

NEWS EXPRESS

Distribution

Software

Hardware

Headline

Event

組み込み機器市場へ進出する Linux

身近なさまざまな製品にマイクロプロセッサが組み込まれていることが、2000年問題の時に話題になった。いまやAV機器や家電製品など、家にある電気製品でCPUが使われていないほうが少ないくらいだろう。

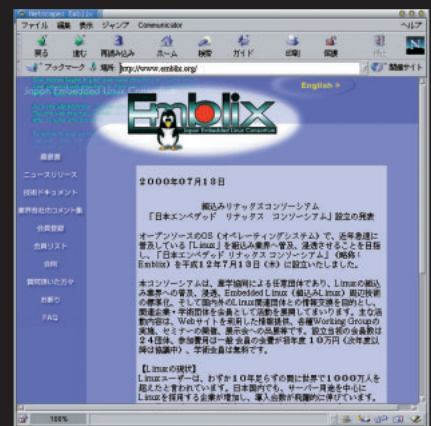
そういった機器は一般に価格が安く、部品コストを下げるため、メモリやI/Oポートを内蔵した1チップCPUが使われている。搭載されているメモリの量もPCなどに比べると非常に小さなものだ。

1チップCPUといっても、8ビットや16ビット、最近では32ビットのものも使われているようで、アーキテクチャの違いだけでなく、移植性を考えれば組み込み用OSとデバッグをサポートする開発環境が必要になる。組み込み用OSとしては、現在は歴史のあるITRONや専用のリアルタイムOSが多く使われている。なぜなら、そういう製品のプログラムはバグがあってはならないため、十分な実績と信頼性が要求されるからだ。

しかし、そういった組み込み用途にも使えるLinuxがすでに製品化されている。リネオ (Lineo) 社のembedixや、Lynux Works社のBlueCat Linuxなど、ほかにリアルタイム処理用のLinuxが多数開発されている。リネオはOpenLinuxを開発しているCaldera Systemsの関連会社である。

そんな中、組み込みLinux (Embedded Linux) のインターフェイスの標準化と普及を目的とした「日本エンベデッド リナックス コンソーシアム」(略称: Emblix) が7月13日に設立された。

ライセンス料が不要で、多くのCPUにすでに移植されているLinuxが、組み込み用途に向いていないはずがない。近い将来、家電製品や携帯電話などにも、組み込みLinuxが使われていくことだろう。



Hardware

発売日

2000年7月6日

Pentium Xeonを最大4基搭載可能なLinuxサーバ
SGI 1450URL <http://www.sgi.co.jp/>

日本SGIは、ハイパフォーマンスLinuxサーバ「SGI 1450」を7月6日より発売した。SGI 1450はSGI 1400の後継機種で、Pentium Xeon 700MHzを最大4基搭載可能な4-Wayサーバである。4U（高さ17.8cm）サイズのラックマウントタイプで、価格は256Mバイトメモリ、9.1Gバイトハードディスクを搭載した最小構成で132万円からとなっている。

メモリは最大8Gバイト（Linuxの場合4Gバイト）ハードディスクは36.4GバイトのSCSIハードディスクを5台搭載することにより最大182Gバイト、64

発売	日本SGI株式会社
TEL	0120-161-086
価格	132万円～

ビット対応を含むPCIスロットを8本備え、Dual Ultra 160 SCSIコントローラを搭載している。

RAIDハードディスク以外にも、電源を3個、ファンを6個と冗長化して、64ビットPCIスロットもホットスワップ可能になっている。

OSは、Red Hat Linux、TurboLinux、Windows NT 4.0 / 2000が対応する。

また、SGI 1450をベースにDB7 UDB、WebSphereを統合した「SGI e-Linux Pack」というパッケージモデルも提供される。



Hardware

発売日

2000年7月下旬

PowerPC搭載LinuxPPC採用の超小型サーバ
OpenBlockSURL <http://www.plathome.co.jp/>

ぶらっとホームは、外形寸法が84（W）×118（D）×52（H）mmと超小型なオープンマイクロサーバ「OpenBlockS」（オープブロックS）を7月下旬より発売した。OpenBlockSは、10/100BASE-TXと10BASE-Tの2ポートのネットワークポートを持ち、CATVやADSLモデムでインターネットに接続する場合に、簡易ファイアウォールやNATアドレス変換、DHCPサーバ機能を利

発売	ぶらっとホーム株式会社
TEL	03-3251-2600
価格	4万9800円

用できる。ネットワークの設定はWebブラウザで容易に行える。

CPUはPowerPC 860T、メモリ16Mバイト、4MバイトのフラッシュROM、シリアル1ポートを搭載している。OSはLinuxPPCを採用し、コンパクトフラッシュ用スロットにメモリカードを装着することで、メールサーバ、Webサーバ、DNSサーバの機能を持たすことができる。



Hardware

発売日

2000年7月25日

TurboLinux ServerをバンドルしたPCサーバ
apricot LinuxインターネットエントリーサーバセットURL <http://www.melco.co.jp/>

三菱電機は、apricotスリムタワーにTurboLinux Server日本語版6.1のインストール代行サービスをバンドルした「Linuxインターネットエントリーサーバセット」を7月25日より発売した。

同セットのapricotスリムタワーは、CPUにCeleron 566MHz、メモリ128Mバイト、10Gバイトハードディスク、24倍速CD-ROM、15インチディスプレイ、キーボード、マウスなどを標準装備

発売	三菱電機株式会社
TEL	03-3218-9064
価格	33万8000円～

している。ネットワークに10/100BASE-TXを2ポート装備していて、プロキシサーバや簡易ルータとして利用することもできる。

サポートサービスとして、OSのアップデート時にアップデートCDが配布（1年間4回）され、オプションでテクニカルサポートをターボリナックスジャパンが直接行う「TurboSupport OEM」サポートパックが用意されている。



Hardware

発売日

2000年7月17日

ラインナップを拡充し、TurboLinux Serverに対応
PRIMERGY LinuxサーババンドルタイプURL <http://www.fujitsu.co.jp/>

富士通は、7月17日よりPCサーバ「PRIMERGY Linuxサーババンドルタイプ」に、ディスクアレイ構成タイプなど4モデル6タイプを追加し、従来のOpenLinux eServer 2.3日本語版に加えTurboLinux Server日本語版6.1に対応した。

追加されたモデルは、ES200、ES210、ES280、MS380で、SCSIハードディスク3台で構成するRAIDディスクアレイのタイプが用意されるのは

発売	富士通株式会社
TEL	03-3216-7976
価格	20万8000円～

ES280とMS380のモデル。

オプションとして、Webやメールサーバ、RAID構築などのインストールサービスを提供し、SupportDesk契約を結ぶことで、Linuxを搭載したプラットフォームの動作保証を行い、インストール時のトラブル、XやApacheなどの設定、運用時のハードウェアとLinuxの障害切り分けに関するQ&Aを、1年間インシデント制限なしで対応する。



Hardware

発売日

2000年7月24日

サーバタイプのウイルスチェッカー搭載Linuxサーバ Xuni-L S Lite

URL <http://www.10art-ni.co.jp/>

テンアートニは、InterScan VirusWallのLinux版をバンドルしたコンパクトなLinuxサーバ「Xuni-L S Lite」(テンユニ-エル エス ライト)を7月24日より発売した。

Xuni-L S Liteは、アクアリウムコンピューターの「white neon」をベースに、メモリとネットワークカードを増設した本体を使用している。CPUはNS GXM233MHz、メモリは128Mバイト、ハードディスクは6.1Gバイト、10/100BASE-TXを2ポート備えていて、OSにはRed Hat Linuxを採用し、InterScan

発売	株式会社テンアートニ
TEL	03-5298-2929
価格	48万円

VirusWall専用にチューニングしている。

InterScan VirusWallは、トレンドマイクロが開発したサーバタイプのウイルスチェッカーで、インターネットゲートウェイ上で動作させることで、SMTP、HTTP、FTPトラフィックをリアルタイムに監視し、インターネットからのメールやダウンロードファイルに混入したウイルスの発見/除去を行う。

Xuni-L S Liteの価格は、5インシデントまでの電話によるサポートと、InterScan VirusWall 30ユーザー分のライセンスを含めて48万円となっている。



Software

発売日

2000年9月8日

高い変換効率を誇るLinux対応日本語変換ソフト ATOK X for Linux

URL <http://www.justsystem.co.jp/>

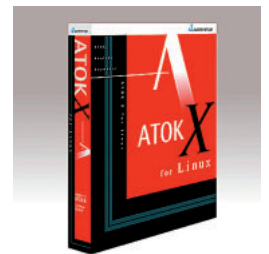
ジャストシステムは、日本語変換ソフトATOKのLinux対応版「ATOK X for Linux」を、9月8日より発売する。昨年7月より「ATOK12 SE for Linux」をLinuxディストリビューションにバンドルしていたが、今回、Windows版やMacintosh版のATOKとほぼ同等の機能を備えた単体パッケージとして販売する。

ATOK X for Linuxは、LinuxのX Window System上で動作し、ユーザーインターフェイスにGTKツール

発売	株式会社ジャストシステム
TEL	03-5412-3939
価格	9800円

キットを利用して使いやすさを実現した。文字パレット、辞書ユーティリティなどの入力支援ツールも用意されている。

動作確認済みのディストリビューションは、LASER5 Linux 6.2、OpenLinux 2.3日本語版、Red Hat Linux 6.2J、TurboLinux Workstation日本語版6.0で、そのほかのディストリビューションも動作確認作業を行っている。



Software

発売日

2000年7月4日

バーチャルマシン機能で複数OSを同時に稼働させる VMware 2.0 for Linux

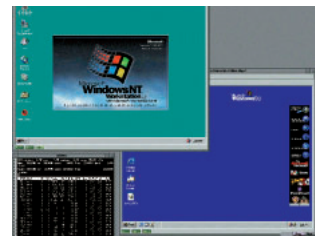
URL <http://www.netserve.co.jp/>

ネットサーブは、1台のPC上で複数OSを同時に稼働させるバーチャルマシン機能を提供するソフトウェア「VMware」を、7月4日から発売した。ソフトウェアをインターネットからダウンロードし、ライセンスキーを購入するダウンロード版が3万9500円。CD-ROM付きのパッケージ版製品が4万8000円で、30日間のサポートが付属している。

発売	株式会社ネットサーブ
TEL	03-3498-1814
価格	3万9500円(ダウンロード版)

ホストOSをLinuxとする「VMware 2.0 for Linux」と、Windows NT / 2000とする「VMware 2.0 for Windows NT and Windows 2000」の2種類があり、どちらも同じ価格だ。

VMwareを使ってLinux上でWindowsなどの他OSを動作させ、Linuxにはないアプリケーションなどを利用することができる。



第1回TurboLinuxソフトウェアコンテスト

PENGUINS@Work!

ターボリナックス ジャパンは、第1回TurboLinuxソフトウェアコンテスト「PENGUINS@Work!」(ペンギン・アット・ワーク)を実施する。

TurboLinuxで稼働するソフトウェアであれば、オープンソースソフトに限らず応募でき、発表/未発表も問わないので、腕に自身のある人はどんどん応募してデッカいクリスマスプレゼントをゲットしよう。

コンテストへのエントリーを8月31日までにしようと、もちろん特製Tシャツがもらえるのだ。ソフトウェア作品の締め切りは10月31日で、本誌も参加する審査結果は12月25日に発

表される。ソフトウェアは団体部門と個人部門の2部門に分けて審査され、それぞれ最優秀賞には「LinuxWorld/New Yorkにペアでご招待」+副賞がもらえる!

なお、最優秀賞作品は、ターボリナックスがワールドワイドに提供するOS製品にバンドルするために、各国語対応を含めて開発をバックアップする。詳細は、同社のWebサイトで。

ターボリナックス ジャパン
<http://www.turbolinux.co.jp/>





Linuxカーネル2.4.0-test5リリース

2000年7月28日

Linuxカーネル開発版の最新バージョン「2.4.0-test5」が7月27日、リリースされた。このバージョンでは、FATファイルシステムのファイル名をEUCやシフトJISに変換する機能が追加された。これによって、日本語ファイル名を含むWindowsのファイルシステムを、Linuxで扱うことが容易になる。

Linuxは「NLS」(National Language Support)という機能を持っている。NLSとは、他のOSのファイルシステムで使われるファイル名の文字コードを、任意の文字コードに変換する仕組みだ(変換されるのはファイル名のみで、ファイルの内容は変わらない)。今回追加されたEUC/シフトJISのサポートはこの機能の一部で、カーネル再構築の際に「NLS_CODEPAGE_932」を有効にすることにより利用可能になる。実際の利用には、mountコマンドのオプションに「-o codepage=932,ioccharset=euc-jp」などと指定すればよい。

Linuxカーネルの開発体制は、マイナーバージョン(バージョンの2けた目の数字。たとえば2.4.0なら「4」)が偶数の「安定版」と、奇数の「開発版」の2つに分かれている。新機能は主に開発版に追加される。安定版は実運用のために安定性を重視している。現在、最新の安定版である「2.2.16」にNLS_CODEPAGE_932のサポートは存在しないが、「2.2.17-pre9」にはサポートが取り込まれており、安定版カーネルでも早い時期に同機能が利用可能になるものと思われる。

Linuxカーネル2.4.0-test5のダウンロード

(<ftp://ftp.jp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.0-test5.tar.bz2>)

ジャストシステム、「Muffin」のTechnology Preview版を公開

2000年7月27日

ジャストシステムは7月26日、プレゼンテーションツール「Muffin」Technology Preview版の提供を開始した。Muffinは、「一太郎Ark for Java」に続くJavaで記述されたアプリケーションで、開発中であることが6月29日にアナウンスされていた。

Muffinはアウトラインプロセッサベースのプレゼンテーションツール。画面は2ペイン構成で、左側に表示されるアウトライン編集画面で文章を編集、それにスタイル情報を付加したものが右側のスライド編集画面に表示されるという構成になっている。

Muffinが出力するファイルの形式は、XHTMLとCSS2、JavaScriptのいずれかなので、Webブラウザで開いてプレゼンを行うことが可能だ。

今回のTechnology Preview版で作成したデータは、Internet Explorer 5.0ではタイトルのフォントが大きくなりすぎて正常に表示できないなどの問題点が残っている。これらが解決されれば、標準的なフォーマットでファイルを保存して、プレゼンテーション資料をそのままWebで公開可能というのは大きな利点だろう。

ジャストシステムでは、今回のTechnology Preview版を使用したユーザーの意見・望をWebで求めている。

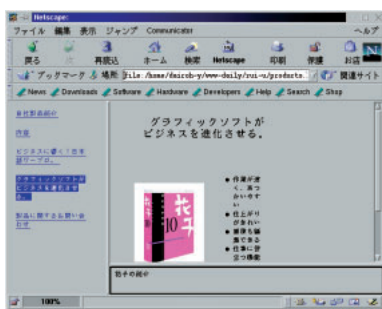
なお、ジャストシステムは、Muffinのほかに、「Choco」という表計算ソフトウェアをJavaで開発中だ。

ジャストシステム

(<http://www.justsystem.co.jp/>)

Muffin Technology Preview版のダウンロード

(<http://www.justsystem.co.jp/ark/muffin/tp/muffintp1.html?akt>)



YDCがミラクル・リナックスと提携

2000年7月25日

横河電機グループのワイ・ディー・シー(以下YDC)とミラクル・リナックスは7月25日、全面的な業務提携を発表した。

これによりYDCは、システム構築やプロダクトのOSとして、ミラクル・リナックスのディストリビューション「Miracle Linux」を標準採用する。また、Miracle Linuxのコンサルティングやエンジニアリング事業も推進していき、24時間365日のフロントサポートサービスを行う。一方のミラクル・リナックスは、営業・技術の両面からYDCを支援していく。

ミラクル・リナックスは日本オラクルの子会社で、LinuxとRDBMSに特化した事業を行なうために2000年6月に設立された。同社は現在、Oracle 8iに最適化されたディストリビューション「Miracle Linux」を、ターボリナックス ジャパンなどの協力を得て開発している。

YDCは、Miracle Linuxを中心としたLinux関連ビジネスで、初年度に15億円、3年後に40億円の売り上げを目指すという。

ワイ・ディー・シー

(<http://www.ydc.co.jp/>)

ミラクル・リナックス

(<http://www.miraclelinux.com/>)

StarOfficeがオープンソースに

2000年7月20日

Sun Microsystemsが無償配布していたマルチプラットフォームのオフィスソフトウェア「StarOffice」(日本語版は商標の問題により「Sun Office」となる予定)が、オープンソースとなることが決定した。今後は、StarOffice用のWebサイトOpenOffice.orgにより、管理/公開されていく予定。

ライセンスには、GNU General Public License (GPL) とSun Industry Standard Source License (SISSL) の2つが適用される。これは、GPLが場合によっては企業活動に(ソースコード公開義務などにより)馴染まないことを考慮したためだという。

ソースコードはCVSで管理され、誰でも

もOpenOffice.orgから閲覧することが可能になる。今後、OpenOffice.orgは、XMLによるオフィスファイルフォーマットや、API仕様などをIETFやW3Cに提出するなどして、オフィス製品の標準仕様を作りあげていくとしている。

OpenOffice.org
(<http://www.openoffice.org/>)

Red Hat、クラスタ向け ディストリビューションをリリース

2000年7月11日

米Red Hatは7月10日、「Red Hat Linux」とクラスタソフトウェア「Piranha」を組み合わせたクラスタ向けディストリビューション「Red Hat High Availability Server 1.0」を発表した。同製品は、複数のサーバに負荷を分散することによって、1台がダウンしても全体としてはサービスを継続することができ、サーバの信頼性向上を実現することが可能にするもの。

クラスタは最小で2台から構成することができるが、ロードバランシング（負荷分散）を行うシステムでは3台以上が必要。その場合、外部のネットワークから見えるのは、トラフィックを分配する「ロードバランサ」とその待機系のみで、内側に置かれた残りのノードは、ロードバランサから転送されるリクエストを処理する。一部のノードがダウンしても、ロードバランサがそれを検出して他にトラフィックを転送するため、信頼性が向上する。

価格は1995USドル。電話およびWebによるインストール・設定のサポート（1年間）が付属する。

米Red Hat (<http://www.redhat.com/>)

Mozilla M16対応の日本語パック 正式リリース

2000年7月10日

7月7日、Mozillaを日本語化する「Mozilla日本語パックM16正式版」が公開された。6月13日にリリースされた最新版のMozilla M16（Milestone 16）に対応しており、これを適用することで、デフォルトでは英語でしか表示されないダイアログボックスのメッセージやメニューを日本語化することができる。なお、オリ

ジナルのMozillaでも日本語のWebページは問題なく表示できる。

また、日本語パック公開とほぼ同時に、Mozilla.gr.jpサイトが公開された。このサイトは、これまで有志によってばらばらに行われてきた活動をまとめ、日本のユーザーから利用しやすくすることを目標にしているという。現在、日本語ドキュメントやリンク集などが用意されている。Mozillaメーリングリストや、日本語版Bugzillaは移行準備中となっており、これからコンテンツが増えていく予定だ。（メーリングリストの運用は7月29日、日本語版Bugzillaの移転は、7月25日に、それぞれアナウンスされた。）

Mozilla日本語パックのダウンロード
(<http://www.mozilla.gr.jp/jlp/>)



Samba 2.0.7 日本語版リリース

2000年7月8日

日本Sambaユーザ会は、「Samba 2.0.7 日本語版（以下2.0.7 日本語版）」をリリースした。Sambaは、UNIX系OS上でWindowsネットワーク（SMB:Server Message Blockを使用したネットワーク）を実現するソフトウェア。Sambaを使用すると、Windowsのファイル共有やプリンタ共有などを、UNIX系OS上から提供することができる。

2.0.7 日本語版では、以下の3点が大きく改善されている。（1）Windows 2000への対応：Windows 2000で起きる不具合を改善。日本Sambaユーザ会は、Windows 2000を含んだネットワークでSambaを使用している場合は、2.0.7 日本語版にバージョンアップすることを推

奨している。（2）SWAT（Samba Web Administration Tool）の国際化版を搭載：SWATは、WebブラウザからSambaの設定を行うことができるツール。2.0.7 日本語版では、Webブラウザの言語設定に合わせて自動的にWebページの表示言語を切り替えることが可能になった。（3）機種依存文字サポート：ファイル名などにおける日本語固有の問題をほぼ解決。日本Sambaユーザ会は、Samba 2.0.5a日本語版から2.0.7 日本語版へのバージョンアップにおいて重要な注意点があるので、ドキュメントを確認してほしいと呼びかけている。注意点とは、おもに文字コードとsmb.confの取り扱いについてであり、2.0.7 日本語版に含まれる「RELNOTE.ja」ファイルなどに詳細が記述されている。

7月8日正午現在、2.0.7 日本語版はソースコードのみの公開で、アーカイブファイルのサイズは約4.6Mバイト。

日本Sambaユーザ会
(<http://www.samba.gr.jp/project/samba-ja/>)

IBMのPC「Desktop On-Call」が Linuxをサポート

2000年7月6日

日本アイ・ビー・エムは、遠隔地にあるマシンをWebブラウザからリモート操作できるソフトウェア「Desktop On-Call V4.0」を7月5日に発表した。今回のバージョンアップでは、操作対象となるプラットフォームとして「Red Hat Linux 6.2J」と「TurboLinux Workstation 日本語版6.0」、「Caldera OpenLinux eServer 2.3 日本語版」を新たにサポートした。

上位製品となるProfessional版には、LinuxのほかにAIX 4.3.3、Solaris 7 / 8のサポートも追加されている。

Desktop On-Callは、マシンの画面をネットワーク経由で転送することによって、手元のWebブラウザから自由に操作できるソフトウェアだ。遠隔地のマシンにDesktop On-Callをインストールしておけば、画面を表示するマシンでは、Javaアプレットが動作するだけなので、Java対応のWebブラウザがあればどこからでも操作できるのが特徴だ。

「V4.0」は新たに、「グリーティング・

テキスト・チャット機能」をサポートしている。これは、遠隔操作のセッションを開始する前に、事前予告としてテキストメッセージを送ることを可能にする機能だ。ヘルプデスク業務などに対する対応を強化したものだという。この機能はオフ設定にすることも可能。

価格は、Desktop On-Call V4.0「PCリモコン」が1万1800円、Desktop On-Call V4.0「PCリモコン」バージョンアップが6800円、Desktop On-Call V4.0 Professionalが2万3600円。

なお、同様の機能を実現するソフトウェアとしては、AT & T Laboratories Cambridgeが開発したフリーソフトウェア（GPL）「VNC」などが存在する。VNCはX Window System、Windows、MacOSに対応しており、専用のクライアントかJavaアプレットにより操作を行うことが可能。また画面全体を転送するわけではないが、遠隔地にXクライアントが、操作する側にXサーバがインストールされていれば、ネットワーク透過なXプロトコルを用いて遠隔操作することもできる。

プレスリリース

(<http://www.jp.ibm.com/NewsDB.nsf/2000/07053>)

米IBM、IBM Java2 SDK 1.3 Linux版をリリース

2000年7月4日

米IBMは、「IBM Developer Kit for Linux, Java 2 Technology Edition, Version 1.3」をリリースした。

同製品は、IBMの開発したJVMや、JITコンパイラが含まれ、米Sun MicrosystemsのJava互換性試験をパスしているもの。前バージョンの1.1.8は、SunのJDKに比べてパフォーマンスが高かったことで知られている。

現在、同社のWebサイトでは以下のようなパッケージが配布されている。(1) Developer Kit package : Javaアプリケーションやアプレットを開発するために必要な、Javaコンパイラ、Javaクラスライブラリ、Javaアプレットビューア、Javaインタプリタや、JNI (Java Native Interface) のためのライブラリとヘッダ

ファイル、デモアプレットなどが含まれる。ファイルサイズは28Mバイト。(2) Runtime Environment package : Javaアプリケーションを実行するために必要なJavaインタプリタなどが含まれる。ファイルサイズは16.4Mバイト。(3) Java Authentication and Authorization Service (JAAS) package : Java 2で提供されているセキュリティ機能を拡張するAPIを含む。ファイルサイズは289Kバイト。(4) Java Communications application programming interface package : JavaアプリケーションからシリアルポートやパラレルポートにアクセスするAPIを含む。ファイルサイズは42Kバイト。

なお、Java2 SDK 1.3の動作には、カーネル2.2.x、glibc 2.1以降、XFree86 3.3.x以降が必要となる。

米IBM (<http://www.ibm.com/>)

「XFree86 4.0.1」リリース

2000年7月3日

XFree86 Projectは、「XFree86 4.0.1」をリリースした。XFree86 4.0のバグフィックスとドライバの追加が主な内容となっている。

「XFree86 4.0」からの主な変更点は以下の4点。(1) 新たに、「Intel i810」、「Matrox G400 / G200 (AGP対応カードのみ)」、「ATI Rage 128」のDRI (Direct Rendering Infrastructure) 用ドライバを追加。(2) GUIでの設定ユーティリティ「xf86cfg」を追加。(3) Sun SPARCプラットフォーム上のLinuxをサポート。(4) Xlibの国際化関連のバグフィックス。

7月27日現在、x86用Linux (glibc 2.1、2.0、libc5)、Alpha用Linux、FreeBSD 3.x、NetBSD 1.3、OpenBSD 2.7などのプラットフォームに対応したバイナリがリリースされている。

ダウンロードは、XFree86 ProjectのFTPサイトやミラーサイトから可能。

(今月号の付録CD-ROMにも、ソースおよびglibc 2.1、2.0を使用したLinux用のバイナリを収録している。)

XFree86 Project

(<http://www.xfree86.org/>)

MySQL、ライセンスをGPLに変更

2000年6月29日

6月28日、リレーショナルデータベース「MySQL」のライセンスがGPLに変更された。これまでのMySQLのライセンスでは、商用利用や改変についての制限が大きく、ライセンス料を払わなければならない場合もあったが、今回のライセンス変更によりその必要がなくなる。

ライセンス変更の対象となるのは、最新のベータバージョン「3.23.19」から。以前より、MySQLの古いバージョンについてはGPLでリリースされていたが、今回は全面的にGPLを採用したのが大きな変化だ。

同時に、米VA Linux社がMySQLに出資することも発表された。また、VA Linuxは、同社の運営する「SourceForge.net」のプロジェクトのひとつとしてMySQLをホスティングしていく。SourceForge.netは、高速なFTPサーバやCVSリポジトリ・Webサーバなどをオープンソース開発者に対して提供しているオープンソース支援サイト。

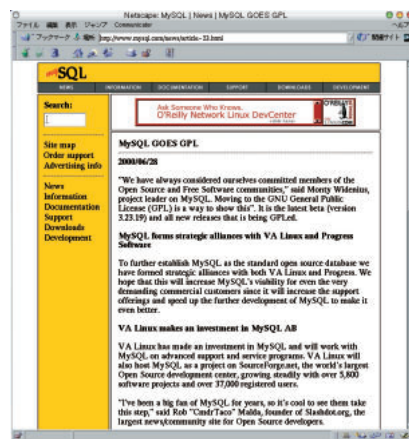
VA Linuxが運営する有名なオープンソース系のサイト「Slashdot」、「Freshmeat」、「Linux.com」、「SourceForge」はすべて、MySQLを利用しているという。

プレスリリース

(<http://www.mysql.com/news/article-23.html>)

MySQL

(<http://www.mysql.com/>)



Distribution

新着ディストリビューション

TurboLinux Server 日本語版 6.1

安定した人気を誇るTurboLinuxのサーバ版がより強力になった。大容量メモリ、Raw I/Oをサポートし、データベースサーバなどで威力を発揮する。メンテナンスCDを1年間4枚無償提供するなど、サポート体制も強化。Linuxサーバを業務に使う、まさにプロフェッショナルユースのディストリビューションだ。

Slackware Linux 7.1

玄人好みの伝統的ディストリビューションがバージョンアップ。収録アプリケーションも大幅に増え、バイナリディスクが2枚構成に。安定バージョンのアプリケーションを中心としたベーシックシステムはもちろん、最新バージョンのコンポーネントでシステムを構成することも可能だ。古くさいなんてもういわせない!

Vine Linux 2.0 for PPC

安定した日本語環境で人気を博しているVine Linuxの、PowerMacintosh版が登場した。x86版Vine Linuxゆずりの特長と、Mac-On-LinuxなどPowerPC Linux独自のアプリケーションが融合した新しいディストリビューションだ。葡萄味の林檎を味わってみよう!

TurboLinux Server 日本語版 6.1

ターボリナックス ジャパンは、6月16日にサーバ用途向けLinuxディストリビューション、TurboLinux Server 日本語版6.1を発売した。価格は3万9800円、アカデミック版の価格は2万4800円だ。なお、TurboLinux Server 日本語版の旧バージョンユーザーは優待価格の2万4800円で購入できる。

本製品は、Linuxによるサーバ構築に特化したディストリビューションで、セキュリティの強化とデータベース運用に向けたチューニングが特徴となっている。

カーネルは2.2.15を拡張したもので、4Gバイトまでのメモリを利用可能。Unified IDEパッチも適用されているので、より多くのIDEコントローラに対応する。標準Cライブラリはglibc 2.1.3、XFree86は3.3.6を採用している。標準のデスクトップ環境は、Sawfish (旧Sawmill) + GNOMEだ。

企業での導入時に重要となるサポー

トは、90日間3インシデントまでのインストールおよびインターネット系サーバ設定サポート、重要なセキュリティホールが発見された場合にサーバ管理者へメールで通知するサービスが無償で提供される。さらに特筆すべきは、1年間4回のメンテナンスCDの無償送付だ。メンテナンスCDには、アップデートモジュールが収録され、TurboLinuxのパッケージインストールツールTurbopkgによるシステムの更新が可能。このほか、サーバ構築から運用までをカバーする有償サポートも用意されている。



サーバに特化したシステムチューニング

通常のLinuxカーネル2.2は、2Gバイトまでのメモリをサポートするが、TurboLinux Server 日本語版6.1では4Gバイトまでのメモリを管理できるよう強化している。これにより、大容量のメモリを必要とするWebサーバやデータベースサーバの構築も可能になる。

また、データベースサーバの信頼性やパフォーマンスの向上に欠かせないRaw I/Oをサポート。さらに、Raw I/Oアクセス時のAsynchronous I/Oサポートにより、ディスクアクセス速度も向上させた。TurboLinux Server 日本語版6.1は、Oracle 8i Workgroup

Server (8.1.5) の対応OSとして認定されている。

このほか、インストーラも一新され、システム管理者の用意する設定ファイルに従い、同じ構成で複数のサーバを構築することができるようになった。



サーバ版ならではのバンドルソフト

バンドルされる主な商用ソフトウェアは表1のとおりだ。サーバ運用には欠かせないデータバックアップソフトウェアや、システム管理の手間を軽減するソフトウェアが中心になっているのわかるだろう。

データバックアップソフトのBRUは、コマンドライン版と、Xで動作するGUI版 (画面1) の2つがインストールされる。コマンドライン版は、tarによく似た構文で利用でき、シェルスクリプトなどから呼び出して使うのに便利だ。一方、GUI版は、対話的にバックアップを作成したり、データをリストアするのに向いている。テープドライブはもちろん、アーカイブファイルにバックアップを作成することができるので、システム用とは別のハードディスクをバックアップ専用にするのもよいだろう。

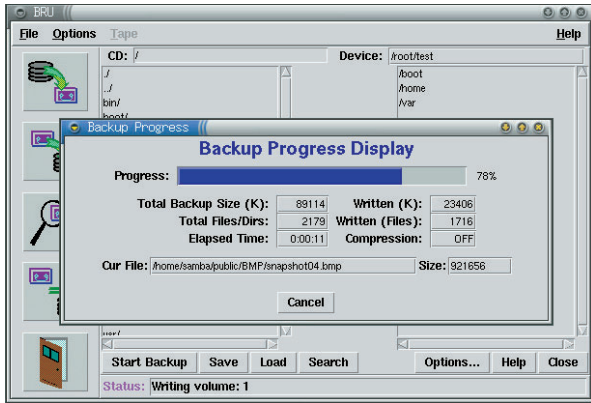
HDE Linux Controller 1.0 (画面2) は、各種サーバデーモンの起動 / 停止



製品名 TurboLinux Server 日本語版6.1
 価格 3万9800円
 問い合わせ先 ターボリナックス ジャパン株式会社
 03-5766-1660
<http://www.turbolinux.co.jp/>

製品名	概要
BRU	データバックアップソフトウェア
HDE Linux Controller 1.0	Webベースシステム管理ツール
RSA SSL ライブラリ	128ビットインターネットセキュリティコンポーネント
Norton Ghost	ディスクイメージバックアップツール
RYOBI フォント	TrueType フォント5書体
EnlightenDSM	システム管理スイート

表1 収録される主な商用アプリケーション
 メンテナンスCDでHDE Linux Controller 2.0アップデートプログラムを無償配布予定



画面1 ディスクバックアップソフトウェアBRU GUI版を使えば、必要なディレクトリだけバックアップするのも簡単だ。

画面2 HDE Linux Controller 1.0 サーバソフトウェアの設定だけでなく、バックアップやメーリングリストの管理機能なども持っている。



や、ユーザー管理、ログ管理など、数多くのサーバ管理をWebブラウザから行うツールだ。サーバは、ひとたび設定が済み、実稼働を始めるとオフィスの隅などに設置されることが多いため、リモートメンテナンスができると嬉しい管理者も多いだろう。

なお、TurboLinux Server 日本語版6.1の登録ユーザーには、HDE Linux Controller 2.0が収録されたメンテナンスCDが無償提供される予定だ。

ディスクイメージバックアップツールのNorton Ghostは、ハードディスクやパーティション単位でバックアップを作成するソフトウェアだ(画面3)。ディスク/パーティションの複製を作ることにも可能だし、イメージファイルを作成することもできる。容量を指定して複数のイメージファイルに分割し、CD-Rに焼くことで、リカバリCD-

ROMを簡単に作成できる。Norton Ghostは、DOSで動作し、ブート可能なDOSシステム入りのフロッピーディスクで供給される。このため、インストール作業が不要なのは嬉しいが、イメージファイルをFATパーティションにしか作成できないので注意してほしい。



安全指向のセキュリティアプローチ

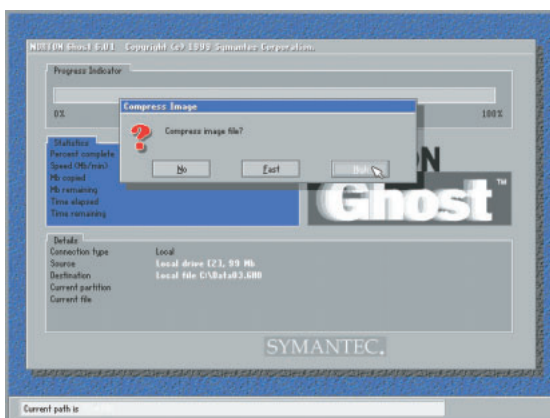
サーバ運用で重要なことのひとつがセキュリティ対策だ。TurboLinux Server 日本語版6.1では、「Deny All」のセキュリティアプローチを採用している。すなわち、初期設定では、すべてのサーバサービスが利用禁止となっており、必要なサービスについては、セキュリティポリシーに合わせて管理者が利用可能にしていくという形態になる。

インターネットスーパーサーバinetdから起動されるサーバプログラムの利用可否は、TurboPortCfgプログラムで設定できる(画面4)。また、このツールでtcp_wrappersによるアクセス制限も設定可能だ。

高いセキュリティレベルを確保するには、日ごろの情報収集が欠かせない。年4回送付されるメンテナンスCDや、ターボリナックス ジャパンのWebサイトで配布される最新モジュールを導入して、セキュリティ対策は怠らないようにしよう。

付録CD-ROMに関するご注意

本誌付録CD-ROMに収録されているのは、TurboLinux Server 日本語版6.1のFTP版です。商用ソフトウェアは含まれていません。また、ターボリナックス ジャパンからサポートを受けることもできません。



画面3 Norton Ghost フロッピーディスクから起動して使う。CD-Rにイメージを焼いて自分だけのリカバリCD-ROMも作ることが可能だ。



画面4 TurboPortCfg inetdから起動されるサーバの利用可否を設定できる。初期状態ではすべて無効(利用不可)。

Slackware Linux 7.1

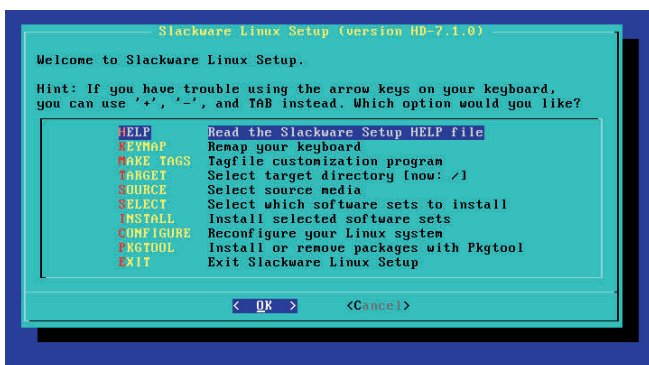
The Slackware Linux Projectは、6月24日にSlackware Linux 7.1を発表した。The Slackware Linux ProjectのFTPサイト(ftp://ftp.slackware.com/)などから入手できるほか、Walnut Creek CDRomから4枚組のCD-ROMが39.95USドルで販売されている。日本国内では、ぶらっとホーム(http://www.plathome.co.jp/)で取り扱っており、Webの通信販売では3376円で購入できる。編集部では、同社店頭において3480円で購入した。

Slackware Linuxは、老舗のディストリビューションで、パワーユーザーを中心に、根強い人気を保っている。

1999年10月に発表されたSlackware Linux 7.0までは、バイナリディスク1枚にすべてのプログラムが収録されていたが、7.1では収録しているプログラムが増加したため、各種プログラムの実行ファイルは、バイナリディスクとContribディスクに分けられている。

最新のシステムコンポーネント

Slackware Linuxは、7.0からカーネル2.2 + glibc2というシステムコンポー



画面1 Slackware Linuxのインストーラ
インストール後の設定変更にも利用できる。

ネットを採用している。今回発表された7.1では、安定版カーネルの最新版2.2.16、標準Cライブラリglibc 2.1.3を採用している。このカーネルには、Alan Coxによるバグフィックスパッチも適用されている。

X Window Systemは、XFree86 3.3.6がインストールされる。ContribディスクにはXFree86 4.0も収録されているので、あとから入れ換えることも可能だ。Contribディスクの内容は、FTPサイトでも公開されているので、必要なパッケージだけを選んで入手してもよいだろう。

インストーラは、Slackwareの伝統通り、テキストベースのメニュー形式だ(画面1)。メニューの項目を上から順に選ぶことで、パッケージのインストール、ネットワーク設定、X Window Systemの設定を行うことができる。インストールが終了したあとも、rootユーザーになって、setupコマンドを実行することで、このインストーラを起動して各種設定を変更できる。

Slackware Linuxは、RPMやdebなどのようなパッケージ管理システムを持たず、tar + gzipでまとめられたバイナリをインストールするシンプルな方式を採用している。このため、インストールするソフトウェアの選択やバージョン管理はすべてシステム管理者にゆだねられる。初心者には取っつきにくいかもしれないが、システムに精通した熟練管理者にとってはかえってありがたいだろう。



充実したソフトウェア群

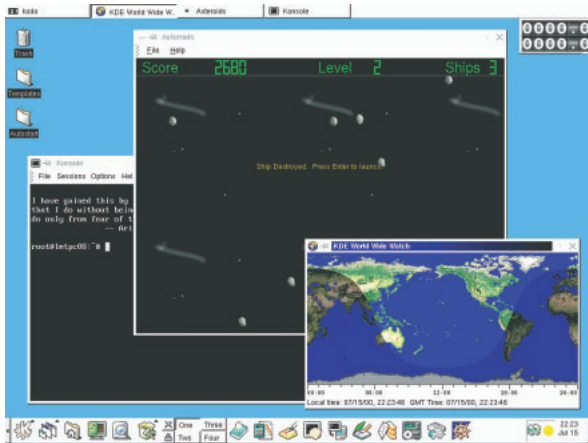
基本システムと、標準的なアプリケーションプログラムはバイナリディスクに収められているので、通常の利用にはこれだけでも十分だろう。バイナリディスクに収められているプログラムをすべてインストールすると、およそ1Gバイトのディスク容量が必要になる。

X Window Systemのデスクトップには、Sawfish 0.28 + GNOME 1.2.1(画面2)、KDE 1.1.2(画面3)の環境

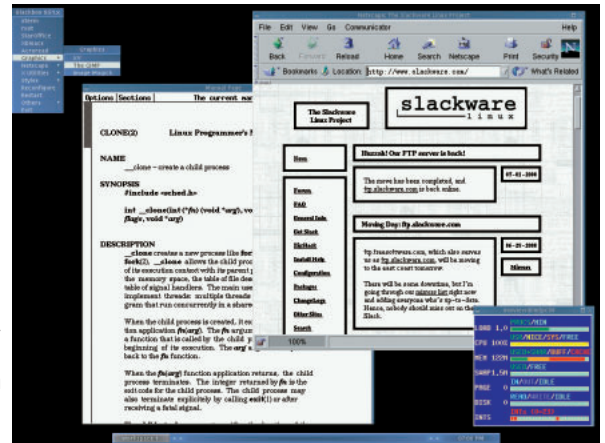


画面2 GNOME 1.2.2を採用

各種リソースメータなど、アプレットとして動作するソフトウェアが多数追加されている。



画面3 KDE 1.1.2のデスクトップ
安定版のKDE 1.1.2を選ぶことも可能だ。



画面4 ウィンドウマネージャ Blackbox
ゴテゴテした画面はキラリだという方にイチ押し！ 軽くてイカしたデザイン。

が用意されている。このほか、ウィンドウマネージャとして、Enlightenment、Window Maker、fvwm2などを選ぶこともできる。Sawfish + GNOMEというデスクトップ環境は、最近のディストリビューションの多くで採用されているが、GNOME 1.2をいち早く採用しているのが特徴的だ。さらに、Contribディスクには、最近話題のウィンドウマネージャBlackbox（画面4）や、KDE 2.0に向けた版の1.90 / 1.91（画面5）も収録されている。Contribディスクには、このほかにもgcc 2.95.2、AfterStep 1.8.0などの最新ソフトウェアのほか、XEmacs、openmotifなども含まれているので、先進的なユーザーにとっては興味深いところだ。
バイナリディスクのkernelsディレク

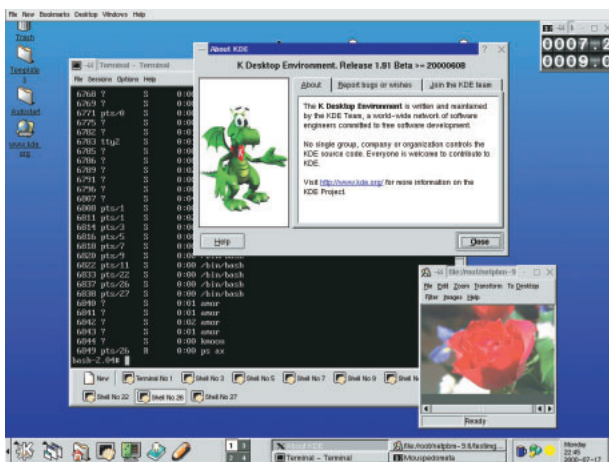
トリには、さまざまなSCSIカードなどに対応するカーネルイメージが多数用意されているので、環境に合わせてインストールするとよいだろう。
この中には、UltraDMA/66モードにも対応するUnified IDEパッチをあてたカーネルも用意されている。編集部では、ALI Aladdin VチップセットのSocket 7マザーボードのマシンにインストールをしたが、標準のカーネル2.2ではこのチップセット内蔵のIDEコントローラでDMAモードを使うことができない。そこで、Unified IDEパッチをあてたカーネルに入れ換えてみたところ、めでたくUltraDMA/33モードで使えるようになった。hdparmコマンドでデータ転送速度を測ってみると、標準カーネルでは6.66 Mバイト /

秒だったのが、17.98 Mバイト / 秒まで向上した。



Linuxマスターに最適

古典的なディストリビューションといわれてきたSlackware Linuxだが、今度のバージョンアップはとてもアグレッシブだ。以前からの特徴である見通しのよいシステム構成はそのままに、最新のソフトウェア環境も提供する。
自力で日本語化しようというパワーユーザーや、日本語環境はなくとも最新のソフトウェアに触れたいという方にお勧めしたい。



画面5 KDE 1.91のデスクトップ
次期バージョンのKDE 2.0環境を先取り。端末エミュレータkonsoleは、複数の画面を切り替えて使えるようになった。



製品名 Slackware Linux 7.1
価格 39.95 USドル
問い合わせ先 Walnut Creek CDROM
<http://www.cdrom.com/>

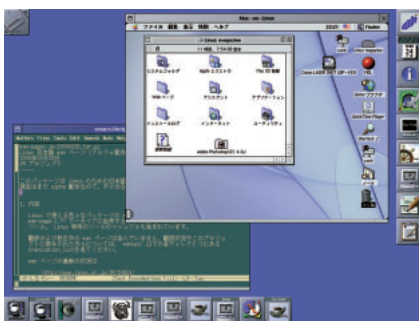
Vine Linux 2.0 for PowerMac

葡萄味の林檎登場！？

Vine Linux 2.0 for PowerMac (以下Vine/PPC)は、Project Vineが開発したVine LinuxのPowerPC対応版だ。PowerPCを搭載し、PCIバスを持ったMacintoshに対応したディストリビューションである(iMac、iBookはPCIバスを持たないが動作可能)。

Vine/PPCは、既存のPowerPC系ディストリビューションの成果をベースに、x86版Vine Linuxの「使いやすい日本語環境を提供する」というコンセプトで作られている。さらに、Mac-On-LinuxなどPowerPC版ならではのソフトウェアを収録している。

現在のところ、Vine/PPCはx86版のVine Linuxと違って、市販はされていない。FTPサイト(<http://www.ring.gr.jp/archives/linux/Vine/Vine->



画面1 PowerPC用でもVine Linuxなので、デフォルトのウィンドウマネージャはもちろんWindow Makerだ。Mac-On-Linuxのウィンドウ以外は、x86版Vine Linuxと見分けがつかない。

2.0/ISO/VinePPC-2.0.image など)からCD-ROMイメージを入手して、自分でCD-Rに焼く必要がある。

現行のPowerPC用のディストリビューションは、カーネル2.2.15を採用している例が多いが、Vine/PPCはx86版Vine Linuxとの共通化のため、2.2.14が用いられている(表1)。またCライブラリは2.1.3、XFree86は3.3.6が採用されている。デスクトップ環境は、GNOME 1.0.55、KDE 1.1.2が収録されている(標準でインストールされるのはGNOMEのみ)。だが、デフォルトのウィンドウマネージャは、Vine Linuxの慣習に基づいて、Window Makerとなっている。

日本語入力システムは、Canna(3.5b2)とFreeWnn(1.1.1)が収録されており、自分に合ったものを選べるようになっている。

PowerPC系ディストリビューションには、MacOSと共存するためのブートローダが付属している。もちろんVine/PPCにも、BootX、yabootの2種類が収録されており、機種や用途によって使い分けられるようになっている。

BootXは、MacOSと共存させることを前提にしたブートローダで、MacOSの起動時にダイアログからMacOS/Linuxを選択して起動できる。

それに対してyabootは、iMac以降のデスクトップ/PowerBook用に搭載さ

れているOpen Firmwareを利用してLinuxを起動するもので、LinuxとMacOSと共存させることも、Linux専用Macとして設定することも可能だ。

PPC系LinuxとVineのいいとこどり

編集部では、384Mバイトのメモリを搭載したiMac DV Special Edition(PowerPC G3 400 MHz)にVine/PPCをインストールして、評価を行った。

PowerPC系Linuxで問題になるのが、画面描画速度だ。ところが今回のVine/PPCでは、以前ほど遅さが気にならなかった。使用している機種も違うので単純に比較はできないが、ドライバの熟成度も上がってきていると思われる。また、デフォルトで、このように最適な設定が得られるのは、初心者だけでなく使い慣れたユーザーにとってもありがたいことだ。

PowerPCならではのアプリケーションといえば、なんといってもMac-On-Linuxだろう(画面1)。これは、PowerPC用Linux上で動作するPowerMacintoshのエミュレータだ。異なるCPU間のエミュレーションを行うわけではないので、動作は比較的速い。また、ネットワークやサウンドなどの機能もそのまま利用できる。編集部で試した際にも、特に設定することもなく社内のLANに接続し、リモートのファイルにアクセスすることができた。

iMacにVine/PPCをインストールして、世界でもっともスタイリッシュなLinuxマシンを使うのもいいかもしれない。

パッケージ名	Vine/PPC	Vine Linux 2.0CR
kernel	2.2.14-19v132	2.14-1v16
glibc	2.1.3-4v11	2.1.2-17v12
XFree86	3.3.6-13v18	3.3.6-13v13
gnome-core	1.0.55-1v13	1.0.55-1v13
kdebase	1.1.2_jp_19991029-4	1.1.2_jp_19991029-2

表1 主要コンポーネントのバージョンVine/PPCは、x86版Vine Linuxよりも後にリリースされたため、一部のパッケージのバージョンやリリースナンバーが新しくなっている。

Distribution ▶▶▶

特報

Project Vine、レッドハットとの協力関係を強化

Project Vineは7月4日、Vine Linuxのサポート体制を改善するため、レッドハットとの協力関係を強化し、同時に商用版Vine Linuxの販売元を技術評論社からレッドハットに変更することを発表した。

Project Vineによれば、Vine Linuxリリース当初の予想よりも企業ユーザーからのサポートの要望が多く、企業向けサポート体制を構築する必要が生じたため、今回の関係強化にいたったとのことだ。

商用版Vine Linuxは、次期バージョンである2.1以降、レッドハットから発売される。また、Vine Linuxに対して、Red Hat Linuxと同等の個人/企業向けサポートがレッドハットが

ら提供される。現行バージョンの2.0については、発売から1年間は技術評論社でサポートを受け付けるが、その後はレッドハットに引き継ぐ予定。

同時に、Red Hat Linux日本語版の開発に、Project Vineが全面的に協力することになった。Red Hat Linux 6.2Jに対するアップデートとして、「Red Hat 6.2J Update with Vine Linux Technology」CD-ROMが提供される。

なお、今後もVine Linuxの開発は、Red Hat Linuxとは独立して進められる予定。

Project Vine (<http://vinelinux.org/>)

レッドハット株式会社 (<http://www.redhat.com/jp/>)

Red Hat Linux 6.2J改訂版リリース

レッドハットは、「Red Hat Linux 6.2J 改訂版」を7月21日に発売した。パッケージ構成は、Red Hat Linux 6.2Jに「Red Hat 6.2J Update with Vine Linux Technology」CD-ROMを追加したもの。このCD-ROMにより、Project Vineやオープンソースコミュニティの成果物である、日本語化されたLinux環境をRed Hat Linuxでも利用できるようになる。主な収録ソフトは、X-TTに対応したXFree86 3.3.6、安定版GNOMEの最新版、1.2.3、各種のウィンドウマネージャ（Enlightenment 0.16.4、amaterus 0.32.5、Sawfish 0.28.1）、Emacs 20.7、XFree86用日本語フォントなど。

また、デラックス版とプロフェッショナル版には、韓国

Hancom Linuxが開発した、「HancomWord」が収録されている。HancomWordは、Microsoft Wordのドキュメントが扱える日本語ワープロソフト。

そのほかの変更点は、安定版カーネル2.2.16の採用、PostgreSQL 7.0.2の収録、Oracle 8iのインストールガイドとインストーラ用パッチの収録など。

価格・商品構成は6.2Jから変更はなく、サーバ用のプロフェッショナル版が2万9800円、デスクトップ用が1万2800円、商用ソフトを含まないスタンダード版が3980円。

レッドハット株式会社 (<http://www.redhat.com/jp/>)

組み込み用途Linuxのコンソーシアム設立

7月13日に、組み込み用途でのLinux普及を目指す「日本エンベデッドリナックスコンソーシアム」(略称:Emblix)が設立された。

Emblixは、産学協同の団体で、組み込み用途向けLinux(Embedded Linux)に関する周辺技術の標準化、国内外のコミュニティとの情報交換、各種の啓蒙活動を行う予定。メンテナンス、サポート、品質保証といった、Embedded Linux採用の障壁になりがちな課題を解決していくとしている。

会長は、早稲田大学工学部助教授の中島達夫氏、副会長

には、ターボリナックス ジャパンのJohn Cheuck氏、ガイオ・テクノロジーの國峯幸雄氏が選任されている。

主な参加団体は、早稲田大学、豊橋技術科学大学、ハードウェアベンダーの東芝、NEC、日立製作所、富士通、三菱電機、Linuxディストリビュータのターボリナックス ジャパン、レッドハットなど。

日本エンベデッドリナックスコンソーシアム

(<http://www.emblix.org/>)

米TurboLinux社がAlphaシステム用Linuxをリリース

7月18日に、コンパックのAlphaプロセッサを搭載したシステム用のLinuxが米TurboLinux社からリリースされた。

64ビットCPUであるAlphaプロセッサを搭載した、コンパック製のシステムに対応しており、Alphaの特長である高速な浮動小数点演算能力を活かすように最適化されている。

TurboLinux、コンパックの両社は、膨大な計算能力が必要

になる金融機関、インターネットサービスプロバイダ、CGプロダクションなどに、費用対効果の高いソリューションを提供できるとしている。

TurboLinux Inc. (<http://www.turbolinux.com/>)

コンパック (<http://www.compaq.com/>)

Products

- 38 低価格インターネットシェアリングBOX
bRoad Lanner BRL-01
- 40 初心者でもカンタンにLinuxの設定ができるGUIツール
HDE Linux Controller 2.0 Standard Edition
- 42 MacOS上でx86版Linuxを動作させる
Virtual PC 3.0 日本語版 with TurboLinux CE
- 43 ミラーリング&デュアルCPUで高機能を実現したPCサーバ
DualCeleronPC DC40SV-CD

低価格インターネットシェアリングBOX



bRoad Lanner BRL-01

急速に普及しているCATVやxDSLによるインターネット接続では、1台のPCでの利用が前提になっていることが多い。しかし、LAN環境にある複数台のPCから利用したいと考えるのは当然だろう。本機はこのような要望に応えるためのNAT/IPマスカレード機能を持ったイーサネットルータだ。

製品名 bRoad Lanner BRL-01
価格 2万4800円(参考価格)
問い合わせ先 プラネックスコミュニケーションズ株式会社
TEL 03-3256-9101
<http://www.planex.co.jp/>

ISDN網とダイヤルアップルータを使った接続であれば、LAN内の複数のPCからインターネットを利用することは当然のように可能だ。しかし、CATVやxDSLを使用したインターネット接続では、ダイヤルアップで接続する場合と同様に、インターネットに接続しているマシンでしかインターネットが利用できない。CATVやxDSLを使用して複数台のPCからインターネットを利用するには、LAN内にルータを設置する必要がある。

もちろん、Linuxを使えばルータ機能を実現することは可能だ。しかし、ファイアウォールの設定やNAT/IPマスカレードの設定など、かなりの知識と手間が必要となる。それに、常にLinuxサーバを稼働しなければならないので、電気代や設置スペースなど、家庭内での使用には難がある。それらの問題を一手に解決するのがbRoad Lanner BRL-01(以下、BRL-01)だ。

BRL-01の接続は、ツイストペアケーブル(ネットワークケーブル)を使っ

て、LAN側ポートにハブを接続し、WAN側ポートにケーブルモデムなどを接続するだけだ。LAN側は10BASE-T/100BASE-TX自動認識に対応しているが、WAN側は10BASE-Tのみとなっている。なお、WAN側の回線速度は1.5Mbpsまでとなっている。

電源はACアダプタを使って供給し、消費電力は最大5Wと非常に経済的だ。

設定はWebブラウザがtelnetを使って行う。このため、特殊なソフトウェアをインストールする必要がない。設



写真1 BRL-01の背面
左からWAN側ポート（ケーブルモデムなどの接続）、LAN側ポート、電源コネクタ、パワーLED、リセットスイッチ。電源スイッチは付いていない。

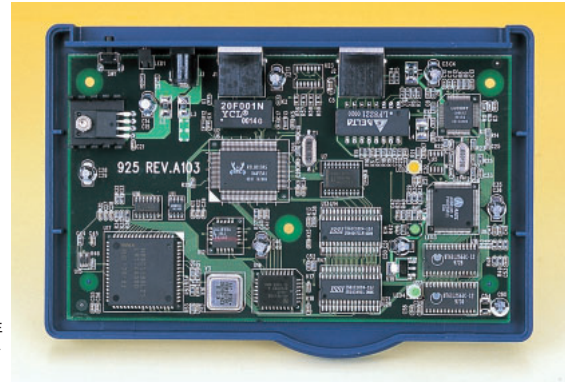


写真2 本体内部を開けたところ
ACアダプタを使用するために、本体は非常にコンパクトにまとめられている。ファンレス設計で全く音はしない。

定方法も、「簡単設定」を選べば、ほとんど何も設定することなく、ルータとして稼働させることが可能だ。もちろん、「アドバンス設定」で詳細に設定することもできる。

また、遠隔地からインターネット経由でBRL-01の設定を行うこともできる（設定を禁止することも可能）

豊富な機能

イーサネットルータとしての機能は必要十分だ。NAT / IPマスカレード機能を搭載し、最大253台までのPCでアドレスを共有することができる。もちろん、DHCPサーバ機能も持っており、自動的にIPアドレスを割り振ることも、MACアドレスを使って固定IPアドレスを割り当てることも可能だ。WAN側ではDHCPクライアント機能を持っており、プロバイダのDHCPからアドレスを自動取得することもできる。

また、無接続状態が続く場合に自動的に回線を切断するように設定することも可能だ（分単位で指定）。従量課金制のプロバイダを利用している場合に特にありがたい機能だ。

その他の機能として、「バーチャルコンピュータ機能」、「ローカルサーバ機能」、「IPフィルタリング機能」などがある。

バーチャルコンピュータ機能は、LAN上の特定のPCをインターネットから参照させるための機能だ。BRL-01へのパケットはすべて指定したPCに転送される。

ローカルサーバ機能は特定のポート

をLAN上のコンピュータに割り振る機能だ。この機能を使えば、FTPはマシンA、HTTPはマシンBといった割り振りが可能になる。

IPフィルタリング機能は、特定のマシンやユーザーが使用するプロトコルを制限するためのものだ。FTPを禁止したり、telnetを禁止したりすることができる。さらに、TCP / IP以外のプロトコルを通さないようになっている。これらの機能により、LAN内は完全に独立したセグメントにすることができるので、セキュリティも堅牢なものとなる。

また、ルーティングテーブルも設定することができるので、より柔軟なネットワーク構成が可能になっている。

ファームウェアはtftpプロトコルによってバージョンアップが可能だ。

なお、BRL-01はxDSLにも利用できるイーサネットルータであるため、

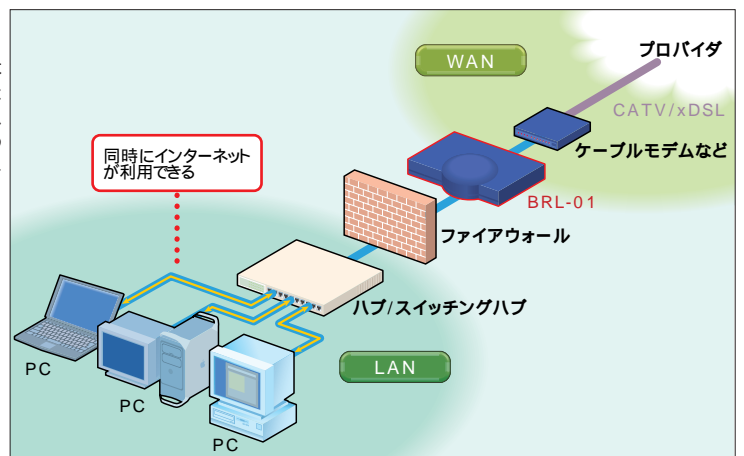
PPPoE（Point To Point Protocol Over Ethernet）にも対応している。PPPoEとは、xDSLで使用されるプロトコルで、TCP / IPのパケットにDSLヘッダを付加したものである。

家庭内LANにお勧め

CATV / xDSL向けのイーサネットルータはいくつも発売されているが、他製品が5万円前後なのに対し、BRL-01は機能も同等で2万4800円と非常に低価格である。消費電力が少ない、ファンレス設計など、家庭内LANやSOHOでの利用に特にお勧めだろう。

CATV / xDSLプロバイダによっては、ルータなどを使って複数のPCからの同時アクセスを禁止していたり、別途契約が必要な場合があるので注意していただきたい。

図1 ネットワーク構成図
LAN内とWAN側は別のネットワークセグメントとなる。ハブによって複数台のPCからインターネットが利用できる。





HDE Linux Controller 2.0 Standard Edition

Linuxの各種設定はテキストファイルを編集することでできるという便利な面はあるが、設定ファイルの記述方法が複雑だったり、設定ファイルがあちこちのディレクトリに散在しているなど、非常にわかりにくい。そんな難物の設定を一手に引き受けてくれるのがこのソフトだ。

製品名	HDE Linux Controller 2.0 Standard Edition
価格	3万円 (アカデミックパック 1万9800円)
問い合わせ先	株式会社ホライズン・デジタル・エンタープライズ TEL 03-5456-3260 http://www.hde.co.jp/

Webブラウザ上でLinuxを設定管理するツールとして定評のあるHDE Linux Controller 1.0が8月25日にバージョンアップし、HDE Linux Controller 2.0 Standard Edition (以下、HLC 2.0)として発売される。バージョン1.0から特に大きな機能追加はされていないが、ユーザーインターフェイスは一新され、より直感的に使うことができるようになっている。また、設定画面の階層も見直され、多くても3回のクリックで目的の画面に到達できるようになった。



対応ディストリビューション

HLC 2.0は、Linuxの各種設定ファイルを読み込み、現在の設定をわかりやすく表示し新たに設定された内

容を設定ファイルに書き戻すようになっている。このため、設定ファイルの書式や保存場所のディレクトリ構造の関係で未対応のディストリビューションで動作させてはいけない。現在対応しているディストリビューションは以下の通りである。

Red Hat Linux 6.0/6.1/6.1J/6.2/6.2J
Turbo Linux Server **日本語版** 6.0/6.1
LASER5 Linux 6.0 Server Edition

なお、本記事では 版を使用したため、製品版と異なる可能性があることを了解いただきたい。



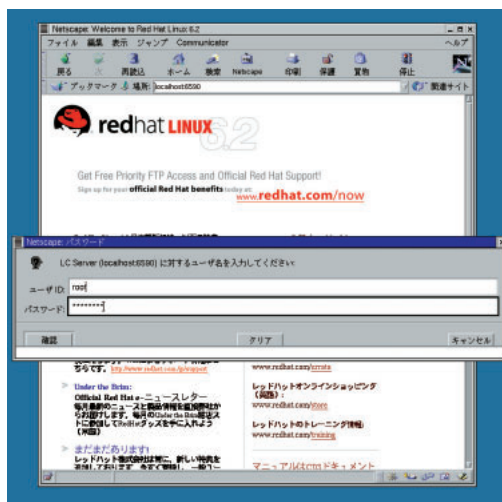
豊富な機能と使いやすさ

HLC 2.0はWebベースのアプリケー

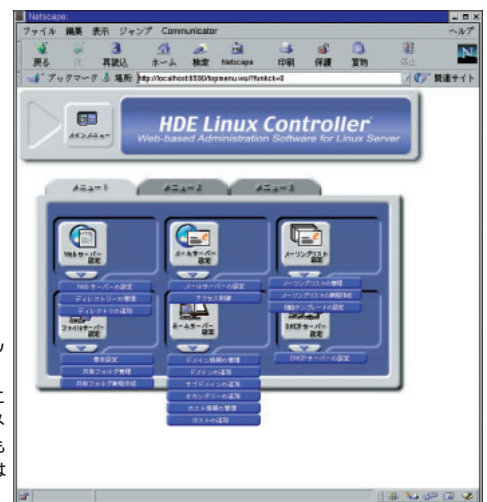
ションとして機能する。LinuxのWebブラウザをはじめ、WindowsやMacintoshのWebブラウザから利用できるため、ネットワークを経由して遠隔地からもサーバの設定を行うことが可能だ。ただし、SSL (Secure Socket Layer)には対応していないのでセキュリティに関して注意が必要だ。

HLC 2.0で設定管理できる機能は豊富で、ほとんどの設定管理が行えるようになっている。なお、以下のような機能が用意されている。

- Webサーバ設定
- メールサーバ設定
- メーリングリスト設定
- ファイルサーバ設定
- ネームサーバ設定
- DHCPサーバ設定



画面1 HLC 2.0のログイン画面
http:// <ホスト名> :6590/に接続する。HLC 2.0を使用するには、rootユーザーのパスワードを入力する必要がある。



画面2 HLC 2.0のトップメニュー
[]をクリックすることで、ショートカットメニューを表示することもできる。メニュー構造はカスタマイズ可能だ。

- ・ネットワーク設定
- ・セキュリティ
- ・ユーザー/グループ設定
- ・バックアップ管理
- ・サーバステータス確認
- ・ソフトウェア管理
- ・初期セットアップウィザード

サーバを構築する場合に「初期セットアップウィザード」を使えば、ウィザード形式で質問に答えるだけで、「オールインサーバ」「インターネットサーバ」「ファイルサーバ/イントラネットサーバ」向けにネットワークや起動デーモンなどが適切に設定される。初心者でも簡単にサーバを構築できる機能だ。

また、メールサーバやネームサーバ、セキュリティなどの難しい設定も、現在のネットワーク構成やシステムを解析して、推奨値が表示されるようになっている。どのように設定したらよいかわからない場合でも安心だ。

各設定画面はタブ構造のメニューになっている。機能の分け方も直感的で理解しやすい(画面2)。設定値や機能がわからなくてもヘルプ画面



画面3 セキュリティの設定

現在の設定値が表示され、GUIで簡単に変更できる。[?]をクリックするとヘルプが表示される。

が表示される(画面3)。また、よく使う機能を同じタブにまとめて表示できるように、各メニューはカスタマイズ可能になっている。

ログの表示機能は特に便利な機能だ(画面4)。多くの情報の中から重要なものを色分けして表示するようになっているので、重要なログ情報を探しやすい。もちろん、検索することもできる。lessやgrepなどでログを見るよりもずっと便利に使うことができる。

さらに、この手の管理ツールとしては珍しくバックアップ機能も用意されていて、定期バックアップなどの設定が可能だ。

このように、HLC 2.0を使えばLinux設定管理の負担を大幅に軽減できる。Linuxの設定はviがあれば十分だという人も、一度使ってみると手放せなくなるだろう。

気になるサポートは、電話、メールによる90日間3インシデントとなっている。

このように、非常に便利なHLC 2.0だが、設定内容をまとめてファイルに保存できる機能があるとより使いやすだろう。複数のサーバを管理する場



画面4 サーバステータスの表示

各種ログが表示できる。重要なログ情報は色分けされて表示される。画面では一般的なログ情報からloginに関するログを検索して表示した。

合や、設定値だけをバックアップしておく場合に有効だ。ぜひ今後のバージョンで対応していただきたい。

また、より多くのディストリビューションに対応していただけるとありがたい。



なお、より高度な設定を行うことができる上級者向けのバージョンがProfessional Edition(以下、Professional版)として9月に発売される予定だ。

Professional版では、「SSL(Secure Socket Layer)対応」、「UPS設定」、「APOP設定」、「バーチャルドメイン設定」、「quota設定」など、より多くの機能に対応する。特に、インターネット経由で設定を行う場合はセキュリティが問題となるので、リモート設定を行いたい場合はSSLの使えるProfessional版を使用するほうが良いだろう。また、Professional版ではユーザーアカウントの一括登録などの機能が搭載される予定になっている。小規模サーバや初心者向けがStandard版、中規模以上のサーバ向け製品がProfessional版ということになるだろう。

お詫び

今月号の付録CD-ROMに収録したHDE Linux Controller 2.0 trial版にはインストールスクリプトが収録されていません。インストール方法は下記URLを参照してください。

<http://www.ascii.co.jp/linuxmag/HLC/>



Virtual PC 3.0 日本語版 with TurboLinux CE

Virtual PCは、PowerPCを搭載したMacintosh上でx86マシンをエミュレートするソフトウェアだ。本製品は、仮想x86マシン用のOSとしてTurboLinuxを同梱している。動作には350 MHz以上のPowerPC G3 / G4を搭載したMacintoshが必要だ。

製品名	Virtual PC 3.0 日本語版 with TurboLinux CE
価格	1万9800円
問い合わせ先	株式会社アスキー ネットメディア事業部 TEL 03-5351-8652 http://www.ascii.co.jp/netmedia/

Virtual PCは米Connectix社が開発した、MacOS上で動作するx86マシンエミュレータソフトだ。本製品には、x86版のTurboLinux CE (Connectix Edition) が付属している。

必要なスペックは、CPUが350 MHz以上のPowerPC G3 / G4、1.1 Gバイト以上のハードディスク容量、Virtual PC用に割り当てるメモリが96Mバイト以上となっている。



インストールは簡単

本製品に含まれるTurboLinux CE (TLCE) はVirtual PC専用であり、インストール時に設定で頭を使う必要はまったくない。Macintosh用の標準的なインストーラでVirtual PC本体とともにインストールされる。このへんの簡単さは、Macintoshのソフトならではという感じだ。

TLCEはTurboLinux 6.0 Workstation (Shiga) がベースになっており、カーネルは2.2.13、Cライブラリは2.1.2が採用されている。6.0のデフォルトのデスクトップ環境はGNOMEだが、TLCEではKDE 1.1.2が標準になっている。なおTLCEには、製品版のTurboLinuxに同梱されているATOKなどの商用アプリケーションは付属していない。

CD-ROMドライブ、ネットワークイ

ンターフェイスなどのハードウェアは、TLCE上からも利用可能だ。特にネットワークについては、MacOS上でLANに関する設定ができていれば、TLCEからもそのままLANに接続できる (画面1)。PowerPCネイティブなLinuxPPCがあるのに、わざわざx86版をエミュレータで動かす意味があるのか? という人もいるだろう。しかしLinuxPPCをMacintoshにインストールしようと思ったら、MacOSパーティションのバックアップをとったうえで、パーティションの切り直しなどの作業が必要になる。だがVirtual PCならそのような手間をかけずにLinux環境を利用できる。それにVirtual PC終了時には、TLCEの状態を保存しておくことができるため、ノートPCでおなじみのレジューム機能を持ったLinuxマシンのような使い方ができる。使う時にいちいちLinuxの起動を待つ



画面1 Virtual PC TurboLinux CEのデスクトップをウィンドウモードで表示している。もちろんこれは「はめ込み合成」ではない。全画面表示への切り替えも可能だ。

ている必要がないのは、ありがたい。

Virtual PCは通常のMacintoshアプリケーションであるため、コピー/ペーストといったMacOSの標準的な機能はそのまま利用できる。そのため、TLCE / MacOS間で双方向のテキスト受け渡しが可能であり (画面2)、作業によってはMacintoshとLinux PCを2台並べて使うよりも効率的だ。

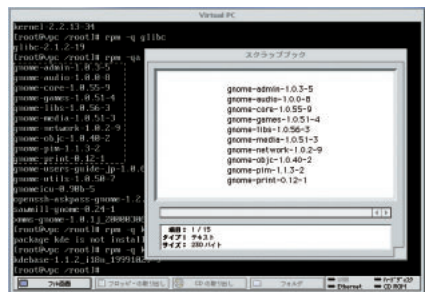


デスクトップ環境は少々重い

編集部では、iMac DV Special Edition (G3 400MHz、メモリ384Mバイト) を用いてテストを行った。

テキスト画面のコンソールでの動作は、エミュレータとは思えないほど軽快で、十分に実用になる。

しかし画面表示が遅いため、X Window Systemの利用は、つらいかもしれない。今後の改善が望まれるところだ。



画面2 Virtual PCとMacOSの連携 Virtual PCは、Macintoshのアプリケーションとして動作しているので、当然両方向のコピー/ペーストができる。



DualCeleronPC DC40SV-CD

サーバ機には高性能と高信頼性が要求される。本機は、CeleronとIDEハードディスクという比較的安価なパーツで構成しているが、デュアルCPUでSMPマルチプロセッサに対応し、2台のハードディスクでミラーリングを行うことで信頼性を確保している。

製品名	DualCeleronPC DC40SV-CD
価格	19万8000円
問い合わせ先	株式会社アスキーイーシー TEL 03-5351-8992 http://www.e-sekai.com/pc/

DualCeleronPC DC40SV-CDは、アイ・オー・ネットが開発・販売している低価格で高性能なPCサーバで、アスキーイーシーが運営するECサイト「e-sekai」のオンラインPCショップより購入できる。

DC40SV-CDは、CPUにCeleron 400MHzを2基搭載しており、LinuxやWindows NT/2000といったSMP（対象型マルチプロセッサ）に対応するOSを使用することで、1つのシステムで2つのプログラムを同時に実行することができる。

そして、2台の20GバイトIDEハードディスクを内蔵し、IDEmirror 2000（以下IDEmirror）によってハードウェアミラーリングをサポートしている。ドライブ自体はパワースイッチ付きのリムーバブルケースに収

められていて、ホットスワップが可能になっている。

DC40SV-CDには、OSがインストールされていないが、LASER5 Linux 6.0 Rel.2フリー版がバンドルされている。早速CD-ROMからインストールしてみたところ、ハードディスクはIDEmirrorが認識され、Linuxからは1台のドライブに見える。グラフィックカードにRIVA TNT2を選択し、インストールが終了すると、SMP対応カーネルを使うように自動認識されている。ネットワークカードは、DEC互換チップを使用しているため、インストール後に手で設定する。

デュアルCeleronの効果は、xosviewを使ってX上でCPUの稼働状

況をグラフ表示することで確認できる。複数のターミナルから同時にコマンドを入力すると、2つのCPUに自動的に振り分けられて実行される。

Celeron 400MHzというのは、今では速いCPUではないが、デュアルプロセッサなのでトータルパフォーマンスは高い。手頃な価格で性能と信頼性をバランスよく追求したPCサーバである。



写真2 ミラーリングを行うIDEmirror2000
IDEハードディスクのミラーリングはIDEmirror2000が行う。ホットスワップに対応し、故障交換後のディスク再構成も自動的に行われる。

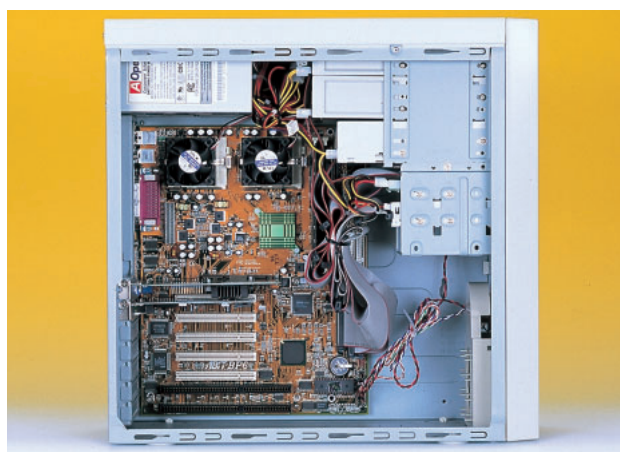


写真1 デュアルCeleronを実現したBP6マザーボードを使用
IDEmirror2000を経由して2台のハードディスクへ接続する。

CPU	Celeron 400MHz (Socket 370) × 2
メモリ	128MバイトPC100 SDRAM (空き2スロット)
マザーボード	ABIT BP6 (Intel 440BX)
ハードディスク	20Gバイト × 2 (IDEミラーリング)
CD-ROM	40倍速CD-ROM
フロッピードライブ	3.5インチ (1.44Mバイト)
グラフィックスカード	RIVA TNT2 VANTA (AGP)
ネットワーク	10/100BASE-TX
ハードディスクインターフェイス	UltraDMA/33 × 2, UltraDMA/66 × 2 (HPT366)
シリアルポート	RS-232C × 2
パラレルポート	1ポート
USBポート	2ポート
拡張スロット	PCI 5スロット (空き4) ISA 1スロット (空き1)
キーボード	PS/2日本語109キーボード
マウス	PS/2マウス (2ボタンホイール付き)
ケース	ミドルタワー
外形寸法 (mm)	200 (W) × 420 (D) × 414 (H)
電源	250W/ATX

表1 DualCeleronPC DC40SV-CDの主な仕様

夏休み

フリーソフト

夏ですね。夏といえば海に山に.....といきたいところですが、キンキンにクーラーが利いたお部屋でじっくりコンピュータに向かうのもオツなもの。正統派Linuxerなら、ここはやはり窓でも林檎でもなくペンギンとたわむれるのが正調。お勧めはフリーソフト三昧というわけで、少し遅めのお中元「Free Application Showcase」超拡大版を総力特集でお届けします。

文：出井 一、山岸 典将、あべひろのぶ、松原 健一、編集部

Text : Hajime Dei, Norimasa Yamagishi, Hironobu Abe, Kenichi Matsubara, Linux magazine

Photo: Shuichi Mito

大会





まず、インストールしてみよう

UNIX系OSのソフトウェアは、WindowsやMacintoshと違って、ソースファイルで提供されるのが一般的だ。tarコマンドでソースやドキュメントをひとまとめにし、gzipなどで圧縮した、アーカイブファイル(「tarボール」「tar玉」と呼ばれる)で配布されていることが多い。

Linuxの場合は、tarボール形式以外に、Red Hat系ディストリビューションが採用しているRPM(Red Hat Package Manager)形式のパッケージで配布されることも多い。使っているディストリビューション用のパッケージが提供されているなら、rpmコマンドやGUIのパッケージユーティリティで簡単にインストールできる手軽さが受けて、普及が進んできた。

パッケージには、ほかにもDebian GNU/Linuxで用いられているdeb形式などがある。使っているディストリビューション用のパッケージが提供されていない場合には、「Alien」というユーティリティを用いるとよいだろう(画面1)。これは、tarボール、RPM、

debなどパッケージ間の変換を行うPerlスクリプトだ。

ここでは一般的な、RPMのバイナリパッケージのインストール方法、ソースRPMパッケージからバイナリを作成する方法、そしてtarボールからビルドする方法の3通りを説明する。ここまで理解していれば、ほとんどのフリーソフトを利用できるはずだ。



RPMのバイナリパッケージインストール

RPMのバイナリパッケージは、拡張子が「i386.rpm」のファイルだ。ディストリビューションによっては、i586、i686となっていることもある。これらのファイルには、ソースからビルドされた実行ファイル、データ、ドキュメントなどが含まれているので、インストールすれば実行のための環境が整う。インストール時にはライブラリなどの依存関係をチェックしてくれるし、パッケージのアンインストールやアップデートも容易に行える。

なおインストール/アンインストー

ルには、root権限が必要になるパッケージが多い。

rpmコマンドを利用する方法

RPMパッケージのインストール、アンインストール、中身の確認などを行うには、コマンドライン版のrpmコマンドか、GUI版のGnoRPMやKpackageなどを用いる。まずは基本中の基本であるrpmコマンドからだ。

インストール時には、インストール用の-iオプションではなく、アップデート用の-Uオプションを使うほうがよいだろう。古いパッケージがインストールされているかどうかを調べて、新規インストールか更新かを判断してくれる。インストール経過を「#」のグラフで表示する-vhオプションと組み合わせて、

```
# rpm -Uvh hoge-1.2.3-4.i386.rpm
```

とすればよい。

必要なライブラリパッケージがインストールされていない場合には、「が必要です」などのメッセージが表示されてインストールに失敗する。この問題を解決するには、必要とされるパッケージを先にインストールすればよい。依存関係を無視する--nodepsオプションや、ほかのパッケージからインストールされたファイルを無条件で置き換える--forceオプションを使うと、一見インストールできたように見える。だが、たいてい思ったように動かない結果に終わるので、安易に使わないほうがよい。

一方、パッケージをアンインストー



画面1 AlienのWebサイト。
<http://kitenet.net/programs/alien/>



ルする場合は、-eオプションとパッケージ名を指定して、

```
# rpm -e hoge
```

とすればよい。パッケージファイル名ではなく、パッケージ名を指定する点に注意しよう。

GUIによるバイナリパッケージのインストール

GUIを使ってRPMのパッケージを扱うソフトとしては、GNOME上で動作する「GnoRPM」、KDE上で動作する「Kpackage」などがある。

GnoRPMのウィンドウ(画面2)には、インストール済みのパッケージがグループ別のフォルダに分類されている。各パッケージは右側の領域にアイコンで一覧表示され、右クリックメニューから「情報(クエリー)」を選択すると、パッケージに関する説明や含まれているファイルを確認できる。

パッケージをインストールするには、「インストール」ボタンを押してダイアログを開いてパッケージを選択していけばよい。

一方、Kpackageのウィンドウ(画面3)にもインストール済みのパッケージがグループ別にツリー表示される。こちらは、ツリー中にパッケージ名そのものが表示されており、これらをクリックすると、右側の領域にそのパッケージに関する説明や含まれているフ



画面2 GUIパッケージ管理ツール。GNOMEには、GnoRPMが用意されている

ファイルが表示される。

パッケージをインストールするには、ツールバー左端の「パッケージを開く」ボタンを押す。インストールするパッケージを選択していけばよい。



ソースパッケージからバイナリを作成する

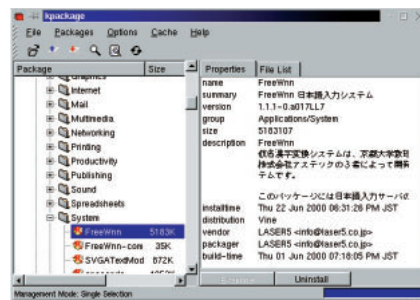
ひとくちにRPM系といっても、ディストリビューションごとに微妙な差異があるため、あるディストリビューション用に作られたバイナリパッケージが、別のディストリビューションでは使えないことがある。

そのような場合にはソースRMPパッケージ(拡張子が「src.rpm」)から、バイナリパッケージを作成(リビルド)してやればよい。rpmコマンドの--rebuildオプションを使って、

```
# rpm --rebuild hoge-1.2.3.4.src.rpm
```

とすればよい。

ソースパッケージには、元になるソースファイルからプログラムを作成するための手順を示したspecファイルが含まれており、リビルドするとspecファイルに基づいて、現在のシステム環境(ライブラリなど)に合わせてプログラムが作成される。リビルドに成功したら、/usr/src/redhat/RPMS/i386(ディストリビューションにより異なる)に置かれたバイナリパ



画面3 RPMだけでなく、debパッケージなども扱える、Kpackage

ッケージをインストールすればよい。



tarボールからビルドする

RPMパッケージが用意されていないなら、tarボールから自分でソースをコンパイル(ビルド)してプログラムを作成しよう。

まず、tarボールを展開する。「hoge-1.2.3.tar.gz」というファイルを展開するには、

```
$ tar xvzf hoge-1.2.3.tar.gz
```

とする。カレントディレクトリにサブディレクトリ(たいていは「hoge-1.2.3」)が作成され、そこにファイル一式が展開される。

次に、README、INSTALLなどのドキュメントを参照する。実際には、「configure」という名のスクリプトを実行して、環境に合わせたMakefileを生成し、その後makeコマンドを実行するという流れになっているプログラムが多いが、念のため確認しておこう。

実際のビルドは、

```
$ ./configure
```

としてMakefileを作り、

```
$ make
```

でコンパイルが行われる。その後、

```
$ su
```

```
# make install
```

とすることで、プログラムやデータが適切なディレクトリにインストールされるようになっていることが多い。

掲載フリーソフト早見表

エンターテイメント		
glTron	ライトサイクルで爆走する3Dアクションゲーム	50
Tux Racer	ペンギンが雪山を滑り降りる3Dアクションゲーム	50
XRACER	雪山を舞台にした3Dレースゲーム	51
Pingus	ペンギンが主人公になったレミングスのクローンゲーム	51
Rocks'n' Diamonds	エメラルドとダイヤモンドをゲット!	52
FALLING TOWER	崩れるタワーをジャンプ!ジャンプ!	52
XMris	リンゴを落として敵をやっつける!	53
Circus Linux	コミカルな風船割りアクションゲーム	53
Xboing	美しいグラフィックとサウンドのブロック崩しゲーム	54
BZFlag	ネットワーク3Dタンクバトルゲーム	54
Xkobo	360度バリバリシューティング野郎	55
Cosmic Shooter	縦スクロールの定番シューティングゲーム	55
Missile Command	ミサイルを迎撃して街を守れ!	56
Batalla Naval	ネットワーク対応の索敵ゲーム	56
Freeciv	戦略シミュレーションゲームCivilizationIIのクローン	57
RealTimeBattle	プログラムしたロボットを戦わせるシミュレーションゲーム	57
XShipWars	宇宙船で自由に宇宙を飛び回るゲームシステム	58
Flight Gear Flight Simulator	世界各地を飛行できるフライトシミュレータ	58
Japanese NetHack	ペットを連れて、さあ冒険に出かけよう	59
Cannon Smash	ネットワーク対戦も可能な本格卓球ゲーム	59
Gtulpas	リアルな拳撃の3Dビリヤードゲーム	60
XTokkaeTama	「たま」をとっかえて連鎖を狙え!	60
GNOME 3D Tetris	空間把握能力が問われる3D落ちモノパズル	61
GemDropX	同じ色の宝石を並べろ!	61
x777	数字合わせ落ちものパズルゲーム	62
XPuzzleTama	「ぶよぶよ」タイプの落ちものパズルゲーム	62
blackbox	隠されたクリスタルの場所を見つける	63
KGryzzles	KDEベースの一笔書きパズルゲーム	63
Kishido	36種類の石を配置するパズルゲーム	64
Kpooka	思考力が試される論理型パズルゲーム	64
SameGame for X Window	X版「さめがめ」	65
GSokoban	GNOME上で動作する「倉庫番」風パズルゲーム	65
Heroes	華麗な画面とエフェクトの新世代ワームゲーム	66
xrenju	X対応の本格派連珠ゲーム	66
四川省 for X11	麻雀牌を使ったパズルゲーム	67
PySol	200種類以上のソリティアを集めたカードゲーム	67
The Ace of Penguins	ソリティアゲームパック	68
Onsoo	海外生まれの花札パズルゲーム	68
Open Universe	太陽系の惑星や衛星などを3D表示するシミュレータ	69
Xplns	アストロアーツが贈る天文シミュレータ	69
Xplanet	ルートウィンドウにリアルな地球を表示する	70
Xcruise	ファイルシステムの宇宙空間を旅しよう	70
GnomeKiss	世界に広がる着せ替えシステムの輪	71
ActX	あなたのデスクトップのマスコット	71
Penguineyes	デスクトップにペンギンを!	71
xpenguins	かわいいペンギンがデスクトップを走りまわる	71
マルチメディア		
CompuPic	30種類以上の画像形式に対応した画像ソフト	72
gPhoto	さまざまなデジカメの画像を扱える画像ソフト	72
GQview	ワンクリックで画像を表示するビューア	73
KShow	サムネイルにも対応したイメージカタログビューア	73
Morpheus	3Dモデルをさまざまな角度から眺められるビューア	74
Xfrackey	さまざまなフラクタル図形を高速に表示	74

xmrm	形と模様の変化を個別に制御できるモーフィングソフト	75
aKtion!	マルチメディアプレーヤー-xanimのフロントエンド	75
FreeAmp	テーマにも対応した高速MP3プレーヤ	76
Grip/GCD	CDリッパやMP3エンコーダと連動するCDプレーヤ	76
kmp3	KDEデスクトップ環境で動作するMP3プレーヤ	77
MpegTV Player	多機能なMPEGビデオプレーヤ	77
mpg123	コマンドラインで動作するMP3プレーヤ	78
Xmms	X上で動作する定番のMP3プレーヤ	78
eXtace	再生中のサウンドをさまざまに視覚化する	79
Gmurf	WAVとMP3に対応したウェブエディタ	79
Synaesthesia	演奏中のCDの音などをグラフィカルに表示	80
TiMidity++	高性能なソフトウェアシンセサイザ	80
Visual Sound Analyzer	サウンドを視覚化するGNOMEアプレット	81
アプリケーション		
Koffice	待望のKDE版オフィススイート	82
USI	高性能な日本語オフィスソフト	83
MagicPoint	日本語対応の定番プレゼンテーションソフト	83
Dia	フローチャートなどのダイアグラムを描画	84
Tgif	定番のドローツール	84
Sketch	高性能なドロー系グラフィックツール	85
SciGraphica	散布図などの科学技術系2次元グラフを描画する	85
Gimp	もっとも有名なビットマップ系グラフィックツール	86
Blender	高性能3Dモデリング/アニメーションツール	86
Giram	入門者向け3Dモデリングソフト	87
ImageMagick	数多くのフォーマットに対応した画像ツール	87
ne	Vzライクな特徴を持つ高速テキストエディタ	88
Bluefish	日本語に対応した使いやすいHTMLエディタ	88
Web Designer	日本語をカラー構文表示できるHTMLエディタ	89
CodeCommander	プログラム開発に適したテキストエディタ	89
KDevelop	KDE上で統合開発環境を提供する	90
MMM	日本語対応の軽量WWWブラウザ	90
w3m	テキストベースの軽快なWebブラウザ	91
Sylpheed	安定して使いやすいメールクライアント	91
The N-Tool(Nmail4)	日本製の高性能メーラ/メールチェッカー	92
Wanderlust	Emacs上で動作する先進的なメールクライアント	92
GnomeICU	日本語対応のGNOME版ICQクライアント	93
KXicq	KDEのICQチャットクライアント	93
Liece	Emacs上で動作する国産のIRCクライアント	94
X-Chat	GTK+を利用した高性能なIRCクライアント	94
micq	テキストベースで軽快に動作するICQクライアント	95
GtkIpmMsg	サーバーいらずのメッセージ交換ソフト	95
IglooFTP PRO	2ペイン構成の定番FTPクライアントソフト	96
gFTP	シンプルなX対応FTPクライアント	96
yafc	ファイル名補完機能やカラーls機能を備えたFTPクライアント	97
caitoo	ドラッグ&ドロップ操作にも対応したダウンローダ	97
Downloader for X	複数ファイルを効率よくダウンロード	98
Knapster	充実した検索機能の多機能Napsterクライアント	98
karchie	Archieサーバからプログラムをゲット!	99
KWebGet	Webページを取得するwgetのフロントエンド	99
Glitter	ネットニュースからバイナリファイルを取り出す	99
KDbg	gdbのGUIインターフェイス	99
ユーティリティ		
kruiser	KDE版「エクスプローラ」?	100
Cxplorer	付属ソフトの豊富な日本語対応ファイルマネージャ	100



gentoo	GUIで設定できる2画面操作のファイルマネージャ	101
Vide	viライクなキー操作の2画面ファイルマネージャ	101
Explorer2fs	Windowsからext2ファイルシステムにアクセス	102
LinNeighborhood	Sambaの共有ディレクトリをGUIでマウント	102
mfm	DOS形式のフロッピーを簡単に扱えるファイルマネージャ	103
Kleandisk	不要なファイルを検索して削除やアーカイブを行う	103
GnoZip	GUIのアーカイバ	104
LnXZip	RPMの作成も可能な統合アーカイバ	104
TkZip	RPMやZIP / LHAも扱えるアーカイブマネージャ	105
kpackage	GUI操作で各種パッケージを管理	105
KPackView	あらゆるフォーマットに対応したパッケージビューア	106
korganizer	シンプルで使いやすいスケジュール管理ソフト	106
nittei	スケジュール管理を行える日本語カレンダー	107
X Desktop Calender	Xのルートウィンドウに常駐するカレンダー	107
MemoPanel	GNOMEのパネルにメモを貼りつける多機能ソフト	108
LinBot	Webページのリンク切れをチェック	108
kwebwatch	webページ更新チェッカー	109
WWWcp	Webサイトのページをまとめてコピー	109
fetchmail	リモートのメールサーバからメールを一括転送	110
PPxP	多機能で使いやすいIPPP接続ツール	110
Eterm	テーマ機能を備えるターミナル	111
Gaby	気軽に使えるパーソナルデータベース	111
Joe's QVocabulary	英単語などの語彙学習用の簡易データベース	112
M3X	XMLを利用した音楽メディア用データベース	112

KEbook	日本生まれの電子辞書検索ソフト	113
Ktransrator	po / potファイル編集ツール	113
glclock	OpenGLを利用したリアルな3D時計	114
HDBENCH clone	CPU / ビデオ / ディスクのベンチマークを計測	114
gtkfind	強力なファイル検索をGUIで行う	115
reXgrep	GUIで手軽に文字列検索	115
gtkdiff	2つのファイルの違いをカラー表示	116
bvi	vi風のキーバインドで使えるバイナリエディタ	116
Figaro's Password Manager	各種パスワードの暗号化と取りだしを行う	117
GAG	グラフィカルなブートマネージャ	117
ttftool	TrueType / Type1 フォントの設定をGUIで行える	117
Xalf	アプリ起動中に砂時計カーソルを表示	117

システム管理ツール

HardDrake	各種ハードウェア情報の表示と設定	118
Ethereal	パケットを監視するネットワークアナライザ	118
ntop	ネットワークトラフィックをWebブラウザで監視	119
KNETLoad	KDE用ネットワークモニタリングツール	119
Kmap	GUI操作でポートスキャンを行なう	120
GKrellm	CPUやディスクの負荷をグラフ表示	120
KCPULoad	CPUの動作率をグラフィカルに表示	121
LavaPS	プロセス情報をグラフィックで表現	121
GProc-applet	動作中のプロセスを監視するアプレット	121
gPS	プロセス情報をグラフィカルに表示	121

今回の特集で紹介するソフトウェアは、大別してXで動作するものとコンソールで動作するものがある。また、別途ライブラリやスクリプト言語が必要なものもある。ソフトウェアごとの動作条件の概要は、以下のアイコンで示しているので参考にしてほしい。

必要と思われるものについてはそれぞれの紹介記事の本文でもふれているが、動作環境や利用条件の詳細については、各ソフトウェアを使用する前に、ホームページの掲載内容やパッケージに同梱されているドキュメントをよく読んで、必ず確認するようにしよう。

- X** X Window System上で動作する
- コンソール** コンソール上で動作する
- GTK+** GTK+ツールキットが必要
- GNOME** GNOMEライブラリが必要
- KDE** KDEライブラリが必要
- Qt** Qtライブラリが必要
- Tcl/Tk** Tcl/Tk実行環境が必要
- Perl** Perl実行環境が必要
- Python** Python実行環境が必要
- Mesa** OpenGLライブラリMesaが必要
- Emacs** Emacsが必要
- 日本語可** 日本語が利用可能、もしくは日本語化パッチが存在する

ペンギンが雪山を滑り降りる3Dアクションゲーム

Tux Racer

バージョン: 0.12.1

ライセンス: GPL

<http://tuxracer.sourceforge.net/>

X Mesa Tcl/Tk

Linuxのマスコットであるペンギン「Tux君」を操作して雪山を滑り降りる3Dアクションゲーム。まだ未完成でラップタイムを保存することはできないが、雪や氷の上を猛スピードで滑降し、勢いをつけてジャンプするときの迫力はなかなかのものだ。変化に富んだコースが9種類用意されている。コース用データはTux Racer本体とは別のファイルで配布されているので注意しよう。RPMパッケージを利用する場合は、「データ 本体」の順番でインストールする必要がある。

起動するとコース選択画面が表示される(画面1)。現在のコースの全容が表示されており、n/pキーで順次、1~9キーで直接コースの切り替えが可能だ。mキーを押すと、現在のコースが左右反転したミラーコースに変わる。sキーを押すとプレイ開始だ。

Tux君がトコトコと登場して、雪山を滑降し始める(画面2)。滑降中は、jキーとlキーで左右に向きを変えられるが、スピードがついていると急には向きが変わらないし、地面の状態(雪、氷、土など)にも影響される。木や山に衝突してスピードが落ちてしまったら、kキーでパドリングして勢いをつけよう。衝突を避けるためにスペースキーでブレーキをかけることもできる。また、1~3キーで視点を変更でき、通

常の視点(1キー)のほか、Tux君により接近した視点(2キー)やTux君が見えない視点(3キー)でプレイできる。

無事にコースの端まで到達するか、あきらめてEscキーを押すとゲームオーバーだ。途中で空高くジャンプしたり、剣山のような岩の間をすり抜けたり、先の見えない暗闇の中を滑り降りたりと選択したコースによってさまざまなパリエーションを味わえる。なお、設定ファイル(/.tuxracer)を書きかえることにより、画面サイズや画面表示に関する設定、操作に使うキーなどをカスタマイズできる。Tux君の向きを通常とは逆にすることも可能だ。



画面1 コース選択画面では、コース全体の構造が表示される

画面2 コースの先を読みつつ、木や岩の間をすり抜けよう



ライトサイクルで爆走する3Dアクションゲーム

glTron

バージョン: 0.59

ライセンス: GPL

<http://www.gltron.org/>

X Mesa

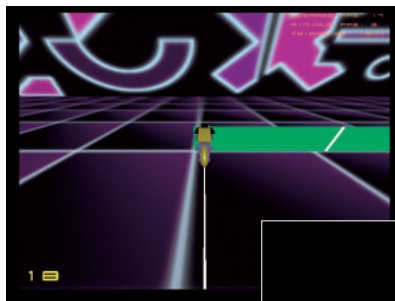
世界初のCG映画「TRON」に登場するライトサイクルゲームをX上で実現した3Dアクションゲーム。四角いアリーナの中で、未来のバイク「ライトサイクル」を互いの航跡に接触しないように走らせるサイバーゲームだ。なお、3D表示にはOpenGL(またはMesa)を使用するが、オプション設定でテクスチャなどの画面効果をすべてオフにすれば、3Dアクセラレータなしでもプレイ可能だ。なお、Esoundサウンドドライバ(esd)を利用している環境では、起動前に“esdctl off”として、機能を一時停止しておく必要がある。

タイトル画面から[GAME] [START GAME]を選択するとプレイ画面に切り替わる。最初にF10キーを一回押して、カメラモードを「ライトサイクル後方」に変更するとプレイしやすい。スペースキーを押すとプレイ開始だ。

ライトサイクルは、aキーで左、sキーで右に方向を変える。アリーナの壁や、ライトサイクルが通った跡に作られる光の壁に衝突しないように走り続けよう(画面1)。衝突するとライトサイクルは破壊され、最後まで残ったプレイヤーが勝者となる。プレイヤーは同時に4名まで参加でき、通常はプレイヤー1を人間が受け持つ。

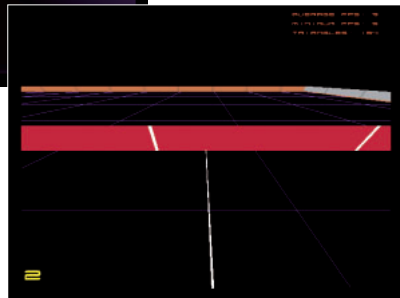
テクスチャーなどの画面効果や画面解像度はタイトル画面の[VIDEO]で、BGMや効果音については[AUDIO]でそれぞれ設定できる。また、[GAME SETTINGS]の「SHOW 2D MAP」をオンにして、現在の壁の状態を2Dマップに表示するように設定すると、ぐっと走りやすくなる。操作に使用するキーを変更することもできる。

3Dアクセラレータのない環境でプレイする場合は、初期設定のままでは満足な速度が得られない。“gltron -gwtbmx1”とすると、あらかじめテクスチャなどの画面効果をオフにした状態で起動するので、なんとかプレイできる速度になるはずだ(画面2)。



画面1 ライトサイクルを壁にぶつからないように走らせる!

画面2 この画面なら3Dアクセラレータがなくてもプレイ可能だ





雪山を舞台にした3Dレースゲーム

XRACER

バージョン：0.96.6

ライセンス：GPL

<http://xracer.annexia.org/>

X

Mesa

雪山に設置されたコースを浮遊式クラフトで駆け巡る3Dレースゲーム。コースやクラフトのデザインはシンプルだが、低い視線がもたらすスピード感はなかなかのもので、自動操縦や加速といったパワーアップアイテムも用意されている。なお、OpenGL互換ライブラリのMesa 3Dを利用して3D表示を行うため、実用的な速度でプレイするには、3Dアクセラレータ（Voodoo系、Riva、GeForceなど）が必須だ。

XRACERは、ソースとデータが別々に配布されている。実行の際は、環境変数XRACER_HOMEにデータを展開したディレクトリを「export XRACER_HOME=/usr/local/lib/xracer」などとして設定する必要がある。ライブラリ関係では、Mesa 3.0本体とGLUTが必要だ。これらのライブラリは、手持ちの3Dアクセラレータを利用する状態でビルドされている必要がある。詳細は「The Mesa 3D Graphics Library」（<http://www.mesa3d.org/>）などを参照されたい。

起動するとメニュー画面が表示され、Sキーでプレイ開始だ。画面サイズ（初期設定は640×480ドット）は、起動時に-sオプションで変更できる。たとえば、1024×768ドットの画面でプレイするには、「xracer -s 1024x768」とすればいい。なお、プレイ直前に異常終了し

てしまう場合は、サウンドドライバ（esd）の有無を確認しよう。

操作にはキーボード・マウス・ジョイスティックを利用できる。細かなハンドル操作を行うにはマウスかジョイスティックが必須だ。マウスを使うには、プレイ中の画面でDキーを一度押せばいい（再度押すとキーボード操作に戻る）。マウスの左右の動きがハンドル操作、左ボタンがアクセル、右ボタンがブレーキだ。

降りしきる雪の中を猛スピードで走り抜けるには、コースを熟知していることはもちろん、コース内によくつが設けられている加速ゾーンやアイテムゾーンをうまく通過し、それらを活用する技術も重要だ。アイテム（シールド・加速・自動操縦など）を取得すると、そのシンボルが画面上部に表示され、スペースキーを押した時点で使われる。レースは同じコースを3周するまで続けられ、ゴール時にあなたの順位や合計タイム、ラップタイムなどが表示される。今後はネットワークへの対応なども行われる予定だ。



画面1 激しい吹雪の中、猛スピードでコースを駆け抜ける

ペンギンが主人公になったレミングスのクローンゲーム

Pingus

バージョン：0.4.0

ライセンス：GPL

<http://pingus.seul.org/>

X

昔懐かしいゲーム、レミングスのクローン。次々に現われるペンギンを制限時間内に出口から脱出させるのが目的だ。

デフォルトでは、フルスクリーンモードで起動するが、「-F」オプションを与えると、ウィンドウモードで起動する。動作は、ウィンドウモードの方が若干安定しているようだ。

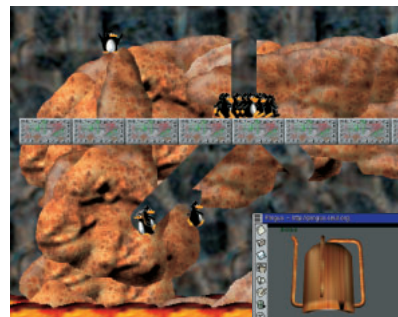
起動後に画面の出力先を聞かれるので、「&」をつけてバックグラウンドで起動してはいけない。出力先は、ドキュメントによれば「X、svgalib、フレームバッファ」が選べるようだが、実際に選べるのはXだけなので、「x11」と入力しよう。

[World]をクリックすると、ステージを選ぶことができる。まず最初は、簡単な「Tutorial」から始めて、コツをつかむといいだろう。スタートさせると、上からペンギンが降ってくる。彼らは、何も指令を受けてないうちは、ひたすらまっすぐ歩き、障害物にぶつくと折り返す習性がある。画面左に表示される指令をマウスで選択し、次にターゲットのペンギンをクリックすると、その指令を与えたことになる。指令には、仲間をせき止める、自爆する、橋を作る、穴を掘る（真下、真横、斜め）、壁を登る、飛ぶ（高いところから安全に着陸する）の6つが

ある。

ペンギンが出口まで行く道作りが、完璧にできたらと思ったら、スペースキーを押すとペンギンの動作が一気に加速し、早くクリアすることができる。ここでもし間違っていると、ペンギンの大量虐殺となってしまい、動物愛護団体から抗議を受けることになってしまうのであった。

以前のバージョンと違い、0.4.0からライブラリをスタティックリンクするようになったため、今まで必要だったライブラリについても、必要がなくなった。Webサイトにあるバイナリのtarボールを展開するだけで、すぐに始めることができるぞ。



画面1 なかなか思ったようには動いてくれないペンギンたち



画面2 マップを自分で作って遊ぶこともできる。ペンギン大虐殺に飽きたら、次はオリジナル面だ！

エメラルド、ダイヤモンドをゲット!

Rocks'n'Diamonds

バージョン : 1.4.0pl1

ライセンス : GPL

<http://www.artsoft.org/rocksndiamonds/>

X

穴を掘りながらエメラルドやダイヤモンドなどの宝石を集めるアクションパズルゲーム。このゲームの作者は「Mirror Magic」や「affenspiel」といったパズルゲームと同じである。

システムは異なるが「Bouldar Rush」や「Sokoban」と同系統で、とにかく敵の攻撃や障害物をうまく回避しながら、宝石を集めていくことが目的だ。具体的には、キャラクターを操作してフィールドに点在するすべてのエメラルドやダイヤモンドを集めることにより、フィールドのどこかにある扉が開くので、そこにたどり着くことを目的とする。扉をくぐるとその面はクリアとなり、新しい面がロードされる。

地中には宝石だけではなく、いろいろな障害物が埋まっている、たとえば、岩の下を掘ると下に落ちてくる。岩の下を掘るときは下敷きにならないように一目散に逃げよう。爆弾を仕掛けることもできるが、下手をすると8方向に爆発してしまうので、さらなる注意が必要だ。各ステージには制限時間が設けられていて、決められた制限時間内にすべての宝石を集めなければならない。

ゲームを起動した後の初期画面では次の操作を行うことができる。

NAME : プレイヤー名の変更 / 設定

LEVEL : ゲームレベルの選択

HALL OF FAME : ハイスコアの表示

LEVEL CREATOR : レベルエディタの実行

INFO SCREEN : ゲームに関する情報の表示

START GAME : ゲーム開始

SETUP : サウンド、スクロール、記録などの設定

QUIT : ゲーム終了

バックグラウンドミュージックの設定は、画面右側の操作パネルでも行えるようになっている。ゲーム開始後のキャラクターの操作は、カーソルキーでキャラクターの移動、右Shiftキーでスナップ、左Shiftキーでダイナマイトの設定となる。

アクションパズルゲームとしては出現する障害物、アイテム、敵キャラの種類も多く、かなり難易度は高い。4人までのマルチプレイヤー機能も持っており、複数プレイでの対戦もなかなか盛り上がりそうだ。



画面1 障害物に気をつけながら宝石を集めよう

崩れるタワーをジャンプ! ジャンプ!

FALLING TOWER

バージョン : 2.6

ライセンス : フリー

<http://mmm.uec.ac.jp:8081/club/koken/kudoh/soft/xjump.html>

X

崩れるタワーを足場から足場へとジャンプして上へ上へと登っていくアクションゲーム。足場が上から下へとゆっくり落ちてくるので、その間をキャラクタをジャンプさせながら飛び移っていく。足場を踏み外すと落下してしまいゲームオーバーとなる。足場とキャラクタしか登場しない非常にシンプルな構成である。

作者のホームページには、NetBSDまたはFreeBSDでしか動作確認していないと書いてあるが、Vine Linux 2.0では問題なくコンパイルできた。ただ、オリジナルソースだと999段までしかカウントされないので、上級者は注意が必要かも。筆者はこのゲーム、異常に下手なのでオリジナルの3桁で問題ないが.....(^_^)。そこで、JG-1.0に収録したバイナリパッケージでは、10桁までカウント可能のようにパッチをあててある。

ゲーム開始後のキーバインドは次のようになっている。

- 、j、4 : 左移動
- 、l、6 : 右移動
- 、z、スペース、8 : ジャンプ

筆者のお勧めキーポジションは、右手人差し指を キー、右手薬指を キー、右手中指を キーに置く方法。高得点に挑戦したい人はどうぞ。

少しやってみた感じだと、心持ち床が滑りやすいような気がするので、最初はまずキャラクタの操作に慣れることが重要だ。初めのうちはうまくコントロールできず、10段くらいしか記録が伸びないかもしれないが、ここでめげずとにかく操作に慣れるまでやり続けることが肝要。筆者は根性がないのと、この手のアクションゲームは苦手なのでなかなか高得点が出せずにめげているが、1000段オーバーという信じられない達人もいるので、ぜひこれを目標にして頑張ってもらいたい。

また、左右の壁をうまく使ったジャンプや二段ジャンプを使うと結構高くジャンプできる。ゲームをスムーズに進めるには、この技をマスターしておいてもらいたい。

README.eucファイルに記述されている内容にしたがって、picture.xpmというファイルを準備すれば、自分の好きな画像データをキャラクタとして使うこともできるようだ。ただ、ぱっと見たところ実行ファイルの再コンパイルが必要のようだ。



画面1 とにかく「慣れ」が肝心。根気よくチャレンジしよう



リンゴを落として敵をやっつけろ!

XMris

バージョン: 4.0.5

ライセンス: GPL

<http://www.cs.bris.ac.uk/~nathan/xmrisc/>

X

リンゴを落として敵をやっつけるゲーム。妖精を動かすことによって地中に穴を掘ることができる。ゲームに出現する敵キャラは妖精を追いかけ、妖精は敵キャラにぶつかると死んでしまう。妖精はボールを敵キャラにぶつけるか、地中に埋まっているリンゴをうまく敵キャラの上に落とすことで敵キャラを倒すことができる。

このゲームの目的は、ガーデンと呼ばれる面をクリアしていくこと。ガーデンのクリア条件は、次のいずれかである。

- 地中に埋まっているチェリーを全部取る
- 出現する敵キャラを全部倒す
- 途中で出現するダイヤモンドを取る

基本的には を目標として操作するのが普通だが、 または が成立しそうときは、迷わずそちらを選択しよう。また、ゲーム途中で出現するボーナスキャラは高得点なので、忘れずにゲットしよう。

コンパイル、インストールは通常のパターンでほとんど問題なくできると思うが、デフォルトの設定だと日本語キーボードでは操作しづらい。

コミカルな風船割りアクションゲーム

Circus Linux

バージョン: 1.0.1

ライセンス: GPL

<http://www.newbreedsoftware.com/circus-linux/>
<http://www.devolution.com/~slouken/SDL/> (SDL)

X

Mesa

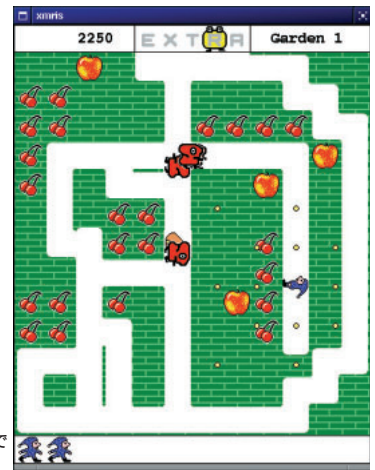
昔懐かしいアーケードゲーム「サーカス サーカス」によく似たゲーム。シーソーを利用して2人のピエロを交互にジャンプさせ、画面上方を左右に流れる風船を割っていく。ルールは単純ながら、コミカルなキャラの動きや効果音、BGMが気分を盛り上げてくれる完成度の高いゲームだ。設定によっては、ピエロを跳ね返す障害物を追加したり、2人で交互にプレイすることもできる。動作にはクロスプラットフォームなマルチメディアライブラリのSDLとSDL Image、SDL Mixerの各ライブラリが必要だ。

起動すると、コミカルなBGMとともにタイトル画面が表示される(画面1)。ソロプレイなら「ONE PLAYERS」、2人でプレイするなら「TWO PLAYERS」(交互に独立した面をプレイ)か「TWO PLAYERS COOP」(相手の続きをプレイ)をマウスでクリックすればいい。オプションとして、「BARRIERS」(障害物が現れる)、「BOUNCY BALLOONS」(風船に当たると跳ね返る)、「CLEAR ALL」(すべての風船を割るまで新しい風船が現れない)を設定できる。複数の組み合わせも可能だ。なお、ハイスコアは、これらのオプションの組み合わせ(8通り)のそれぞれについてトップのスコアだけが記録される。

日本語キーボードを使っている場合は、コンパイル前に、ソースファイルXmrisc-ad.srcの先頭部分を“ * Up:Up ”、“ * Down:Down ”、“ * Left:Left ”、“ * Right:Right ”というように修正したほうがよいだろう。

ボールを投げる際の注意点は、投げたボールは自分の「向いている方向」に飛ぶということ。敵キャラに追いつかれているときにボールを投げても前に飛んでしまい、敵キャラをそのボールでやっつけることはできない。また、ボールは連続して投げることはできず、一度投げると次に投げられるようになるまで少し時間がかかるので、タイミングを計って投げるようにしよう。

ゲームの途中でアルファベット型の敵キャラが出現する場合がある。この敵キャラは“ EXTRA ”のいずれかのアルファベットの形をしており、EXTRAの5種類すべての形の敵キャラを倒すと、使用できる妖精の数が1つ増えるので、アルファベット型の敵キャラが出てきたら積極的に狙っていこう。ただ、他の敵キャラと比べて少し賢いので、なかなかEXTRAをそろえるのは結構難しいかもしれない。



画面1 独特の絵柄のキャラが素敵である

シーソーの操作にはマウスを使用する。左クリックでピエロが落ちるので、マウスを左右に移動させて受け止めよう(画面2)。受け止めた反動でシーソーの上にいるピエロがジャンプする。こうして交互にピエロをジャンプさせ風船を割っていくのだ。一度に多くの風船を割るには、できるだけシーソーの端のほうでピエロを受け止めて高くジャンプさせる必要があるが、それだけミスする危険も増す。なお、キーボードによる操作も可能だが、シーソーを思い通りの位置に止めるのは難しいので、マウスで操作することをお勧めする。



画面1 風船が乱れ飛ぶタイトル画面。障害物などの設定も可能だ

画面2 シーソーを動かし、ピエロをジャンプさせて風船を割れ!

美しいグラフィックとサウンドのブロック崩しゲーム

Xboing II

バージョン : 2.4

ライセンス : フリー

<ftp://ftp.x.org/contrib/games/>

X

美しいグラフィックとサウンドが特徴のブロック崩しゲーム。パドルを左右に動かして、ボールをはじき返してブロックを壊していく。全てのブロックを壊すとラウンドクリアとなり、次のラウンドに進むことができる。

ゲーム中に、マウスの左ボタンをクリックすることで弾を発射してブロック壊すことができるので大変便利だが、弾がボールに当たるとボールを壊してしまうので弾を発射するときは気をつける必要がある。

ゲーム中の主なキー操作は次の通り。

スペース : ゲーム開始

j / : パドルを左に移動

l / : パドルを右に移動

k : 弾を発射

ESC : 初期画面に戻る

i : アイコン化

h : スコア画面での表示切り替え

p : ポーズ

s : 画面効果のオン / オフ

d : ボールの破棄

c : スコア画面での表示の切り替え

w : スタートのレベルの設定

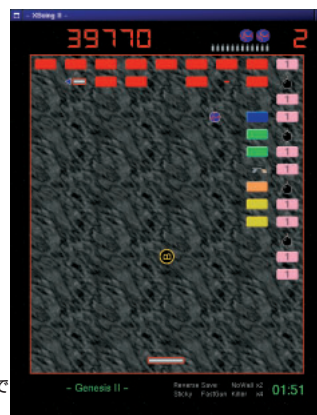
q : ゲーム終了

ゲーム中には、いろいろなアイテムが出現する。たとえば、「Extra Ball」や「Multi Balls」などが出現したら、ボールが増えるので、積極的に狙っていこう。逆に「Instant Death!」にボールが当たるとボールが壊れてしまう。また、「Reverse Control」に当たると操作の左右が逆になり一気に難しくなるので、できるだけ当てないように頑張ろう。

ゲームをしていてあまりにもマシンの性能が高くゲームが速すぎる場合は、「-speed」オプションを使えばゲームの速度を調整できるので、マシンが速すぎたり遅すぎたりする人は試してみてください。

“-sound”オプションを指定するとサウンド機能が有効になるが、ISA形式のサウンドカードを使う場合など、性能の低いマシンで遊ぶ場合は、サウンド機能を有効にするとかえって動作がギクシクして遊びづらい場合があるので注意が必要だ。

画面1 なつかしの「ブロック崩し」がXで楽しめる



ネットワーク3Dタンクバトルゲーム

BZFlag

バージョン : 1.7d

ライセンス : GPL

<http://bzflag.sourceforge.net/>

X

Mesa

障害物の置かれたフィールド内で戦闘を繰り広げるマルチプレイヤー3Dタンクバトルゲーム。ローカルなネットワーク上でサーバを起動して対戦するほか、インターネット上のサーバに接続してプレイすることも可能だ。

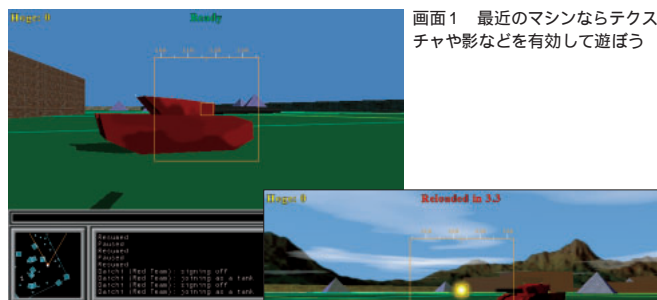
起動するとタイトルとメニューが表示される。まずは、[Options]を選択して、画面表示に関する設定を変更する。初期設定値はかなり抑え目になっているが、最近のマシンなら「On/On/On/On/Linear/Low/On/On」程度の設定は軽くこなせるはずだ(画面1)。3Dアクセラレータカードがあれば、最高画質に設定しても快適にプレイできる(画面2)。

ローカルネットワーク内で対戦する場合、マシンのどれかひとつでサーバを起動する必要がある。サーバ設定画面では、ゲームの種類や最大プレイヤー数などを設定する。あとは、各クライアントでサーバのホスト名を指定してゲームを開始すればいい。

プレイ中の画面にはフィールド内のようすが3D表示され、左下にはレーダー、その右にはシステムメッセージや会話のログ、右下には各チーム(色別)のプレイヤー数が表示される。タンクの操作にはマウスを

使用する。カーソルを画面の上下に動かすと前進/後退、左右に動かすと旋回だ。移動速度はカーソルが画面中央からどれだけ離れているかに比例する。レーダーを使って敵を視界に捕らえたら、画面中央に来るように移動しつつ左ボタンで弾を発射しよう。なお、連射できる弾数(初期値は1発)を超えると、再充填に3秒を要する。

一緒に遊んでくれる仲間がいない場合、起動時に“-solo 台数”というオプションを付ければ、指定した台数のロボット戦車がゲームに登場する。また、サーバ検索機能を使ってインターネット上で運用されているサーバを一覧表示し、選択したサーバに接続してプレイすることも可能だ。



画面1 最近のマシンならテキストチャットや影などを有効して遊ぼう

画面2 3Dアクセラレータがあれば最高品質の画面でプレイ可能だ



360度バリバリシューティング野郎

Xkobo

バージョン: 1.11

ライセンス: フリー

<http://seki.math.hokudai.ac.jp:20080/xkobo-current.html>

X

360度スクロール可能な面クリアタイプのシューティングゲーム。ステージごとに設置された要塞をビームを使って攻撃・破壊していく。マップ上のすべての要塞を破壊すると、その面をクリアしたことになる。要塞からは多くのビームが発射されるし、マップ上に登場する敵機からも激しく攻撃されるので、とにかく攻撃と防御を効率的に。

カーソルキーの操作で自機の方向を変えられる。また、キーとキーを同時に押すことで右斜め上45度に方向を変えることも可能。ただし、自機を静止させることはできないので、常にどちらかの方向に向かって進むことになる。ビームの発射はShiftキーまたはマウスの左ボタンクリック。これで自機の前方向にビームが発射される。Shiftキーを押せばなしにすることで連射が可能なので、とにかく方向を変えながらビームを連射して敵機や要塞を破壊しよう。

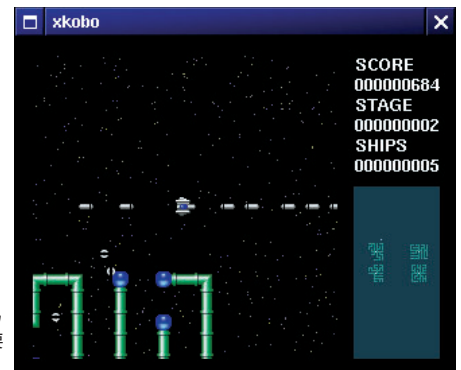
このゲームの目的である要塞を破壊するルールであるが、敵の要塞は中心部から枝葉が伸びており、その先に青い玉が設置されている。自機をうまくコントロールして、青い玉にビームを当てることで、その玉と玉につながっている要塞の枝の部分が爆発する。青い玉をいくつか破壊すると、要塞の中心部にあるピンクの玉を攻撃できるようになる。こ

のピンクの玉を破壊すれば、要塞全体を破壊できる。画面右のレーダーにすべての要塞と自機の位置がマップ表示されるので、これを見ながら全要塞を効率的に破壊しよう。

ゲームを始めてしばらくすると白い敵機が出てくる。この敵機は直線的に動き、かつ非常にスピードが速いので、攻撃するか逃げるかを即座に判断すべし。また、3面あたりから緑の玉が出現する。この緑の玉は、ビームを当てても破壊できないので、気をつけてほしい。

このほかにもリング状の敵機など、とにかくいろいろな敵が登場するのでじゃんじゃん撃ちまくって敵機をやっつけよう。ただ、このゲームの目的は敵機をやっつけることではなく要塞を壊すことにある。敵機を深追いすることは禁物だ。ピンクの玉を破壊することを常に意識して、そのために邪魔となる青い玉や敵機を破壊していくのがコツ。

作者のホームページから入手できるソースコードはC++で記述されており、各ディストリビューション上でコンパイルが可能だ。JG-1.0にも収録されているので、Vine Linux 2.0 ユーザーであれば、コンパイルの必要なくバイナリファイルを直接入手することも可能だ。



画面1 右下の全体マップで白く見える部分が要塞のコアだ

縦スクロールの定番シューティングゲーム

Cosmic Shooter

バージョン: 0.94

ライセンス: フリー

<http://jg.linux.or.jp/>

X

オールドツクスな縦スクロールのシューティングゲーム。自機を左右に動かして、上から降ってくる敵を倒す。敵を攻撃するための武器はミサイルとボムがあり、ミサイルは無制限に使えるが、ボムは画面上の敵をすべて倒すことができるので使える数に制限がある。

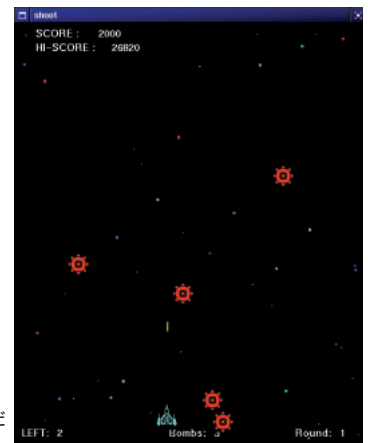
このゲームはラウンド制を採用しており、1ラウンドは8レベルから構成される。レベルごとに敵が変わっていき、ラウンドの終わりに登場するボスを倒すことによりラウンドクリアとなって、次のラウンドへと進むことができる。自機はゲーム開始時には3機だが、スコアが2万点、4万点、8万点、16万点、32万点に到達するごとに1機ずつ増える。すべての自機を失うとゲームオーバーとなる。ラウンドをクリアした時点でラウンド数×1万点のボーナスが入るので、まずは集中してラウンドクリアを目標に頑張ろう。

コマンドラインで“shoot”と入力することで遊ぶことができるが、起動オプションを指定することもできる。“-delay 数字”はゲームのスピードの調節、“-mono”で白黒画面になり、“-nostars”で背景画面の星が非表示になる。また、“-round 数字”または“-level 数字”として開始ラウンドや開始レベルを指定できる。

ゲーム中の主なキー操作は次のとおり。

- j / / 4 : 自機を左に移動
- l / / 6 : 自機を右に移動
- s / x / スペース : ミサイル発射
- a / z / k : ボム発射
- r : 連射モード切り替え
- e : 連射スピードアップ
- w : 連射スピードダウン
- p : ポーズ
- q : ゲーム終了

攻略するには、ゲーム中に登場するパワーアップアイテムを効率的にゲットすることと、ボスに対してはヒットアンドアウェイ戦法（上にいるときにミサイルを撃ち込んで、下にいるときは逃げる）をとることが重要となる。



画面1 シンプルなシューティングだが、それだけにアツくなる

ミサイルを迎撃して街を守れ!

Missile Command

バージョン: 0.99.7

ライセンス: GPL

<http://missile.sourceforge.net/>

X

空から降ってくるミサイルを撃ち落として街を守るゲーム。システムは非常にシンプル。ゲームを開始すると、空からミサイルがじゃんじゃん降ってくるので、右、中、左に設置された砲台から迎撃ミサイルを发射して、ミサイルを撃ち落として街が破壊されるのを防ぐのが目的だ。

初期画面でEnterキーを押すとゲーム開始。空からミサイルが降ってくるので、マウスポインタを動作させてターゲットを動かし、目標位置を決めよう。そこでマウスの左ボタンをクリックすると、設定した目標位置に向かって迎撃ミサイルが发射され、目標位置に到達すると爆発する。爆発の範囲内に敵ミサイルがある場合は、それが爆発して得点が加算される。迎撃ミサイルの爆発に巻き込まれなかったミサイルは地上に到達する。そのとき、地上に街や砲台があると爆破されてしまう。うまく迎撃ミサイルを操作して地上の街を守り抜こう。

各砲台から发射できるミサイルの数は限られていて、撃ちまくるとすぐなくなってしまう。また、砲台は合計3カ所あるので、どのミサイルをどの砲台から狙うかを即座に決めることが重要だ。

1つの面で降ってくるミサイルの数が決まっており、その時点で、街

が最低1つでも壊されずに残っていれば、面クリアとなり得点が加算される。すべての街が壊された時点でゲームオーバーになる。

ゲーム中のキー操作は次のとおり。

マウス移動: 目標位置の移動

左ボタンクリック: 左の砲台から迎撃ミサイルを发射

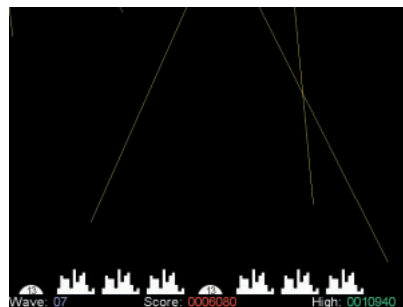
中ボタンクリック: 中央の砲台から迎撃ミサイルを发射

右ボタンクリック: 右の砲台から迎撃ミサイルを发射

Ctrl / Shift / スペース: 一番近い砲台から迎撃ミサイルを发射

スコアアップを目指すには、とにかくミサイルをできるだけ画面の上方で壊すことを心がけよう。そのためには、マウスポインタを素速く動かすことと、迎撃ミサイルを素速く发射することが重要である。マウスクリックによる迎撃ミサイル发射は、ボタン選択を迷いやすいのでお勧めできない。やはり迎撃ミサイル发射は迷わずCtrlキーで。

なお、コンパイルにはSDLライブラリとSDL_imageが必要となる。



画面1 とにかくすばやいマウス操作が必要だ

ネットワーク対応の素敵ゲーム

Batalla Naval

バージョン: 0.74.0

ライセンス: GPL

<http://batnav.sourceforge.net/>

X

GTK+

Batalla Navalではプレイヤーは互いに相手の見えない軍艦の位置を当てていく。そう、かつては潜水艦ゲームと呼ばれていたゲームだ。

サーバ/クライアント方式のマルチプレイヤーによるネットワーク対戦をプレイするには、最初に専用のサーバを起動させておく必要がある。コマンドラインで“gbnserv”とすれば立ち上がる。プレイ人数は最高で7人まで。対戦相手としてAIも用意されていて、1人で遊ぶことも可能だ。クライアントは“gbcnclient”で立ち上げるが、このときサーバの名前とポートを聞かれるので、事前にそれらのデータをほかのプレイヤーに教えておく必要がある。

ルールはいたって単純だ。それぞれの軍艦は大きさに違いがあり、いくつかのマス占有して、すべてのマスが見つされるとその艦は撃沈となる。すべての軍艦を撃沈されたら負けだ。画面左には自分の艦の配置マップが、右には他のプレイヤーのそれぞれに対応した攻撃用のマップが表示される。プレイヤーごとに順番に、右側に表示されたそれぞれのプレイヤーのマップのマスをクリックしていくと、もしそこに艦が存在すれば艦のマークが出る。そして、その艦が撃沈することができればxマークが付く。各プレイヤーのマップは上部にあるタブをクリ

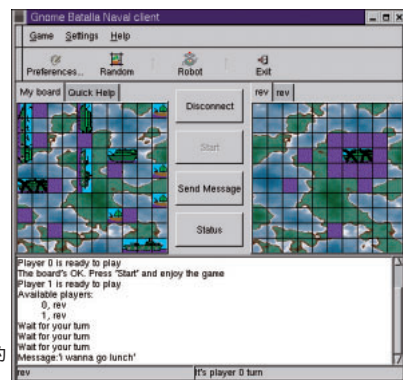
ックすると表示されるが、3人以上でプレイする際にも、自分の番には1人のマップの1カ所しか調べることができない。どの相手を狙っていくかも状況を判断しながら、変えていく必要があるだろう。

簡単なメッセージ送信システムも用意されており、威嚇やはったりをかますこともできる。もっとも、これはすべてのプレイヤーに送信されてしまうので、集中攻撃を行うなどといった、秘密の作戦会議には向かない。また、下部のステータスウィンドウには、各プレイヤーの状況や、メッセージが表示される。

派手な画面などはないが、そのぶんネットワークの通信量は少ないので回線が細くても問題なく遊べる。遠くの仲間と遊んだり、ちょっとした息抜きにはいいだろう。

0.74.0ではGNOME専用となっているが、1つ前のバージョンの0.60であれば、Windowsやncursesによるターミナルによるクライアントも存在する。なお、最近は他のホストからのアクセスを制限しているディストリビューションもあるので、うまくサーバに接続できない場合には、サーバとなっているマシンのアクセス制限をチェックしてみよう。

画面1 敵の心理を読みつつ、的確に攻撃しよう。





戦略シミュレーションゲーム CivilizationIIのクローン

Freeciv

バージョン: 1.11.0

ライセンス: GPL

<http://www.freeciv.org/>

X

GTK+

有名なマルチプレイヤー戦略シミュレーションゲーム「CivilizationII」のクローン。プレイヤーは、文明を発展させながら、他民族との外交と抗争を行い、世界制覇を目指す。ネットワークによる最大14人のマルチプレイヤーによる同時プレイができる。インターネット上には多数のFreecivサーバが立ち上がっており、世界中の人とプレイすることができる。もっとも最初は、自分でサーバを立ち上げて用意されたAIのプレイヤー相手にプレイして慣れていくほうがいいだろう。

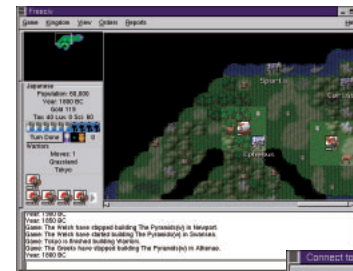
まずcivserverを起動しておき、その後civclientを立ち上げる。そして[Connect]をクリックし、ローカルのcivserverに接続する。AIプレイヤーを使うのなら、civserverに“create foo”と入力すれば、fooというAIプレイヤーが作成される。civserverにstart命令を与えれば、ゲームが始まる。

あとは、画面上に表示されるキャラクターを移動させて、探検させ、都市を建築し、敵を倒していく。もちろん、単純に領土を広げるだけではなく、税金を取って資金を作ったり、人口を増加させて兵士を増やし、その能力をあげていくといった作業が必要になってくる。日本語のマニュアルがないのでちょっとつらいが、繰り返しプレイしているう

ちに、コマンドの使い方もわかってくるだろう。重要なできごとにはダイアログボックスが表示されるし、画面に表示されない部分で起こったことについては、下部のメッセージウィンドウに表示される。

ゲームに慣れてきたら、マルチプレイに挑戦してみよう。civclient起動時のダイアログで[Metaserver]を選んで「Update」すると、インターネット上で立ち上がっているサーバが表示される。ここでサーバを選んで「Connect」すると、世界中の人との対戦を楽しむことができる。

Red Hat、Debian、Slackwareといったディストリビューションのバイナリパッケージも用意されているが、バージョンが古いものもあるので、最新バージョンを使いければ、ソースからビルドするようにしよう。クライアントはLinuxだけでなく、WindowsやAmiga用もある。



画面1 敵を倒し、領土を広げ、国力を増強せよ



画面2 腕に覚えがあるなら、世界中のプレイヤーに挑戦してみよう

プログラムしたロボットを戦わせるシミュレーションゲーム

RealTimeBattle

バージョン: 1.0.4

ライセンス: GPL

<http://realtimebattle.sourceforge.net/>

X

GTK+

日本語

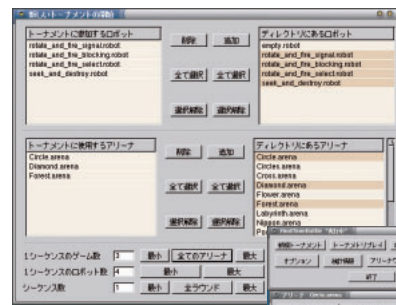
プログラムしたロボットをアリーナで戦わせるシミュレーションゲーム。1対1の対戦から、最大120体が同時に戦うバトルロイヤルまで、さまざまな条件での戦闘が可能だ。ロボットは、実際の物体のように慣性や摩擦の影響を受け、エネルギーを消費する。専用言語ではなく、C、C++、Perl、Java、Schemeなど、たいいていのプログラム言語でロボットを作成できるのが最大の特徴だ。日本語カタログ(ja.po)が標準で含まれており、メニューやダイアログが日本語で表示できる。

RealTimeBattleでは、同じロボット同士の一連のゲームを「シーケンス」、シーケンスの集合を「トーナメント」と呼ぶ。トーナメントを開始するには、トーナメント条件を設定するウィンドウで、参加するロボットの種類と舞台となるアリーナを選択する(画面1)。「1対1の総当り戦」や「全ロボットのバトルロイヤル」なども可能だ。

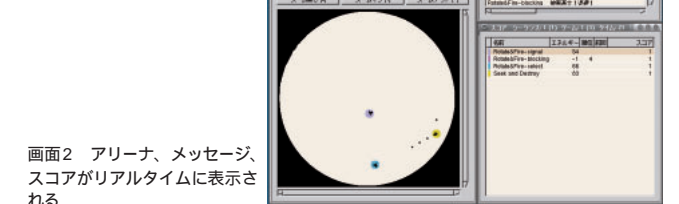
トーナメントを開始すると各シーケンスのゲームが順番に行われ、各ロボットのプログラムが起動して戦闘が行われる(画面2)。ロボットはタイヤ付きの乗物で、周囲の情報を得るレーダーと、敵のロボットなどを破壊するキャノンを備えている。壁や敵のキャノンに当たらないように移動しながらキャノンを撃ち、敵のエネルギーを0にして破壊すると

というのが一般的な動作だ。アリーナ内のロボットが1台以下になるか、タイムアウトになるとそのゲームは終了する。

ロボットのプログラムは、RealTimeBattle本体から子プロセスとして起動され、標準入出力(stdin/stdout)を利用して本体とメッセージ交換を行う。このため、たいいていのプログラム言語でロボットを作成できる。メッセージの種類や受け取り方などについては、Webサイトのマニュアルを参照されたい(日本語マニュアルもある)。「MrFrost」(<http://artscene.fluxus.lt/rdb/>)には、ユーザーが作ったさまざまなロボットが登録されており、ダウンロード可能だ。



画面1 参加するロボットや舞台となるアリーナを選択する



画面2 アリーナ、メッセージ、スコアがリアルタイムに表示される

宇宙船で自由に宇宙を飛び回るゲームシステム

XShipWars

バージョン : 1.33g

ライセンス : GPL

<http://fox.mit.edu/xsw/>

X

複数のプレイヤーが宇宙船をリアルタイムに操作するサーバ/クライアント方式のマルチプレイヤーゲームシステム。宇宙船や宇宙ステーション、惑星、ワームホールなどが美しくデザインされ、効果音も凝ったものが使われている。標準でスタートレックTNG/DS9の舞台となるユニバース（宇宙）が用意されているほか、付属のエディタを使って自分でユニバースをデザインすることも可能だ。

本体のほか、基本データ、画像、サウンドがそれぞれ配布されている。本体のtarボールには、クライアントとサーバ、サーバ用モニタ、ユニバース作成エディタが含まれる。インターネット上のサーバに接続して遊ぶ場合はクライアントのみが必要となる。

クライアントを起動すると、ウィンドウが開いて美しいタイトルが表示される（画面1）。[CONNECT] ボタンを押してユニバースリストウィンドウを開き、接続するサーバをリストから選択しよう（[TWU] が [Dogstar] がお勧め）。サーバに接続すると、即座にプレイが始まる。プレイ中のウィンドウは、左側に自船のコンソール、中央に周囲の宇宙を表示するビュースクリーン、右側にスキャナのコンソール、下にメッセージボックスという構成になっている（画面2）。自船は / キーで

加減速、 / キーで方向転換する。また、Tabキーで周囲の物体がスキャンされ、順番に右のコンソールに表示される。武器は数字キーで切り替え、スペースキーで発射だ。

なお、XShipWars自体には特定の目的は設けられておらず、各ユニバースの設計者が目的を設定する。たとえば、[TWU] では基本的な戦闘方法の学習、[Dogstar] ではプレイヤー間のコミュニケーション（チャット）が目的だ。ローカルネットワークでサーバを立ち上げれば、こころゆくまで宇宙船の操作を練習したり、仲間内だけでプレイを楽しんだり、ユニバースの設定を自由に変更できる。



画面1 美しいタイトル画面が気分を盛り上げる

画面2 中央のビュースクリーンをはじめ、さまざまな情報が表示される



世界各地を飛行できるフライトシミュレータ

Flight Gear Flight Simulator (FGFS)

バージョン : 0.7.4

ライセンス : GPL

<http://www.flightgear.org/>

X

Mesa

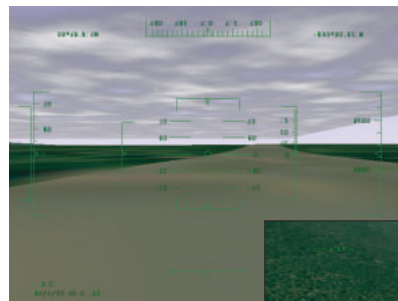
マルチプラットフォームで動作するオープンソースのフライトシミュレータ。ちゃんとしたフライトモデルを用いるシミュレータなので操縦は結構難しいが、そのぶんリアルな機体の挙動を楽しめる。また、オプションとして世界各地のシーナリーデータが用意されており、飛んでみたい地域のデータをダウンロードして追加できる。快適な速度でプレイするには、3Dアクセラレータカードが必須だ。

FGFSはソース一式と、ベースパッケージ、ドキュメントに分けて配布されている。このほかにOpenGL（またはMesa）、コンストラクションツールのSimGear（<http://www.simgear.org/>）、ゲームライブラリのplib（<http://www.woodsoup.org/projs/plib/>）が別途必要だ。

起動すると、現在時刻のアリゾナの空港からシミュレーションが開始される（画面1）。飛行機の操作には、キーボードがジョイスティックを使用する。 / キーでエレベータ（上下の向き）、 / キーでエルロン（左右の傾き）、Insert/Enterキーでラダー（左右の向き）を変更し、Page Up/Downキーでスロットル（加減速）を調整する。現在の状況はHUD（ヘッドアップディスプレイ）のインジケータやパネルの計器で確認できる。視点の変更やオートパイロットなども用意されている。

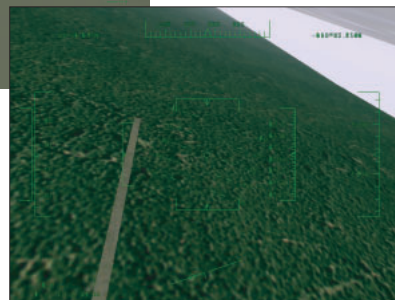
空港から離陸するには、スロットルを開いてスピードを上げつつ、ラダーを操作して滑走路に並行に走るように調整。離陸可能なスピードに達したら、エレベータを少しだけ操作して機首を上に向け、あとはこれを維持すればいい。こう書くと簡単そうだが、最初のうちは離陸するだけでも結構難しい（画面2）。

操縦を失敗して地面に墜落すると機体が逆さになって操作不能になるので、最初の状態に戻してやり直そう。なお、手軽に空の散歩を味わいたいなら、機体の高度や速度、日時などを起動時オプションで指定して、昼間の飛行の状態ゲームを始めるといい。



画面1 Flight Gearは本格的なフライトシミュレータだ

画面2 離陸したアリゾナの空港を上空から眺める





ペットを連れて、さあ冒険に出かけよう

Japanese NetHack

バージョン : 1.1.5

ライセンス : GPL

<http://www.jnethack.org/jnethack.html>

X

GTK+

日本語可

プレイするたびにダンジョンのマップが変化するロールプレイングゲーム。この手のゲームの決定版といえるNetHackを日本語化したものだ。GTK+を利用したグラフィカルな画面でプレイでき、メッセージが日本語化されているため、英語が苦手な人でも大丈夫だ。

まずは、プレイするキャラクタを選択する。キャラクタごとにクセがあり、初期装備や戦い方などが大幅に異なる。最初は、戦闘のしやすい野蛮人や洞窟人、ワルキューレなどがお勧めだ。

操作は1文字のコマンド入力で行う。上下左右への移動はviライクなk/j/h/lキーで行う(斜め方向の移動も可能)。進行方向に怪物がいる場合は、移動ではなく怪物への攻撃となる。このほかにも、よく使うキーが数多くある。はじめのうちはジャンル別に分類されたメニューでコマンド入力しながら、対応するキー操作を少しずつ覚えていこう。

階層別のダンジョンはだいたい25~30階の深さで、各階はいくつかの部屋が通路で結ばれている(画面1)。すべての部屋を探索しながら怪物と戦い、落ちてアイテムやお金を拾いつつ、下への階段を探す、というのが基本的な行動パターンだ。アイテムを売買する店や王座などの特殊な部屋、ノームの鉱山、城、ゲヘナといった「特別レベル」

も存在する。

また、地面に落ちているアイテムのうち、巻物や魔法書、薬、指輪などには仮の名前が付けられており、実際に使ってみるまで効能がわからない。さらに、使用結果を踏まえてプレイヤーが名付ける必要のあるアイテムもある。たとえば、装備している武器が青く輝いたら「武器に魔法をかける」(enchant weapon)巻物だ。

ダンジョンで出会うたいの怪物は攻撃的だが、中立的・友好的なものもある。特に、プレイ開始直後に側にいる犬(または猫)は、あなたとともに行動するペットで、怪物を攻撃してくれたり、呪われていないアイテムだけ拾ったり、店から品物をちょろまかしてくれる大切な存在だ。名前を付けて可愛がってあげよう(餌をやるのを忘れずに)。

行動の自由度が極めて高いNetHackには、ソースコードを読まないといけないような仕掛けが数多く隠されている。最近では日本語で書かれた攻略用の文書も充実しており、初心者なら「猿でもできるJNetHack」さらに深い知識が必要になったら「JNetHack Spoilers」(<http://www.ke.ics.saitama-u.ac.jp/matsuda/nethack/jspoiler/>)を参照しよう。



画面1 メインウィンドウにダンジョンのマップが表示される

ネットワーク対戦も可能な本格卓球ゲーム

Cannon Smash

バージョン : 0.4.2

ライセンス : GPL

<http://www.utmc.or.jp/nan/csmash/index.html>

X

Mesa

本格的な3D卓球ゲーム。プレイヤーのタイプが3パターン用意されており、多彩な戦術が要求されるスポーツとしての卓球を楽しめる。快適にプレイするには3Dアクセラレータが必須。コンピュータ相手のソロプレイと人間同士のネットワーク対戦が可能だ。試合形式でプレイするモードのほか、ラリーを続けるだけの練習モードも新たに用意された。

ソロプレイの場合は「csmash」で起動する。ネットワーク対戦の場合は、片方が-sオプションを付けてサーバとなり、もう一方が-cオプションを付けてクライアントとして起動すればいい。まずは、自分のプレイタイプを「Pen Attack(前陣速攻型)」、「Pen Drive(ドライブ主戦型)」、「Shake Cut(カット主戦型)」の中から選択する(画面1)。ソロプレイの場合、対戦相手となるコンピュータのプレイタイプはランダムに決定される。

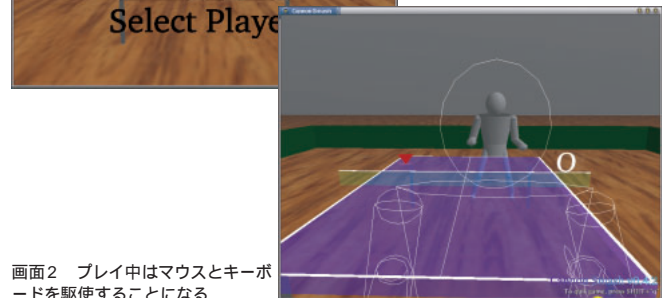
プレイヤーは、球を打ち込む位置をキーボードのフルキー部分で指示しつつ、プレイヤーの移動とショットをマウスで行う。マウスのボタンをクリックすると、サービスした球がキーボードで指定した位置のインジケータ()に向かって飛ぶ。相手プレイヤーがレシーブすると、返ってくる球の軌道が点線で表示される。最適な打点が赤く表示される

ので、赤い点のそばに球が来るのを待ってボタンをクリックしよう。タイミングが合えば強い球を打てるし、そうでなければ返球が浮いたり空振りしてしまう。相手が返す球の軌道によっては、マウスを動かしてプレイヤー、を移動させてからでないと、最適打点を示す赤い点が表示されない場合もある。

ルールは実際の卓球に基づいており、5球ごとにサーブをチェンジし、先に21ポイントを取ったほうが勝ちだ(1セットマッチ)。5ポイント先取や11ポイント先取に変更したり、コンピュータの強さやサウンドドライバを切り替えることもできる。



画面1 3種類のプレイタイプから自分の好みのものを選択しよう



画面2 プレイ中はマウスとキーボードを駆使することになる

リアルな挙動の3Dビリヤードゲーム

Gtulpas

バージョン: 1.0.0

ライセンス: GPL

<http://www.suse.cz/gtulpas/>

X

Mesa

GTK+

GNOME

9ボール、スヌーカー、キャロムをプレイできる3Dビリヤードゲーム。3Dアクセラレタカードは必要ない。優秀な物理エンジンを搭載しており、ボールに捻りや押し引きを加えた際の挙動はなかなかリアルだ。ウィンドウを分割して複数の視点からテーブルを眺めたり、自分でゲームのルールを追加してプレイすることもできる。動作にはGNOMEやMesaのほか、GtkGLArea、guileが必要だ。

ウィンドウには、中央のメインビューのほか、左側に3つのミニビューが用意され、さまざまな角度からテーブルを眺められる(画面1)。ミニビューを消すことも可能だ。用意されているゲームは、ポケット(穴)のないテーブルで行う三つ球のキャロム、ポケット台で行うおなじみの9ボール、少し大きめのポケット台と小さな球で行うスヌーカーの3種類で、[File] - [Select game]で切り替える。現時点では、人間同士の対戦のみ可能だ。

ゲーム中の操作はマウスとキーで行う。基本的な操作は、「英字キーを押しながらマウスを左ドラッグする」というもの。たとえば、Cキーを押しながらマウスを上下にドラッグすると、キューのストローク(手球を撞く動作)になる。プレイヤーの視線はマウスの左/中ドラッグで

変更できるが、これだけでは思い通りのアングルにすることが難しいため、0~9キーで代表的な視線に切り替えられる。お勧めは、キューの向きに視線を連動させる0キーだ。左ドラッグだけでキューの向きを変えられ、手球が常にビューの中心に位置するので狙いをつけやすい。Ctrlキーによる微調整も可能だ。

右上に表示された手球の撞点を変更すると、手球を撞く際に捻りや押し引きのアクションを加えられる。キューの向きと撞点が決まったら、Cキーを押しながらドラッグして手球を撞こう。直前のショットを取り消したり、リプレイすることも可能だ。



画面1 初期設定ではウィンドウに4つのビューが表示される

画面2 おなじみ9ボール。ファール後は手球がフリーボールになる

「たま」をとっかえて連鎖を狙え!

XTokkaeTama

バージョン: 1.0a

ライセンス: フリー

<http://www.hry.info.gifu-u.ac.jp/~hideki/index.html>

X

黄、黄緑、マゼンタ、紫、青の5色の「たま」をとっかえて消していくパズルゲーム。「一人ゲーム」と「二人ゲーム」のモード選択が可能で、「一人ゲーム」の場合はコンピュータとの対戦になる。どちらの場合も、たまを送って相手のフィールドをたまで埋め尽くせば勝ち。「二人ゲーム」の場合は、勝ち負けが決まった時点でゲームオーバーとなり初期画面に戻る。「一人ゲーム」の場合は、次のレベルに進むことができる。

このゲームには、丸いたまと四角いたまの2種類のたまが登場し、丸いたまは縦または横に3個くっつけることで消すことができる。丸いたまが消えるときに、上下左右に四角いたまがある場合は、そのたまを丸くすることができ、これを繰り返してフィールドに積み重なったたまを消していくのが基本ルールだ。

また、フィールドに積み重なっているたまの位置をうまく取り替えて丸いたまを一回の操作で連鎖させて消すことで、相手に「おくりたま」を送って攻撃することができる。おくりたまの発生条件は2つ以上の連鎖が発生したときのみで、単純にたまを消すだけでは相手にたまは送られない。

とにかくこのゲームの攻略のカギは連鎖にあり、適切なタイミングで相手におくりたまを送りつけるのが秘訣だ。フィールドにある程度の数のたまがたままってないと10連鎖以上の大連鎖は期待できない。フィールドにたまをためるのは、相手におくりたまをもらうか、bキーを使って自分でたまをせり上げるかのどちらかだが、一歩間違えると、自分のフィールドを一杯にしてしまい、そのままゲームオーバーになってしまう危険性も高いので注意が必要だ。特にレベルが上がってくると敵の攻撃も速くなるので、bキーを使っているところではないかも:-)。

連鎖の方式は、基本的なものとして、「縦連鎖」、「横連鎖」、「くの字連鎖」の3種類を身につけておきたい。これらの連鎖は組み合わせで発生させたり、いろいろ応用が可能なので、あれこれやってみよう。経験的には16~17連鎖が限界だと思うが、ぜひこのあたりに挑戦してほしい。

画面1 7連鎖発生中! 攻略の鍵は連鎖にあり





空間把握能力が問われる3D落ちものパズル

GNOME 3D Tetris (gno3dtet)

バージョン：1.6.0

ライセンス：GPL

<http://webdat.com/seb/3dtetris.html>

X GTK+ GNOME

さまざまな立体ブロックを操作して平面上に敷き詰める3D落ちものパズルゲーム。回転軸が3つもあるため、複雑な形のブロックを思い通りに回転させるには優れた空間把握能力が要求される。ブロックはワイヤフレームで表示されるため、3DアクセラレータカードやMesaライブラリは必要ない。

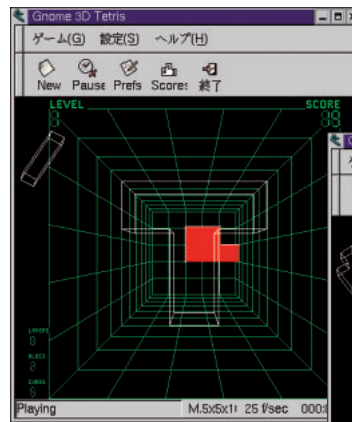
コマンドラインから起動するか、GNOMEメニューの[ゲーム]-[3D Tetris]を選択すると、盤面を含むウィンドウが開く。ツールバーの[New]ボタンでプレイ開始だ。ワイヤフレームの立体ブロックが登場し、ゆっくりと画面奥に向かって下降していく(画面1)。

操作にはキーボードを使用する。ブロックの移動はカーソルキー、各軸に対する回転はそれぞれq/aキー、w/sキー、e/dキーで行う。思い通りの位置と向きになったら、スペースキーを押してブロックを一気に底まで落下させよう。

底に積まれたブロックは、高さに応じて色分け表示される。右側に色のサンプルが表示されるので参考にしよう(画面2)。ブロックにより隙間なく平面を埋めると、その平面のブロックが消え、それより上のブロックが1段下がる。積まれたブロックが盤面の上まで達するとゲーム

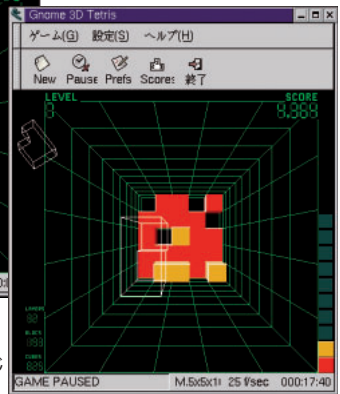
オーバーだ。ハイスコアを出せば名前を登録でき、ツールバーの[Score]ボタンで参照できる。

難易度の初期値は「Medium」に設定されている。もし、平面をひとつも揃えられないままゲームオーバーになるようなら難易度を変更しよう。ツールバーの[Prefs]ボタンを押して設定ダイアログを開き、[Game difficulty]の設定を[Easy]に変更すればいい。もちろん、[Hard]に変更して、さらにシビアなゲームをプレイすることも可能だ。設定ダイアログでは、このほかに盤面のサイズや操作に使用するキー、サウンドなどをカスタマイズできる。



画面1 立体のブロックを思い通りに回転させるのは難しい

画面2 積まれたブロックは、高さに応じて色分け表示される



同じ色の宝石を並べろ！

GemDropX

バージョン：0.7

ライセンス：GPL

<http://www.newbreedsoftware.com/gemdropx/>

X

同じ色の宝石3つ以上を並べて消す1人用の落ちものパズルゲーム。起動すると初期画面が表示され、スペースキーまたはEnterキーの入力によってゲームが開始される。ゲーム開始時には画面上部に4種類の宝石(gem)が並んでいる。一定時間が経過すると宝石の数が増えて徐々に宝石が下に降りてくる。宝石が一番下まで降りてしまうとゲームオーバーとなるので、フィールドが宝石で満たされないように、キャラクターを動かしてフィールドから宝石を吸い取り、宝石を投げ返して宝石を並べて消去させよう。

宝石が降りてくる間隔は最初はゆっくりで、ゲームが進むにつれてだんだん短くなる。底部の警告灯が宝石が降りるときの合図なので、警告灯の点灯をチェックするようにしよう。

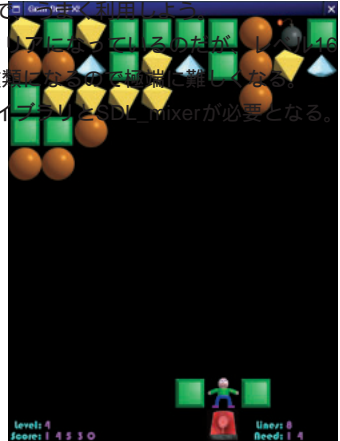
宝石を投げ返したとき、同じ種類の宝石が縦に3つ以上揃うと宝石を爆発させることができる。そのとき同じ種類の宝石が左右にあれば、連鎖により左右の宝石も爆発させることができる。一定数の宝石を消すとゲームがレベルアップして速度が速くなり、宝石の数も多くなるので、クリアするのが大変になる。

ゲーム中のキー操作は次のとおり。

- k/ : キャラクタを左に移動
- l/ : キャラクタを右に移動
- z/ : 宝石を取る
- a/ : 宝石を投げ返す
- Enter : 宝石を下降させる
- q / Esc : ゲーム終了
- p / スペース : ゲーム中断、任意のキーで再開

また途中で、隣接する宝石を爆発させる「ボム」、時間の進みを一時停止できる「ストップウォッチ」、宝石の種類に関係なく隣接するすべての宝石と連鎖爆発させられる「ワイルドカード」の3種類のアイテムが出現する場合がある。これらのアイテムは、連鎖に含めて爆発させることで、いろいろな効果があるので、

このゲームはレベル20で全面クリアを目指すのだが、レベル14以降は宝石の種類が4種類から8種類になるので極端に難しくなる。なお、コンパイルには、SDLライブラリとSDL_mixerが必要となる。



画面1 連鎖とアイテムをうまく使ってクリアを目指そう

数字合わせ落ちものパズルゲーム

X777

バージョン:0.1

ライセンス:フリー

<http://jg.linux.or.jp/>

X

落下するキューブに書かれた数字を合わせて消していくパズルゲーム。L字型につながった3個のキューブには1~7の数字が書かれており、順番に上から落ちてくる。そのキューブを積んでいき、上下または左右に隣接したブロックの合計を7にすることによって、キューブを画面から消すことができる。すなわち、「1」と「6」、「2」と「5」、「3」と「4」と書かれたキューブを隣接させることが消す条件なのだ。注意しておきたいのは、「7」と書かれた緑色のキューブ。このキューブは特別で、縦か横に最低3個つなげるのが消すための条件である。

[Start] ボタンでゲームを開始。次のキー操作でキューブを回転させたり、移動させたりして積んでいこう。

/h : キューブを左に移動させる

/l : キューブを右に移動させる

/j : キューブを反時計回りに回転させる

/k : キューブを時計回りに回転させる

スペース : キューブを下まで落とす

「ぶよぶよ」タイプの落ちものパズルゲーム

XPuzzleTama

バージョン : 1.5b

ライセンス : フリー

<http://www.hry.info.gifu-u.ac.jp/~hideki/index.html>

X

先に紹介した「XTokaeTama」と同じ作者の落ちものパズルゲーム。類似したシステムだが、こちらはいわゆる「ぶよぶよ」系の落ちものパズルゲーム。ただ、仕掛けは違っており、こちらのほうが難易度は高くなっている。

基本的なルールは5色の丸い「たま」や四角い「たま」が2つつ上から落ちて来るので、丸いたまを3つつつけることでそのたまを消すことができる。四角いたまは凍っておりそのままでは消せない。まわりの丸いたまが消える時の熱で解凍して丸いたまにする必要があるのだ。画面右にある [Timer] の表示が0になる前に [Task] の個数だけたまたまを消せないと、冷凍されたたまがせり上がってくるので注意が必要だ。ただし、そのせり上がってきたたまを有効に利用して大連鎖を引き起こすこともできる。たまたまでフィールドを埋め尽くされてしまうと、その時点でゲームオーバーとなる。

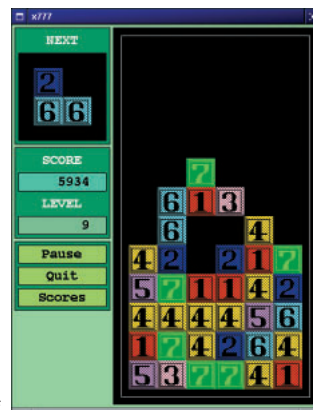
遊び方は、1人ゲームまたは2人ゲームを選択できる。1人ゲームは“xpuzzletama”と入力して起動する。2人ゲームの場合は、先に起動する人が“xpuzzletama -net”で起動し、次に起動する人が、“xpuzzletama -host ホスト名”で起動すれば準備OKである。

落ちものパズルゲームなので連鎖を狙いたいところだが、このゲームで連鎖を狙うのはかなり難しい。したがって、落ちてくるキューブを地道に1組ずつ消していくのが必勝法だ。また、キューブは書かれている数字によって色が違うので、完璧に消せる色の組み合わせを覚えて、脊髄反射で対応できるようになるまで徹底的に鍛えよう。

また、「7」のキューブは3個つなげる必要があるので、すぐにたまってしまふ可能性が高い。できる限り2個ずつつなげておいて、次の「7」が出てきたらすぐに消せるよう準備しておこう。

公式なホームページは存在しないが、筆者が幹事をしているJG (Japanese Games and amusement) のホームページからソースコードまたはVine Linux 2.0に対応したrpm形式のバイナリパッケージが入手できるようになっている。基本的なライブラリしか必要としないので、Vine Linux 2.0以外のディストリビューションでも簡単にコンパイルできるはず。まだやったことがない人はぜひ挑戦してほしい。

シンプルではあるが、計算を使っているので一瞬の判断に戸惑うこともあり、中毒性も高い。JGに収録したゲームの中でも非常に人気の高いゲームのひとつである。



画面1 地道にコツコツ消していくのがコツだ

ゲームの進行も1人ゲームと2人ゲームの場合で異なる。

1人ゲームの場合

通常のゲームで遊ぶ場合、起動画面で [スコアアタック] を選択してスペースキーを押す。ゲーム中は前述した、「Timer」表示が無くなる前に「Task」分のたまを消すことが当面の目標だ。これに失敗すると冷凍したたまがせり上がってくる。

2人ゲームの場合

こちらは相手と対戦することが目的なので、連鎖を起こして、相手にたまを送って攻撃する。

また、特別に白いたまと黒いたまがあり、白はまわりのたまを解凍し、逆に黒はまわりのたまを凍らせるという効果がある。特に、白いたまが2個くっついて出てきたら連鎖のチャンス。冷凍たまの群れに対しては横2つつ分のスペースを空けておこう。



画面1 連鎖の数だけ相手にたまを送られる



隠されたクリスタルの場所を見つける

blackbox

バージョン : 1.1

ライセンス : GPL

<http://members.linuxstart.com/karlb/blackbox.html>

X

画面に隠されたクリスタルの場所を推定するパズルゲーム。Penguin CommandやBumpRaceなどでも有名なKarl Bartel氏の作品でコンパイル、動作にはSDLライブラリが必要だ。

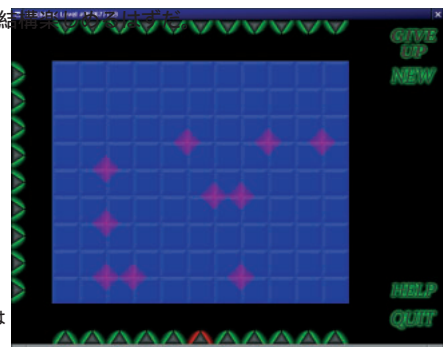
ゲームを開始すると画面中央に大きなフィールドとその上下左に赤色または緑色のレーザー台が表示される。画面中央のフィールドは、縦8×横10の合計80の格子点があり、この80の格子点のうちのいくつかにレーザー光線を反射する機能を持ったクリスタルが隠されている。レーザー台は上下にそれぞれ11台ずつ、左側に9台の合計31台設置されている。

このゲームの目的は、ユーザーがレーザー光線を発射して、その軌跡をもとに、クリスタルの隠された場所を推定して当てることである。まずは、レーザー台をマウスの左ボタンでクリックして、そのレーザー台から実際にレーザー光線が発射させてみよう。レーザー光線は画面中央のフィールドに隠されたクリスタルに当たると反射して方向を90度変えるので、指定したレーザー台から発射したレーザー台がまっすぐに画面中央のフィールドを通り抜けなかった場合は、少なくとも一つのクリスタルに当たったと考えられる。このようにして、レーザー光線の軌跡

から、クリスタルが隠された場所を推定していくのである。ここだと考えた場所をマウスの左ボタンでクリックすると、ユーザーの推定結果として画面上にマークが置かれる。ユーザーが置いたマークの位置とレーザー光線の軌跡との整合性がとれた場合に、レーザー台の色が赤から緑に変わる。緑色のレーザー台は、ユーザーが置いたマークによって正常にレーザー光線が届いているということなので、すべてのクリスタルの上にマークを置ければ、すべてのレーザー台の色が緑色になる。

やってみて思ったのは、画面の色使いや効果音などの雰囲気がいまいちサイバーしていて非常にかっこいいということ。ゲームのシステム自体はメインスィーパーのような感じで隠されたアイテムを探すというもののだが、実際にやってみると、結構複雑でどれか1つのレーザー台を緑に変えようとしてマークを置くと、今まで緑色だったレーザー台が赤色に戻ってしまったりする場合も多々ある。

ウィンドウサイズが結構大きいので、解像度800×600 / 16bpp以上のX環境で、できればサウンドカードを搭載したマシンでやってみてほしい。雰囲気だけでも結構



画面1 単純そうだが推理はかなり難解だ

KDEベースの一筆書きパズルゲーム

KGryzzles

バージョン : 0.1.6

ライセンス : フリーソフト

<http://www.hess-brothers.de/linux/kgryzzles/>

X

Qt

一筆書きの原理を応用したシンプルなパズルゲーム。ドイツのChristoph Hess氏の作品で、コンパイルと動作にはKDEとQtが必要。オリジナルゲームは作者の兄弟であるEkkehard Hess氏が数年前に発明したものだそう。

起動すると画面上に縦10×横16の大きさのフィールドが出現し、そのフィールド内に複数の青色のボールと1つの赤色のボールが表示される。赤色のボールはプレイヤーのボールで、「」」「」」「」」「」キーを使って操作する。しかし、プレイヤーのボールは通常の青色のボールの上にはしか移動できず、さらに、プレイヤーのボールが通常のボールの上を通るとそのボールは画面上から消滅してしまう。したがって、赤色のボールを動かすことによって青色のボールは次々にフィールド上から消えてしまうことになる。

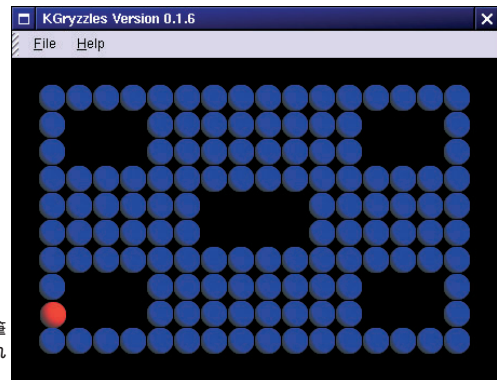
このゲームの目的は、一筆書きの要領でプレイヤーの赤色のボールをうまく動かして、通常の青色のボールをフィールド上から全部消してしまうことである。慎重に赤色のボールを動かさないと、すぐに手詰まりになってしまうので注意が必要だ。昔よく遊んだ一筆書きの応用なので、子供の頃の経験を思い出しつつ挑戦しよう。

フィールドの青い玉をすべて消すと面クリアになり次の面に進むことができる。このゲームは全部で50面あるらしいが、パズルゲーム好きのわりに、要領の悪い筆者はまだ最後までたどり着けていない。シンプルながらハマりそうなので、少しずつ解いていくのが面白いゲームなカモ。

解き方のコツとしては、制限時間がないので、とにかく慌てて動かさないことが肝要。特に、ボールが一列に並んでいるところを消す場合は、後戻りできないので注意が必要だ。

各面の開始点は最初から決まっている。全パターンを追いかけたわけではないが、終了点も決まっている場合が多いので、慣れてくればかなりパターン化できてるのではないだろうか。

とにかく筆者は今、50面クリアして「KGryzzles Master」になるべく熱い気持ちで挑戦している。読者のみなさんも一度挑戦してみたいかがだろうか？



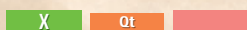
画面1 ようは一筆書きなのだが、これが結構大変なのだ

36種類72個の石を配置するパズルゲーム

Kishido

バージョン : 1.0.2

ライセンス : GPL

<http://www.Informatik.Uni-Oldenburg.DE/km/kishido/>

「石道」と呼ばれる日本発のパズルゲームのクローン（オリジナルの石道はPC-8801シリーズ用で、1990年4月にアスキーから発売された）。縦8×横12のマスの目に区切られたボード上に、一定のルールに基づいて72個の石をすべて配置することがゲームの目的だ。上下左右を石で囲まれたマスほど石を置くための条件はきつくなるが、それだけ高得点のチャンスになる。

コマンドラインから起動するか、KDEメニューの[ゲーム]-[kishido]を選択するとゲームが始まる（画面1）。石のマークと色はそれぞれ6種類ずつあり、同一の石が2つずつ（計72個）用意されている。ゲーム開始時には、ボードの四隅に1つずつ、中央に2つの石がすでに置かれている。右上に次の石が表示されているので、その置き場となるボード上の空きマスをクリックしよう。石を置けるのは、少なくとも1つの石が周囲（上下左右の4マス）に置かれているマスだ。また、新たに置く石のマークと色は、以下の条件を満たす必要がある。

周囲の石が1つ：マーク、色のいずれかが周囲の石と一致

周囲の石が2つ：マークが周囲の石の一方、色がもう一方と一致

周囲の石が3つ：マークが周囲の石の2つと、色が残りの1つと一致（逆も可）

周囲の石が4つ：マークが周囲の石の2つと、色が残りの2つと一致

周囲の石が多いほど条件が厳しいが、それだけ得点も高い。周囲の石が1つだと1点だが、2つなら5点、3つなら20点、4つだと100点も入るのだ。なお、マウスの右ボタンをクリックすると、石を置ける場所が赤く反転する（画面2）。ルールを覚えるまではこの機能を活用するとよいだろう（使いすぎるとハイスコアが記録されない）。すべての石をボードに置か、石を置ける場所がなくなるとゲームオーバーだ。



画面1 四隅と中央に石が置かれた状態でゲームが始まる

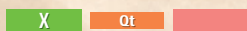
画面2 右ボタンクリックで石の置ける場所を確認できる

思考力が試される論理型パズルゲーム

KPooka

バージョン : 0.3

ライセンス : GPL

<http://www.ucs.co.za/pgr/kpooka/>

不定形生物の「プロブ」を操作して、他のプロブを押しつけながら出口を目指すパズルゲーム。ペンゴや倉庫番といった名作パズルの流れをくむ論理型パズルで、アクション要素がない分、盤面の先を読む力が要求される。パズルは全部で15レベル用意されており、後半には動かせないブロックや別の場所に転送されるテレポーターが登場して難易度が高くなる。

コマンドラインから起動するか、KDEメニューの[ゲーム]-[KPooka]を選択すると、最初のレベル（面）が表示される（画面1）。なお、次回からは、各ユーザーがまだ解いていないレベルが自動的に表示されるようになる。すでに解いたレベルは自由に往来可能だ。

盤面に並ぶ四角いブロックは、上下左右がパイプで接続されている。目玉のある赤いプロブをカーソルキーで操作し、「E」と表示された出口のブロックまで導くことがゲームの目的だ。いくつかのブロックには赤いプロブの邪魔をする緑のプロブが配置されていて、これを押しつけながら移動する必要がある。ただし、倉庫番やペンゴと同様の制限として、「対象となるプロブの後ろが空きブロック」の場合しか押すことはできない。押されたプロブは、ほかのプロブまたは盤面の端にぶつか

るまで自動的に移動する。

はじめの数レベルは、基本的なテクニックを習得するために用意されているのでそれほど難しくはないだろう。このゲームのコツは、クリアするために必要な道順をゴールから逆向きに考えることだ。レベルが進むにつれ、押すことのできない灰色のブロックや、別の場所に瞬間移動するテレポーターといったギミックも登場する。手詰まりになったら、Escキーを押してそのレベルの初期状態に戻そう。ライフなどの回数制限は設けられていないので、何度でもやり直すことができる。

画面1 赤いプロブで緑のプロブを押しながら出口を目指す



画面2 灰色のブロックは動かせないので注意しよう



X版「さめがめ」

SameGame for X Window

バージョン: 3.02 種別: フリー

<http://www.asahi-net.or.jp/fj9y-ynym/works.html>

X

DOS版、Windows版、Macintosh版でもすっかり有名になった定番パズルゲーム「さめがめ」のX Window System版。もちろんオリジナルは福本栄治氏が開発したUNIX上のtty上で動作する版で、シンプルながら未だにユーザーは多い。また、Emacs上で動作するe-lisp版も作成されており、こちらも段位認定機能などのオリジナル機能が追加されて、根強い人気を誇っている。最近ではGNOMEやKDEなどのデスクトップ環境にも移植されつつあるようで、このゲームをやったことのない人はほとんどいないのではないだろうか。

ルールは単純で、同じ色の玉どうしは上下左右に2個以上つなげることで画面上から消すことができる。玉が空白になった部分には上から玉が詰められる。また、縦一列の玉がなくなるとその右の列が左に詰めてくる。

得点は「一度に消した玉の個数-2」の2乗が加算される。具体的には別掲の表のようになる。たとえば、一度に20個消すと18の2乗で324が得点として加算される。

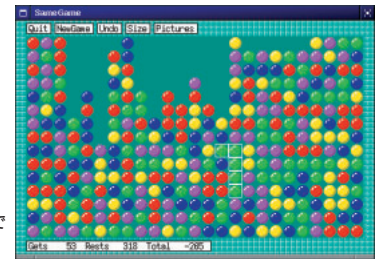
玉を取ることができなくなった時点でゲームオーバーとなり、玉が残った場合は玉の数だけ得点が減点され、残り玉が0の場合は、パーフェ

クト賞としてボーナス点（通常は1000点）が加算される。

このゲームは時間を競うものではなく、高得点またはパーフェクトを目指すのが中心になる。どちらの場合も、いかに一度に同じ色の玉を同時に消せるかがポイントになるので、とにかく1つの色に着目してみよう。特に横方向のつながりが重要で、縦一列にその玉が存在しない列があればその列で分断されてしまう。

筆者は長年このゲームで遊んでいるが、筆者の記憶の中では一度もパーフェクトを出したことがなく、このゲームでパーフェクトを出すことが夢である。なお、X版ではWindows用のBMP形式の駒データを使用することもできるので、自分の好きな画像を駒にして遊ぶことも可能だ。

一度に消した玉の個数(個)	得点(点)
2	0
3	1
5	9
10	64
15	169
20	324



画面1 キミはパーフェクトを達成できるか？

GNOME上で動作する「倉庫番」

GSokoban

バージョン: 1.0 ライセンス: GPL

<http://home.swipnet.se/darshiva/g Sokoban/>

X

GTK+

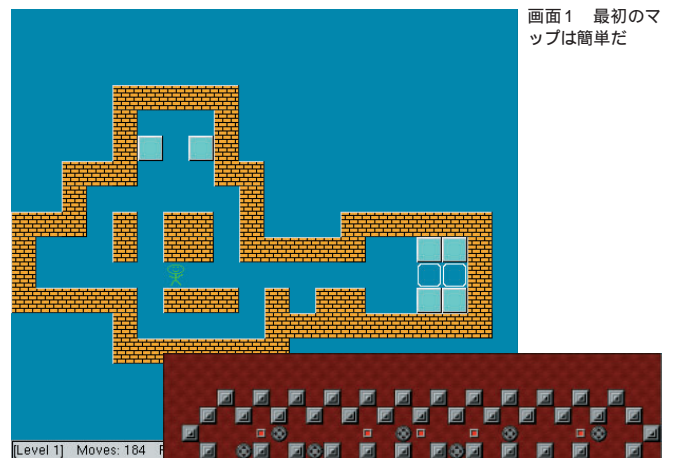
「倉庫番」とほぼ同じシステムのパズルゲーム。倉庫番は20年近く前からあるパズルゲームで、ゲーム自体をやったことがなくても、名前は知っているという人も多いだろう。倉庫の中にあるキャラクタを操作して、荷物を押して所定の位置に持っていくという単純なゲームだが、単純なだけに短時間でも簡単に遊べ、ついついハマってしまう。

ルールは「荷物は押すことしかできない」、「一度に押すことのできる荷物は1つだけ」の2つのみ。単純だが、パズルはルールが単純だからこそ、奥が深く難しいものができる。難しい面では、よく考えて荷物を動かさないと、すぐに手詰まり状態になってしまう。

ゲームに使うのは、マイキャラを操作するためのカーソルキーだけ。数多くのマップが用意されているし、一度解いたマップでも、もう一度少ない手順で終わらせることができるように挑戦したくなる。Undo機能もついているので、「しまった」と思ったときには、やり直しも効くが、Undoはやっぱり悔しい。

キャラクタも選べるようになっている。また、キャラクタのグラフィックの形式は通常のxpm形式なので、その気になれば、自分でキャラクタを作ることも可能だ。

tarボールでソースが配布されているが、通常の手順でビルドすると、実行時にエラーが出る。「./configure」時に「--prefix=/usr」オプションをつけてビルドするとよいようだ。GNOME用としては「G」 sokobanという名前になってはいるが、GTK+がインストールされていれば、KDEでも動作する。



画面1 最初のマップは簡単だ



画面2 最終マップ。こんなの解けるの??

華麗な画面とエフェクトの新世代ワームゲーム

Heroes

バージョン：0.4

ライセンス：GPL

<http://www.realtech.scene.org/heroes/>
<http://www.ggi-project.org/> (GGI/GII)

X

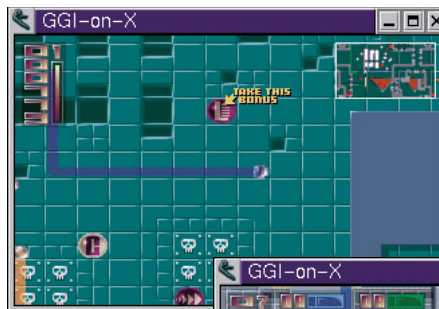
次第に身体が伸びるワームを操作するアクションゲーム。古典的なワームゲームと異なり、数々のアイテムや凝ったステージ構成、派手な画面エフェクトや効果音でユーザーを飽きさせない。最大94レベルをプレイ可能な「QUEST」モードをはじめ、プレイモードが豊富に用意されている。画面を分割表示して人間同士で対戦することも可能だ。動作にはGGI/GIIライブラリが必要となる。

起動すると、凝った画面エフェクトのタイトルが表示される。なお、画面表示サイズが320×200ドットと小さいので、/etc/X11/XF86Configの画面モードに「320x200」を追加しておき、起動後にCtrl - Alt - +/-キーで画面モードを切り替えるとよいだろう。メインメニューで[PLAY]を選択し、[QUEST]を選択するとプレイ開始だ。

操作にはカーソルキーを使用する。他のワームや通路に挟まれないようにワームを操作しよう(画面1)。さらに、左上のゲージにパワーが残っていれば、右Shiftキーで一時停止、右Ctrlキーで加速できる。ステージ上には、加速レーンや地雷原などのギミックのほか、ワームの長さを伸縮する「L」「L」、スピードを増減する「S」「S」など各種のアイテムが用意されている。紫は自分、白は他のワームに作用するア

アイテムだ。いずれかのワームの長さが10以上になると、次のレベルに移動するための特別なアイテムが現れる。

ゲームモードを選択する前に、一番上の[ONE PLAYER]を[TWO PLAYERS]に変更しておく、プレイ画面が左右に分割されて対戦プレイが可能になる。左側のプレイヤーはe/s/d/xキーでワームを操作する。最後まで生き残ったものが勝者となる「DEATH MATCH」、制限時間内に自分と同色の宝石を数多く集めたものが勝者となる「COLOR」など、5種類のゲームモードが用意されている。



画面1 カーソルキーでワームを操作してアイテムを取ろう



画面2 2人同時プレイでは、画面が左右に分割表示される

X対応の本格派連珠ゲーム

xrenju

バージョン：0.4

ライセンス：フリー

<http://www.techfirm.co.jp/masaaki/xshisen.html>

X

日本語

X上で動作する本格的な連珠ゲーム。筆者は五目並べと連珠の違いがよくわかってないが、作者は本格的な連珠ゲームを目指しているとのこと。いい機会なのでここで連珠のルールについてふれておこう。

ゲームは2人の競技者(対局者)によって行う。

先手が黒石を、後手が白石を持ち、盤の中央から一手ずつ交互に打つ。タテまたはヨコまたはナナメのいずれかに、先に五つ連続した形(五連)を並べたほうが勝ち。

黒には禁手(きんて)を設け、五連ができる前に禁手を打ったときは禁手負けになる。禁手には「三々」、「四々」、「長連」がある。

白には禁手はない。白の長連は五連と同様に勝ち。白の三々や四々は打ってさしつかえなし。

どちらにも五連ができないときは引き分けになる。

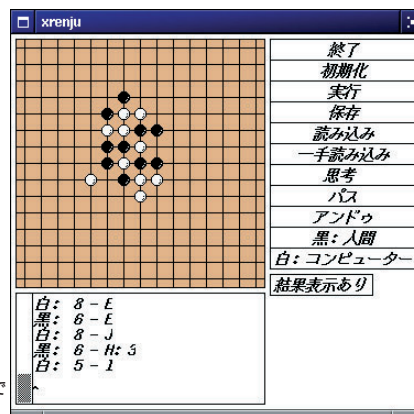
平たく言うと、自分の石を先に連続して5つ並べたほうの勝ちとなる。このゲームは基本的にコンピュータと対戦する仕様になっており、コンピュータ側の思考ルーチンもかなり本格的。初心者だと簡単に負けて

しまうだろう。

コンピュータに勝つためには、コンピュータの体制が整う前に一気に攻撃する先手必勝作戦か、早め早めにひたすら防御に徹する作戦が良いのではないだろうか。

また、最新バージョンの0.4では、Web上のCGIプログラムで連珠を楽しむためのwwwrenjuというパッケージも付属する。作者のホームページでは、このプログラムを使用したWeb上で動作する連珠プログラムがCGIプログラムとして実際に動作していて、Webブラウザから遊ぶことができるようになっている。

作者自身はあまり強くない、次の課題は思考ルーチンと言っているが、なかなかどうして、そこそこ遊べるプログラムに仕上がっているので、今後のバージョンアップも楽しみである。思考ゲームが好きな方にはお勧めだ。



画面1 本格的な連珠をXで楽しもう



麻雀牌を使ったパズルゲーム

四川省 for X11

バージョン: 1.36 ライセンス: フリー

<http://www.techfirm.co.jp/masaoki/xshisen.html>

X 日本語可

X上で動作する麻雀牌を使ったパズルゲーム。麻雀牌を使うので「上海」と間違われやすいが、盤面上から牌を2個ずつ取り除くという手段は同じでも、牌の並べ方や取り除くルールなどが全く違っている。

起動して、メニューバーの[ゲーム]-[スタート]を選択するとゲーム開始となり、144枚の麻雀牌が縦8枚×横18枚の形でランダムに並べられる。この状態で、マウスの左ボタンで牌を直接選択して、盤面上から2個ずつ取り除いていき、最終的にすべての牌を取り除くことがゲームの目的だ。牌を取り除くことができるのは、以下の条件を両方満たした場合のみである。

- 2つの牌が同じ種類のものであること。
- 2つの牌が牌のないところを2曲がり以内で結べること。つまり、2つの牌の中心から中心までを牌のないところを通って水平または垂直の連続した線分3本以下で結べること。

文章で書くと一瞬難しそうなルールのように思えるが、実際にやってみると簡単なので、まずは挑戦してみよう。“-demo”オプション付き

で起動すると「自動デモ」モードで起動するので、初心者はずこのモードでルールを覚えよう。

手詰まりになると自動的にダイアログがポップアップして終了となるのだが、たまに取り除くことができる牌がなかなか見つからないことがある。このとき便利なのがサーチ機能で、メニューバーの[補助操作]-[サーチ]で呼び出すことができる。ただし、この機能を一度でも使うとハイスコアが記録されないので注意が必要だ。1つ前の操作を取り消すことのできるアンドゥ機能も持っており、[補助操作]-[一手戻り]で呼び出すことができる。こちらもサーチ機能と同様、一度でも使うとハイスコアは記録されなくなる。

「さめがめ」のように空いた領域に牌が落ちてきて詰まってしまう「重力」モード（通常は牌を取り除くとその領域は空き領域となる）や、牌の数が通常の4倍で遊べる「超特大」モードも面白いのでお勧め。

コンパイルには、MotifまたはAthena WidgetとXpmライブラリが必要。Athena版のバイナリパッケージがJG-1.0に収録されている。



画面1 似てはいるが「上海」とは違うゲームだ

200種類以上のソリティアを集めたカードゲーム

PySol

バージョン: 4.41 ライセンス: GPL

<http://pysol.tsx.org/>

X Python Tcl/Tk

バリエーション豊かなカードゲーム。バージョンアップされるたびにゲームの種類が増え、おなじみの「クロンダイク」や「フリーセル」をはじめ、最新版の4.41ではなんと200種類以上にも及ぶトランプや花札の一人遊び（ソリティア）をプレイできる。さらに、プラグイン形式で新たなゲームを追加できるほか、さまざまなデザインのカードを集めたカードセット、専用のBGMのセットやそれを演奏するためのサウンドサーバなども別途配布されている。

起動すると、BGMの演奏とともにクロンダイクのカードが配られてプレイできる状態になる（画面1）。なお、次回からは、終了前にプレイしていたゲームが自動的に開始される。別のゲームに切り替えるには、[File]-[Select game]以下から選択する。最初は、[Popular games]以下のクロンダイク、フリーセル、スパイダーといった代表的なゲームをプレイするとよいだろう。

ゲーム中の操作にはマウスを使用する。左ボタンのクリックでタロン（置札）から新しいカードを引き、ドラッグでタブロー（一時的なカード置き場）に移動するというのが基本的な操作だ。このほか、中ボタンのクリックで積まれている途中の表向きカードを確認でき、ダブル

クリックか右ボタンクリックでファウンデーション（台札）へカードを移動できる（一部の動作は自動化できる）。見事にゲームをクリアすると、Congratulation画面が表示される（画面2）。

ルールがよくわからない場合は、ツールバーの[Rules]ボタンを押してルールの解説を表示しよう。説明は簡潔な英語で書かれているので、読むのはそれほど難しくはないはずだ（基本となるゲームとの相違点だけが書かれている場合もある）。また、操作を遡ってやり直せるアンドゥ、動かせるカードを教えてくれるヒント、コンピュータがプレイするデモなどの機能を使うと素早くルールを把握できるだろう。



画面1 クロンダイクをはじめ、200種類以上のゲームをプレイ可能



画面2 見事にクリアすると、このCongratulation画面が表示される

ソリティアゲームパック

The Ace of Penguins

バージョン: 1.1

ライセンス: GPL

<http://www.delorie.com/store/ace/>

X

X上で1人で遊べるソリティア形式のゲームを集めたゲーム集。作者のDJ Delorie氏の話によれば、収録されているゲームはほとんどWindows上で動作するものと同等であるが、若干内容がブラッシュアップされているとのこと。最新版の1.1では、「freecell」、「golf」、「mastermind」、「merlin」、「minesweeper」、「pegged」、「solitaire」、「taipei」、「thornq」と9種類のソリティア形式のゲームが収録されている。誌面の都合もあるので、2つのゲームだけを紹介しよう。

freecell

ご存じの方も多いと思うが一応紹介しておく、Windowsに収録されている「フリーセル」とほぼ同じで、4つのフリーセルを一時的なカード置き場として使いながら、すべてのカードを種類毎にホームセルに移すことがゲームの目的である。最終的には、4種類のカードを数字の小さい順に1ずつ大きくなるようにすべてホームセルに積み重ねることが必要となる。

taipei

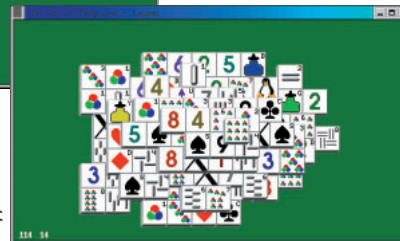
こちら大変メジャーな「上海」タイプのソリティア形式のゲームである。

同じ種類の牌を2個ずつ消していくことができ、その時の条件として「上に他の牌が載っていないこと」、かつ「左または右が他の牌と接していないこと」の2つをクリアしている必要がある。このゲームで使用している牌データはオリジナルっぽくてなかなか面白い。また、「taipedit」というエディタがついてくるのも特徴。

これ1本インストールするだけでも楽しく遊ぶことができるだろう。収録されているゲームは基本的なものばかりなので、空いた時間にも気軽に遊ぶことが可能になっており、Linux初心者への導入用アプリケーションとしてもお勧めだ。



画面1 Windowsでもおなじみ「freecell」



画面2 牌デザインが独特な「taipei」

海外生まれの花札パズルゲーム

Oonsoo

バージョン: 1.2

ライセンス: GPL

<http://www4.ncsu.edu/bwmott/www/onsoo/>

X

たいへん珍しい花札を使ったソリティアタイプのパズルゲーム。トランプのカードを使ったパズルゲームは数多く存在するが、花札を使ったものはあまり出会ったことがない。花札に不慣れなユーザは、まず[Help]をクリックして、札の種類(松、桐、坊主など全部で12種類ある)とその強さの順序(各種類ごとに4枚ずつある)を覚えてほしい。

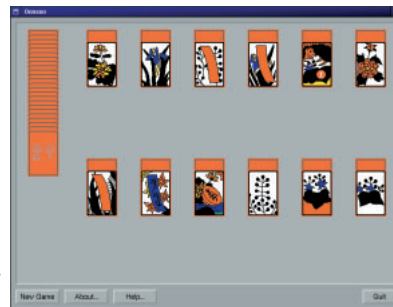
[New Game]をクリックすると、ゲームがスタートする。まず、最初に12枚の札が開かれる。その12枚の中でつなげられる札が見つかった場合は、移動元の札をドラッグして移動先の札にドロップしよう。移動できるかどうかは先ほどの[Help]で表示された強さの順序で判断される。ユーザの選択が正しいと札は移動して、その下の札が開かれる。間違えた選択をした場合は、札は移動せずに元の位置に戻る。

遊んでいて、つなげられる札がなくなった場合は、左に積まれている札の山をクリックしてみよう。すると、今まで開かれていた12枚の札の上に新たに12枚の札が開かれ、再度札の移動が可能となるので、またつなげられる札を移動させていく。このとき、2枚以上の札を同時に移動したい場合は、その札同士がつながっていて、さらに上に別の札が重なっていない場合に限られる。また、左の札の山のオープンが2回目

が最後になってしまうので、つなげ忘れのないように慎重に操作しよう。

このゲームでの操作はマウスの左ボタンしか基本的に使用しない。メニューの決定、札の移動などすべてマウスの左ボタンだけで遊ぶことが可能だ。基本的にソリティアタイプのゲームなので、偶然性がゲーム進行に影響する割合が高い。逆に、次にどんなカードが開かれるのか、札の山をオープンすると次に何が出てくるのか、あのカードはどこに埋まっているのか、がわからないところがゲーム性となっていて面白いところである。

とはいつてもまったく戦略がないわけでもない。たとえば、松や坊主の平札は同じものが2枚存在し、強さもまったく同じであるので、どちらをどちらの上に重ねてもかまわない。たとえば、フィールド上に松の平札が2枚とも開かれているとしよう。この時の重ね方の選択が今後の展開に大きな分かれ道になることが多い。基本セオリーだと、2枚の平札の下の札のつながりやすさを考慮して選択すればよいのだが、たとえそれを心がけていてもクリアが難しいゲームだ。一度や二度であきらめず、クリアできるまで何度も何度も挑戦してみよう。



画面1 作者はきっとオリエンタル好きなのだろう



太陽系の惑星や衛星などを3D表示するシミュレータ

Open Universe

バージョン：1.0 beta3 ライセンス：GPL

<http://openuniverse.sourceforge.net/>

X Mesa

太陽、惑星、衛星、小惑星などの天体や、スペースシャトルなどの人工天体を3D表示する太陽系シミュレータ（以前は「Ssystem」と呼ばれていた）。天体写真を利用した美しいテクスチャが貼られた天体を、さまざまなカメラモードで眺められる。ソフトウェアのみでも実行できるが、3Dアクセラレータがあれば、よりスムーズな表示が可能だ。

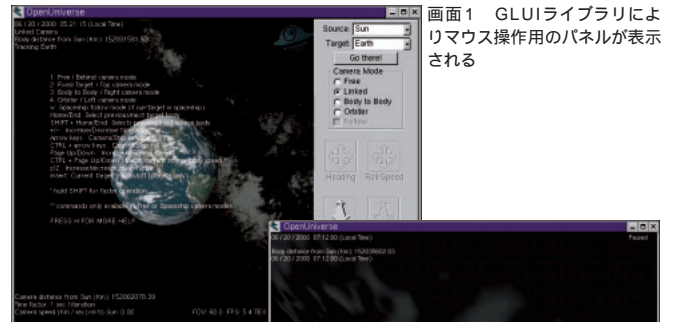
起動すると、数種類あるスプラッシュ画面のひとつを表示しながらデータが読み込まれ、地球をバックに使用キーの説明が表示される。通常は、操作にはキーボードを使用する。各キーに機能が割り当てられており、慣れてくると素早い操作が可能だ。マウスで操作するには、別途配布のGLUIライブラリと組み合わせる必要がある（画面1）。

表示対象（Target）となる天体・宇宙船をHome / Endキーで選択し、4種類のカメラモードを1~4キーで切り替える。天体の位置は実時間に基づいて1秒ごとに更新されるが、+ / - キーで時間スケールを増減可能だ。惑星や恒星の名前を表示したり、近くの天体に対する向きと距離をマップ表示することもできる。気に入った構図が得られたら、pキーで表示を一時停止し、Shift - Sキーでスナップショットを撮ろう。

こうした操作が面倒な人向けに、ユーザーが特に操作をする必要の

ない「デモモード」も用意されている。dキーでデモモードに移行すると、一定時間ごとに表示対象が切り替わり、自動的にカメラが適切な位置に移動する。再度dキーを押すとデモモードが終了する。

スペースシャトル「エンデバー」、アポロ13号、ハッブル望遠鏡（HST）、ステーションVなどの人工天体は、単にアングルを変えて眺めるだけでなく、ユーザーが操縦することも可能だ（画面2）。人工天体を表示した状態で、wキーを押して「カメラ追尾モード」に切り替えればいい。以後、カーソルキーで人工天体の向き、PageUp / Downキーで速度を変えられるようになる。



画面1 GLUIライブラリによりマウス操作用のパネルが表示される



画面2 スペースシャトル「エンデバー」を操縦することも可能だ

アストロアーツが贈る天文シミュレータ

Xplns

バージョン：3.1.0Rel2 ライセンス：フリー

<http://www.astroarts.co.jp/products/Xplns/index-j.html>

X

星空をウィンドウに表示する天文シミュレータ。「ステラナビゲータ for Windows」を開発したアストロアーツ製のソフトだ。Xplnsはステラナビゲータ1.0のサブセットで、6.5等星までの恒星や約1万個の星雲・星団、すべての星座、惑星、太陽、月、四大小惑星、彗星などの天体の運行を正確にシミュレートする。

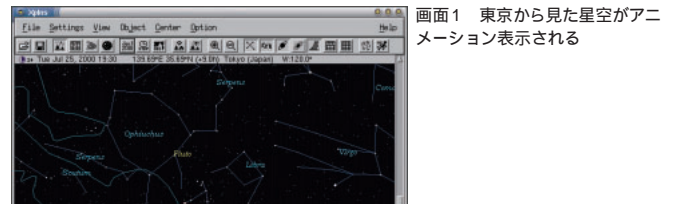
ステラナビゲータの商用コードを含んでいる関係から、バイナリ形式でのみ配布されている。glibc2を採用しているRed Hat系ディストリビューションでは、RPMパッケージを利用すればいい。追加恒星データも一緒にインストールしよう。

表示方法の切り替え、現在地・時刻の設定、表示の拡大・縮小、陽光のオン/オフ、星座や惑星名などの表示といった設定のほとんどはツールバーのボタンで行える。また、天体をクリックすると、天体情報ウィンドウが開いて天体の名前などが表示される。画面をドラッグして、視線を上下左右に移動することも可能だ。

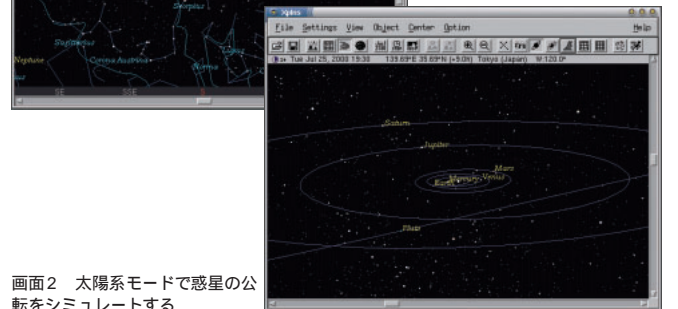
地上から見た星空のアニメーションを表示するには、ツールバーの左から3番目の [Horizontal mode] ボタンを押して地平座標モードに切り替え、右端の [Play] ボタンを押す。初期設定では、東京から見た3

分ごとの星空の変化がアニメーション表示される（画面1）。[Daylight] ボタンをオンにして昼間の星空を眺めたり、[Date and Time] ボタンや [Place] ボタンで日時と場所を変更することも可能だ。

次に、太陽系の惑星の動きを眺めてみよう。[Solar System mode] ボタンを押して太陽系モードに切り替えると、太陽を中心として各惑星の軌道が表示される。拡大・縮小や視線移動を行って適当にアングルを決め、[Time step] ボタンを押して時間間隔を3分から1日に変更してみよう。[Play] ボタンを押すと、太陽系の各惑星が公転しているようすが確認できるはずだ（画面2）。



画面1 東京から見た星空がアニメーション表示される



画面2 太陽系モードで惑星の公転をシミュレートする

ルートウィンドウにリアルな地球を表示する

Xplanet

バージョン : 0.72

ライセンス : GPL

<http://xplanet.sourceforge.net/>

X Mesa

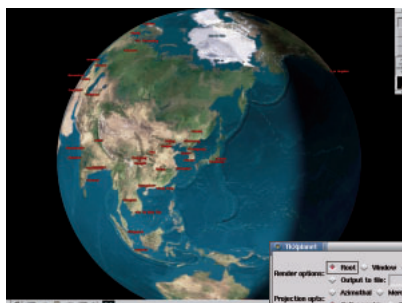
Xのルートウィンドウ(背景)に地球の画像を表示するソフト。この手のソフトとしては「xearth」が有名だが、Xplanetでは各種のマップ画像を利用して、各種図法により投影された地球(またはその他の惑星・衛星など)をリアルに表示する。一定時間ごとにXplanetを呼び出して再描画を行うxplanetbgや、設定をGUIで行うtkxplanetなどの関連ソフトも同梱されている。

マップ画像は、「XGLOBE & XPLANET MAPS」(<http://www.radczyberzine.com/xglobe/>)や「The Planetary Maps Hub」(<http://www.lanacs.ac.uk/postgrad/thomasc1/render/maps.htm>)「Planetary Image Maps」(<http://homepage.mac.com/jhasting/>)といったWebページから別途取得する。あまり大きな画像を扱うと実行時に大量にメモリを消費する。ルートウィンドウと同じ程度の画像を使うと画質がよくなるので、Gimpなどで適当なサイズに縮小してから使うとよいだろう。

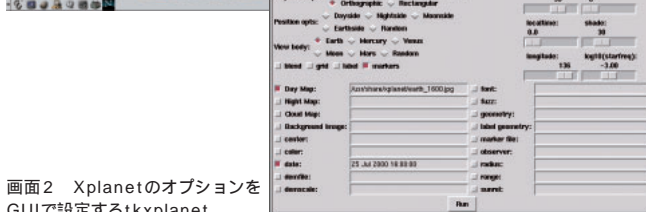
“xplanet -i earth_800.jpg -p or”のように、-iオプションで地球のマップ画像のファイル名を指定すると、ルートウィンドウに丸い地球が表示される(画面1)。-pオプションは表示図法を指定するもので、この例では正射投影図法が使われている。このほか、矩形図法、メルカト

ル図法、モルヴァイデ図法、正距方位図法による表示も可能だ。コマンドラインオプションの指定が面倒なら、tkxplanetでGUIを使って設定することもできる(画面2)。

一定時間ごとに表示を更新するには、xplanetbgコマンドを利用する。オプションはxplanetと同じだ。初期設定では、5分ごとにxplanetが呼び出される(-waitオプションで変更可能)。また、xplanetをアニメーションモード(-aオプション)で起動すると、ルートウィンドウではなく通常のウィンドウに画像が表示され、キー操作などを使って回転速度や方向などを変更できる(Mesaライブラリが必要)。



画面1 ルートウィンドウに地球を表示する



画面2 XplanetのオプションをGUIで設定するtkxplanet

ファイルシステムの宇宙空間を旅しよう

Xcruise

バージョン : 0.24

ライセンス : GPL

<http://tanaka-www.cs.titech.ac.jp/%7Eeuske/prog/index-j.html>

X

ディレクトリを銀河、その中のファイルを惑星といった具合に、ファイルシステムを3次元の宇宙空間に見立てて、ファイルの一覧表示を行う一風変わったソフト。惑星の色や大きさがファイルの属性やサイズに対応しており、自由な宇宙の航海を楽しみながらLinuxのディレクトリ/ファイル構成に対する理解を深められる。なにより、単に宇宙をさすらっているだけでも充分楽しい。

起動すると、ウィンドウが開いて宇宙空間が表示される。最初は、ファイルシステム全体を俯瞰する位置から始まるので、ルートディレクトリを示す大きな楕円と、いくつかのディレクトリの銀河の楕円が表示されるだけだ(画面1)。宇宙空間の移動にはマウスとキーボードを利用する。画面中央部の十字カーソルを移動させることで進行方向を変え、左ボタンを押すと前進、中ボタンを押すと後退する。一時停止はfキー、終了する際はqキーを押せばいい。

銀河にある程度接近するとその銀河に入ったとみなされ、ディレクトリ内のファイルが惑星として表示される。惑星の大きさや色、位置などは一定のルールに基づいて決められている。このため、接近したディレクトリの内容によって、さまざまな銀河の概観を楽しめる(画面2)。

具体的には、各惑星の大きさ(質量)はファイルサイズに対応しており、惑星の色はファイル属性により決定される(読み込む権限を持っていないファイルは紫色など)。また、銀河内での惑星の位置はファイル名の長さや類似度に応じて決められる。

なお、シンボリックリンクはワームホールとして扱われ、リンク元のファイルとは緑の曲線で結ばれる。たとえば、銀河「/bin」に入って惑星「vi」を眺めると、銀河内のワームホール「ex」や「view」などとリンクで結ばれているのを確認できる。もちろん、別の銀河へ遥かに伸びるワームホールも存在する。



画面1 起動時にはルートディレクトリの構成が表示される



画面2 ディレクトリに接近すると、ファイルの惑星が表示される



世界に広がる着せ替えシステムの輪

GnomeKiss

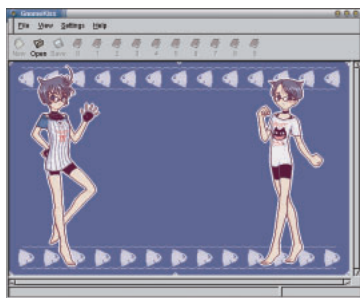
バージョン : 0.4 ライセンス : GPL

<http://www.ecs.soton.ac.uk/~nj198r/code/>

X GTK+ GNOME 日本語可

日本発の着せ替えシステム「KISS」の拡張仕様である「French KISS」に対応したGNOMEアプリケーション。Cherry KISSや拡張パレット、半透明セルなどの機能に対応しており、世界中で作られたさまざまなKISSデータを読み込んで、着せ替えやパズルなどのコンテンツを楽しむことができる。

[ファイル] - [開く] で設定ファイル(拡張子.cnf)を選択すると、対応するKISS画像が表示される。まずは、画像全体が表示されるようにウィンドウサイズを調整しよう。あとは、マウスのドラッグを利用して、自由に服や靴、アクセサリなどのパーツを着せ替えればいい。着せ替えた状態が「ビュー」として設定されているデータの場合は、[ビューの追加]以下の項目を選択するか、0~9キーを押すことで、それぞれの着せ替え状態を表示できる。



あなたのデスクトップのマスコット

ActX

バージョン : 0.99.3p13 ライセンス : フリーソフト

<http://www.astr.tohoku.ac.jp/~taji/ActX/>

X 日本語可

「お座りマスコット」、英語圏では「Window Sitters」と呼ばれる X Window System上でマスコットを表示するプログラム。作者の努力により、ほとんどのウィンドウマネージャに対応している(日頃使っているWindow Makerでも問題なく動作した)。Xのフォーカスされたウィンドウのタイトルバー上にxpmで書いたファイルを貼付けることによって、アクティブなウィンドウのほうにあたかもマスコットが乗っているような効果を実現している。また、画像を切り替えることにより、アニメーション機能をも実現している。自分で好きなマスコットを作って、アニメーションさせることも可能だ。ふだんから、Xの画面がなんとなく寂しいなあと感じている方にとってはお勧めのソフトだ。キャラクタの女の子のグラフィックも、みんなかわいいので一度試してみてもいい。



デスクトップにペンギンを!

Penguineyes

バージョン : 0.10 ライセンス : GPL

<http://www.crathva.fsnet.co.uk/penguineyes.html>
<http://kopje.koffie.nu/ivo/tuXeyes/> (tuXeyes)

X GTK+

Xeyes (カーソルを見つめる目玉)のGTK+版。Linuxのマスコットキャラとしておなじみのペンギン「Tux君」がデスクトップに常駐し、マウスカーソルを目で追いかける。Tux君のサイズは、中ボタンドラッグやマウスホイールの回転により自由に変更できる。

外観を切り替えるテーマ機能を持っている。用意されているテーマには、Tux君の目玉の形を赤い点やハートに変更したり、おなかに時計を表示するといったバリエーションのほか、デザイン自体をGnuの「ヌー」や巨大な「ロボペンギン」、首から上だけの「Linus氏」などに切り替えることもできる。カーソルを追いかけるLinus氏の目つきは結構セクシーかも。なお、これらのテーマを切り替えた直後に表示がおかしくなる不具合があるが、一度左右反転などの操作を行うと正常に表示されるようになる。



かわいいペンギンがデスクトップを走り回る

xpenguins

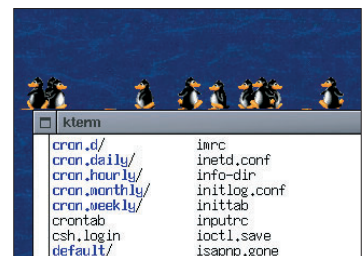
バージョン : 1.2 ライセンス : GPL

<http://www.met.rdg.ac.uk/~swrhgnrj/xpenguins/>

X

バックグラウンドを小さなペンギンが走り回るアミューズメントソフトウェア。Pingus (<http://pingus.seul.org/>)というLinux上のレミングスクローンソフトウェアから飛び出してきたかわいいキャラクタがバックグラウンド上を走り回る。起動するとバックグラウンドはペンギンたちでいつもにぎやか、という感じになる。

“-penguins” オプションで登場するペンギンの数を指定ができるが、あまり多くしすぎると、リソースを消費しすぎて、本当にやるべき作業ができなくなってしまうので注意しよう。ちなみに筆者の環境では、256羽モード(上限値)で起動したら、まったく仕事にならなくなってしまった。経験値から言って50羽くらいが適当か? 余談だが、ソフトウェアを途中で終了させたときの、ペンギンたちがブクッと膨れて破裂してしまうアニメーションが少し寂しさを感じさせるのであった。



30種類以上の画像形式に対応した画像管理ソフト

CompuPic

バージョン：4.6/5.1 (ベータ) ライセンス：シェアウェア (\$39.95)

<http://linux.compupic.com/>

X

Windowsソフトとして実績のある洗練された画像管理ソフトのLinux版。30種類以上の画像形式に対応しており、サムネイル(縮小画像)表示により大量の画像を効率よく管理できる。また、スライドショーやファイル管理、画像の加工といった機能も充実している。非商業利用の場合はフリーで使えるが、30日の試用期間を過ぎるとLZW圧縮を利用したGIF/TIFF形式を扱えなくなるので注意されたい。

起動すると、ディレクトリツリーとファイル一覧で構成されるウィンドウが開く(画面1)。ファイル一覧の表示形式は、サムネイル、詳細、一覧、ボタン状サムネイルで切り替え可能だ。画像ファイルのサムネイルは、はじめて表示する際に自動的に作成される。サムネイルをクリックしてファイルを選択すると、ディレクトリツリー表示の下に少し大きめのイメージが表示され、ダブルクリックかEnterキーにより実際の画像がフルスクリーンで表示される。

画像を連続表示する「スライドショー」では、一定時間(初期設定では3秒)で自動的に次の画像に切り替わるほか、カーソルキーやスペースキーで前後の画像に切り替えることも可能だ。同様の機能として、フルスクリーンを3x3のエリアに分割し、左上から順番に画像を表示

していく「マキシショー」も用意されている(画面2)。連続写真など、シリーズものの画像を眺めるのに便利だ。

大量の画像を管理している場合は「Advanced Batch Conversions」機能を活用したい。画像サイズや格納パスの変更、テキストや画像のオーバーレイといった作業内容をあらかじめダイアログで設定しておき、選択した複数のファイルに対して一括実行できる。このほか、複数のサムネイルをまとめたインデックス画像を作成したり、電子メールで送るためのグリーティングカードを作成するなど、ユニークな機能が豊富に用意されている。



画面1 フォルダツリーとファイル一覧表示で構成されるウィンドウ



画面2 複数の画像を並べて表示する「マキシショー」機能

さまざまなデジカメの画像を扱える画像ソフト

gPhoto

バージョン：0.4.3 ライセンス：GPL

<http://www.gphoto.org/>

X

GTK+

日本語可

デジタルカメラの画像を扱うことを目的に作られた画像ソフト。対応しているデジカメは100機種以上にも及ぶ。パソコンと専用ケーブルで接続されたデジカメに対し、ハードディスク上のディレクトリを扱う感覚で、サムネイル(縮小画像)や画像の取得、削除などを行える。なお、日本語に対応するには、ビルド前にsrc/main.cの“gtk_init(...);”の前に“gtk_set_locale();”を追加する必要がある。

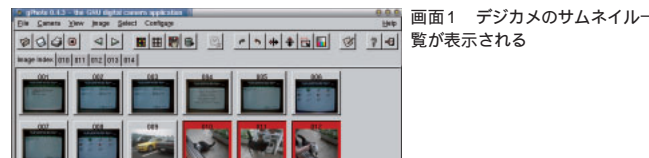
初めて起動したときには、デジカメの機種と使用するシリアルポートを選択して設定を保存する必要がある。ディレクトリのサムネイル一覧から選択した画像を表示する設定により、一般的な画像管理ソフトとして使うことも可能だ。続いて、パソコンとデジカメをケーブルで接続し、デジカメの電源を入れた状態で[Camera]-[Camera Summary]を選択して正常に接続されているかどうか確かめよう。

デジカメで撮影した画像を取得するには、最初にサムネイル(縮小画像)の一覧を取得し、それをインデックスとして使用する(画面1)。なお、取り出したい画像の番号があらかじめわかっている場合は、サムネイルなしのインデックスを使うこともできる。

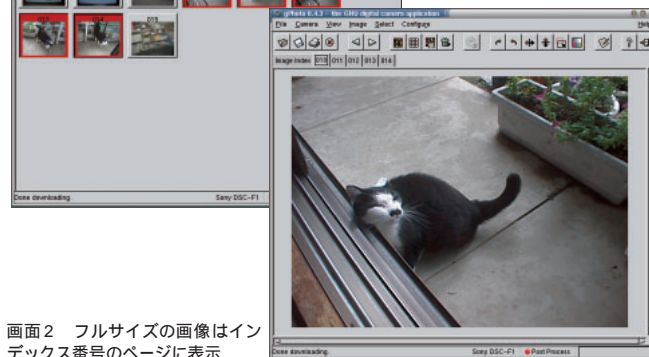
サムネイルを見ながら、フルサイズ画像を取得したいものを選択し

(複数選択可)、ツールバーの[Get Selected Images]ボタンを押すと、選択したインデックスのフルサイズ画像がデジカメから取得され、独立したページに表示される。ツールバーのボタンを使って、左右/上下の反転や90度回転、サイズ変更、カラーバランス調整などの画像処理が可能だ。JPEG、GIF、TIFF、PNG、BMPなどの画像形式でファイルに保存したり、プリンタで印刷することもできる。

なお、USBやIEEE1394経由での接続にも対応し、GTK+やコマンドラインなどのインターフェイスを切り替えて使える次世代バージョン「gPhoto2」の開発が現在進められている。



画面1 デジカメのサムネイル一覧が表示される



画面2 フルサイズの画像はインデックス番号のページに表示



ワンクリックで画像を表示するビューア

GQview

バージョン : 0.8.2

ライセンス : GPL

<http://gqview.sourceforge.net/>

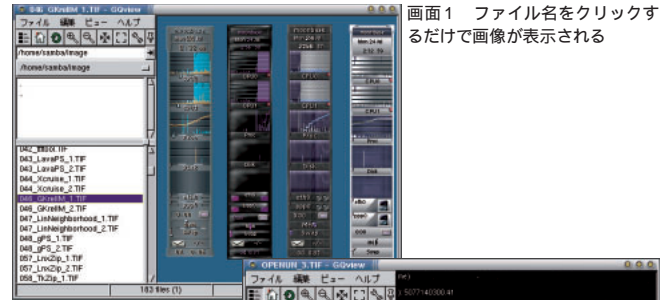
X GTK+ 日本語可

ファイル名をクリックするだけで画像を表示できる画像ビューア。サムネイル(縮小画像)表示やフルスクリーン表示、スライドショーなども可能だ。日本語カタログが標準で用意されており、メニューやオプションは日本語で表示される。

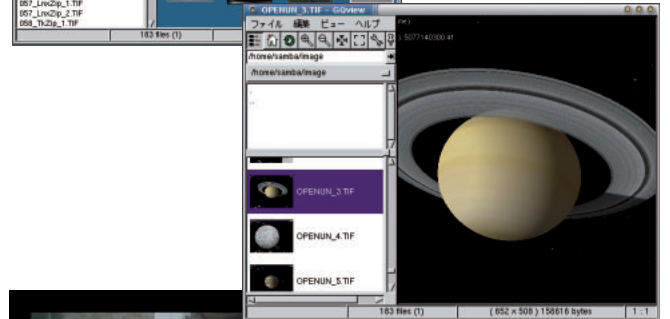
ウィンドウの左側にファイル一覧、右側に選択した画像のイメージが表示される(画面1)。ファイル一覧はサムネイル(縮小画像)表示に切り替えることも可能だ(画面2)。ファイル名やサムネイルをクリックすると、その画像が右側に表示される。画像を順次眺めるには、画像上でクリックするだけでいい。

また、右ボタンメニューにより、画像を独立したウィンドウに表示したり、フルスクリーン表示(画面3)に切り替えることもできる。外部ソフトの起動やファイル処理も可能だ。

キーボードにも同様のコマンドが割り当てられている。たとえば、フルスクリーン表示(トグル)はVキー、前後の画像への切り替えはPageUp / Downキーだ。さらに、画像にフォーカス(黒枠)がある場合は、カーソルキーで画像をスクロールしたり、スペースキーで順次画像を切り替えて表示することもできる。



画面1 ファイル名をクリックするだけで画像が表示される



画面2 一覧表示をサムネイルに切り替えることも可能だ



画面3 画像サイズが大きい場合はフルスクリーン表示を活用しよう

サムネイル表示に対応したイメージビューア

kShow

バージョン : 0.2.1

ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

JPEG、GIF、XPMが扱えるイメージビューア。表示形式として、サムネイル形式、スライド表示などが選択可能で、通常のイメージビューアよりも重要な優れものだ。

インストールは、通常の“./configure” “make” “make install”で行えるが、KDEとQtライブラリが必要になる。インストールすると、アプリケーションスタータの[グラフィック]に登録され、このメニューから選択して起動できるようになる。

[ファイル]メニューから[開く]を選択すると、ファイルやディレクトリを選択できるダイアログが開く。ここで、ディレクトリを指定すれば、そのディレクトリにある表示可能な画像ファイルが読み込まれるので、あとは、これをどのように閲覧するかを指定すればよい。[イメージ]メニューから[情報]を選択すると画像に関する情報が表示され、同じく[リスト]を選択すると、現在読み込まれているカタログのファイルがリストアップされる。[サムネイル]を選択すると、サムネイル形式で表示されるようになる。また、画像上でマウスの右クリックで現れるメニューで、画像の等倍、拡大、縮小などの表示モードを切り替えることもできる。

自分自身用のカタログを作成し、そのカタログを編集することも可能だ。[イメージ]メニューから[リスト]を選択すると、リストダイアログが表示され、ここで[保存]を選択することによって、新しいカタログとして登録される。このカタログは[ファイル]メニューの[カタログを開く]から利用できる。

このほか、画像ファイル操作としては、ファイルを指定の場所にコピーする機能も用意されている。[オプション]メニューで表示されるダイアログで設定可能な項目も2つ紹介しておこう。[ウィンドウ]タブでは、ウィンドウの仕様を変更できる。画像のサイズ(縦横)に合わせてウィンドウの大きさを随時変更させるか、それとも固定の大きさを使用するかを指定する。デフォルトでは、画像の大きさに合わせて、ウィンドウの大きさも変更されるようになっている。[イメージ表示]タブでは、スライド形式で表示する際の詳細設定を行う。[Delay Time]で次の画像を表示するまでの時間、[次へ]、[戻る] [ランダム表示]のラジオボタンで表示する順番、[ループ]でスライドをループさせるかどうかをそれぞれ指定する。



画面1 サムネイル表示。複数の画面に分かれる場合は下部にあるボタンで表示を切り替える

3Dモデルをさまざまな角度から眺められるビューア

Morpheus

バージョン : 0.3

ライセンス : GPL

<http://wine.sexcity.pl/morpheus/><http://www.student.oulu.fi/~jlof/gtkglarea/> (GtkGLArea)

X

Mesa

GTK+

GNOME

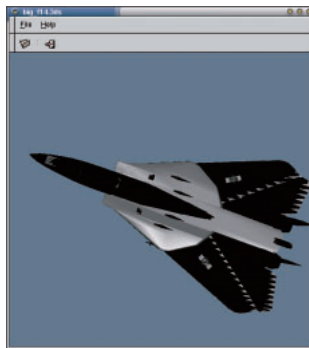
3Dモデルを表示し、さまざまな角度から眺められるGNOME用の3Dモデルビューア。Morpheus本体とライブラリのlibmorphで構成される。このほか、ツールキットのGtkGLAreaが別途必要となる。

現時点对応している3Dモデルは、LightWaveのデータ形式(拡張子lwob)と3D Studioのデータ形式(拡張子3ds)の2種類。POV-RayやDXF形式などに対応する予定もあるようだ。

なお、3Dモデルのデータは付属しないので、「3Dカフェ」(<http://www.3dcafe.com/asp/meshes.asp>)などのWebサイトからフリーなデータを入手しよう。

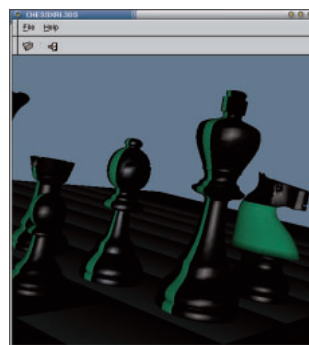
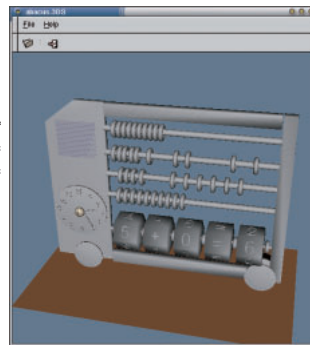
オープンダイアログで3Dモデルのデータを選択するか、GMCなどのファイルマネージャからMorpheusのウィンドウにデータをドラッグ&ドロップすると、OpenGL(またはMesa)でレンダリングされた画像がウィンドウに表示される(画面1~3)。

ウィンドウのサイズを自由に変更できるほか、左ボタンのドラッグでレンダリング画像を回転させたり、中ボタンを上下にドラッグして拡大/縮小することも可能だ。



画面1 サンプルデータは付属しないので別途入手する必要がある

画面2 画面ではわかりづらいが、3Dモデルなので視点を変えて見ることもできる。



画面3 レンダリングした画像は拡大表示も可能

さまざまなフラクタル図形を高速に表示

XFrackey

バージョン : 2.51

ライセンス : フリー

<http://www.gk.dtu.dk/~hwj/xfrackey/index.html>

X

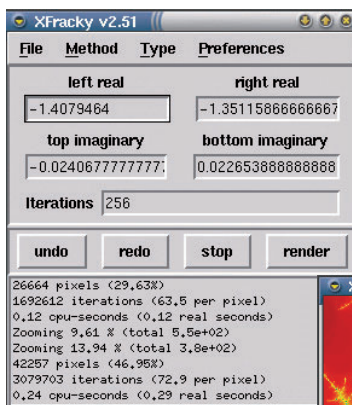
Tcl/Tk

マンデルブロート集合やジュリア集合などを利用したさまざまなフラクタル図形を、境界スキャンと反復追跡により瞬時に表示するソフト。一般的な2D平面図のほか、3D俯瞰図の表示も可能だ。

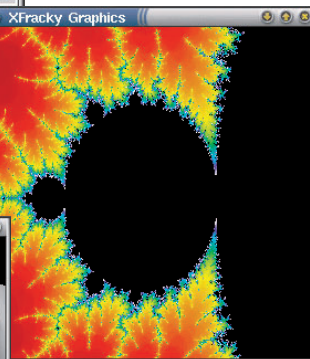
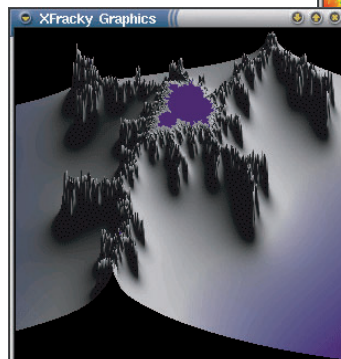
起動すると、メインウィンドウとグラフィックスウィンドウが開いてマンデルブロート集合の2D平面図が表示される。メインウィンドウで表示範囲などのパラメータを数値で設定するか(画面1)、グラフィックスウィンドウ上でドラッグして範囲を指定して[render]ボタンを押すと、指定範囲のフラクタル図形が描画される(画面2)。高速なアルゴリズムを採用しているため、表示は一瞬で行われる。

3D俯瞰図に切り替えるには、メインウィンドウの[Method] - [3d]を選択後、[render]ボタンを押して再描画すればいい(画面3)。また、フラクタルの種類は[Type]メニューで切り替えられる。

このほか、一定時間(初期設定は5秒)ごとにさまざまなフラクタル図形を自動的に切り替えて表示するデモモードや、気に入ったフラクタル図形のパラメータをファイルに保存する機能、フラクタル図形をTGA形式の画像ファイルとして保存するエクスポート機能などが用意されている。



画面1 メインウィンドウでフラクタル図形のパラメータを設定



画面2 フラクタル図形の表示は一瞬で完了する

画面3 斜め上方向から眺めた3D俯瞰図の表示も可能だ



形と模様の変化を個別に制御できるモーフィングソフト

xmrm

バージョン : 2.0

ライセンス : GPL

<http://www.cg.tuwien.ac.at/xmrm/>
<http://bragg.phys.uwm.edu/xforms> (Xforms)

X

映画やCMでおなじみのモーフィングを行うソフト。2つの画像から、複数の中間画像を自動的に生成してアニメーション表示する。動作にはハイカラー以上の環境とツールキット「Xforms」が必要だ。Xformsをスタティックリンクしたバイナリ版も配布されている。

いきなりデータを作成するのは大変なので、まずはバイナリ版のアーカイブに含まれるサンプルデータを使ってみよう。morph.prjというプロジェクトを読み込むと、白熊と豹の画像が多数のベクトル(矢印)付きで表示される(画面1)。これらのベクトルは2つの画像の対応する部分を示している。サンプルの場合はすでに設定済みだが、自分でデータを作成する場合はベクトルを根気良く設定する必要がある。続いて、左上の[Number of Frames]で適当なフレーム数(30以上)を設定して[CALCULATE]ボタンを押すと、中間画像が自動的に生成され、[ANIMATE]ボタンでアニメーションが表示される(画面2)。アニメーションの速度を変更したり、中間画像をコマ送り表示することも可能だ。

左下の2つのグラフを操作すると、モーフィングの進行を調整できる。左側のグラフが模様などの細部の特徴、右側のグラフが形などの大ま

かな特徴に対応している。途中から急激に模様を変化させたり、途中で形が逆戻りするといった制御が可能だ。このほか、模様の変化速度を部分的に変えたり(Detail Map Morph)、1画像だけを利用して形を変化させる(Warp Source Image)機能も用意されている。

生成された画像はTIFF形式で個別に保存できる。ファイル名には自動的に3桁の連番が付くので、GIMPなどを使ってアニメーションGIFに変換したり、付属のxmrm_mpegを使ってMPEG形式の動画に変換すればいい。なお、xmrm_mpegの実行にはMPEGエンコーダやTIFFファイルからのコンバータなどが別途必要だ。



画面1 まずはサンプルデータをもーフィングさせてみよう

画面2 生成結果をアニメーション表示。コマ送りも可能だ

GUIのビデオプレーヤ

aKtion!

バージョン : 0.4.1

ライセンス : GPL

<http://www.geocities.com/SiliconValley/Haven/3864/aktion.html>

X

Qt

aKtion!は、Qtライブラリを使ったGUIのインターフェイスを持つビデオプレーヤだ。マルチメディアプレーヤ「xanim」(<http://xanim.va.pubnix.com/home.html>)のフロントエンドとして動作するため、xanim 2.70.7.0以降が別途必要となる。

xanimは、AVI、Quicktime、MPEGといった一般的なビデオフォーマットをはじめとする、さまざまなマルチメディアフォーマットに対応している。高機能なマルチメディアプレーヤだが、その多くをコマンドラインオプションで指定しなければならず、使いにくさは否めない。aKtion!を使えば、手軽にいろいろなビデオファイルを鑑賞できるようになる。ただし、MPEGに関しては、xanimでサウンドがサポートされていないので、aKtion!でも当然サウンドは再生されない。

aKtion!では、再生するファイルの指定、ボリューム調整、カラーマッピングなどをGUIで設定できる。また、再生中に「c」キーを押すことにより、動画フレームのキャプチャーも可能だ。Qtライブラリを使っているため、外観や操作性は他のKDE用のソフトと統一されている。

Red Hat 6.x用のRPMパッケージが用意されているが、そのほかのディストリビューションではtarボールのソースから自分でビルドする必要

がある。インストールするとKDEの[マルチメディア]メニューの中にaKtion!が追加され、さらにMIMEタイプへの関連付けもされるのでKFMの中の動画ファイルをクリックするだけで、再生できるようになるのは親切だ。xanimは公式のWebページではソースしか配布されていないが、<http://www-chaos.umd.edu/dsweet/XAnim>で、Red Hat用のRPMパッケージを入手することができる。

なお、xanimは24bitカラーには対応していない。そのため24bitカラーでXが起動されていると、aKtion!でビデオファイルを指定したとき、「Error Reading file info.」というメッセージが出る。そのときはいったんXを終了後、「startx --bpp 16」などとして、24bitカラー以外のモードを指定して、Xを起動しなおそう。



画面1 各種フォーマットのビデオデータをGUIで手軽に再生できる

画面2 オプション指定のウィンドウ。xanimに渡すオプションを直接指定することも可能

テーマにも対応した高速MP3プレーヤ

FreeAmp

バージョン : 2.0.8

ライセンス : GPL

<http://www.freeamp.org/>

X

GTK+

LinuxやWindows 9x/NTなどで動作する完全にフリーなMP3プレーヤ。元々はクールな外見と安定性で定評のあるプレーヤだったが、バージョン2以降では外見を変化させる「テーマ」、ディスク上のMP3ファイルやプレイリストを一括管理する「My Music」、ダウンロードマネージャなど大幅に機能が強化された。EsounDサウンドデーモンや、ALSAサウンドドライバにも対応している。

ウィンドウのデザインはシンプルなパネル風で、大きさは3段階に変更できる(画面1)。再生や停止などのボタン類は説明するまでもないだろう。なお、サウンドデーモンEsounD(esd)を利用している場合は、FreeAmpの出力モジュールを初期設定のsoundcard.pmoからesound.pmoに変更する必要がある。

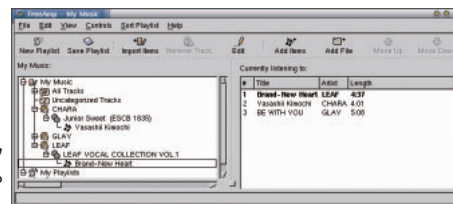
FreeAmpには、ディスク上に散在するMP3ファイルやプレイリストを検索して一括管理する「My Music」機能が用意されている。My Musicウィンドウの左側には検索されたMP3ファイルやプレイリストがツリー表示され、右側には現在のプレイリストの内容が表示される(画面2)。プレイリストに曲を追加するには、ツリー中のMP3ファイルやプレイリストをダブルクリックするだけでいい。

FreeAmp用の「テーマ」(いわゆる「スキン」)は、Xmms用ほど数は多くないものの、FreeAmpのWebサイトや米コカ・コーラ社のWebサイト(http://www.coca-cola.com/moments/turn_up/mp3_index_frm.html)などから入手できる。取得したテーマは、`/usr/share/freeamp/themes`ディレクトリにコピーしておけばいい。

また、Real Jukeboxのダウンロード用形式であるRMP(Real Music Package)に対応したダウンロードマネージャを内蔵しており、Real Jukebox用の合法的なMP3ファイルが置かれたWebサイトからMP3ファイルをダウンロードできる。



画面1 シンプルなパネル風のウィンドウは3段階に変更可能だ



画面2 散在するMP3ファイルを一括管理する「My Music」

CDリッパーやMP3エンコーダと連動するCDプレーヤ

Grip/GCD

バージョン : 2.94

ライセンス : GPL

<http://www.nostatic.org/grip/>

X

GTK+

日本語可

CDリッパーやMP3エンコーダと組み合わせて、MP3作成のフロントエンドとして使えるCDプレーヤソフト。フロントエンド部分を持たないGCDも用意されている。いずれもCDDDBに対応しており、インターネット上のCDDDBサーバから音楽CDのタイトルや曲目情報を取得可能だ。また、CDDDBからは取得したり登録したりすることはできないものの、日本語のタイトルや曲目も問題なく入力できる。

Gripを起動すると、上部に曲名などが並んだトラックリスト、下部に操作のボタンが並んだウィンドウが開く(画面1)。トラックリスト部分は、リップや設定変更、バージョン表示などの画面に切り替えられる。また、タイトルや曲名を入力するディスクエディタ、ボリュームコントロール、プレイモードコントロールなどの表示をオン/オフするボタンも用意されている。

音楽CDからMP3ファイルを作成するには、音楽CDをCD-ROMドライブに入れ、トラックリストからリップしたい曲(複数可)を右クリックで選択する。選択が完了したら、[Rip]ページに切り替えて[Rip + Encode]ボタンを押せばいい。あとは、自動的にCDリッパーとMP3エンコーダがGripから起動されてリップとエンコードが行われる。使用

するCDリッパーやMP3エンコーダは自由に設定できる。たとえば、国産の高速MP3エンコーダ「午後のこゝだ」を利用する場合、実行ファイルを「gogo」、コマンドラインを「%f -b %b」に設定すればいい。なお、内蔵のCDリッパーとしてはcdparanoiaライブラリが使われている。

一方、GCDのほうは、GripからCDリッパーやMP3エンコーダのフロントエンド部分を取り去ったもの(画面2)。それ以外の機能(CDDDBなど)はGripとまったく同じだ。GCDのファイルサイズはGripの半分以上なので、MP3のエンコードなどを行わず、純粋にCDプレーヤとして使うだけならGCDのほうがお勧めだ。



画面1 トラックリストとボタンのみ表示される起動直後のGrip

画面2 フロントエンド関係の部分を取り除いたGCD



KDEデスクトップ環境で動作するMP3プレーヤ

kmp3

バージョン : 1.0

ライセンス : GPL

<http://www.kde.org/>



その名前のとおり、KDE上で動くMP3プレーヤ。インストールすると、KDEのスタートメニューにある[マルチメディア]にメニュー項目として登録される。Kパネルに小さなドック化アイコンとして表示することもできる。標準的な機能を備え、ドック化してコンパクトにすることで他の作業の邪魔にもならないコンパクトなプレーヤだ。

まずはじめに、再生リストエディタを開き(CDプレーヤの取り出しボタンの部分をクリックする)再生するMP3データをリストアップしよう。ここで編集したリストを保存し、あとから再び読み込んで再生することも可能だ。もちろん、それぞれの曲のタグ情報も編集できるようになっている。タグ情報の編集には、タグエディタを使用する。エディタ起動は、メインパネルの「MP3」のロゴ部分をクリックすればいい。タイトルとアーティスト名をはじめ、ジャンル、この曲に対するコメントなどを自由に書き込むことができる。

このほかの再生、早送り/巻戻し、一時停止といった音楽を演奏するための操作インターフェイスは、CDプレーヤにそっくりなので(画面1)、あらためて説明するまでもないだろう。

なお、Kmp3のバージョン1.0-prexを使用していて、Kmp3を終了し、

再びKmp3を起動した場合、MP3ファイルの再生ができないという問題が報告されている。それを回避するための方法を紹介しておこう。すでに終了してしまった場合は、/.kde/share/apps/kmp3/にあるdefaultファイルを空にし、それからKmp3を起動するようにしよう。終了する前であれば、再生リストを空にし、[適用]ボタンを押してから終了するという手順を踏めばOKだ。

この問題は、正規バージョンになった時点で解消されており、最新バージョンを使用している限り問題はない。しかし、とりあえず古いバージョンを使用していて、この問題が発生している場合、その都度再生リストにmp3ファイルを読み込ませるのは面倒なので、読み込ませた再生リストを名前を付けて保存しておき、終了する際には現在読み込んでいるリストを空にして、次回起動する際にはこれを読み込ませるといふ具合にすれば、かなり楽ができる。

付け加えると、このようにしておけば、自分のお気に入りの曲目リストをいくつも用意して気分によってリストをチェンジすることもできるのだ。「ものは考えよう」ということで、MP3のあるLinuxライフをエンジョイしよう!



画面1 CDプレーヤと同じ感覚で操作できる

多機能なMPEGビデオプレーヤ

MpegTV Player

バージョン : 1.1.4

ライセンス : フリーソフト

<http://www.mpegtv.com/>



MpegTV Playerは、MPEG1形式の動画や音声、さらにVIDEO CDを再生することのできるプレーヤだ。コマンドラインからファイルを指定するmtvpと、GUIのコントローラを起動するmtvがある。再生能力はMPEGプレーヤの中でも比較的高いほうだろう。入力データは、ファイルだけでなく、標準入力から受け取ったり、URLで指定することも可能だ。

コントラスト、ガンマ値のほか、再生レート、音声のクオリティといったオプションも設定可能だ。GUI版で指定できるオプションは、コマンドライン版でも指定できるので、単にMPEGファイルを見たいだけであれば、コマンドライン版でも十分だろう。ただし、フレーム操作などの細かい操作が必要な場合には、GUI版を使ったほうが簡単だ。

また、コマンドライン版では、さまざまな環境変数を設定することにより、動作を制御できる。これには、GUI版のオプション画面で設定できないものも含まれている。たとえば、HTTPプロキシのホスト名を設定する“MTVP_HTTP_PROXY_HOSTNAME”や、使用するオーディオデバイスを指定する“MTVP_AUDIO_DEVICE_MAIN”などだ。これらのオプションはコマンドラインで“mtvp -h”とすることで確認で

きる。通常は指定する必要がないものがほとんどだが、うまく再生できない場合などに、試してみるとよいだろう。

フルスクリーンモードで再生することもできるが、この場合Simple DirectMedia Layer Library (SDL)が必要となる。なお、マルチメディアプラグインのPluggger (<http://www.hubbe.net/hubbe/pluggger.html>)をインストールすることにより、Netscape Navigator内でMpeg TVを使って直接MPEGファイルを再生することも可能だ。

実行ファイルはglibc、libc5に対応し、x86、LinuxPPC、Alphaといった幅広いプラットフォーム向けに、RPM、deb、tarボールなどの形式で用意されている。同梱のコマンドライン版のmtvpのみ、個人、教育用などの非営利使用に限りフリー。GUIのコントローラを利用するmtvはシェアウェア(10USドル)。未登録の場合、再生時間30秒という制限がある。



画面1 GUI版はコントロールパネルが表示される

画面2 plugggerでNetscape Navigatorに表示されるMPEGデータ

コマンドラインで動作するMP3 プレーヤ

mpg123

バージョン: 0.59r

ライセンス: フリー

<http://www-ti.informatik.uni-tuebingen.de/hippm/mpg123.html>

コンソール

コマンドラインで動作するMP3プレーヤ。MPEG 1 (および2) の Audio Layer1 ~ 3に対応している。XmmsやFreeAmpのようなGUIを持たない代わりに処理はそれだけ軽くて済む。フロントエンドから制御するためのリモートインターフェイスや、デコードしたデータをWAVEファイルに保存したり標準出力に書き出す機能も用意されている。

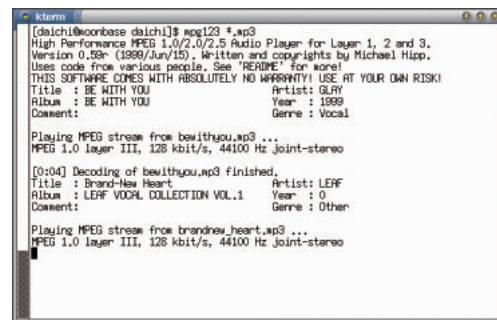
MP3ファイルを再生するには、ktermなどのコマンドラインで“mpg123 ファイル名”とする(画面1)。複数のファイル名を指定すると、それらを順番に再生する。Ctrl - Cキーで次の曲にスキップでき、1秒以内にCtrl - Cを2回押すとmpg123自体が終了する。なお、通常は再生中のファイル名、形式、ビットレート、サンプリング周波数などが表示されるので、バックグラウンドで再生する場合はメッセージ表示を抑制する-qオプションを指定するとい。

また、ファイル名の代わりに「http:」で始まるURLを指定すると、HTTPを利用したストリーム再生が行われる。ただし、ある程度高速な回線でないリアルタイム再生は難しいので、-bオプションによるバッファ指定を併用する必要がある。

ファイル名やURLが列挙された「プレイリスト」を利用するには、

“mpg123 -@ ファイル名”のように-@オプションでプレイリストのファイル(拡張子m3u)を指定する。なお、Netscapeのヘルパーアプリとして、MIMEタイプに「audio/x-mpegurl」、拡張子「m3u」に対して“mpg123 -@ %s”(または“kterm -e mpg123 -@ %s”)を設定すると、Webページ上のプレイリストを利用した再生が可能だ。

コマンドラインで使うのはちょっと.....という人は、X上で動作するmpg123用のフロントエンドがいくつか作られている。たとえば、Tcl/Tkで書かれたtk3play(画面2)やGQmpegなどだ。これらはmpg123をバックグラウンドで実行することでMP3の再生を行う。



画面1 再生中のタイトルなどが端末画面に表示される



画面2 Tcl/Tkを利用したmpg123のフロントエンド「tk3play」

X上で動作する定番のMP3 プレーヤ

Xmms

バージョン: 1.2.2

ライセンス: GPL

<http://www.xmms.org/>

X

GTK+

xmmsは、X Window System上では定番ともいえるMP3プレーヤだ。多くのディストリビューションでは、標準でインストールされている。インストールされない場合も、パッケージが用意されている場合がほとんどだろう。

画面や操作方法も、Windowsでやはり定番といえるWinAMPとほとんど同じだ。WinAmpを使ったことがあれば、操作にとまどうことはないだろうし、たとえ初めてでも、カーステレオのパネルのような画面は、すぐに扱えるようになるだろう。

xmmsとはX MultiMedia Systemの略で、MP3、WAV、MODのファイルフォーマットに標準で対応しているほか、プラグインシステムにより、VQF、MIDIといったフォーマットや、ラジオの受信(もちろん、ラジオチューナーカードは別途必要)などの、さまざまな音源に対応する優れものだ。さらには、インターネット上に存在するMP3ファイルを、URLの指定により演奏することもできる。

ビジュアル面でも、表示画面を変更する「スキン」機能をはじめとして、派手なボリュームメータの追加から、フルスクリーンでのさまざまな特殊効果の表示まで、プラグインによりいろいろと楽しめるように

なっている。これらのスキンやプラグインは、上記のxmmsの公式サイトで、いろいろと見つけることができる。また、スキンに関してはWinAMPのものがそのまま使えるので、<http://www.winamp.com/>で探してみるのもよいだろう。スキンそのものは、画面を構成するBMPファイルをZIPでまとめただけなので、自分で作成することも可能だ。

イコライザやプレイリストといった必要と思われる機能も充実しており、高機能マルチメディアプレーヤとして、しばらくは定番の座を譲ることはないだろう。



画面1 イコライザやプレイリストを表示させてみた



画面2 自作のスキンで画面を変えたxmms



再生中のサウンドをさまざまに視覚化する

eXtace

バージョン : 1.2.25 ライセンス : GPL

<http://extace.sourceforge.net/>
<http://www.fftw.org/> (FFTW)

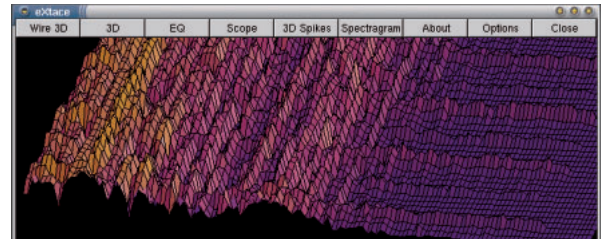
X GTK+

再生中のサウンドを高速フーリエ変換 (FFT) して得られたスペクトル情報をリアルタイムに視覚化するソフト。

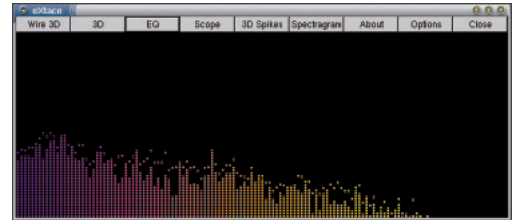
表示方法は、3Dワイヤフレーム、サーフェス、スパイク、グラフィックイコライザ、グラディエントスコープ、スペクトログラムの6種類で、再生中もボタン操作でこれらを自由に切り替えられる。動作にはFFTWライブラリが必要だ。

起動すると、上部にボタンが並んだウィンドウが開く。その後、EsounD (esd) を利用したサウンド再生ソフト (FreeAmpなど) を使用すると、FFTにより得られたスペクトル情報がリアルタイムに3Dサーフェス表示される (画面1)。

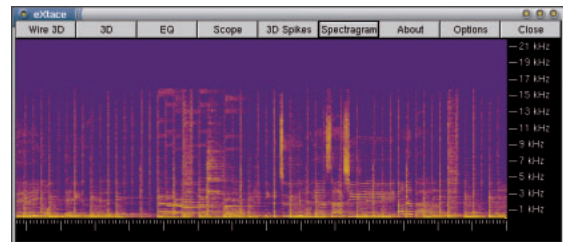
表示方法を変更するには、ウィンドウ上部のボタンを押す。サウンド再生中の切り替えも可能だ。3D表示は、サーフェス以外にワイヤフレームとスパイクなどが用意されており、表示の傾きや更新速度を [Direction] ウィンドウで変更できる。イコライザ (画面2)、グラディエントスコープ、スペクトログラム (画面3) への切り替えも可能だ。また、[Option] ボタンで開くダイアログでは、バンド数やディケイ速度などを変えられる。



画面1 再生されるサウンドのスペクトル情報を3Dサーフェス表示



画面2 2Dのイコライザ表示。ディケイをかけたときカッコいい!



画面3 スペクトログラム (スペクトルの時間的変化) の表示も可能だ

WAVとMP3に対応したウェーブエディタ

Gmurf

バージョン : 0.6 ライセンス : GPL

<http://www.epita.fr:8000/epx/projs/gmurf/>

X GTK+ Tcl/Tk

Gmurfは、WindowsのWAVデータとMP3フォーマットに対応したウェーブエディタだ。ミキサー機能も付いており、ラインやマイクからの入力やボリューム、バランスの調整もできる。

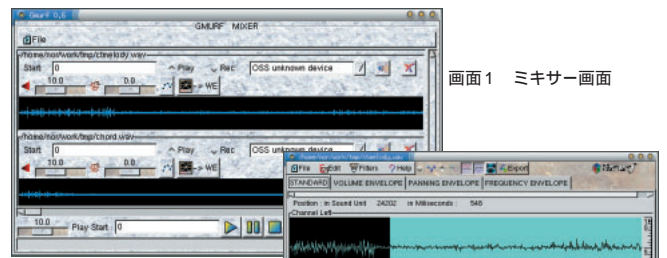
ウェーブエディタとしては、波形のズームイン/アウト、削除・コピー・ペーストはもちろんのこと、プラグインを利用することによって、エコーやリバーブ、周波数の変更など、さまざまなエフェクトをかけることができる (標準で用意されているフィルタは表1を参照)。どの操作も波形を見ながら直感的に行えるのが特徴だ。作者のWebページにはプラグインの作成方法も解説されているので、欲しいエフェクトを自分で作ることも可能だ。

ウェーブエディタでフィルタをかけたものを、そのまま保存してもよし、いくつかのファイルをミキサーで合成して保存してもよい。

インストールは、Red Hat用のRPM/パッケージとソースファイルが用意されている。サウンドデバイスはOSSとEsounDをサポートしているが、最新のEsounDではうまく動作しないので、バージョン0.2.8を使う必要がある。なお、コンパイルと実行には、GTK+のバージョン1.2およびtcl/tkが必要となる。

表1 用意されているフィルタ

Sox	Fender Vibro-Champ effect
	Phaser
	Lowpass Filter
	Highpass Filter
	Flanger
	Echoing
	Chorus
Gmurf	Bandpass Filter
	Echoes Generator
	Volume adjustment / Saturation
	Reverse wave
Experimental	Loop wave
	Frequency Modulator
	Sox Reverberation
	Expander



画面1 ミキサー画面



画面2 各ファイルの編集画面

演奏中のCDの音などをグラフィカルに表示

Synaesthesia

バージョン: 2.1

ライセンス: GPL

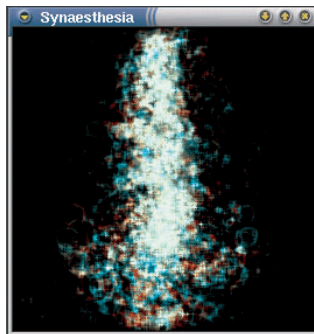
<http://yoyo.cc.monash.edu.au/pfh/synaesthesia.html>

X

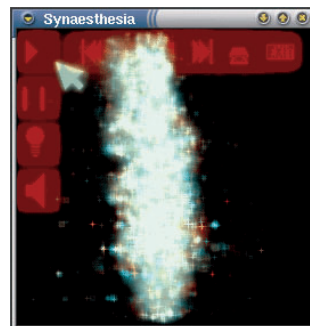
再生している音をきらめくグラフィックで視覚化してくれる一風変わったCDプレーヤ。グラフィックは、音の位置や周波数成分などに応じて変化する。バージョン2以降では、従来の「スター」に加え、「ウェーブ」や「フレーム」などの新たなエフェクトが追加され、それぞれのパラメータも調整できるようになった。

“synaesthesia cd”として起動すると、CD-ROMドライブにセットされた音楽CDを自動的に再生しながら視覚化する(画面1)。再生・停止、トラック移動、エフェクト・ボリュームの変更などの操作はウィンドウ上のアイコンで行う(画面2)。これらのアイコンは、グラフィック表示の邪魔にはならないよう、一定時間操作しないと自動的に隠れてしまう(マウスを動かすと再び表示される)。

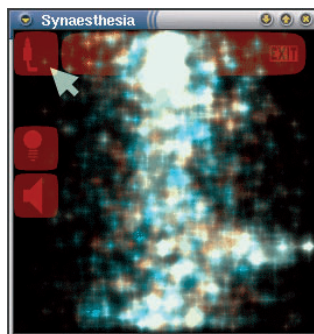
このほかにも、他のソフトの標準出力をパイプ経由で受け取ったり、サウンドカードのライン入力をモニタして、同様のグラフィックを表示することも可能だ。たとえば、“mpg123 -s MP3ファイル名 | synaesthesia pipe 44100”とすると、mpg123でデコードされたMP3データをパイプ経由で受け取って視覚化と再生を行う(画面3)。



画面1 再生される音をきらめくグラフィックで視覚化する



画面2 先送りなどのアイコンは操作している間だけ表示される



画面3 mpg123でデコードされた音をパイプ経由で受け取る

高機能なソフトウェアシンセサイザ

TiMidity++

バージョン: 2.9.5

ライセンス: GPL

<http://www.goice.co.jp/member/mo/timidity/>

X

コンソール

日本語可

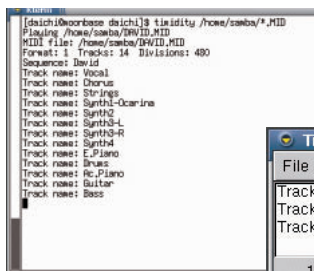
MIDI音源を持たないマシンでも、MIDIデータの再生を可能にするソフトウェアシンセサイザ。あらかじめ用意した音源データ(GUS互換のパッチファイル)とMIDIデータの情報から実際の音の波形を生成し、リアルタイムに音楽を再生してくれる。生成した音をWAVEデータとしてファイルに保存したり、HTTPプロトコルなどを利用してネットワーク上のファイルを再生することも可能だ。

なお、音源データはTiMidity++のアーカイブには含まれていないので、適当なパッチファイルを別途取得する必要がある。たとえば、作者の出雲正尚さんのページには、合計18Mバイト程度の高品質なパッチファイルが設定ファイル付きで用意されている。これで大きすぎるなら、松本庄司さんのページ(<http://www.i.h.kyoto-u.ac.jp/~shom/timidity/>)にある、10Mバイトや4Mバイトのパッチファイルを利用するといいい。

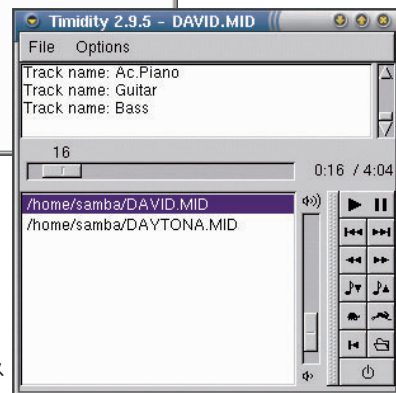
TiMidity++を使ってMIDIデータを再生するには、“timidity ファイル名”とする(複数指定可)。標準MIDI形式、レコンポーザ形式など各種のMIDI形式に対応しているほか、“timidity 'アーカイブ名#ファイル名'”という書式で、アーカイブ内のファイルを直接再生することも可能だ。ファイル名の部分にワイルドカードを使って複数のファイルを再生した

り、ファイル名を省略してアーカイブ内のすべてのファイルを再生することもできる。

こうしたコンソールベースでの利用のほか、TiMidity++にはGTK+やTcl/Tk、ncurses、slang、Motif、Emacsなどを使った各種インターフェイスも用意されている(ビルド時に指定する必要あり)。これらを利用するには、-iオプションに続けてインターフェイスの種類を指定すればいい。たとえば“-ig”とするとGTK+インターフェイスが選択され、GUI操作でMIDIファイルのオープンやプレイリストの編集、再生・スキップといった操作を行える(画面2)。



画面1 通常はコンソールベースで動作する



画面2 GTK+インターフェイスによる表示(-igオプション)



サウンドを視覚化するGNOMEアプレット

Visual Sound Analyzer (VSA)

バージョン : 0.9.7 ライセンス : GPL

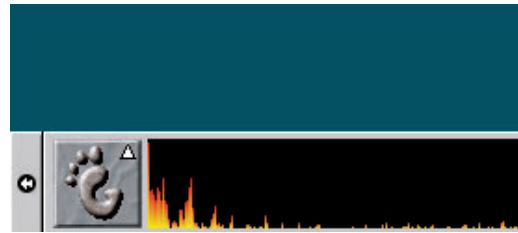
<http://vsa.linuxcore.com/>
<http://www.fftw.org/> (FFTW)

X GTK+ GNOME 日本語可

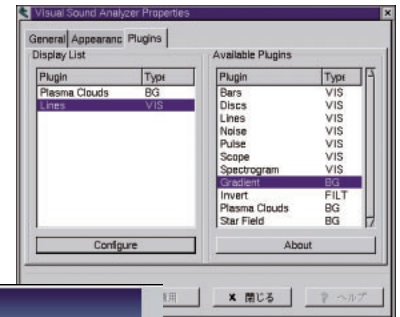
再生中のサウンドを高速フーリエ変換 (FFT) し、スペクトラムアナライザ風のグラフィックをリアルタイムに表示するGNOMEアプレット。背景やグラフの表示がプラグイン形式になっており、好みに合わせて自由に組み合わせられるのが特徴だ。実行には高速フーリエ変換ライブラリFFTWが別途必要だ。

Esoundが動作している状態でVSAを起動すると、GNOMEのパネルに表示領域が確保され、Esound対応のWAVEプレーヤーやMP3プレーヤーでサウンドが出力される際に、スペイン風のグラフィックが表示されるようになる (画面1)。

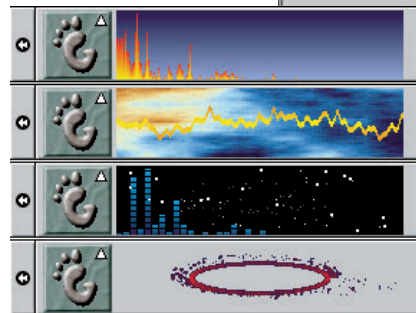
プロパティダイアログの [Plugin] ページでは、グラフや背景の表示に使われるプラグインを変更できる (画面2)。グラフ (BIS)、背景 (BG)、フィルタ (FILT) のプラグインが10個以上用意されており、自由に組み合わせて使用できる (画面3)。同種のプラグインを複数同時に使うことも可能だ。登録や表示順の変更はドラッグ&ドロップで行う。各プラグインの細かい設定も変更可能だ。



画面1 サウンドの出力に応じたグラフィックが表示される



画面2 プロパティダイアログでプラグインを設定する



画面3 グラフや背景のプラグインを複数組み合わせられる

Column

フリーソフト探しましよ

Windowsに比べると、まだまだ数が少ないLinuxのフリーソフトだが、それでも、用途や好みにあったソフトを探し出すのは結構大変なもの。開発プロジェクトごとに独自のWebサイトを設けて公開していることが多く、gooやinfoseekなどの一般の検索サイトでは情報を絞り込むのが難しい。

そんなときは、専用のソフトウェアアーカイブサイトを利用しよう。こうしたサイトの中では、WindowsやMacのフリー/シェアウェアを集めた「窓の杜」と「Vector」が国内ではポピュラーな存在だろう。

Linux関連の国内サイトとしては「ペンギン活用委員会」 (http://www.pcunix.org/linux_soft/) がお勧め。ソフトウェア自体を集めたアーカイブサイトではないが、カテゴリごとに分類されたソフトウェア紹介ページに、それぞれの入手先がリンクされていて便利だ。

海外に目を向けると、巨大なアーカイブサ

イトも存在する。ここでは、代表的な2つのサイトを紹介しておこう。

SourceForge

オープンソースソフトウェアの開発をトータルにバックアップするネットワークプロジェクト。開発プロジェクトのサイト運営のサポート (サーバスペースの提供、変更履歴やバグレポートの管理) を目的としているが、それぞれの成果物もダウンロードできるよう

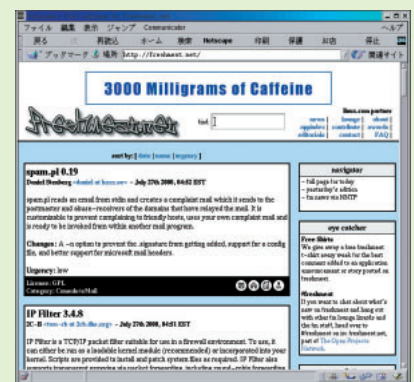


SourceForge <http://sourceforge.net/>

になっている。現在、4000近くのプロジェクトが登録されている。

freshmeat

毎日更新されているLinuxアプリケーションを紹介するサイト。トップページにはその日に更新されたソフトの情報が紹介されている。数え切れないほどの登録数だが、インデックスや検索機能も用意されていてとても便利。あなたがお探しのソフトも、きっと見つかるはずだ。



freshmeat <http://freshmeat.net/>

待望のKDEオフィススイート

KOffice

バージョン：1.91

ライセンス：GPL

<http://koffice.kde.org>
X KDE Qt

KOfficeは、KDE-2.0で新たに加わったオフィスアプリケーションの総称だ。執筆時点でのバージョンはKOffice-1.91で、KOfficeおよびKDE-2.0は、現在開発途中のためベータ版となっている。また、KOfficeをインストールするには、Qt-2.1.1とkdesupport-1.91がインストールされている必要がある。

KOfficeには、KChart、KSpread、KWord、KPresenter、KIllustratorが含まれている。Linuxにおいては、これまでオフィスアプリケーションが少なく、ビジネスの場面では活用の場が少ないといわれてきた。KOfficeの登場によって、この状況も大きく変わるのではないかと期待されている。執筆時点では、KOfficeならびにKDE-2.0はベータ版であったため、日本語が完全ではなかった。今後、正規リリースまでには、日本語対応も安定してくると思われる。

今回は概要だけを簡単に紹介しておこう。

KChart

グラフ作成ツール。さまざまなデータからグラフを作成でき、豊富なグ

ラフスタイルを選択することが可能だ。

KSpread

表計算ソフト。スプレッドシートを使って、見積り書、請求書など、さまざまな文書を作成できる。

KWord

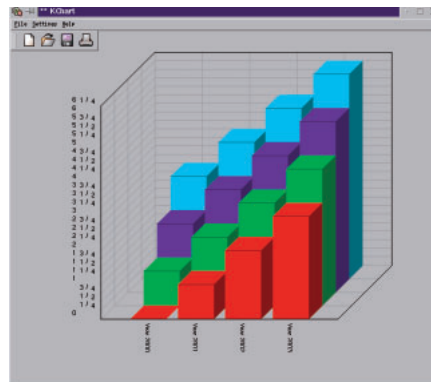
ワードプロセッサ。報告書、稟議書など、さまざまなレイアウトの文書作成が可能だ。

KPresenter

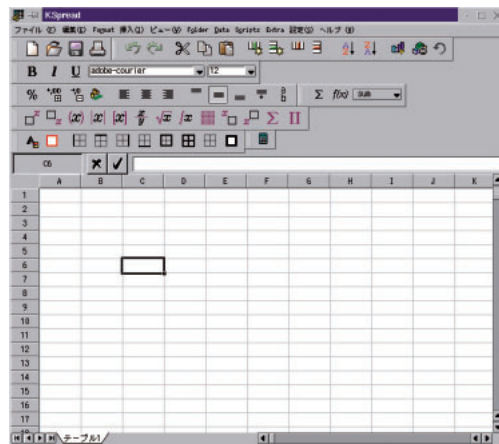
プレゼンテーションツール。プレゼンテーション用の資料作成から、実際のクライアントへのプレゼンテーションまで行える。

KIllustrator

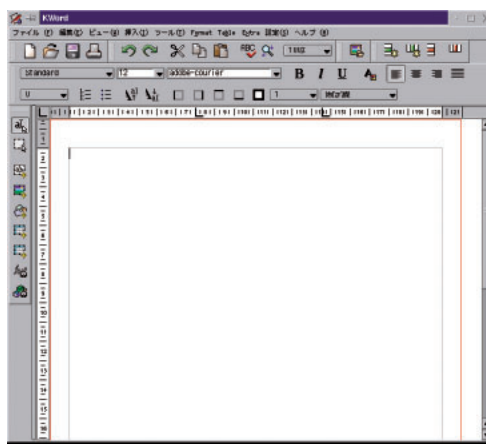
ベクトルグラフィックツール。図、ポスター、イラストが作成できる。



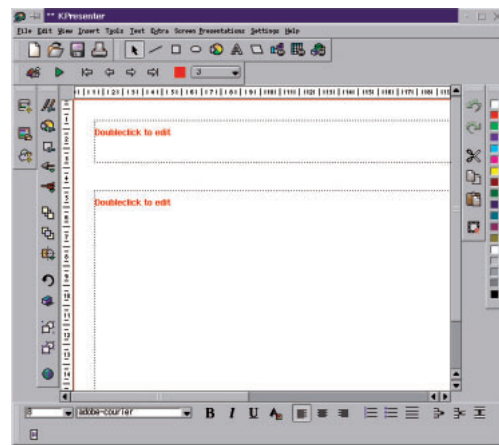
画面1 グラフ作成ツール「KChart」。画面のような立体グラフも作成できる



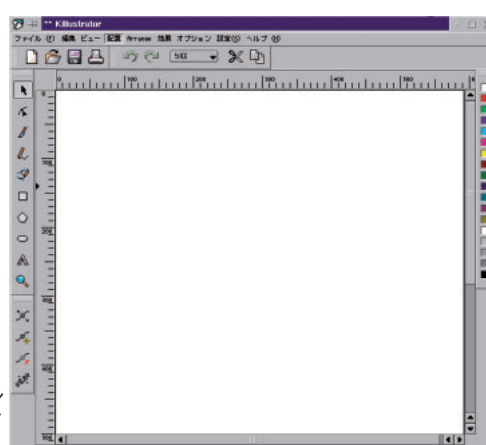
画面2 「KSpread」のメインウィンドウ。基本的な操作や関数の設定を行うためのツールバーを備える



画面3 KSpreadと同様にツールバーを使った基本的な編集機能を備えた「KWord」。



画面4 プレゼンテーションツール「KPresenter」。ベータ版のためメニューが日本語化されていないが、今後対応されるはず



画面5 ドロー系グラフィックツール「KIllustrator」。現時点では独自のファイルフォーマットのみをサポート



高機能な日本語オフィスソフト

USI (JPW/JSIAG/JEGON)

バージョン：1.0.6.5

ライセンス：GPL

<http://www2.tky.3web.ne.jp/metal/vMacj.html>

X

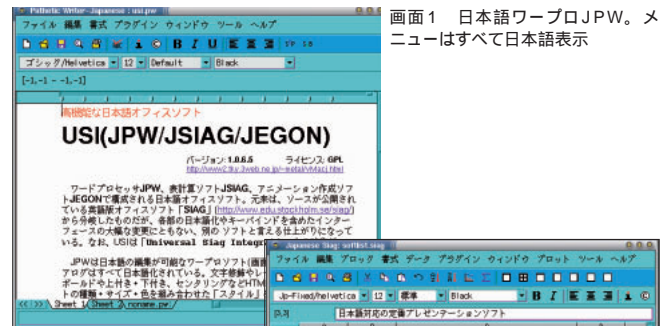
日本語可

ワードプロセッサJPW、表計算ソフトJSIAG、アニメーション作成ソフトJEGONで構成される日本語オフィスソフト。元来は、ソースが公開されている英語版オフィスソフト「SIAG」(<http://www.edu.stockholm.se/siag/>)から分岐したのだが、各部の日本語化やキーバインドを含めたインターフェイスの大幅な変更などにとまない、別のソフトといえる仕上がりになっている。なお、USIは「Universal Siag Integration」の略称だ。

JPWは日本語の編集が可能なワープロソフト(画面1)で、メニューやダイアログはすべて日本語化されている。文字修飾やレイアウトに関しては、ボールドや上付き・下付き、センタリングなどHTML程度のレベル。フォントの種類・サイズ・色を組み合わせた「スタイル」を登録して切り替え可能だ。文書管理はExcel風で、それぞれの文書(バッファ)に複数の「シート」が含まれており、下部のタブで切り替える(分割表示も可能)。ファイル形式は、JPW独自形式のほか、テキストやリッチテキストフォーマット(RTF)、HTMLなどをサポート。印刷はPostscript形式で、gvによるプレビューが可能だ。

一方、JSIAGは日本語が使用できる表計算ソフト(画面2)。バッフ

アやシートの扱いはJPWと同様で、ウィンドウ分割による複数シートの同時表示も可能だ。セルに入力するデータは、ラベル・数値・演算式の3種類で、ラベルを入力する場合はスペースキー、演算式の場合は=キーを最初に押す必要がある(編集時はスペースキーで統一)。セルの位置は、Excel風の「A1」形式でも、行と列を数字で指定する「1,1」形式でも指定できる。ファイル形式は、JSIAG独自形式以外に、CSVやテキスト、HTML、ロータス123などにも対応している。印刷はJPWと同様にPostscript形式で、gvでプレビュー・印刷できる。指定範囲のセルの数値をGnuplotでグラフ化することも可能だ。



画面1 日本語ワープロJPW。メニューはすべて日本語表示



画面2 日本語表計算ソフトJSIAG。セルに日本語を入力可能だ

日本語対応の定番プレゼンテーションソフト

MagicPoint

バージョン：1.07a

ライセンス：GPL

<http://www.mew.org/mgp/>

X

日本語可

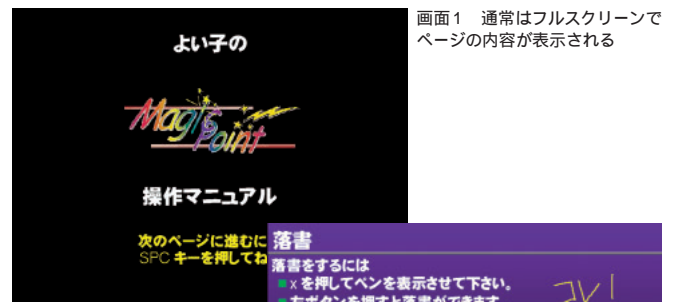
スライドやOHPの代わりにパソコンのディスプレイや液晶プロジェクタでプレゼンテーションを行うソフト。WYSIWYG形式で編集するWindowsのPowerPointとは異なり、テキスト形式のmgpファイルから各ページの画面を生成する。TrueTypeやType1フォントによる高品質な文字に加え、背景にグラデーションを使ったり、JPEG/PNGなど各種画像ファイルを表示することも可能だ。

通常は、各ページの内容がフルスクリーン表示される(画面1)。オプション指定によりウィンドウ表示もできる。画面操作はキーボードとマウスを使用する。スペースキー、キー、マウスの左ボタンなどで次のページに進み、Delキー、キー、マウスの右ボタンで前のページに戻る。「ページ番号」+Gキーで指定したページに直接ジャンプしたり、Ctrlキーで左下に表示されるページ一覧をマウスでクリックしてページを切り替えることも可能だ。

ちょっと変わった機能として「落書きモード」がある。これは、Xキーを押すとマウスポインタがペンの形に変わり、マウスのドラッグ操作でページ上に描画できるというもの(画面2)。プレゼンテーション中に内容を補足したり、アンダーラインを引くのに使えらるだろう。Shift-X

キーで色を変更することもできる。描いた内容は、Lキーを押すか、ページを切り替えると消去される。

mgpファイルはテキスト形式なので、ふだん使っているエディタで作成可能だ。行指向のデータ形式で、先頭が「%」で始まる行はコマンドと解釈される。文字サイズ指定やセンタリングなどのコマンドが豊富に用意されており、複数のコマンドを「,」で区切って1行に並べられる。付属のツールを使えば、画像データを直接ファイルに埋めこんだり、PostScript形式に変換したり、ネットワーク上でプレゼンテーションすることも可能となる。



画面1 通常はフルスクリーンでページの内容が表示される

画面2 ページに自由に書き込みを行える「落書き」機能

フローチャートなどのダイアグラムを描画

Dia

バージョン：0.85/0.84（日本語版）ライセンス：GPL

<http://www.lysator.liu.se/alla/dia/dia.html><http://www2.familie.ne.jp/mituiwa/#dia>（日本語版）

X

GTK+

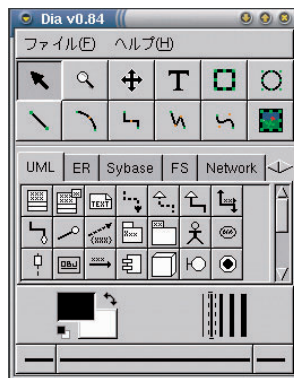
日本語可

フローチャートや回路図、組織図、ネットワーク構成図など、科学・ビジネス分野で用いられる「ダイアグラム」を効率よく描けるソフト。基本オブジェクトのほか、フローチャート、回路図、ネットワーク構成図、データベース、UMLなど分野別のオブジェクトが用意されている。なお、オリジナルのDiaは、日本語カタログが付属するものの、ダイアグラム中では日本語が使えない。このため、ダイアグラム中や印刷時に日本語を利用可能にするパッチが別途配布されている。

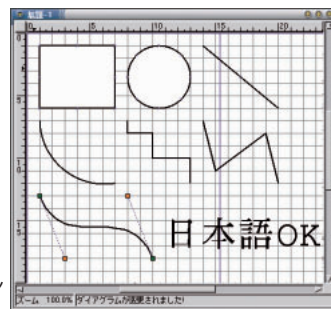
起動すると、さまざまなボタンが並んだメインウィンドウが開く（画面1）。分野別のオブジェクトはタブ付きページで分類されている。これらのボタンを使ってビューウィンドウにオブジェクトを配置する。各オブジェクトには「ハンドル」（緑の点）が表示され、ドラッグにより自由に大きさや長さを変更できる（画面2）。

生成したオブジェクトの属性は、左ボタンのダブルクリックで開くプロパティダイアログや、中ボタンのクリックでポップアップするオブジェクトメニューで変更する。オブジェクトメニューは、ジグザグ線、連続線、ベジエ曲線のように、ハンドルの追加が可能なオブジェクトのみ用意されている。

複数のオブジェクトを選択して位置合わせを行ったり、グループ化してひとつのオブジェクトとして扱うことも可能だ。また、面タイプのオブジェクトに用意されている「接続点」（×印）に、線タイプのオブジェクトを結合すると、オブジェクトを移動しても結合は解けず、自動的に線の長さが増えるようになる。このほか、複数のレイヤにオブジェクトを配置して前後関係を調整することも可能だ。作成したダイアグラムは専用形式で保存できるほか、別形式へのエクスポートも可能。ダイアグラムの印刷はDia上から直接行える。



画面1 メインウィンドウには、描画用のボタンが並んでいる



画面2 7種類の基本オブジェクト。ハンドルで大きさなどを変更可能

定番のドローツール

Tgif

バージョン：4.1pl34

ライセンス：フリー

<http://bourbon.cs.umd.edu:8001/tgif/>

X

日本語可

Tgifは、昔からX Window System上で使われている、Xfigと相壁をなすドローツール。基本のファイル形式は独自のものだが、Postscript、EPS、Xbitmap、GIF、PNGといった一般的な形式での入出力が可能となっている。ドローツールでは比較的珍しい、ページという概念を持ち、このページをつなげて使うこともできる。

ウィンドウ左側のパネルから図形を選び描画し、矢印で図形を選択できる。描画できる図形は、文字、矩形、円、楕円、開いた線、閉じた線、円弧、扇形、角丸矩形、自由曲線だ。上のパネルは選択した図形の属性を変更するためのもので、色や太さ、線のスタイル、字体といったさまざまな属性を変更することができる。図形を選択していないときに、このパネルをクリックすればデフォルトの属性を設定できる。また、表示の拡大縮小、ページスタイルの変更も可能だ。

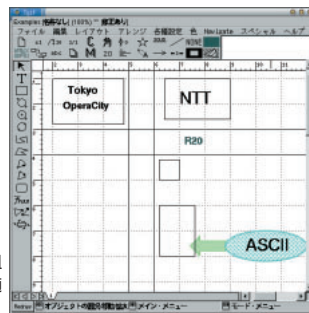
さらに、図形の整列、グループ化、ロックという、描画ツールには必須ともいえる機能もある。図形をパーツにして再利用するといったことも簡単にできるので、複数の同じような図形を使うときには重宝するだろう。ウィンドウの下部には、ページ移動のためのタブのほかに、Windowsのステータスバーに相当する情報が表示される。

ベジエ曲線が利用できないため、グラフィックデザインを行うというような凝った用途にはちょっと厳しいが、簡単な地図を作ったり、図を描いたりといった作業には向いている。スライドショーやハイパーリンク、そして外部コマンド起動機能もあるので、デモやプレゼンテーションといった作業にも使うことができる。

バグフィックスや新しい機能の追加も頻繁に行われているので、安心して使い始めることができるだろう。

なお、Tgifでのファイルの出力は標準のTgif形式以外は、Print（印刷）メニューから行う。つまり、まず上部パネルから出力形式を選んでおき、Printする。もちろん、ここでプリンタのマークを選べば、Postscript形式でプリンタに対して出力されることになる。

実行に必要なファイルは1つで、一次配布元においてはRPMやdebになっているわけではない。必要なライブラリ（glibc2）さえあれば、特にインストール作業をしなくても使うことができる。インターフェイス自体は昔から変わっていないので、最近のGTK+やQt、Tcl/Tkなどを使ったものに比べると見劣りはするが、逆に、個々のライブラリに依存する部分が少ないため、どのような環境でも安心して動作させることができるといえるだろう。見た目のシンプルさとは逆に、かなり高機能な使いのであるツールだ。



画面1 tgifはこのような、簡単な図形を組み合わせた地図などを手軽に作成するのに適している



高性能なドロー系グラフィックツール

Sketch

バージョン : 0.6.7

ライセンス : GPL

<http://sketch.sourceforge.net/>

X Tcl/Tk Python

Sketchは、tgifと同じドロー系のグラフィックツールだ。tgifと比較すると、Sketchは、イラストなどの作成に向いているようだ。

LinuxのグラフィックツールといえばGimpがあまりにも有名だが、ビットマップ系のGimpに対して、Sketchはドロー系といえる。ビットマップ系のグラフィックツールは、データを点で持っているため、拡大/縮小、回転を行うと、画像がくずれてしまう。それに対してドロー系のツールは、データを線/面の形式で保持しているため、拡大縮小を行っても、元の形を維持できる。

SketchはGimpほど有名ではないが、やはりLinux上ではAdobe Illustratorの代わりになるといってもいい、強力なツールだ。趣味でコンピュータイラストを作成するといった用途であれば、十分すぎるほどの機能を備えているといってもよいだろう。

また、操作方法に関しても直感的で、メニューの中をさまようこともない。気がつかないのは、図形を回転させるのに、2回クリックしなければならないことくらいだろう。

Sketchは、市販のドロー系ツールと同様に、ベジェ曲線にも対応している。ベジェ曲線は豊かな曲線の表現に、なくてはならないものであ

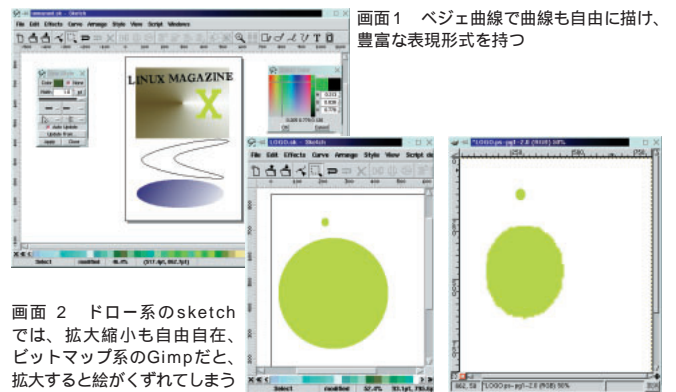
り、イラスト作成には必須の機能といっていられる。

Sketchのデータ形式は独自のものだが、読み込みはAdobe Illustrator、Windows Meta File、Corel CMX、xfig形式などに対応し、Postscript形式とAdobe Illustrator形式などで保存できる。現実の用途には十分だろう。

残念ながら、日本語は扱えないが、日本語の部分のみは他のツールを利用するとしても、使う価値は十分にあるツールだ。

インストールはRPMとdebパッケージ、そしてソースのtarボールが用意されている。ただしTcl/TkやPythonおよびそれに関連するライブラリが必要なため、環境によっては他のパッケージをインストールする必要があるかもしれない。実際に筆者の環境(Kondara)ではpython-imaging-1.0-5.i586.rpmが必要だった。

Paint (ビットマップ) 系の雄であるGimpと、ドロー系のSketchがあれば、Linuxでもグラフィックを自由自在に扱えるだろう。



画面1 ベジェ曲線で曲線も自由に描け、豊富な表現形式を持つ

画面2 ドロー系のsketchでは、拡大縮小も自由自在、ビットマップ系のGimpだと、拡大すると絵がくずれてしまう

散布図などの科学技術系2次元グラフを描画する

SciGraphica

バージョン : 0.3 ライセンス : GPL

<http://magnet.fsu.edu/~feiguin/scigraphica/>

<http://magnet.fsu.edu/~feiguin/gtk/> (GtkExtra)

X GTK+

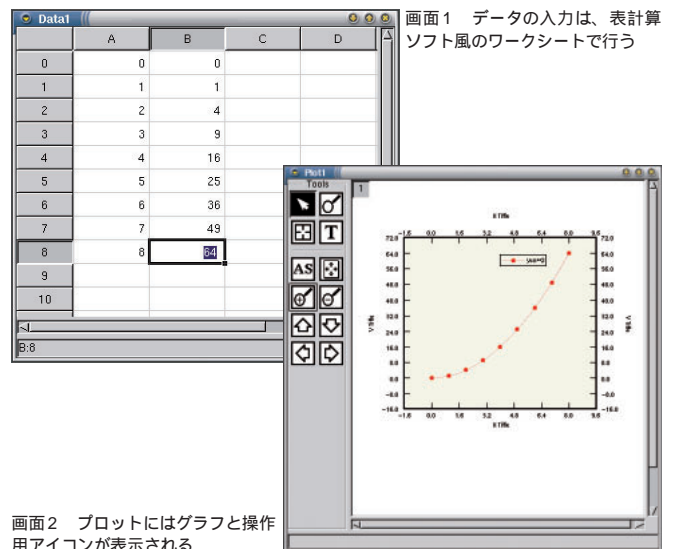
科学技術系2次元グラフ描画ソフト。データの入力を行う「ワークシート」と、グラフを描画する「プロット」で構成され、複数のワークシートとプロットを自由に組み合わせられるのが特徴だ。グラフのスケールやパラメータはGUIで調整でき、複数のレイヤを重ねて表示することもできる。動作には同じ作者によるGTK+用ウィジェットセット「GtkExtra」が別途必要だ。

メインウィンドウでは、データ入力を行うワークシートが一覧表示される [Worksheets] ページと、グラフを描画するプロットが一覧表示される [Plots] ページをタブで切り替えられる。ワークシートは表計算ソフト風のグリッド形式で、グラフの表示に利用する数値データを入力する(画面1)。テキストファイルやほかのワークシートから数値データをインポートすることも可能だ。一方、プロットには操作用のアイコンとグラフが表示される(画面2)。

ワークシートのデータをプロットのグラフに表示するには、「ワークシートのどの列(カラム)をグラフのX、Y軸の値として使うか」を設定する必要がある。この組み合わせを「データセット」と呼ぶ。ひとつのワークシートから複数のデータセットを作成したり、異なるワークシ

ートのデータセットをひとつのプロットで同時に使用することも可能だ。また、データセットごとに、データ間の線の結び方やデータ点の記号、各部の色などを独立して設定できる。さらに、グラフを描画する「レイヤ」を複数作成することにより、複数のグラフを並べたり、スケールの異なるグラフを重ねて表示できる。

ワークシートの内容やプロットの設定を含む全データは、ひとつのプロジェクトとしてXML形式で保存される。また、グラフはPostScript形式で保存して、gvなどによりプレビュー・印刷が可能だ。



画面1 データの入力は、表計算ソフト風のワークシートで行う

画面2 プロットにはグラフと操作用アイコンが表示される

もっと有名なビットマップ系グラフィックツール

Gimp

バージョン : 1.1.18

ライセンス : GPL

<http://www.blender.nl/>

X GTK+ 日本語可

GimpはLinux上で動作するグラフィックツールとしてはもっとも有名なものだろう。そのため、ほとんどのディストリビューションでは標準でインストールされる。そうでなくても、パッケージ自体は用意されていることが多い。

これだけ有名になった理由は、市販ソフトに劣らぬ機能の豊富さのためだ。ビットマップ系グラフィックソフトの最高峰といえるAdobe Photoshopと比較されることが多いが、実際その機能はPhotoshopにも劣らないものだ。

扱うことのできるファイル形式も、BMP、GIF、PNG、TIFF、Postscript、XBM、WMFなど、多岐にわたる。もちろん、レイヤ、フィルタといった機能も備えている。また、マクロ言語であるScript-fuによるロゴ作成の機能などは、それだけでも単独のツールとして十分に通用するほどのものだ。そしてPhotoshopに比べると少なかったプラグインも、最近では充実し始めている。

逆に高機能すぎて、そのすべての機能を使いこなすには、過去にフォトタッチソフトをある程度使いこなしている人でないと、なかなか難しいかもしれない。しかし、Gimpのインターフェイスはフォトレタ

ッチソフトとしては標準的なものだ。有名なソフトのため、Web上のドキュメントや書籍も豊富なので、それらを参照すれば初めてでもそれほど困らないだろう。

さまざまな機能を使っていくと、多数のウィンドウが開くので操作にとまどうかもしれないが、画像ウィンドウの上で右クリックをすれば、いつでもメニューが開くので、何かするときは「絵の上で右クリック」と覚えておこう。

現在の安定版は1.0.4だが、より多くの機能を備えた開発版の1.1.18も実際には十分に安定しているため、こちらを常用してもいいだろう。



画面1 スタートアップ画面には、BUGSとあるが、バグだらけというわけではない



画面2 ツールパレットもバージョンアップとともに変わっていく

画面3 文字を打ち込むだけでScript-fuで簡単にロゴを作ることができる



高機能3Dモデリング/アニメーションツール

Blender

バージョン : 1.71

ライセンス : フリーウェア

<http://www.blender.nl/>

X Mesa

Blenderは、Not a Number製の高性能な3Dモデリング/アニメーションツールだ。フリーではあるが、市販のモデリングツールに遜色のない、個人で使うには十分すぎるほどの機能を持っている。3Dモデリングに興味があるのならば、まず、このフリーのツールを使っている人試してみるのもよいだろう。もちろん、VRML、DXFといった汎用的な形式での出力をサポートしているので、このツールで作ったデータは、他のツールで活用することも可能だ。

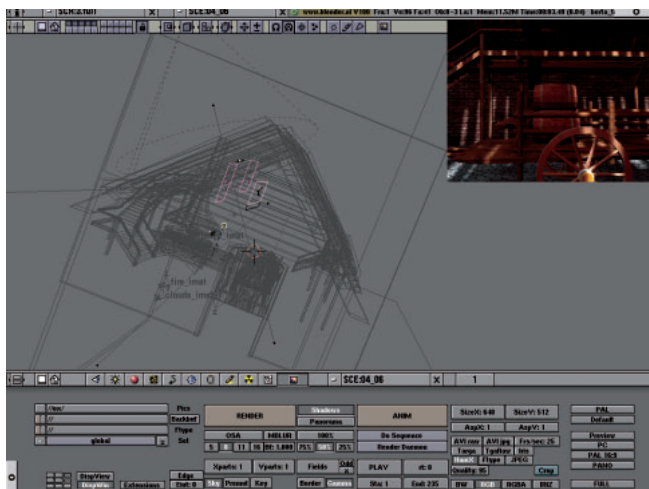
起動後、マウスの左ボタンで位置決定、中ボタンを押してドラッグで視野回転、SHIFTキーを押しながらの中ボタンドラッグで視野の平行移動、CTRLキーと中ボタンドラッグで視野のズームイン、ズームアウトといった操作が行える。

メニューはSHIFT-Aで出すことができる。後はADDやEDITで物体やライトといったオブジェクトを加えたり、編集してみよう。F12ボタンでレンダリングされる。レンダリングウィンドウを消すにはF11ボタンだ。あとはいろいろといじって試してみたい。

マニュアルはWeb上はかなりできのよいものが(英語ではあるが)公開されているので、紙のマニュアルがなくても困ることはないだろう。

英語が苦手ならば、日本のBlenderユーザー有志によるバージョン1.5用の日本語チュートリアルマニュアルを参考にさせてもらおう(http://www.dims.or.jp/blender/stuff/bl_manjp.html)。

インストールはtarボールのバイナリを展開するだけだ。Mesaライブラリを内部に持ったStatic版と、Mesaライブラリを別途必要とするDynamic版がある。



画面1 Blenderの機能を使いこなせば、こんな画面も作れる



入門者向け3Dモデリングソフト

Giram

バージョン: 0.1.3

ライセンス: フリー

<http://www.minet.net/giram/>

X

GTK+

3DレンダリングソフトPOV-Rayを利用する3Dモデリングソフトだ。高機能とはいえないが、そのぶん一般のモデリングソフトにありがちな難解さはなく、3Dモデリング入門者にもわかりやすいインターフェイスになっている。メニューやドキュメントは英語だが、操作方法で悩むことはあまりないだろう。

自前の簡単なレンダリングエンジンもあるが、フリーの3Dモデラーとしてはもっとも有名な、POV-Rayのフロントエンドとなるツール、povfrontも持っており、POV-Rayを呼び出してレンダリングを行うことができる。

対応しているファイル形式は、DXF形式とPOV形式の2種類だ。

XYZ軸の各方向から見たウィンドウと、設定したカメラの位置から見たウィンドウの4つのウィンドウを表示することができ、見た感じは一般の3D CADに近い。カメラ位置からのウィンドウは、通常のワイヤフレーム、見えない部分の線を表示しないワイヤフレーム、簡易レンダリング、POV-Rayレンダリングの4通りで表示できる。

起動すると、簡単なメニューウィンドウが表示される。ここで、既存のファイルを読み込むなら、「LOAD POV」か「LOAD Auto CAD DXF」

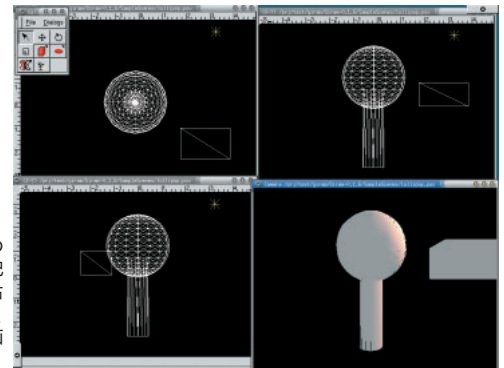
を選んでファイルを読み込む。

新規のファイルを作成するなら、そのまま「File」メニューの「New View」から、各方向のウィンドウを開くことができる。あとはメニューウィンドウの中から図形を選び、各ウィンドウに配置、右クリックで選択して移動していく。描画する図形はメニューウィンドウ上で右クリックすると、切り替えることもできる。

また各方向の表示ウィンドウで右クリックすると、メインメニューにはない、色の選択、回転、大きさの変更や、レンダリングなどを行うことができる。

tarボールによる配布のみなので、自分で展開、ビルドして、インストールする必要がある。

まだ未完成な部分が多いとドキュメントにもあるが、動作そのものが若干不安定なこともあった。ただし、開発は着々と進んでいるようで、将来が楽しみなソフトである。



画面1 通常、このようなウィンドウ配置で作業を行う。右下のウィンドウは、カメラ位置からの画像

数多くのフォーマットに対応した画像ツール

ImageMagick

バージョン: 5.2.2

種別: フリー

<http://www.wizards.dupont.com/cristy/ImageMagick.html>

X

Tcl/Tk

Python

ImageMagickは数多くの画像フォーマットに対応した、画像処理ツールとライブラリ集だ。対応するフォーマットは、よく使われるJPEG、GIF、PNG、TIFF、PDFをはじめに、なんと68種類以上にも及び、

コマンドラインから使うツールには、以下のようなものがある。

convert 画像フォーマット変換

mogrify 拡大縮小、回転など

montage 複数の画像を並べる

combine 複数の画像の合成

identify 画像情報の取得

import 画面のスクリーンショットを撮る

X上で動作するdisplayコマンドでは、拡大縮小、回転などの単純なものから、画像フォーマットの変換、減色増色やシャープネスなどの特殊効果も行うことができる。またCやperlから使うことのできるライブラリも用意されている。

convertコマンドによるGUIの画像処理はGimpほどの機能はないが、

ImageMagickの一番の真価は、コマンドラインツールとライブラリにあるといえるだろう。特に、複数の画像ファイルに対して一定の処理をしたりするときには、シェルスクリプトと組み合わせるなどして、強力な味方となってくれるはずだ。

新フォーマットへの対応、新機能の追加、そして効率化など、常にバージョンアップを続けているのは心強い。

ImageMagickもかなりメジャーなツールなので、ほとんどのディストリビューションにもパッケージは付属している。また、Webサイトでの配布はtarボールによるソースだが、特に問題なくビルドできるだろう。



画面1 JPEG、PNG、BMPなどポピュラーなもの以外にも、非常に多くの画像フォーマットに対応している



画面2 Gimpほどではない簡単な画像処理なら、ImageMagickでも手軽にできる

Vzライクな特徴を持つ高速テキストエディタ

ne

バージョン : 3.05

ライセンス : GPL

<http://hp.vector.co.jp/authors/VA018758/ne/index.html>

コンソール

日本語可

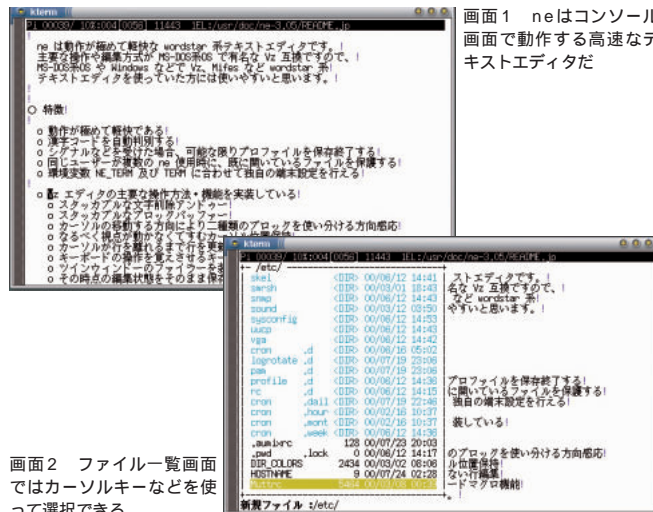
コンソール画面で軽快に動作する日本語対応のテキストエディタ。MS-DOS用の高速テキストエディタとして一世を風靡したVzエディタによく似た機能やインターフェイスが特徴だ。VzやMIFESのようにモードレスなWordstar系エディタを愛用していた人には、viやEmacsよりもはるかに使いやすいだろう。

「ne ファイル名」として起動すると、コンソール画面にそのファイルの内容が表示される(画面1)。なお、各行の末尾にある青い「!」は改行コードを示している(表示しない設定も可能)。文字コードがJISやシフトJISのファイルは、読み込む際に日本語EUCに自動変換され、保存時に元の文字コードに戻される。行番号の表示やフリーカーソル、オートインデントなどの設定は、実行中にF4キーで変更できる。

初期設定では、Vzライクなキー割り当てが行われている。最初からファンクションキーやカーソルキーが使えるので、PC UNIXの初心者にも使いやすいだろう。たとえば、ファイル関係の操作はF1キーのメニューから選択でき、ファイル一覧画面(画面2)ではカーソルキーとEnterキーを使って読み込むファイルを選択できる。一方、常にホームポジションに手を置いておきたい中・上級者は、Ctrl-QやCtrl-K、

Esc (Ctrl-[) キーで始まる2ストロークキーを利用しよう。たとえば、ファイルのオープンやセーブは、Esc (Ctrl-[) キーに続けてOキー(オープン)やSキー(セーブ)を押すことで実行できる。こうしたキー割り当ては設定ファイルの書き換えで変更可能だ。

ところで、neではVzと同様にスタックابلなカット(コピー)&ペースト用ブロックバッファを採用している。つまり、カットを複数行してからペーストすると、最後にカットした内容から順番にペーストされる。また、文字単位の削除についてもスタックابلなバッファが用意されており、最後に削除した文字から順にCtrl-Uキーでペーストできる。



画面1 neはコンソール画面で動作する高速なテキストエディタ

画面2 ファイル一覧画面ではカーソルキーなどを使って選択できる

日本語に対応した使いやすいHTMLエディタ

Bluefish

バージョン : 0.4 ライセンス : GPL

<http://bluefish.openoffice.nl/><http://www.weblint.org/> (Weblint)<http://www.sfc.keio.ac.jp/~mimasa/jweblint/index.html.en> (jweblint)

X

GTK+

日本語可

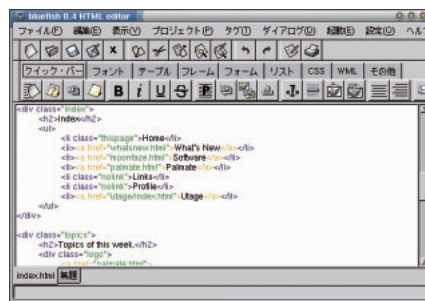
カラー構文表示に対応したタグ挿入型のHTMLエディタ。ツールバーのボタンを使ってタグを効率よく入力できる。日本語カタログによるメニューの日本語化や、日本語の入力・編集にも対応している(日本語EUCのみ)。また、Weblintを利用したWebページの文法チェックや、Netscapeによる実際の表示の確認が可能だ。

起動すると、2段構成のツールバーを持つウィンドウが開く(画面1)。使い方は一般的なタグ挿入型のHTMLエディタと同じで、ツールバーのボタンを押してカーソル位置にタグを挿入し、要素をキーボードから入力する。もちろん、タグ自体をキーボードから入力してもいい。タグ入力用のツールバーはジャンル別にページ分けされており、目的のタグを素早く見つけられるようになっている。<A>やのように属性の指定が必須のタグでは、それぞれ専用のダイアログが開いて属性値を設定する(画面2)。スタイルシートにも対応しており、各種のスタイル設定やタグへのクラス追加がボタン操作のみで可能だ。

タグ、属性、属性値などをカラー表示する色分け(カラー構文表示)機能は、初期設定ではF5キーを押さないと表示が更新されないが、編集行やウィジェット全体を自動更新する設定も可能だ。ただし、現時

点ではマルチバイト文字の色分け処理に不具合があり、日本語を含むHTMLファイルではタグの色分けをオフにする必要がある。

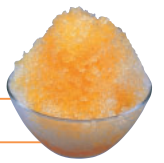
HTMLファイルの構文チェックを行うには、設定ダイアログの[起動]ページの[Weblintコマンド]を、Weblintの日本語対応版であるjweblintのフルパスに変更する。Bluefishからjweblintを起動すると、HTMLファイルのチェック結果が別ウィンドウに表示され、結果をダブルクリックすると編集領域の対応行が反転表示される。



画面1 タグ挿入用のボタンはジャンル別に分類されている



画面2 の設定ダイアログでは画像をプレビューできる



日本語をカラー構文表示できるHTMLエディタ

Web Designer

バージョン : 0.1.6 ライセンス : GPL

<http://webdesigner.linuxave.net/>

X GTK+ GNOME 日本語可

カラー構文表示を持つタグ挿入型のHTMLエディタ。ツールバーのボタンを使ってタグを効率よく入力でき、外部ブラウザで実際の表示を確認できる。編集時のタグの文法が表示されるヘルプバーや、複数のファイルをまとめて扱うプロジェクト機能も用意されている。

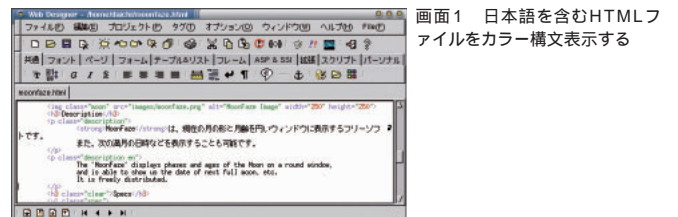
日本語カタログ (ja.po) は標準で付属しているが、日本語EUCでテキストを入力・編集するには、srcディレクトリのwebdesigner.cの“gtk_init (...);”の前に“gtk_set_locale ();”を挿入する必要がある。また、poディレクトリのja.gmoをビルドする前に削除しておかないと、日本語メニューの表示がおかしくなる。

起動すると、Tipsウィンドウとともに、数多くのボタンが並んだツールバーを持つウィンドウが開く (画面1)。使い方は一般的なタグ挿入型のHTMLエディタと同じで、ツールバーのボタンを押したり、[Tags]メニュー以下の項目を選択すると、対応するタグがカーソル位置に挿入される。タグ挿入用のボタンはツールバーの2段目にあり、タブ付きページでジャンル分けされている。

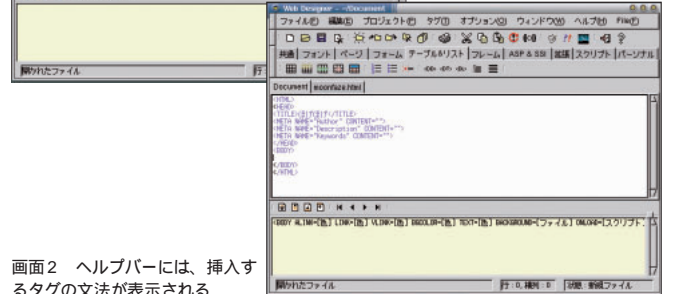
ファイル編集用のペインでは、HTMLのタグやC言語の予約語などがカラー表示される。複数のファイルを同時に編集でき、上部のタブで切

り替えられるほか、複数のウィンドウを開くことも可能だ。なお、先に紹介したBluefishと違って日本語を含むHTMLファイルもカラー表示できるが、カラー表示が更新されるのはファイル読み込み時などに限られる。

編集用ペインの下には「ヘルプバー」が用意され、挿入したタグの意味や利用できる属性などの文法が日本語で表示される (画面2)。ヘルプバーが必要なければ、非表示にすることも可能だ。作成したHTMLファイルをいったんセーブすれば、外部ブラウザ (初期設定はNetscape) により実際のページの見栄えを確認できる。



画面1 日本語を含むHTMLファイルをカラー構文表示する



画面2 ヘルプバーには、挿入するタグの文法が表示される

プログラム開発に適したテキストエディタ

CodeCommander

バージョン : 0.9.6 ライセンス : GPL

<http://codecommander.sourceforge.net/>

X GNOME GTK+

CodeCommanderは、GTKを利用したプログラマー向けのマルチファイルエディタだ。

検索、置換、行ジャンプ、undo、redoなどの基本的なエディタの機能はもちろん完備している。さらに、使用するプログラミング言語に合わせて、構文強調のためのフォントおよび表示色の変更や、自動インデントも行ってくれるので、コーディングの際の見通しもよくなる。

もちろん、「Build」メニューやアイコンから、コンパイル、デバッグ、実行などを行うこともできる。これらは自動的にターミナルが開き実行される。

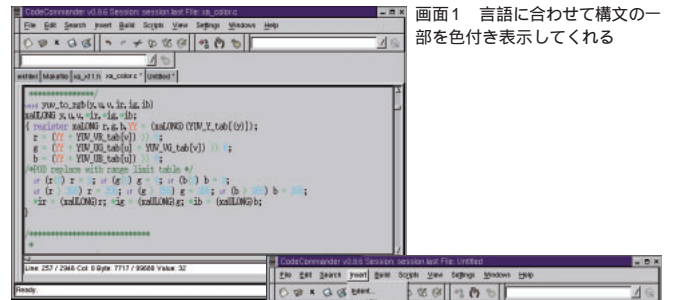
対応している言語は、C、C++、Perl、bash、HTML、Java、PHPなどと、かなり広範囲にわたっている。

現在編集しているファイル群をセッションとして保存する機能もあるので、複数のファイルを同時に、ある程度の期間編集を続けるときにはうれしいだろう。こういった機能はEmacsにもあるが、Emacsに比べれば動作の雰囲気はWindowsのエディタに近く、UNIX系OSよりもWindowsに慣れたユーザーなら、試してみる価値があるエディタだ。

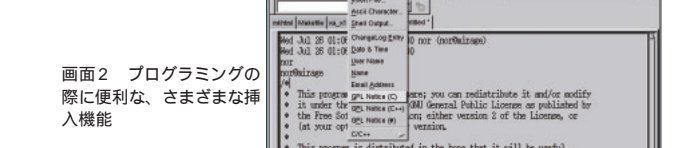
そのほかGPLの注意書などを挿入する機能や、単語を選択してman

ページを検索する機能など、プログラマーのための機能が多数用意されている。

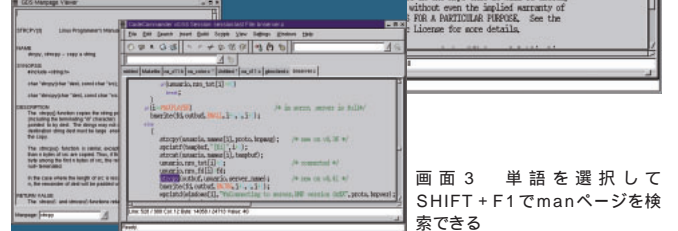
RPMパッケージとソースファイルのtarボールが用意されている。残念ながら、現在のバージョンでは日本語が使えない。



画面1 言語に合わせて構文の一部を色付き表示してくれる



画面2 プログラミングの際に便利な、さまざまな挿入機能



画面3 単語を選択してSHIFT+F1でmanページを検索できる

KDEアプリケーションの統合開発環境

KDevelop

バージョン: 1.2

ライセンス: GPL

<http://www.kdevelop.org/>

X

KDE

Qt

Kdevelopは、KDE上で動作する総合開発環境アプリケーションだ。Qtアプリケーション、KDEアプリケーションを簡単に、そして効率よく開発するためのツールとして、大変いい仕上がりとなっている。デバッグツールである「KDbg」、poファイルの翻訳とメッセージカタログの作成を行うためのツール「ktranslator」と組み合わせることにより、よりいっそう効率のよい開発が可能となる。プロジェクト管理機能やウィザードによるアプリケーションのサポート、インターフェイス部品（ウィジェット）をデザインするための専用エディタなどプログラム開発環境としての基本的なものはすべて備えている。リファレンスの検索が簡単にできるのも特徴のひとつだ。

KDevelopを起動すると、インストールして最初の起動の場合、セットアップの画面が表示される。画面の指示に従って、Qtドキュメントのパス、KDELIBSドキュメントのパスを指定しよう。セットアップが正しく行われるとKDevelopが起動する。

新規プロジェクトの作成

[プロジェクト]メニューから[新規]を選択すると「アプリケーション

ウィザード」が起動する。画面左側のツリーから作成したいプログラムの種類を選択し、[次へ]ボタンを押して次の画面に移る。必要な事項を入力し、[作成]ボタンを押すとプログラムの雛型が作成される。

ウィジェットの作成

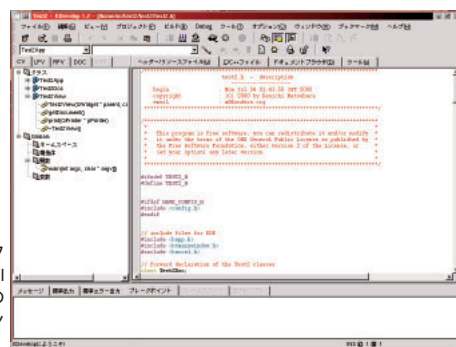
[ビュー]メニューから[ダイアログエディタ]を選択すると、ダイアログエディタの画面が表示される。プレビュー画面を使って快適にウィジェットをデザインをすることが可能だ。

プログラムのビルド

[ビルド]メニューから[Make]あるいは[全てをリビルド]を選択する

プログラムのデバッグ

KDbgがインストールされていれば、[Debug]メニューからシームレスにデバッグ作業を行える。



画面1 ユーザーインターフェイスはVisual C++などWindowsのものによく似たデザインだ。

日本語対応の軽量Webブラウザ

MMM (も~)

バージョン: 0.416 ライセンス: フリー

<http://pauillac.inria.fr/furuse/jmmm/>
<ftp://ftp.kurims.kyoto-u.ac.jp/pub/lang/caml-light/> (Objective Caml)

X

Tcl/Tk

日本語可

Netscapeより軽量なWebブラウザ。オブジェクト指向言語の「Objective Caml」と、そのTcl/Tkインターフェイスである「CamlTk4」ライブラリを使って作られている。Objective Camlで書かれたアプレットによる機能追加も可能だ。日本語化されたTcl/Tkと組み合わせれば、日本語の表示（JIS / シフトJIS / 日本語EUCに対応）や、kinput2を利用した入力も問題なく行える。

最初に、Objective Camlをインストールし、CamlTk4ライブラリ、MMMの順でObjective Camlによるコンパイルを行う（詳しい手順は、Webサイトの「インストール」ページに書かれている）。なお、Objective Camlにはバイトコードを生成する「ocamlc」とネイティブコードを生成する「ocamlopt」の2種類のコンパイラがある。どちらを利用して構わないが、CamlTk4とMMMでコンパイラを統一する必要があるようだ。なお、アプレットはネイティブ版では利用できない。

“/usr/local/lib/mmm/mmm.bin”（ネイティブ版はmmm.x.bin）として起動すると、ボタンやタコメータ（回転計）を持ったウィンドウが開いて、MMMに内蔵された著作権ページが表示される（画面1）。使い方はNetscapeなどと同じで、青いアンダーラインのリンクをクリックした

り、[Back]ボタンや[Forward]ボタンを押したり、[Open URL]に直接URLをキー入力することでWebページを閲覧できる。

サポートしているHTMLのバージョンは2.0と3.2で、テーブル、フレーム、フォーム、一部のCGIにも対応している。JPEG形式やGIF形式（アニメーションGIFを含む）の画像にも対応しているが、初期設定では画像はページを読み込んだだけでは表示されず、メニューの[Document] - [Load Images]を選択するか、Alt-iキーを押した時点で読み込まれる。面倒なら、ページの読み込みと同時に画像を読み込むように設定することも可能だ（画面2）。



画面1 MMMはHTML 3.2に対応した軽量Webブラウザだ

画面2 設定を変更すれば画像を自動的に読み込むことも可能



テキストベースの軽快なWebブラウザ

w3m

バージョン: 0.1.10

ライセンス: フリー

<http://ei5nazha.yz.yamagata-u.ac.jp/aito/w3m/eng/>

コンソール

w3mはテキストベースすなわち、ターミナル上で使うWebブラウザだ。同様のソフトとしてはLynxがあるが、Lynxにない特徴としては、

- ・ tableの表示
- ・ 軽快な動作と、サイズの小ささ

などがある。

w3mを使ってみるとその動作の軽さに驚くだろう。特に文字中心のサイトを見る場合には、一度使うと手放せなくなるほどの軽さだ。

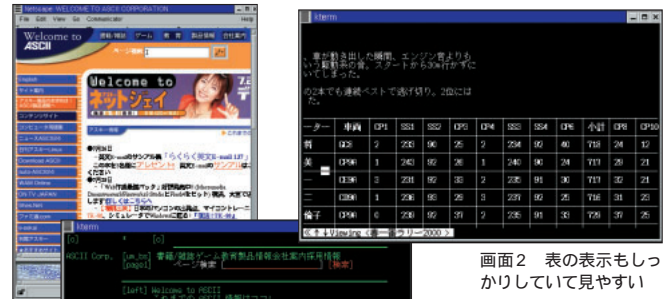
また、カーソルキーやマウスに対応しており画面内を自由に移動できるので、ターミナルからはみ出してしまふ部分も非常に読みやすい。キー割り当ての変更など、各ユーザーの使い勝手を重視した設計もうれしい。操作がわからなくなったときには、Insertキーを押せば簡単なヘルプが表示されるので、戸惑うこともないだろう。

また、ほかのブラウザにはない機能として、Webサーバを使わずにCGIスクリプトを起動する、Local CGIという機能がある。これは、w3m自体がCGIスクリプトを起動して、その出力を表示するというものだ。自作のCGIスクリプトを、実際にWebサーバに置く前にテストするときなどにも、有用な機能だ。

インストールは、tarボールで配布されるソースをビルドすることになる。多くの場合と異なり、./configureを実行するといくつか質問が表示される。英語の質問だが、それほど難しいものではないので、あまり悩むようなことはないだろう。

今後の開発予定だが、余計な機能を入れて動作を重くしたくないため、バグフィックスのみだそう。昨今の高機能を追求し肥大化するソフトとは逆を行くが、機能ばかり増えれば使いやすいというものではない。こういったソフトこそ、気楽かつ快適に使えるツールとして、愛用したいものだ。

なお、作者によるとw3mはページだということだ。オプションを何も指定しないで起動するとそのまま終了したり、キャッシュ機能を持っていないのはそのためだ、としている。



画面2 表の表示もしっかりしていて見やすい

画面1 アスキーのホームページを Netscape Navigatorとw3mで表示してみた。w3mでも十分実用になることがわかる

安定して使いやすいメールクライアント

Sylpheed

バージョン: 0.3.21

ライセンス: GPL

<http://sylpheed.good-day.net/>

X GTK+ 日本語

日本製の軽快で使いやすいメールクライアント兼ニュースリーダー。Windows系メールクライアントと同様の外観と、MewやWanderlustといったEmacs系メールクライアントの操作感をあわせ持つ。複数アカウントへの対応をはじめ、メール一覧のスレッド表示、受信メールの振り分け、ドラッグ&ドロップによるファイルの添付、クリックブルURLなど機能も充実してきた。

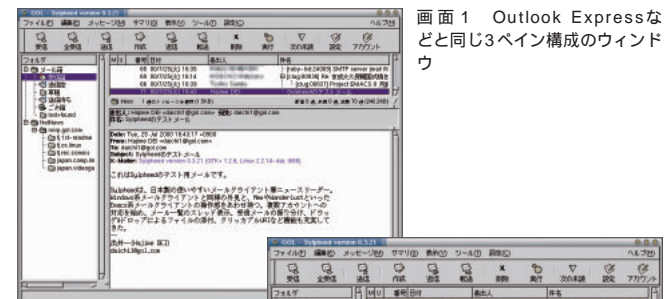
メインウィンドウの構成はOutlook ExploressなどのWindows系メールクライアントでおなじみの3ペイン方式だ(画面1)。設定によっては、フォルダツリーやメッセージビューを独立したウィンドウとして表示することもできる。

Sylpheedでは、1人のユーザーが複数のアカウントを切り替えて使用できる(現在のアカウント名はウィンドウの右下に表示)。メールの受信では、現在のアカウントのメールを受信するだけでなく、登録したすべてのアカウントのメールを一度に受信したり、ローカルなメールボックスの内容をインポートすることも可能だ。

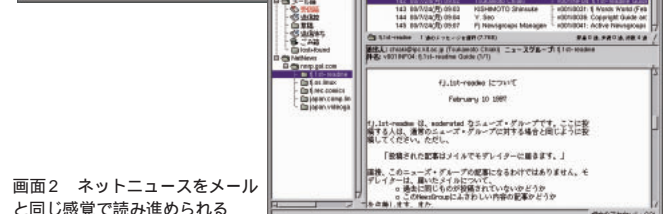
受信したメールは[受信箱]フォルダに格納され、タイトルや日付、送信者などがサマリビューにスレッド表示される。未読メールを順番に

閲覧するにはスペースキーを押すだけでいい。次の未読メールへの切り替え、画面に収まりきらない本文のスクロール、次のフォルダへの切り替えなどを自動的に処理してくれる。もちろん、大量のメールを受け取る人には必須の振り分け機能も用意されている。

このほか、ネットニュースを購読するニュースリーダとしての機能も備えている(記事を投稿する機能は用意されていない)。利用するニュースサーバや購読するニュースグループの名前を登録すれば、ニュースグループがフォルダとして表示され、メールを読むのと同じ感覚で閲覧できる(画面2)。



画面1 Outlook Expressなどと同じ3ペイン構成のウィンドウ



画面2 ネットニュースをメールと同じ感覚で読み進められる

日本製の高性能メールクライアント / メールチェッカー

The N-Tool (Nmail4)

バージョン: 1.2

ライセンス: GPL

<http://nmail.hiug.ne.jp/>

X

GTK+

GNOME

日本語可

メールクライアント「Nmail4」を中心とした日本製のメール関連ソフト集。見た目はWindows系メールクライアントそのもので、受信したメールの振り分けや、アドレス帳によるアドレス管理も可能だ。さらに、複数のメールアドレスを切り替えられたり、送信メールをいったん保存して、あとでまとめて送信できるなど、PPPでインターネットを利用するユーザーに適した機能を持つ。

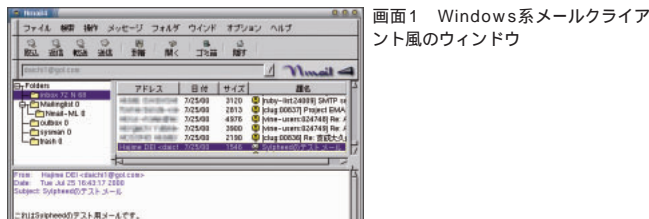
初めて起動したときは、自動的に設定ツール（nsetup4）に切り替わり、メールアドレスやサーバ名などの設定を行う。パスワードは、入力後にEnterキーを押し、再入力後にもう一度Enterキーを押すことではじめて保存されるので注意してほしい。

Nmail4のウィンドウ構成はWindows系メールクライアントで一般的な3ペイン方式（画面1）だが、メールの内容を独立したウィンドウに表示することも可能だ。ユーザーによるドラッグ&ドロップや、振り分け機能によりメールをフォルダに分類できる。なお、初期設定ではメールのヘッダをすべて表示する設定になっているので、簡略化表示（From / Date / Subjectのみ）に切り替えるといいだろう。

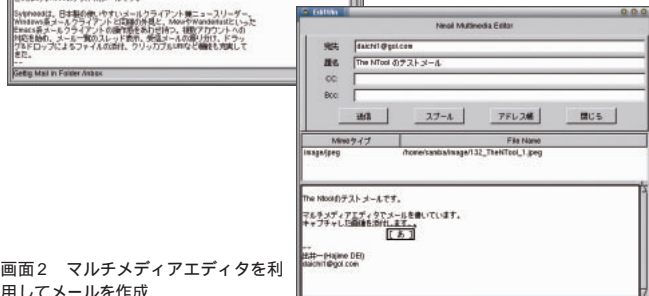
メールの作成は「マルチメディアエディタ」と呼ばれるメール送信ウ

ィンドウで行い、文章以外にも簡単な操作で画像などのファイルを添付できる（画面2）。設定によっては、Emacsなどの外部エディタを利用することも可能だ。作成後のメールはいったんスプールに保管され、あとでまとめて送信される。

The N-Toolには、Nmail4以外にもいくつか関連ソフトが含まれている。そのうちのひとつを紹介しよう。popcheckerは10分ごと（変更可能）にPOPサーバに接続して新着メールの有無を調べるメールチェッカーだ。Nmail4の設定を参照して動作するため、POPサーバ名などを設定する必要はない。



画面1 Windows系メールクライアント風のウィンドウ



画面2 マルチメディアエディタを利用してメールを作成

Emacs上で動作する先進的なメールクライアント

Wanderlust

バージョン: 1.1.1/2.2.18 () ライセンス: GPL

<http://www.gohome.org/wl/>

Emacs

日本語可

Emacs / XEmacs / Mule上で動作する先進的なメッセージ管理ソフト。POP3 / IMAP4やMH形式対応のメールクライアントとして、あるいはNNTP対応のニュースリーダとして、日本語を含むメッセージを统一的に扱える。XEmacsではツールバーやフォルダアイコンなどがグラフィックで表示され、マウスによる操作も可能だ。なお、Emacs-lispのみの実装で、MHなどの外部ソフトは不要だが、MIMEモジュール（SEMIまたはtm）をあらかじめインストールしておく必要がある。

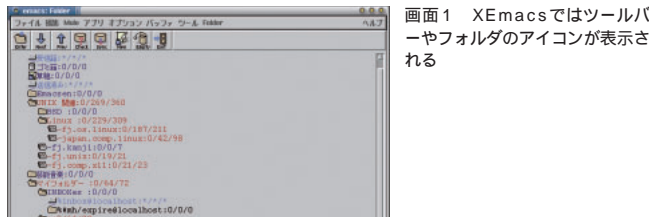
Emacs上で「M-x wl」としてWanderlustを起動すると、美しいロゴが表示されたあと、「フォルダモード」に移行し、メールやニュースなどを格納するフォルダがツリー状に並ぶ（画面1）。POP3 / IMAP4のメールスプールもフォルダとして扱われるので、いちいちローカルなフォルダ（+inboxなど）にメールを取りこむ必要はない。ユーザーがフォルダ構成を変更したり、フォルダを追加することも可能だ。

フォルダ上でスペースキーを押すと「サマリーモード」に移行し、フォルダ内のメッセージがスレッド順に表示される（画面2）。未読メッセージを順番に読んでいくには、スペースキーを押すだけでいい。マルチパートMIMEによる添付ファイルや、メッセージの振り分け（リファ

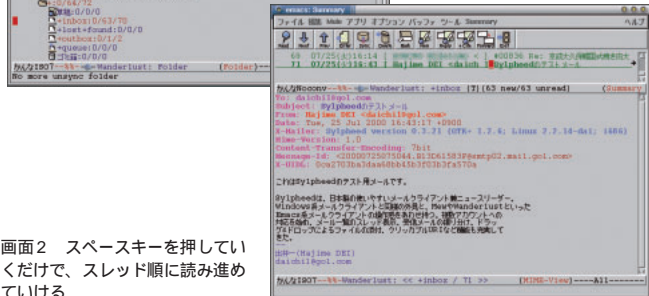
イル）にも対応している。

メッセージの作成時には、Emacsの強力な編集機能をそのまま利用できる。マルチパートMIMEによるファイルの添付も可能だ。なお、オフライン状態で作成したメッセージはキューに保管され、オンラインになった時点で一括送信される。

このほか、ZIP、LHAなどのアーカイブを通常のフォルダのように扱うアーカイブフォルダ、古くなったメッセージを自動的に削除（あるいはアーカイブフォルダに移動）するエクスパイア、SPAMの排除に役立つスコアリングなどの便利な機能が用意されている。



画面1 XEmacsではツールバーやフォルダのアイコンが表示される



画面2 スペースキーを押すだけで、スレッド順に読み進めていく



日本語対応のGNOME版ICQクライアント

GnomeICU

バージョン : 0.93/0.94 devel ライセンス : GPL

<http://gnomeicu.gdev.net/>
<http://northeye.org/gnomeicu-ja/> (日本語サイト)

X GTK+ 日本語可

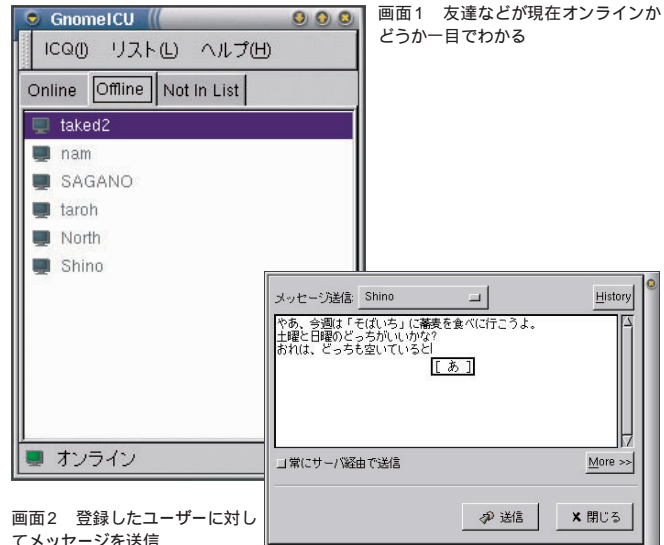
ユーザー数が数千万人を超えるコミュニケーションソフト「ICQ」互換のクライアントソフト。メッセージ交換やチャット、ファイル転送など、ICQの主要な機能をほぼカバーしている。ふだんはGNOMEのパネルに常駐しており、クリックするとウィンドウが開く。なお、これまではメニュー表示やメッセージを日本語対応にするパッチが別途用意されていたが、次バージョンから本体に取り込まれることになった。

初めて起動した場合は、ユーザー識別用のICQ番号の入力を求められる。これまでにICQを使ったことのない人は、ここでICQユーザーとしての新規登録を行う。WindowsなどでICQを使っている人は、自分のICQ番号を調べておいて入力すればいい。どちらの場合も、ほかのユーザーに公開する情報（名前、ニックネーム、メールアドレスなど）を、差し支えない範囲で設定しておこう。

GnomeICUのウィンドウを開いたら、まずはICQを使っている友人などをコンタクトリストに登録しよう。ICQ番号を直接入力するほか、名前やメールアドレスによる検索も可能だ。相手側の設定によっては、登録時に認証が必要な場合もある。登録が完了すると、そのユーザーがオンライン（ICQ使用中）の場合は [Online] ページ、そうでなければ

[Offline] ページにニックネームが表示される（画面1）。

コンタクトリストに登録したユーザーとメッセージの交換を行うには、相手のニックネームを右クリックして [メッセージの送信] を選択し、メッセージをダイアログに入力して [送信] ボタンを押せばいい。逆に、メッセージを受信すると、そのユーザーのアイコンが変化し、ダブルクリックするとメッセージの内容が表示される。チャットやファイル転送も同様に右クリックメニューから選択するが、相手がオンラインの場合にしか利用できない。



画面1 友達などが現在オンラインかどうか一目でわかる

画面2 登録したユーザーに対してメッセージを送信

KDEのICQチャットクライアント

KXicq

バージョン : 0.4 ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

KDEのICQクライアント。ICQとは、メッセージ、URL、データなどを交換したり、ときにはチャット機能も提供するコミュニケーションサービス。KXicqは日本語でのチャット機能は対応していない。ただし、Kondara MNU/Linuxのパッケージについては日本語対応してある。Kondara以外のRed Hat系ディストリビューションを使っている場合は、specファイルの “ %define kkdir /usr/kde1x ”、 “ %define qtdir /usr/lib/qt-1.44 ” の部分を環境に合わせて編集してからリビルドしよう。その他のディストリビューションの場合は、kxicq1x-0.4-3k3.nosrc.rpm に含まれるパッチをすべてあててからリビルドしてほしい。

ほかのICQユーザーと通信するには、まず、自分のICQ UIN（ICQ番号）を取得する必要がある。KXicqの初回起動時に、登録ウィザードが起動し、すでにUINを持っている場合には、それを入力するプロンプトが表示される。持っていない場合には、KXicqがただちにUINの取得を試みる。このとき、とりあえずパスしておいて、のちほど改めて自分の個人情報をICQサーバに送信することもできる。個人情報は、オンラインで接続しているほかのユーザーからも閲覧できるので、何の情報を開示するかは、慎重に選択しよう。

オンライン状態では、相手がどのような利用状態にあるのかを知ることができる。また、自分自身の利用状態は、[ファイル] メニューの [状態] から選択すればいい。

コンタクトリストにユーザーを追加するには、[ファイル] メニューから [ユーザを追加/検索] を選択する。同じ画面で、ICQサーバに登録しているユーザーを検索することもできる。逆に、ICQサーバに登録しているユーザーが検索によってあなたを見つけ、コンタクトリストに登録した場合は、その旨を知らせるメッセージが届けられる。

KXicqではメインウィンドウのレイアウトを変更することも可能だ。それぞれのコンタクトリストの上部にある横線をドラッグすることで場所を移動したり、メインウィンドウから外したりできる。また、各リストの右上にある ボタンをクリックすると、リストを移動できなくなり、x ボタンをクリックするとそのリストを表示無効にできる。表示無効にしたリストを再び元に戻すには、ビューメニューから [オンライン / オフラインを表示する] を選択すればいい。



画面1 日本語対応バージョンのKXicq

Emacs上で動作する国産のIRCクライアント

Liece

バージョン: 1.4.3

ライセンス: GPL

<http://www.unixuser.org/ueno/liece/index.html.ja>

Emacs

日本語可

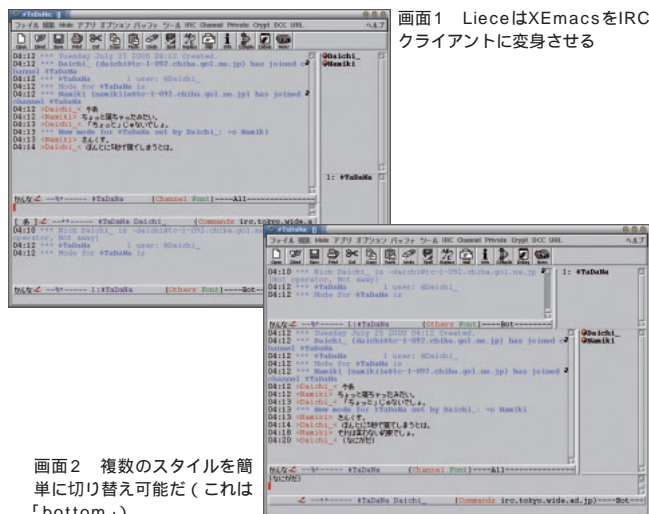
EmacsおよびXEmacsで動作する国産のIRCクライアントソフト。Windows用のIRCクライアント「CHOCOA」によく似たインターフェイスを持つ。最初から文字コード変換機能が備わっており、パッチなどで修正することなく日本語を扱える。動作には、APEL (A Portable Emacs Library) などのEmacsライブラリが必要だ。

Lieceでは、接続するサーバやポート番号、ユーザー名やニックネームなどを、設定ファイル(/liece/init.el)から読み込む。tarボール展開先のdocディレクトリにあるサンプル(sample.dot.liece)をリネームコピーして、内容を書き換えるとよいだろう。

Emacs上で“M-x liece”として起動すると、タイトル画面が表示され、設定ファイルで指定したIRCサーバに接続する。あとは、通常のIRCクライアントソフトと使い方は同じだ。チャンネルに参加するjoinなどのIRCコマンドは、Lieceのメニューから選択するだけで送信できるので初心者にも使いやすい(画面1)。もちろん、Emacsに手が馴染んでいる人向けに、Ctrl-Cで始まる2ストロークのキー操作(たとえば「C-cj」でチャンネルに参加など)も豊富に用意されている。なお、XEmacsでは、マウスの右クリックメニューから項目を選択することに

よってもコマンド入力を行える。

Lieceの特徴のひとつとして、画面(バッファ)の分割を「スタイル」と呼ばれる機構で柔軟に設定できる点あげられる。付属の「top」、「middle」、「bottom」の3スタイル(初期設定はmiddle)のほか、自分でカスタマイズしたスタイルを利用することも可能だ。スタイルを変更するには、メニューの[IRC]-[Change window style]を選択するか、「C-cs」を押してから、希望のスタイルを入力すればいい。その後、[IRC]-[Reconfigure windows]を選択するか「C-cr」を押すと、実際にウィンドウの分割状態が変化する(画面2)。



画面1 LieceはXEmacsをIRCクライアントに変身させる

画面2 複数のスタイルを簡単に切り替え可能だ(これは「bottom」)

GTK+を利用した高機能なIRCクライアント

X-Chat

バージョン: 1.4.2 (日本語版) / 1.5.5 (開発版) ライセンス: GPL

<http://xchat.linuxpower.org/>http://www.on.cs.keio.ac.jp/yasu/jp_gnome.html (日本語化)

X

GTK+

日本語可

X上で動作するグラフィカルなIRCクライアントソフト。IRC (Internet Relay Chat) は、TCP/IP環境で動作するサーバクライアント方式の分散型マルチユーザーチャットシステムで、X-ChatなどのIRCクライアントを利用してIRCサーバに接続し、同じチャンネルに参加している複数の相手と文字による会話を楽しめる。

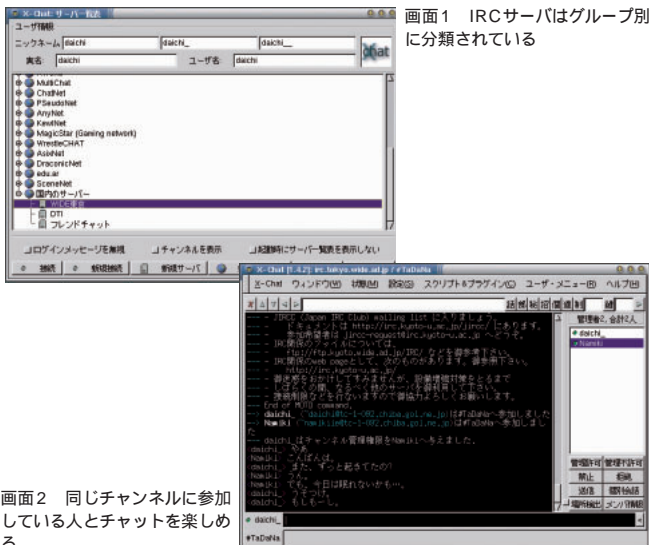
IRCでは日本語の文字コードとしてJISが使われるため、X-Chatで日本語を使えるようにしたパッチが別途配布されている。3つのパッチをすべて当ててから、“./configure --enable-libjcode” “make” とすると、日本語対応のX-Chatをビルドできる。

起動すると、メインウィンドウとともにサーバリストウィンドウが開く(画面1)。接続先のIRCサーバはグループ別に分類管理されており、ダブルクリックするだけで接続できる。初期設定では海外のサーバしか登録されていないので、国内のIRCサーバ(irc.tokyo.wide.ad.jpなど)を登録して接続するといい。

IRCでは、「/」で始まるコマンドを入力することでサーバにさまざまな要求を伝える(たとえば、チャットするためのチャンネルに参加するには“/join #チャンネル名”と入力する)。X-Chatでは、一部のコマン

ドをGUI操作で代替できる。たとえば、チャンネルの参加者一覧は常にウィンドウ右側に表示されており、ボタンや右クリックメニューで参加者の情報確認や管理権限の委譲などが可能だ。IRCに慣れていない人にも使いやすい機能だろう。

このほか、IRCサーバを介さずに他のIRCクライアントと直接接続するDCCや、複数のIRCコマンドを組み合わせる新しいコマンドを作成するユーザー定義コマンド、Perl/Pythonスクリプトやプラグインによる機能拡張など、中・上級者向けの機能も用意されている。



画面1 IRCサーバはグループ別に分類されている

画面2 同じチャンネルに参加している人とチャットを楽しむ



テキストベースで軽快に動作するICQクライアント

Micq

バージョン: 0.4.5/0.4.0 (日本語版) ライセンス: フリー

<http://phantom.iquest.net/micq/>
<http://www.tahoo.org/taku/unixicq.html> (日本語化)

コンソール 日本語可

ktermなどのコンソール画面で動作する軽量なICQクライアントソフト。表示はカラーで見やすく、カーソル制御を行わないため非力なマシンでも軽快に動作する。ktermをシフトJIS漢字モードに切り替えれば、そのまま日本語メッセージのやりとりが可能だが、日本語EUCモードで使うためのパッチも別途配布されている。

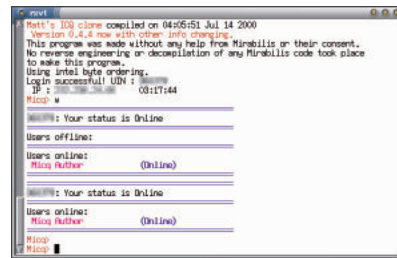
初回起動時には、ユーザー識別用のICQ番号 (UIN) の入力を求められる。入力を省略すると、MicqがICQユーザーとしての登録を自動的に行って新たなICQ番号を取得する。「Micq>」というプロンプトが表示されたら接続成功だ。

Micqでは、すべての操作をコマンドラインからのコマンド入力で行う。たとえば、コンタクトリストに登録したユーザーがオンラインかどうか調べるには“w”と入力してEnterキーを押せばいい(画面1)。オンラインのユーザーだけ表示するなら“e”、コンタクトリストへの登録は“add 相手のICQ番号 ニックネーム”だ。

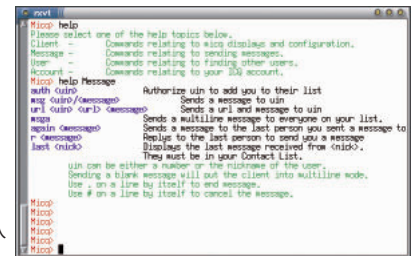
メッセージを送信するには、最初の一度だけ“msg 相手のICQ番号/メッセージ”として、ICQ番号を指定する必要がある。以降は、“again メッセージ”で同じ相手にメッセージを送ることができる。また、到着

したメッセージに対しては、“r メッセージ”とするだけで返信可能だ。なお、“/help ジャンル名”とすることにより、用意されているコマンドの一覧がジャンル別に表示される(画面2)。

くどうたく氏が作成した日本語化パッチを利用すれば、送受信時に文字コード変換が行われ、ktermの日本語EUCモードのまま使えるようになる。ただし、パッチは0.4.0用なので、最新の0.4.5に対しては一部手作業で修正しなくてはならない。また、Makefile内の言語の設定は英語のままにしておく必要がある。



画面1 コンソール画面でICQを利用できる



画面2 操作はすべてコマンド入力で行う

サーバいらすのメッセージ交換ソフト

Gtklpmmsg

バージョン: 0.2.3 ライセンス: asis/GPL

<http://free.prohosting.com/nobori/gtklpmmsg.shtml>

X GTK+ 日本語可

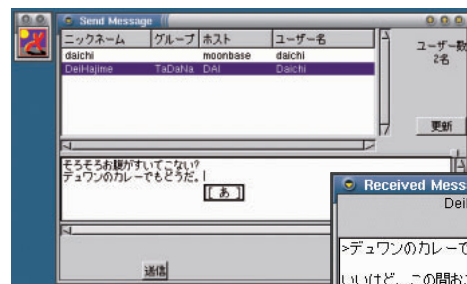
シンプルメッセージ送受信ソフト「IP Messenger」のGTK+版。IP Messengerは、サーバを用意する必要がなく気軽に使えるのが特徴で、イントラネットでのメッセージ交換に威力を発揮する。Gtklpmmsgのほかにも、Windows版やMac版、X11R4版、Java版のクライアントソフトが作られているので、イントラネットに接続されたさまざまなOSのマシン同士で相互にメッセージをやりとりできる。

起動すると、小さなアイコン状のウィンドウが開く。ウィンドウマネージャの設定でタイトルバーを表示しないようにして使うとよいだろう。このウィンドウをクリックすると、現在IP Messengerを使っているユーザーの一覧を含むウィンドウが開く(画面1)。メッセージを送るには、送信先となるユーザーを選択し、下部のエディットボックスにメッセージ内容を入力してから[送信]ボタンを押せばいい。一度に複数の人にメッセージを送ることも可能だ。

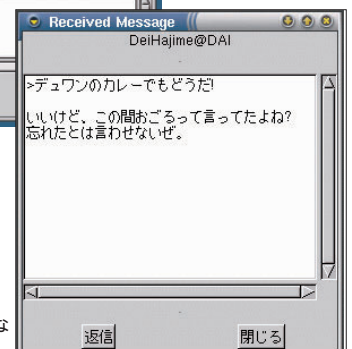
なお、Gtklpmmsgから送ったメッセージは、デフォルトで「封書」扱いとなる。このため、送信先の画面には送信元のユーザーのニックネームなどがまず表示され、相手が[開封]ボタンを押すとメッセージの内容が表示される。また、送信元には相手が開封した時刻が表示される。

ほかのクライアントから送られてくるメッセージは、そのまま内容が表示されるものもあれば(画面2)、Gtklpmmsgと同様に封書扱いで送られてくるものもある。封書扱いの場合は、[開封]ボタンを押してメッセージの内容を表示しよう。届いたメッセージに返事を出すときは、[返信]ボタンを押して返事の内容を書けばいい。自動的に相手のメッセージが引用された状態で書き始められる。

設定を変更するには、アイコン状のウィンドウを右クリックし、メニューから[設定]を選択する。このダイアログでは、自分のニックネームやグループなどを変更可能だ。



画面1 送信先のユーザーを選んでメッセージの内容を入力する



画面2 受信したメッセージはこのようなウィンドウに表示される

定番FTPクライアントソフト

IgllooFTP PRO

バージョン: 1.0.0 ライセンス: シェアウェア (\$29.95)

<http://www.littleigloo.org/iglooftp.php3>

X GTK+ 日本語可

PC雑誌やWebサイトで数多くの賞を獲得しているFTPクライアントの定番ソフト。ローカル/リモートファイルの同時表示、ツールバーや右クリックメニューによる操作、FTPサイトのブックマーク管理など、とても使いやすいソフトに仕上がっている。試用期間30日のシェアウェアだが、機能制限はいっさい設けられていない。

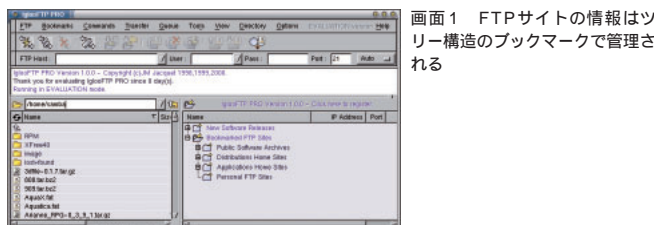
起動すると、多数のペイン(領域)に分割されたウィンドウが開く。FTPサイトに接続していない状態では、右のペインにはFTPサイトのブックマークとIceWalker.com提供の新作ソフト情報がツリー表示される(画面1)。ブックマークは、グループ別のフォルダで分類されており、著名なディストリビューションやプロジェクトのFTPサイトが登録済みだ。もちろん、ユーザーによる追加も可能だ。

ブックマークを登録したFTPサイトについては、ブックマークをダブルクリックするだけで接続処理が行われる。FTPサイトに接続すると、右ペインがリモート(接続先のFTPサイト)のファイル一覧に切り替わり、左右のペインにローカルとリモートのファイルが同時に表示された状態になる(画面2)。

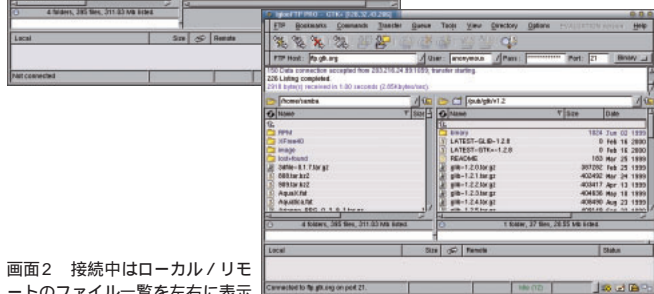
どちらのファイルに対しても、クリックで選択後、右クリックメニュー

で閲覧、編集、削除、属性変更などの操作を行える。付属のビューアやエディタは日本語を問題なく扱えるし、外部エディタを利用することも可能だ。ファイル転送に関しては、選択したファイルを即座に転送する方法と、ファイルの情報をいったんキューに溜めておき、まとめて一括転送する方法が用意されている。

このほか、クリップボードにFTPのURLがコピーされると、即座に接続を開始する「クリップボード監視」機能や、FTPサーバ間でディレクトリ内のファイルをまるごと転送する「FXP transfers」機能など、ユニークな機能が満載だ。



画面1 FTPサイトの情報はツリー構造のブックマークで管理される



画面2 接続中はローカル/リモートのファイル一覧を左右に表示

シンプルなX対応FTPクライアント

gFTP

バージョン: 2.0.7 ライセンス: GPL

<http://gftp.seul.org/>

X GTK+ 日本語可

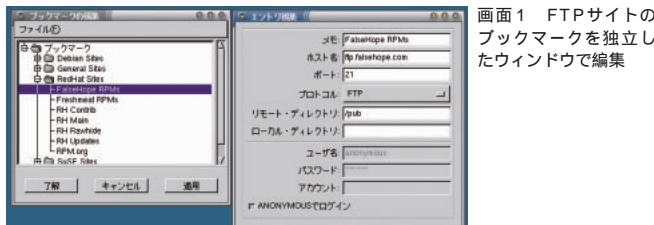
ローカルとリモートのファイル一覧が左右のペインに表示されるシンプルな2パネル構成のFTPクライアント。日本語カタログが標準で用意されており、日本語環境ではメニュー表示やオプション設定ダイアログ、各種のメッセージに至るまで日本語で表示される。

FTPサイトに接続するには、ホスト名などを直接ツールバーの各エリアに入力する方法のほか、あらかじめブックマークに登録しておいて[ブックマーク]以下のメニューから選択することも可能だ。gFTPのブックマークはフォルダで分類されており、独立したウィンドウでその内容を編集できる(画面1)。Red Hat、Debian、SuSeといったディストリビューションや、GTK+、GNOME、KDEなどのプロジェクトのFTPサイトが最初から登録されており、ユーザーがフォルダやFTPサイトを新規登録することも可能だ。

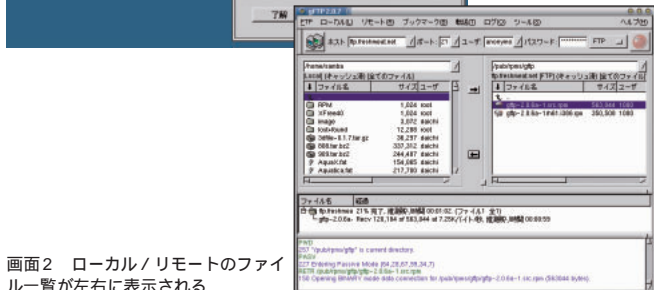
FTPサイトに接続すると、右ペインがリモート(接続先のFTPサイト)のファイル一覧に切り替わり、左右のペインにローカルとリモートのファイルが同時に表示される。あとは、目的のファイルをクリックで選択し、中央部の矢印ボタンを押すだけで、即座にダウンロード/アップロードできる。ファイル転送中は、その下のペインにツリー形式で転送状

況が表示される(画面2)。

このほか、右クリックメニューで選択したファイルの閲覧や編集、削除、名前変更といった操作が可能だ。gFTPの内蔵ビューアは日本語EUCの表示に対応しているが、使いなれた外部ソフトを利用することもできる。[FTP] - [オプション]で設定ダイアログを開き、[閲覧プログラム]に日本語対応のビューアやエディタを設定すればいい。なお、エディタは内蔵していないため、gFTPでファイルを編集するには、あらかじめ設定ダイアログの[編集プログラム]にEmacsなどのエディタを設定しておく必要がある。



画面1 FTPサイトのブックマークを独立したウィンドウで編集



画面2 ローカル/リモートのファイル一覧が左右に表示される



ファイル名補完機能やカラーls コマンドを備えたFTPクライアント

yafc

バージョン : 0.5.6

ライセンス : GPL

<http://www.stacken.kth.se/~mhe/yafc/>

コンソール 日本語可

コンソールベースで動作する軽快なFTPクライアントソフト。標準のftpコマンドに似たシンプルなコマンドラインインターフェイスながら、ファイル名補完やエイリアス、カラーlsといった便利な機能・コマンドが満載だ。また、日本語端末 (ktermなど) で動かせば、日本語を含むテキストを問題なく表示できる。

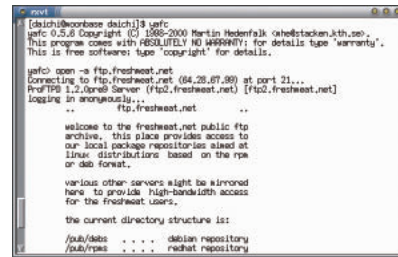
インストール後は、tarボールに含まれる「yafcrc.sample」を、yafcを利用するユーザーのホームディレクトリの「.yafc/yafcrc」にコピーし、Anonymous接続時のパスワード (ユーザーのメールアドレス) などの設定を行う必要がある。

起動すると「yafc>」というプロンプトが表示されるので、openやlsなどのコマンドを入力してサーバの接続やファイル一覧表示を行う (画面1)。なお、コマンド一覧はhelp、各コマンドの詳細は“コマンド名 help”で確認できる。

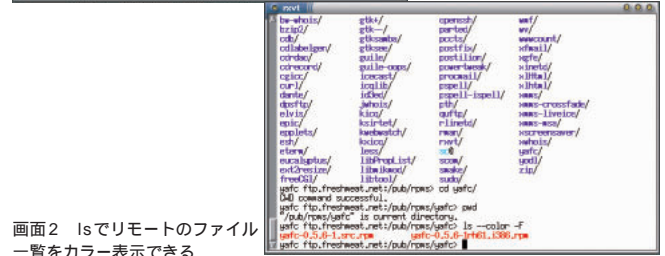
readlineライブラリを使用しており、bashのコマンドラインと同様の編集機能やヒストリー機能を利用可能だ。ふだんコマンドラインで作業することが多い人には、GUIのFTPクライアントよりも使いやすいだろう。さらに、リモートとローカルのファイルをシームレスに扱うための

工夫が随所に見られる。たとえば、リモートのファイルに対するTabキーによるファイル名補完や、“ls --color”によるカラー表示など (画面2)。なお、エイリアス機能を利用して“alias ls 'ls --color'”とすれば、以後“ls”とするだけでカラー表示が有効になる。

ファイルの送受信はputとgetで行う。単にファイル名を指定するだけでなく、指定したディレクトリ以下のファイルを再帰的に転送したり、あらかじめtagで印を付けておいたファイルをまとめて転送することも可能だ。このほかにも、接続手続きを簡略化するブックマーク機能やオートログイン機能が用意されている。



画面1 yafcのコマンドラインはbashと同様の編集機能を持つ



画面2 lsでリモートのファイル一覧をカラー表示できる

ドラッグ&ドロップ操作にも対応したダウンロードマネージャ

Caitoo

バージョン : 0.6.5

ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

Caitooは統合デスクトップ環境KDEのダウンロードマネージャだ。ダウンロードマネージャとは、ファイルをダウンロードするという作業はもちろんのこと、あとでまとめてダウンロードできるようにリストに記録しておいたり、ファイルの拡張子別に格納するディレクトリを決めたりと細かくケアしてくれるアプリケーションである。

インストールすると、KDEのスタートメニューの[インターネット]サブメニューに登録される。また、コマンドラインから起動することもできる。コマンドラインから起動する場合、“caitoo http://xxx.xxx...”とタイプすることで、転送を行うURLを指定できる。

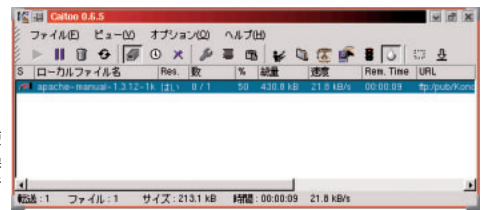
ダウンロードしたいファイルまたはプログラムがある場合、通常は[ファイル]メニューから、[転送を開く]を選択する。URLを入力するプロンプトが表示されるので、そこにお目当てのURLを入力しよう。

Netscapeでサイトを探索中に、ダウンロードしたいファイルやプログラムがあった場合は、なにもCaitooを使用しなくともネットスケープでそのままダウンロード保存してしまえばいいが、ほかのソフト、たとえばメーラなどで、知人がURLを紹介してくれたり、仕事でデータの取得を行わなければならない、といった状況になったとき、いちいちブラウザ

ウザを起動するのはちょっと.....と思うことがあるだろう。こんなとき、Caitooは大変便利だ。

たとえば、NetscapeやkfmでダウンロードしたいURLを見つけたとしよう。そのURLをドラッグし、caitooのメインウィンドウにドロップすると、caitooのダウンロードリストにそれが追加され、ダウンロードが開始される。クリップボードにURLをコピーし、それをメニューの[転送を貼り付け]を選択することでリストに加えることもできる。この操作は、Netscapeやkfmだけでなく、Kmailやその他のURLを頻繁に扱うようなKDEアプリケーションでドラッグ&ドロップや「クリップボードに複写」に対応しているソフトでも行える。

ダウンロードの状況をログウィンドウで確認する機能も備えている。また、メニューの設定で「サウンドを使う」をチェックしていれば、何らかのアクションが起きた際に、サウンドを奏でて知らせてくれる。なお、ドック化して使用すれば、貴重なデスクトップスペースを使用せず、ウィンドウを開かなくともウィジェットにURLをドラッグできる。ドック化するには、ツールバーのドック化ボタンをクリックするだけでいい。ドック化されたドックウィジェットを右クリックし[元に戻す]を選択することで、ウィンドウを元に戻せる。



画面1 Caitooを使えばシンプルなGUI操作でダウンロードを行える

複数ファイルを効率よくダウンロード

Downloader for X

バージョン: 1.17

ライセンス: フリー

<http://www.krasu.ru/soft/chuchelo/>
X GTK+ 日本語可

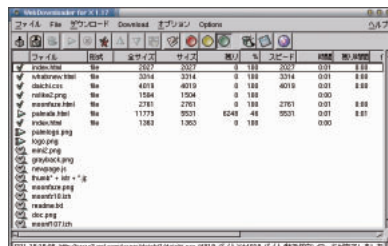
HTTP / FTP プロトコルを利用してファイルのダウンロードを行う自動巡回ソフト。指定されたURL以下のファイルを再帰的に取得したり、ファイルの種別を限定することも可能だ。このほか、中断した箇所から受信を再開するレジューム、指定した時間に受信を開始するタイマー予約といったさまざまな機能を備えている。日本語カタログが付属しており、日本語表示で使えるのもうれしい。

起動すると、上下2ペイン構成のウィンドウが開く。上のペインにはダウンロード待ちリスト(キュー)、下のペインには各種メッセージがそれぞれ表示される(画面1)。ダウンロードするファイルの登録は、直接URLをキー入力するほか、Webページのリンクをブラウザからドラッグ&ドロップするか、クリップボード監視機能を利用してURLをクリップボードにコピーしてもOKだ。登録したファイルはキューの末尾に追加され、ツールバーの[ダウンロードの再開/停止]ボタンを押すと、キューの先頭から順番にダウンロードされる。

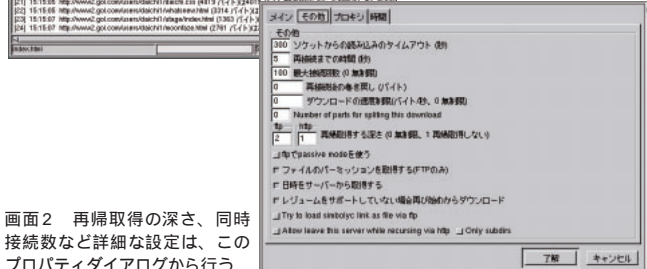
Webページ内の画像やリンク先のページを取得したり、FTPサイトのディレクトリの内容をまとめて取得するには、各ファイルのプロパティ(画面2)で、[再帰取得する深さ]の設定値を変更する必要がある。1

段階の再帰を行う場合は「2」、制限なしの再帰取得は「0」、再帰取得しないなら「1」を設定する。Webページの初期値は1なので、URLで指定したページのHTMLファイルしか取得しない。一方、FTPの初期値は2なので1段階の再帰取得が行われる。このほか、受信を開始する日時を設定することも可能だ。

全体の設定を変更するオプションダイアログでは、同時にダウンロードするファイル数や受信速度の制限値も設定可能だ。また、プロパティで個別に設定する内容、たとえばサブフォルダやプロキシサーバなどの初期値もこのダイアログで設定できる。



画面1 上のペインにダウンロード対象のファイルが並び



画面2 再帰取得の深さ、同時接続数など詳細な設定は、このプロパティダイアログから行う

検索機能が充実した多機能Napsterクライアント

Knapster

バージョン: 0.2.1

ライセンス: GPL

<http://knapster.netpedia.net/>
X KDE Qt

Napsterサーバに登録されているMP3ファイルリストから検索したり、見つけた曲をダウンロードしたりといった、さまざまな機能を備えたKDEのNapsterクライアント。動作には、KDE-1.1.2以降およびQt-1.44以降が必要となる。

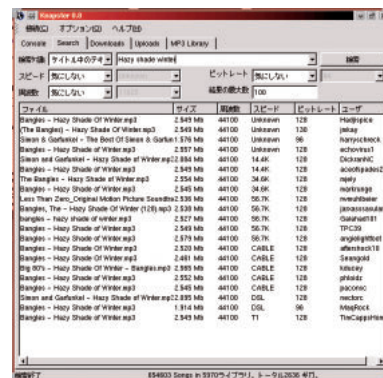
Knapsterを使用するためには、まず最初に[オプション]メニューの[設定]から自分のユーザー情報を入力し、napsterサーバに接続する。すでに別のユーザーがパスワードやユーザー名を使用している場合もあるので、そのときは、設定画面でパスワードなどを入力し直して再度送信してみよう。

認証が完了したら、次は曲の検索だ。[Search]タブで、探している曲の曲名あるいはアーティスト名をテキストフィールドに入力し、[検索]ボタンをクリックするか、あるいはEnterキーを押そう。もっと細かく検索条件を指定することも可能だ。スピード、周波数、ビットレートなど、いろいろ指定することで、検索対象を絞り込むことができるようになっていく。サーバに検索したファイル名が登録されていれば、検索結果として[Search]タブ画面の下のリストボックスに表示される。このフィールドには、ファイル名、サイズ、周波数、スピード、ピ

ットレート、ユーザの各項目が表示される。ここに表示されたファイルが必ず存在し、確実に入手できるというわけではないので、注意してほしい。

サーバとの通信中のエラーや検索結果をユーザーに知らせる目的で、通信記録が[Console]タブの画面に表示される。[Console]タブのウィンドウは上下2つに分かれていて、上部に通信結果が表示される。最下部にあるテキストボックスは「コマンドフィールド」と呼ばれ、サーバに送る各種コマンドを入力できる。ここで使用できるコマンドは、IRC(Internet Relay Chat)のコマンドに似ており、「/」ではじまる。このコマンドフィールドにおいて「/help」とすれば、使用できるコマンドの一覧が表示される。また、「/list」で利用可能なチャンネルのリストが表示される。その他の使用できるコマンドについての詳しい説明は、付属のヘルプを参照してほしい。

Knapsterは通知機能も備えている。検索などのタスクを実行中に、ほかのアプリケーションにカレントを移していても、タスクが終了するか、何らかの結果(イベント)が発生した時点で、Knapsterがピープ音を鳴らして知らせてくれるので、安心して他の作業に集中することができるだろう。





Archieサーバからプログラムをゲット!

karchie

バージョン : 1.1.2

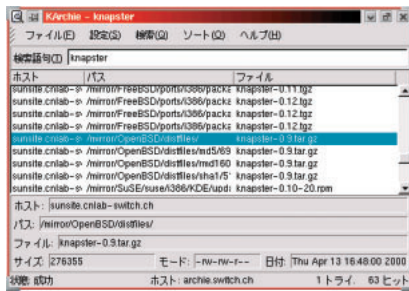
ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

世界中に広がる開発ネットワークを持つLinuxを使用していて、さまざまなプログラムを、ディストリビューション単位で、あるいはバージョン単位で比較してみたいと思ったことはないだろうか？ または、最新版のプログラムをダウンロードしたいのに、どこにあるのかわからないなんてことは？ そんなとき、karchieが大いに役立つはずだ。karchieは、プログラム名やファイル名をArchieサーバから検索し、その検索結果から欲しいファイルをダウンロードするためのアプリケーションだ。

検索モードも正規表現を使用したアバウトな検索から、正式なファイル名による精密な検索まで幅広く対応しているので、状況に応じて検索モードを変更することで便利な検索を実現できる。



Webページを取得するwgetのフロントエンド

KWebGet

バージョン : 0.5/0.6 (KDE2用)

ライセンス : GPL

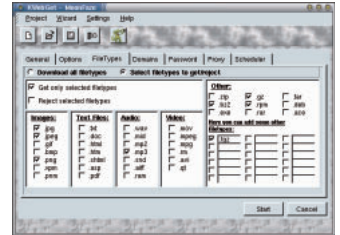
<http://www.kpage.de/en/kwebget/>

<http://www.gnu.org/software/wget/wget.html> (wget)

X Qt KDE 日本語可

WebサイトやFTPサイトのファイルを一括取得するツール「wget」のフロントエンド。取得を開始するURLをはじめ、リンクをたどる数や、対象とするドメイン、ファイルタイプなどの設定を、複雑なコマンドラインオプションの代わりに、わかりやすく分類されたパネルで行える。

起動すると、最初に2つの実行モードを選択するウィンドウが開く。初心者向けのウィザードモードでは、保存先ディレクトリやwgetの動作など基本項目について、順を追ってウィザード形式で設定する。最後の画面で [Start-Now] を選択して [Finish] ボタンを押すと、wgetが実行されてファイルの取得が行われる。もう一方のアドバンスモードでは、ジャンル別にページ分けされたウィンドウが開き、wgetの詳細な設定を行うことができる(画面2)。設定を「プロジェクト」としてファイルに保存し、再利用することも可能だ。



ネットニュースからバイナリファイルを取り出す

Glitter

バージョン : 0.9.1

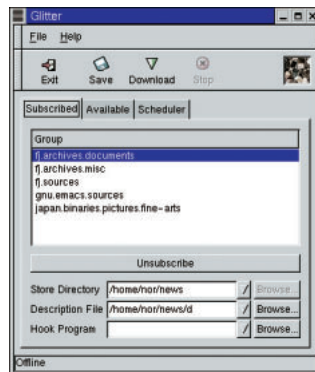
ライセンス : GPL

<http://www.kde.org/>

X GNOME GTK+

glitterはネットニュースに投稿されるバイナリファイルを切り出して保存するための専用ツールだ。取得したいニュースグループを指定し、そこに投稿されたすべてのバイナリファイルを指定したディレクトリに保存する。さらにglitterはスケジューラも内蔵しており、ネットワークがすいている時間や、夜間を指定して動作させることができる。

最初の起動時に、まずニュースサーバの名前を設定する。続いて購読したいニュースグループを選択して、保存すべきファイルの種類を拡張子で指定するればよい。設定が終わったら「Download」アイコンをクリックすると、ニュースサーバにアクセスが始まり、設定したすべてのニュースグループに対して切り出しがはじまる。



GNU debugger (gdb) のGUIインターフェイス

KDbg

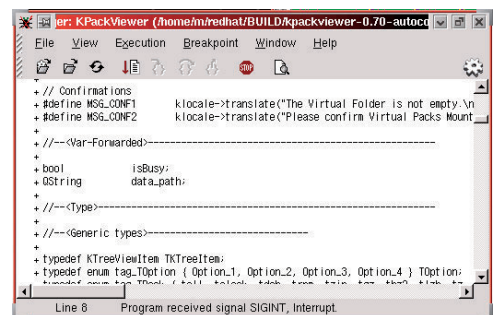
バージョン : 0.5/0.6 (KDE2用)

ライセンス : GPL

<http://www.kde.org/>

X Qt KDE

KDbgは、GNU debugger (gdb) のKDE GUIインターフェイスだ。gdbは高機能なデバッグアプリケーションで、多くのプログラマーに利用されている。使用方法は、基本的にgdbと同じだ。まず、デバッグしたいプログラムを“-g”オプションを付けてコンパイルする。そして、[file]メニューから [Executable..] を選択し、デバッグしたいプログラムの実行ファイルを選択すればいい。ほかに、ソースコードの表示や実行中のプログラムのアタッチ、ブレイクポイントの設定などもGUIで行える。Kdevelopがインストールしてあれば、Kdevelop上から起動することもでき、非常に効率よくアプリケーションの開発を行える。



KDE版エクスプローラ

kruiser

バージョン: 0.4

ライセンス: GPL

<http://www.kde.org/>

X KDE Qt

kruiserは、KDE用エクスプローラとも呼べるWindows 9xライクなリック&フィールを備えたファイルマネージャだ。機能的にも、Windowsを使った経験をお持ちの方には馴染みのものばかりだ。

インストールすると、KDEのスタートメニューの[ユーティリティ]にその名も「KDE Explorer」として登録される。ここから起動してもいいし、Windowsライクな使い方をお望みなら、ショートカットをデスクトップ上に作成するのもよいかもかもしれない。

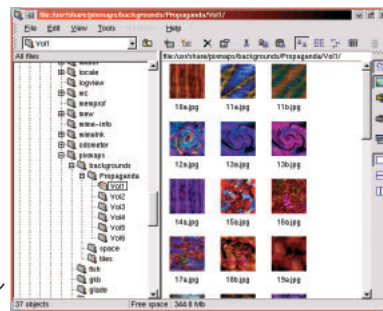
kruiserは非常に多くの機能を実装しながら、使い方はいたって簡単だ。ディレクトリ構造が左側のペインにツリー表示され、右側のペインにカレントディレクトリのファイルリストが表示される、非常にわかりやすい構造になっている。もちろん、必要に応じて仕切りを移動し、レイアウトを変更することも可能だ。また、mountデバイスにも対応しているので、マウントしているデバイスの内容もシームレスに閲覧することができて大変便利だ。さらに、圧縮したファイルの内容も、そのファイルを右クリックし、メニューから[Archiver]を選択することで展開せずに見ることができる。右端のviewToolバーにある圧縮アイコンをクリックすることで、左側のディレクトリビューに圧縮ファイル

の中がツリー表示される。

kruiser上では、ディレクトリ作成から、ファイル削除までほとんどの作業をGUIで行える。当然、ドラッグ&ドロップにも対応しているので、ファイルやディレクトリをカットまたはコピーし、別の場所に貼り付けたり移動したりすることも可能になっている。ファイルをリネームするには、ファイル名をクリックし、入力を受け付ける状態にしてから、新しい名前を入力してEnterキーを押すだけ。

FTPサーバに接続して、そのディレクトリ構造を見ることも可能。FTPやSMBに接続するためには、まず[Tools]メニューの[Connect New device]を選択する。プロパティウィンドウが表示され、このウィンドウでホスト名や接続先のタイプ、名前などを入力して[OK]ボタンをクリックすれば接続完了だ。うまく接続できた場合は、左側の「Remote_Computer」部分に新しい接続が表示される。右ペインのファイルリストウィンドウでは、ファイルのタイプに合わせて表示されるアイコンが決められるが、メインウィンドウの右端にあるツールバーの[Show Image]を有効にしておくと、画像ファイルはサムネイル形式で表示されるようになる。また、[Tools]メニューの[find]を選択してKfindを起動し、ファイルの検索を行う機能も備えている。

画面1 画像ファイルはサムネイルでも表示可能



付属ソフトの豊富な日本語対応ファイルマネージャ

Cxplore

バージョン: 1.3.4

ライセンス: GPL

<http://www.threeweb.ad.jp/hatakeda/cxplore/index.htm>

X GTK+ 日本語可

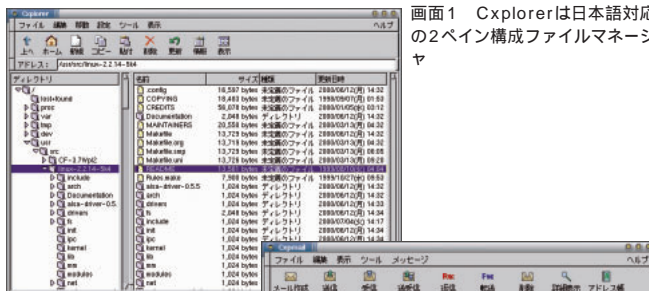
日本語ファイル名に完全対応した国産のファイルマネージャ。メニューやダイアログは最初から日本語で表示される。将来的にはGNOMEやKDEのような統合デスクトップ環境の構築を目指しており、簡易テキストエディタなどの関連ソフトが付属する。

起動すると、ディレクトリツリーとファイル一覧の2ペイン構成のウィンドウが表示される(画面1)。ディレクトリツリーやファイル一覧で使われるフォルダアイコンは、MacOS風のデザインのほか、Windows、KDE、GNOMEのスタイルに変更可能だ。ファイルマネージャとしての操作は、GMCやKFMと似たごく一般的なもので、処理するファイルを選択し(範囲/複数選択も可能)、ツールバーのボタンや右クリックメニュー、ショートカットキーなどで処理を実行する。

ツールバーの[ツール]メニューからは、Cxploreと一緒に配布されている付属ソフトを呼び出せる。拡張子の関連付け(Cxpapp)などCxploreに密着したソフトだけでなく、カレンダー(Cxpcalen)や計算機(Cxpcalc)、アドレス管理(Cxpaddr)などのアクセサリ的なソフトや、簡易テキストエディタ(Cxpnote)やメールクライアント(Cxpmail)、イメージビューア(Cxpimage)などのアプリ的なソフト

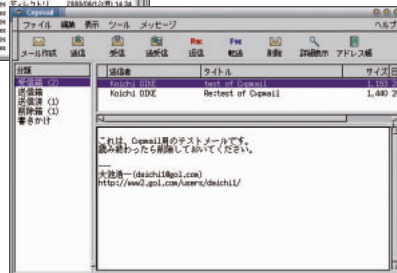
まで揃っている(単独実行も可能)。このほか、ウィンドウマネージャやバイナリエディタ、アーカイブ管理ソフトなども予定されている。

Cxpnoteは、日本語を入力・編集できる簡易テキストエディタで、コピー&ペーストなどの基本的な機能が用意され、設定ファイルなどの閲覧や変更であれば十分に使えるレベルだ。表示に使われるフォントは、3種類のサイズ(14/16/24ドット)を切り替えられる。Cxpmail(画面2)は日本語対応のメールクライアントで、POP3/SMTTPによるメールの送受信、受信メールの自動振り分け、複数アカウントの切り替えに対応している(ファイルの添付は未対応)。



画面1 Cxploreは日本語対応の2ペイン構成ファイルマネージャ

画面2 付属ソフトのCxpmailはかなり使えるメールクライアントだ





柔軟な設定が可能なファイルマネージャ

gentoo

バージョン: 0.11.14

ライセンス: GPL

<http://www.obsession.se/gentoo/>

X GTK+ 日本語

2つのディレクトリのファイル一覧が表示され、コピーや移動の結果を確認できる2ペイン構成のファイルマネージャ。ボタンの内容やキーバインドなどあらゆる設定をダイアログ上で変更でき、ファイル一覧に関しては左右のペインで異なる設定も可能だ。

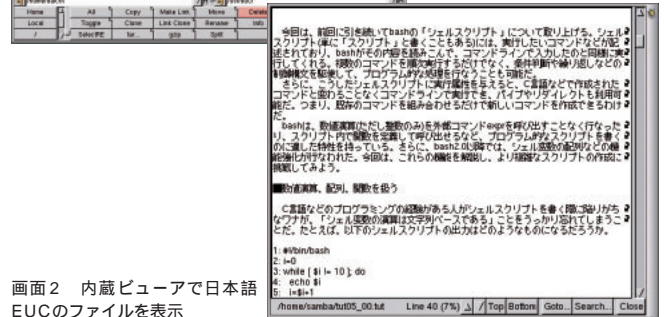
なお、ソースファイルgentoo.cの“gtk_init (...);”の前に“gtk_set_locale ();”という行を追加するだけで、ファイル一覧や内蔵ビューアで日本語が扱えるようになる。また、最初はアイコンが正常に表示されないで、右下の [Configure] ボタンで設定ダイアログを開き、[Paths & Hide] ページの [Icons] に、アイコンデータをコピーしたディレクトリを設定する必要がある。

ウィンドウは、左右のファイルリスト表示ペインと、その下に並んだコマンドボタンで構成される (画面1)。各ペインには、現在フォーカスしているディレクトリのファイル一覧が表示されている。コピーや削除、閲覧などのファイル操作を行うには、ファイルをクリックで選択し (ドラッグによる範囲指定も可能) 目的のコマンドボタンを押せばいい。コマンドボタンは、設定ダイアログの [Buttons] ページで追加や修正を行える。ボタンを日本語表記にすることも可能だ。

ファイル名をダブルクリックした場合は、「ファイルタイプ」に応じた処理が行われる。たとえば、テキストファイルの場合は内蔵ビューアで内容が表示され (画面2)、アーカイブならばファイル一覧、画像ファイルならばxvでイメージが表示される。ファイルタイプは、単に拡張子で判別するだけでなく、ファイル名の正規表現パターンによるマッチやファイル属性などを組み合わせた柔軟な設定が可能だ。一方、ダブルクリックに対する動作や表示色・アイコンの設定は「ファイルスタイル」として独立しており、複数のファイルタイプで同じファイルスタイルを共有できる。



画面1 2つのディレクトリのファイル一覧を同時に参照できる



画面2 内蔵ビューアで日本語EUCのファイルを表示

viライクなキー操作の2画面ファイルマネージャ

Vide

バージョン: 0.2.6

ライセンス: GPL

<http://vide.sourceforge.net/>

X GTK+ 日本語

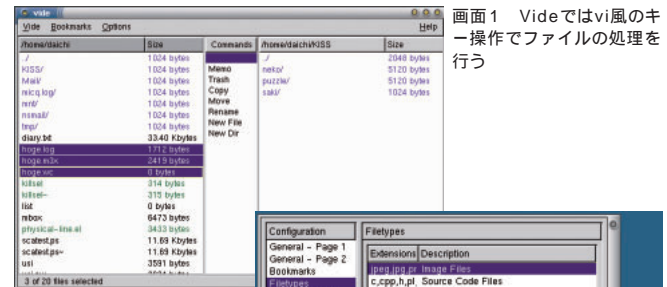
2つのディレクトリのファイル一覧が表示され、コピーや移動の結果を確認できる2ペイン構成のファイルマネージャ。viライクなキー割り当てを採用していることが最大の特徴で、viのキー操作になじんでいる人に特に勧めたい。h/j/k/lキーによるカーソル移動をはじめ、「:」コマンドによるファイル処理など、キーボードから手を離すことなくあらゆる処理を行える。なお、日本語環境では、そのままダイアログの表示などが文字化けしてしまうので、ソースファイルsrc/vid.cの“gtk_init (&argc, &argv);”の前に“gtk_set_locale ();”を追加してからビルドしよう。

起動すると、左右に2つのファイルリスト、中央部にコマンドリストを持つウィンドウが開く (画面1)。カーソルを1行下に移動させるにはjキー、上に移動させるにはkキーを押せばいい。メニュー項目もhjklキーで移動できる。なお、カーソルを持つ「アクティブ」なリストは、インデックス部分が濃い灰色で表示される。カーソルの切り替えには、Tab (あるいはスペース) キーを利用する。

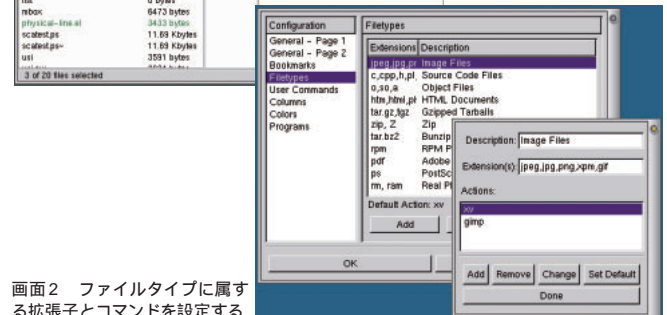
カーソルをテキストファイルまで移動させてEnterキーを押すと、ビューア (初期設定ではrxvt上のvim) が起動する。一方、画像ファイル

などの場合は、ファイルタイプに応じたコマンドが実行される。ファイルタイプに属する拡張子と実行するコマンドは、設定ダイアログの [FileTypes] ページで設定されており (画面2)、ユーザーによるファイルタイプの追加や設定の変更が可能だ。

コピーや削除といったファイル処理を行うには、コロン (:) に続けてコマンドを入力してEnterキーを押す。たとえば、ファイルを削除するには、“:e”とEnterキーを押せばいい。通常はカーソル位置のファイルが対象となるが、/キーに続けてパターンを指定して複数のファイルを対象にすることもできる。



画面1 Videではvi風のキー操作でファイルの処理を行う



画面2 ファイルタイプに属する拡張子とコマンドを設定する

Windowsからext2ファイルシステムにアクセス

Explore2fs

バージョン: 1.0.0pre3

ライセンス: GPL

<http://uranus.it.swin.edu.au/jn/linux/explore2fs.htm>
Windows 9x / NT4 / 2000

Windows 9x / NT / 2000でLinuxのext2ファイルシステムを読み書きするファイルマネージャ。ハードディスク上のLinuxパーティションをはじめ、ext2ファイルシステムでフォーマットされたフロッピーやZIPなどにも対応している。

ルック&フィールはエクスプローラによく似た2ペイン構成で、左側にはext2ファイルシステムのパーティションのディレクトリ構造がツリー表示され、右側には選択したディレクトリのファイル一覧が表示される。ファイルの表示方法は、大小のアイコン表示・一覧表示・詳細表示を切り替え可能だ。

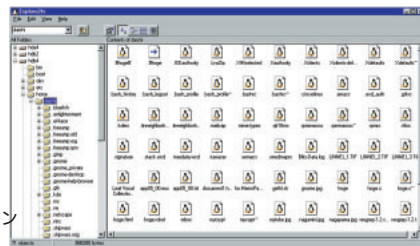
ファイルに対する操作は、エクスプローラと同様にドラッグ&ドロップや右クリックメニューを使用する。たとえば、Windowsのドライブにファイルをコピーするには、Explore2fsからエクスプローラなどにファイルをドラッグ&ドロップすればいい。また、テキストの内容を閲覧するには、右クリックメニューから[View]を選択する。文字コードを自動判別するテキストビューアを用意すれば、日本語EUCで書かれたテキストの閲覧も可能となる。

詳細表示でユーザー名やグループ名を表示するには、「/etc/passwd」

(シャドウ化済みのもの)と「/etc/group」をWindows側の適当なフォルダにコピーし、オプションダイアログの[Users]ページで指定すればいい。以後、ユーザーIDとグループIDの数値に対応する名前が表示されるようになる。

一方、書きこみに関する機能はより危険でテストが足りないため、初期設定では無効になっている。これを有効にするには、オプションダイアログの[Debug]ページで[Enable write support]をチェックし、Explore2fsを再起動する必要がある(起動時に確認される)。これで、エクスプローラなどからExplore2fsにドラッグ&ドロップでファイルをコピーしたり、右クリックメニューでのファイル名や属性の変更、シンボリックリンク作成などが可能になる。なお、WindowsからコピーしたファイルはユーザーIDとグループIDがともに0になるので、プロパティの[Attribute]ページで適当な数値に変更しよう。

Explore2fsをWindowsとLinuxのデュアルブートマシンで利用すると、起動時スクリプトなどの設定ミスをシングルモードなどを使わずに修正できるので便利だ。ただし、ファイルに設定されたアクセス権限に関係なく全ファイルを読み書きできるため、大きなセキュリティホールにもなりかねない。他人が直接パソコンに触れられる環境での運用には注意が必要だ。



画面1 Linuxのパーティションの内容が表示される

Sambaの共有ディレクトリをGUIでマウント

LinNeighborhood

バージョン: 0.6.0

ライセンス: GPL

<http://www.bnro.de/schmidjo/>

X GTK+ 日本語可

Sambaで共有しているWindowsマシンや他のLinuxマシンのディレクトリを階層的に表示し、マウント/アンマウント操作をGUIで行えるようにするソフト。実行には、smbmount、smbumount、smbclientなどのツールが必要だ。smbmountとsmbumountは、カーネル2.2以降ではSamba 2.xに付属するが、オプションなので付属しないディストリビューションもあるので確認しておこう。

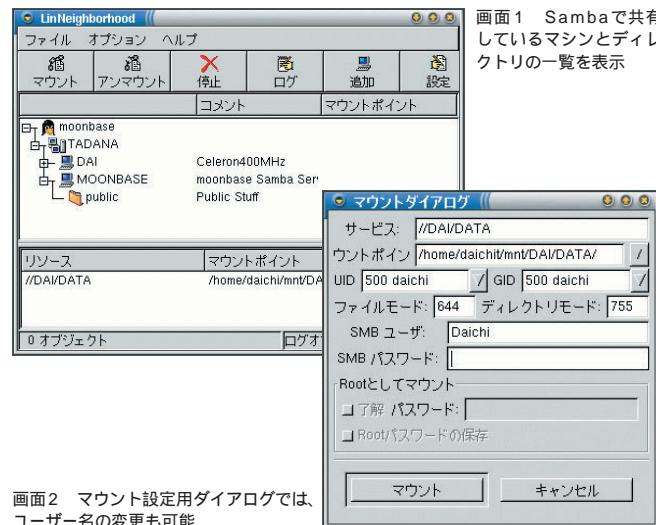
起動すると、マシンやディレクトリのツリー表示と、実際にマウントされたディレクトリのリストが表示される(画面1)。はじめて起動した場合は、設定ダイアログでワークグループ名を設定し、[Save]ボタンで設定をファイルに保存する必要がある。

ウィンドウ上のホスト名を右クリックし、[クイックブラウザ]を選択すると、ワークグループ名の下にLinuxマシンやWindowsマシンの一覧がツリー表示される。これらのマシン名をダブルクリックすると、共有ディレクトリやプリンタの一覧が表示される。

共有ディレクトリをダブルクリックすると、マウント設定用のダイアログが開く(画面2)。ここでは、マウントポイントやユーザー名の変更を行える。通常はそのまま[マウント]ボタンを押せばいい。マウント

に成功すると、マウントポイントを使って共有ディレクトリのファイルにアクセスできるようになる。

マウント時に任意のファイルマネージャを起動することも可能だ。設定ダイアログの[マウント後]ページで[マウント後にファイルマネージャを実行]をチェックしよう。初期設定ではxtermでMIDNIGHT COMMANDERが実行される。このほか、KDEやGNOME向けの設定も最初から用意されていて、リストから選択できるようになっている。なお、日本語化されたMIDNIGHT COMMANDERを使う場合は、xtermでは文字化けしてしまうのでrxvtに変更するとよい。



画面1 Sambaで共有しているマシンとディレクトリの一覧を表示

画面2 マウント設定用ダイアログでは、ユーザー名の変更も可能



DOS形式のフロッピーを簡単に扱えるファイルマネージャ

mfm

バージョン : 1.5

ライセンス : GPL

<http://www.core-coutainville.org/mfm/>

X GTK+ 日本語可

MS-DOSファイルシステムでフォーマットされたDOSやWindowsのフロッピーを、Linux上で簡単に扱えるファイルマネージャ。mountコマンドでいちいちフロッピーをマウントすることなく、フロッピー上のファイル一覧を参照し、ファイルのコピーや削除を行える。こうした機能は、mfmの内部でmtools (mdir/mcopy/mdelなどからなるツール群) を呼び出すことで実現しているため、実行にはmtoolsが必要となる。もっとも、ほとんどのディストリビューションにはmtoolsが最初から含まれているので新たにインストールする必要はないだろう。

起動すると、左右にファイルリスト表示ペインを持つウィンドウが開く(画面1)。最初はどちらもカレントディレクトリのファイルが一覧表示されている。ディレクトリを切り替えるには、ファイル一覧中のディレクトリをダブルクリックすればいい。フロッピーのファイル一覧を表示するには、MS-DOSファイルシステムのフロッピーをFDDに挿入した後、どちらか一方のペインの右上のリスト内容を「Hardisk」から「a:」に切り替える。

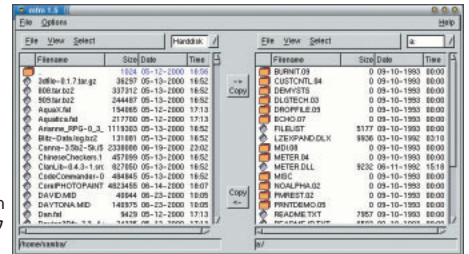
mtoolsはVFATにも対応しており、Windowsの長いファイル名もそのまま表示できる。また、ソースファイルmfm.cの“gtk_init (...)”の

前に“gtk_set_locale ();”という行を追加すると、Linuxファイルシステム上の日本語EUCのファイル名が表示されるようになる。ただし、MS-DOSファイルシステムの日本語ファイル名は、シフトJISで記録されているため文字化けしてしまう。

フロッピーとハードディスクの間でファイルのコピーを行う場合は、対象となるファイルをクリックで選択する。Shift - クリックによる範囲指定やCtrl - クリックによる複数選択、[Select] - [All]による全ファイル選択も可能だ。続いて、中央部の[]または[]ボタンを押すと、選択したファイルがまとめてコピーされる。

一方、ファイルの削除や名前の変更、印刷は、各ペインに用意されたメニューバーの[File]メニュー以下の項目か、右クリックメニューの項目から選択する。たとえば、ファイルを削除するには、それらを選択したあとでメニューの[File] - [Delete]を選択するか、右クリックメニューの[Delete]を選択すればいい。

なお、FDDが複数接続されていたり、mtoolsが/usr/bin以外にインストールされている場合は、[Options] - [Configure mfm] で設定ダイアログを開いて、FDDのドライブ名やmtoolsのインストール先ディレクトリの設定を変更する必要がある。



画面1 DOSやWindowsのフロッピーのファイル一覧が表示される

不要なファイルを検索して削除やアーカイブを行う

Kleandisk

バージョン : 1.1.1

ライセンス : GPL

<http://www.casema.net/buursink/kleandisk/>

X KDE Qt 日本語可

異常終了したソフトが作成するcoreなど、一般的には不要とされるファイルをディスク上から検索し、クリーンアップ処理を行うソフト。検索条件を設定したファイルグループを複数登録して、さまざまな種類のファイルを分類できる。単にファイルを削除するだけでなく、アーカイブしたり移動したりしておき、後で復活させることも可能だ。

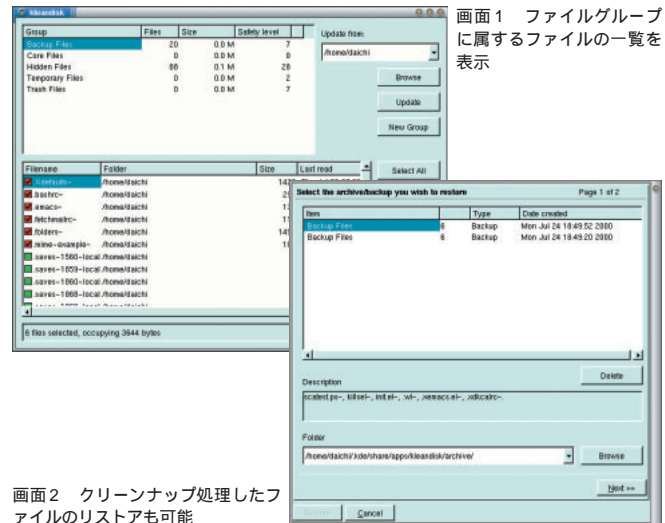
起動すると、クリーンアップとリストアのどちらを行うかを選択するメインウィンドウが開くので、[Clean Up] ページのボタンを押してクリーンアップ用のウィンドウを開こう(画面1)。最初から「Backup Files」や「Core Files」などいくつかのファイルグループが用意されている。ユーザーがファイルグループを追加したり、検索条件を変更することも可能だ。

検索を行うには、検索開始ディレクトリを設定して[Update] ボタンを押せばいい。このほか、指定した日時に自動的に検索を行うスケジュール機能も用意されている(ファイルグループ単位で設定可能)。いずれにせよ、検索開始ディレクトリ以下から、各ファイルグループに属するファイルが検索される。

検索後、それぞれのファイルグループをクリックすると、そのグルー

プに属するファイルやディレクトリのリストが一覧表示される。クリーンアップを行うには、リスト中のファイルにチェックを付けてから[Cleanup] ボタンを押せばいい。処理の種類(削除、アーカイブ、移動)や詳細な内容は、ウィザード形式で設定できる。

クリーンアップ処理したファイルは、バックアップしないで削除した場合を除いて復活可能だ。メインウィンドウの[Restore] ページのボタンを押し、リストア用ウィンドウを開こう(画面2)。過去にクリーンアップ処理したファイルのグループ名が一覧表示され、グループ単位やファイル単位でリストアすることができる。



画面1 ファイルグループに属するファイルの一覧を表示

画面2 クリーンアップ処理したファイルのリストアも可能

GUIのアーカイバ

GnoZip

バージョン: 0.1

ライセンス: GPL

<http://www.geocities.com/SiliconValley/9757/gnozip.html>

X

GTK+

GnoZIPは、GUIで動作するアーカイブ処理ツールだ。WindowsのWinZipに相当するツール。ただしGnoZipは、外部のコマンドラインで動作するアーカイバを、呼び出す形で動作する。tar、gzip、bzip2、lhaなど、一般的に使われるアーカイバについては対応しているが、外部ツールが存在しない場合は、もちろんそのアーカイブ形式は利用できない。

ほとんどの場合は問題ないはずだが、使う前にも、「Setting」 - 「Preferences」の「Executables」タブで指定されている、外部ツールのパスを確認しよう。

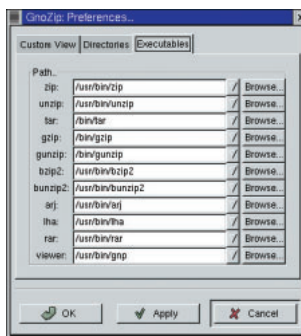
展開する場合は、「Open」でアーカイブファイルを開くか、ドラッグ&ドロップでアーカイブファイルをGnoZipに落とすと、格納されているファイルの一覧が表示される。ここで展開するなら「Extract」をクリックする。ただし、アーカイブ内の特定のファイルのみを展開するといった使い方は、まだできないようだ。展開する前に、中のファイルを見ることができるのは便利だし、コマンドラインを使うのが苦手であれば、有用なツールとなるだろう。

アーカイブの作成は、「New」でファイル名とアーカイブ形式を指定し、まず空のアーカイブファイルをつくり、「Add」で追加していく。もちろ

ん、「Open」で開いたアーカイブファイルに対しても、追加は可能だ。

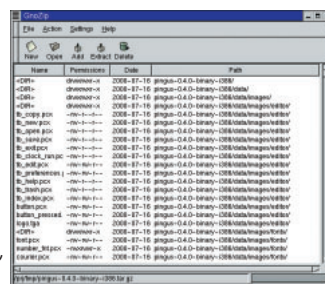
展開については、どのファイルも問題なく展開できたが、作成や追加は、若干動作が不安定なところもあった。作成したアーカイブは念のため再度開いて、中身を確認した方が良さそう。また、「Delete」や「Action」 - 「Test Archive」などは、メニューには存在していてもまだ動作しないようだ。もっとも、毎月のように新しい機能が備えられたバージョンが、Webサイトで公開されているようだ。まだバージョンも0.1.0ということなので、今後開発が続けば、さらに有用なツールになるだろう。

WebサイトにはRPMパッケージとソースのtarボールが用意されている。GTK+さえインストールされていれば、GNOME上でなくても動作した。ただし、KFMなどからのドラッグ&ドロップは効かないようだ。



画面1 必要な外部プログラムは、その存在とパスを確認しておく

画面2 GUIのウィンドウで中身を確認しながら展開できるのはやはり楽だ



RPMの作成も可能な統合アーカイバ

LnxZip

バージョン: 0.1

ライセンス: GPL

<http://visionary-hawk.webjump.com/lnxzip.html>

X

GTK+

GNOME

日本語可

tar / gzip / bzip2 / zip / arjなどでアーカイブ・圧縮されたファイルを統一的に扱うフロントエンド。見た目はWindows用の統合アーカイブソフト「WinZip」に似ており、アーカイブの形式を気にすることなく、ファイルの追加や削除、閲覧を行える。

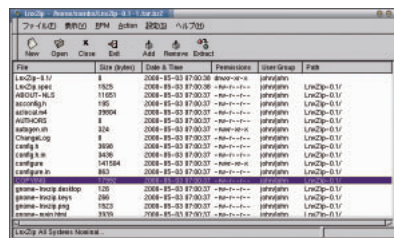
起動すると、ツールバーとファイルリストで構成されたウィンドウが開く(画面1)。既存のアーカイブを開く場合は、[Open]ボタンを押してダイアログからアーカイブを選択する。また、[New]ボタンでアーカイブを新規作成することも可能だ。いずれの場合も、ウィンドウにはアーカイブに含まれるファイルが一覧表示される。

ファイルの展開、追加、削除、閲覧などの操作には、ツールバーのボタンや右クリックメニューを利用する。たとえば、ファイルを選択してから右クリックメニューの[View]を選択すると、別ウィンドウにその内容が表示される。日本語EUCのテキストも表示可能だ。

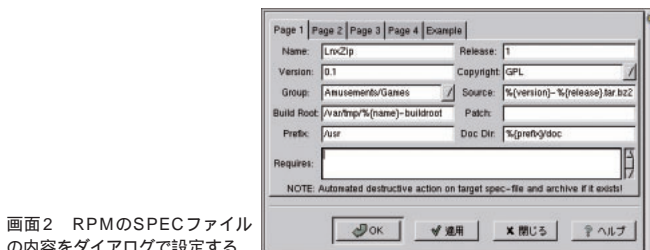
LnxZipの大きな特徴は、RPMパッケージの作成を支援する機能を持っていること(画面2)。まずは、RPMパッケージの核となる「SPECファイル」をダイアログで設定して作成する。続いて、指定したディレクトリ内のソースファイルをtar + bzip2でアーカイブ・圧縮し、最後に

rpmコマンドでソースとバイナリの各パッケージを作成する。こうした作業をGUIによる操作で進められるのだ。

なお、RPM作成機能を使う場合、あらかじめ以下の修正を行っておく必要がある。まず、設定ダイアログの[General]ページの設定を2か所修正する。1つめは[RPM]の内容で、「」を使わないフルパスに書きなおす(末尾には「/」が必要)。2つめは、[Auto Name Patch File]のチェックを外しておく。このほか、初回起動時にホームディレクトリに作成される「.rpmmacros」の最後の行「%_tmppath ...」の末尾に改行を追加する必要がある。



画面1 ウィンドウにはアーカイブ内のファイルが一覧表示される



画面2 RPMのSPECファイルの内容をダイアログで設定する



RPMやZIP/LHAも扱えるアーカイブマネージャ

TkZip

バージョン : 1.0.15

ライセンス : asis

<http://www.pcnet.com/proteus/TkZip/TkZip.html>

X Tcl/Tk 日本語可

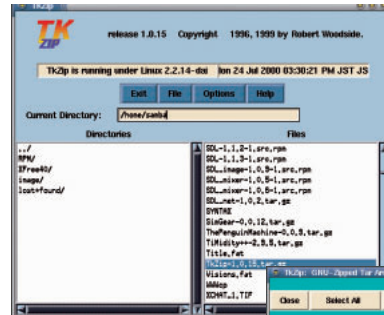
Tcl/Tkで書かれたアーカイブマネージャ。tar + gzipされたtarボールだけでなく、RPM形式のパッケージや、Windowsで使われるZIP、LHA、RAR形式のアーカイブにも対応している。いずれの形式の場合も、アーカイブ内のファイル一覧がウィンドウに表示され、ファイルの展開やビューアによる閲覧といった処理を統一されたGUIで操作できる。

起動すると、カレントディレクトリのサブディレクトリとファイル一覧が表示されたウィンドウが開く(画面1)。一覧中のアーカイブをダブルクリックすると、アーカイブ内のファイル一覧が別ウィンドウに表示される(画面2)。実際には、tarボールならtar / gzip、ZIPならzip / unzip、LHAならlhaといった具合にTkZip内部で対応するコマンドを実行している。もちろん、アーカイブにファイルを追加したり、新たなアーカイブを作成することも可能だ。

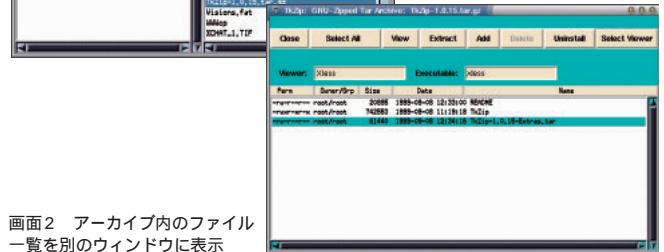
アーカイブ内のファイルの閲覧や展開を行うには、まず対象とするファイル名をクリックで選択し(複数選択や範囲選択も可)、閲覧なら[View]ボタン、展開なら[Extract]ボタンを押す。閲覧に使われるビューアは、[Select viewer]ボタンで変更できる。MIMEタイプの設定により、ファイル名をダブルクリックするだけで、拡張子に応じたコマ

ンドを起動することも可能だ。

[Options]以下のメニューにより、動作を細かくカスタマイズすることもできる。たとえば、ファイルの上書き時の警告を無効にするには、[Options] - [Warn of Overwrites]のチェックを外せばいい。逆に、上書きを完全に禁止するには[Options] - [Never Overwrite Existing Files]をチェックする。画面表示に関しては、[Options] - [Set Appearance]で開くダイアログで、メインウィンドウやアーカイブ形式ごとにウィンドウの背景色などを細かく設定できる。これらの設定は、[Options] - [Save Options]で設定ファイルに保存しておこう。



画面1 メインウィンドウはディレクトリとファイル一覧で構成



画面2 アーカイブ内のファイル一覧を別のウィンドウに表示

GUI操作で各種パッケージを管理

kpackage

バージョン : 1.3.10

ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

kpackageは、Slackwareのtgzパッケージ、Debianのdebパッケージ、Red HatのRPMパッケージ、BSDパッケージの管理を行うGUIインターフェイスだ。パッケージのインストール、アンインストールはもとより、現在インストールされているパッケージに含まれるファイルのリスト表示といった機能を備えている。パッケージをインストール、アンインストールする場合は、root権限でkpackageを起動する必要がある。

kpackageのウィンドウ構成は2ペイン方式になっている(画面1)。左側のペインにパッケージのツリー構成、右側のペインにパッケージやパッケージに含まれるファイルが表示される。ファイルマネージャなどによくみられるウィンドウ構成なので、通常のファイルを扱う感覚で操作できるのがうれしい。KDEドラッグ&ドロップにも対応しており、kfmなどのアプリケーションとの間でマウスを使ってパッケージを操作できる。たとえば、kfmで表示しているパッケージファイルをドラッグし、kpackageにドロップしてパッケージ情報を確認するといった操作が可能だ。

kpackage自体はインターフェイスなので、各パッケージを扱うためには、その種類ごとに、パッケージの処理を行うソフトウェアが別途必

要となる。以下に、まとめておくので参考にしてほしい。

Debianパッケージ

debパッケージをインストール/アンインストールするには、パッケージマネージャ「dpkg」が必要。

BSDパッケージ

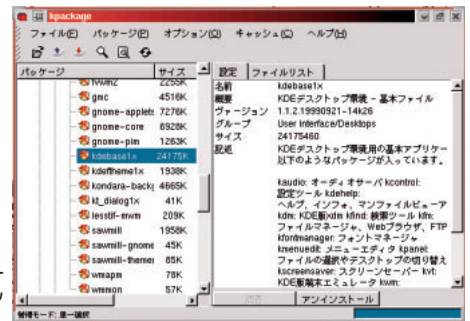
「pkg_info」、 「pkg_add」、 「pkg_delete」が必要。

Slackwareパッケージ

「installpkg」と「removepkg」が必要。

Red Hatパッケージ

「rpm」と「rpm-devel」が必要。RPMパッケージをkpackageで利用するには、RPMデータベースを初期化しておく必要がある。“rpm -qa”とタイプした際に“unable to open...”というエラーメッセージを受け取ったときは、“rpm --rebuilddb”を実行してRPMデータベースの再構築を行わなければならない。



画面1 ファイルマネージャのような感覚でパッケージを管理できる

あらゆるフォーマットに対応したパッケージビューア

KPackViewer

バージョン : 0.70

ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

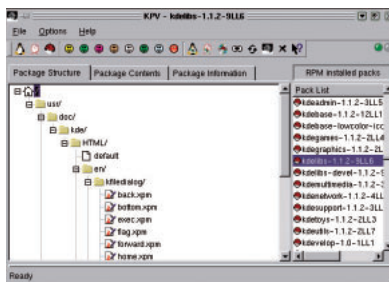
KPackViewerは、複数のファイルフォーマットに対応した高機能なパッケージビューアだ。zip、tar + gzip、tar + bzip2、LHA、gzip、ARJ、RARなど大変多くのフォーマットに対応している。また、Slackwareのtgz、Debianのdeb、Red Hat系のRPMといった各ディストリビューションのパッケージフォーマットにも対応しており、上記のアーカイブファイルと同じように扱える。

たとえば、tar + gzip形式のアーカイブの中身を閲覧したい場合は、[File]メニューから [Open TarGZip packages] を選択する。ファイルセレクタが開くので、目的のアーカイブを選択すれば、アーカイブに含まれるファイルがメインウィンドウの左側にリスト表示される。また、上部のタブで「ツリー表示」、「リスト表示」、「詳細情報」を切り替えることも可能だ。

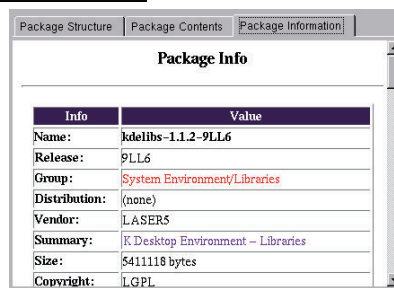
RPMパッケージや、debパッケージも同様の方法で閲覧できる。さらに、RPMパッケージなどのディストリビューションパッケージフォーマットでは、現在インストールされているパッケージの情報も閲覧することができる。その場合も、上部のタブで表示形式を切り替えられる。パッケージのグループ情報を閲覧したい場合は、右クリック - [Show

RPM Groups...] を選択すればいい。

KPackViewerはビューアだが、アーカイブの閲覧だけでなく、それを展開する機能も備えている。アーカイブを展開するには、右クリック - [Extract to...] を選択し、ファイル展開したい場所を指定する。ディストリビューションパッケージのインストールはサポートしていないが、同じくKDEのアプリケーションである「kpackage」とあわせて使用することで、柔軟なパッケージ管理が可能となるだろう。



画面1 KpackViewerのメインウィンドウ



画面2 RPMパッケージの情報も表示できる

シンプルで使いやすいスケジュール管理ソフト

korganizer

バージョン : 1.1.2

ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

korganizerはスケジュール管理ソフトだ。多くの人間でプロジェクトを進める場合などに、お互いの情報や進行状況を交換することができるため、さらに効率的に仕事をすすめることができるだろう。プロジェクトなどに関わっていない場合でも、このソフトは非常に役に立つ場面があるはずだ。

起動はKDEのスタートメニューまたはコマンドラインから行う。起動されると、Kパネルのタスクトレイエリアにドックアイコンが格納される。このアイコンは「alarm daemon」を表している。alarm daemonはメモリ上に常駐し、スケジュールの予定の時刻がくるとサウンドやアラート表示でこれを知らせる機能を持っている。

korganizerでは、会議、外出などの項目を「アポイントメント」と呼ぶ。新たにアポイントメントを登録するには、[アクション]メニューから [新規アポイントメント] を選択して、編集ウィンドウを開き、必要事項を入力していけばいい。このウィンドウでは、毎週、毎月といった決められた周期に応じた予定も記録できる。入力が終わったら、[保存し閉じる]メニューを選択してアポイントを登録する。また、ビューエリア上でダブルクリックすると、自動的に1時間分のアポイントメント

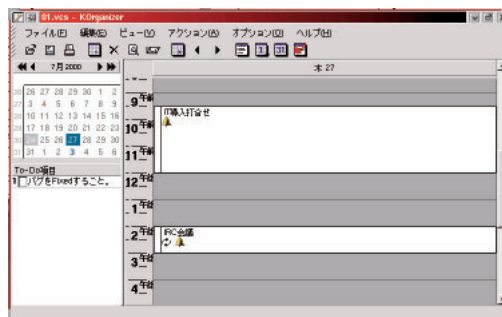
が挿入される。こちらの方法も覚えておくといいだろう。

内容を変更したいときは、メインウィンドウ右側の「ビューエリア」からアポイントメントをクリックし、[アクション]メニューから [アポイントメント編集] を選択すればいい。

アポイントメントエリアをドラッグし、左側の日付ナビゲータの別の日付にドロップすると、その日付にアポイントメントを移動することもできる。予定が入っている日は、左上の日付ナビゲータに太字で表示されるので一目でわかる。

多くのアポイントの中から特定のものを検索したい場合もあるだろう。[アクション]メニューの [検索] から開く検索ウィンドウで、検索が可能になっている。検索文字列にはワイルドカードも使用できる。

さまざまなkorganizerのオプションは、[オプション]メニューの [オプションの編集] で設定する。インターフェイスに関する設定から、ユーザー自身の個人情報まで、このウィンドウで編集できるので、いろいろ試してみよう。



画面1 スケジュール管理ソフトとしては標準的なインターフェイスを持っているため、初めてでもとまどうことはないだろう



スケジュール管理を行える日本語カレンダー

nittei

バージョン：1.80 ライセンス：GPL

<http://member.nifty.ne.jp/seto-yoneji/nittei.html>

X 日本語可

スケジュール管理を行えるカレンダーソフト。国産アプリなのでパッチなどを当てなくても最初から日本語が使える点がうれしい。スケジュールは分単位で設定でき、予定時刻（あるいは指定した時刻）になるとウィンドウが開いてユーザーに通知してくれる。また、指定時刻にプログラムを自動実行したり、毎日・毎週・毎月の繰り返しスケジュールを設定することも可能だ。今年から施行された祝日月曜化法による成人の日・体育の日の移動に対応しているほか、ユーザーによる休日や記念日の追加も可能となっている。

起動すると今月のカレンダーが表示され（画面1）、カレンダーの日付をクリックするとスケジュール入力用のウィンドウが開く。スケジュール1件につき1行を使用し、複数のスケジュールも入力できる。通知機能を使うには、予定時刻を「(時:分)」という形式で先頭に指定すればいい。通知をさかのぼる時間（分単位）を「(時:分)(分)」という形式で指定することも可能だ。予定時刻を記述しなかった場合、通知などのスケジュール管理からは外れるものの、入力した内容はそのまま保存されるので、日記として使うこともできる。

習い事や雑誌発行日のように、日/週/月単位で繰り返されるスケ

ジュールについては、専用のウィンドウが用意されており、時間や曜日を指定してスケジュールを入力するだけでいい（画面2）。このほか、年単位の誕生日や記念日などの場合は、休日に類するものとして、休日ファイル（nit_holiday）に記述を追加する。

このようにして設定したスケジュールは、1日ずつ表示するだけでなく、1か月分のスケジュールをまとめて表示したり、HTMLファイルとして出力してWebブラウザで閲覧や印刷を行うこともできる。また、キーワードを指定して、それを含むスケジュールを表示する検索機能も用意されている。



画面1 nitteiのカレンダー。もちろん日本語で表示される

画面2 毎週決まった曜日に繰り返されるスケジュールを設定

Xのルートウィンドウに常駐するカレンダー

X Desktop Calender (xdkcal)

バージョン：0.9c ライセンス：GPL

<http://www.shiratori.niec.tohoku.ac.jp/jir/linux/products/xdkcal/index-j.html>

X 日本語可

Xのルートウィンドウ（背景）に透明なカレンダーを貼りつけるソフト。年/月/日/曜日の表示色などを自由にカスタマイズできる。なお、週や月の名前はロケールデータベースから読み込まれるが、ほとんどのディストリビューションには対応する日本語の文字列データベースが用意されていないため、英語表記で表示される。

起動すると、Xのルートウィンドウの左上隅に、今月の小さなカレンダーが表示される。カレンダーの表示位置や使用する文字のフォント・色などの設定は起動時オプションで設定する。文字の色は、デフォルト、休日、土曜、今日、月名など細かく分けて設定可能だ。

たとえば、座標(512,16)にフォントセット「-alias-*」を使って、デフォルト色を黒として表示するなら、

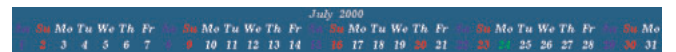
```
$ xdkcal -x512 -y16 -f '-alias-*' -C black &
```

とすればいい（画面1）。このほか、月や週の名前を表示しないようにするオプションや、横一列に日付を表示する（画面2）といったオプションも用意されている。

オプションで指定した内容は、そのまま設定ファイルに保存され、次回からは自動的に読み込まれる。フォントセットを細かく指定するなど、コマンドラインでいちいち記述するのが面倒な場合は、設定ファイル（.xdkcalrc）を直接エディタで書き換えてもいい。いったん好みの設定が決まったあとで、一時的に別の設定を試してみる場合には、オプションの内容を設定ファイルに保存しないようにする-Nオプションをあわせて指定するとよいだろう。



画面1 Xのルートウィンドウにカレンダーが表示される



画面2 横一列に日付を表示するスタイルも用意されている

GNOMEのパネルにメモを貼りつける多機能ソフト

MemoPanel

バージョン：6.5

ライセンス：GPL

<http://www.geocities.co.jp/NeverLand/1645/memopanel2.html>

X GTK+ GNOME 日本語可

GNOME環境で使われる「パネル」にメモを貼りつけるアプレット。日本で作成された国際化アプリで、そのまま日本語の文章をメモできる(画面1)。時計やヘッドラインニュース表示、メールチェッカーとしても使える多機能ソフトだ。

起動すると、起動した日時が表示されたメモがパネルに表示される。同時に複数のMemoPanelを利用することも可能だ。内容を変更するには、メモ上で右クリックして[プロパティ]を選択し、メモの内容のほかさまざまな項目を設定するダイアログを開く(画面2)。設定内容は自動的に保存され、次にGNOMEを起動した際には、同じ位置に同じ内容のメモが表示される。

複数行にわたるメモを書くには、改行したい位置に“%n”を付けられればよい。このほか、現在の日付で置きかえられる“%d”など、dateコマンドと同じ形式の文字列(strftime)を利用できる。単にメモを書いた日時を簡単に入力できるだけでなく、指定秒数ごとに表示を更新する「ディレイ」機能と併用すれば、そのままMemoPanelを時計アプレットとして使うこともできる。

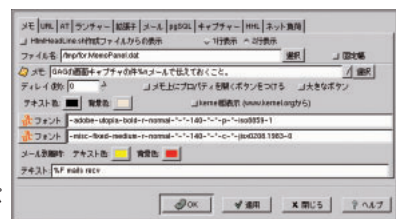
ヘッドラインニュースの表示は、同じ作者によるシェルスクリプト

「HtmlHeadLine.sh」と組み合わせることで可能になる。HtmlHeadLine.shがfreshmeatやslashdotなどのニュースサイトから取得した情報をファイルに保存し、MemoPanelがそのファイルの内容を1、2行ずつ読み進めて、順番にメモとして表示する。

また、IMAP / POP3対応のメールチェック機能を有効にすると、一定時間ごとに新着メールをチェックし、もしあればその数をメモに表示してメールクライアントを起動してくれる。このほか、URL呼び出しやアラーム機能、ランチャ、画面キャプチャ、ネットワーク負荷表示などさまざまな機能が用意されている。



画面1 GNOMEのパネルにメモが表示される。日本語も使用可能



画面2 メモの内容などはプロパティダイアログで設定する

Webページのリンク切れをチェック

LinBot

バージョン：1.0

ライセンス：GPL

<http://starship.python.net/crew/marduk/linbot/>

コンソール Python

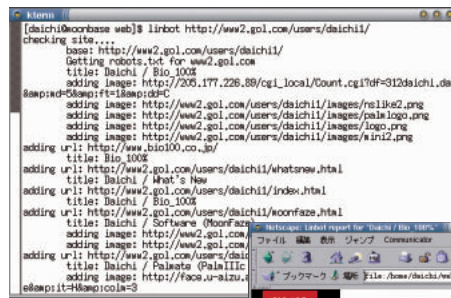
Webページのリンク切れなどを調べるコンソールベースのツール。自動的にリンクをたどって複数のWebページを調べ、それぞれのリンク先のページが存在するか、作成日時が古いものはないか、受信速度が遅いものはないかなどを調べて報告してくれる。チェック結果はHTMLファイルに出力され、Webブラウザで閲覧可能だ。

インストールの際は、LinBot本体(linbot.py)へのシンボリックリンクを/usr/local/binに作成するとよいだろう(詳細はINSTALLを参照)。なお、LinBotのWebサイトのURLが変更されたため、config.pyの128行目と140行目の“skyport”を“python”に修正する必要がある。これを忘れると、チェック結果の表示に使われるスタイルシートを読み込めず、結果の表示が汚くなってしまうので注意されたい。

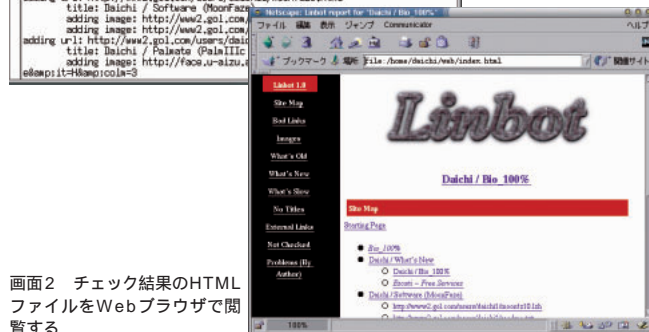
使い方は簡単で、コマンドラインでチェックするURLを指定してLinBotを起動すればいい。たとえば、“linbot http://www.hoge.com/hoge/”とすると、LinBotが自動的にそのページのリンクを検出し、リンク先のページが存在や作成日時、受信速度などを調べてくれる(画面1)。リンクをたどる数や、外部リンクのチェックの有無、外部リンクとみなす文字列パターン、チェック結果の保存先などはすべて

コマンドラインオプションで指定する。

チェック結果は複数のHTMLファイルとして出力されるので、NetscapeなどのWebブラウザを利用して、リンク切れ、古いページ、新しいページ、遅いページなどの状況を確認できる(画面2)。たとえば、Webサーバとして運用しているマシンでcronシステムにより定期的にLinBotを起動し、チェック結果をWebサイト上に保存するように設定しておけば、定期的にチェックが行われ、結果を簡単に閲覧できるわけだ。



画面1 Webページのリンク切れを自動的にチェックしてくれる



画面2 チェック結果のHTMLファイルをWebブラウザで閲覧する



Webページ更新チェッカー

KWebWatch

バージョン: 0.71 ライセンス: GPL

<http://www.kde.org/>

X KDE Qt

更新されたURLをチェックするためのアプリケーション。定期的（あるいは不定期）に更新されるURLを管理するのに大いに役立つ。

KWebWatchはインストールすると、KDEのスタートメニューの[インターネット]サブメニューに登録されるので、そこから起動するか、あるいはコマンドラインから起動する。

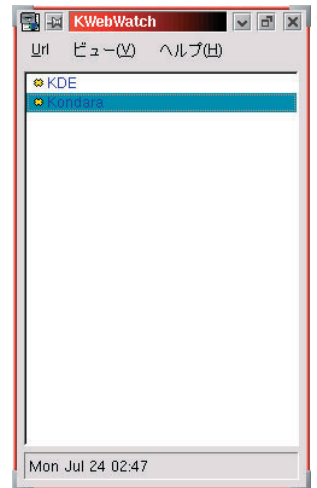
URLの更新をチェックするには、まず、そのURLをデータベースに登録する必要がある。[Uri]メニューの[新規]を選択すると、新しいURLを登録するためのウィンドウが開く。そこに、URL、URL名、チェック間隔などを入力し、[はい]ボタンをクリックすると登録が完了する。ここでいう「URL名」というのは、リストにこのURLが表示される時の名前になるので、わかりやすい名前にしておこう。

URLのチェックは、自動的でも手動的でも可能だ。KWebWatchは指定された間隔が経過すると自動的にアドレスをチェックするが、[Uri]メニューから[今すぐチェック]を選択することで、時間に関係なくチェックすることができる。初めて、そのURLをチェックしたあとは、必ず「更新されている」という印が表示される。次回のチェックの際に、更新されていない場合は、緑色のボールが名前の先頭に付き、途中でエ

ラーが発生してチェックできなかった場合には赤いボールが付く。

KWebWatchでは、いくつかの方法でチェックの精度を調整できる。たとえば、広告バナーがアクセスのたびに切り換わるようなサイトでは、通常のチェック方法では毎回更新されたと解釈してしまふ。このようなときは、[Uri]メニューの[設定]で表示されるダイアログで、[最適推測する時にタグを無視]をチェックすることで正確さを保持できる場合もある。

Kパネルのタスクトレイエリアに格納することも可能で、ドック化するには[ビュー]メニューの[パネルにドック化]を選択すればいい。パネルに格納されている間もKWebWatchはURLのチェックを続ける。その際に、もしURLの更新を見つけた場合は、ドックアイコンの色を茶色から緑に変更し、更新があったことをユーザーに通知する。更新があった場合に、ドックアイコンを左クリックすると更新されている可能性のあるURLのリストが表示され、その中の一つをクリックすると、ブラウザが起動する。デフォルトのブラウザにはkfmが設定されているが、これはNatscapなどほかのブラウザに変更可能だ。



画面1 KDEらしいシンプルなお顔が特徴だ

Webサイトのページをまとめて取得

WWWcp

バージョン: 1.96 ライセンス: フリー

<http://www.ff.iij4u.or.jp/rewsirou/WWWcp/WWWcp.html>

コンソール 日本語可

HTTPプロトコルを利用してWebページを自動巡回するPerlスクリプト。コマンドラインで指定したURLのファイルを取得し、そこに含まれるリンクを再帰的にたどって、そのサイトのページや画像をすべて取得する。パスワード付きのページやCGIのページの取得も可能だ。このほか、取得する（あるいはしない）ファイルのパターン指定や、ファイルの途中から読み出すレジューム機能を備えている。

コマンドラインにURLと保存先ディレクトリを指定して、

```
$ WWWcp http://www.hoge.com/~hoge/ hoge/
```

とすると、「http://www.hoge.com/hoge/」で取得できるHTMLファイルや、その中に記述されている画像、リンク先のHTMLファイルなどを再帰的に取得して、ローカルなディレクトリhoge以下にツリー構造をなすように格納する（画面1）。初期設定では、指定したURLのファイル名の直前までを「基準パス」とし、それを含まないリンク先は無視するため、外部サイトへのリンクはたどらない（-oオプションで変更可）。また、すでに保存先ディレクトリに同名のファイルが存在する場合は、

更新されているファイルだけを取得する。

ファイルを1つだけ取得するには、-1オプションを付け、URLとファイル名を記述すればいい。たとえば、

```
$ WWWcp -1 http://www.hoge.com/~hoge/ hoge.tar.gz
```

とすると、hoge.tar.gzのみ取得する。

また、転送途中に接続が切断された場合は、-rオプションを追加して、

```
$ WWWcp -1r http://www.hoge.com/~hoge/ hoge.tar.gz
```

とすると、レジューム機能が働いて、中断した部分から転送が再開される。URLを1行に1つずつ列挙したファイルを利用してファイルを受信することも可能だ。

このほか、起動時のオプション指定により、取得するファイルや対象外とするファイルのパターン指定、漢字コードの変換、クッキーの取得・送付、古いファイルを上書きするかどうか、といった設定を行える。よく使うオプションは、ホームディレクトリのWWWcprcに1行1引数形式で書いておくといい。



画面1 指定したURL以下のページを再帰的に取得する

リモートのメールサーバからメールを一括転送

fetchmail

バージョン : 5.4.3

ライセンス : GPL

<http://www.tuxedo.org/esr/fetchmail/>

コンソール

リモートのメールサーバからPOP3 / IMAP4プロトコルなどを使ってメールを取得し、ローカルのメールスプールや別のメールサーバに転送してくれるツール。PPP接続のユーザーがオフラインでメールを読んだり、複数のメールサーバのメールを1つのサーバに集めたりする場合などに使える。設定によりサーバにメールを残しておくことも可能だ。

メールサーバ名、プロトコル、サーバ用のユーザー名とパスワード、メールの送付方法などは、ホームディレクトリの設定ファイル(.fetchmailrc)に記述するか、起動時のオプションで指定する。なお、平文のパスワードが含まれるため、設定ファイルは他人に見られないように属性を「600」に変更しておこう。

たとえば、POP3サーバから取得したメールをローカルのメールスプールに転送する場合、

```
poll POPサーバ名 proto POP3
username POPユーザー名 password POPパスワード
no mimedecode
no rewrite
```

多機能で使いやすいPPP接続ツール

PPxP

バージョン : 0.99120923

ライセンス : フリー

<http://www.dsl.gr.jp/manabe/PPxP/>

コンソール

日本語

国産のPPP接続ツール。PPPをインストールしたppxpdデーモンと、インターフェイスを受け持つコンソールが独立しているのが特徴だ。通信が必要になった時点でPPP接続を確立するオンデマンド接続や、複数のローカルホストで同時に利用できるIPマスカレードなど数多くの機能を備えている。動作にはトンネルデバイスのuserlinkが必要だ。

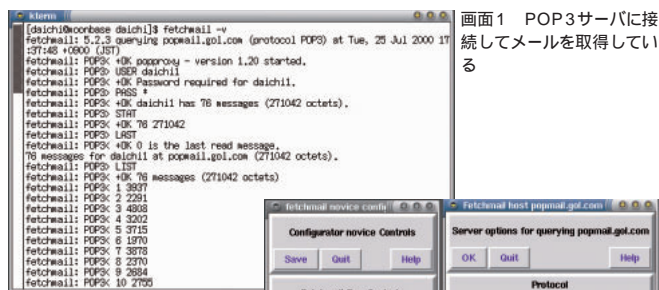
標準コンソールであるppxpコマンドは、「ppxp>」というプロンプトにset、connectなどのコマンドを入力することによって設定や接続処理を行う(スクリプトによる一括処理も可能)。この方法は、初心者には敷居が高いため、対話的に設定や接続を行えるクイックダイアルアップ機能が用意されている。

「ppxp -C qdial」として起動するか、「ppxp>」プロンプトでqdialコマンドを実行すると、クイックダイアルアップの画面に切り替わる(画面1)。電話番号(複数指定可)やPPP接続用のログインユーザー名・パスワードなどを設定しよう。[ダイヤル]を選択すると、現在の設定でPPP接続が行われる。正しく接続できるのなら、[保存]を選択して現在の設定をスクリプトファイルに保存しておこう。次回からは起動時に「ppxp スクリプトファイル名」とするだけで自動的に設定される

```
mda "/bin/mail %T"
```

と設定ファイルに記述すればいい。別のサーバにメールを転送する場合は、sendmailなどのMTAの設定が別途必要になる。

fetchmailを実行すると、実際にサーバに接続してメールを取得する(画面1)。通常は処理が終わるとfetchmailは終了するが、バックグラウンドで動作する「デーモンモード」も用意されており、cronシステムを使うことなく定期的なメールの取得が可能だ。設定ファイルの記述が面倒な人は、GUIで設定ファイルを作成できるフロントエンドを利用するとよいだろう。Pythonで書かれたfetchmailconf(画面2)などが作られている。



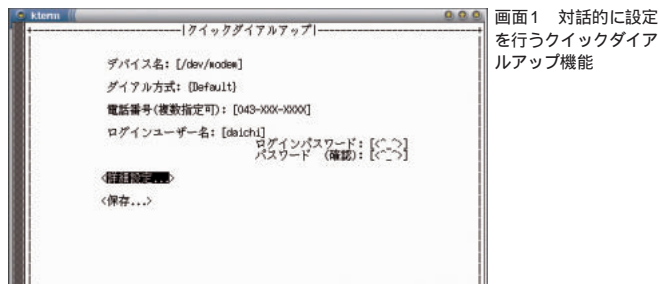
画面1 POP3サーバに接続してメールを取得している



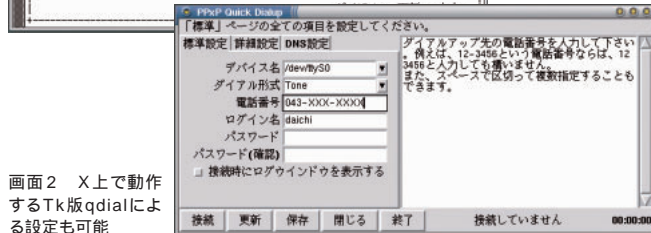
画面2 GUIで設定ファイルを作成できるfetchmailconf

ので、connectコマンドで接続するだけでいい(切断はdisconnectコマンドで行う)。

PPxPのtarボールには、Xpm版のxppxp、Tk版のtkPPxP/qdial、forms版のfppxpなどのグラフィカルなコンソールが付属する。また、GTK+版のGTK-ppxpや、GNOMEのパネル上で動作するPPxPアプレットなども別サイトで配布されている。たいいていのコンソールでは、設定用のスクリプトファイルが実行に必須となる。ppxpコマンドのクイックダイアルアップ機能を利用するか、Tk版のqdial(画面2)を利用してスクリプトファイルを作成しておくといいい。



画面1 対話的に設定を行うクイックダイアルアップ機能



画面2 X上で動作するTk版qdialによる設定も可能



テーマ機能を持ったターミナル

Eterm

バージョン : 0.9

ライセンス : GPL

<http://www.eterm.org/>

X

日本語可

Etermはrxvtをベースに、Enlightenment向けにつくられたターミナルプログラムだ。もちろんfvwm、AfterStepやKDEなどのウィンドウマネージャ (kwm) 上でも動作するが、すべての機能を利用するにはEnlightenmentを使う必要がある。Etermは独自のテーマ機能を持っており、画像や色などの外観を設定することができる。

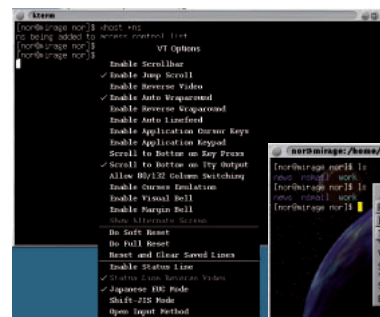
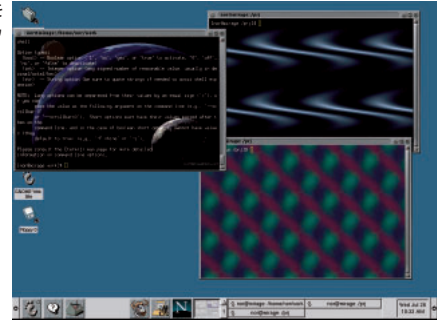
ktermと異なり、Ctrlとマウスの中ボタンでは、「スクロールバーの表示 / 非表示」、Ctrlと右ボタンで「メニューボタンの表示 / 非表示」が切り替えられる。もちろん、テーマに設定されていない好きな画像や色を選ぶこともできるし、背景を透明にすることも可能だ。

Webサイトではtarボールのソースが配布されているので、自分でビルドする必要がある。ビルドにはEnlightenmentで利用しているImlibが必要だ。また、日本語を使うにはkanjiオプションをつけてコンパイルし、使用するフォントを指定しなければならない。Etermもかなりメジャーなターミナルソフトなので、各ディストリビューションにパッケージが入っている場合も多い。最新版が必要でなければ、そちらを使うほうが無難だろう。

現在公開されているのは、開発版の0.9系列と安定版の0.8.10である。

0.9ではテーマのフォーマットの変更、Enlightenmentと自動的に同期を取るAuto modeなどが追加された。

画面1 にぎやかなEtermをたくさん開いたデスクトップ。



画面2 ktermと比較すると、Etermはメニューそのものも派手派手だ。

気軽に使えるパーソナルデータベース

Gaby

バージョン : 1.9.22

ライセンス : GPL

<http://gaby.netpedia.net/>

コンソール

GTK+

GNOME

日本語可

住所録やCD目録などを気軽に管理できるパーソナルデータベースソフト。データベースサーバを使うほどではない小規模なデータの管理に役立つ。日本語のデータも問題なく入力・編集可能だ。データベースの構造を定義している記述ファイルを書きかえることで、項目の表示順の変更や新たな項目の追加を柔軟に行える。また、プラグインによる機能拡張にも対応している。

なお、バージョン1.9.22のconfigure.inには不具合があり、そのままではGabyを作成できない。configure.inの106行目の“gui=false”を“gui=true”に修正してからビルドしよう。

日本語表示でGabyを利用するには、“LANGUAGE=ja gaby”として起動する。使い方のヒントなどが表示される [Tip of the day] ウィンドウとともに、住所録 (フォーム) を含んだメインウィンドウと、電話帳 (拡張リスト) のサブウィンドウが開く。

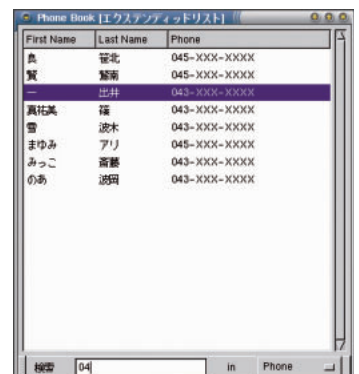
Gabyではデータの入力や閲覧を各種のビューで行う。基本的なビューは、データの入力・編集に適した「フォーム」と、多数のレコードを同時に閲覧できる「リスト」の2つ。このほかにも、インクリメンタル検索機能を持つ「拡張リスト」や、プラグインで実現される「キャン

パス」などのビューが用意されている。

まずは、住所録 (フォーム) にデータを入力してみよう (画面1)。ツールバーの [新規] ボタンを押すとデータベースにレコードが登録され、次のレコードの入力に切り替わる。一方、電話帳 (拡張リスト) には、住所録で入力したデータのうち、姓名と電話番号だけが一覧表示される。住所録でデータを登録・修正すると電話帳のリストも自動的に更新されるし、リストのレコードをクリックするとその内容が住所録に表示される。インクリメンタルサーチでリストを絞り込むことも可能だ。



画面1 メインウィンドウには住所録のフォームが表示される



画面2 電話帳 (拡張リスト) はインクリメンタルサーチ機能を持つ

英単語などの語彙学習用の簡易データベース

Joe's QVocabulary (QVocab)

バージョン: 0.22.3

ライセンス: GPL

<http://www.qvocab.seul.org/>

X Qt KDE 日本語可

英単語 / 熟語とその意味といった対になる言葉を学習するための簡易データベース。単語の入力や修正、検索、テストなどの機能を持つ。それぞれの単語はテストの結果によって5段階に分類されるため、不得手な単語だけを集中して学習することも可能だ。日本語対応のKDE上で使えば、修正なしで日本語を扱える。

最初に、単語を登録するためのデータベースを新規作成しよう。データベースは複数作成でき、実行中に切り替えられる。一覧表示されたデータベース名をダブルクリックすると、そのデータベースが読み込まれる。なお、データベースの内部は、「First」～「Fifth」の5ファイルに分かれており、登録した単語はテストの結果により自動的にこれらのファイル間を移動する。

データベースに単語を登録するには、[Input] ボタンを押して入力用ウィンドウに切り替え、[1st/2nd language] の両フィールドに、対になる言葉（たとえば、英単語 / 熟語とその訳語など）を入力すればいい。どちらのフィールドにも、複数の単語をカンマで区切って並べられる。一方、すでに登録したレコードの編集や検索は、[Edit words] ボタンで開く編集用ウィンドウで行う。このウィンドウには、入力時の設

定内容や、過去の質問数・不正解数などの情報が表示され、それぞれの内容を修正できるようになっている。単語の検索も可能だ。

ある程度単語を登録したら、テストを実行しよう。[Ask words] ボタンを押すと、テスト用ウィンドウに切り替わる（画面2）。最初に、上部の3つのボタンで質問と答えに使われる内容を指定する。すると、その下の2つのフィールドの一方に質問が表示されるので、もう一方にキーボードを使って答えを入力すればいい。キー入力の代わりに、カンマで区切ったデータの内容を分割して、質問または答えに利用するというような設定にもできる。



XMLを利用した音楽メディア用データベース

M3X

バージョン: 0.1.12

ライセンス: GPL

<http://madhouse.lonyay.edu.hu/m3x/>

X GTK+ GNOME 日本語可

音楽CDなどの音楽メディア用の軽量データベースソフト。アルバムのタイトルやアーティスト名、各トラックの曲名などをGUIにより登録できる。日本語のデータも問題なく利用可能だ。SQLサーバなどは必要なく、データはXML形式で保存される。

起動すると、データベースのウィンドウが開く（画面1）。まずは[Add] ボタンを押して、新しいエントリを設定しよう。アーティスト名、タイトル、コメント、ジャンル、リリース年などの項目が用意されており、日本語のデータも設定できる（画面2）。トラックについては、[Add] ボタンで開くダイアログを使って曲名や長さを登録する。

このほか、コンソールスペースでデータを操作する「m3x-text」も用意されており、[ファイル] - [Launch m3x-text] で起動できる（画面3）。m3x-textでは、load、ls、cd、getといったコマンドを使ってデータベースの読み込みやエントリの表示が可能だ。ただし、日本語を含むデータを登録している場合、内蔵ターミナルでは文字化けしてしまう。日本語を表示するには、設定ダイアログの[M3X-Text Preference] ページで外部ターミナルとしてktermを利用するように設定すればいい。





日本生まれの電子辞書検索ソフト

KEBook

バージョン : 0.3.3 ライセンス : GPL

http://www.os.rim.or.jp/s_mitu/library/mysofts.html

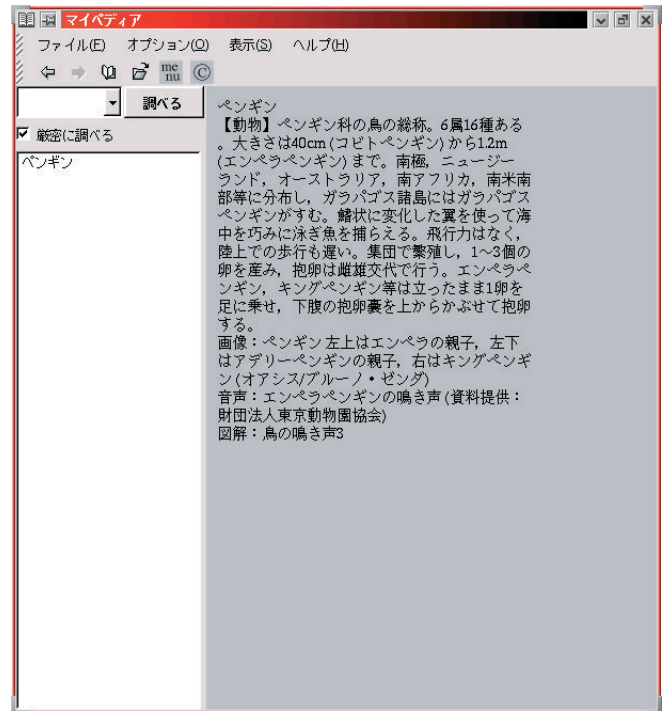
X KDE Qt

KEBookは、単語を電子辞書で調べるための国産KDEアプリケーションだ。日本KDEユーザ会のホームページによると、日本初のKDEアプリケーションであるようだ。EB、EBG、EBXA、EBXA-C、S-EBXA、EPWING形式の電子辞書に対応している。なお、動作には別途EBライブラリが必要となる。EBライブラリは、CD-ROMブックにアクセスするためのライブラリだ。日本語版の公式ホームページ (<http://www.sra.co.jp/people/m-kasahr/eb/index-ja.html>) も開設されているので、一度見ておくといいだろう。

KEBookを使用するためには、まずはじめに、辞書の入ったディレクトリと辞書を指定しなければならない。[オプション] メニューから [辞書の入ったディレクトリを選ぶ] を選択すると、ディレクトリセレクトウィンドウが表示される。そのディレクトリに複数の辞書が含まれている場合、[オプション] メニューの [辞書を選ぶ] を選択し、使用したい辞書を指定する。

単語を調べるには、メインウィンドウの左上にあるコンボボックスに単語を入力して [調べる] ボタンをクリックすればいい。検索結果は、ウィンドウの右側のパネルに表示される。複数の候補が見つかった場合

は、左下のリストボックスに一覧表示されるので、そこから該当するものを選択しよう。



画面1 国内で広く普及している電子辞書が、KDE上でも利用可能に

po / potファイル編集ツール

Ktranslator

バージョン : 0.6.1 ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

ローケルデータであるpo / potファイルを編集するKDEユーティリティ。Ktranslatorを使用することにより、メッセージカタログの翻訳作業を統一的に、しかも効率よく処理できる。

ktranslatorを使用するためには、あらかじめいくつかの設定を行う必要がある。[オプション] メニューから [一般設定] を選択し、設定画面を起動しよう。設定項目は以下のとおりだ。

ファイル

kde-i18n CVSモジュールのコピーのフルパスを指定する。カタログマネージャを使用するためにはこの指定が必要となる。

個人情報

ユーザーの個人的なデータを設定する。この情報は「オートヘッダ」機能で使用される。

翻訳

このセクションでは、スクロールに関するいくつかのオプションを確認することができる(たとえば、[次のメッセージ] または [前のメッセージ] ボタンを押す条件など)。

検索

検索機能に関するオプションを指定する。

ヘッダ

poヘッダ用のテンプレートを指定できる。

設定が完了したら翻訳作業を行えるようになる。新規にpoファイルを翻訳する (msgstrを翻訳する) には、翻訳したいpoファイルを右下のエディタに置く。

古いバージョンのメッセージカタログ、またはpoファイルを使って翻訳作業をする場合は、画面右下に新しいlmsfstrを置き (NewStr)、画面左下に古いバージョンを置く (OldStr)。それから、[前のメッセージ] (または [次のメッセージ]) ボタンを押すことで、以前の (改定が必要とされる) メッセージに進む事ができる。[->] - [古いのを新しいのにコピーして前に行く] と [->] - [古いのをコピーして次へ行く] では、古い文字列OldStr (左下) の内容をNewStrエディタにコピーして、その後前あるいは次のメッセージに進む事ができる。[.poファイルから翻訳を検索] または [.moファイルから翻訳を検索] を使って、古い翻訳を検索することも可能。Kdevelopから起動して利用すれば、効率よくアプリケーション開発を進められるだろう。



画面1 ローケルデータの作成が効率よく行える。古いバージョンと比較しながらの更新にも役立つ

OpenGLを利用したリアルな3D時計

glclock

バージョン: 6.0Beta6.0

ライセンス: フリー

<http://www.daionet.gr.jp/masa/glclock/index.html>

X

Mesa

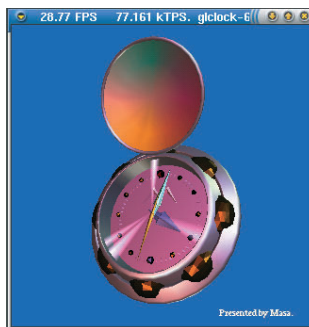
OpenGL (あるいは互換ライブラリのMesa) とGLUTを利用して、リアルな懐中時計を表示するソフト。美しい時計を手元に置いて、さまざまな角度から眺めて楽しむことができる。起動時オプションの指定により、ソフトウェアによるシミュレーションでも快適に表示されるシンプルな時計から、3Dアクセラレータが必須のゴージャスなものまでバリエーションに富んだ時計を表示可能だ。

起動すると、フタ付きの円形の時計が表示される(画面1)。起動オプションで時計の形や質感を変えられるが、いちいち指定するのが面倒なら同梱のシェルスクリプトを利用しよう。半透明の「crystalclock」(画面2)や大理石 & 木目調の「mwclock」などのバリエーションを手軽に楽しめる。なお、末尾に「_L」が付いたシェルスクリプトは、外形を六角形にして処理を軽くしたものだ。パフォーマンスの低いマシンでは、こちらのシリーズを使うとよいだろう。

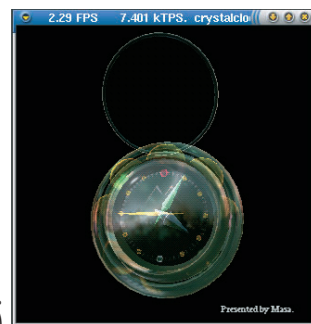
ウィンドウ上でマウスをドラッグすることにより、時計の移動・回転・拡大やフタの開閉が可能だ。また、特定のキー操作により、時計を照らすライトの数や種類などを変更したり、モーションブラー、アンチエイリアシング、被写界深度ブラー、フォグなどの特殊効果をオン/

オフしたりできる。特殊効果をオンにすると、3Dアクセラレータなしのマシンでは目に見えて表示速度が遅くなるので気をつけよう。

このほか、「-SA」オプションを付けて起動することで、大きなウィンドウ中を時計が回転しつつ移動するスクリーンセーバー的な使い方も可能だ。バージョン6.0からは、ベンチマークソフトの「glclock Mark」が独立した実行ファイルとして用意され、26項目のさまざまな設定で時計を表示してマシンのOpenGL性能を測定し、テキストログやHTML形式のグラフで結果を表示してくれる。Webサイトに用意されたデータベースとあなたのマシンの結果を比較してみよう。



画面1 マウス操作により回転・移動やフタの開閉が可能だ



画面2 半透明のcrystalclock、3Dアクセラレータなしではつらい

CPU / ビデオ / ディスクのベンチマークを計測

HDBENCH clone

バージョン: 0.14.0

ライセンス: GPL

<http://www.enjoy.ne.jp/gm/program/hdbench/index-ja.html>

X

GTK+

日本語可

有名なWindows用国産ベンチマークソフト「HDBENCH」にそっくりのインターフェイスを持つ、UNIX系OS用の国産ベンチマークソフト。グラフ表示など一部実装されていない機能があるものの、基本的な使い方はHDBENCHと同じで、簡単に各種のベンチマークを計測できる。

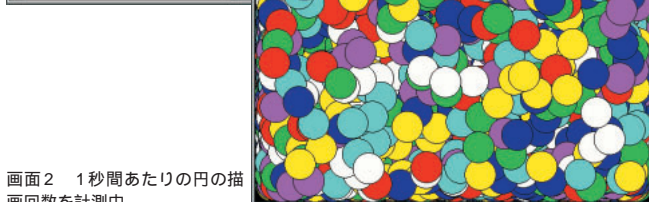
起動すると、いくつかのボタンが並んだウィンドウが開く(画面1)。HDBENCH cloneでは、大きく分けてCPU、ビデオ、ディスクの3種類のベンチマークを計測できる。これらは、[CPU] [VIDEO] [DISK] ボタンでそれぞれ個別に計測することもできるし、[ALL] ボタンでまとめて順番に計測することも可能だ。

CPUに関しては、1秒間あたりの浮動小数点演算および整数演算の計算回数、メモリ転送速度を計測する。ビデオでは、1秒間あたりの矩形・円・テキストの描画回数(画面2)、3秒間のスクロールでの描画回数、5秒間のイメージの描画回数を計測する。ただし、イメージについては65536色(色深度16ビット)モードでしか計測できない。ディスクについては、1秒間の書き込み/読み込み速度を計測する。計測時に使用するドライブ(マウント後のディレクトリ)とファイルサイズ(1Mバイト~2Gバイト)はユーザーが自由に変更できる。

計測結果は、[コピー] ボタンでクリップボードにコピーしたり、[印刷] ボタンで印刷して保存できる。また、[システム] ボタンを押すと、CPUの種類やメモリ容量などの情報が別ウィンドウに表示される。なお、本家HDBENCHの特徴のひとつである、「複数の計測結果を棒グラフを使って比較する」機能は、現時点では実装されていない。自分のマシンの相対的な性能を知りたい人は、HDBENCH cloneのWebサイトにさまざまなマシンの計測結果をまとめたデータベースが用意されているので、そちらを参照するとよいだろう。



画面1 ボタンを押すだけで簡単にベンチマークを計測できる



画面2 1秒間あたりの円の描画回数を計測中



強力なファイル検索をGUIで行う

gtkfind

バージョン: 1.1 (日本語版) ライセンス: GPL

<http://www.oz.net/~mattg/download.html>
<http://www.yk.rim.or.jp/kawagisi/gtkfind-jp.html> (日本語版)

X GTK+ 日本語可

ファイル名、日時、属性、サイズなど、さまざまな条件によるファイル検索を行うソフト。コマンドラインで複雑なオプションを指定しなければならないfindコマンドに比べ、検索条件をGUIで設定できるので、初心者でも複雑な検索を行える。検索結果は専用のウィンドウに表示するだけでなく、標準出力に出力して他のツールと組み合わせることも可能だ。ウィンドウの表示や検索条件で日本語を使うためのパッチが別途配布されている。

gtkfindを単独で使用する場合は、ktermなどのコマンドラインで“gtkfind &”として起動する。一方、gtkfindをfindコマンドの代わりとして使う場合には、“gtkfind -vanish | sort”のように、「|」(パイプ)を使って他のコマンドと接続し、検索結果を標準出力へ出力するよう設定すればいい。なお、“-vanish”は、検索後にgtkfindを自動的に終了するオプションだ。

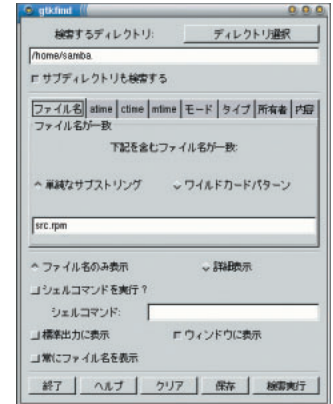
ウィンドウには、上部に検索ディレクトリ、中央部にさまざまな検索条件を設定する「カード」、下部には検索後の処理の選択や検索を開始するボタンなどが並んでいる(画面1)。カードは、ファイル名に含まれる文字列やアクセス日時、属性、含まれる文字列など8種類用意されて

おり、複数のカードで条件を設定すると、それらをすべて満たすファイルだけが検索される。

検索条件にあてはまるファイルに対してどのような処理を行うかを設定したら、[Find] ボタンを押して検索を実行する。単にファイル一覧をウィンドウに表示するだけでなく、外部コマンドを実行したり、検索結果を標準出力に出力してパイプで他のコマンドに渡すなど、さまざまな使い方が可能だ。

検索したファイルに対してコマンドを実行する場合は、[Run a shell command?] をチェックして、コマンドを含んだコマンドラインの内容を [Shell command] に設定する。コマンドライン中では、検索ファイル名を「%0」で参照可能だ。たとえば、“rm %0”と設定すれば、検索したファイルをrmコマンドで削除できる。また、ファイル名の一部だけを参照することも可能だ。

findコマンドの代わりに他のコマンドとパイプで接続して使う場合、検索結果が標準出力に出力されるように、[Print to stdout] をチェックする必要がある。[Print to window] のチェックを外して、結果のウィンドウ表示を抑止しておくともいだろう。両方をチェックしておき、ウィンドウ表示を確認に使うことも可能だ。



画面1 さまざまな検索条件をGUIを使って指定できる

GUIで手軽に文字列検索

reXgrep

バージョン: 1.2 ライセンス: GPL

<http://rexgrep.tripod.com/rexgrepmain.htm>

X GTK+ 日本語可

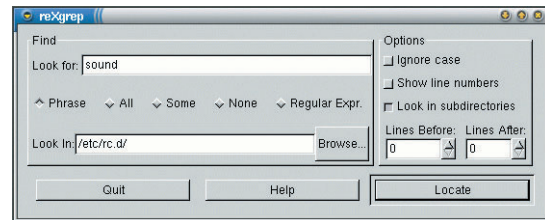
指定した文字列を含む行を複数のファイルから検索して表示するgrepのフロントエンド。文字列やオプションをGUIで設定でき、初心者にとってより理解しやすいPhrase / All / Some / Noneといった簡単な検索モードを備えている。サブディレクトリ以下のファイルを検索対象に含めたり、検索された前後の行をあわせて表示することもできる。

ダイアログや検索結果で日本語を表示するには、rexgrep.cの272行目の“gtk_init (...);”の前に“gtk_set_locale ();”を挿入し、display.hの87行目の“gtk_font_load (...)”を“gtk_fontset_load”に変更する必要がある。また、検索はgrepを内部で起動することにより実現しているため、サブディレクトリ以下を再帰的に検索する機能を持つGNU grep 2.3以降が必要だ。

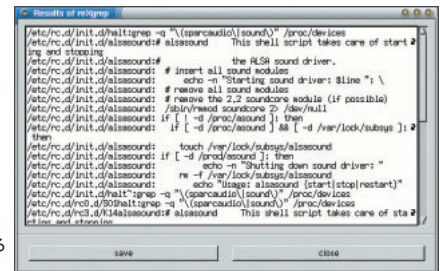
起動すると、検索文字列やオプションを設定するためのウィンドウが開く(画面1)。使い方は簡単で、[Look for] に検索する文字列、[Look in] に検索対象となるファイルを指定して(ワイルドカードによる指定も可) [Locate] ボタンを押せばいい。reXgrep内部でgrepが起動され、検索結果が別ウィンドウに表示される(画面2)。

検索モードは、5つのラジオボタンで切り替える。これらの違いは文

字列中の空白の扱いだ。初期設定の [Phrase] では空白も検索文字列に含まれるのに対し、[All] / [Some] / [None] では空白を区切り文字とみなして、複数の文字列に対する検索を行う。たとえば [All] の場合は、空白で区切られた複数の文字列をすべて含む行だけが検索される。最後の [Regular Expression] は、grepと同様の正規表現を直接指定するモードだ。このほか、大文字・小文字を区別しない [Ignore case] や、サブディレクトリ以下も検索する [Look in subdirectories] などのオプションも用意されている。



画面1 検索文字列とファイル名を指定して [Locate] ボタンを押す



画面2 検索結果を表示するウィンドウ。保存も可能だ

2つのファイルの違いをカラー表示

gtkdiff

バージョン: 1.0.2/1.5.1 (開発版) ライセンス: GPL

<http://www.ainet.or.jp/inoue/software/gtkdiff/index.html>

X GTK+ GNOME 日本語可

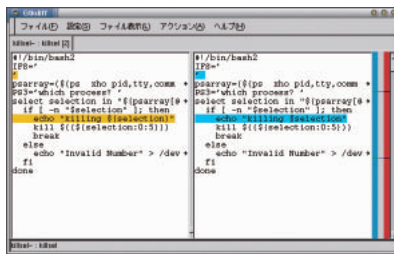
2つのファイルの内容を並べて表示し、両者の相違点をカラーで表示するソフト。プログラムのソースや各種設定ファイルなど、内容の一部を追加・変更・削除することの多いファイルに対して使うと、どの部分が増え減ったのか素早く把握できる。

“gtkdiff ファイル1 ファイル2”のように、比較する2つのファイルを指定して起動すると、ウィンドウが開いて両者の内容が左右の領域に表示される(画面1)。内容の異なる行の背景色はオレンジ色と空色で表示されるので、一目で違いを判別できる。画面領域が狭い環境では、[設定] - [画面表示]で[1画面表示]に切り替えるといいだろう。こちらは、2つのファイルの内容をひとつにまとめ、相違部分だけを背景色を変えて上下に並べて表示してくれる。なお、ウィンドウの左側には、各ファイル内での相違点とその対応がオレンジ色・空色のバーとそれを結ぶ線で示される。

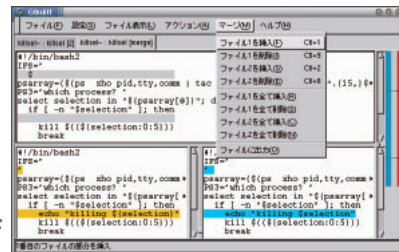
[アクション]メニュー以下の項目を選択するか、Ctrl - NまたはCtrl - Pキーを押すと、前後の相違点に即座に切り替わる。なお、[設定] - [行の折り返し]をチェックした状態ではこれらの機能は動作しないため、通常は行を折り返さない設定で使うことをお勧めする。

このほか、2つのディレクトリを比較して、追加・変更されたファイルを一覧表示する機能も用意されている。コマンドラインで2つのディレクトリを指定して起動すると、片方だけに含まれないファイルや、同名だが内容の異なるファイルのリストが表示され、よく似た2つのディレクトリの内容の違いを簡単に把握できる。

開発版の1.5.1では、2つのファイルを統合して新たなファイルを作成するマージ機能が追加された。2つのファイルの相違点を3分割されたウィンドウ上で確認しつつ、どちらを採用するか(あるいは削除するか)を[マージ]以下のメニューで選択できる(画面2)。



画面1 内容が異なる部分は背景色付きで表示される



画面2 開発版ではマージ機能が追加されている

各種パスワードの暗号化と取り出しを行う

Figaro's Password Manager (FPM)

バージョン: 0.50alpha ライセンス: GPL

<http://www.figaro.org/fpm/>

X GTK+ GNOME 日本語可

複数のユーザー名とパスワードを暗号化して保存し、復号化して取り出せるGNOMEアプリ。Webベースの掲示板など、Linuxへのログイン以外の場面で使われるパスワードの管理に便利だ。保存した情報は、クリップボードやプライマリセクションにコピーしたり、起動するコマンドのコマンドラインで利用できる。

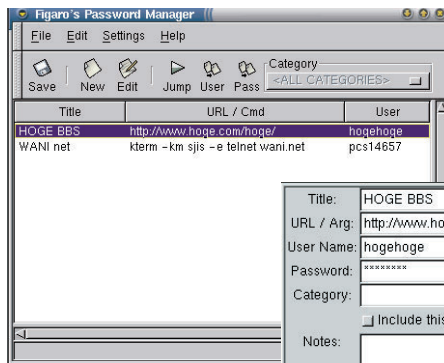
起動すると、最初にFPM用のパスワードの入力を求められる。これは、起動時のセキュリティを保つとともに、FPMで管理する情報を暗号化/復号化する際のキーとしても使われる。続いて、設定済みのエントリのタイトル、URL/コマンド、ユーザー名が一覧表示されたウィンドウが開く(画面1)。

エントリを新規作成するには、ツールバーの[New]ボタンを押せばいい。ダイアログが開くので、タイトルやユーザー名、パスワードを設定しよう(画面2)。Webベースの掲示板の場合であれば、[Launcher]を[Web]に設定し、[URL/Arg]に掲示板のURLを設定する。入力したパスワードは、通常「*」でマスクされており、[Show Password]ボタンで内容を確認できる。パスワードを自動生成することも可能だ。

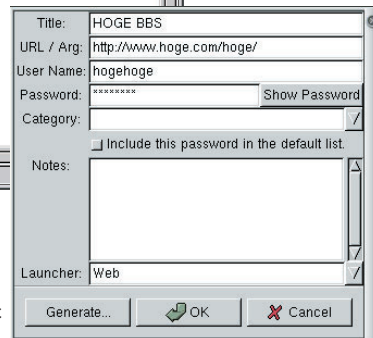
設定完了後、情報を取り出したいエントリを選択し、ツールバーの

[User]や[Password]ボタンを押すと、クリップボードとプライマリセクションにユーザー名やパスワードの内容がコピーされ、Ctrl - Vキーやマウスの中ボタンでアプリケーションにペーストできる。

[Jump]ボタンを押すと、Netscapeが起動され、指定したURLのページを表示する。このとき、ユーザー名はクリップボード、パスワードはプライマリセクションにコピーされており、それぞれAlt - Vキーとマウスの中ボタンでペーストすることができる。このほか、sshや任意のコマンドを起動する設定も可能で、ユーザー名やパスワードの取り出し方法を柔軟に設定できる。



画面1 パスワードを管理しているエントリの一覧が表示される



画面2 ユーザー名やパスワードなどを設定する



vi風のキーバインドで使えるバイナリエディタ

bvi

バージョン : 1.2.0

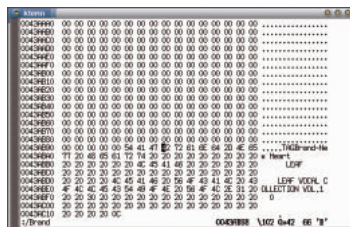
ライセンス : GPL

<http://bvi.linuxave.net/>
<http://www.chez.com/prigaux/hexedit.html> (HexEdit)

コンソール

ファイルの内容を16進数とASCII文字列で表示し、バイナリファイルの内容を閲覧したり、書き換えたりできるバイナリエディタ。キーバインドはviに準拠しており、おなじみのh/j/k/lキーでのカーソル移動をはじめ、iキーやaキーによる挿入、xキーやdキーによる削除、「:」で始まるex系のコマンドなど、viで使用するキー操作のほとんどがバイナリファイル向けに見事に昇華されている。

ただし、バイナリファイルでは「行」という概念が存在しないため、dキーの2度押しによる1行削除などの行ベースのコマンドは存在しない。一方、16進数表示部とASCII表示部のカーソル切り替えにTabキーが使われていたり、バイナリサーチや論理操作 (AND/ORやシフト、ローテート) を行えるなど、viには存在しないコマンドもある。動作が細かな点で異なるコマンドも存在するため、bviのHTML形式のマニュアルには、bviとviでの各キーの動作の違いが詳しく記述されている。



グラフィカルなブートマネージャ

GAG

バージョン : 3.1

ライセンス : GPL

<http://raster.cibermillennium.com/gageng.htm>

日本語可

Windows、Linux、BeOSなど9種類のOSをブートできる、スペイン生まれのグラフィカルなブートマネージャ。ブート用のフロッピーやハードディスクのMBR (マスターブートレコード) にインストールすることで、OSの切り替えをアイコンメニューから行えるようになる。

GAGはフロッピーディスクイメージとして配布されているので、ハードディスクのMBRにインストールする場合にも、いったんブート用フロッピーを作成する必要がある。その後、作成したブート用フロッピーでPCを起動し、きちんと各OSがブートすることを確認してから、ハードディスクのMBRにGAGをインストールするという手順になる。なお、すでにMBRにliloをインストールしている場合には、Linuxのパーティションのブートセクタにliloをインストールしなおす必要がある。



TrueType / Type1 フォントの設定をGUIで行える

ttftool

バージョン : 4.3.3/2.9 (RPM) ライセンス : GPL

<http://www2.famille.ne.jp/mituiwa/#font>

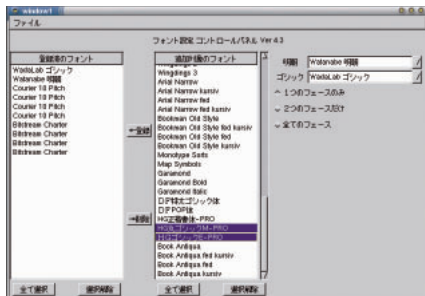
X

GTK+

日本語可

TrueTypeフォントやType1フォントのインストールや設定を専用のコントロールパネルで行うソフト。X-TTやVFlibなどでこうしたスケラブルなフォントを扱う際、従来はさまざまな設定ファイルをエディタを使ってひとつひとつ書きかえる必要があった。このttftoolを使えば、フォントを一覧から選択してインストールするだけで、あらかじめ用意された「テンプレート」の設定に従って各種の設定ファイルを自動的に作成してくれる。特に、一度に複数のフォントを追加する場合などに便利なツールだ。

tarボールとRPMパッケージの両方で配布されているが、RPM版はバージョンが古くType1フォントに対応していない。最新版はtarボールによる配布のみなので注意してほしい。



アプリ起動中に砂時計カーソルを表示

Xalf

バージョン : 0.4

ライセンス : GPL

<http://www.lysator.liu.se/altic/projects/xalf/>

X

GTK+

X用アプリケーションが起動中であることを、砂時計カーソルなどの方法でユーザーにフィードバックするソフト。砂時計カーソルのほか、GNOMEやKDEのタスクリストにカッコ付きでアプリ名を表示する、砂時計のスプラッシュスクリーンを表示する、アニメーションする星印を表示する、という計4種類のフィードバック方法が用意されている。これらは起動時のオプションで使い分ける (初期設定はタスクリストへの表示)。なお、一部のラッパースクリプト (netscapeなど) に対しては、ウィンドウが開いているのにタイムアウトになるという問題が生ずるため、関連ツールの「xalfoff」をスクリプト中に書き加えるなどの修正が必要となる。また、GNOMEの最新版「GNOME 1.2」では、Xalfがビルトインサポートされている。



各種ハードウェア情報の表示と設定

HardDrake

バージョン : 0.8.1

ライセンス : GPL

<http://www.linux-mandrake.com/harddrake/>

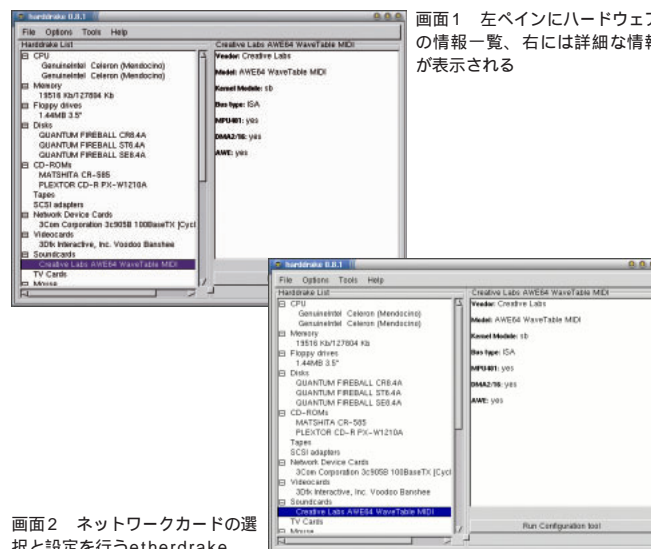
X

GTK+

Windowsの「デバイスマネージャ」のように、ハードウェア情報の表示や設定をX上で行うソフト。ハードウェアの設定をGUIを使って簡単に行うことを目標として、MandrakeSoftが中心となって開発しており、MandrakeだけでなくRed Hatなど他のディストリビューションでも動作する。現時点では、HardDrake本体に加えて、ハード情報の検出を行う「detect」、ALSAサウンド設定モジュール「sounddrake」、イーサネット設定モジュール「etherdrake」がそれぞれ提供されている。このほかにも、マウスやプリンタなどに関する同様の設定モジュールを開発中のようだ。

スーパーユーザー（root）になった状態でHardDrakeを起動すると、detectによりハードウェア情報が自動的に検出され、HardDrakeのウィンドウに表示される（画面1）。表示は左右のペインに分かれている。左側には、CPU、メモリ、ディスク、CD-ROMドライブ、ネットワークカード、ビデオカード、サウンドカードといったハードの分類の下に、detectで検出されたハードウェアがツリー表示される。これらの分類をクリックして、ツリー表示されたハードウェアを選択すると、右側のペインに詳細情報が表示される。

ネットワークカードやサウンドカードを選択した場合は、情報の下に [Run Configuration Tool] ボタンが表示される。このボタンを押すと、それぞれに対応する設定モジュール（etherdrakeとsounddrake）が起動する（画面2）。いずれも、リストから自分のカードを選択するだけの簡単設定だ。ISAデバイスの場合は、I/OアドレスやIRQなどの設定も同時に行う。なお、これらの設定モジュールにより/etc/conf.modules内のネットワークカードやサウンドカードに関する設定が書き換えられるので、バックアップをあらかじめとっておくといいたいだろう。



画面1 左ペインにハードウェアの情報一覧、右には詳細な情報が表示される

画面2 ネットワークカードの選択と設定を行うetherdrake

パケットを監視するネットワークアナライザ

Ethereal

バージョン : 0.8.10

ライセンス : GPL

<http://ethereal.zing.org/>
<ftp://ftp.ee.lbl.gov/> (libpcap)

X

GTK+

イーサネットやPPP接続用のデバイスを監視し、そこを流れるTCPなどのパケット情報を取得するネットワークアナライザ。取得されたパケット情報をGUIにより確認できる。キャプチャや表示の際に、フィルタを使って情報を絞り込むことも可能だ。動作にはパケットキャプチャライブラリlibpcapが別途必要となる。

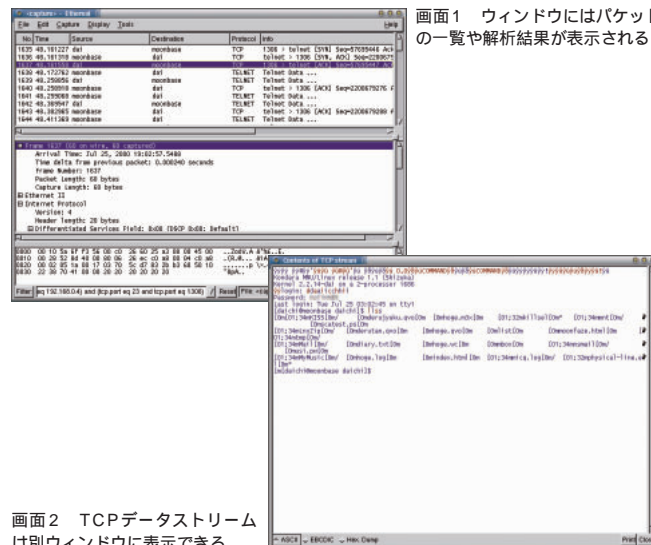
スーパーユーザー（root）になった状態で起動すると、3つのペイン（領域）で構成されウィンドウが開く。メニューの [Capture] - [Start] でキャプチャ開始用のウィンドウが表示され、インターフェイスの選択やフィルタの設定を行える。[OK] ボタンでキャプチャ開始。キャプチャ中は状態表示ウィンドウにパケットの種類別のパーセンテージなどが表示され、[Stop] ボタンを押すとキャプチャが終了する。

キャプチャ終了後のウィンドウの上部ペインには、取得したパケットの取得日時、送信元、送信先、プロトコル、簡単な説明が一覧表示される（画面1）。強力な表示フィルタを使って、表示するパケットを絞り込むことも可能だ。

一覧中のパケットをクリックすると、そのパケットを解析した結果が中央のペインにツリー状に表示され、その下のペインにはダンプリスト

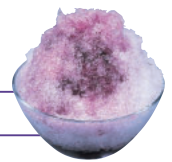
が表示される。ツリーの一部をクリックすると、ダンプリスト中の対応する部分が太字に変化するなど、細かな工夫がなされている。また、TCPデータストリームの内容（テキスト）を別ウィンドウに表示することも可能だ（画面2）。

このほか、取得したデータをファイルに保存しておいて、あとから読み込んだり、ほかのツールの出力（たとえば、tcpdumpのパケットモニターファイルなど）を取り込んでEtherealで解析するという機能も用意されている。



画面1 ウィンドウにはパケットの一覧や解析結果が表示される

画面2 TCPデータストリームは別ウィンドウに表示できる



ネットワークトラフィックをWebブラウザで監視

ntop

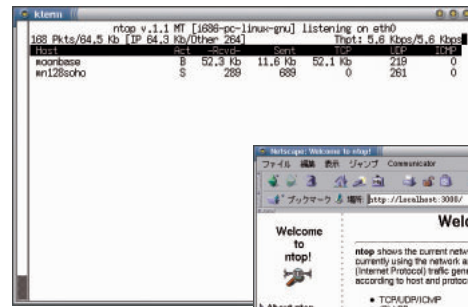
バージョン : 1.1/1.3.1 (snapshot) ライセンス : GPL

<http://www.serra.unipi.it/~ntop/>

ネットワークトラフィックを監視して端末画面やWebページに表示するソフト。ntop (network top) という名前は、端末画面を利用するインタラクティブモードの表示が、プロセス一覧を表示するtopコマンドに似ていることに由来する (画面1)。

ただし、ntopで特筆すべきなのはインタラクティブモードではなく、もう一方のWebモードのほうだ。起動時に-wオプションとポート番号 (3000など) を指定することでHTTPサーバとして動作し、トラフィックの監視結果をWebページに表示する。

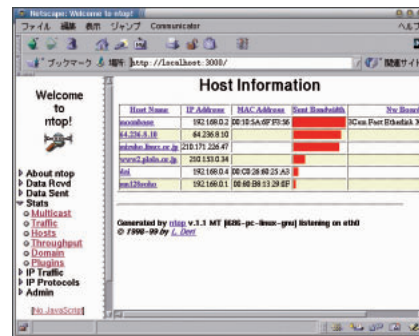
NetscapeなどのWebブラウザを使って、ホスト名とポート番号からなるURL (ローカルホストの場合は「http://localhost:3000/」) を指定すると、ntopのトップページが表示される (画面2)。左側の階層メニューをクリックすることで、送受信データのデータ量をドメイン別に表示するページや、IPトラフィックの統計を表示するページなどに切り替わる (画面2)。表示される結果は、背景の色分けや各種のグラフ、リンクなどがセンスよく使われていてわかりやすい。さらに、ntopにアクセス可能なユーザーを管理することも可能だ。



画面1 インタラクティブモードはtopコマンド風の表示だ



画面2 HTTPサーバとして動作するWebモードのページを表示



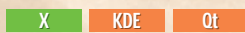
画面3 結果はグラフや表でわかりやすく示される

KDE用ネットワークモニタリングツール

KNETLoad

バージョン : 1.2 ライセンス : GPL

<http://www.kde.org/>



ネットワークの通信速度をKパネル上でダイアグラムとして表示するアプリケーション。デバイス、比率、受信/送信の色、更新頻度などを変更でき、さまざまなモニタリングが可能だ。CPUの使用率をモニタリングする「KCPULoad」と作者が同じで、外観や使用方法などが類似している。

インストールすると、KDEのスタートメニューの[システム]に登録されるので、そこから起動するか、あるいはコマンドラインから起動することができる。起動されると、Kパネルのタスクトレイエリアの右隅に格納され、ただちにモニタリングが開始される。

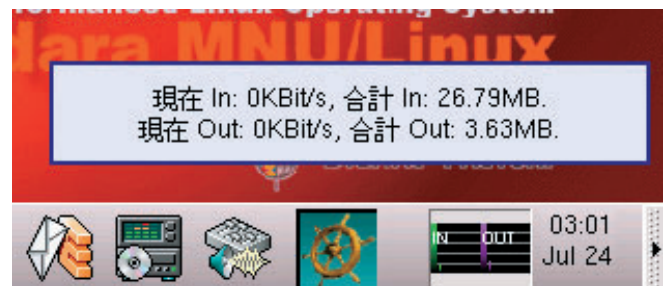
KパネルのKNETLoad上で右クリックすると、KCPULoadとほぼ同じメニューが表示されるが、[デバイス]と[比率]という独自のサブメニューがある。KNETLoadはネットワークを対象としたモニタリングアプリケーションで、さまざまなデバイスに対応している。[デバイス]サブメニューで、モニタリングしたいデバイスに合わせて設定を変更しよう。[比率]サブメニューは使用するデバイスの通信速度に合わせて設定する。リストにない比率を使用したい場合は、サブメニューの一番下にある「その他」を選択するとプロンプトが表示されるので、このプ

ロンプトに使用したい比率を入力すればいい。単位はkbpsだ。

もうひとつKCPULoadと違う点に、[色]のサブメニューがINとOUT用の2種類あることが挙げられる。同じ色にするもよし、違う色にするもよし、好みに応じて変更できるようになっている。色はINとOUTそれぞれに設定できるが、残念ながらスタイル(描画形式)は1つしか設定できない。

KNETLoadは必要に応じて、[動作]メニューにおいてオン/オフを切替えることもできる。また、メニューの一番上にある[ソフト]という項目がある。これはグラフのカーブをソフトにするかハードにするかを切り替えるためのものだ。メニューの中程にある[クリア]は、それまでに読み込んだ数値をクリアし、同時にグラフもリセットする。

なおKCPULoadと同様に、Kパネルのタスクトレイに格納されているKNETLoadを左クリックすると、現在のINとOUTの数値が文字で表示される。この表示を消すには、もう一度、タスクトレイエリアのKNETLoadを左クリックするか、表示されているエリアを右クリックすればいい。



GUI操作でポートスキャン

Kmap

バージョン : 0.7.2

ライセンス : GPL

<http://www.edotorg.org/kde/kmap/><http://www.insecure.org/nmap/> (nmap)

X

Qt

KDE

ポートスキャンツール「nmap」の設定や実行、結果の整理をGUIで行うKDE用のフロントエンド。自分の管理しているマシンが外部に開放しているポートをチェックし、不要なサービスを起動しないようにするのはセキュリティ保護の基本。Kmapを使えば、nmapのさまざまなコマンドラインオプションをGUI上で設定でき、スキャン結果の情報を整理してウィンドウで確認できる。

起動すると、シンプルなウィンドウが開く。使い方は簡単で、[Remote hostname] にポートスキャンの対象となるホスト名かIPアドレスを入力する。IPアドレスの場合は、複数のホストを一括スキャンすることも可能だ。ツールバー左端の [Start scanning] ボタンを押すと、nmapによるスキャンが実行される。ステータスバーに「Nmap : exited normally」と表示されればスキャンは終了だ。

スキャン結果は、3種類のウィンドウで表示される。なかでも、ポートとサービス名を一覧表示する「Port View」と、ホストごとにポートをまとめた「Information View」が重要だ(画面1)。新たなスキャンを実行すると、これらの表示は自動的に更新される。

Kmap自身やnmapに対するオプション設定は、ツールバーの [Setup

options] ボタンで開く設定ダイアログで行う(画面2)。設定項目は多岐にわたるため、詳細はnmapのマニュアルを参照されたい。

不要なサービスが有効になっていることがわかったら、対応するデーモンを起動しないように設定ファイル(/etc/inetd.confなど)を修正するなどの対策をとったほうがよい。なお、自分の管理外のマシンに対してポートスキャンを行うと、それだけでクラックと見なされる可能性があるのだからくれぐれも注意してほしい。



画面1 ポートスキャンの結果を整理して2つのウィンドウに表示

画面2 設定ダイアログではnmapに対する細かい設定が可能だ

CPUやディスクの負荷をグラフ表示

GKrellM

バージョン : 0.10.4

ライセンス : GPL

<http://web.wt.net/~billw/gkrellm/gkrellm.html><http://www.muhri.net/> (テーマ)

X

GTK+

CPUやディスクの負荷、プロセス数、PPPやイーサネットのパケット流量、メモリやスワップの使用量などを監視して、リアルタイムにグラフやメータで表示するモニタリングソフト。このほか、カレンダーやファイルシステムのマウント、メールチェックなど豊富な機能を内蔵しており、プラグインによる機能の拡張やテーマ(スキン)を利用した外観の変更も可能だ。

起動すると、縦型の小さなウィンドウが開いて、CPU・プロセス・ディスクのグラフ、メモリやスワップのメータなどが表示される(画面1)。情報は1秒ごとに更新され、グラフの縦方向のスケールは測定値に応じて自動的に調整される。また、各グラフともシアンとオレンジの2色で異なるデータを表示する。たとえば、CPUのグラフではユーザータイムとシステムタイム、ディスクのグラフではリードとライトといった具合だ。なお、マルチプロセッサシステムでは、使用しているCPUの数だけグラフが表示される。すべてのCPUを合わせた「コンポジットCPU」のグラフを表示することも可能だ。

ホスト名表示部を右クリックして [Configuration] を選択すると、設定ダイアログが開く(画面2)。設定項目は左側のツリーでジャンル

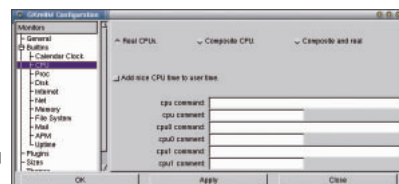
別に分類されており、各項目の表示の有無や計測単位量の設定などが可能だ。また、CD-ROMなどのマウント/アンマウントをボタン一発で行ったり、ローカルやリモートのメールボックスにメールが届いているか調べてメールクライアントを起動することもできる。

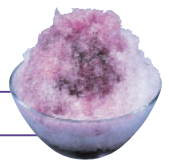
別途配布のプラグインを利用すれば、MP3プレーヤや天気予報、ミキサーなどの機能をGKrellMに追加できる。また、テーマサイトの「MUHRI DOT NET」には100種類以上のテーマが用意されており、手軽に外観を変えることも可能だ。



画面1 テーマ(スキン)により外観を簡単に切り替えられる

画面2 設定ダイアログで表示内容やグラフのサイズを変更する





プロセス情報をグラフィックで表現

LavaPS

バージョン : 1.13

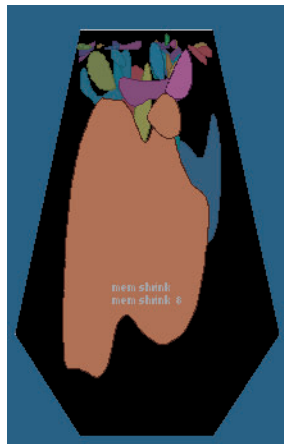
ライセンス : GPL

<http://www.isi.edu/~johnh/SOFTWARE/LAVAPS/>

X Tcl/Tk

プロセス情報を不定形な「プロブ」状のグラフィックで表示するソフト。各プロセスの情報をプロブの大きさなどのアナログな要素によって表現するため、現在のシステムの状況を感覚的に理解できる。さらに、ポップアップウィンドウで各プロセスに関する詳しい情報を表示したり、プロセスにシグナルを送信することも可能だ。

プロブとプロセスは1対1で対応しており、プロセスが生成されると新たなプロブが追加され、プロセスが終了すると対応するプロブが消滅する。各プロセスの情報は、対応するプロブの大きさや動き、色などによって表現される。たとえば、プロブの大きさはメモリの消費量、動きはCPUの使用量に比例している。



画面1 LavaPSは単なるプロセスモニターであるが、ついつい見入ってしまいがちだ

タスクトレイでCPUの稼働率を表示

KCPULoad

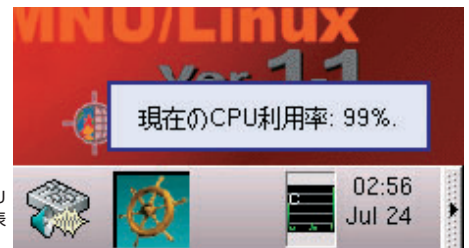
バージョン : 1.4

ライセンス : GPL

<http://www.kde.org/>

X KDE Qt

KCPULoadは、CPUの稼働率をKパネル上でダイアグラムとして表示するツールだ。SMPシステムのマシンでは、2つのCPUの利用率の合計を表示する。起動すると、Kパネルのタスクトレイエリアの右隅に格納され、すぐにモニタリングが開始される。もちろん、好みに応じてモニタリング形式の変更も可能だ。変更できる項目は、色、スタイル、更新間隔、ラベル、グリッドの有無、カーブのソフト/ハードなど。これらの項目設定は、KパネルのKCPULoad上で右クリックして行う。ちなみに、左クリックすると現在のCPU利用率が文字で表示される。KCPULoadの数値は、CPUの稼働率の指標であり、実際のCPUの使用率ではないので、注意しよう。



画面1 パネル上でCPU稼働率をコンパクトに表示する

動作中のプロセスを監視するアプレット

GProc-applet

バージョン : 0.6.0Pre2

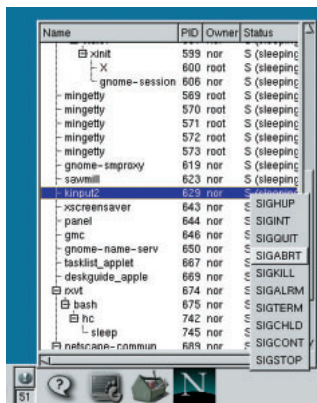
ライセンス : GPL

<http://gproc.cjb.net/>

X GNOME

GProcは、現在動作中のプロセスを監視、およびコントロールするGNOMEアプレットだ。

ふだんはパネル上にプロセス数のみを表示しているが、このアプレットの上部のアイコンを右クリックすると、現在動作しているプロセスのプロセスID、所有者、ステータスの一覧が、親子関係を含めて表示される。さらに、表示されたそれぞれのプロセスを右クリックで選択し、左クリックすると、シグナルのメニューが表示される。そして右クリックすれば、そのプロセスに対してシグナルを送ることができる。サーバプログラムの設定を変更するために、何度もSIGHUPシグナルを送ったりする際には、このツールが役に立つだろう。



画面1 GNOMEパネルに小さくおさまるGProc-applet

プロセス情報をグラフィカルに表示

gPS

バージョン : 0.8.1

ライセンス : GPL

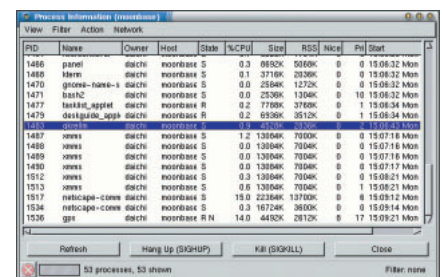
<http://www.gps.seul.org/>

<http://www.student.nada.kth.se/~f91-men/gps/> (gps)

X GTK+

従来psコマンドやtopコマンドを使って表示していたプロセス情報をGTK+を利用してグラフィカルに表示するソフト。topコマンドと同じ形式でプロセス情報が表示され、項目名の内容によってソートしたり、特定のプロセスだけに表示を絞り込んだりできる。また、メニューやボタン操作によりプロセスにKILLなどの各種シグナルを送ることも可能だ。

起動すると、プロセスIDやプロセス名、所有者、CPU使用率、nice値など現在のプロセス情報がウィンドウに表示される。これらのプロセス情報の各項目をキーとして昇順/降順にソートできるほか、プロセスを選択して詳細情報を表示したり、プロセスの親子関係をツリー表示することもできる。また、CPUとメモリの状態をグラフ化して表示する機能も用意されている。



画面1 プロセスの状態を確認しながら、シグナルを送ることが可能

Software Catch up

韓国で誕生した日本語ワープロ

HancomWord

HancomWordは、韓国HancomLinux社で開発されたLinux用の日本語ワープロだ。X Window System上で動作し、日本語だけでなく韓国語や中国語の入力も可能な多言語ワープロでもある。今回は、このHancomWordのレポートをお届けする。

文：塩田紳二
Text : Shinji Shioda

価格 9800円（インターネット販売）
問い合わせ先 HancomLinux Japan, Inc.
TEL : 03-3257-0490
URL : <http://hancom.com/jp/>

HancomWord（画面1）は、韓国HancomLinuxの日本語ワープロだ。このHancomLinuxは、韓国のソフトウェアハウスHaansoftの子会社で、Haansoftは韓国のベストセラーワープロであるアレアハングル（Windows用）のメーカーとして有名である。韓国では、400万人のユーザーがいるという。

このHancomWordも、今年5月のLinux World Expo/Tokyo 2000のときにはアレアハングル2000という名前でもデモされていたが、日本語パッケージの出荷にあたりこの名称に変更された。なお、このHancomWordはレッド

ハットのRed Hat Linux 6.2J改訂版にバンドルされている。

日本と同じ2バイトコードを使う韓国で作られたHancomWordは、基本的には多言語ワープロであり、日本語版とは、デフォルトの文字コードやメニューなどが日本語化されているものといえる。

なお、このHancomWordはLinux上のWindowsアプリケーション実行環境であるWineを利用して動作する。この点では、CorelのWordPerfectなどと同じだが、配布されているバイナリを見る限り、Windows用のバイナリを直接

Wineで動かしているのではなく、Wineの提供するAPIを使って動作しているようである。おそらく、Windows用のソースコードをWineAPIを使う形でコンパイルして、Linux用バイナリを作っているのだと思われる。

HancomWordは単体のワープロで、表計算やプレゼンテーションソフトなどを含むオフィススイートではない。このため、現在のところ他のアプリケーションで作成した文書ファイルなどを埋め込んだり、リンクするようなことはできない。しかし、文書ファイル中の表では計算式による各種演算が可能だ。また、背景を指定して文書をページごとにフルスクリーンで表示するプレゼンテーション機能を持つ。ある意味、ワープロを中心としたオフィス用ソフトという感じである。

また、日本語版ではMicrosoft Wordや一太郎といったWindows用ワープロの文書ファイルの読み込みに対応しているほか、Lotus1-2-3のスプレッドシートファイル（WKx形式）やHTML、RTF（Windowsで広く使われるリッチテキストファイル。書式付きの文書をテキスト形式のコマンドで表現したもの。文書ファイルの交換用）などに

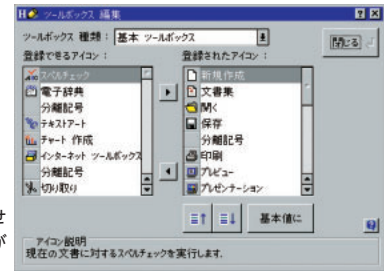


画面1
今回は、Red Hat Linux 6.2Jを使ってHancomWordを動かした。画面右上にあるのが、起動用のHancomShell。



画面2 HancomWordは文書ごとにウィンドウが開くようになっている。タイトルバーの下にメニューバー、さらにその下にツールバー、ルーラーが配置される。なお、水平スクロールバーはステータスバーと一体になっている。

画面3 ツールバーは、表示させるボタンやその順番をユーザーがカスタマイズすることが可能。



画面4 編集エリアで右クリックすると、コンテキストメニューが表示される。メニュー内容は、カーソル位置にあるオブジェクトによって変化する。



も対応している。

なお、今回の評価は、Red Hat Linux 6.2J改訂版にバンドルされるものを利用し、Red Hat Linux 6.2J上で評価した。

GUIまわり

HancomWordは、1つのウィンドウに1つの文書を表示する形式のウィンドウ構造を持ち(画面2)、各文書ファイルウィンドウは、メニューバー、ツールバー、ルーラー、文書表示領域、ステータスバーなどを持つ。ある意味、GUIの世界ではごくあたりまえの構成になっている。ただし、水平方向のスクロールバーが、ステータス行と同じになっているなどちょっとした特徴がある。また、ツールバーが最近のWindowsアプリケーションで使われるReBarコントロール(左側に縦線が入っており、そこを掴むことで移動などが可能になるツールバー構成用のGUIコンポーネント)に似ているものの、ツールバーをマウスで動かしたり、フロートウィンドウとしてメインウィンドウから切り離すなどの操作が行えないなどの違いもある。

ツールバーは、最初から登録されている5つのツールバー(標準、作画、

表編集、書式とインターネット)のみであり、各ツールバー上のボタンの有無や順番は設定が可能なもの(画面3)、ユーザーが任意のボタンを組み合わせることで新しいツールバーを作ることができない。また、このツールバーの設定では、「インターネット」という名称のツールバーが設定対象となっているのだが、このツールバーはメニューから表示を指定することができなかった。

文書編集エリアの上端と左端には、ルーラーを表示することができ、水平ルーラーでは左右のインデント(マージン)の設定やタブ設定が行える。ただし、ページ全体の余白の調整はマウスからは行えず、ページ設定(HancomWordの用語では「編集用紙」)で用紙サイズなどととも上下左右の余白やヘッダ、フッタの位置を指定するしかない。

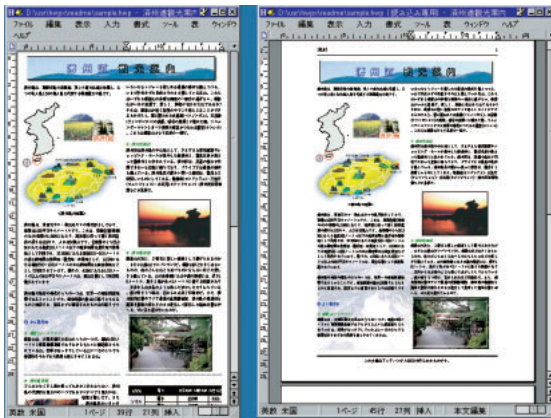
文書編集エリアでは、マウスの左ボタンで選択とキャレットの移動、右ボタンでコンテキストメニューとなる(画面4)。このコンテキストメニューは、マウスカーソルの下にあるオブジェクト(文字列や表、図版など)によりメニュー内容が切り替わるものだ。ただし、メインメニューを含めて、メニューをカスタマイズする機能はない。

なお、HancomWordの起動は

GNOMEやKDEのアプリケーションメニューにHancomLinuxというメニューが登録され、そこから行うことができる(画面5)。HancomWordを直接起動することも可能だし、HancomShellという起動用のアプリケーションを最初に起動しておき、そこから起動することもできる。このHancomShellは、4つのボタンがある小さなウィンドウとなっており、それぞれのボタンは、「HancomWordの起動」、「設定変更」、「ヘルプ」、「終了」の機能を持つ。簡単なランチャー(Microsoft Officeというショートカットバー)だが、現時点では、HancomWordのみがメインアプリケーションであり、必ずしもHancomShellを起動する必要はなさそうである。



画面5 HancomWordをインストールすると、GNOMEやKDEのメニューに起動用の項目が登録される。



画面6 ドラフトモード(左)とページモード(右)。ページモードでは、ヘッダやフッタを含め印刷イメージで表示される。



画面7 Hancowordは段組みの表示が可能。文章は、Microsoft Word 2000の文書を読み込んだもの(手動にて修正してある。本文参照)。

文書編集機能

Hancowordには、ヘッダやフッタを含めて印刷イメージに近い形でページを縦に並べて表示するモードと、ページ区切りのみを表示するドラフトモードの2つの編集モードがある(画面6)。また、編集エリアにグリッド(格子)を表示することも可能だ。図版や表の枠、編集記号などについては、設定により非表示とすることもできる。ただし、文章を階層構造で表示するアウトライン機能はない。

本誌の誌面のように1ページを複数のコラムにわたる「段組み」を行うこともできる(画面7)。海外で作られたLinux用ワープロなどでは、この段組みの機能がないことがあるが、アジア圏では段組みは必須ともいえる。その点、こうした段組みに対応しているの

はありがたい機能である。ただし、筆者がMicrosoft Word 2000の段組みを指定した文書を読み込んだところ、元文書に指定してあった段組みは解除されてしまった。おそらくインポート用の文書変換フィルタの仕様だと思われるが、ぜひ対応してほしい機能である。

文書は、テンプレートを元に作成することができ、テンプレートには予め図版や文書を埋め込んでおくことができる(画面8)。また、これとは別にスタイル機能(画面9)があり、文書中で指定したスタイル名と設定される属性を切り替えることができる。これは、HTMLというCSS(Cascading Style Sheet)に似た機能で、スタイルセットをまとめて切り替えることで、指定したスタイル名の表示属性が変わるものだ。このスタイル機能は、前述のテンプレートとは独立しており、テンプレートを使って文書を作成したあとに、スタイルだけをまとめて切り替えることができるようになっている。Microsoft Wordなどでは、テンプレートの中にスタイルが登録されており、基本的にはスタイルとテンプレートは1対1の対応となっている。スタイルを変更することもできるが、テンプレートを更新してしまうと、以後、そのテンプレートで作成する文書のスタイルも変更されてしまう。

文書中には、

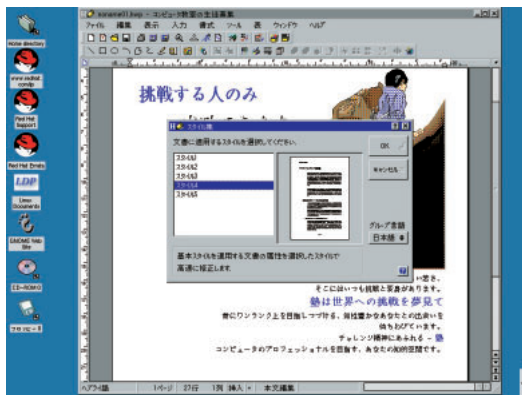
- グラフィック
- ベクトル図形
- 罫線
- 表
- テキストボックス
- 特殊コード
- 数式

といったオブジェクトを配置することができる(画面10)。グラフィックは、BMPやJPEGといったビットマップファイルやEPSなどで、ファイルを指定して直接ページ内に埋め込むか、文書ファイルにリンクのみを埋め込むかを選択できる。これに対してベクトル図形は、矩形や円、矢印線といった基本図形を組み合わせて描画を行うものだ。このベクトル図形で作成したクリップアート集(Hancowordの用語だと図片集)も付属している。

表は簡易計算機能があり、1つの表内で各セルに入力した数値の合計なども別のセルに入れることが可能だ(画面11)。Hancowordには、現在のところ連携できる表計算ソフトはないが、これを使うことで文書中に計算式を含んだ表を入れることができる。また、この表とは別に罫線文字を使った罫線をカーソルキーを使って引くことがで



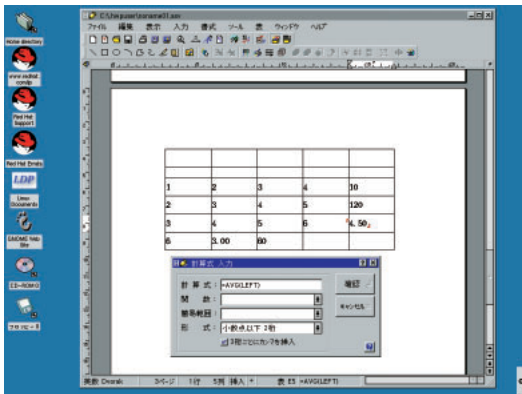
画面8 文例集にあるテンプレートから文書を作成することができる。テンプレートは、HancowordでHWT形式で保存することで作成できる。



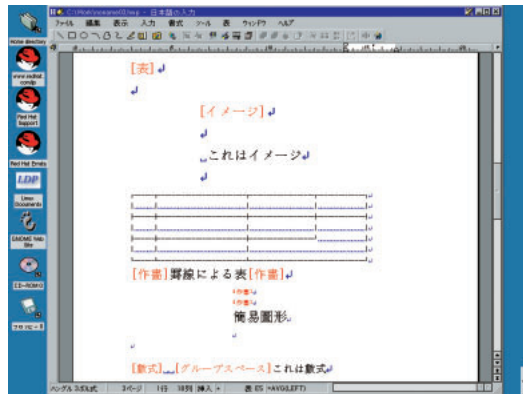
画面9 テンプレートとは別に、見出しや本文に対する書式設定をまとめた「スタイル」を別途変更することが可能。



画面10 HancmWordは文書内に表やイメージ(ビットマップ)、ベクトル図形、数式といったオブジェクトを配置することができる。



画面11 表内では計算式を使うことができ、これを使うことでちょっとした集計表などを作ることができる。



画面12 文書内に配置されるオブジェクトは、「組版符号」を表示させるモードではこのように赤い文字で表示される。これは、画面10を表示させたもの。

きる。国産のワープロなどでは、ごく当たり前に装備されていた機能である。Microsoft Wordなどでは表の罫線を使ってこの機能を実現している。大きな違いは、表は本文テキストとは独立したオブジェクトであるのに対して、罫線文字を使った罫線はあくまでも本文扱いであることである。テキスト入力では罫線が崩れないように罫線保護の機能もある。このあたり、やはり米国製ソフトの移植ものとは違った感覚で、国内のユーザーにはありがたい機能だ。

テキストボックスは、本文とは別オブジェクトになる矩形で、ここに文字を入れることができるものだ。実際にはベクトル図形の一つである。

特殊コードは本文中に埋め込まれるもの(画面12)で、たとえば、現在の日付や編集中の文書ファイル名などである。これは文字として扱われるが、ひとまとまりのオブジェクトとなる。

Microsoft Wordというフィールドコードに相当するものだ。また、この機能を使って、文中に数学記号を使った数式やルビを実現している。

ほとんどのオブジェクトは、本文の回り込み(オブジェクト左右に本文が表示される)、回避(オブジェクトの左右には本文が表示されず、オブジェクトの下から本文の続きが表示される)、別レイヤ扱い(本文がオブジェクトに重なって表示される)といった指定が可能になっている(画面13)。

その他

そのほか原稿用紙への印刷を行う機能がある。この場合には、プレビュー時に原稿用紙を含めてプレビューが行える。また、原稿用紙は、200字、400字、1000字のものが切り替えられる。

表内では計算式が利用でき、合計や



画面13 各オブジェクトは、サイズやページ内の位置、本文との関係などを設定することができる。

平均、三角関数、logなど26個の関数を用意されている。ただし、用意されているのは、数学、統計関係の数値関数のみである。表は、Lotus1-2-3やExcelなどで使うA1形式(表の列がA、B、C.....、行が1、2、3.....となっていて、この組み合わせでセルを指定する方法)でセル指定が行え、これを関数などの引数として利用する。また、計算式の置かれているセルの上下、左

右方向に並んでいるすべてのセル範囲を表すキーワード（LEFT、RIGHT、ABOVE、BELOW）が関数の引数として利用できる。また、数値の表示形式も指定することができる。

Microsoft Wordにあるように、マウスを使って表を書いていくこともできる（これは、前述の罫線描画とは別機能）。セルの結合なども可能で、セル内に斜め線を引くこともできる。

プレゼンテーション機能は、文書の各ページに背景をつけてスライドショー表示が可能になるものだ（画面14・15）。ページ内にグラフィックやベクトル図形なども配置できるので、表現力としては普通のプレゼンテーションソフト並といえる。プレゼンテーション専用ソフトとの違いといえば、表示項目やページ切り替え時にワイプなどの効果を付けたり、スライド（ページ）ごとに並べ替えを行う機能がないということぐらいだ。

Wineを使って機能を実現しているため、システム側のフォントではなくHancomWord自身が持つフォントのみを利用する。このため、インポートした文書などで指定されていたフォントは、読み込み時に置き換えが行われる。また、印刷時には用意されたプリンタドライバを使うのだが、現在のところPostscriptプリンタを前提としている

ようである。このため、フォントもPostscriptを前提としたものなのである。ただし、画面表示に使われるフォントには、一部アウトラインフォントでないものがあり、ドットフォントを無理矢理拡大した表示（ギザギザ）になるものがある。

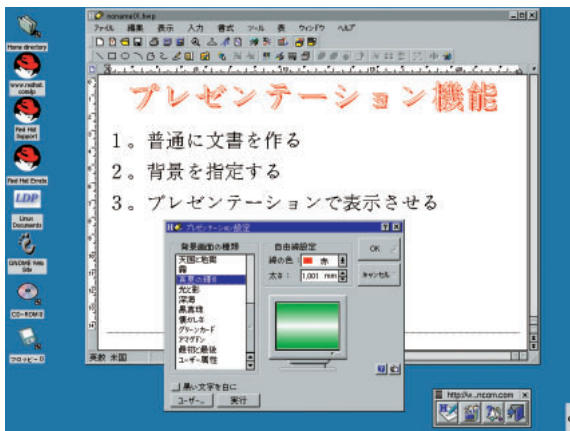
HancomWordは多言語に対応しており、欧文や日本語のほかに韓国語や中国語、ヘブライ語などの入力、編集が可能だが、日本語の入力にはX Window Systemの入力メソッドを利用する。このため、Xで日本語入力が可能になっているなら、Shift+Spaceキーで日本語入力モードとなる。また、韓国語用の入力メソッドはHancomWordが持っているが、中国語の入力メソッドは組み込まれておらず、システム側で用意する必要がある。

この入力言語に合わせてキーボードレイアウトを変更することができる（画面16）。欧文用には、QUERTYとDOVORAKの2つが用意されており、このほか全角アルファベットを入力するためのキーボードレイアウトなどがある。このキーボードレイアウトは、ステータスバーやタイトルバーの右側に表示され、現在のレイアウトが判断できるようになっている。また、現在のレイアウトを表示させ、それをソフトウェアキーボードとして利用するこ

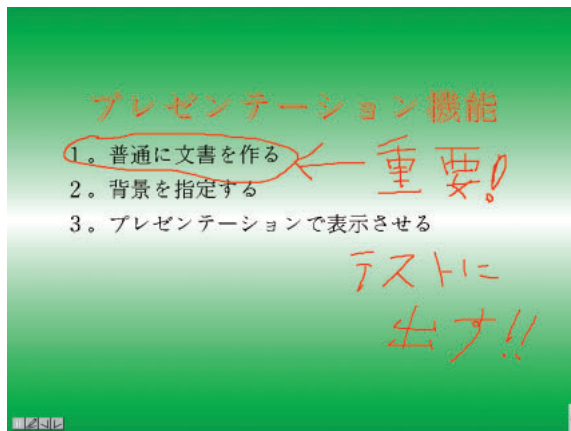
とも可能だ（画面17・18）。キーボードレイアウトは、2つの切り替えキーを使って、最大4つをキー操作で切り替えて利用することができる。標準では、この切り替えキーに左右のCtrl+Spaceキーが使われており、このキーの組合せで切り替えができる。また、グラフィックキャラクタや数学記号などをキーボードに割り当てて、直接入力するといった使い方も可能になっている。

こうした多言語処理の場合、キーボードレイアウトや入力メソッドなどは、本来はシステム側が用意するのがスジなのだが、Windows環境ではWindows自体のローカライズ処理が必要で、多言語の同時利用には難しい部分がある。HancomWordは、Windows用ワープロ（アレアハングル）をベースにしているため、このように入力関連機能を自身で持っているのだと思われる。

このほかに目立った機能としては、正常終了しなかった場合の編集中ファイルの回復機能がある。これは、一定時間ごとに編集中のファイルをセーブしておき、システム側の要因でHancomWordが終了してしまったとき（俗にいう落ちたとき）に、再度起動するとセーブしていなかったファイルを回復する機能である。Windowsなどでも、落ちる可能性は低くなったとは



画面14 プレゼンテーション機能は、文書に背景を指定し、プレゼンテーションメニューを使うことでフルスクリーンで表示できる。



画面15 プレゼンテーション表示状態では、マウスで書き込みも行える。

いえ皆無ではなく、こうした安全対策は必須といえる。また、現在のWineを使ったアプリケーションは、ウィンドウ操作の途中で画面がフリーズすることがときどきあり、こうした機能がないと長時間の作業をムダにする可能性が高い。Linux用のワープロはいくつも登場しつつあるが、こうした機能を持つものはあまり多くない。

ファイルのロード/セーブ時には専用のダイアログが表示され、ツリーの有無や新規ディレクトリ作成などの機能が利用できる。

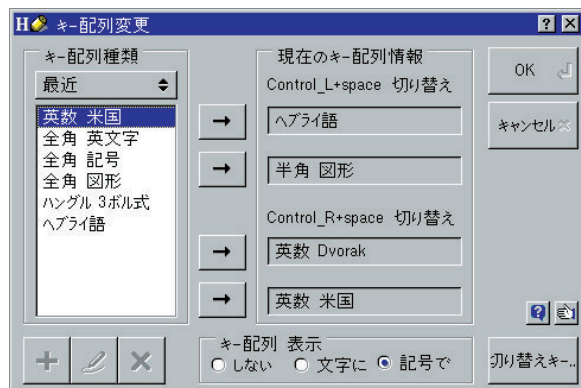
なお、Wine側の設定で自分のホームディレクトリがCドライブに、ルートディレクトリ以下がDドライブとして定義されている。また、ホームディレクトリにはワークフォルダとしてworkという名前のフォルダが作られる。

全体的な印象

今回評価したバージョンでは、ダイアログボックス内の表示やツールチップにハングル表示が出るなど、ちょっとした表示の不具合がみられる。

このほか、HTML形式でのセーブでは文字がすべて化けてしまう現象が見られた。セーブ用のダイアログでも、文字コードはシフトJISしか指定できず、多少ちぐはぐな感じがみられる。HancomWordオリジナルの形式でのロード/セーブは問題ないが、HTMLは比較的よく利用すると思われるので、

画面16 HancomWordは、いくつかの言語に対して入力方法を提供している。これは、キーボードレイアウトを変更する機能で、切り替えキーでキーボードレイアウトを最大4つまで切り替えることができる。なお、日本語の入力はこのキーボードレイアウトとは別になっており、Shift+Spaceキーで強制的に日本語入力モードとなる。



もう少しクオリティアップを望みたい。

また、このHancomWordだけではないのだが、Wineを使ったアプリケーションでは、下になっているウィンドウを移動させようとドラッグした場合などにX Window System全体がフリーズするなどの現象が時々みられる。CorelのWordPerfectでも同様の現象が見られたので、Wine側の問題（あるいは、X Windows Systemやウィンドウマネージャとの関係か）と思われる。それ以外では、HancomWordだけが動かなくなることもなかったため、このあたりがもう少し解消されればある程度実用になるといった感じである。

今年になり、日本語対応予定のものを含めて、Linux上のオフィスソフトが増えてきたようである。日本語対応予定の話があるのは、SunOffice（海外ではStarOffice）や、ApplixWareなどだ。また、ジャストシステムの一太郎ArkもLinuxでの動作実績があり、そう考えると、そろそろどれを使うか「選択」する時期に入りつつあるといえ

る。おそらくは、今年から来年にかけて、この中で安定性の高いもののシェアが伸びていくことと思われる。選択できるとはいえ、現状では機能よりも安定性（利用中にクラッシュしない）ことのほうが重視されるからだ。そうなると、日常的なちょっとした作業をLinux上のワープロで行うユーザーも増えてくるとと思われる。

Linuxをデスクトップで使うなんてことは、一昨年ぐらいは誰も考えなかったが、今ではもう夢物語ではない。

そんななかでのHancomWordの登場はタイムリーだ。国産のワープロソフトさえいくつも数がないなかで、韓国ソフトハウスが日本語ワープロに取り組んだ意欲は高く買いたい。韓国語版の長いノウハウに支えられて、最初からかなりの高機能を実現し、さらにアジア圏のワープロ事情にも精通しているアドバン스는大きい。9800円という手軽な価格も魅力だ。日本にオフィスを構えるなど、やる気も十分。今後に大いに期待したい。



画面17 現在選択中のキーボードレイアウトを表示させる機能もあり、このダイアログはソフトウェアキーボードとしても利用できる。ハブライ語のキーボードレイアウトを表示させたもの。

画面18 ハングル入力時のキーボードレイアウト。





にわか管理者に贈る!

root見栄講座

その管理! 間違ってます!

文: 竹内充彦

Text: Michihiko Takeuchi

いきなりシステム管理をまかされた!

突然、会社や学校で複数の人達が使うLinuxマシンのシステム管理をまかされた。これまでは一人で使ってきたので気にしたこともなかったが、そもそもシステム管理って何をすればいいのだ?

LinuxはもともとUNIXを手本にしたこともあり、マルチタスク・マルチユーザーOSである。Windowsとは異なり、複数の人が同時に1台のマシンを利用することを想定している。このとき、ユーザーが勝手にシステムの構成を変えられるようだと困った事態になってしまう。重要なファイルを勝手に削除されたり、いきなりシステムを再起動されたりしてはたまらない。

そのため、Linuxではシステム管理者と一般ユーザーとが明確に切り分けられ、システムの構成を変更できるのはシステム管理者だけに限定している。一般ユーザーが立ち入ることのできない部分、すなわちシステムの状態を維持・管理、あるいは変更するのがシステム管理者の役目なのだ。

これまで一人でLinuxを使ってきたのなら、ある部分のシステム管理はこなしてきたといえる。しかし、一人のときにはいい加減にやっていたことでも、複数の人を使うマシンでは間違った管理かもしれない。今までは、多少トラブルでも自分さえ我慢したり苦労すればよかった。しかし、これからはそうはいかない。自分のちょっとしたミスや怠惰が、ほかのユーザーの作業をフイにしまったり、イライラをつのらせてしまいかねないのだ。

ここはひとつ、「ちゃんと管理してますよ」というところを見せようじゃないか!



システム構築編

パーティションを 分けるべし!

システム管理の仕事は、システムを構築するときからすでに始まっている。Linuxを新たにインストールするときに、必ず登場するのがパーティションテーブル操作ツールfdiskだ(画面1)。Linuxでは、単一のファイルシステムで全体を構成することもできるが、複数のファイルシステムを使って全体を構築することもできる。

なんだかわからないからお勧めの分け方でいいやとか、面倒くさいから今までどおり1つでいいやなんて考えてないだろうか。たしかに個人での利用や、趣味性の強い利用の場合はそれでもかまわない。しかし、仕事や研究に利用するシステムの場合そうはいかない。ちゃんとディレクトリの意味と構成を考えてファイルシステムを構築しなければイケナイのだ!

各ディレクトリの意味を知っておこう!

そもそもLinuxのディレクトリ構成を把握しているだろうか?

```
Using /dev/hda as default device!
Command (m for help): p
Disk /dev/hda: 255 heads, 63 sectors, 524 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1 *           1           492    3951953+   83  Linux
/dev/hda2             493          524     257040    5  Extended
/dev/hda5             493          524     257008+   82  Linux swap

Command (m for help): █
```

画面1
多くのインストーラがインストール時にパーティション分割ツールを起動する。パーティションは1つでいいのだろうか?

一般的なLinuxのディレクトリ構成は図1のようにになっている。これらのディレクトリが、いったい何のためにあるのかをここで解明していく。存在理由がわかれば、ディスクのレイアウトにも自然と考えが及ぶようになる。また、ディレクトリ内のファイルについても次第に見当がついてくるはずだ。

あわせて、ディレクトリ名の読み方も紹介しておく。/etc/servicesファイルの場合、「スラッシュ イーティーシー」の下のサービスズというファイルに……」なんて言っているのはカッコ悪い。いっばしを気取って「スラ エトセ サービスズに」と言えば、ほかのシステム管理者と会話するときにも気後れしないはずだ。

/

Linuxのファイルシステムの頂点に位置するディレクトリだ。起動時に最初にマウントされる。通常、ここにはファイルを置いたりしないが、例外的にcshやGNOMEなどで使う設定ファ

イルの雛形が置かれている。

ここはツリー構造を持つディレクトリの根の部分に相当することもあり、ルートディレクトリと呼ばれる。ここを基点にパスを示す場合は「スラ」と言うことが多い。

bin

実行可能なコマンド(バイナリ: binary)を収めておくためのディレクトリがbinだ。/の直下にあるbinは、bash、cat、more、mkdir、tar、viなど、昔からある基本コマンドが収められている。逆に言えば、運用するにつれ消費量が増えるということはないだろう。

読み方は「ピン」。「bin」と名の付くディレクトリは/usr/binや/usr/X11R6/binなど、ほかの場所にもあるので、人に伝えるときには、明示的に「スラピン(/bin)」と言うこともある。

dev

Linuxでは、ハードウェアデバイスを仮想化し、システム側からはデバイスをファイルと同様に扱えるというUNIXの設計思想をそのまま受け継いでいる。プログラムを組む場合、ファイルを扱うのと同様にデバイスをOpenしてRead/WriteしてCloseするという手順でアクセスできるようになっているのだ(そのためUNIXやLinuxはプログラマーに人気がある)。その際に利用するデバイスファイルが収められているのがここだ。プログラマーでない一般ユーザーには、/dev/null(何もないデバイス)以外はあまり関わりがないかもしれない。初級管理者は、ディスクのマウントなどで意識することになるだろう。後から消費量が増えたりしない。

devはここ以外にないが、語感がいい

いためか、「スラデヴ (/dev)」と呼ん
だりする。

home

一般ユーザー用のホームディレクト
リが収められているのがここだ。通常
はこのディレクトリの下にユーザー名
が付いたディレクトリを作成し、そこ
をデータファイルの保存領域や作業領
域として使わせる。ユーザー数やユー
ザーの作業内容によって、このディレ
クトリに必要な容量は大きく異なる。
たとえば、Webサーバを起動している
場合、通常は、各ユーザーのWebペ
ージデータが /public_html、つまり
/home/<ユーザー名>/public_html
に置かれることになる。画像を置いて
いたりすれば当然ディスクの消費量も
増えるのだ。運用開始後に消費量が劇
的に増える可能性を秘めている。この
ディレクトリは例外的に「ホーム」と
省略されることが多い。

lost+found

ファイルシステムごとに自動的に作
られるので構築時に気にする必要はな
い。ファイルシステムがクラッシュし
た場合など、fsckコマンドでサルベ
ージされたファイルがここに収められ
る(詳しくは、本誌2000年4月号「Linux
日記」参照)。ここに拾い上げられた
ファイルの正体がわかるようになった
ら、もはやベテラン管理者である。こ
のディレクトリは「ロスト アンド ファ
ウンド」と読む。

mnt

フロッピーディスクドライブやCD-
ROMドライブなど、リムーバブルメ
ディアをマウントするためのディレク
トリーがmntだ。なので「マウント」と読
む。ここはマウントポイントになるデ

ィレクトリだけあればいいので消費量
もほとんどないし、増えない。

root

スーパーユーザーであるrootユー
ザーのホームディレクトリがここだ。シ
ステム管理者は、管理作業をする際に
suコマンドで一時的にrootユーザーに
なる機会が多くある。そのときに作業
用の一時ファイルを所かまわず作成し
てしまうと大変なことになる。何しろ
他のユーザーには、たとえそれがいら
ないファイルだとわかっている、消
すことはできないからだ。rootユー
ザーで作業するときにはここで作業す
るよう心掛けよう。

かつてはrootのホームディレクトリ
は/(こちらも「ルート」)だったのに、
今は/rootである。/と区別するために
「スラルート」と読むべきだろう。

tmp

一時的な(テンポラリ:temporary)
ファイルを置いておくディレクトリがこ
こだ。「テンプ」と読む。プログラムが
バッファの内容を一時的に書き出した
りするのに使われるが、誰でも読み書
き可能なため、ユーザーが一時ファイ

ルを作成することもできてしまう。し
かし、管理者としては、ユーザーがこ
こに必要なファイルを置いたりしないよ
うに警告する必要がある。システムよ
っては高速性を追求するために、/tmp
にメモリファイルシステム(RAMディ
スク)をマウントしていることがあるか
らだ。その場合は、マシンが再起動す
るたびに内容が失われる。自動的に消
えることを逆手に取って、いらぬフ
ァイルを置くツワモノもいたりする。

「tmp」という名のディレクトリも
いろいろな場所にあるので明示的に
「スラテンプ」と呼ぶこともある。

ちなみに、「テンプディレクトリ」と
いったときに、「temp」を思い浮かべ
るのはUNIXを使ったことがない
Windowsユーザーだ。

var

「バァー」と読む。ここも一時的に
ファイルを置いておくディレクトリだ
が/tmpとは事情が異なる。たとえばサ
ーバが受け取ったメールや、プリンタ
に送り出すデータ、システムの状態や
プログラムの動作を記録するログ(log)
ファイルなどが数多く収められている。
ログファイルは、システム管理者がシ

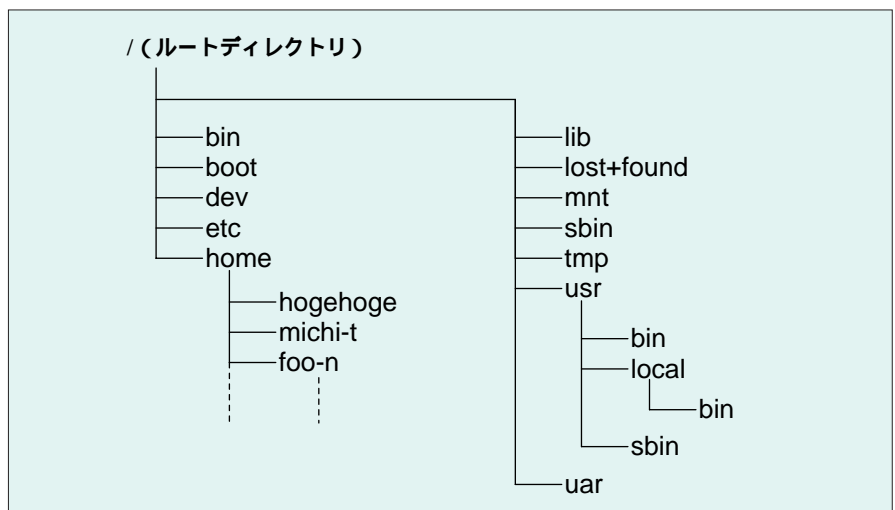


図1 標準的なLinuxのディレクトリ構成

システムの動作状態を知るのにとっても役に立つ。システムクラッシュの原因追求にも役に立つのだ。システムのサービス次第だが、メールやプリンタのプール量によっては運用後に劇的に増える可能性を秘めている。

ここにもtmpがある。/var/tmpだ。ここは/tmpと違いシャットダウン時にも内容が残っているため、最近ではviも一時ファイルをここに作る。編集中にトラブルがあってシステムが再起動を余儀なくされても、編集内容を復活させることができるからだ。

boot

Linuxのカーネルやシステムマップファイルが収められている。システム起動時にLILOからこのディレクトリにあるカーネルがメモリに読み込まれ、晴れてLinuxが起動する。つまり、Linuxの核が収められているディレクトリなのだ。そのまま「スラブート」と読む。

etc

名前からは想像がつかないが、ここにはシステムやアプリケーションで利用する重要な設定ファイルが数多く収められている。システム管理者が何かを設定するときには必ずここにアクセスすることになるだろう。

ラテン後の「et cetera (エトセトラ)」からきているため、「エトセ」と読む。/etc内のファイルは多種多様で

あり、すべてのファイルについて把握している人は少ないだろう。

lib

ここは、プログラムにリンクするライブラリファイルや、各種モジュールファイルが収められている。ユーザーや管理者がこの中に直接アクセスすることは少ないだろう。「リブ」と読む。

sbin

主にシステム管理に必要なコマンドが収められている。ここにあるコマンドはrootユーザーでしか実行できないので、一般ユーザーには無縁のディレクトリだ。管理者はここが/usr/sbinのコマンドによくお世話になる。「エスビン」と読む。

usr

ユーザーが利用するアプリケーションやそれに付随するファイルが収められている。

/usr/binにはEmacsをはじめとするユーザーコマンドがぎっしり詰まっているし、/usr/X11R6にはX Window System関連のファイルがすべて収められている。システムによっては、後からインストールする非標準的なアプリケーション関連ファイルが/usr/local以下にこれまたドッサリと置かれているかもしれない。とにかく、このディレクトリの下には、やたらと容量を消費するディレクトリがわんさかあるし、

アプリケーションをインストールしていくとアツという間に容量が足りなくなる。/homeと並んで消費量が多い。

読み方は「ユーザー」だ。もともとUNIXはキャラクタベースのOSであったため、なるべくタイプ量を減らすために省略された名称が多く使われている。「tmp」や「bin」などもいい例だ。



パーティションを分割する意味

ざっとディレクトリ構成を紹介したが、では一体なぜパーティションを分割するのだろうか？ これにはいろいろな側面がある。もちろん昔のUNIXシステムではハードディスクユニット自体の容量が少なかったため、複数台で構成せざるをえなかったという事情もあるだろう。しかし、今や80Gバイトのハードディスクが手頃な値段で購入できるのだ。それでもパーティションを分割する。実はハードディスクの容量とは関係ない。パーティション分割の本当の理由は、危険性の回避と管理のしやすさにあるのだ。

ファイルシステムクラッシュに備える

Linuxの運用において、常にファイルシステムがクラッシュする危険性がつきまとう。原因はハードウェアの故障であったり、ソフトウェアのバグであったり、突然の停電だったり、クラッカーの攻撃だったり、さまざまである。このとき、パーティションを分

Column

ピンチになったら スーパーユーザーに変身!?

システム管理者の特権であるスーパーユーザーroot。スーパーの名にふさわしく、一般ユーザーにできないことが何でもできてしまう。にわかシステム管理者がちょっとしたミ

スをしただけでも被害甚大だ。

たとえば、あるディレクトリを削除しようとして、間違えて/を削除してしまった……なんてことになったら目もあてられない。

従って、たとえシステム管理者であっても、ふだんの作業は極力一般ユーザー用のアカウントで行うようにしたい。そして、どうしても一

般ユーザーで対処しきれない場合には、suコマンドを実行して一時的にrootに変身するのだ。

suコマンドを入力すると、rootのパスワードを求められる。正しく入力すれば、その瞬間からあなたはスーパーユーザーだ。もちろん作業が終了したらexitで一般ユーザーに戻ることを忘れずに!

割しておくことにより、被害を最小限にとどめることができる。

たとえばパーティションがクラッシュしたとしよう。それも壊滅的に。/binや/usr以下のアプリケーションはバックアップデータやインストールCD-ROMから、元どおりに復旧することもできるだろう。しかし、/var/spoolにクラッシュ直前に受信したメールや、ユーザーが直前まで作成していた/home以下のデータは復旧することができない。もし、これらのディレクトリが別のパーティションにあれば、その領域だけでもクラッシュから逃れることができるかもしれないのだ(もちろん可能性の問題であり、原因が何かにもよるが)。また、/varが生き残っていれば、ログを参照することで、クラッシュの原因を特定することができるかもしれない。

Linuxは仮想記憶を使っている。物理的なメモリが足りない場合、ディスクをメモリの代わりに使う。メモリの内容とディスクの内容をとっかえひっかえして、あたかも広大なメモリ空間があるかのように振る舞うのだ。このメモリ内容とディスク内容の入れ替えをスワップと言う。そしてメモリ代わりに使うディスクの領域こそがswapだ。もし、スワップ領域が他のファイルシステム中にあると、スワップができなくなる危険性もある。

こうした危険性を考え、被害の拡大を防ぐためにパーティションを分割しておくべきだ。場合によっては、壊滅的なクラッシュをパーティションが食い止めてくれることもある。

ただし、いくらパーティションを分割しても、所詮同じハードディスクユニット内の話である。ユニットそのものが故障してしまえばいくら分割しておこうが意味はない。リソースに余裕

があるのであれば、ファイルシステムごとにハードディスクユニットも分けてしまうのが正しい。

ファイルシステムフルに備える

利用が進むうちに、ハードディスクの容量が足りなくなってきたなんてこともある。

たとえば、プログラムのコンパイル中に「File system is full」のメッセージが表示されて、makeが異常終了するなんてことが起きてくる。では、ディスクの容量を増やそうということになる。このとき、パーティションが分割されていると作業もやりやすくなる。容量不足のパーティションの内容をバックアップして、新しいディスクを用意し、フォーマットしてからマウントし、内容をリストアすればいい。システム全体が1つのパーティションの場合、非常にやっかいなことになる。ほとんど再インストールに近い。

またあとで紹介する、ユーザーごとのディスク使用量を制限する場合、単一のパーティションで構成されていると難しい。/を含むパーティションには制限を課すことができないのだ。従って、/と/homeなどを別パーティションにしておくべきである。

運用後に次第にディスクを消費していくであろう、/usrや/homeはこうした観点からも別パーティションにして

おき、手軽に差し替えが可能なようにしておく、後の苦勞が減る。

結局どう分ける?

以上のことを考えに含め、パーティションを分割しよう。ちなみに筆者はよく以下のように分割する。「なんだ某インストーラお勧めパターンと同じじゃん」などと言うことなかれ。同じでもいいのだ。意味を把握して分割することが重要なだから。

```
/
/usr
/var swap
/home
```

むしろ、にわか管理者のあなたにとって難しいのは、構築するシステムの/homeや/usrはどれぐらいのディスク容量を必要とするかの見極めかもしれない。この辺のことは、ユーザー数や作業内容、起動するサービスによってまるで違って来る。すでに周囲に何台かシステムが稼動しているならば、それを基に割り出すことも可能だ。しかし、今回が初稼動であれば見込み違いも十分ありうる。そんなときは新たなディスクを調達してマウントし直そう。そうすればもう「にわか管理者」から「見極めのできるちょっとだけベテラン管理者」にステップアップだ!

Column

PATHに「.」を入れちゃダメ!

bashにはコマンド検索パス(PATH)というものが設定されている。これは「set」と入力することで確認できる。ここに記述されているディレクトリにある実行形式ファイルは、フルパスで指定しなくても実行できる。便利な機能なのだが、rootのPATHにカレ

ントディレクトリ(.)を含めては絶対にいけない。検索順序によっては、「ls」コマンドを起動するつもりがカレントディレクトリにある偶然lsという名前の実行ファイルが起動してしまうかもしれないのだ。ひょっとすると、それがファイルを削除するコマンドだったり、パスワードファイルを書き換えるプログラムだったりするかもしれない。常に危険がつきまわっていることをお忘れなく!

ユーザー管理編

ユーザーはグループ化すべし!

ユーザーアカウントの作成は専用のツールを使えば簡単だ(画面2)。ところで、このときグループはどうしているだろうか? ふだん一人で使っている場合、グループについてはあまり考えていないことが多い。それではイカン!! グループはユーザーを効率良く管理するために欠かせない要素なのだ。初めからキチンとユーザーのグループ化を考えるようにしないと收拾がつかなくなり、トンデモないことになる。

たとえばlinuxconfでユーザーアカウントを作成する場合、グループ欄の入力は省略可能だ。グループを省略して作成するとどうなるだろうか。/etc/

passwdと/etc/groupの両ファイルはリスト1・2のようになる(それぞれのフィールドの見方については図2・3を参照)。このようにグループを指定しないとユーザーの数だけグループが作成されてしまうのだ。これではグループ化されていないのと同じである。つまりログイン名のみで管理すると変わらない。せっかくのグループ管理機能が使えなくなる。



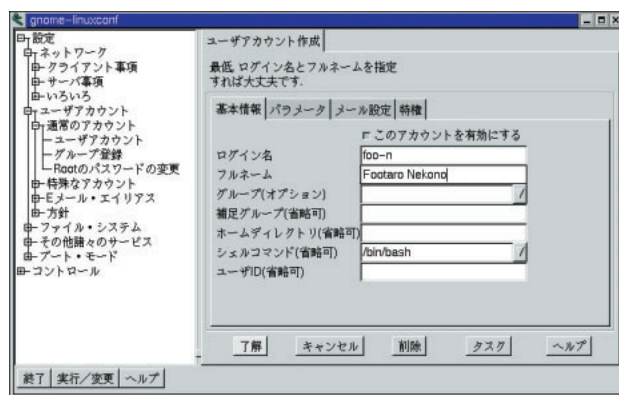
グループ化することの意義とは?

たとえば部署や研究室のようにチームで共同作業を進めているとしよう。

未発表のプロジェクトに関する論文や資料は、チーム内で共有したいが、ほかの人たちには見せたくないということがある(いわゆる部外秘というやつ)。このような欲求に応えるのがグループ管理機能だ。

プロジェクトチームのグループprj1を作成し、そのメンバー間でファイルを共有した例が画面3である。グループprj1に属するユーザーfoo-nの所有するファイルbugaihi.txtを「ls -l」で表示させたところだ。行頭にあるパーミッションが「-rw-rw----」になっているのに注目してほしい。所有者とグループに読み出し(r)と書き込み(w)のパーミッションが設定されている。ほかのユーザーには何も許可されていない(パーミッションの見方は図4参照)。

こうしておけばチーム内の秘密が他に漏洩することがなくなる。複数のユーザーが利用するマシンでは、こうした管理も必要になるのだ。



画面2
linuxconfによるユーザーの追加。グループ欄は省略することもできるが.....

グループの作成とユーザー登録

では、グループを作成しよう。新規グループの作成は、ほとんどのユーザーアカウント作成ツールで作成できる。作成したら、ユーザーアカウントの作成時にグループを正しく指定するようにする。

ユーザーは複数のグループに所属することができる。/etc/passwdに記述されているグループIDがそのユーザー

リスト1 グループを省略した場合の/etc/passwd (抜粋)

```
michi-t:x:500:500:Michihiko Takeuchi:/home/michi-t:/bin/bash
sunao-s:x:501:501:Sunao Shibahara:/home/sunao-s:/bin/bash
foo-n:x:502:502:Footaro Nekono:/home/foo-n:/bin/bash
```

リスト2 グループを省略した場合の/etc/group (抜粋)

```
:*:500:
:*:501:
:*:502:
```

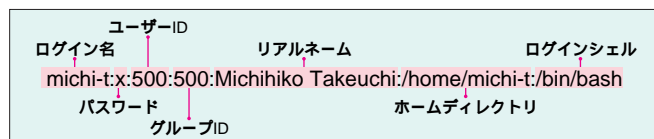


図2 passwdフィールドの見方

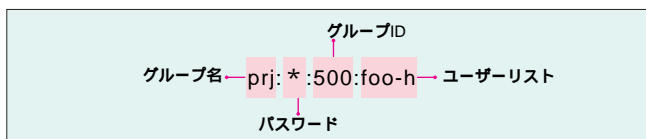


図3 groupのフィールドの見方

する。そしてグループの特殊ボタンである [このグループ権限を固定 (Set GID)] ボタンも押しておく。 [ファイル所有権] の [グループ] を正しく選んでおくことを忘れずに。

コマンドラインから作成する
コマンドラインから作成するには以下の手順で行う。

```
# mkdir /home/prj1
# chgrp prj1 /home/prj1
# chmod 2770 /home/prj1
```

chmodでGIDを指定する場合、8進ならば2000を足すか、文字ならば「g+s」で指定できる。覚えておこう。

SetGIDを指定したディレクトリは「ls -l」で見ると画面5のように見える。グループの実行パーミッションにsが付いているはずだ。

これでグループのユーザーがこのディレクトリにファイルを作成すると、所有者には作成者が、グループにはprj1が設定されるのだ。もちろん、ディレクトリを新たに作成してもSetGIDが継承される。便利！便利！



ユーザーに注意してもらうこと

共有ディレクトリもできて、これで共同プロジェクトもスムーズに……、と言いたいところだが、実はまだ問題もあったりする。ファイルの削除についてだ。もっともこれはシステム管理者のあなたではなく、共同作業をするグループのユーザーが注意することだ。システム管理者のあなたがユーザーにレクチャーすればあなたの株も上がるかもしれないぞ。

どういうことかということ、いくらファイルのグループに対するパーミッションを読み出しだけに指定しても、そのファイルが収められているディレクトリに書き込み権限 (w) があれば削除することができてしまうのだ。つまりグループのユーザーなら誰にでも削除できる。共有ディレクトリに置いた大切なファイルがグループ内の誰かに間違っって削除されてしまったなんてこともあり得るのだ。これを回避するにはファイルにStickyビットを設定する。

指定方法はSetGIDと同様である。GNOMEからなら目的のファイルのプ

ロパティの特殊ボタンにある [Sticky] ボタンを押す (画面6)。コマンドラインからなら、

```
# chmod 1640 bugaihi.txt
```

と指定する。Stickyビットを設定する場合、8進数なら1000を足すか、文字なら「+t」を指定する。

Stickyビットが設定されていれば、ディレクトリに書き込みのパーミッションがあっても、削除できなくなる。「ls -l」で確認すると、画面7のように表示される。パーミッションの表示の末尾にTが付いているのが目印だ。

重要なファイルにはStickyビットを活用するようにユーザーに理解させよう。

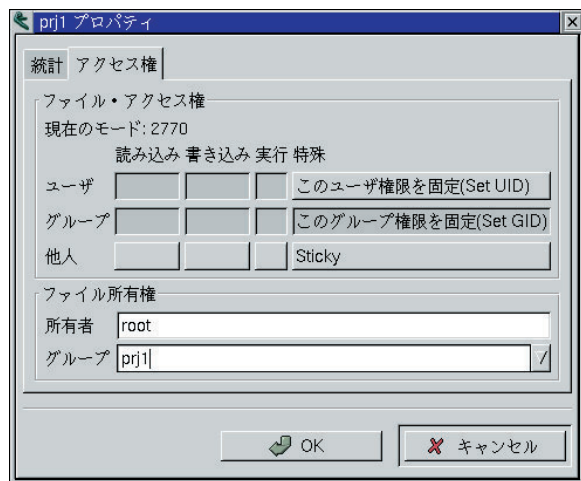
また、新しく作成されるファイルのパーミッションを自動的に設定したい場合は、umaskコマンドを使う。umaskコマンドは、パーミッションを8進数の補数で指定する。たとえば、660というパーミッションを設定したいのであれば、017 (777 - 660 = 017) を指定する。

```
$ umask 017
```

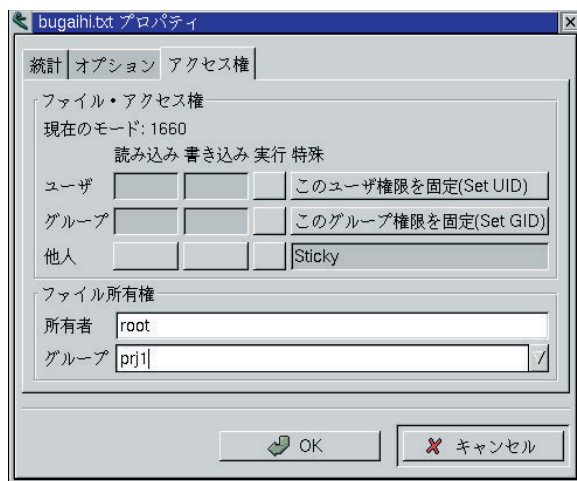
これで今後、新しく作成されるファイルのパーミッションが指定できる。

```
drwxrws---  2 root  prj1  1024 Jul 23 16:34 prj1/
```

画面5 SetGIDを指定した共有ディレクトリ



画面4 アカウントの作成画面



画面6 Stickyビットの設定



ユーザーがいなくなったら?

会社を退職したり、部署が異動になったり、卒業したりしたユーザーがいた場合、作成したユーザーアカウントをなんらかの方法で削除しなければならない。特に、会社を退職した場合などは早急にアカウントを削除する必要がある。退社した人を信頼していないわけではないが、システム管理者たるもの、つねにシステムの安全を最重視しなければならない。逆に、何かの問題が発生した場合、退社した人にいらぬ疑いがかかってしまう可能性もあるのだ。そういった要因を排除するのもシステム管理者の重要な仕事だ。

ユーザーの削除は結構大変である。単にユーザーアカウントを削除するだけではダメなのである。

最初にユーザーが作成したファイルがシステムに残っていないかを確認する。もし、残っているようだったら削除する必要がある。ユーザーはシステムのいたるところに痕跡を残しているハズだ。注意深く探すようにしよう。特に、/var/spool/mailにあるユーザーのメールスプール、/tmpや/var/tmpにあるファイル、cronやatコマンドのジョブなどもないか調べる必要がある。

全部探すのは大変だ? そんなときは、findコマンドを使う。

```
# find / -user hogehoge
```

これでhogehogeユーザーが所有権を持つファイルを全て表示することができる。不必要なファイルであれば削除し、必要なファイルであればファイルのパーミッションを変更したり、適切な場所へ移動しておく。次に、グループからユーザーを削除する。そしてユーザーアカ

ウントの削除だ。ユーザーの削除はrootユーザー権限で次のようにして行う。

```
# userdel -r hogehoge
```

これでユーザーhogehogeが削除された。「-r」オプションは、ユーザーhogehogeのホームディレクトリも削除する。しかし、システム管理者たるもの、ユーザーのホームディレクトリやファイルを削除する前に、念のためにバックアップを取っておくぐらいの慎重さが要だ。

ユーザーアカウントのロック

派遣社員や、異動などでユーザーがまた同じシステムに戻ってくる可能性がある場合は、いきなり削除しないほうがいいだろう。あるいは、会社を辞めた場合でも、「やっぱりこっちの会社がいいや」と戻ってくるかもしれない。「え? 削除されちゃったの?」と悲しい目で見つめられるかもしれない。そんなときは、ユーザーアカウントを削除せずにロックするといいたいだろう。ロックされたユーザーはユーザーアカウントはそのまま残っているものの、ど

んなパスワードを入力してもログインできない。ユーザーが再び戻ってきたときは、ロックを解除するだけですむ。

ユーザーアカウントをロックするには、passwdコマンドを使う。

```
# passwd -l hogehoge
```

```
Changing password for user hogehoge
Locking password for user hogehoge
passwd: Success
```

こうしておいて、ユーザーが決して自分の元には戻ってこないと確信してから、「さようなら」とユーザーアカウントを削除してもいい。

もし、ユーザーが「もう一度やり直したい」と戻ってきたら、「お帰り」とロックを解除してあげよう。

ユーザーアカウントのロックを解除する場合も、passwdコマンドを使う。

```
# passwd -u hogehoge
```

```
Changing password for user hogehoge
Unlocking password for user hogehoge
passwd: Success
```

これで元のさやに戻ったことになる。

```
-rw-r--r-T 1 foo-n prj1 10 Jul 23 16:34 hoge
Stickyビットが設定されている
```

画面7 Stickyビットが設定されたファイル

Column

削除されたユーザーのファイル

削除したユーザー名と同一の名前でもう一度ユーザーアカウントを作成すれば、以前のユーザーアカウントで作成したファイルにアクセスできるかもしれない。実はこれは間違いだ。Linuxはユーザー名でファイルのパーミッションをチェックしていない。UIDというユーザーIDによって管理されてい

る(本文図2参照)。このため、同一のユーザー名でも、ユーザーIDが違えば違うユーザーとして扱われる。逆に、新しいユーザーに古いユーザーIDが割り当てられると、以前のユーザーIDで作成されたファイルの所有権も割り当てられてしまう。このような混乱を避けるために、ユーザーIDは社員番号や学籍番号など、ユニークなものを使用するようにするのが上級システム管理者だ。くれぐれもユーザーIDを使い回さないように。

バックアップ編

面倒でも バックアップすべし!

一人でシステムを使ってきたユーザーにとっては、バックアップがなかなか身につけていないことが多い。とくにこれまでファイルシステムにトラブルを抱えたことがなければなおさらである。なんか面倒とか、バックアップデバイスがないとか、やり方がよくわからないとか、理由はさまざま。しかし、今やシステム内には多くのユーザーの作業の成果が収められているのだ。これをきちんとバックアップしておかないとイザというときにシステム管理者の面目丸つぶれだ。そんなことではイカン！バックアップはマメにやればだんだん楽になってくるのだ。

バックアップの種類

ちょっと乱暴な分け方ではあるが、バックアップにはフルバックアップと部分バックアップがある。ファイルシステムを隅から隅までバックアップするのがフルバックアップ（フルダンプとも言う）。対して、ファイルシステムの一部だけをバックアップするのが部分バックアップだ。

フルバックアップをしておけば、たとえファイルシステム全体が失われても元通りに復旧できる。その代わりにバックアップ作業自体に時間がかかる。

部分バックアップは、どこのバックアップを取っておくべきかという見極めが必要なのと、システムがすべて失われた場合、バックアップだけでは復

旧できないこともある。その代わりにバックアップ作業自体は時間もかからず手軽だ。主にデータファイルのみバックアップしておくのに使われる。

どちらも一長一短といったところかもしれない。しかし両者のいいところ取りをしたバックアップ方法もある。インクリメンタルバックアップだ。これは、まず最初にフルバックアップをしておき、その後は更新のあったファイルのみバックアップするというものだ。たとえば、運用次第だが数か月に1回フルバックアップをしておき、あとはその後の更新部分を日々バックアップしていくというような使い方ができる。

バックアップメディア

ここで紹介するツールはいずれも古くからあり、バックアップメディアも本来はテープを対象としている。しかしテープドライブは容量当たりのメディアの価格はともかくとして、ドライブの価格が思いのほか高い。それならいっそのこと、別に同容量のハードディスクを用意して、そちらにバックアップしてしまったほうが安い。容量の問題を抜きにしても、リムーバブルメディアとしてはMOやZipのほうが利用価値が高く、そちらの装備率のほうが圧倒的に高いと言っていい。

ここではそうした風潮に合わせて、テープメディアではなくディスクメディアへのバックアップを考えたい。幸

いバックアップツールはいずれもファイルに対して書き込めるのでディスク上にもバックアップ可能である。

dumpとrestore

インクリメンタルバックアップをサポートしているバックアップツールの定番がdumpとrestoreだ。dumpコマンドはバックアップレベルを指定できるようになっており、レベル0～9を指定する。次回バックアップ時に、同じかそれ以上のレベルを指定すると、前回バックアップ時からの更新情報のみをバックアップしてくれる。従って、レベル0を指定してフルバックアップを取り、以降1～9まで順に数を増やしていけばいいことになる。しかし、実際には指定したレベルよりも低いレベルでバックアップした日付を元に、その日以降に変更されたファイルをバックアップするので、正確に1ずつインクリメントしていく必要はない。フルバックアップに0、月イチや週イチのバックアップに3や6、毎日のバックアップに9などという使い方をしてもいい。

dumpでバックアップ

まずバックアップの例を示そう。フルバックアップには時間がかかるので、作業はユーザーが利用していない時間帯を選びたい。不意に誰かがログインしてこないようにシングルユーザーモードで作業を進めるといいだろう。それ以降のインクリメンタルバックアップでは、マルチユーザーモードでもかまわないかもしれないが、ユーザーが利用していない時間帯を見計らってサッサとやっておこう。

ここでは4Gバイトのバックアップ用ハードディスクをマウントしてそこにバックアップを取ってみる。ext2でフ

フォーマットされたハードディスク（ここではhdb1）を/mnt/bkupにマウントしよう。

```
# mkdir /mnt/bkup
# mount /dev/hdb1 /mnt/bkup -t ext2
```

マウントされたハードディスクに、/mnt/bkup/home_lv10.bakという名のアーカイブを作成し、/homeのバックアップを取るには画面8のように入力する。オプションの「0」はレベル指定、「u」はアップデート指定（バックアップ日時が/etc/dumpdatesに記録される）だ。「B」はメディアのサイズ指定だと思ってほしい。dumpのデフォルト設定では、2300ft.のテープに1600bpiで記録するので、1巻あたり約5Mバイトということだ。途中でテープ交換と言われても困るので（交換できないし）ここでは「B」オプションで適当にレコード数を広げている。

以降は、レベルを増やして差分のバックアップを取る。差分が少なければMOやZipにも取れるかもしれない。システム次第だ。

この方法には、/以下のバックアップが取れないという欠点もある。これは、バックアップ用のディスク自体もバックアップしようとしてしまうからだ。その場合はアーカイブファイル指定の代わりに、マウントされていないRawデバイスを指定しなくては行けない。つまり、ファイルシステムを作成せず、ディスクに直接データを書き込むのだ。この場合、アーカイブファイル名でなく「/dev/hdb1」のように指定する必要がある。というか本来はテープデバイスを相手に、そのように使うのが正しいのだ。しかし、ハードディスクでは巻き戻しや先送りができないので、つねに1つのバックアップデータしか書

き込むことができない。このため、Rawデバイスを指定する方法ではハードディスクの容量が大量に余る場合に無駄が生じる覚悟をしておこう。

restoreでリストア

さて、ここでの解説事項が役に立つ日が来ないことを祈りつつ、dumpでバックアップしたデータのリストア方法を紹介しておこう。

まず、リストアするディスクを準備する。

```
# mkfs /dev/hda2
```

次に、正しいマウントポイントにマウントする。

```
# mount -t ext2 /dev/hda2 /home
```

インクリメンタルバックアップのデータをリストアするには古い順にたどればよい。まずレベル0のデータをリストアする。コマンドラインは以下のよ

うになる。

```
# restore rf /mnt/bkup/home_lv10.bak
```

以降、順にレベル3、6、9のようにリストアしていく。このときrestoreはカレントディレクトリを基点にリストアするので注意が必要だ。/homeをリストアする場合、カレントディレクトリを/にしておかなければならない。

システムがクラッシュしていないとはいえ、リストアの方法は万一のときに備えて練習しておいたほうがいい。バックアップ用のハードディスクを買ってきたら、それをアーカイブ用とターゲットファイルシステム用にパーティション分割して、練習しておこう。



cpioを使ったフルバックアップ

容量の少ないメディアにバックアップを取るにはcpioが便利だ。もちろん大容量のハードディスクにフルバックアップを取るにのにも使えるが、それ

```
# dump -0uB 4000000 -f /mnt/bkup/home_lv10.bak /home
DUMP: Date of this level 0 dump: Mon Jul 24 00:09:56 2000
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/hda2 (/home) to /dev/hdb1
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 21501 tape blocks on 0.00 tape(s).
DUMP: Volume 1 started at: Mon Jul 24 00:10:03 2000
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: DUMP: 20282 tape blocks on 1 volumes(s)
DUMP: finished in 29 seconds, throughput 699 Kbytes/sec
DUMP: Volume 1 completed at: Mon Jul 24 00:10:32 2000
DUMP: Volume 1 took 0:00:29
DUMP: Volume 1 transfer rate: 699 KB/s
DUMP: level 0 dump on Mon Jul 24 00:09:56 2000
DUMP: DUMP: Date of this level 0 dump: Mon Jul 24 00:09:56 2000
DUMP: DUMP: Date this dump completed: Mon Jul 24 00:10:32 2000
DUMP: DUMP: Average transfer rate: 699 KB/s
DUMP: Closing /dev/hdb1
DUMP: DUMP IS DONE
#
```

画面8 dumpコマンドの実行例

はdumpのほうが便利かもしれない。cpioはcpコマンドのように手軽に使い、バックアップメディアの容量がいっぱいになった場合にそれを検出し、メディアの交換を促してしてくれるのだ。

ただしcpコマンドよりもシンプルで、ディレクトリを再帰的にコピーしてくれるようなオプションがない。従って、すべてのファイルに対してフルパスでファイル名を渡してやらなければならないのだ。げげっ、/usr/X11R6/bin/xeyesとかいうフルパスをすべてに指定するなんてやってられないぞ、と早まるなかれ。findを組み合わせて使うことでそんな心配は無用なのだ。findは「-depth」オプションと「-print」オプションを付けると、すべてのファイル名をフルパスで出力してくれる。

カレントディレクトリ以下のすべてのファイルを/mnt/bkupにマウントしたディスクのbkfileというアーカイブファイルにバックアップするには以下のように指定する。

```
# find . -depth -print | cpio -o > /mnt/bkup/bkfile
```

このとき、cpioに「-oH tar」というオプションを指定するとアーカイブファイルがtar形式になる。展開時にtarを使うこともできるのだ(ただし複数枚のメディアに分かれた場合はダメ)。

さらにfindの使い方、インクリメ

ンタルとはいかないが、更新されたファイルだけを拾い集めてバックアップすることができる。findには「-mtime」というオプションがあり、これに続けて+/-で日数を指定すると更新日付による抽出が可能になる。試しに、

```
# find / -mtime -3
```

と指定してみよう。すると、3日以内に更新されたファイルを見つけ出してくれるはずだ。先ほどのcpioとの組み合わせで使う場合、findにこのオプションを指定することで能率よくバックアップを取ることができる。ただしdumpと違うのは、消されたファイルの情報が反映されないで、リストア時に過去のバックアップをすべてリストアすると、大昔に消したはずのファイルまで復活してきて混乱するかもしれない。

さて、cpioでバックアップしたアーカイブファイルをリストアするには以下のように指定する。

```
# cpio -id < /mnt/bkup/bkfile
```

cpioのリストアでは、アーカイブ形式を自動判別してくれる。従ってtar形式でバックアップしようがcpioネイティブ形式でバックアップしようがリストア時のオプション指定は同じである。

cpioもリストア時のカレントディレ

クトリはアーカイブ作成時と合わせておかなければならないので注意しよう。



tarを使った部分バックアップ

部分的にバックアップを取るならば、tarコマンドを使うのが手軽だ。よくインターネット上からアーカイブファイルを持ってくると、「~.tar.gz」という拡張子が付いている。そう、あの形式でファイルをまとめてバックアップしようというのだ。

tarでフォルダ/home/foo-nをまるごとアーカイブして、gzipで圧縮したファイルを/mnt/moにマウントされたMOに保存するなら以下のように入力する。

```
# tar cfB - /home/foo-n | gzip -9 > /mnt/mo/foo-n.tar.gz
```

tarのオプション「c」はアーカイブの作成、「f」オプションは「-」とセットで標準出力への書き出しを意味している。「B」オプションはブロック化の指示だが、ここではパイプでつなぐ時のおまじないと考えておけばいい。gzipは「-9」オプションを指定することで、圧縮率を最大にしている。

このとき、対象となるフォルダを絶対パス指定で/home/foo-nとしているが、リストア時にも絶対パスで復元されることに注意すること。また、シンボリックリンクをたどって元のファイルをアーカイブに含めたい場合は「h」オプションを併せて指定する。

アーカイブの中にどのようなファイルが格納されているかを表示させるには、以下のように入力する。

```
# tar tfz /mnt/mo/foo-n.tar.gz | less
```

Column

rootはtelnetでログインできない!

telnetで離れた端末から管理作業をしようと思ったのに、ログインできない。万能なrootのはずなのになぜ? 確かにrootは万能なのだが、telnetによるログインはできない。

これは外部からrootのアカウントがハッキングされた場合に備えての安全策なのである。rootアカウントでは、セキュアな端末からしかログインできないように初期設定がなされているのだ。

この設定は変更できるが、お勤めはしない。普通にログインしてsuを使うべきだ。

これでアーカイブ内に格納されているファイルが一覧表示される。tarの「t」オプションはアーカイブの内容の一覧表示、「f」オプションはファイルからの入力またはデバイスを使用するということだ。「z」オプションはgzipへフィルタするという意味だ。つまり、tarからgzipを使うということになる。

このアーカイブファイルを展開するのは以下の手順だ。

```
# tar xfz /mnt/mo/foo-n.tar.gz
```

tarの「x」オプションは展開を意味している。後のオプションは先ほどと同じだ。

tarは、最近ではバックアップ以外にも使う機会が多いので、使い方をマスターしておくとかと便利だ。練習しておこう。



どこをバックアップすべきか?

フルバックアップはともかくとして、日常のバックアップはどの部分をしておくべきだろうか? 一般的には/homeや/var、せっせと設定した/etcがバックアップ対象にあげられる。/usr以下も/usr/localにアプリケーションを追加しているのであれば、バックアップ対象になる。

たとえば/binや/libは、インストール後に手を加えていなければ、インストールCDで十分バックアップになる。



定期バックアップは自動実行!

毎日のバックアップも基本的にはユーザーが利用していない時間帯にやる。いい心がけだ。しかし、そのために早朝出勤(登校)や居残りをしているのはイカン! いや、立派な行いではあるのだけれど。

毎日のバックアップは、容量も少ないのでバックアップメディアを交換する必要がない場合が多い。これは何を意味するかというと、メディア交換のために人が見張ってなくてもいいということになる。つまりあなたがシステムに張り付いてなくてもバックアップができてしまうということだ。

コマンドを定期的に行うにはcrontabを使う。Linuxでは常にcronデーモンが起動していて、crontabに記述したコマンドは、キッチリ指定の時刻にこなしてくれるという優れたものだ。

使い方は時刻指定以外は簡単だ。crontabにはいろいろと複雑な時刻設定ができる。まずはリスト5を参照してもらおう。これはレベル9のバックアップを午前4時2分に行うというものだ。行頭の空白で区切られた数字と*の並びが時刻の指定で、その右にコマンドラインが続く。時刻指定の並びは次のようになっている。

分 時 日 月 曜

これとリスト5を見比べれば「02 4」とあるのが午前4時2分なのはすぐに飲み込めるだろう。*の部分は指定なしということになる。つまり毎月、毎日、すべての曜日である。

ちなみにこの時刻指定が複雑なのは、範囲や割合を指定できるからだ。たとえば午前1時から午前6時までの間、10分ごとに、という場合は、

```
* /10 1-6 * * *
```

となる。/に続けて数値を指定するとその数値分スキップする。毎月15日の午

後10時30分なら、

```
30 22 15 * *
```

となる。曜日の指定は0~7で指定できる。0と7が日曜日に相当する。ウィークデイ(月~金)の昼12時を指定するのなら、

```
0 12 * * 1-5
```

となる。

さて、実際にcrontabを登録してみよう。リスト5のような内容をファイルとして作成しておき、

```
# crontab ファイル名
```

と指定すればいい。あるいは、

```
# crontab -e
```

と入力するとエディタが起動する(通常はvi)ので、その場で入力してもいい。登録されたcrontabは、

```
# crontab -l
```

で確認できる。crontabの機能自体は一般のユーザーでも利用可能である。

些末な話だが、どうもcronを設定するときに00分に指定するユーザーが多い。重たいコマンドを夜中の空いている時刻に実行しておこうというのはわかるのだが、なぜか2:00、3:00、4:00などキッカリの時刻に設定する。みんなが設定するので当然その瞬間にシステムは重くなる。2:20とか3:38とかに設定すれば早く終わるのにだ。

リスト5 crontab

```
02 4 * * * /sbin/dump -9uB 230000 -f /mnt/mo/home_lv19.bak /home
```

ファイル管理編

ディスク消費量を制限すべし!

運用が進むうちにディスクの消費量もかさんでくる。中にはMP3データやインターネットからダウンロードした巨大画像なんかを平気でホームディレクトリに置いているユーザーもいる。さらには、CD-ROMイメージをそのままホームディレクトリに抱え込んでいなんて不届き者もいたりする。

これではいくら大容量のディスクを装備しても追いつかない。なんとかしなければ。

ディスク大尽は誰だ?

まずはディスクの消費量を調べてみよう。とりあえず、どのパーティショ

ンがどれくらい消費しているのかを知るにはdfコマンドを実行すればいい。

```
# df
```

結果は画面9のように表示される。ファイルシステムごとに、全体の容量、使用量、残り容量、消費率、マウント位置が表示される。画面9では/homeが89%消費されており、残りわずかなのがわかる。

うぬぬう、許せん! いったい誰がこんなにディスクを消費しているのだ!? それを調べるのがduコマンドとsortとコマンドの組み合わせだ。/home以下をランキングするなら次の

ように指定する。

```
# du -s /home/*/ | sort -nr
```

結果は画面10のようになる。上から消費量の高い順になっている。こうしたユーザーには、不要なファイルを削除してもらうよう頼むか、当座必要のないファイルをリムーバブルメディアにバックアップしてもらう。

ディスク使用量の上限を設定しよう

あとからこうした事態にならないように、あらかじめユーザーごとにディスクの消費量に制限を課すことができる。Linuxにはquotaという機能があり、これを使うとユーザーごとにディスクの消費量に制限を課すことができる。

quotaの起動

まずはquotaをLinux起動時に機能させよう。/etc/rc.dの下にリスト6のようなrc.quotaというファイルを作る。作成したら、chmodでパーミッションを「rwxr-xr-x」に変えておこう。

次に/etc/rc.d/rc.localにリスト7の内容を追加しておく。さらに、quota

```
# df
Filesystem 1K-blocks    Used   Avail Capacity  Mounted on
/dev/hda1    893047    73182  748422     9%   /
/dev/hda2   2032623   1807847 220314     89%  /home
/dev/hda3   2047238   1292064 591395     69%  /usr
/dev/hda4    499303     6343  453016     1%   /var
```

画面9 dfコマンド実行

リスト6 /etc/rc.d/rc.quotaの内容

```
#!/bin/sh
if [ -x /sbin/quotacheck ]
then
    echo "Checking quotas. This may take some time."
    /sbin/quotacheck -avug
    echo " Done."
fi

if [ -x /sbin/quotaon ]
then
    echo "Turning on quota."
    /sbin/quotaon -avug
fi
```

```
# du -s /home/*/ | sort -nr
409784 /home/foo-n
4722 /home/httpd
4369 /home/michi-t
1564 /home/ftp
412 /home/sunao-s
1 /home/samba
```

画面10 ディスク消費者ランキング

リスト7 /etc/rc.d/rc.localに追加

```
if [ -x /etc/rc.d/rc.quota
]; then
    echo -n "Starting quota "
    /etc/rc.d/rc.quota start
    echo "."
fi
```

```
リスト8 /etc/fstabにusrquotaを追加
/dev/hda5      swap      swap      defaults 0 1
/dev/hda1      /          ext2      defaults 1 1
/dev/hda2      /home     ext2      defaults,userquota 1 1
/dev/hda3      /usr      ext2      defaults 1 1
/dev/hda4      /var      ext2      defaults 1 1
/dev/cdrom     /mnt/cdrom iso9660   user,ro,noauto,exec 0 0
```

を設定するファイルシステムのルート、つまり/homeのパーティションなら/homeの直下にquota.userというファイルを作成する。中身はあとから書き込むので必要ない。ただし、ユーザーに中身を見られないように、パーミッションを「rw-----」に設定しておこう。

最後に/etc/fstabに、どのファイルシステムにquotaを適用するかを記述する。ここでは/homeなので、リスト8のようにusrquotaの記述を追加すればいい。

ここまで済んだらシステムを再起動する。

quotaの設定

ではさっそくquotaを設定しよう。quotaの設定にはedquotaというコマンドを使う。foo-nというユーザーにquotaを設定したければ次のように指定する。

```
# edquota foo-n
```

これでエディタが起動するので、ユーザー名の次の行にリスト9のように指定する。quotaではディスクの消費量を、サイズとファイル数の両面から制限できる。blocksがサイズの指定（Kバイト単位）で、inodesがファイル数の指定である。それぞれlimitsにsoftとhardの記述がある。softは制限を超えると警告されるが、猶予時間を与えられるという、文字どおりソフトな制限だ。hardはまったくの限界で、それを超えてディスクを消費することはでき

ない。softとhardを組み合わせることもできる。ここではhard設定についてのみ指定している。20000Kバイト（=20Mバイト）とファイル数500という絶対的な制限にしてみた。

さて、quotaの設定が終わったら内容を確認してみよう。quotachekを実行してからrepquotaを使う。結果は画面11のようになる。これで設定したユ

ーザーは20Mバイトかファイル500個までの制限を受ける。

一般のユーザーからでも、自分に課せられたquotaだけはチェックすることができる。quotaコマンドを入力すればいい。画面12のような結果が表示される。ファイルが作成できない旨のエラーが出た場合はquotaを確認してみるように勧めよう。

```
リスト9 edquotaで記述する内容
Quotas for user foo-n:
/dev/hda2: blocks in use: 1, limits (soft = 0, hard = 20000)
          inodes in use: 1, limits (soft = 0, hard = 500)
```

```
# quotacheck -a
# repquota -a
```

User	used	Block limits			File limits			
		soft	hard	grace	used	soft	hard	grace
root	-- 302	0	0		25	0	0	
michi-t	-- 101	0	20000		82	0	500	
sunao-s	-- 71	0	20000		72	0	500	
foo-n	-- 1590	0	20000		280	0	500	

画面11 quota設定の確認

```
$ quota
Disk quotas for user foo-n (uid 502):
```

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hda2	1590	0	20000		280	0	500	

画面12 一般ユーザーからのquota確認

Column

ディスクの空き容量はこまめにチェック!

ユーザーから、「File system is fullでファイルが作れない!」などと言われないように、常にディスクの空き容量には注意を払ってお

こう。管理者として恥ずかしすぎる。どこかにcoreファイルなどの不必要なファイルがないか、logファイルが巨大になりすぎていないか、誰かが異常にディスクを圧迫していないかなど、つねに目を光らせていなければならないのだ。

プロセス管理編

常に監視すべし!

なんかシステムが重い……。実行結果がなかなか表示されない。なんだか知らないけど、ちょっと待ってればすぐに収まるだろう。いやいや、ちょっと調べてみようではないか! その昔 sendmail を利用したインターネットワーク事件が起きたときにも、システムが急激に重くなり始めたことに気が付いたシステム管理者が、最初に事の重大さに気が付いたのだ。

Linux はマルチタスクのシステムであることは冒頭でも述べた。複数のプロセスに、人間にはわからないほど小

刻みに分けたCPUの作業時間を割り当てて、あたかも同時に動いているかのように見せている。しかし、タスク数が多くなると刻んだCPUの作業時間が1つのタスクにまわってくるまでに時間がかかるのは当然だ。そうなると、システムが重いと感じるようになる。

また、メモリ消費量が多い場合も重く感じるはずだ。スワップについては述べたが、スワップの頻度が多くなると、ディスクアクセスが余計に増えることになる。極端な場合、メモリ内容とディスク内容の入れ替えの作業自体に忙しくなり、本来のタスクがなかなか進まなくなる。このような状態をスラッシングと呼ぶ。

こんな状態ではユーザーも仕事にならない。管理者の出番だ。

ロードアベレージを監視しよう!

システムにどのくらいの負荷がかかっているかを調べるために、ロードアベレージを監視しよう。ふだん、GNOME を使っているのであれば、xload (画面13) を起動するか、GNOME のパネルにロードアベレージ表示アプレットを置くようにしよう (画面14)。グラフの山が高い状態ならばCPUの負荷が高いということだ。

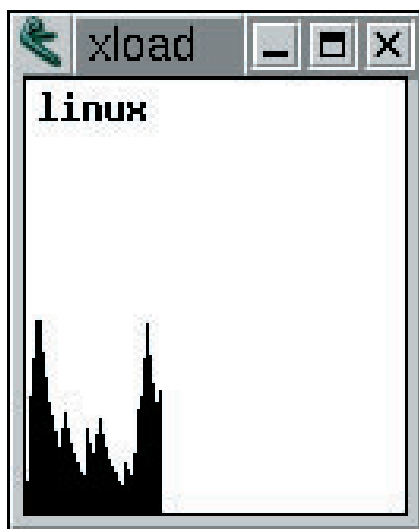
グラフの山が高くなっているようなら、もう少し詳しく調べるほうがいいだろう。CPUのここ数分間の負荷を調べるには、w コマンドを使う。一般ユーザーでも実行可能だ。

\$ w

結果は画面15のようになる。まず1行目に注目しよう。左から現在時刻、システムが最後に起動してから今までの日数と時間、現在ログイン中のユーザー数、そして過去1分、5分、15分のロードアベレージが表示されている。

そして、2行目以降は端末ごとのCPU負荷を示している。ユーザー名 (USER)、端末 (TTY)、どこから接続しているか (FROM)、ログイン時刻 (LOGIN@)、何もしていない時間 (IDLE)、その端末から消費されたCPUタイムの合計 (JCPU)、WHAT に表示されているプロセスが消費したCPUタイム (PCPU)、現在のプロセス (WHAT) となっている。

CPUタイムは、末尾にsが付いてれば秒数、:で区切られている場合はmm:ssで分:秒の表示になる。画面15では、ユーザーfoo-nがCPUタイムを消費していることがわかる。



画面13 xload



画面14 GNOMEアプレット

```
$ w
5:33pm up 14 days, 1:57, 3 users, load average: 8.04, 7.30, 5.22
USER  TTY  FROM          LOGIN@  IDLE   JCPU   PCPU   WHAT
michi-t pts/0  -             3:03pm  37:00s 0.16s  0.13s  -bash
foo-n  pts/1  192.168.0.2   5:08pm  0.00s  9:13   5:21   bash
sunao-s pts/2  192.168.0.3   2:34pm  1:24   0.13s  0.03s  bash
michi-t pts/3  -             5:13pm  0.00s  0.10s  0.05s  w
```

画面15 wコマンド


```
$ ps U foo-n
  PID TTY          STAT       TIME COMMAND
 11739 pts/3        S           0:00 -bash
 11758 pts/3        R          37:23 latex
 11759 pts/3        R           1:02 latex
 11762 pts/3        R           0:20 platex
```

画面16 psによるユーザーごとのプロセス表示

```
$ free
              total        used         free       shared    buffers     cached
Mem:          127904       124604         3300        67348        2620       47336
-/+ buffers/cache:    74648         53256
Swap:         257000       134098       122902
```

画面17 freeコマンドによるシステム情報の表示



不要なプロセスを終了させる

いくらシステムに負荷がかかるといっても、必要なプロセスならばかたがない。まずはどんなプロセスが実行されているのかを調べる必要がある。プロセス単位でのCPU負荷を探ってみることにしよう。

プロセス単位で情報を表示するにはpsコマンドを使う。通常は自分が実行しているプロセスが表示されるが、ここでは、ユーザーfoo-nの実行しているプロセスをすべて表示してみよう。ユーザーを指定するには「U」オプションに続けてユーザー名を指定する。

```
$ ps U foo-n
```

結果は画面16のようになる。PIDはプロセスを特定するためのIDで、プロセスごとに固有に振られている。TTYはどの端末から実行されているかということだ。STATはプロセスの状態を示している。Rは実行中、Sはスリープ(待機)で、Wはスワップアウトされている。Tは中断中で、Zはゾンビ(親プロセスが正しく終了処理をしなかったりして残ってしまったプロセスの残骸)だ。TIMEは累計のCPU利用時間だ。

プロセスを終了させるkill

画面16で言えばプロセスID11758が怪しい。ユーザーに確認したら本人は

止めたつもりようだ。終了したのではなくサスペンドしているだけだろうか? 了解を取ったら、強制的にプロセスを終了させよう。

プロセスを終了させるにはkillコマンドを使う。killはプロセスにシグナルを送るというコマンドだが、これを利用して終了シグナルを送ればいいのだ。まずは通常の終了を促すTERMシグナルを送る。

```
# kill -TERM 11758
```

通常はこれで終了する。しかし中にはタチの悪いプロセスもあり、これでは終わらないこともある。そういうときは強制的に終了させるKILLシグナルを送る。

```
# kill -KILL 11758
```

これで終了するはずだ。ただしKILLシグナルで強制的に終了させるとゾン

ビが残ったりすることもあるので、最初はTERMシグナルで終了を試みるようにしたい。ゾンビとして残ったプロセスはリブートするまで消えない。ちなみに殺しても死なないからゾンビという。冗談のようだが、UNIXの正しい用語だ。



メモリの状態を知る

プロセス以外にもシステムを重たく感じさせる要素としてメモリ不足がある。実際、自分のシステムのメモリ消費量も気になるところだ。これはfreeコマンドで確認できる。

```
# free
```

物理的なメモリ容量と、使用量、空き領域、カーネルが利用している共有メモリ、バッファ、キャッシュの情報も併せて表示される(画面17)。そして最下行にはスワップについての情報

Column

ログイン時にユーザーに告知したい!

システムを利用するユーザー全員に呼びかけるにはいくつか方法があるが、ログイン時にメッセージを出すというのは効果的だ。/etc/motdに書き込んだ内容は、毎回ログイン時に表示される(画面18)。

定期バックアップのお知らせや、不要なファイル削除の呼びかけなど、使い方はいろいろある。活用するようにしよう。



画面18 /etc/motdの内容はログイン時に表示されるのでユーザーに注意を促せる

も表示されている。スワップ領域の消費量があまりにも多いようであれば、メモリの増設を考えたほうがいいのかもれない。それだけメモリからはじき出されているプロセスがあるのだ。



リアルタイムに監視する

ここまでで紹介したプロセス管理やシステムの状態を常時監視するにはtop

コマンドも使える。topコマンドは常に新しい情報を画面に表示し続けるので、もしデスクトップが広ければどこかに表示させておくといい。

topの起動は簡単で、

```
# top
```

ただだ。すると端末画面いっぱいに行中のプロセス情報が表示される(画面19)。リアルタイムに表示が変わっていくのがわかるはずだ。この情報を見てシステムが現在どのように稼働しているかを把握できるようになれば、にわか管理者を脱したといえるだろう。

それでは画面の最初の5行に表示される内容の説明を順にしていこう。

最初の行は左から、現在の時刻、「up」の次の数値はシステムが起動してから時間、ログインしているユーザーの数、「load average」はシステムの負荷状態だ。順に1分、5分、15分間に実行を待っていたプロセスの平均数が表示されている。

2行目はプロセスの状態を表している。「processes」は実行されているプロセスの合計だ。この行には、スリープ中(sleeping)、実行中(running)、ゾンビ状態(zombie)、停止中(stopped)のプロセス数が表示される。

3行目はCPUの状態だ。ユーザー(user)やシステム(system)がどのくらいのCPUタイムを使用したかが表示される。「nice」とは、タスクの優先度(通常は0)、「idle」はシステムの待ち状態の割合を示している。

続く2行はメモリの使用状況だ。順に、メモリの総量、メモリ使用量、空きメモリ、共有メモリ、バッファに使用されているメモリ、スワップメモリ総量、スワップメモリの使用量、スワップメモリの空き、キャッシュメモリ

画面19 topはリアルタイムにプロセスやメモリの状態を監視できる

パラメータ	意味
PID	タスクごとのプロセスID
USER	タスクの所有者(実行者)
PRI	タスクの優先順位
NI	タスクのnice値。通常は0。負のnice値は優先順位が低い
SIZE	タスクが使用しているコードとデータとスタック領域の合計(キロバイト)
RSS	タスクが使用している物理メモリの使用量(キロバイト)
SHARE	タスクによって共有されるメモリ量
STAT	タスクの状態。Dは連続スリープ状態、Rは実行中、Sはスリープ状態、Tは停止中またはトレース中、Zはゾンビ状態を表す
LIB	使用ライブラリページの数
%CPU	プロセッサごとの全CPUパワーに対するタスクが使用したCPUパワーの割合
%MEM	タスクが使用している物理メモリの割合
TIME	タスクがスタートしてから使用した全CPUタイム
COMMAND	タスクのコマンド名

表1 topコマンドで出力されるパラメータの意味

コマンド	意味
スペース	最新の情報に更新
Ctrl + L	画面表示が乱れたときの書き換え
hまたは?	ヘルプの表示
u	ユーザー別プロセス表示。ユーザー名を問い合わせる
k	プロセスの終了。TERMを試みて、ダメならKILLで終了する
i	待機中のプロセスを表示/非表示
#数字	画面に表示するプロセス数を指定する
r	renice。実行優先順位を変える
l	ロードアベレージの表示/非表示
m	メモリ情報の表示/非表示
t	プロセス情報の集計を表示/非表示
c	コマンドラインのフルパスを表示/非表示
M	プロセスのランキングをメモリ使用率順に並べる
P	プロセスのランキングをCPU使用率順に並べる
T	プロセスのランキングを累計実行時間順に並べる

表2 topの主なコマンド

の使用量になっている。

スワップメモリの使用量が増えくると、ディスクアクセスが頻繁に発生し、システムが遅くなっていくので注意が必要だ。

残りの行に表示されるのは、現在実行中のタスクのリストで、CPUに負荷をかけている順だ。上にあるほど負荷がかかっている。表1は各パラメータの意味だ。

画面19では、rootユーザーが実行しているXが最もCPUパワーを使っているということになる。なんと、メモリの40%近くをXが占領している。CPUパワーも7%近く占有している……などということが瞬時にわかるのだ。topコマンドで表示される情報はリアルタイムで変化するのでなかなか興味深いものがある。コマンドやプログラムがどのくらいシステムに負荷をかけるのかがわかれば、どのくらいのメモリを搭載すればよいのかもおのずとわかってくるだろう。

topは対話的にコマンドを受け付ける。主なコマンドは表2のようになっている。



より詳しくシステムの負荷を調べる

メモリの使用率やスワップの発生をより詳しく調べるには、vmstatコマンドが便利だ。vmstatコマンドは多くのUNIXにあるコマンドで、古くからシステムの負荷を調べるために使われている。vmstatはシステムを監視する間隔と回数をオプションに指定して実行する。たとえば、5秒間隔で10回調査するには、

```
$ vmstat 5 10
```

とする。1回目はシステムの起動時か

らの平均値が表示される。なので、少なくとも2回以上の調査をするようにしたほうがいだろう。vmstatは多くの情報を提供するのできっとシステムの状態を把握するのに役立つはずだ(画面20)。

表示される情報は表3のようになっている。

vmstatは標準出力に結果を出力するので、ファイルなどにリダイレクトすればログとして残すことができる。たとえば、

```
# vmstat 3600 >/var/vmstat.log &
```

としておけば、1時間ごとのシステムの

負荷の記録を取ることができる。記録したログから、システムの負荷状態を分析することができるだろう。

特に、cronやatなどで実行するコマンドを指定している場合は、実行される時間帯の負荷を調べておくことは重要だ。ほかのユーザーも何かを実行しているかもしれないので、知らない間に負荷が高くなっていることがある可能性がある。その時間にログインしているユーザーも調べておくといい。

なお、必要なプロセスのみが実行されている状態なのに、メモリの使用率やスワップの発生率が高い状態が続くようであれば、メモリを増設するなどの対策が必要になる。

```
$ vmstat 10 10
procs          memory      swap          io           system      cpu
 r  b  w  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id
0  0  0  9396 4284 13204 16272  0  0  0  0  17  14  0  0  13
0  1  0  9396 4144 13236 16272  0  0  2  3 141  28  0  0  99
1  0  0  9396 1152 16384 16052  0  0 153 18 1475 330  1  3  96
1  0  0  9396 3292 17848 12440  0  0 132 48 1538 307  1  3  96
0  1  0  9396 1476 19748 12388  0  0 114 71 1576 263  1  3  96
1  0  0  9396 1288 20784 11536  0  0 126 43 1456 275  1  4  95
1  0  0  9396 1436 20848 11312  0  0 146 44 1625 321  2  6  92
1  0  0  9396 1004 21292 11312  0  0 133 41 1492 280  1  3  96
2  0  0  9396 1040 21252 11312  0  0 137 47 1571 311  2  5  94
1  0  0  9396 1404 20896 11312  0  0 138 47 1585 307  1  2  96
```

画面20 vmstatはシステムの状態を出力する

パラメータ	意味	
procs	r	実行待ちプロセスの数
	b	連続スリープ中プロセスの数
	w	スワップアウト中で、実行できるプロセスの数
memory	swpd	使用中の仮想メモリの総計(キロバイト)
	free	アイドル状態のメモリの総計(キロバイト)
	buff	バッファとして使用しているメモリの総計(キロバイト)
swap	si	ディスクからメモリにスワップした総計(キロバイト/秒)
	so	メモリからディスクにスワップした総計(キロバイト/秒)
io	bi	ブロックデバイスに転送したブロック数(ブロック/秒)
	bo	ブロックデバイスから転送されたブロック数(ブロック/秒)
system	in	1秒あたりの割り込み数
	cs	1秒あたりのコンテキスト切り替え数
cpu	us	ユーザータイム(全CPUタイムのパーセンテージ)
	sy	システムタイム(全CPUタイムのパーセンテージ)
	id	アイドルタイム(全CPUタイムのパーセンテージ)

表3 vmstatコマンドで表示されるパラメータの意味

ソフトウェアインストール編

RPMで 簡単インストール!

運用が始まると、次第にユーザーからクレームがつく。ソフトウェアがインストールされていないと言われたり、バージョンがよそのシステムより古いと言われたりするようになってきたりする。「なくても気合で何とかしなさい!」とか、「古くても我慢しなさい」などとは言えない。システム管理者は、ちゃんとソフトウェアのインストールやアップデートをしないとイカンのだ。

ソフトウェアのインストール

Linuxでソフトウェアのインストールという難しいようなイメージがあったのは、今はもう昔の話である。ソフトウェア開発者の努力により、配布されるソフトウェアはパッケージ化されており、誰でも簡単にインストールできるようになっている。バイナリファイルで配布されているものはコンパイルというフェイズすら省略しているお手軽さだ。

ソースプログラムで配布しているも

のでも、ユーティリティにより、gccを使ってコンパイルし、正しいディレクトリに置き、設定ファイルを配置するといった作業が自動化されている。パッケージファイルさえ入手できれば、インストール作業自体は、実は手間いらずなのだ。

では、パッケージをどこで手に入ればいいのか? それも簡単! インターネット上にもたくさんあるし、本誌付録のCD-ROMの中にもギッシリ詰まっているぞ!

とっても簡単RPM形式

Linuxにソフトウェアをインストールする方法はいくつかある。最近ではRPM (Red Hat Package Manager) 形式というパッケージファイルで配布されているものが多い。RPM形式はインストールとアンインストールが簡単、バージョン管理が楽ちんといふことづくめなので人気がある。なんといつてもインストール後の諸設定までしてく

れたりするのがいい。

そのRPM形式のパッケージを扱うコマンドがrpmだ。rpmは多機能なコマンドで、豊富なオプションを持っているが、ふだんよく使うものは表4に示すものぐらいだから覚えておこう。

rpmコマンドを使って、RPM形式のパッケージファイルからソフトウェアをインストールするには、

```
# rpm -ivh ファイル名
```

と入力すればいい。

パッケージのファイル名は比較的わかりやすく付けられている(図5)。パッケージ名とバージョンがハイフンで区切られ、バージョン名の後ろにハイフンがあれば続いてリリース、そしてピリオドで区切られたアーキテクチャ名、そして最後に拡張子(.rpm)が付く。

この図はWebサーバApacheのものだが、バージョンは1.3.12でLASER5 Linux用にリリースされていることを示している。

アーキテクチャは、i386とあればPC用にコンパイルされたバイナリパッケージを示す。コンパイルせずにすぐインストール可能だ。srcとあればソースプログラムで提供されるパッケージである。コンパイルが必要だが、通常はrpmコマンドがインストール時に勝手にやってくれる。

これらを踏まえ、インストールを指定すると、

```
# rpm -ivh apache-1.3.12-2LL2.i386.rpm
```

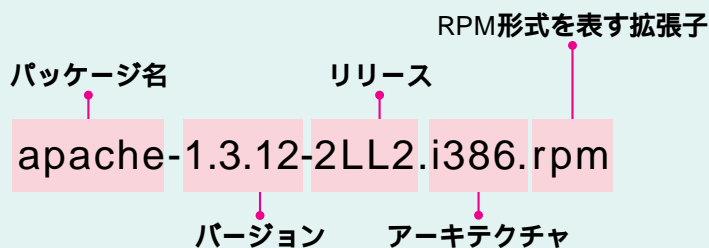


図5 RPM形式のパッケージファイル名

オプション	意味
-i	パッケージのインストール
-U	パッケージのアップグレード
-e	パッケージの削除
-q	パッケージに関する情報を得る

表4 rpmコマンドの主なオプション

と指定すればいい。rpmコマンドでインストールすると、そのインストール情報は/var/lib/rpm/にあるデータベースに登録される。この情報はアンインストール時や、パッケージの情報を得る時に利用される。

インストールしたソフトウェアをアンインストールするには、

```
# rpm -e パッケージ名
```

と入力する。パッケージ名(たとえば「apache」)を指定すると、rpmコマンドがデータベースから検索し、パッケージに含まれるファイルをすべてうまくアンインストールしてくれる。



古いソフトをアップデート

ソフトウェアのバージョンを管理するのも重要な作業だ。運用形態によっては、いたずらに最新バージョンを追えばいいというわけではないが、バグの修正やセキュリティの向上などメリットはたくさんある。可能な限り最新バージョンにしておいたほうが何かと安心だ。

バージョンの調べ方

rpmを使って現在インストールされているソフトウェアのバージョンを調べるには、

```
# rpm -q パッケージ名
```

と入力する。もちろんrpmのデータベースを検索しているので、残念ながらrpmコマンドを使ってインストールしたソフトウェアについてしか調べることはできない。

パッケージ名は、モジュールの先頭の名前を指定すればいい。たとえば

gimpのバージョンを調べると以下のようになる。

```
# rpm -q gimp
gimp-1.0.2-rh52_L5_1
```

現在インストールされているのは、gimpのバージョン1.0.2であることがわかる。その後ろに付いているのは、どうやらRed Hat Linux 5.2とLASER5 Linux用にリリースされたバージョンのようだ。

gimpのサイト<http://www.gimp.org/>を調べてみると、現在gimpは1.1.23のRPM形式がリリースされているので、ここは新しいバージョンにアップデートしてみることにしよう。

アップデート

現在インストールされているソフトウェアをアップデートするには、

```
# rpm -Uvh ファイル名
```

と入力する。

本誌先月号付録のLASER5 Linux 6.2のディストリビューションCD-ROMから1.1.23を持ってきて、アップデートすることにする。ディストリビューションCD-ROMをマウントして、/mnt/cdrom/LASER5/RPMSの中か

らRPMパッケージを持ってくる。

以下のように入力すればアップデートされる。

```
# rpm -Uvh gimp-1.1.23-1LL1.i386.rpm
```



rpmその他の使い方

rpmコマンドは、インストールやアップデートに使う以外にも役に立つ。ここでいくつか例を紹介しておこう。

ディレクトリの中を見ていて、これは何のファイルだろう? と思ったことはないだろうか。manを参照してもわからない。もしそのファイルがRPMパッケージからインストールされたものであれば、rpmコマンドで調べることができる。

たとえば「libjs.a」というファイルがあったとする。このファイルは何が利用しているのだろうか?

```
$ rpm -qf libjs.a
```

```
kdesupport-1.1.2-31v2
```

「あ、なんだKDEサポートで利用するファイルだったのか!」という具合だ。気になるファイルは検索してみよう。

逆に、Apacheのパッケージがどんなファイルをインストールしているか

```
kterm
[root@linux /root]# rpm -qi Mesa
Name           : Mesa                               Relocations: (not relocatable)
Version        : 3.2                               Vendor: Red Hat, Inc.
Release        : 1                               Build Date: Thu 03 Feb 2000 01:40:47
  AM JST
Install date:  Mon 24 Jul 2000 10:45:55 PM JST   Build Host: porky.devel.redhat.com
Group          : System Environment/Libraries   Source RPM: Mesa-3.2-1.src.rpm
Size           : 4443139                          License: GPL/XFree86
Packager       : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL            : http://www.mesa3d.org/
Summary        : A 3-D graphics library similar to OpenGL.
Description    :
The Mesa 3-D graphics library is a powerful and generic toolset for
creating hardware assisted computer graphics. To the extent that Mesa
utilizes the OpenGL command syntax or state machine, it is being used
with authorization from Silicon Graphics, Inc. However, the author
(Brian Paul) makes no claim that Mesa is in any way a compatible
replacement for OpenGL or associated with Silicon Graphics, Inc. Those
who want a licensed implementation of OpenGL should contact a licensed
vendor. However, Mesa is very similar to OpenGL, and you might find
Mesa to be a valid alternative to OpenGL.
[root@linux /root]#
```

画面21 Mesaについての説明が表示される。正体不明のパッケージはなくなる!?

については、

```
# rpm -ql apache
```

と入力して知ることができる。げげっ、こんなにたくさんのファイルをインストールしてるのか！と認識を新たにすることもかもしれない。

インストールされているパッケージがいったい何のためのものなのか知りたくなることがある。そんなときはやはりrpmのデータベースに問い合わせるといい。Mesaって何するものか？と思ったら、

```
# rpm -qi Mesa
```

と入力する(画面21)。すると、インストール情報に含まれるパッケージの解説が表示される(ただし英語だ

が.....)。「あ、OpenGL関連のグラフィックライブラリなのね！」ということがわかるのだ。



GnoRPMを使う

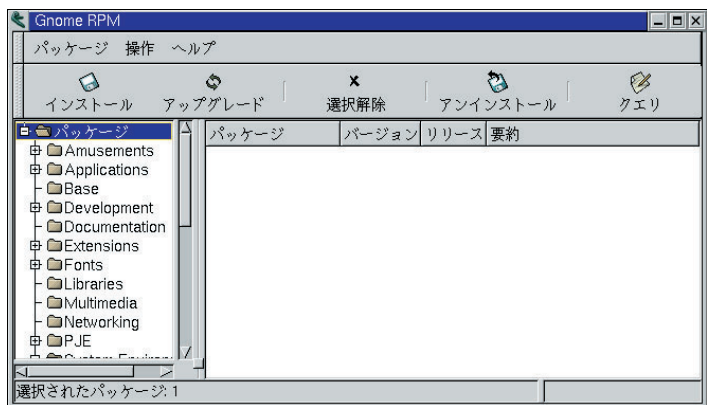
RPM形式のパッケージならインストールが簡単なことがわかってもらえただろうか。しかし、コマンドラインからの利用はどうも.....というにわか管理者諸氏には、GnoRPMというものもある。GNOME環境で動作するRPMパッケージユーティリティだ。

GNOMEのメインメニューから[システム] - [GnoRPM]で起動できる(画面22)。GnoRPMのウィンドウ左側にある一覧は現在インストールされているパッケージ群だ。ファイルマネージャと同様、カテゴリー分けされたツリーをクリックしながらたどることが

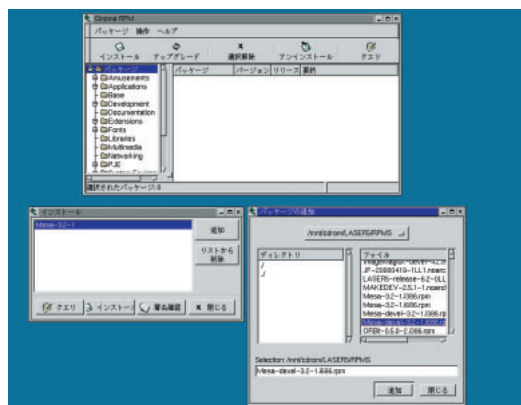
できる。非常に便利だ。

GnoRPMはここで紹介したrpmコマンドの機能はすべて網羅している。ここでは、GnoRPMを使った新規インストールを例に手順を紹介する。インストールしたいRPM形式のパッケージファイルをどこかに用意しておこう。インターネットからダウンロードしてきてもいいし、パッケージが入ったCD-ROMをマウントしてもいい。

用意できたら、GnoRPMの[インストール]ボタンをクリックする。[インストール]ダイアログボックスが表示されるので[追加]ボタンをクリックする。[パッケージの追加]ダイアログボックスが表示されるので、用意したパッケージファイルを選択して[追加]ボタンをクリックする(画面23)。[インストール]ダイアログボックスにパッケージが追加されたことを確認して



画面22 GnoRPMはファイルマネージャに似たインターフェイスで直感的にわかりやすい



画面23 インストールするパッケージファイルを選択する。ここではマウントしたCD-ROMから選択している



画面24 インストールの様子は棒グラフで逐次表示される

画面25 Mesaが登録された。もちろん、後でアンインストールすることも可能だ



ダイアログボックス上にある [インストール] ボタンをクリックする。

これでインストールが実行され、進行状況が表示される (画面24)。きちんとRPMのデータベースにも登録されていることがわかる (画面25)。

tarボールからインストール

RPM形式に対して、tar形式のアーカイブ (tarボールと呼ぶ) による配布も主流である。サイトによってはRPM形式とtar形式の両方のアーカイブを用意してくれている。RPM形式をサポートしていないディストリビューションでは、こちらを使うことになる。

tarボールの展開方法はすでに述べた。ここではtarボールからソースプログラムを展開し、コンパイルしてインストールする手順を紹介しておこう。

tarボールで配布されるアーカイブからは、通常、ソースプログラムやドキュメント一式を収めたディレクトリが展開される。そこからmakeコマンドを使ってインストールするのだ。

RPMに比べると若干難しく映るかもしれないが、さして難しいことはない。基本的な手順は決まっていると言ってもいい。展開して、作成されたディレクトリがMesaだとすれば、

```
$ cd Mesa
$ ./configure
$ make clean
$ make
$ su
# make install
```

という手順になる。

これらの配布形態では、ディレクトリ内にconfigureという実行形式のファイルがあり、これを実行することでシ

ステムの情報を踏まえたMakefileが作成される。Makefileはmakeコマンドの動作を定義したファイルだ。

make cleanで、念のため中間ファイル (コンパイルの過程で作成される各種ファイル) を一掃しておく。その後、makeでコンパイルを開始する。makeの実行時間はコンパイル対象により異なる。

コンパイルが終了したら、suコマンドでrootユーザーになり、make installでインストールする。

アンインストール

上記の方法でインストールしたソフトウェアのアンインストールの方法は以下の手順になる。

```
$ cd Mesa
$ su
# make uninstall
```

展開されたディレクトリに移動し、ルート権限になり、make uninstallでアンインストールする。

アンインストールする際にもMakefileが必要なことに注意してほしい。「インストールが終わったから捨てちゃった」なんてことをすると後悔することになる。tarボールの展開ディレクトリは、キチンと/usr/local/srcなどに保管するなりしておこう。

gzとbz2圧縮

インターネット上で流通するtarボールは、.tar.gzという拡張子以外に、.tar.bz2という拡張子を持つものもある。これは、gzipの代わりにbzip2という圧縮ツールで圧縮されているだけで、gzipが使えるならビビることはない。展開するためのオプションも、先述したgzipとまったく同じである。

インストーラによるインストール

ソフトウェアによっては、シェルスクリプトなどで作成されたインストーラが付属している場合もある。この場合は、そのシェルスクリプトを実行すればインストールすることができる。

Column

rm コマンドは慎重に!

rootでrmコマンドを実行する場合は注意したい。一般ユーザーがいくらファイルパーミッションに「rwxr--r--」と設定していても、

rootの前には何の効力もない。もし、にわか管理者でrmコマンドに不安があるならば、rmコマンドは常に「-i」オプションを付けて実行しよう。そうすれば削除する前に一度確認を求めてくれるので、ついウツカリの危険性を減らせるはずだ。rootユーザーは慎重に。

Column

設定ファイルの更新はバックアップを取ってから

システム管理作業に設定ファイルの変更はつきものだ。このとき、エディタで変更する前に必ず設定ファイルのバックアップを取っておこう。設定をちょっと間違えただけで、さまざまなソフトウェアが起動しなくなる危

険性があるからだ。

バックアップといっても何も大袈裟なものを言っているのではない。fstabを変更する前に、cpコマンドでfstab.oldとかいう名前にして複製しておくという話だ。決して、いきなりバリバリとエディタで書き換え始めてはいけないうい! 何度も変更するのであれば、fstab.000714.bakなどと日付を入れるのもいいだろう。

Linux プレインストール製品 13 機種

Linuxサーバ 導入ガイド

文：編集部

Text：Linux magazine

photo：Shuichi Mito (Dee)



国内の手元PCメーカーの多くは、Linuxの動作確認情報をWebページなどで公開しているが、プレインストールマシンは発売していない。しかし、従来はWindows NT/2000サーバ機として販売していたPCサーバに、今年になって大手メーカーが続々とLinuxをプレインストールして発売している。

ユーザー（＝管理者）自らが、Linuxをインストールする場合、ユーザー自身がLinuxの動作を確認する必要があり、サーバ機能のセットアップやトラブルに対処するには知識と経験が要求される。その点、プレインストールマシンだと、インストールの手間がかからないのですぐに使うことができ、もちろん動作保証されているのでハードウェアとの相性問題がないなどのメリットがあり、故障した際の修理に関しても、オンサイトまたはセンドバック保証が付いていて安心である。

また、サーバ機には、デュアルプロセッサ対応や、データ信頼性を確保するために、ホットスワップが可能なRAIDハードディスクを内蔵しているといった特徴がある。

そこで、仕事で使うのに信頼のおけるLinuxプレインストールマシンを、一挙に紹介しよう。

今回紹介するLinuxマシンの中には、ラックマウントタイプのサーバが多く含まれている。Webサーバやデータベースマシンは、複数台のマシンでシステムを構成するため、タワー型よりスペース効率がよく、管理がしやすいといったメリットがあるためだ。

ラックマウント用サーバは、主にISP (Internet Service Provider) や ASP (Application Service Provider) 向けだ。19インチラックに、ラックマウントタイプのサーバを設置すれば、ラックのサイズにもよるが、2U (8.8cm) サイズのサーバで19～21台、1U (4.4cm) サイズのサーバなら40台程度のマシンを1本のラックに内蔵できる。

ふだん、あまりお目にかからない1Uや2Uといった薄型サーバ機の特徴を見よう。

写真1は、ノーザンライツコンピュータの2Uサイズサーバ「NL Server 2100」の内部を上から見たところだ。左側が前面パネルになり、フロッピー、CD-ROMドライブ、ハードディスクが並べられている。ハードディスクは上下に2段並べて合計4台で、ホットスワップ可能なRAID構成が取れるようにしている。

ハードディスクのすぐ右にファンが2個並んでいて、マザーボードに向かって風を送っている。デスクトップやタワー型のケースでは、背面パネルにファンが付いているが、ラックマウントタイプでは、背面パネルには入出力用のコネクタや、拡張ボードが置かれるために、ファン用のスペースが取れないためだ。右側上部は電源である。

このマザーボードは、グラフィックコントローラやネットワーク、SCSIコントローラをオンボードに搭載している。そしてメモリやデュアルCPU対応のSlot 1ソケットなどを、効率よく配

置している。

2Uサイズといえども、PCIスロットなどのオプションカードをマザーボードに垂直に立てるスペースは取れないため、ライザカードと呼ばれるスロット部分拡張ボードを利用して、マザーボードと平行してオプションカードを接続するようになっている。写真ではマザーボード上にPCIコネクタが見えているが、実はそれは使われないのである。

1U超薄型ラックマウントの特徴

今回紹介する1Uサイズのサーバ機で

は、PowerApp.webとNL Server 1100、VC7850-1UIの3機種が、同じIntel製のケース、マザーボードを使っているようだ。前面のプラスチックカバー部分は独自のものであり、ラックにマウントした状態ではそうはわからないだろう。

1Uサイズラックマウントケースだと、パーツの実装にはさらに工夫を要する。ストレージ系はケースの厚みが気になるし、マザーボード上に載せるパーツでは、マザーボードを実装するための高さも計算に入れる必要があり、

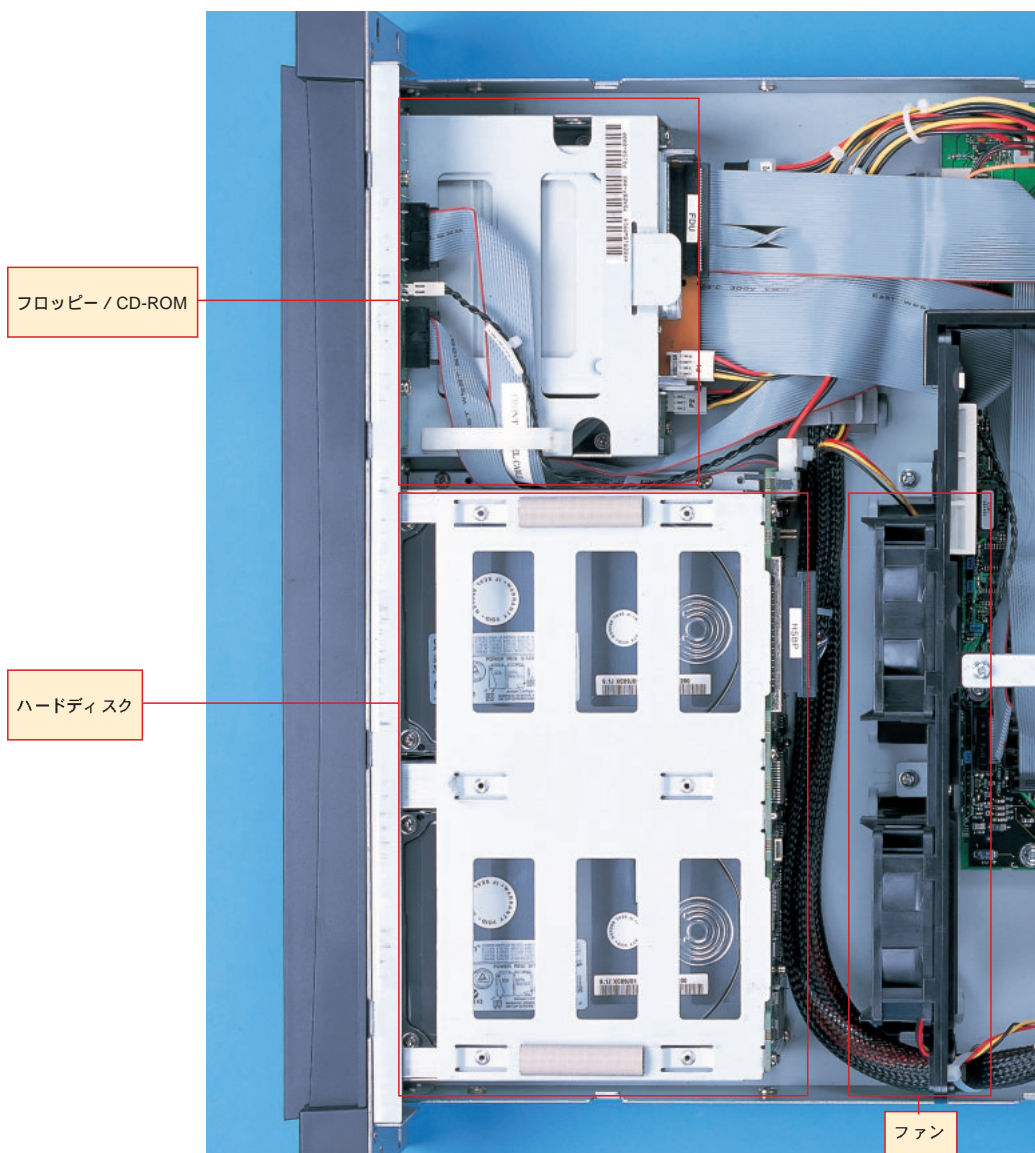


写真1 NL Server 2100



そこに付いているコネクタの高さなども影響する。

CPUは、Slot 1タイプは使えない。ソケットタイプでもCPUファンの高さは気をつけないといけない。装着できてもファンから風が流れるようなスペースが上部に必要だ。そのためCPUファンを使わずに、大きめのヒートシンクを採用している場合もある。

メモリは斜めに傾けて装着できるようなDIMMソケットを使っている。大容量のDIMMモジュールを利用する場合、基板の高さが問題になることがある。

CD-ROMやDVD-ROMドライブなど5インチドライブベイに実装するタイプは、高さ約4.2cmとギリギリである。フロッピーディスクドライブはハードディスクと同じ1インチサイズなので大丈夫だ。CD-ROM、フロッピー共にスリムタイプを使用すれば重ねて装着できる。PCI、ISA、AGPカードなども縦に実装するのは無理なので、ライザカードを使用して、マザーボードと平行に実装することになる。それでも1枚分しか高さは取れず、2スロット確保するには平面に広げて装着し、なおかつ1スロットは高

さが低いカードしか使えないようになっている。優先度の低いISA、AGP用のコネクタは省略されている場合も多い。

ハードディスクは熱に弱い

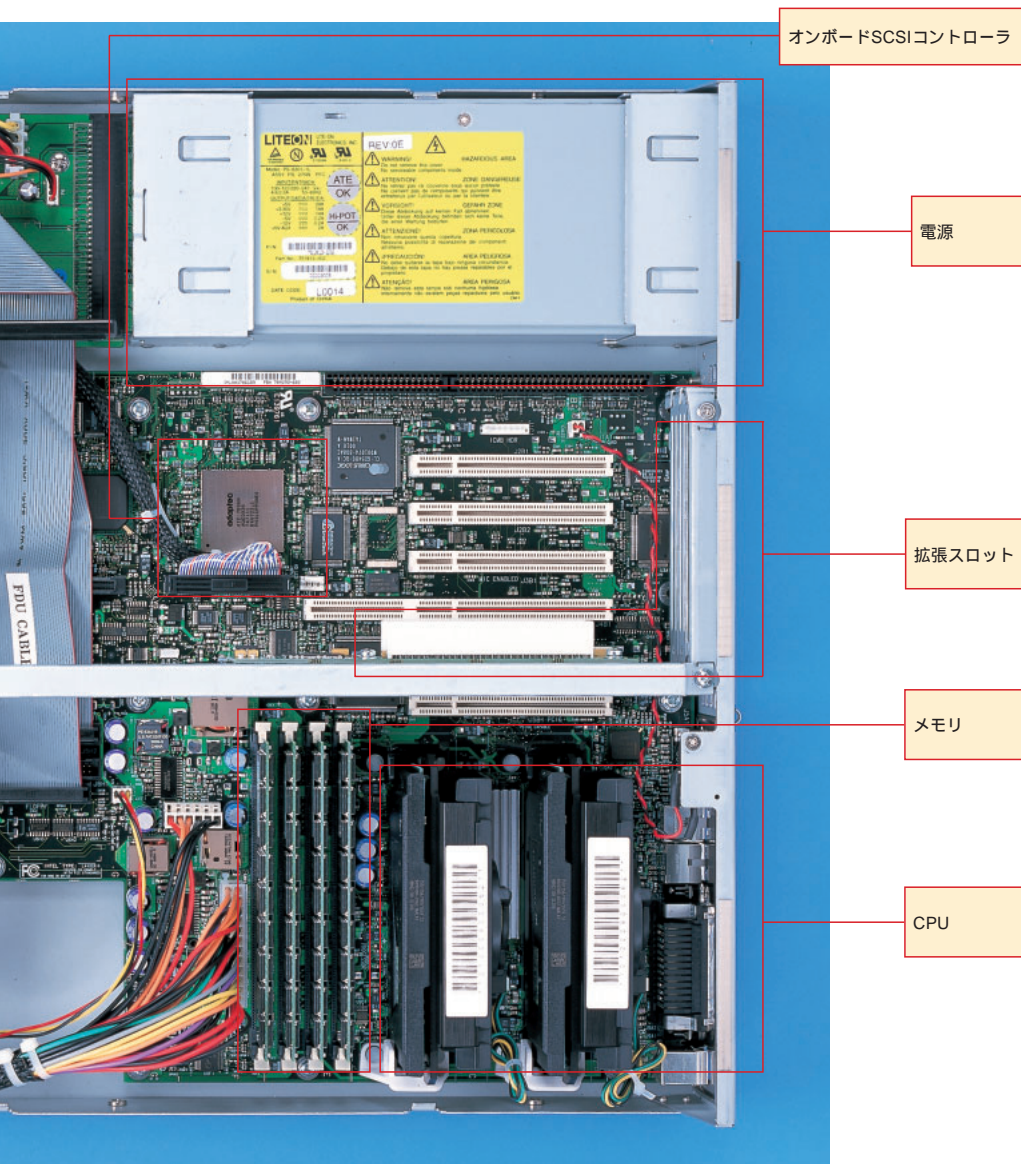
実際にラックにマウントすると、びっしりと重なるため、ケースパネルからの放熱は期待できない。というより相互に暖め合ってしまうような状況になる。そのためにもファンで強制的に排気することが必要で、なおかつ前面パネルから吸い込んで、背面へいかに排気するかというケース内部の空気の流れ方も重要だ。

特に、スピードを追求した7200rpmや10000rpmといったSCSIハードディスクは、その高速回転のために大きく発熱する。しかし、ドライブメーカーのスペックを見るとわかるが、ほとんどのドライブの動作温度は5度から、50度または55度となっていて、意外と熱に弱いのである。高温での動作を続けると、熱膨張による歪みでディスクの軸受けの摩耗が進みやすく、コントロールボード上のパーツが故障したりする可能性も高くなる。

メーカー保証

ほとんどの場合、1年間のセンドバックまたはオンサイト保証が付いている。標準ではオンサイト保証といっても営業時間内の受付で対応は翌営業日ということもあるが、24時間体制で受け付けたり、即日修理といった緊急度の高いサポートが受けられるオプション契約を用意しているメーカーもある。

また、メーカーやサポートを行う関連会社によって、Webサーバやメールサーバなどのセットアップサービスを有償で行ってくれるところもあって、Linuxサーバの導入はますますしやすくなってきている。



RAID対応の2Uラックマウント高性能サーバ

PowerEdge 2450

デルコンピュータ株式会社

044-556-6190

http://www.dell.com/jp/

33万3000円～

デルコンピュータは、サーバ製品「PowerEdge (パワーエッジ)」の全製品ライン (PowerEdge 1300、2400、4400、6400、2450、6450) で、Red Hat Linux 6.2Jを標準搭載したモデルの販売を2000年5月より開始している。

また、エントリーモデルのミドルタワー PowerEdge 1300には、Turbo Linuxモデルもあり、CPUはPentium 700MHz、メモリ64Mバイト、9GバイトUltra2Wide SCSIハードディスクという構成で、17万800円である。

PowerEdge 2450

PowerEdge 2450は2U (高さ8.8cm) サイズのケースを採用していて、データセンターなどの大規模サーバシステムの構築に向いている。

CPUは、FSB 133MHzのPentium (600 / 667 / 733 / 866MHz) で、デュアルプロセッサ構成が可能。RCC社のLE64チップセットを採用し、メモリはECC付きで最大2Gバイトまで拡張できる。最大転送速度160Mバイト/秒のUltra3 (Ultra160) SCSIコントロ

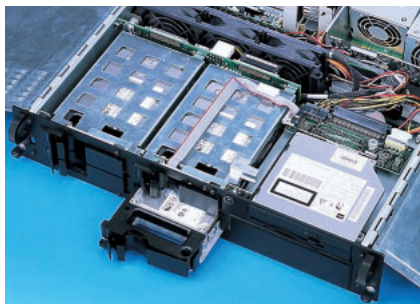


写真2-1 ホットスワップ対応のドライブベイ
9Gまたは18GバイトのSCSIハードディスクを4台まで内蔵可能。CD-ROM下部にもペリフェラルベイがある。

ーラをオンボードに搭載し、10000rpmと高速な9Gまたは18GバイトのSCSIハードディスクを4台まで搭載できる。オプションのRAIDコントローラを利用することで、前面パネルから交換できるホットスワップに対応する。

また、内蔵用DDS4テープドライブ (20G / 40Gバイト) や、外付けDLTテープドライブなどのバックアップ装置、

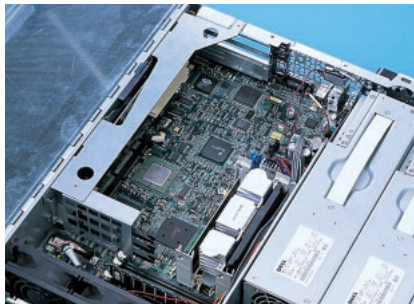


写真2-2 デュアルCPU対応のマザーボード
チップセットは、RCC LE64を採用。オンボードにほぼすべての機能を搭載している。



外付けRAIDシステム、ギガビットネットワークアダプタといった多くのオプションが用意されている。

ATI Rage IIcグラフィックスコントローラ、Intel 82559ネットワークコントローラをオンボードに搭載しているため、2Uサイズの限られた空間に設けられたPCIスロット×3ポートをすべて拡張のために使えるのは便利だろう。

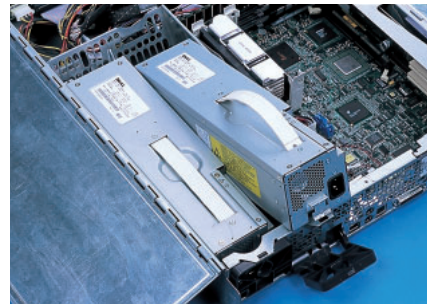


写真2-3 専用ファン内蔵の330W電源
システムの信頼性を高めるために、オプションで電源のホットスワップ対応 (二重化構成) が可能。

製品名	PowerEdge 2450
CPU	Pentium 600 / 667 / 733 / 866MHz (シングル/デュアルCPU)
メモリ	64M ~ 2Gバイト ECC SDRAM (PCI133、DIMMスロット×4)
チップセット	RCC LE64 3.0
ハードディスク	9G / 18GバイトUltra3 (U160)SCSI 10000rpm (最大4台)
SCSIインターフェイス	オンボードAdaptec AIC-7899 (Ultra160/m)
RAIDコントローラ	オプション (PowerEdge RAIDコントローラ)
グラフィックス	ATI Rage IIc (4Mバイト)
ネットワーク	10/100BASE-T (Intel 82559) × 1
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	24倍速 ATAPI CD-ROM
拡張スロット	PCIスロット×3
I/Oポート	RS-232C × 2、パラレル×1、USB × 2、PS/2 × 2
キーボード	日本語106キーボード
マウス	2ボタンマウス
ケース	2Uラックマウント
電源	330W (オプションで二重化構成ホットスワップ対応)
外形寸法 (mm)	424 (W) × 84 (H) × 668 (D)
重量	25kg (最大)
OS	Red Hat Linux 6.2J
価格	33万3000円～



1UサイズのラックマウントWebサーバ

PowerApp.web 100 Linuxモデル

デルコンピュータ株式会社

044-556-6190

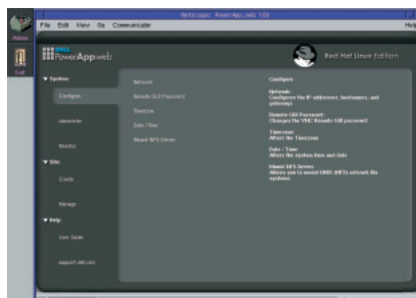
http://www.dell.com/jp/

20万8000円～

デルコンピュータは、ISP / ASP向けの高性能アプライアンスサーバ製品として「PowerApp (パワーアップ)」シリーズを2000年6月より発売している。Webサーバ用のPowerApp.web、Webキャッシング機能によりネットワークトラフィックを削減し、Webサーバの負荷を分散するPowerApp.cacheの2シリーズが用意されていて、1U (高さ4.4cm) サイズの「PowerApp.web 100」と「PowerApp.cache 100」、2U (高さ8.8cm) サイズの「PowerApp.cache 200」がある。いずれもラックマウント専用の薄型のケースを採用している。

PowerApp.web 100 Linuxモデル

OSにカスタマイズされたRed Hat Linux 6.2J + Apacheを搭載した「PowerApp.web 100 Linuxモデル」はPentium 600MHz、64MバイトECC SDRAM、9GバイトSCSIハードディスク、10/100BASE-Tネットワークカード×2、スリムCD / フロッピーの構成で20万8000円となっている。



画面3-1 設定専用ツールstartadmintool
Webブラウザからリモートでサーバの管理が行える、PowerApp.web専用の設定ツールが用意されている。

PowerAppシリーズは、複数のマシンをラックに設置して使用するために便利なツールが用意されている。独自のKick Startユーティリティは、導入時のセットアップが簡単に短時間でできる。あらかじめ設定を記録したフロッピーを挿入して起動すると、最短の場合1台あたり約5分でセットアップできる。

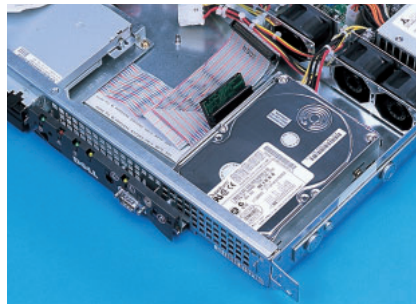


写真3-1 Ultra3 SCSI対応ハードディスク
9Gまたは18GバイトのUltra160対応ハードディスクを2台まで内蔵可能。



また、キーボードとマウスは付属しないため、Webベースの管理ツール (画面3-1) を使用して、ネットワーク環境で各種設定を行うことができる。

サポートとして、3年間保証、電話での無償インストールサポート (30日間) ハードウェアのトラブルシューティングと診断を24時間電話で行うテクニカルサポートが用意されている。

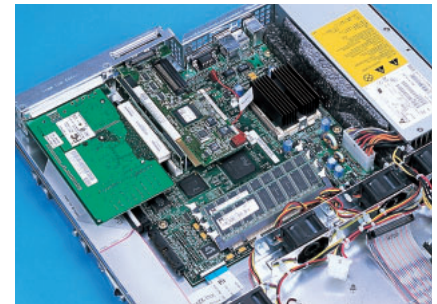


写真3-2 FC-PGA CPU対応のマザーボード
2つのPCIスロットには、SCSIコントローラとグラフィックスカードが接続されている。

製品名	PowerApp.web 100
CPU	Pentium 600 / 650 / 700MHz
メモリ	64M ~ 512Mバイト ECC SDRAM (PC100、DIMMスロット×4)
チップセット	Intel 440BX
ハードディスク	9G / 18Gバイト Ultra3 (U160) SCSI 7200rpm (最大2台)
SCSIインターフェイス	Ultra3 SCSIコントローラ
RAIDコントローラ	
グラフィックス	Matrox Millennium G200 SD (8Mバイト)
ネットワーク	10/100BASE-T (Intel 82559) × 2
フロッピーディスク	3.5インチ 1.44Mバイト (スリムタイプ)
CD-ROM	40倍速ATAPI CD-ROM (スリムタイプ)
拡張スロット	PCI × 2 (空き0)
I/Oポート	PS/2 × 2
キーボード	オプション
マウス	オプション
ケース	1Uラックマウント
電源	125W
外形寸法 (mm)	425 (W) × 43 (H) × 550 (D)
重量	10kg (最大)
OS	Red Hat Linux 6.2 + Apache
価格	20万8000円～

1U / 2UサイズのラックマウントLinuxサーバ

NL Server 1100/NL Server 2100

ノーザンライツコンピュータ株式会社

0120-508-138

http://www.nlcomputer.com/

1100 (58万6000円～) / 2100 (69万円～)

ノーザンライツコンピュータは、ISP / ASP向けラックマウントサーバとして、1Uサイズの「NL Server 1100」を2000年7月から、2Uサイズの「NL Server 2100」を同5月から発売している。OSには、Red Hat Linux 6.2Jをプレインストールし、さらに、セキュリティーホールなどに対応するために、常にネットで公開された最新のパッケージをインストールしている。

NL Server 1100

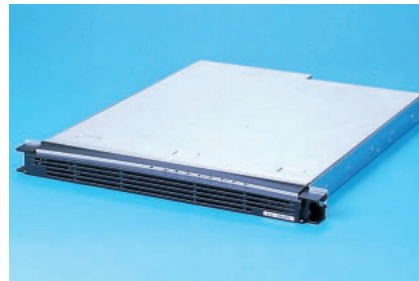
NL Server 1100の標準構成は、CPUにPentium 750MHz、メモリは128MバイトECC SDRAM、ハードディスクは10.2GバイトのIDEハードディスク、オンボードに10/100BASE-T対応ネットワークコントローラを2ポート、グラフィックスはATI RAGE

PRO TurboをPCIスロットに装着している。

NL Server 2100

NL Server 2100は、CPUにPentium 800MHzを最大2個まで搭載可能で、メモリは標準で128MバイトECC SDRAM、ハードディスクは9GバイトのUltra2Wide SCSIをオンボードのUltra2 SCSIコントローラで接続する。4台までのSCSIハードディスクを内蔵できるホットスワップ可能なドライブベイを装備していて、ソフトウェアRAIDに対応、オプションでハードウェアRAIDにも対応している。

グラフィックスにはオンボードGD5480、ネットワークは10/100BASE-Tに対応したIntel Pro/100+を搭載している。



NL Server 1100



NL Server 2100

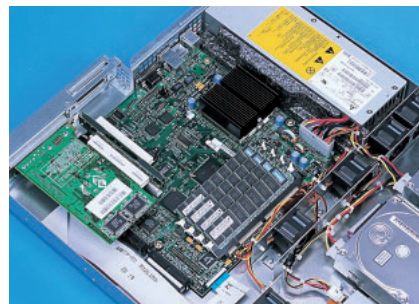


写真4-1 NL Server 1100のマザーボード
CPUはFC-PGAソケットPentium 750MHz、メモリモジュールは25度に傾けて装着するようになっている。

製品名	NL Server 1100	NL Server 2100
CPU	Pentium 750MHz～	Pentium 650MHz～800MHz (シングル/デュアルCPU)
メモリ	128M～1Gバイト ECC SDRAM (PC100、DIMMソケット×4)	128M～2Gバイト ECC SDRAM (PC100、DIMMスロット×4)
チップセット	Intel 440BX	Intel 82443GX
ハードディスク	10.2G～30GバイトIDEハードディスク (最大2台)	9G～36Gバイト Ultra2 Wide/LVD SCSI (最大4台)
SCSIインターフェイス		オンボードAdaptec AIC7896 Ultra2/LVD SCSI
RAIDコントローラ		
グラフィックス	ATI RAGE PRO Turbo PCI (8MバイトSGRAM)	オンボードGD5480 (2MバイトSGRAM)
ネットワーク	オンボード10/100BASE-T (Intel 82559) ×2ポート	オンボード10/100BASE-T (Intel 82559)
フロッピーディスク	3.5インチ 1.44Mバイト	3.5インチ 1.44Mバイト
CD-ROM	オプション (スリムタイプ40倍速ATAPI)	オプション (スリムタイプ40倍速ATAPI)
拡張スロット	PCIスロット×2 (1つはハーフレングス)	PCIスロット×2
I/Oポート	RS-232C ×1、USB ×2、PS/2 ×2	RS-232C ×2、パラレル ×1、PS/2 ×2
キーボード	オプション	オプション
マウス	オプション	オプション
ケース	1Uラックマウント	2Uラックマウント
電源	125W	275W
外形寸法 (mm)	425.45 (W) × 43.18 (H) × 558.80 (D)	480 (W) × 88 (H) × 612 (D)
重量	10.5kg (最大構成の場合)	22.7kg (最小構成の場合)
OS	Red Hat Linux 6.2J	Red Hat Linux 6.2J
価格	58万6000円～	69万円～



RAID搭載の高信頼性Linuxサーバ

NL Server 3000

ノーザンライツコンピュータ株式会社

0120-508-138

http://www.nlcomputer.com/

85万8000円～

NL Server 3000

NL Server 3000は、CPUにPentium (600～750MHz) を搭載し、シングルプロセッサとデュアルプロセッサの両仕様が用意されている。メモリはECC (エラー検出訂正機能) 付きのSDRAMで、256Mバイトから最大1Gバイトまで拡張できる。ネットワークカードは10BASE/100BASE-Tに対応したDECの21140AFを使用している。

ハードディスクはRAID構成になっていて、標準で9.1Gバイト (7200rpm) Ultra2 Wide SCSIのドライブを3台使ったRAID5である。RAID5ではデータとパリティを複数のドライブに分散して記録する。同時に書き込みを行うためスピードが速く、冗長性を持たせてあるため、ドライブが1台故障しても、そのほかのドライブのデータを使用して正しいデータを読み出すことが可能である。RAID専用のコントローラは、MylexのAcceleRAID250でLinuxにも対応している。ドライブベイは5台分あり、18.2Gバイトのハードディスクを5台接続すれば70Gバイト (RAID5コンフィグレーション時の実容量) の容量を内蔵できる。消費電力の大きいSCSIハードディスクに対応できるように、400Wと大容量の電源が付いている。

タワー型のケースは、高さは46cmでふつうのミドルタワーと変わらないが、ドライブベイが前面全体にあるため、奥行きが62cmと長くなっている。5Uサイズのラックマウント仕様とするこ

ともできる。高速ハードディスクの冷却のためドライブベイ部分専用のファンが2個、ケース背面にもファンが2個付いていて冷却性能はかなり高い。

サーバ用途には欠かせないバックアップテープ装置もオプションで提供されており、本体に内蔵することができる。OSには、Red Hat Linux 6.2Jがプレインストールされている。

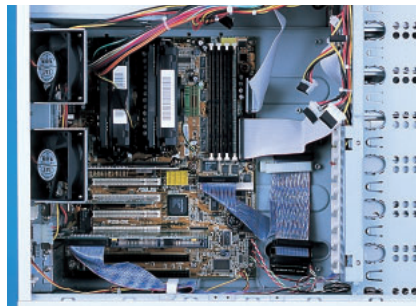


写真5-1 NL Server 3000の内部
デュアルPentium やRAIDハードディスクの発熱に対応するため強力なファンが付いている。



1年間の無償オンサイト保証と、3か月間の電話またはメールによるテクニカルサポートが受けられる。そして、故障の際には北海道から沖縄まで全国60カ所あるサービス拠点から、担当技術者が派遣される。



写真5-2 NL Server 3000のドライブベイ
標準でRAID5をサポートし、SCSIハードディスク5台まで装着できる。故障時にはホットスワップが可能。

製品名	NL Server 3000
CPU	Pentium III 600～750MHz (シングルまたはデュアル)
メモリ	256M～1Gバイト ECC SDRAM (PC100、DIMMスロット×4)
チップセット	Intel 440BX
ハードディスク	9Gバイト (Ultra2 Wide/LVD) × 3 RAID5 (最大5台)
SCSIインターフェイス	Adaptec 7980+3860 Ultra2/LVDオンボード
RAIDコントローラ	Mylex AcceleRAID250 Ultra2 Wide
グラフィックス	ATI XPERT98 (8MバイトSDRAM)
ネットワーク	10/100BASE-T (DEC 21140AF)
フロッピーディスク	3.5インチ (1.44Mバイト)
CD-ROM	40倍速 ATAPI CD-ROM
拡張スロット	PCIスロット×4 (空き2) ISAスロット×2
I/Oポート	RS-232C×2、パラレル×1、USB×2、PS/2×2
キーボード	オプション
マウス	オプション
ケース	ASUS AS-30 Tower (5Uラックマウントにコンバーチブル)
電源	400W ATX
外形寸法 (mm)	220 (W) × 460 (H) × 620 (D)
重量	25kg
OS	Red Hat Linux 6.2J
価格	85万8000円～

1UサイズのラックマウントLinuxサーバ

Trus-1

ぶらっとホーム株式会社

03-3251-2600

http://www.plathome.co.jp/

31万8000円～

ぶらっとホームから、1Uおよび2Uサイズのラックマウントサーバ「Trus」タイプが発売されている。

1U（高さ4.4cm）サイズには、SCSIハードディスクを搭載した「Trus-1」と、IDEハードディスクを搭載した「Trus-1i」の2種類あり、それ以外の基本構成はCPUにCeleron 466MHz、メモリ64Mバイトとなっている。

2U（高さ8.8cm）サイズには、CPUにPentium 600MHzを採用した「Trus-2」と、CPUにPentium 600MHzのデュアルプロセッサを採用した「Trus-2D」の2種類あり、それ以外の基本構成は、メモリ128Mバイト、9.1GバイトUltra2Wide SCSIインターフェイスのハードディスクとなっている。

Trus-1

Trus-1は、CPUにCeleron 466MHz。メモリは標準で64Mバイトだが、3つあるDIMMスロットに256MバイトのSDRAMを装着することで、最大768Mバイトまで拡張が可能になっている。ハードディスクは9.1GバイトのUltraWide SCSIが採用されている。

小型のマザーボードには、SCSIコントローラ、グラフィックス、ネットワークコントローラがオンボードで搭載されている。SCSIコントローラはAdaptecのAIC-7890が、グラフィックスはC & T社の69000が、ネットワークコントローラは10/100BASE-T対応のIntelの82558Bが使われている。

拡張スロットは、PCIとISA共通の

スロットが1つだけある。そのほかのI/Oポートとして、シリアル1ポート、パラレル1ポート、キーボードとマウス用のPS/2が2ポートあり、背面にコネクタが接続されている。

ケース内部には、大きな空きスペースが見られるが（写真6-1）、5インチベイ、拡張スロット、3.5インチ内蔵ベ

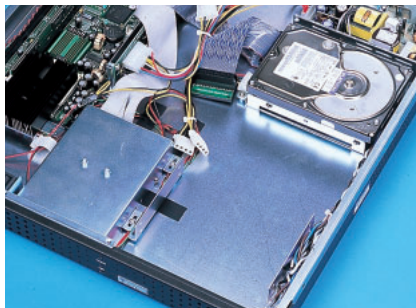


写真6-1 Trus-1の内部
5インチベイ相当の空間が空いている。実際に使用する場合には、ぶらっとホームの営業と相談が必要。



イのどれかとして利用できる（同時には使用できない）。

OSは、表のように多くのディストリビューションに対応している。

1年間のセンドバック保証がついていて、2年目以降は、センドバック保守サービスを申し込むか、スポットでの有償修理を依頼することになる。

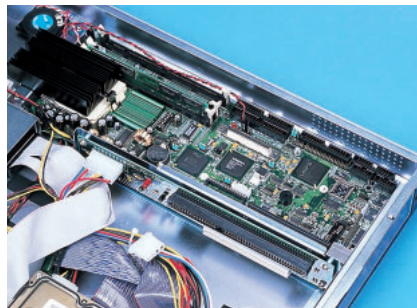


写真6-2 コンパクトなマザーボード
すべての機能が1枚のマザーボード上に納められているといえるだろう。PCI/ISAスロットも手前に見える。

製品名	Trus-1
CPU	Intel Celeron 466MHz
メモリ	64M～768Mバイト SDRAM (PC100、DIMMスロット×3)
チップセット	Intel 440BX
ハードディスク	9.1Gバイト7200rpm UltraWide SCSIハードディスク
SCSIインターフェイス	オンボードAdaptec AIC-7890
RAIDコントローラ	
グラフィックス	オンボードC & T 69000 (2Mバイト)
ネットワーク	オンボード10/100BASE-T (Intel 82558B)
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	なし
拡張スロット	PCI (ISA共用) スロット×1
I/Oポート	RS-232C×1、パラレル×1、PS/2×2
キーボード	オプション
マウス	オプション
ケース	1Uラックマウント
電源	80W
外形寸法 (mm)	439.6 (W) × 44 (H) × 482.6 (D)
重量	約5.5kg
OS (オプション)	Red Hat Linux 6.2J、Vine Linux 2.0CR、LASER5 Linux 6.0 Rel.2、TurboLinux日本語版4.0、Slackware 7.0を選択可能
価格	31万8000円～



5Uのラックマウントスペースに4ユニット置けるコンパクトサーバ

VTrus-1i

ぷらっとホーム株式会社

03-3251-2600

http://www.plathome.co.jp/

24万8000円～

ぷらっとホームから、高さ5U (22cm) サイズのコンパクトPC「VTrus」タイプが発売されている。

横幅が1/4サイズの5Uクォーターサイズには、CPUにPentium 550MHz、9.1GバイトSCSIハードディスクを搭載した「VTrus-1」と、CPUにCeleron 466MHz、20GバイトIDEハードディスクを搭載した「VTrus-1i」の2種類ある。

横幅が1/2サイズの5Uハーフサイズには、CPUにPentium 550MHz、9.1GバイトSCSIハードディスクを搭載した「VTrus-2」と、CPUにCeleron 466MHz、20GバイトIDEハードディスクを搭載した「VTrus-2i」、CPUにデュアルPentium 550MHz、20GバイトIDEハードディスクを搭載した「VTrus-2iD」、CPUにデュアルPentium 550MHz、9.1GバイトSCSIハードディスクを搭載した「VTrus-2D」の4種類ある。

VTrusの基本構成は、全機種メモリ128Mバイトとなっている。

VTrus-1i

小型のマザーボードには、グラフィックス、ネットワークコントローラがオンボードで搭載されていて、SCSIコントローラは装着されていない。

拡張スロットは、PCIスロットが3つあり、ISAスロットはない。そのほかのI/Oポートとして、シリアル1ポート、キーボードとマウス用のPS/2が2ポートあり、背面にコネクタが接続されている。

ケースは専用のラックマウントキャリア「VTrusキャリア (別売1万2000円)」を利用することが前提のため、ラックマウント時の取り扱いが便利になるように、取手や固定用の手回し可能なネジが前面パネルに付いている。Trus-1のような1Uサイズのものより、空間に余裕があるのと、前面に付いて

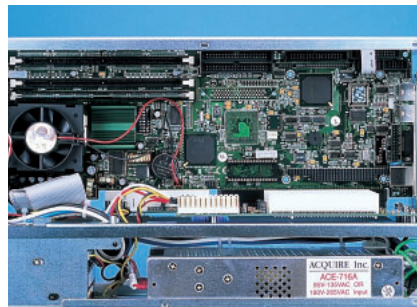


写真7-1 VTrus-1iの内部

マザーボードは、Trus-1と同製品で一部仕様異なる。電源の左(前面寄り)にハードディスクが収納されている。



いる大きめのファンによって放熱面が有利になっているため発熱量の多い高速CPUの搭載が可能になっている。

OSとサポートは、Trus-1と同様なので、前ページを参照していただきたい。



写真7-2 VTrus-1i前面部分

ラック収納時の固定のため、前面パネルは下部のネジ止め部分が長くなっている。上部にはファンを内蔵。

製品名	VTrus-1i
CPU	Intel Celeron 466MHz
メモリ	128M ~ 768Mバイト SDRAM (PC100、DIMMスロット×3)
チップセット	Intel 440BX
ハードディスク	20GバイトIDEハードディスク
SCSIインターフェイス	
RAIDコントローラ	
グラフィックス	オンボードC & T 69000 (2Mバイト)
ネットワーク	オンボード10/100BASE-T (Intel 82558B)
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	なし
拡張スロット	PCIスロット×3
I/Oポート	RS-232C×1、PS/2×2
キーボード	オプション
マウス	オプション
ケース	5U縦型1/4ラックマウント
電源	90W
外形寸法 (mm)	110 (W) × 211.5 (H) × 440.4 (D)
重量	約5.5kg
OS (オプション)	Red Hat Linux 6.2J、Vine Linux 2.0CR、LASER5 Linux 6.0 Rel.2、TurboLinux日本語版4.0、Slackware 7.0を選択可能
価格	24万8000円～

1Uサイズの高性能ラックマウントサーバ

Vintage Rackmount Server VC7850-1UI

株式会社日本コンピューティングシステム

03-3821-3200

http://www.jcsn.co.jp/

34万8000円～

日本コンピューティングシステムは、1Uサイズの超薄型のラックマウントサーバ「Vintage Rackmount 1U」シリーズを発売している。Vintage Rackmount 1Uシリーズは、CPUにPentium またはCeleronプロセッサを搭載したシステムで、BTOによって、CPU、メモリやハードディスク、CD-ROMドライブ、グラフィックスカードなどのパーツを注文時にカスタマイズ可能になっている。

また、プレインストールされるOSも、Red Hat Linux 6.2J、LASER5 Linux 6.2、TurboLinux Workstation日本語版6.0、TurboLinux Server日本語版6.1から選択することができる。

VC7850-1UI

VC7850-1UIは、Intel純正のマザーボードと1Uラックマウントケースを使用している。

標準構成では、CPUにPentium 850MHz、メモリは128MバイトECC SDRAM、ハードディスクは30.7GバイトのIDEハードディスク、オンボードに10/100BASE-T対応のIntel 82559を2ポート搭載している。

1U（高さ4.4cm）サイズなので、フロッピーディスクドライブとCD-ROMドライブは、スリムタイプを使用して上下にまとめて、前面パネル左側に装着されている。

PCIスロットは2ポートあるが、ひとつはハーフレングスタイプで、CanopusのSPECTRA Light T32PCI

というグラフィックスカードが装着されている。これは、アクセラレータにnVIDIA RIVA TNT2 Proを使用し、ビデオメモリには32MバイトのSDRAMを搭載している。

また、シリアルインターフェイスは、背面に1ポートと前面に1ポート、キーボード/マウス接続用のPS/2とUSBインターフェイスが2ポート用意されている。

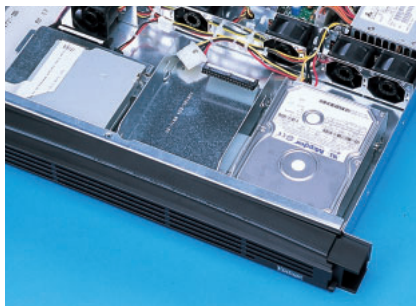


写真8-1 VC7850-1UIのドライブベイ
左側のフロッピードライブの下にスリムタイプCD-ROMが装着されている。ハードディスクは2台まで増設可能。



メモリは25度に寝かせたDIMMスロットが4個用意されている。ハードディスクを増設して2台にすることも可能なので、拡張性は十分といえるだろう。

OSには、TurboLinux Workstation日本語版6.0、またはRed Hat Linux 6.2Jがプレインストールされる。

サポートは、1年間無償オンサイトサービス保証が含まれている。

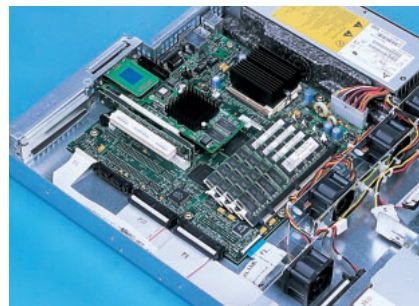


写真8-2 VC7850-1UIのマザーボード
CPUはPentium 850MHzを搭載。メモリは128M x 4枚で512Mバイトに増設している。

製品名	Vintage Rackmount 1U Server VC7850-1UI
CPU	Pentium 850MHz
メモリ	128M ~ 1Gバイト ECC SDRAM (PC100、DIMMスロット×4)
チップセット	Intel 440BX
ハードディスク	30.7Gバイト IDE 7200rpm (最大2台)
SCSIインターフェイス	
RAIDコントローラ	
グラフィックス	RIVA TNT2 Pro PCI (32MバイトSDRAM)
ネットワーク	オンボード10/100BASE-T (Intel 82559) × 2ポート
フロッピーディスク	3.5インチ 1.44Mバイト (スリムタイプ)
CD-ROM	24倍速ATAPI CD-ROM (スリムタイプ)
拡張スロット	PCIスロット×2 (1つはハーフレングス)
I/Oポート	RS-232C × 2、USB × 2、PS/2 × 2
キーボード	日本語109キーボード
マウス	3ボタンマウス
ケース	1Uラックマウント
電源	125W
外形寸法 (mm)	433 (W) × 44 (H) × 565 (D)
重量	約7kg
OS (オプション)	Red Hat Linux 6.2Jデラックス
価格	34万8000円～



Pentium 1GHzを搭載した高性能Linuxサーバ

Vintage Pro Linux VE71000UE

株式会社日本コンピューティングシステム

03-3821-3200

http://www.jcsn.co.jp/

32万4000円～

日本コンピューティングシステムは、LinuxをプレインストールしたミドルタワーPCサーバ「Vintage Pro Linuxシリーズ」を発売している。BTO（受注生産）販売なので、CPU、メモリ、ハードディスク、CD-ROMドライブ、ネットワークカード、ケースなど、すべてのパーツを注文時にカスタマイズ可能になっている。

また、プレインストールされるOSも、Red Hat Linux 6.2J、LASER5 Linux 6.2、Vine Linux 2.0CR、TurboLinux Workstation日本語版6.0、TurboLinux Server日本語版6.1、Kondara MNU/Linux 1.1、Kondara MNU/Linux Server 1.0から選択することができる。

VC71000UE

VC71000UEは、Pentium 1GHzを搭載したVintage Pro Linuxシリーズの最上位モデルである。

Pentium 1GHzの性能を引き出すため、マザーボードにIntel i820チップセットを使用したAOpen AX6C-Lを採用した。i820ではよく使われているSDRAMではなくDirect RDRAMをサポートし、本機では128MバイトのRDRAM（PC800）を搭載している。そのため未使用のメモリスロットには、Continuty RIMMモジュールというバス結線用のダミーカードを装着している（写真9-2）。

標準構成では、30.7GバイトIDEハードディスクを搭載し、グラフィックスカードはAGP 4xに対応したnVIDIA

RIVA TNT2 AGP（32Mバイト）、ネットワークカードは10/100BASE-T対応のIntel Pro/100+を使用している。ハードディスクは、9G～36GバイトのUltra160対応SCSIインターフェイスタイプを選ぶこともでき、その場合にはSCSIコントロールカードも併せて追加する必要がある。

OSには、TurboLinux Workstation

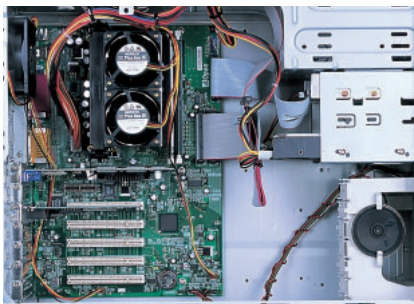


写真9-1 VC71000UEの内部
マザーボードはAOpenのAX6C-Lが使われている。チップセットはIntel i820。



日本語版6.0、またはRed Hat Linux 6.2Jがプレインストールされる。

サポートは、1年間無償センドバックサービス保証が含まれていて、オプション（1万円）で1年間無償オンサイトサービスに切り替えることができる。

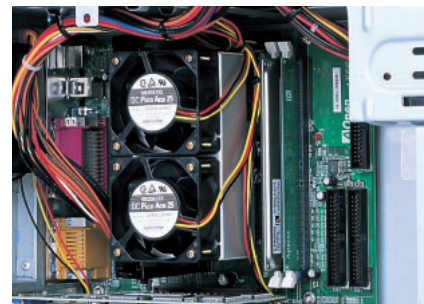


写真9-2 VC71000UEのCPUとメモリ
Slot 1タイプのPentium 1GHzは2つのファンで放熱している。メモリはPC800 RDRAMが使われている。

製品名	Vintage Pro Linux VC71000UE
CPU	Pentium 1GHz
メモリ	128M～512MバイトRDRAM（PC800、RIMMスロット×2）
チップセット	Intel i820（AOpen AX6C-L）
ハードディスク	30.7Gバイト IDE 7200rpm
SCSIインターフェイス	オプション
RAIDコントローラ	
グラフィックス	RIVA TNT2 Pro 32Mバイト AGP 4X（ELSA ERAZOR III Pro）
ネットワーク	10/100BASE-T（Intel Pro/100+）
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	40倍速ATAPI CD-ROM（TEAC CD-540E）
拡張スロット	PCIスロット×5（1つはネットワークカードで使用）
I/Oポート	RS-232C×2、パラレル×1、USB×2
キーボード	日本語109キーボード
マウス	Logitech 3ボタンマウス PS/2
ケース	ミドルタワー
電源	250W
外形寸法（mm）	186（W）×425（H）×465（D）
重量	約16kg
OS	TurboLinux Workstation日本語版6.0またはRed Hat Linux 6.2Jデラックス
価格	32万4000円～

2UサイズラックマウントLinuxサーバ

SGI 1200

日本SGI株式会社

0120-161-086

http://www.sgi.co.jp/

69万円～

日本SGIは、ラックマウント可能な薄型サーバである「SGI 1200」の発売を2000年4月より開始している。SGI 1200の上位機種として、4U（高さ17.8cm）の「SGI 1450」が2000年7月より発売された。SGI 1450は、Pentium Xeon 700MHzを最大4プロセッサ。メモリ最大4Gバイト（Linuxの場合）。36.4GバイトのUltra160 SCSI対応ハードディスクを5台まで収納可能。8スロットのPCI（64ビット対応）、電源の冗長化（3台まで）などの機能を持ったハイエンドサーバである。

SGI 1200

SGI 1200は、SGI 1000サーバファミリーのエントリーモデルで、高さ2Uの筐体に Pentium 550MHzまたは700MHzを2基まで搭載可能。メモリは2Gバイト、ハードディスクは72Gバイトまで拡張できる。マザーボード上に、AdaptecのUltra2 SCSIコントローラ、10/100BASE-T対応Intel 82559ネットワークコントローラ、Cirrus LogicのGD5480グラフィックスコントローラを搭載している。2つのPCIスロットが用意されていて、オプションのRAIDコントローラを装着することで、ホットスワップ可能なハードディスクを4台まで本体に内蔵可能となっている。

Pentium 550MHz、メモリ128Mバイト、ハードディスク9.1Gバイトの最小構成で69万円からとなっている。

従来は、SGI Linuxという独自にカスタマイズしたディストリビューショ

ンを提供していたが、SGI 1200では、Red Hat Linux 6.1をベースとして、機能を拡張するSGI ProPack 1.2というアドインパッケージが提供されている。カーネル2.2.13に対応し、3.8Gバイトまでの物理メモリ対応、I/Oパフォーマンスの向上などを実現する。SGI ProPack 1.2は、SGIの機能拡張部分をRPMパッケージで提供するもので、Red Hat



写真10-1 ホットスワップ可能なドライブベイ
本体カバーはフロント部分を除いて開くようになっている。4台分のUltra160 SCSI対応ドライブベイを持つ。



Linuxのほかに、TurboLinux、SuSEや日本語ディストリビューションにも対応可能となっている。

サポートは、3年間のハードウェア保証、90日間のソフトウェアサポートが付属し、有償サポートによって、同日中のハードウェア/ソフトウェアアップグレードや、24時間体制のオンサイトサービスも受けられる。



写真10-2 SGI 1200のマザーボード
オンボードに機能を盛り込んでいる。PCIスロットに挿入しているカードはテストのため標準では空き。

製品名	SGI 1200
CPU	Pentium 550 / 700MHz (シングル / デュアルCPU)
メモリ	128M ~ 2Gバイト ECC SDRAM (PC100、DIMMスロット×4)
チップセット	Intel 440BX
ハードディスク	9.1G / 18Gバイト Ultra2 SCSI (最大4台)
SCSIインターフェイス	オンボードUltra2 SCSI (AIC-7890)
RAIDコントローラ	オプション
グラフィックス	オンボードGD5480 (2MバイトSGRAM)
ネットワーク	オンボード10/100BASE-T (Intel 82559)
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	32倍速 ATAPI CD-ROM
拡張スロット	PCIスロット×2
I/Oポート	RS-232C ×2、パラレル×1、USB ×2
キーボード	オプション
マウス	オプション
ケース	2Uラックマウント
電源	252W
外形寸法 (mm)	475 (W) × 87.5 (H) × 631 (D)
重量	20.41kg (最小構成の場合)
OS	SGI ProPack 1.2 with Red Hat Linux 6.1
価格	69万円～



OpenGLアクセラレータ搭載のLinuxワークステーション

Silicon Graphics 230 Visual Workstation

日本SGI株式会社

0120-161-086

http://www.sgi.co.jp/

33万6000円～

日本SGIは、OpenGL 3Dグラフィックスをハードウェアサポートするワークステーション「Silicon Graphics 230 Visual Workstation」(以下SGI 230)を2000年5月より発売している。

なお、SGI 230の上位モデルとして、CPUにPentium 800MHzを最大2基まで搭載可能な「Silicon Graphics 330 Visual Workstation」と、Pentium Xeon 800MHzを最大2基まで搭載可能な、「同550」も発売されている。

Silicon Graphics 230

SGI 230は、同社初のLinuxワークステーションで、CPUにPentium 667MHzを採用したモデル(230-667)と、Pentium 733MHzを採用したモデル(230-733)の2種類が用意されている。

230-667は、メモリが128Mバイト、20GバイトのIDEハードディスクを搭載し、230-733は、メモリが256Mバイト、9.1GバイトのUltra2 SCSI対応ハードディスクを搭載している。

VProグラフィックスカードは、4本の256ビットエンジンを独立したパイプ

ライン処理で実行でき、2Dレンダリング以外に、T&L(トランスフォーム&ライティング)のジオメトリ計算をハードウェアで高速に実行する。これによりOpenGLの性能が向上し、スムーズで高速な3Dグラフィックスを実現している。

SGI 230にインストールされているSGI ProPack 1.2 for Linux-230

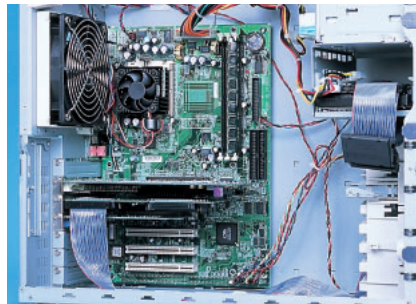


写真11-2 マザーボードと排気用巨大ファン
拡張スロットには、VProグラフィックス(AGP)、SCSIカード、サウンドカードが挿さっている。



Editionによって、3Dグラフィックス機能をハードウェアレベルで高速化したOpenGLを、Linux上でサポートしている。

なお、サウンドカードはCreative LabsのEnsoniqが装着されていて、専用のステレオスピーカが付属する。

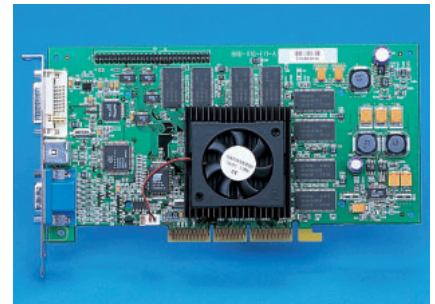


写真11-3 VProグラフィックスカード
nVIDIAのGeForceをベースとしたLSIを使用。ハードウェアサポートされたOpenGL環境を提供する。

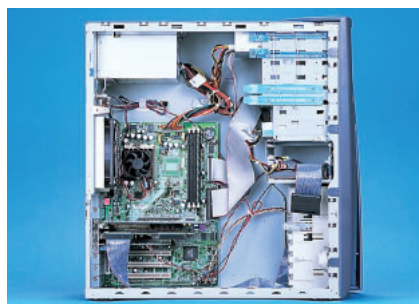


写真11-1 メンテナンスしやすいタワー型ケース
ドライブのネジ部分に装着しワンタッチで着脱可能なレールを採用(ドライブベイ横に予備が付属)。

製品名	Silicon Graphics 230 Visual Workstation
CPU	Pentium 667 / 733MHz
メモリ	128M / 256M ~ 1.5Gバイト ECC SDRAM (PC133、DIMMスロット×3)
チップセット	VIA Apollo Pro133A
ハードディスク	20GバイトIDE / 9.1GバイトUltra2 SCSI (最大3台)
SCSIインターフェイス	Ultra2-LVD/SE SCSI(Adaptec AHA-3950-U2B)
RAIDコントローラ	
グラフィックス	SGI VPro (32M / 64MバイトDDR、AGP 4X)
ネットワーク	オンボード10/100BASE-T (Intel 82559)
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	48倍速 ATAPI CD-ROM
拡張スロット	PCIスロット×5 (空き4、SCSIハードディスクの場合空き3)
I/Oポート	RS-232C×2、パラレル×1、USB×2、PS/2×2
キーボード	日本語109キーボード
	3ボタンマウス
ケース	ミドルタワー
電源	300W
外形寸法 (mm)	210 (W) × 489 (H) × 489 (D)
重量	13.15kg
OS	Red Hat Linux 6.1J、SGI ProPack 1.2 for Linux-230 Edition
価格	33万6000円～

思い切った低価格を実現したLinuxサーバ

Netfinity 1000

日本アイ・ビー・エム株式会社

0120-04-1992

http://www.ibm.co.jp/

13万8000円～

日本アイ・ビー・エムは、2000年4月よりPCサーバ「Netfinity（ネットフィニティ）」の全モデルでLinuxを正式サポートしている。Netfinityは、ミドルタワーのエントリーサーバから、ラックマウント専用モデル、8CPUのSMP対応のエンタープライズサーバまで、16種類もの幅広いラインナップがあり、ユーザーのニーズに合わせてモデルを選択することができる。

Netfinity 1000

Netfinity 1000は、その中の一番下のモデルで、CPUはPentium 600MHz、メモリは64MバイトのECC SDRAM、10.1GバイトのIDEハードディスク、CD-ROM、フロッピードライブ、キーボード、マウスという基本構成の場合、TurboLinux Server日本語版6.0(OEM Version)を同梱して、13万8000円という低価格を実現している。

グラフィックスコントローラのTrio64 3Dと、10/100BASE-T対応のネットワークコントローラをオンボードに搭載しているため、PCI/ISAの拡張スロットは自由に使える。拡張スロットはマザーボード上ではなく、ライザカードを使用して本体内部の仕切り板から下向きに装着するようになっている。PCI×3スロットとISA×3スロットが用意されているが、最近ではISA用カードを使うことがほとんどないので、PCIの数を増やしてほしいところだ。

3.5インチのドライブベイには、ハー

ドディスクを3台まで装着することが可能で、10.1GバイトのIDEハードディスクを増設することで、最大30.3Gバイトまで拡張できる。電源ユニットは330Wと容量は十分である。

OSは、Red Hat Linux 6.1日本語版またはTurboLinux Server日本語版6.0で1000シリーズを除くNetfinityには、Linux版の「ロータス ドミノ」が

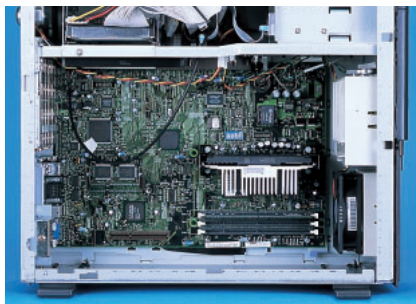


写真12-1 Netfinity 1000 本体内部
少し見えにくい仕切り板から下向きに拡張スロット用コネクタがあり、マザーボードはすっきりしている。



標準添付される。

Netfinityのサポートは、故障の際にメンテナンス要員が24時間/365日対応する1年間オンサイトサービスに加え、3年間の部品保証サービスを実施している。



写真12-2 本体上部にある電源とドライブベイ
ドライブベイには、5インチベイ1台と3.5インチ2台分の空きスペースがある。

製品名	Netfinity 1000
CPU	Pentium 600MHz
メモリ	64M～768Mバイト ECC SDRAM (PC100、DIMMスロット×3)
チップセット	Intel 440BX
ハードディスク	10.1Gバイト IDE (最大3台)
SCSIインターフェイス	
RAIDコントローラ	
グラフィックス	オンボード Trio64 3D (4Mバイト)
ネットワーク	オンボード 10/100BASE-T (Intel 82557)
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	40倍速 ATAPI CD-ROM
拡張スロット	PCIスロット×3、ISA×3
I/Oポート	RS-232C×2、パラレル×1、USB×2、PS/2×2
キーボード	日本語109キーボード
マウス	3ボタンマウス
ケース	ミドルタワー
電源	330W ATX
外形寸法 (mm)	200 (W) × 445 (H) × 492 (D)
重量	17.3kg (最大)
OS	Red Hat Linux 6.1JまたはTurboLinux Sever日本語版6.0
価格	13万8000円～



システム管理ソフト付属のエントリーサーバ

HP NetServer E200

日本ヒューレット・パカード株式会社

03-3335-8333

http://www.jpn.hp.com/go/netserver

22万9800円～

日本ヒューレット・パカードは、HP NetServerシリーズのエントリーモデル「HP NetServer E200」に、Red Hat Linux 6.2Jとシステム管理ソフト「HDE Linux Controller」をプレインストールした「HP NetServer E200 Linuxモデル」(以下E200)を、2000年7月より発売している。

HP NetServer E200

E200は、CPUにPentium 600MHz (FSB 133MHz)を採用。メモリは標準で64Mバイト、ハードディスクは9.1GバイトのUltraWide SCSI対応で、AGPスロットにはATI Rage IIcグラフィックスカードが、PCIスロットには、10/100BASE-Tに対応したネットワークカード、UltraWide SCSI (Adaptec AIC-7880)インターフェイスカードが挿さっている。電源は305Wで、ハードディスクや拡張カードの増設に十分な容量を持っている。

また、Red Hat Linux 6.2Jデラックス製品版パッケージと、ホライズン・デジタル・エンタープライズのHDE Linux Controller製品版が付属している。HDE Linux Controllerは、Webブラウザを使ってLinuxシステムの保守管理が行えるので、専任の管理者がない場合に便利である。なお、HDE Linux Controllerは、8月に新バージョン2.0の発売が予定されているが、E200購入者には無償バージョンアップが提供される。

サポートは、1年間のオンサイト保守

が付属していて、故障連絡の翌営業日に訪問修理が受けられる。オプションの各種サポートパックを利用することで、優先訪問、24時間×7日間訪問修理、6時間ハードウェア修復といった、高い緊急度に対応するサポートが受けられることができる。

また、E200に追加してグループウェアのサイボウズOffice3製品版をインス



トールした「HP NetServer E200サイボウズモデル」も用意されている。サイボウズOffice3を60日間無料で試用することができ、継続して使用する場合には、サイボウズより必要な機能のライセンスを購入する。

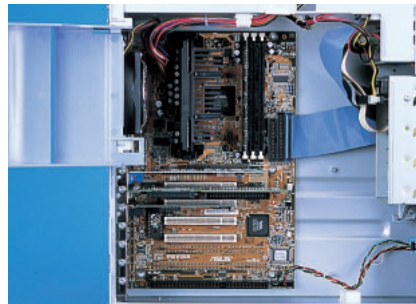


写真13-1 HP NetServer E200マザーボード
チップセットはVIA Apollo Pro133AでFSB 133MHz。ファン部分には開閉式のカバーが付いている。

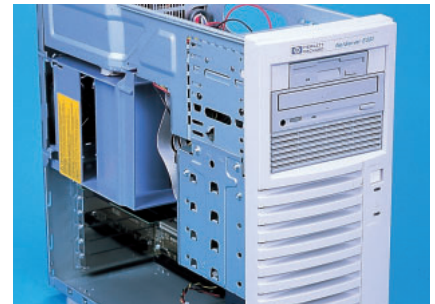


写真13-2 ドライブベイとCPUファン用カバー
ハードディスクは、7200rpmの9.1GバイトのUltraWide SCSI対応。CPUの熱をカバーで覆いファンで排気。

製品名	HP NetServer E200 Linuxモデル
CPU	Pentium 600EBMHz
メモリ	64M ~ 768Mバイト ECC SDRAM (PC133、DIMMスロット×3)
チップセット	VIA Apollo Pro133A
ハードディスク	9.1G / 18.2GバイトUltraWide SCSI 7200rpm (最大4台)
SCSIインターフェイス	UltraWide SCSI (Adaptec AIC-7880)
RAIDコントローラ	
グラフィックス	ATI Rage IIc AGP (4Mバイト)
ネットワーク	10/100BASE-T (Intel 82559)
フロッピーディスク	3.5インチ 1.44Mバイト
CD-ROM	40倍速 ATAPI CD-ROM
拡張スロット	PCIスロット×4 (空き2) ISA×1
I/Oポート	RS-232C×2、パラレル×1、PS/2×2
キーボード	日本語106キーボード
マウス	2ボタンマウス
ケース	ミドルタワー
電源	305W
外形寸法 (mm)	180 (W) × 434 (H) × 497 (D)
重量	16kg
OS	Red Hat Linux 6.2Jデラックス
価格	22万9800円～

隠喩としてのLinuxコンピュータ

Gnutella はインターネットの悪魔、それとも天使？

文：豊福 剛
Text : Tsuyoshi Toyofuku

“ Fire and water may be good servants, but bad masters. ” という諺がある（《火と水は役にも立つが害にもなる バカとはさみは使いよう》の意）。この諺、まるでGnutella（グヌーテラ）のことを言っているようで面白い。

最近ではCDすらほとんど買わなくなり、音楽そのものへの関心も低下していることもあって、MP3まわりの話題については、フォローしていなかった。そのため、Napster（ナップスター）というソフトがあって、ユーザー間でMP3ファイルを共有したり検索できるといったことは、恥ずかしながらGnutellaの後から知った次第だ。

NapsterとGnutella、ユーザーのローカルなディスクにあるファイルを相互に共有しているという点では、どちらも機能的に似たものに思えるが、NapsterがMP3ファイルの交換を目的としているのに対して、Gnutellaはファイル形式を限定していない。そのため、GnutellaというとCDから違法コピーした曲だけでなく、Warezと呼ばれる類のアプリケーション製品の違法コピーを含む、UG（アンダーグラウンド）系の温床といった印象があるのは否定できない。しかも、名前からしてGNUと関係ありそうであるし、ソフトウェアや音楽が商品であることを悪と考えるGNUから派生した、何か新しいプロジェクトなのだろうかと思っていた。

もともとGnutellaは、Winampという有名なMP3再生ソフトを手掛けているNullsoft社のスタッフが開発していたのだが、これをWebで公開した直後に、Nullsoft社の親会社であるAOLの判断で、公開にストップがかかった。しかし、わずか1日足らずの間にダウンロードされたGnutellaは、ハッカーによってリバース・エンジニアリングされ、そのクローンを開発するさまざまなコミュニティが立ち上がっている。Nullsoft社のスタッフがGnutellaと命名した背景には、将来的にGNUソフトウェアにしたい意図があったらしい。

サーバントのアソシエーション

Gnutellaが形成するGnutellaNetにおいては、そこに接続しているすべてのコンピュータが、ネットワークのノードとして位置づけられる。クライアントサーバモデルにおいては、データもトラフィックも、サーバという中心に集中するのに対して、GnutellaNetにおいては、データはすべてのノードに分散している。というのも、

GnutellaNetにおけるデータとは、すべてのノードに存在する公開ファイルの集合であり、ファイルの転送はノード間で直接実行されるため、サーバという特権的な中心は存在しない。すべてのノードが、クライアントであると同時にサーバであることから、Gnutellaではこれをサーバント (Servant) と呼んでいる。サービスを消費する客 (クライアント) ではなく、主人 (マスター) に支配された奴隷 (スレーブ) でもなく、ネットの使徒、しもべとしてのサーバントなのだ。

インターネットでファイルをダウンロードするとき、通常はHTTPやFTPが使われるが、Gnutellaは、もうひとつのファイル転送プロトコルというわけではない。Gnutellaは、検索を分散処理によって実行するところに、機能的な特徴がある。

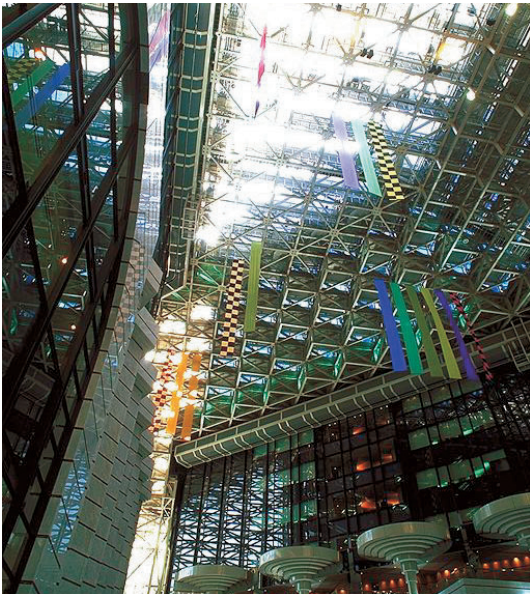
検索にヒットしたファイルをダウンロードするときには、そのファイルが格納されているサーバントとの間で、HTTPによるファイル転送が実行される。このとき転送元のサーバントのIPアドレスは隠されていない。これは転送先についても同様であり、自分が公開しているファイルをダウンロードしているサーバントのIPアドレスもわかる。IPアドレスが公開されてしまう点では、IRCなどと共通するセキュリティ・リスクはそれなりに存在する。

現在のところ、Gnutellaでの検索は、ファイル名が対象になっている。ファイル名による検索という点では、Archieに似ている。ただし、GnutellaNetはサーバントの集合としてダイナミックに形成されるので、検索の対象となるファイルの集合もまた、ダイナミックに変化する。ただし、これらのファイルの集合に対するインデックスが、どこか特定のサーバに置かれていて、そのインデックスに対して検索が行われるのではない。

検索のクエリーを発行したとき、そのクエリーは、まず直接接続されているサーバントに対して送られ、さらにそのサーバントに接続されている別のサーバントに伝播されていく。このように、検索クエリーがサーバントからサーバントへ、次々に連鎖的に伝播されていく。そして、連鎖のどこかにあるサーバントで、検索クエリーにヒットするファイルが存在すれば、そのヒット情報が応答メッセージとして次々に送られてくる。

当然のことながら、GnutellaNetに接続しているときには、他のサーバントが発行した検索クエリーが送信されてくる。すべてがサーバントである以上、クライアントサーバと違って、検索クエリーを送信するだけではない。他





のサーバントからの検索クエリーの受信も同時に実行しているのだし、公開しているファイルの中に、検索クエリーにヒットするものがあったときは、クエリーを発行したサーバントに対して、そのファイルをアップロードすることになる。

MP3 を検索してしまう現実

このように、Gnutellaのネットワーキングに対するアプローチは、ある意味でとてもシンプルだ。GnutellaNetを流れる検索クエリーをモニタしていると、いままさに機能しているネットワークの動態が実感できる。接続中のGnutellaNetを構成するホスト(サーバント)とファイルの数が、時々刻々と増えていくのを見てみると、ダイナミックに生成されるネットワークそのものの感触が伝わってくる。数千ものサーバントが互いに協調しながら形成する一時的自律ゾーンとしてのGnutellaNet。

もちろん、Gnutellaは始まったばかりの試みであり、ソフト自体もけして洗練されたものではない。スケーラビリティやセキュリティなど、解決すべき課題も少なくない。しかし、今後のインターネットにおいて大きな潮流になっていくのは確実だと直感している。Gnutellaに対する期待と興奮は、Mosaicを初めて体験したとき以来のものだ。

ただし現状では、Gnutellaを使ってダウンロードしたいと思うファイルの筆頭は、やはりMP3になるだろう。ある1曲だけ聞きたいのだが、CDを買うほどでもない、そんなときに、Gnutellaで探してみようか、という人は少なくない。

MP3のファイル名の多くは、アーティスト名と曲名で構成されているので、ファイル名に対する検索であっても、十分機能する。個々のサーバントでの検索処理も、ファイル名だけを対象にしているので、それほどの負荷にはならないだろう。1曲あたりおよそ3~5Mバイトとはいえ、転送元の帯域が狭ければ、場合によってはダウンロードに数時間もかかってしまう。ただし、いったんダウンロードしてしまえば、MDと同等の音質の音楽が聴けるわけで、ファイルの容量やダウンロードにかかる時間といったコストに見合うだけの満足度が、MP3にはある。

これが、テキストや画像になると、ファイル名だけの検索では、ほとんど役に立たないだろう。音楽は、アーティスト名や曲名によるラベリングが、非常に強力に機能する例外的なコンテンツであるのだ。それでは電子ブックは

どうか。音楽と同様に、著者や書名、さらにはISBNコードによるラベリングが機能するはずだが、画面で読むという行為は、MP3で音楽を聞くほど手軽なものではない。ただし、音楽ほどではないにしても、入手困難な書籍を電子ブックの形式で入手したいというニーズは、確実に存在するだろう。

NTT ドコモという門番

音楽配信については、NTTドコモによるPHSを使った実験サービスがある。PHS実験端末にダウンロードした曲データは、メモリカードに書き込まれる。これを再生用のプレーヤやパソコンに装着して、再生する。PHSや携帯の電話機と再生プレーヤが一体になった端末も出てくるはずだ。W-CDMAが導入される2001年には、384kbpsでのダウンロードも可能になるらしい。

このような携帯電話によるサービスは、あらゆる意味でGnutellaの対極にある。それは音楽流通の方法論として対極にあるというだけでなく、電話の延長にあるサービスとインターネット的世界観の対立を象徴しているように思う。

インターネット的世界観においては、階層のないフラットなノード間の関係の総体として世界は構成されている。Mosaicの頃のWWWや現在のGnutellaに知的興奮を覚えたのは、そこからインターネットのフラットな構造を読み取ったからだ。

しかし、WWWの爆発的浸透によって、サーバとクライアントの階層構造は強化され、あげくの果てにiモードが登場するに至った。iモードは、実際に使ってみれば、そこそこ便利ではあるらしい。けれども、「iモードで日本のIT革命を」といったオヤジの言い草を耳にするたびに、IT革命の先にある未来は、きっと今より居心地の悪い世界になっている気がして仕方ない。特権的な門番によって独占されたゲートウェイという構造は、必然的にさまざまな利権をはびこらせるだろう。

ほんとうの革命は、Gnutellaのような試みの、さらにその先にあるのだとしたら、IP接続サービスの64kbpsという帯域は、あまりにも貧しすぎる。

Profile

とよふく つよし

1962年生まれ。メディアデザイン研究所技術顧問。訳書に『Javaプログラムクイックリファレンス』『Java分散コンピューティング』(オライリージャパン)『GIMPパーフェクトガイド』(エムディーエヌコーポレーション)などがある。

ITの時代に政府は何をするべきなのだろうか

文：安田幸弘
Text：Yukihiro Yasuda

「イット革命」だ!!

年寄りだからって、コンピュータがわからないなんて思っちゃいけない。最近、Linuxを駆使するスーパーおじいちゃんも少なくないらしい。だけど、ふだん「日本は神の国」だとか、「教育勅語にはいいことも書いてある」なんてことばかり言っている日本の総理大臣が「イット革命」なんて言ってるのを聞くと、「ホントに日本の21世紀は大丈夫なのかなあ」と思ってしまう。おそらく目端の利く官僚に「今、世界はIT革命の時代です」なんて話を吹き込まれたんだろうけど、どうもその官僚たち、特に郵政省あたりの官僚は、10年ぐらい遅れた国際センスでコトを進めようとしているのである。

そもそも、日本の政治家や官僚の中で、「IT革命」ってやつに「革命」なんていう大げさな言葉が入っていることの意味が理解できる人はどれぐらいいるんだろう？ 情報通信分野での進歩が「革命」と呼ばれるのは、これまでのモノの考え方が通用しなくなったということの意味しているわけで、それを昔ながらの価値観や政治手法・行政手法から抜け出せない政治家や官僚があれこれ論じること自体が「わかってねーな」って感じなのだ。

ICANNで国益を叫ぶ時代錯誤

.....そんなことを痛感させられたのが、このあいだ、横浜で開かれたICANN (<http://www.icann.org/>) の会議のときだった。ICANNとは、これまでNSI (<http://www.networksolutions.com/>) が独占してきたインターネットのドメイン名やIPアドレス、プロトコルの割り当てなどを調整する機関として、特定の国家や勢力から独立した組織として発足した非営利団体である。

ユーザーひとりひとりの貢献によってボランタリーに形成されてきたインターネットの伝統を踏まえて、世界中のインターネット利用者なら誰でも、ICANNの決定プロセスに参加できるという開かれた組織であるという点が特徴的だ。

ある意味では理想主義的とも言える地球規模の団体なのだが、ICANNは設立から日が浅い未成熟な団体で、決定すべきこと、決められないことは山のようにある。今回の横浜会議で焦点になったのは、この秋に行われるICANNの理事選挙だった。

理想主義的な建前を持つICANNだが、理想と現実はずし一致しないのは世の常である。特に人事ってのはいろいろ生臭いこともあるんだろう。舞台裏では、さまざまな企業や勢力が主導権争いを繰り広げているらしく、このままではICANNが「サイバー世界のWTO」になってしまうのではないかと危惧する世界各国の市民社会グループが、ICANNへの関心を高めている。横浜での会議にも各国から、企業ばかりでなく、非営利団体や市民社会団体が集まってきた。

シアトルのデモみたいな派手な行動はなかったものの、市民側からICANNに発言していくための戦略をめぐって、連日、夜遅くまで熱心な議論が続けられた。そしてそのときに議論されたことのひとつに、日本がICANNを国益確保の場と位置付けて、理事選挙に秋に予定されているICANN理事選挙で日の丸を背負った日本人理事を送り込み、インターネットの世界をコントロールしようとしているということだった。

もちろん政治家や官僚が国益を論じるのは彼らの仕事であって、それ自体は非難されるようなことじゃない。ITをめぐる交渉の中では、国益を賭けて真正面から交渉しなけりゃいけない分野もあるだろう。また、日本はアジア地域の事情をインターネットの世界に反映させるための努力をしなければならぬ立場でもある。

ところが官僚の要請による数集めの結果、日本からの一般会員の登録数が、世界のインターネット先進国であるアメリカを抜いてダントツの第一位になってしまったから、コトはややこしい。この結果はあまりにも不自然だ。特にICANNに取り組んでいる世界のNGOは、一方的に企業にだけ有利なシステムは市民の権利を奪うと強く反対しているのだが、日本政府がインターネットをIT産業に有利なものにするために組織的な選挙活動をしているというのだから、日本政府のやり方に不満の声が上がったのも当然である。

インターネットの分野は、グローバル、オープンネス、そしてボトムアップというキーワードで動いている。本音はともかく、とりあえず国家だとか自分の利益だとかは棚に上げて(または棚に上げたような顔をして)、世界のインターネットコミュニティ全体の利益を前提に誰にでも納得できるような透明な手続きでコトを進めなければ、まとまるものもまとまらなくなってしまう。もちろん、国際政治の駆け引きはどこかでやらざるを得ないのだけれど、少なくともICANNという場所を露骨に駆け引きの場に作るなんてのは論外だ。郵政省の役人が「eJapan アズ・ナンバーワン」なんて叫んでみせるのは愛嬌としても、役人と財界が「ICANN対策官民会議」なんていう名前の怪しげな会議を開いてみたり、公然と「日本の国益のために日本人理事をICANNに送り込め」なんて発言をしたり、あるいはこっそりと企業に票まとめを依頼して、インターネットの世界を日本に都合がいいようにコントロールしようなんてことを本気で考えてるんだったら、これは相当危ない。

人間の共有財産としてのIT

たとえばどこかの政府がLinuxを自国に有利な

OSにするつもりで、Linuxのソースツリーの管理者として国益を背負った人物を入れると言って来たら、Linuxのコミュニティは納得するだろうか。.....いや、納得する、しないというレベル以前に、あまりにパカパカしくて相手にされないかもしれないけれど。

ICANNはLinuxよりもはるかに現実的な影響力が大きいか、図式としては似たようなもの。もし、政府がインターネットやオープンソースに貢献したいと思うのだったら、国益なんてものを背負った人物を送りこむのではなく、開発資金を援助したり、有益なコードをコミュニティにコントリビュートするほうがずっとウケもいいだろうし、国際貢献にもなる。そして結局は国益とやらにもプラスになるはずだ。

日本はITを景気回復の道具ぐらいにしか思っていないのかもしれない。でも、日本をはじめとする先進国は、いやおうなくITの影響下に置かれる世界に対する責任を負っている。ICANNでも、情報格差の問題や先進国にばかり有利な手続きに対して途上国からさまざまな発言があったのだが、たとえば日本をはじめ、先進国は人類の共有財産であるフリーソフトウェアへの支援などを通じて、安価で優秀なシステムを提供することだってできるだろう。

ITの時代は情報が国境を超えて瞬時に行き来する時代である。日本をはじめとする先進国の政府には、ITを単に金儲けの道具としてだけ考えるのではなく、人類の共有財産として考える視点を持ってもらいたいものだと思う。

Profile

やすだ ゆきひろ

生業はテクニカルライター。原稿書きのかたわら、(株)市民電子情報網のボランティア社長兼技術スタッフとして、NGO向けプロバイダのネットワーク運営に携わる。

ドクターShiodaの

ギョーカイ SnapShot

PC業界今月の話題

文：塩田紳二
Text: Shinji Shioda

- 7・19 アップル、Mac G4 Cube 発表
- 6・28 Intel Willametteの製品名をPentium4に
- 6・23 日立とRambusが和解
- 6・22 マイクロソフト .NET 構想を発表
- 6・19 Windows ME 9月14日より発売開始

暑い日が続きますが、いかがお過ごしでしょうか。楽しい夏休みをお過ごしでしょうか？真夏のパソコンライフは？

我が家では、コンピュータの熱対策が夏の重点項目で、ちょっと油断すると仕事部屋が温室のようになってしまい、あちこちのコンピュータがCPU過熱でピーポー、ピーポー泣き出します。このオーバーヒート警告機能って便利なんだか、不便なんだか.....。



NECは、業務向けタッチパネル装備のノートPCをベースにCrusoeを使った試作機を展示。

インテルはPentium4

インテルは次世代CPUであるWillametteの製品名をPentium4とすることを発表した。まあ、II、IIIと続いできたので、4というのは順当なネーミング。以前から、Pentiumのブランドを生かせる名前にするとはいつかだったので、大方の予想通りというところ。いちおう、ローマ数字からアラビア数



IBMのCrusoe試作機は、ThinkPad 240をベースにしたもの。

字になっているが、これは4からはIVと(5-1)という表記になり、わかりにくいからだと思われる。

Linuxを使ったインターネットアプリケーションを発表したインテルだが、今度はLinux用のUPnP (Universal Plug and Play) 開発キット「Intel Universal Plug and Play Software Development Kit V1.0 for Linux」を発表した。これは、マイクロソフトがSunのJiniに対抗して提唱したUPnP対応のソフトウェアを開発するためのもの。マイクロソフトはどうか知らないが、少なくともインテルは、家庭用ネットワーク機器としてLinuxを採用したものが、いくつも登場すると見込んでおり、それらをサポートするためにこのキットを提供するのだと思われる。

ネットワーク機器は、ほぼオフィスにいきわたり、今後のターゲットは、家庭市場だといわれている。家庭用の無線LAN機器や、電話線を使ったネットワーク機器も登場しているが、安価なイーサネットを使うという方法もある。しかし、問題は企業にはネットワーク管理者はいるが、家庭にはそういう人物がいるとは限らないことだ。このために注目されているのが、JiniやUPnP、あるいは家電メーカー主導のHAViなどである。

こうしたネットワーク環境には、多かれ少なかれ、Linuxなどの安価なOSをベースにした専用機器が入ってくる可能性が高い。というか、このようなネットワーク対応機器を作る場合に、いまではLinuxを採用するところが多くなるだろうということだ。このため、Linux関連でもいまや組み込み系はちょっとしたブームだし、実際、Linuxを組み込んだ機器なんてのもいくつか登場している。インテルは、自社製品、特にPentiumなどのCPUを採用したネ

ネットワーク機器などで、Linuxの採用が増えると見ているし、後押しもしているわけだ。また、WindowsCEなどのマイクロソフト系のOSは、インテルが放っておいてもマイクロソフトがなんとかすることになり、インテルの出番はないからでもある。

Transmeta採用 ノートの出荷は近い

さて、6月末のPC EXPO (米国ニューヨークにて開催)では、TransmetaのCrusoeを使ったノートパソコンの試作機が何社からか展示された(筆者も取材に行きました)。展示したのは、IBM、NEC、富士通、日立で、富士通は基板のみの展示。製品の完成度は高そうで、Crusoeが出荷されれば、そう遠くないうちに製品も出るのではないかと思われる。

インテルのほうは消費電力や熱の問題があって、なかなかクロックを上げることができないでいる。ACとバッテリー駆動時でクロック周波数と電源電圧を変えるSpeedStep対応のCPUは出しているものの、ファンや冷却のための部品が必要。これに対して出展されていたCrusoe搭載ノートパソコンは、ファンがなく、どこを触っても熱を感じられなかった。昨年、ノート分野は

AMDが低価格戦略でインテルの市場を食ったのだが、今年後半以降、Transmetaがこの分野でかなりのシェアを占めるかもしれない。

もう1つ出展されていたのは、Transmetaが設計したWebpadと呼ばれるインターネットアプライアンス。これは、液晶を装備した板状のインターネットアクセス機器で、Crusoeの下位CPUであるTM3200を採用し、OSにはMobile Linuxを使っている。

説明員の話によると、Mobile Linuxはインテルベースのバイナリであり、Crusoeのネイティブコードバイナリではないとのこと(Crusoeはx86バイナリをVLIWのネイティブコードに変換して動作する)。これは、現在Linuxはインテルベースのものをもっとも普及しており、そのまま使ったほうがドライバなどの面で有利だかららしい。また、Crusoeは基本的にAT互換機を構築するためのCPUで、いわゆるノースブリッジ相当の機能をCPU内に持つため、すなわちAT互換機としたほうが簡単に作れるのである。

なお、MobileLinuxは、こうした機器のためのコンパクトなディストリビューションであり、特別なLinuxというわけではないという。実機では、XFree86が立ち上がり、その上で

Netscapeのブラウザが動作していた。

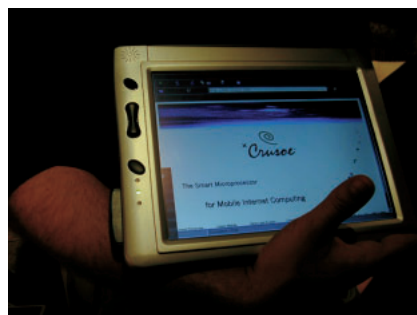
WebPadはおそらく、TransmetaからCPUとともに基本設計データも入手可能と思われ、たとえば、いままでPCを手がけていない家電メーカーなどでも簡単に製造できるように考慮されていると思われる。OSまで提供するのでは、このためで、ほとんど手間なしで機器の製造に入れるのだろう。S3がCrusoe発表後に早々にインターネットアプライアンスを製造することを発表しているが、おそらく製品はこのようなものになると思われる。

もうひとつTransmetaブースでは、Linuxを使った、ゲートウェイサーバも展示されていた。これは、ケーブル接続やADSL接続の場合のゲートウェイサーバとして使われることを想定したもので、ローカル側から見れば、インターネットアクセスのためのサーバ(たとえば、DHCPやDNS)として動作するもの。Crusoeが低発熱であるためファンがなく、また、低消費電力であるためUPSの内蔵が可能で、しかもバッテリーでの長時間動作も可能だという。CrusoeというとノートやPDA的なものが注目されるが、意外に家庭用のサーバ機器なんかにも向いているのかも。

では、今月はこのへんで。



日立は、PriusNoteがベースだが、筐体の色が現行製品とは違っていた。



WebPadは、Linuxを使っている。写真は、Netscapeを使ってTransmetaのホームページを表示しているところ。



WebPadは非常にコンパクトながら、PCカードスロットやUSBインターフェイスを備えている。LCDには、タッチパネルが装備されており、ソフトウェアキーボードで文字入力も可能。

日刊アスキー Linux on Linux magazine

http://www.linux24.com/

日刊アスキー Linuxの舞台裏

～日刊アスキー Linux Weekly Newsletterのアンケート～

日刊アスキー Linuxも、先日無事1周年を向えることができました。日ごろご愛読いただいている読者の皆さま、ありがとうございます。

今回はこれまでに集めた各種アンケートの結果をご紹介します。

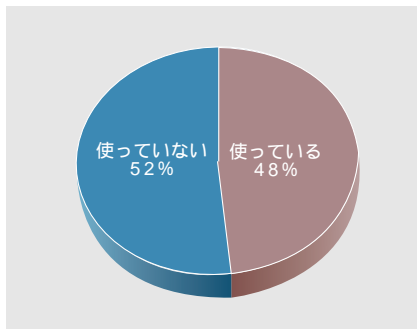
(日刊アスキー編集部・吉川)



Webサイトがオープンしてから1年経ったわけだが、毎週月曜日にお届けしているメールサービス「日刊アスキー Linux Weekly Newsletter (以下 Newsletter)」も、12月1日の創刊0号から、すでに31号を数えるまでになった。

今回は、このメールサービスで毎週行っているLinuxに関するアンケートの結果をご紹介します。Newsletterでは、毎週アンケートを行い、その結果を翌週のメールサービスに掲載している。その週ごとのトピックを取り上げ、それに対する意見をうかがってみようというわけである。だれでも簡単に投票できるので、ぜひご参加いただきたい。自由回答欄も意図的に設けるようにしてあるので、興味のあるお題が出た場合には、存分に主張していただければ

図1 ノートPCとLinux



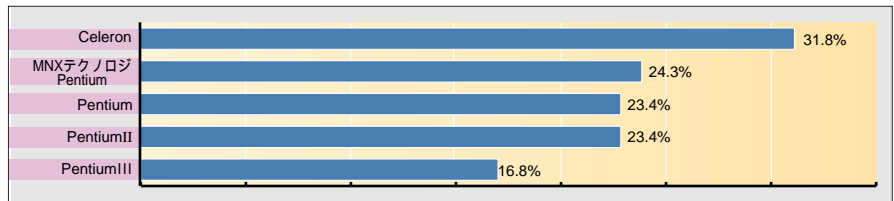
Linuxをノートパソコンで使っていますか？

嬉しい限りである。

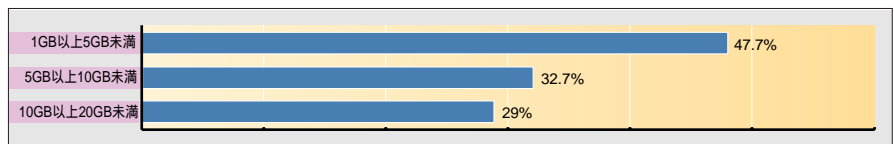
ハードウェア環境とソフト

宣伝はここまでにして、本題に入ろう。まずは、Linuxとハードウェアについてのアンケート結果をご紹介します。該当するアンケートは、2000年1月24日号掲載の「ノートPCとLinux」(図1)と、2000年4月10日号掲載の「Linuxマシンのスペックは？」(図2)

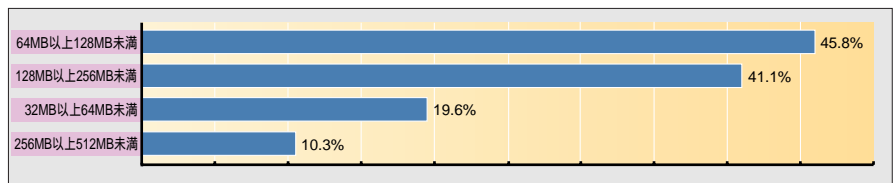
図2 Linuxマシンのスペックは？



x86系プラットフォームでLinuxをお使いの方に質問します。CPUの種類をお答え下さい。(複数回答)



ハードディスクの容量はどれくらいですか？(複数回答)



メモリの容量はどれくらいですか？(複数回答)

である。「ノートPC」のほうだが、実際にノートで使っている人が半数近くにおよぶほか、今後ノートPCでLinuxを使いたいと回答した人が81%もいたのである。IBMやNECなどのハードウェアメーカーがLinux対応を始めている今、環境が整ってきて嬉しいと感じる方も多いのではないだろうか。

また、「Linuxマシンのスペックは？」だが、これはCPU、ハードディスク、メモリ容量についてアンケートしてみ

た。ここから浮かび上がる皆さんお使いのマシンの平均像は、CPU = Celeron、ハードディスク5Gバイト未満、メモリ64~128Mバイト未満といったところである。ご自分のシステムと比べてみていかがだろうか？ それにしても、Celeronは妥当だとして、2位と3位がPentiumとは。KDEやGNOMEを使うとちょっと辛いと思うが、こうしたCPUをまだまだ現役で使えるのがLinuxのいいところなのかもしれない。

続いてソフトウェアである。ここでは、安定性を求める声が目立つことがわかった。抜粋したのは、2000年2月14日号「商用ディストリビューションのバージョンアップについて」(図3)と、2000年3月20日号「Mozillaについて」(図4)、2000年4月3日号「Vine Linuxについて」(図5)である。「商用～」のみは2位だが、ほ

かの2つは圧倒的な差を付けて安定性が1位である。「商用～」にしても、カーネルやライブラリのバージョンアップは、ある意味安定性につながると捉えることもできるだろう。Windows文化に嫌気がさしている人も多いのかもしれない。

ディストリビューション

ところで、みなさんはどのようにディストリビューションを選ぶのだろうか？ 2000年4月24日号「Linuxディストリビューションについて」(図6)では、現在どのディストリビューションを使用しているかを質問し、さらにその理由をうかがった。ちなみに、使われているディストリビューション第1位はTurboLinuxであったが、ディストリビューションの人気は日々入れ替わっているため、アンケート実施日から3

カ月近くたった現在では、また違った結果になるだろう。

むしろ、ディストリビューションの選択理由のほうが、今見ても色あせない数値として見るができるだろう。ここでの第1位は、なんと製品そのもののスペックなどではなく、ディストリビューターの信頼度であった。まずは技術力に定評のある集団が一番強いという健全な構図といえよう。また、2位の「フリー版のできがよいので」ディストリビューションの採用を決めるという結果にも注目したい。あるディストリビューターの方から「フリー版の配付数が増えれば増えるほど、パッケージ版(商用版)の売り上げも伸びる」といった話をうかがったことがあった。ディストリビューションのフリー版は、ディストリビューターにとってマーケティング的にも大切な存在なのかもしれない。

図3 商用ディストリビューションのバージョンアップについて

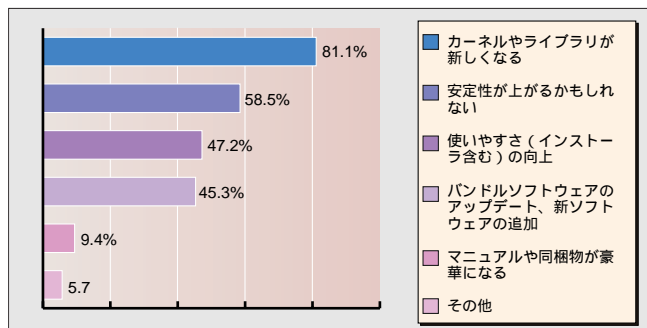
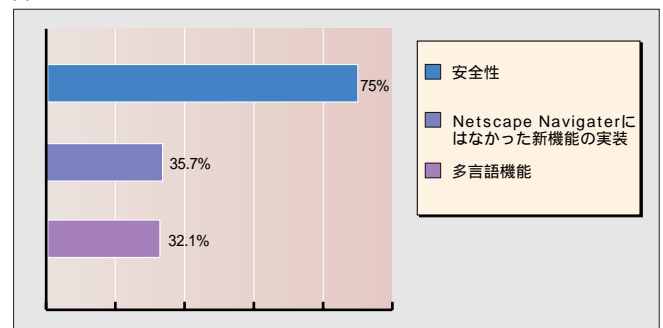
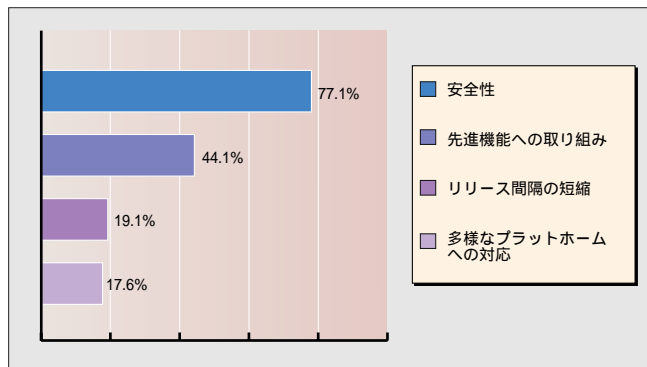


図4 Mozillaについて



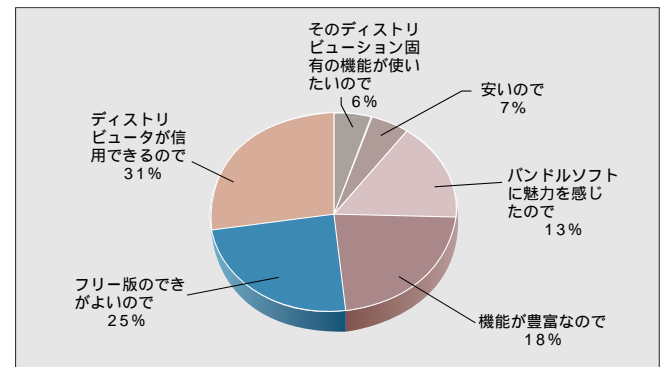
“Mozilla”というソフトウェアに期待することはなんですか？(複数回答)

図5 Vine Linuxについて



今後「Vine Linux」には、どのような点を期待しますか？(複数回答)

図6 Linuxディストリビューションについて



ディストリビューションを選択した理由は？(複数回答)

初級Linuxer養成講座

第12回 リンクについて

前回の本連載で、Linuxでのファイルの移動や削除には「リンク」という概念が関係していることに触れた。Linuxのファイルシステムの特徴であるリンクには、「シンボリックリンク」と「ハードリンク」の2種類があり、これらの特性を知ることでもうまく使い分けることができるのだ。Linuxを使いこなすうえで避けて通ることのできない「リンク」について少し詳しく見てみよう。

文：竹田善太郎
Text：Zentarō Takeda

約10年間、故障することもなく動き続けてきた自宅のビデオデッキが、ヘッドをクリーニングしても再生画面のノイズが消えない状態になってしまった。おそらくヘッドの磨耗が進んだためだろう。ノイズの発生以外に異常はまったくなく、メカニズムの動作や録画の状態も悪くない。おそらく、2~3万円の修理費をかければ、ヘッドを新品に交換してまだ使い続けることができるのだろうが、2万円出すのなら新品が買えるということで、新しいビデオデッキを購入することにした。

筆者はいわゆるAVマニアではないので、ビデオデッキに要求するのは故障せずに、まともな画が映ればよいということだけだ。だから、特に深くは考えずに、いままで使ってきたデッキと同一メーカーの製品を購入した。最近流行の3次元ノイズリダクション機能がついた普及機だ。

さすがに最新機種だけあってメカの動作もすばやく、録画再生機能にとりたてて問題はない。しかし、ノイズリダクションを有効にしたときの再生画像がいただけない。とくに、映画などで近景や遠景に「ぼかし」の効果が使われていたり、霧や霞の中などの「ぼ

んやりとした」場面になると、正視に堪えないほど不自然な画像になってしまう。流れる霧の中で、人物が静かに立っているような場面は、白いのっぺりとした寒天の中で、不定形のアメーバ状生物がうごめいているとしか形容できない画面になるのだ。ノイズリダクションをオフにすれば、そのような不自然さはなくなるのだが、今度は画面全体のノイズが目立つようになって、画質自体が10年前のデッキよりも悪くなってしまった。

どうやら、「3次元ノイズリダクション」のチューニングが、野外でフィルム撮りされた映画ではなく、スタジオでビデオ撮りされたテレビ番組を対象に行われているためらしい。また、ビデオデッキ本体のハードウェアとしての品質は、コストダウンのせいで10年前のものにくらべて格段に落ちており、それを安価なデジタル技術でごまかしているようなのだ。

最近の家電製品の品質の悪さは承知していて、今回のビデオデッキに関してもある意味で予想通りだったので、腹を立てているわけでもなく、落胆しているわけでもないのだが、なん

だかデジタル技術が悪用されているような気がして、ちょっと悲しい。目に見えない部分は徹底的に手を抜いて、うわべだけを最新技術でごまかす、という企業理念が、家電業界に限らず、あらゆる分野で幅を利かせているようだが、そんなことを続けていては、いつか破綻するときがくるのではないかと心配だ。小人の杞憂で済めばよいのだが。

リンクとは

前回、Linuxにおいて「ファイルを削除する」ことは「リンクを外すこと」と同じ意味であると述べた。では「リンク」とは何かを一言で(乱暴に)いうと、目に見えるファイル名のことである。もともとリンク(link)という単語は、「何かと何かを結びつけるもの」というような意味なのだが、Linuxというところのリンクでは、人間が認識できる世界と、ディスク中のファイルの実体(データ)を結びつけるもの、と考えておけば間違いないだろう。

前回説明したように、Linuxのファイルの実体は「iノード」という番号が

らたどることのできるディスク上のデータ領域である。しかしこれでは使いづらいので、「ディレクトリファイル」の中で人間にも読める形の「ファイル名」とiノード番号の対応付けを行って、実際にファイルを使うときにはファイル名を指定すればファイルの実体にたどり着けるようにしてあるわけだ。この「ファイル名とiノード番号の対応付け」こそが「リンク」の正体なのである。極端にいえば、すべてのファイル(名)はリンクであると断言してしまっても間違いではない。

「なんだ、それならわざわざ『リンク』なんてややこしい言葉を使わずに、『ファイル名』といえればいいんじゃないのか」と考えるかもしれないが、まったくそのとおりだ。だから、「リンク」について説明されている文章は、「ファイル名」と置き換えてしまっても間違いにはならないことが多い。

たとえば、前回説明したファイルを削除するコマンド「rm」は、ファイル自体の削除をするのではなく「アンリンク」すなわちリンクを外す処理をしているのだが、これは「ディレクトリファイル中のファイル名を削除する」と言い換えてもよい。実際には、ファイル名を削除すると同時に、そのファイル名と関連付けられている先のiノードに存在するリンクカウントという数値から、1を減算するという処理も行っている(図1)。

逆に、「新しくファイルを作成する」場合は、iノード領域の中から未使用のiノードを探し、そのiノード番号とファイル名の対をディレクトリファイルに追加するとともに、そのiノードのリンクカウントに1を加算する処理が行われる。この処理は新たなリンクを作成すると言い換えてもよいのだ。ここで、リンクカウントとは何

のためにあるのかと思うだろうが、それについてはこれ以降の説明を読めばわかるだろう。

さて、単純に「ファイル名」と言ったほうがわかりやすいのに、わざわざ「リンク」という言葉を使っているのは理由がある。Linuxのリンクでは、同一のiノード、すなわち同一のファイルの実体を指し示す、複数のリンクを作ることができるからだ。つまり、まったく同じ(内容が同一の複製という意味でなくて、実体そのものが同じ)ファイルに対して、複数のファイル名を付けることができるのだ(図2)。

たとえば、1つのファイルに「fileA」という名前と「fileB」という名前の2つのリンクがつけられているとすると、

```
$ cat fileA
```

と、

```
$ cat fileB
```

という2つのコマンドでは、まったく同じ内容が出力される。これだけでは、たとえば、「cp fileA fileB」を実行して同じ内容のファイルを2つ作ってある場合と変わりはない。ところが、fileAとfileBが同一ファイルへのリンクである場合なら、たとえば一方のfileAに対して、

```
$ echo "hoge" >> fileA
```

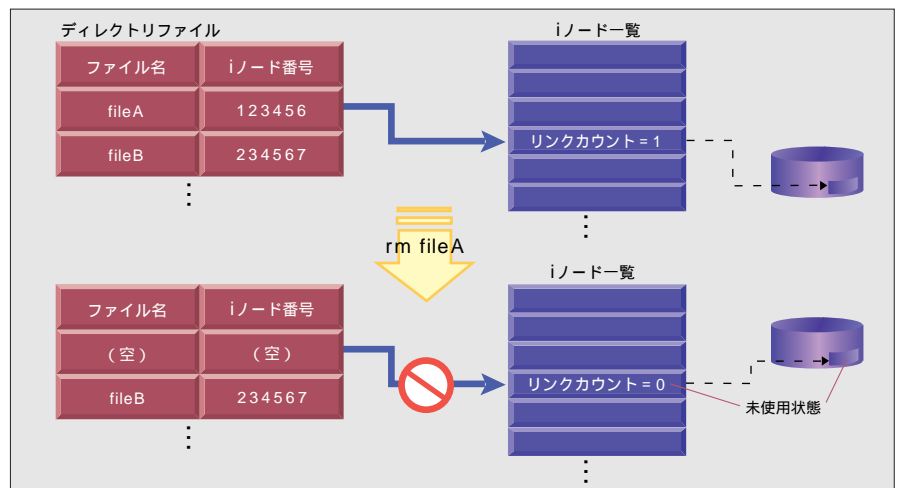
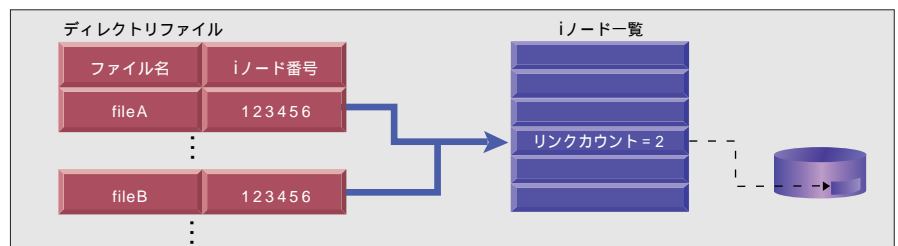


図1 rmコマンドの「アンリンク」処理

rmコマンドでファイルを「削除」するとき、実際にはディレクトリファイル中の該当ファイルの情報(リンク)を削除し、参照先のiノードの「リンクカウント」を1つ減らすという処理が行われている。未使用状態になったiノードとディスク領域は、いずれ再利用される(いつ再利用されるかを、ユーザーが知ることはできない)。



画面2 リンクカウントを調べる

ls -lコマンドの出力を見ると、ファイルのアクセス属性(「-rwxrwxr-x」などという文字列)の次に、数値が表示されているのに気づくだろうか。これが、そのファイル(正確にはそのファイルの実体を示すiノード)のリンクカウントの値である。この値が2より大きい場合は、そのファイルに対して複数のファイル名(リンク)が割り当てられていることを示す。

などを実行してファイルの内容を書き換えた後で、fileBの内容を表示させると、ちゃんと書き換えられた後の内容になっているはずだ(画面1)。

fileAとfileBは同じ実体(iノード)を指しているわけだが、このどちらかをrmコマンドで削除した場合どうなるだろうか。前回から何回か述べたように、rmコマンドは実際には「ファイルの削除」ではなく「リンクの解除」を行うコマンドである。たとえば、fileBを、

```
$ rm fileB
```

を実行して「削除」したとすると、実際には「fileB」というファイル名(リンク)が削除され、それまでfileBが指し示していたiノードの「リンクカウント」から1が減算される。しかし、iノード自体とそれが示している実際のファイルデータには変化はない。このため、fileAに関しては、fileBを「削除」した後も、ファイルの内容には以前と変わらずにアクセスできるのだ。ここでさらに、fileAについても「rm fileA」で削除すると、iノードのリンクカウントは1減って「0」になる。この時点で、初めてiノードが「未使用」の状態になり、いままでファイルのデー

タを記憶するために使われていたディスク上の領域が開放できるようになるのだ。

このように、iノードにある「リンクカウント」という数値は、そのiノード(すなわちファイルの実体)を指し示している「リンク」がいくつ存在するかを示していて、数値が0になれば、もはやそのファイルを指し示すリンクがないことを表す。ただし、リンクカウントが0になっても、すぐにファイル自体がなくなるわけではない。前回説明したように、そのファイルがまだ使用中である限り、iノードとそれが管理するデータ領域はそのまま存在し続ける。

リンクカウントの値は、lsコマンドで簡単に知ることができる。たとえば、

```
$ ls -l /bin | less
```

などで/binディレクトリの内容を表示してみると、アクセス権の次に数値が表示されることに気づくだろう(画面2)。ほとんどのファイルは「1」だが、たとえば「gzip」、「gunzip」、「zcat」などのコマンドファイルについては「3」などと表示されているはずだ。これがリンクカウントの数値で、そのファイル(リンク)が指している先のiノード

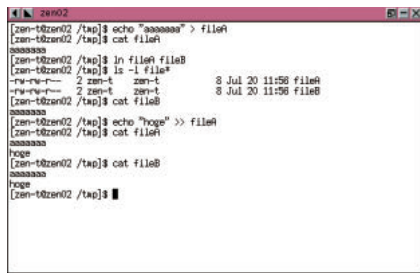
のリンクカウントの値なのである。

ついでによく見てみると、先ほどのgzip、gunzip、zcatの3つのファイルは、ファイルサイズもファイル更新日時もまったく同じことに気づくだろう。実は、これらの3つのファイルの「実体」は同じで、3つの異なる名前のリンクが作られているだけなのだ。これらのコマンドは、同じバイナリファイルで3種類の機能をこなすようになっているのだが、コマンドラインオプションで機能を指定するのは面倒なので、機能ごとに別々の「ファイル名」をつけて、簡単に使い分けられるようになっているのだ。Linuxでは、プログラムの実行時に、そのコマンドがどのファイル名を使って呼び出されたかを簡単に知ることができる。これとリンクを組み合わせれば、1つのコマンドファイルであたかも複数のコマンドファイルがあるかのように見せることができるのだ。

ハードリンクとシンボリックリンク

ところで、いままで説明してきた「リンク」は、ハードリンクと呼ぶことが多い。ハードリンクがあるのなら、その反対のソフトリンクもあるのでは、と考えるだろうが、そのとおりである。ただし、ソフトリンクという言葉はあまり使われず、シンボリックリンクと呼ぶのが普通だ。また、最近では、なんの前触れもなく「リンク」といった場合、ハードリンクではなくシンボリックリンクのことを指すことが多い。

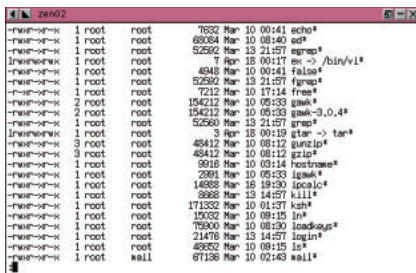
シンボリックリンクはハードリンクと同様に、あるファイルに対して別の名前を付けるために使われる。たとえば、先ほどの例と同じ/binディレクトリの中をls -lコマンドで見ると、



```
[zen@t0zen02 /tmp]$ echo "aaaaaa" > fileA
[zen@t0zen02 /tmp]$ cat fileA
aaaaaa
[zen@t0zen02 /tmp]$ ln fileA fileB
[zen@t0zen02 /tmp]$ ls -l file*
-rw-rw-r-- 2 zen-t zen-t 8 Jul 20 11:56 fileA
-rw-rw-r-- 2 zen-t zen-t 8 Jul 20 11:56 fileB
[zen@t0zen02 /tmp]$ cat fileB
aaaaaa
[zen@t0zen02 /tmp]$ echo "hoge" >> fileA
[zen@t0zen02 /tmp]$ cat fileA
hoge
[zen@t0zen02 /tmp]$ cat fileB
hoge
[zen@t0zen02 /tmp]$
```

画面1 リンクの実験

同じファイルの実体をリンクしているfileAとfileBは、どちらに対する読み書き操作も、結果的には同一の実体に対するアクセスになる。fileAに対して行った変更は、fileBで読み出したときにも反映されている。



```
[zen@t0zen02 ~]$ ls -l /bin | less
-rwxr-xr-x 1 root root 7832 Mar 10 00:141 echo*
-rwxr-xr-x 1 root root 89084 Mar 10 08:140 ed*
-rwxr-xr-x 1 root root 52592 Mar 13 21:157 ggrep*
-rwxr-xr-x 1 root root 7 Apr 18 00:117 ex -> /bin/vi*
-rwxr-xr-x 1 root root 4948 Mar 10 00:141 false*
-rwxr-xr-x 1 root root 52592 Mar 13 21:157 fgrep*
-rwxr-xr-x 1 root root 7212 Mar 10 17:114 free*
-rwxr-xr-x 1 root root 194212 Mar 10 09:333 gawk*
-rwxr-xr-x 1 root root 154212 Mar 10 09:333 gawk-3.0.4*
-rwxr-xr-x 1 root root 52560 Mar 13 21:157 grep*
-rwxr-xr-x 1 root root 3 Mar 18 00:139 star -> tar*
-rwxr-xr-x 1 root root 48412 Mar 10 08:112 tar*
-rwxr-xr-x 1 root root 48412 Mar 10 08:112 gzip*
-rwxr-xr-x 1 root root 9918 Mar 10 03:14 hostname*
-rwxr-xr-x 1 root root 2991 Mar 10 09:333 igawk*
-rwxr-xr-x 1 root root 14888 Mar 18 19:303 ipcalc*
-rwxr-xr-x 1 root root 9888 Mar 13 14:157 kill*
-rwxr-xr-x 1 root root 171332 Mar 10 01:157 ksh*
-rwxr-xr-x 1 root root 15032 Mar 10 08:115 ln*
-rwxr-xr-x 1 root root 75900 Mar 10 08:283 loadavg*
-rwxr-xr-x 1 root root 21476 Mar 13 14:157 lsjar*
-rwxr-xr-x 1 root root 43652 Mar 10 08:115 ls*
-rwxr-xr-x 1 root root 67136 Mar 10 02:143 mail*
```

画面2 リンクカウントを調べる

ls -lコマンドの出力を見ると、ファイルのアクセス属性(「-rwxrwxr-x」などという文字列)の次に、数値が表示されているの気づくだろうか。これが、そのファイル(正確にはそのファイルの実体を示すiノード)のリンクカウントの値である。この値が2より大きい場合は、そのファイルに対して複数のファイル名(リンク)が割り当てられていることを示す。

一覧の中に、

```
awk -> gawk*
```

```
bsh -> ash*
```

というように、「->」という記号の後ろに別のファイル名(ディレクトリ名)が追加されているようなファイル(ディレクトリ)がある。これがシンボリックリンクである。たとえば、「awk」というコマンドを実行すると、実際には「gawk」というコマンドファイルが、「bsh」というコマンドを実行すると、実際には「ash」というコマンドファイルがそれぞれ読み込まれて実行されるのである。

これだけなら、ハードリンクもシンボリックリンクもほとんど違いはない。しかし、シンボリックリンクは通常のファイルだけでなく、**ディレクトリに対するリンクも作成できる**。たとえば、/usrディレクトリの中を見ても、

```
dict -> share/dict/
```

```
doc -> share/doc/
```

```
tmp -> ../var/tmp/
```

といったようなシンボリックリンクがある(ディストリビューションによ

```

zeno2
[zen-10zen02 testdir]# pwd
/tmp/testdir
[zen-10zen02 testdir]# mkdir truedir
[zen-10zen02 testdir]# echo "aaaaa" > truedir/fileA
[zen-10zen02 testdir]# ls truedir
fileA
[zen-10zen02 testdir]# ln -s truedir symboli.cdir
[zen-10zen02 testdir]# ls
fileA
[zen-10zen02 testdir]# echo "cccc" > symboli.cdir/fileA
[zen-10zen02 testdir]# cat truedir/fileA
aaaaa
[zen-10zen02 testdir]#

```

画面3 ディレクトリのシンボリックリンク

ディレクトリに対するシンボリックリンクを「ls」で表示したり「cd」でそのディレクトリに移動すると、リンク先のディレクトリに対する操作が行われる。ただし、見かけ上はあなたもその位置にディレクトリが実存するかのように振舞うので、アプリケーションやユーザーからその違いを意識することは少ない。

て異なる)。これらはリンクが指し示す先が通常のファイルではなく、ディレクトリになっている。ためにこれらのシンボリックリンクに対して「cdコマンド」を使って移動してみると、リンク先のディレクトリに移動する(画面3)。このような、ディレクトリに対するリンクは、ハードリンクでは**原則として不可能**なのである。

ハードリンクの限界

ディレクトリに対するリンクのほかにも、ハードリンクが使えない場合がある。前回、mvコマンドによるファイルの移動について述べたとき、「ファイルのあるファイルシステムから別のファイルシステムに移動する場合、ファイル名の付け替えではなくコピーが行われる」という意味の説明をした。これと同じことがハードリンクの扱いについても言える。ハードリンクは、ファイル名とiノード番号の組み合わせなので、指し示すことのできるファイルは、そのiノード番号が有効なファイルシステムの内部に限られる。たとえば、/tmpと/mntという2つのディレクトリ

が別々のファイルシステム上に存在する場合(容量の関係でそれぞれのディレクトリを2台のハードディスクに分けて配置したときなど)、/tmp/fileAと/mnt/fileBが同一のファイルを示すようなハードリンクは、存在し得ないのである(図3)。

一方のシンボリックリンクについては、このような制限はない。ファイルシステムの場所や種類を問わず、cdコマンドで移動できる先ならどこにでもリンクを作成することができるのだ。

実は、大昔のUNIXには「ハードリンク」しか存在しなかった。ところが、UNIXシステムの規模が大きくなって、1台のマシンで複数のファイルシステムを使うようになってくると、上記のようなハードリンクの制限があると使いづらいということで、新たに「シンボリックリンク」の仕組みが導入されたのだ。

繰り返しになるが、最近では、ハードリンクよりもシンボリックリンクのほうがわかりやすいということから、一般に「リンク」といった場合、シンボリックリンクのことを指すことが多い。

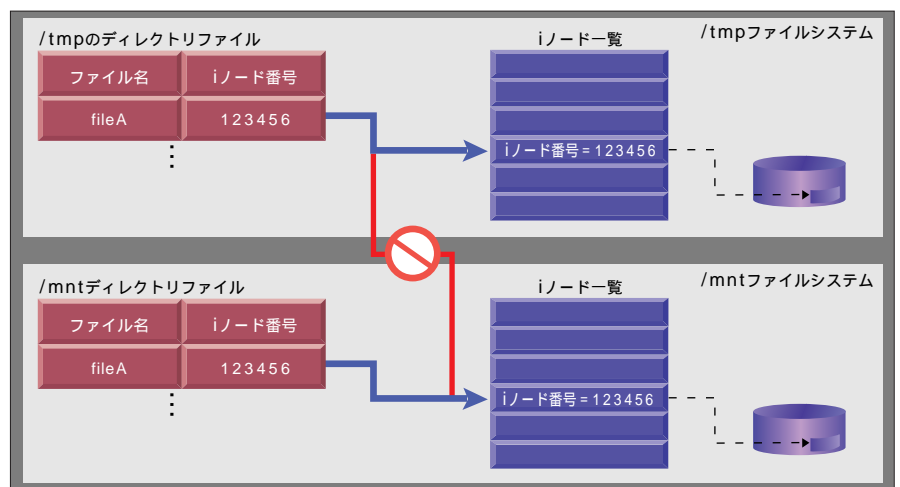


図3 ファイルシステムをまたがるハードリンク

ファイルの実体を管理するiノード番号は、各ファイルシステム(パーティション)ごとに独立して存在するため、別のファイルシステムからそのiノード番号を参照することはできない。このため、iノード番号を共有するような「ハードリンク」を、ファイルシステムをまたがって作成することはできないのだ。

ディレクトリとハードリンク

ところで、さきほど「すべてのファイルは(ハード)リンクである」と述べた。一方、Linuxではディレクトリも「ディレクトリファイル」という一種のファイルにすぎないことを前回述べている。すると、ディレクトリだってハードリンクであるはずだ。事実、そのとおりで、ディレクトリ名もディレクトリファイルの内容を示すiノードへのハードリンクなのである。ただし、ディレクトリへのハードリンクを、一般ユーザーが直接作成したり削除したりすることはできない。「ディレクトリへのハードリンクは原則として不可能」と述べたのはそのためである。実際には、mkdirコマンドやrmコマンドを使ってディレクトリを作成したり削除したりするとき、内部的にハードリンクの作成や削除が行われているのだ。

「ls -l」コマンドでディレクトリ一覧を表示させると、ディレクトリについても「リンクカウント」が表示されているのが分かるだろう。この数値をよく見ると、そのディレクトリの中に

含まれているサブディレクトリの数に2を足した値になっているのに気づくだろうか。これは、すべてのディレクトリは、自分自身を指す「.」というハードリンク(ファイル名)と、親ディレクトリを指す「..」というハードリンクを持っているからだ。

たとえば、Aというディレクトリの中に、B、C、Dという3つのサブディレクトリが存在する場合、Aディレクトリの内容を記したディレクトリファイル(iノード)へのハードリンクは、A自身の中の「.」、B、C、Dそれぞれの中にある「..」、そして、Aディレクトリの親ディレクトリの中にある「A」の、合計5つ存在することになる(図4)。「cd ..」と入力するときの「..」は、単なるおまじないだと思っていた人も多いだろうが(筆者もそうだった)これは実体のないおまじないではなく、「..」というファイル(リンク)が本当に存在しているのである。cdコマンドは「..」を見て、「親ディレクトリへ移動」という処理を行っているのではなく、もっと機械的に「..という名前のディレクトリに移動」しているだけなのである。このあたりの

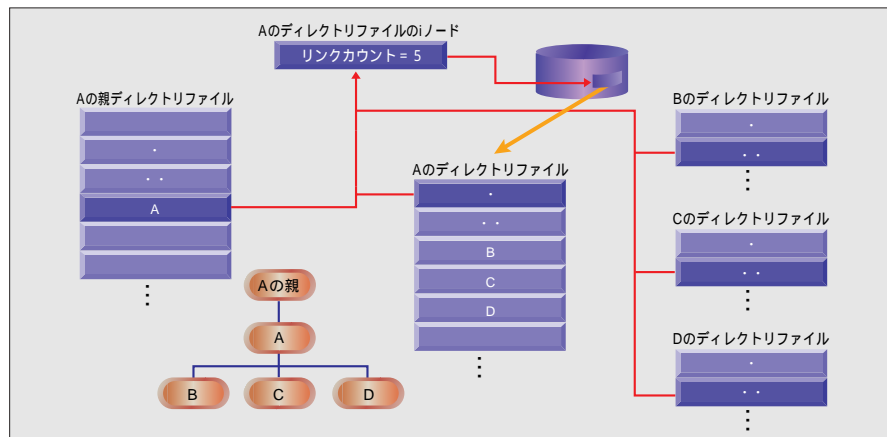
くわしい事情については、本誌2000年3月号と4月号の『Linux日記』を参照してもらいたい。

シンボリックリンクの正体

さて、話を「シンボリックリンク」に戻すことにしよう。われわれは、ファイルやディレクトリを作成したり削除したりするとき、無意識に「ハードリンク」の操作を行っていることになるわけだが、**ふだんの生活**でハードリンクを意識的に作成したり削除したりすることは少ない。ファイルシステムの違いなどを意識する必要があったり、ディレクトリへのリンクをユーザーが作成できなかったりするなど、制約が多いからだ。このため、通常はシンボリックリンクを使うことになる。

シンボリックリンクの正体は、**テキストファイル**である。中身には、リンク先のファイル名やパス名がそのまま記録されている。ただし、シンボリックリンクのファイルには、特別な印(属性)がつけられていて、一般のファイルとは区別されている。これは、ディレクトリファイルの正体がデータファイルなのに、他のファイルとは違う振る舞いをするのと同じである。ls -lコマンドでディレクトリの内容を表示するとき、シンボリックリンクは、そのアクセス属性を表す文字列の先頭に「l」(小文字のL)が表示される。同じ場所に、ディレクトリの場合は「d」が表示され、一般のファイルの場合は「-」が表示されている(画面4)。ちなみに、「ハードリンク」は一般のファイルと区別がつかない(というよりも、すべての一般ファイルはハードリンクである)ので、「-」のままである。

アプリケーションがシンボリックリンクへアクセスしようとする、シン



画面4 ls -lでリンクの種類を見分ける

通常のファイル、通常のディレクトリ、ファイルへのハードリンク、ディレクトリへのシンボリックリンク、ファイルへのシンボリックリンクをつくって、ls -lコマンドで見える。シンボリックリンクは、ファイルのアクセス属性の左端に「l」が、通常のディレクトリには「d」が表示されるが、ハードリンクについては、リンクカウントの値が多い以外、通常のファイルとの見分けはつかない。

ボリックリンクの中身が読み出されて、そこに記述されているファイルなりディレクトリなりへのアクセスに置き換えられる。たとえば、「/var/tmp」というディレクトリへのシンボリックリンクが/usr/tmpという名前が存在する場合(/usr/tmp -> /var/tmp)

```
$ cd /usr/tmp
```

を実行すると、見かけ上は/usr/tmpへの移動だが、実際には/var/tmpディレクトリへ移動することになる。

ところで、シンボリックリンクは、基本的にどのディレクトリやファイルに対しても自由に作成できるため、これが**セキュリティホール**となる可能性がある。このため、Linuxのプログラム（特にネットワークサービスに関連するもの）の中には、シンボリックリンクのリンク先への移動を禁止できるようになっているものが多い。たとえば、Webページの作成時にシンボリックリンクを使うと、うまく動かないことがあるが、これはWebサーバがそのような設定になっているからである。シンボリックリンクを使わないようにするか、あるいはWebサーバの設定を変えなければならない。

```
[zen@t0zer02 testdir]$ echo "aaaaaaaa" > fileA
[zen@t0zer02 testdir]$ mkdir truedir
[zen@t0zer02 testdir]$ ln fileA hardlink
[zen@t0zer02 testdir]$ ln -t truedir symbolicdir
[zen@t0zer02 testdir]$ ln -s fileB symbolicfile
[zen@t0zer02 testdir]$ ls -l
total 12
-rw-rw-r-- 2 zen-t zen-t 11 Jul 20 13:42 fileA
-rw-rw-r-- 2 zen-t zen-t 11 Jul 20 13:42 hardlink
lrwxrwxrwx 1 zen-t zen-t 7 Jul 20 13:43 symbolicdir -> truedir/
lrwxrwxrwx 1 zen-t zen-t 5 Jul 20 13:43 symbolicfile -> fileB
-rw-rw-r-- 2 zen-t zen-t 4096 Jul 20 13:42 truedir/
[zen@t0zer02 testdir]$
```

画面4 ls -lでリンクの種類を見分ける

通常のファイル、通常のディレクトリ、ファイルへのハードリンク、ディレクトリへのシンボリックリンク、ファイルへのシンボリックリンクをつくって、ls -lコマンドで見ている。シンボリックリンクは、ファイルのアクセス属性の左端に「l」が、通常のディレクトリには「d」が表示されるが、ハードリンクについては、リンクカウンターの値が多い以外、通常のファイルとの見分けはつかない。

リンクの作成と削除

ちょっとややこしい話が続いてしまったが、ディレクトリの構成が複雑になってしまい、長いディレクトリ名を入力するのが面倒な場合や、複数のディレクトリから同じファイルを参照できるようにしたい場合、あるいは、ディレクトリやファイルに「別名」をつけたいような場合には、リンクを作成すると便利である。

リンクの作成には、ハードリンク、シンボリックリンクのどちらの場合もlnコマンドを使う。/tmp/fileAというファイルに対するリンクを、カレントディレクトリのfileBという名前で作成したい場合、ハードリンクをつくるのであれば、

```
$ ln /tmp/fileA ./fileB
```

と入力する。同じようなリンクをシンボリックリンクとして作成したい場合には、-sオプションをつけて、

```
$ ln -s /tmp/fileA ./fileB
```

と入力する。ハードリンクを作成する

```
[zen@t0zer02 testdir]$ ls -l
total 4
-rw-rw-r-- 1 zen-t zen-t 11 Jul 20 13:42 fileA
lrwxrwxrwx 1 zen-t zen-t 5 Jul 20 13:43 symbolicfile -> fileA
[zen@t0zer02 testdir]$ rm fileA
[zen@t0zer02 testdir]$ ls -l
total 0
lrwxrwxrwx 1 zen-t zen-t 5 Jul 20 13:43 symbolicfile -> fileA
[zen@t0zer02 testdir]$ cat symbolicfile
cat: symbolicfile: No such file or directory
[zen@t0zer02 testdir]$ rm symbolicfile
[zen@t0zer02 testdir]$ ls
[zen@t0zer02 testdir]$
```

画面5 シンボリックリンクの「幽霊リンク」

シンボリックリンクのリンク先を削除してしまうと、リンクだけは残るのだが、アクセスしようとする「ファイルが見つからない」という意味のエラーが発生する。このような場合は、慌てず騒がず、リンクファイルを削除すればよい。

とき、リンク先とリンク元のファイルシステムが違ったり、ディレクトリへのリンクを作成しようとした場合には、

```
ln:cannot create hard link ...
```

というようなエラーメッセージが表示され、リンクは作成されない。このようなときは、-sオプションを追加して、シンボリックリンクとして作成し直せばよい。リンクの削除は、通常のファイルの削除と同じrmコマンドを使う。これは、ハードリンクもシンボリックリンクも同様である。

```
$ rm ./fileB
```

ところで、ハードリンクの場合は、リンク先とリンク元のどちらを先に削除しても、ファイル本体は最後のリンクがなくなるまでは存在し続けるので問題ないのだが、シンボリックリンクの場合、リンク先の大元のファイルを削除してしまうと、リンクだけが取り残されてしまう。リンクは残っているのでファイルがあるように見えるのだが、実際にそのリンクにアクセスしようすると、no such file or directoryという冷淡なエラーメッセージが表示されることになる（画面5）。このような状態でも、ファイルシステム自体には異常はないし、リンクファイルをrmコマンドで削除すれば元の状態に戻るので心配はいらない。しかし、このような**幽霊リンク**は、アプリケーションやLinuxディストリビューションのバージョンアップをすると発生することが多い。実害はないというものの、**気持ちの良いものではない**ので、見つけたらひとつひとつrmコマンドで退治したほうがよいだろう。

Javaプログラミング入門

Javaで3Dゲームを作ろう

この連載の副題は「Javaで3Dゲームを作ろう」です。前回までは2Dでの実装でしたが、今回からはいよいよ3Dの世界に突入です。3Dといっても、そんなに身構える必要はありません。簡単な変換式で3Dを表現できますので、がんばってください。まずは肩慣らしからいきましょう。

第3回 いよいよ3Dに挑戦!

文: おもてじゅんいち / かざぐるま
Text: Junichi Omote / Kazaguruma

前回の最後のアプレット、「反射するボール (ball2d.java)」を実行してみましたか。確かにボールが反射しながら動いているのですが、よく見ると背景の白い部分にグレーの帯が上下したり、赤いボールの上下が欠けたりしていませんでしょうか。

これは、アプレットの描画のタイミングとディスプレイの垂直同期のタイミングとが微妙にずれてくるために起こる現象です。決して見映えの良いものではありませんし、この状態を長時間見続けるのも不快なものです。さらに今後、壁やブロック、ラケットなどを全部表示することになると、もっとひどいちらつきが起こってしまいます。

このちらつき現象をおさえるために、「ダブルバッファリング」という手法を使います。これは、個々の描画を画面に対して直接行うのではなく、いったんオフスクリーンバッファと呼ばれる、仮想の画面に描画しておき、全部を描き終えてから1画面分の絵をまとめて実際の画面に転送するという方法です。こうすれば、いちいち画面に描画する

よりも速く1画面分を描画することができるため、ディスプレイの垂直同期のタイミングにひっかからないで済むようになるのです。

仮想画面となるバッファ用としてImageクラスを使いますが、paintメソッドのパラメータや個々の描画のためにGraphicsクラスが必要になるため、そのImageクラスに結び付けられているGraphicsクラスを取得します。

リスト1は、前回のリスト3のダブルバッファリング版です。

網掛けの部分でダブルバッファリングのために追加、修正する部分で、その他は前回のリスト3と同じです。動きを比較できるように、アプレット名を変えていますので、リスト1を作るために、前回のリスト3とリスト4をコピーして、新しい~.javaファイルとHTMLファイルを作り、リスト1のように修正してください (code = も変更するのを忘れないように)。

最初に、バッファのためのImageクラスとGraphicsクラスを扱うための変数を宣言しています。

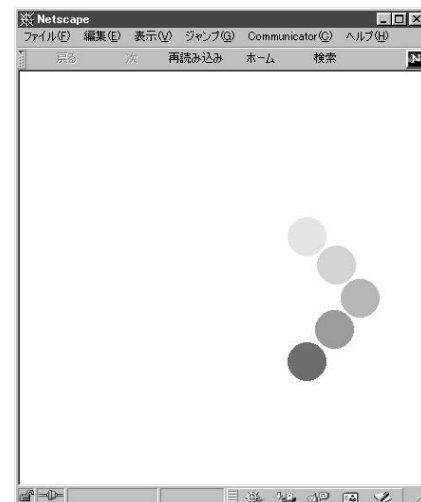
initメソッドでは、バッファのための領域を作成しています。

バッファのための領域は、

```
createImage(400,400)
```

として、400×400画素の実画面と同じ大きさの領域を確保します。

前回のリスト3にはなかったupdateメソッドがありますが、リスト1では、実画面に直接描画を行っていないため



画面1 反射するボールアプレット

に、システム側がアプレットのウィンドウを書きなおしする場合には、アプレットでも必ず描画するようにpaintメソッドを呼んでいます。

paintメソッドでは、今まで実画面に直接描画していたので、描画対象となるGraphicsメソッドはgであったのが、

ここではまずバッファのbuffergに描画してから、それをまるごとgに転送するという方法をとっています。そのために、バッファのImageクラスであるbufferから、getGraphicsメソッドによって、

```
bufferg = buffer.getGraphics()
```

というように、バッファのGraphicsクラスbuffergを取得して、setColorやfillRectなどの描画メソッドと、ball.disp(...)に送るパラメータとしてbuffergを使っています。

これにより、描画は実画面でなく仮想画面であるバッファに対して行われることとなります。

最後の、

```
g.drawImage(buffer, 0, 0, this)
```

は、実画面gのメソッドであり、仮想画面bufferの内容を実画面gの座標(0,0)に転送します。

以上がダブルバッファリングのための修正部分です。注目してほしいのは、ダブルバッファリングを使うようになっても、Ballクラスの部分には何の修正も加えていないということです。つまりBallクラスは、独立したクラスになっており、描画の対象が変わってもボールの動き自体は変わらないので、まったく同じクラスが使えるのです。

このようにJavaのプログラミングでは、個々のクラスの独立性を高めることができます。描画は描画というように、個々のオブジェクト(部品)の動きはそれぞれにまとめてクラス化することを心がけてください。

複雑だけど簡単な 3Dグラフィック

さて、それではいよいよこのボールを3次元の世界で動かしてみましょう。

3Dゲームを作る場合、当然ですが3Dグラフィックの知識が必要になります。画面の中で繰り広げられる世界はまさに現実の立体(3次元)世界であり、

リスト1 ダブルバッファリング版(ball2d2.java)

```
import java.applet.*;
import java.awt.*;

//-----
// Applet Class
//-----
public class ball2d2 extends Applet implements Runnable {
    Thread        t;
    Ball          ball;
    Image         buffer;           // バッファ用
    Graphics      bufferg;        // バッファ用

    public void init() {
        buffer = createImage(400,400); //バッファ領域を作る

        ball = new Ball(400,400);

        t = new Thread(this);
        t.start();
    }

    public void run() {
        try {
            while(true) {
                ball.process();
                repaint();
                Thread.sleep(10);
            }
        } catch(Exception e) {}
    }

    public void update(Graphics g) {
        paint(g);
    }

    public void paint(Graphics g) {
        // バッファのGraphics クラスを取得する
        if(bufferg == null) bufferg = buffer.getGraphics();

        bufferg.setColor(Color.white); // 背景をクリア
        bufferg.fillRect(0, 0, 400, 400);

        ball.disp(bufferg);           // ボール描画
        g.drawImage(buffer, 0, 0, this); //バッファを画面に転送
    }
}

//-----
// Ball Class
//-----
class Ball{
    int        vx,vy;
    int        gx,gy,gr;
    .
    省略
    .
}
```

それを平面（2次元）の画面に写さなければならないのですから、それなりのテクニックが必要になってきます。

といっても、3次元の動きや位置を2次元画面の座標に置き換えるための変換式などは意外と簡単で、プログラムそのものは悩むほど複雑なものにはなりません。ややこしいのはその理屈であって、ややこしいからといって理屈を理解せずに変換式だけを使おうと思っても、結局思い通りの画面を描けなくなるのがオチです。ぜひここで、3Dゲームの基礎として重要な3Dグラフィックの基本をマスターしてください。ちょっと読んだだけではすんなり理解できないかもしれませんが、頭にイメージを描きながら何度も読み返していただければ、きっとものにすることができるようです。

座標変換 3次元から2次元へ

「座標」とは堅苦しい呼び方ですが、グラフィックスとはいってもプログラムで扱うのはとにかく座標に尽きます。2Dグラフィックスなら、ゲーム中の登

場人物の動きや位置も、XとYの座標で表し、現実のディスプレイの表示もXとYの座標ですから、ちょっとした拡大や縮小をすれば問題なく画面に表示できました。

3Dグラフィックスでも座標で表すことには変わりないのですが、厄介なことに3Dの座標はx、y、zの3つで表されているのです。といっても、ディスプレイの画面は依然としてX、Y座標ですから、x、y、zの3つを何とかしてX、Yに変換しなければならないというわけです。だから座標変換なのです。

最初に答えを書いておくと、3D 2Dの座標変換の式は、3Dの座標を小文字のx、y、z、2Dの座標を大文字のX、Yとすると、

$$X = hx / z$$

$$Y = hy / z$$

となります。嘘じゃありません、これだけです。

さんざん難しそうなことをいっておきながら、こんな単純な式だったのかとお叱りを受けそうですが、じつは単

純だから難しいのです。実世界の3次元を画面上の2次元に完璧に変換するという、「これぞコンピュータグラフィックスともいえる処理がたった2つの式でできるのか」という疑問が湧いてきたかもしれませんが、この答えを説明するとなると非常にややこしいのです。

3次元の物体を画面に表示するというのは、**図1**のような位置関係になります。つまり表示したい物体の影を、その物体と（画面を見ている人の）視線との間に置いたディスプレイ画面に投影するということなのです。もちろん実際には黒い影でなく、鏡に写っているような姿を投影します。

次に、**図1**の物体（対象物）を左右に動かしてみたらどう見えるでしょう。物体が左右に動けば、それに合わせて画面上の影も左右に動くはずですが、ですから、画面上の動きは物体の動きに比例します。つまり、物体のx座標が変化したらそれに比例して画面上の影のX座標が変化します。しかも、物体が左右に動いても画面上の影は上下には動きません。x座標の変化はY座標には関係しないのです。すなわち、変換

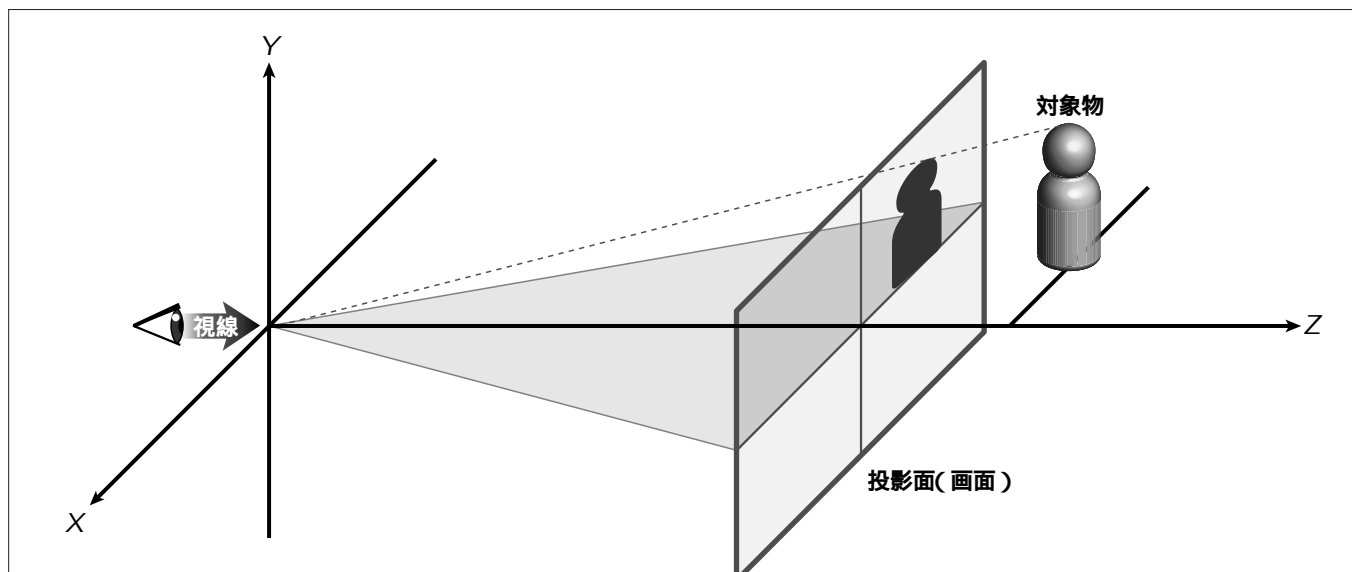


図1 3Dを画面の投影

式ではXの式にyはありませんし、Yの式にxは入ってこないのです。

さて、物体が画面から離れていったらどうなるでしょう。図2を見ると、物体の大きさは変化していなくても、物体が画面から離れて行くと画面上の影(Y)は小さくなっていきます。

つまり 物体のz座標が大きくなると、今度は画面の影のY座標が小さくなるのです。これは、Yはzに反比例するということです。X座標についても同様です。

すなわち、「画面上の座標Xは物体の座標xに比例し、zに反比例する。画面

上の座標Yは物体の座標yに比例し、zに反比例する」ということです。このことを表したのが、先ほどの2つの変換式というわけです。

これで、3D座標は立派に2D座標に変換されるのですが、この式にあるhを適当に変えることで、画面の見え方を変えることができます。図2にあるように、hは視点と画面との距離を表す係数で(実際の距離ではない)、大きければ視野が狭まり、小さければ視野が広がります。

図3、図4はそのサンプルで、どちらも左右に四本の柱が見えています。hが小さいと視野は広がるのですが、やりすぎると中心に集まりすぎて見にくくなります。用途に合わせて調整するのがいいでしょう。

3Dではずむボール

堅苦しい話はこれくらいにして、先ほどのリスト1の「反射するボール」アプレットを、さっそく3D化することにしましょう。

3Dの動きとなれば、上下左右のほか

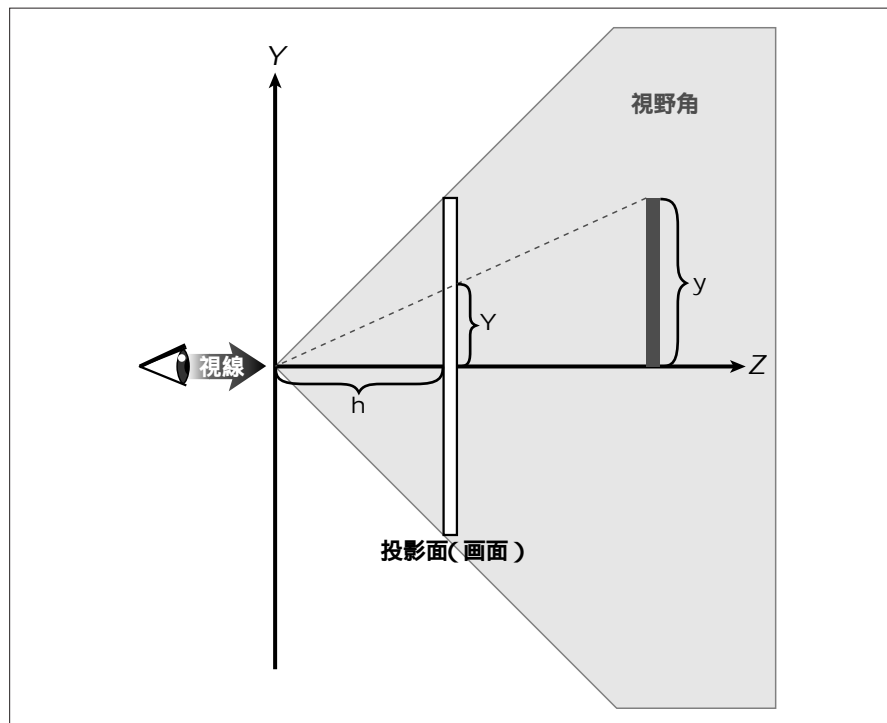


図2 画面上の座標

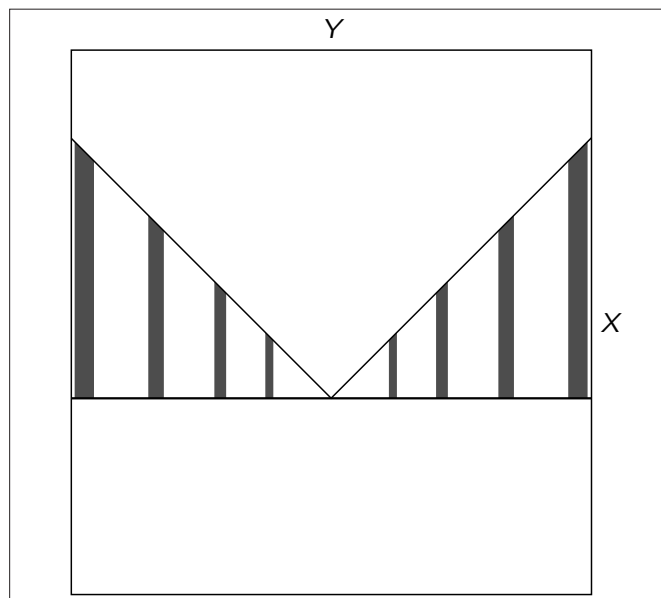


図3 hが大きい場合(視野が狭い)

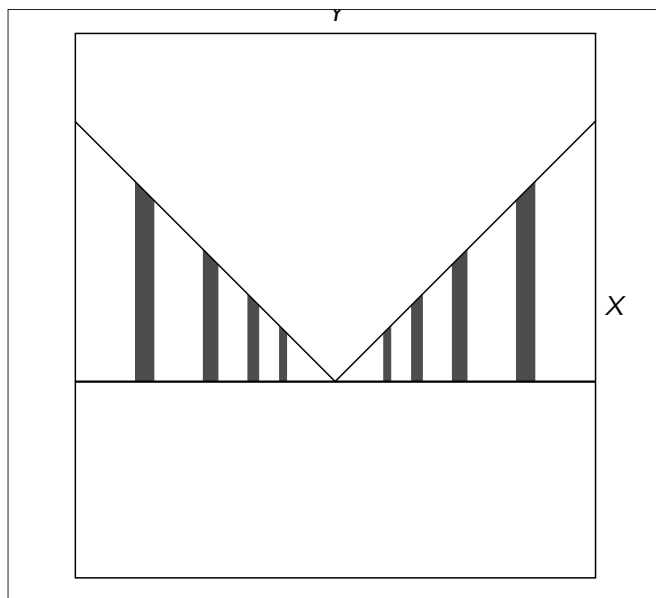


図4 hが小さい場合(視野が広い)

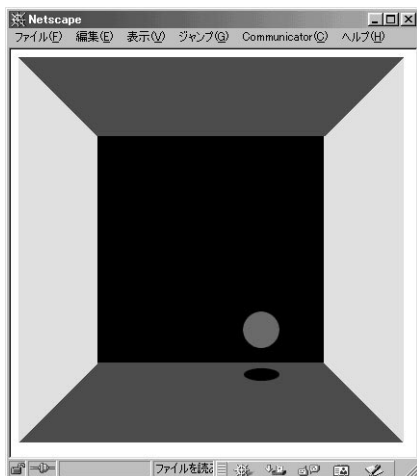
に奥行きを持たせる必要が出てきます。今までと同じような真っ白な背景では、奥行き方向の動きがわかりにくいので、壁と床、天井からなる背景も一緒に描くことにします(画面2)。

リスト2が「3Dではずむボール」のプログラムです。少し長いリストですが、全体の構造を理解してもらうために全部掲載します。ぜひとも、本文の解説を読みながら、実際にプログラム上でどのように書かれているかを十分に理解するようにしてください。単に掲載のリストを入力して実行するだけでは、いざ自分なりにプログラムを改良したり、新しいアプレットを作りたいときに何をどのように書けばよいのかがわからなくなってしまいます。

プログラミング上達の一番の早道は、入門書よりもサンプルプログラムを理解し、参考にすることです。

共通する処理をまとめて書く

前にも説明しましたが、Javaのプログラムはすべてがクラスで構成されています。リスト2も結構長いプログラムですが、よくみると4つのクラス(だけ)で成り立っています。



画面2 3Dではずむボール

```
class ball3d ....
class Disp3D
class Ball ....
class Background ....
```

そもそも本連載で作り上げようとしている「3Dブロックくずしゲーム」を構成する部品は、

- ・ボール
- ・ブロック

- ・ラケット
- ・壁(床、天井)

ですから、これらをそれぞれクラスとして作ってあげればよいのです。

「それならさっそく」とばかりに、各クラスを設計してプログラミングするのですが、ちょっとここで考えてみてください。

これら4つのクラスは、いずれも3D画面上を動き回る(壁は動かないが...

リスト2-1 ball3d.java (Applet Class)

```
import java.applet.*;
import java.awt.*;

//-----
// Applet Class
//-----
public class ball3d extends Applet implements Runnable {

    Thread          t;
    Ball            ball;
    Background      bg;
    Image           buffer;
    Graphics        bufferg;

    public void init() {
        buffer = createImage(400,400);

        ball = new Ball(400,400);
        bg = new Background(400,400);

        t = new Thread(this);           // スレッドを開始する
        t.start();
    }

    public void run() {
        try {
            while(true) {
                ball.process();
                repaint();
                Thread.sleep(40);
            }
        } catch(Exception e) {}
    }

    public void update(Graphics g) {
        paint(g);
    }

    public void paint(Graphics g) {
        if(bufferg == null)    bufferg = buffer.getGraphics();

        bg.disp(bufferg);      // 背景描画
        ball.disp(bufferg);    // ボール描画

        g.drawImage(buffer, 0, 0, this);
    }
}
```

...) ものです。3Dとして扱われるからには、前述の座標変換などの処理が必要になってくるでしょう。また、この4つのクラスにおいて、3Dの処理部分はほとんど同じだと考えられるので、その共通する処理部分を4つのクラスそれぞれについて書くのではなく、共通部分をまとめて書くことができれば楽ですし、あとで修正するときも1カ所を直すだけで済みます。

実はそれを「クラスの継承」という方法で簡単に実現することができます。

本連載の第1回目で、アプレットを

初めて作ったときに、「アプレットはアプレットとしての基本的な機能や性質を持っていなければなりません。そのようなアプレットとしての共通機能は、すでにJavaの中に基本クラスとして用意されています」ということを学んだのを覚えているでしょうか、そしてその基本クラスを引き継いで、自分なりに拡張したクラスを作るときには、

```
class ..... extends Applet
```

というように、extendsを使ってクラ

ス名を書きました。

この方法をクラスの継承といい、新しく作られるクラスはextendsの後に書かれた、元になるクラス（スーパークラス）の機能（メソッド）をすべて持つようになります。

ですから、今回のように3D処理をしなければならないクラスが4つもあるのなら、まず共通する3D処理部分だけの機能を持たせたクラスを作り、ボールやラケットなどのクラスは、すべてその3D処理用クラスをスーパークラスとして継承すればよいこととなります。つまり、

```
class Disp3D
```

という3D処理クラスを作り、

```
class Ball extends Disp3D
```

と、Disp3Dクラスを継承してBallクラスを作れば、BallクラスにはDisp3Dクラスの持つ機能すべてが搭載されることとなります。

では、リスト2で具体的な例を見ていきましょう。ここでは、本ゲームの部品のうち、ボールと壁（床、天井）のみを搭載しています。そして、ボールにも壁にも必要になる3D処理（座標変換）の機能を持つDisp3Dというクラスを作り、BallクラスとBackgroundクラスは、Disp3Dクラスを継承しています。

各クラスの宣言部分だけに注目すると、

```
class Disp3D {
    ....
}
class Ball extends Disp3D {
    ....
```

リスト2-2 ball3d.java (3D Class)

```
//-----
// 3D Class
//-----
class Disp3D {
    int          width,height;
    int          unit3d = 1000;

    Disp3D(int w, int h) {
        width = w;
        height = h;
    }

    // 3D->2D x座標変換
    int d2x(int x, int y, int z) {
        double dx;

        dx = (double)x * 500 / (z+350);
        dx = dx * width / unit3d + width / 2;

        return (int)dx;
    }

    // 3D->2D y座標変換
    int d2y(int x, int y, int z) {
        double dy;

        dy = (double)y * 500 / (z+350);
        dy = dy * height / unit3d + height / 2;

        return (int)dy;
    }

    // 3D->2D 相対座標変換
    int d2r(int r, int z) {
        double dr;

        dr = (double)r * 500 / (z+350);
        dr = dr * width / unit3d;

        return (int)dr;
    }
}
```

```

}
class Background extends Disp3D
{
....
}

```

となっており、BallクラスとBackgroundクラスが、ともにDisp3Dクラスの機能を持つことがわかるでしょう(図5)。

さて、このようにわざわざスーパークラスとして継承したDisp3Dクラスとはどういうものなのでしょうか。Disp3Dには、初期化のためのコンストラクタのほかに、

```

int d2x(int x, int y, int z) {}
int d2y(int x, int y, int z) {}
int d2r(int r, int z) {}

```

という3つのメソッドがあります。

これらは、(x,y,z)の3D座標をそれぞれ、2DのX座標、Y座標、相対座標に変換するためのメソッドで、まさにこれらのメソッドが、先程学習した3D 2Dのための座標変換を行うのです。

BallクラスとBackgroundクラスは、Disp3Dクラスを継承しているので、これらのメソッドを自分のものとして使うことができます。ですから、クラス内ではつねに3D座標を取り扱っており、画面に表示する場合のみこれらの座標変換メソッドを通して実画面用の2D座標を計算すればよいこととなります。BallクラスとBackgroundクラスの内部をじっくり見てみましょう。リスト1と比べるとよくわかりますが、リスト2のBallクラスでは、x,y,zというように3D座標で位置や速度が扱われています。

Disp3Dクラスは、

リスト2-3 ball3d.java (Ball Class)

```

//-----
// Ball Class
//-----
class Ball extends Disp3D {
    int          vx,vy,vz;
    int          x,y,z;           // 3D座標
    int          gx,gy,gr;       // 2D座標、直径
    int          sgx,sgy,sgr;    // 影2D座標

    Ball(int w, int h) {
        super(w,h);
        init();
    }

    void init() {
        x = 0;
        y = 0;
        z = 0;

        vx = 13;
        vy = 15;
        vz = 15;

        gr = 40;
        process();
    }

    void process() {
        x += vx;
        y += vy;
        z += vz;

        int lim = 500 - gr / 2;

        if ( x > lim )          vx = -vx;
        if ( x < -lim )         vx = -vx;
        if ( y > lim )          vy = -vy;
        if ( y < -lim )         vy = -vy;

        if ( z > lim )          vz = -vz;           // ブロック面
        if ( z < 150 ) {       vz = -vz;           z = 150; } // 手前の面

        gx = d2x(x,y,z);
        gy = d2y(x,y,z);
        gr = d2r(100,z);

        sgx = gx;
        sgy = d2y(x,500,z);
        sgr = gr / 2;
    }

    //---- 表示
    public void disp(Graphics g) {
        g.setColor(Color.black);
        g.fillOval(sgx-gr/2, sgy-sgr/2, gr, sgr);

        g.setColor(Color.red);
        g.fillOval(gx-gr/2, gy-gr/2, gr, gr);
    }
}

```

- 3D 2Dの座標変換
- 仮想座標(0~1000)を画面の実座標に変換

という2つの機能を一手に引き受けているため、それを継承するBallクラスとBackgroundクラスなどでは、この2点に関してはなにも考えなくても済むのです。

オブジェクト指向プログラミングでは、これを「隠蔽」と呼んでいます。つまり、スーパークラスが、ある機能を完全に引き受けて、外からそのからくりが見えないようにするのです。クラスを継承したクラスからは、そのからくりをいちいち気にせずに、心おきなくその機能を使うことができるというものです。

隠蔽には、継承していったクラス内でいちいちそのからくりについて考えなくてもよいという利便性だけでなく、そのからくりに変更を加えたいときでも、隠蔽しているスーパークラスだけを変更すれば、その他のクラスには一切手を加えなくてもよいというメリットがあります。

同じようにクラスを作っていくので、このように機能の共通化を図って、うまく継承していくようにすれば、クラスを使ったプログラミングのメリットがわかるようになります。

単に、「なんでもクラスにしてしまえ」というのではなく、できるだけそのプログラムの目的と構造に従ったクラス構成を考えていくことが重要になることでしょう。

リスト2の各部の詳細については次回解説しますので、それまではリスト2を実行しながら、数値をあれこれ変えてみたり、クラスの構成をじっくりと理解すべく復習してみてください。

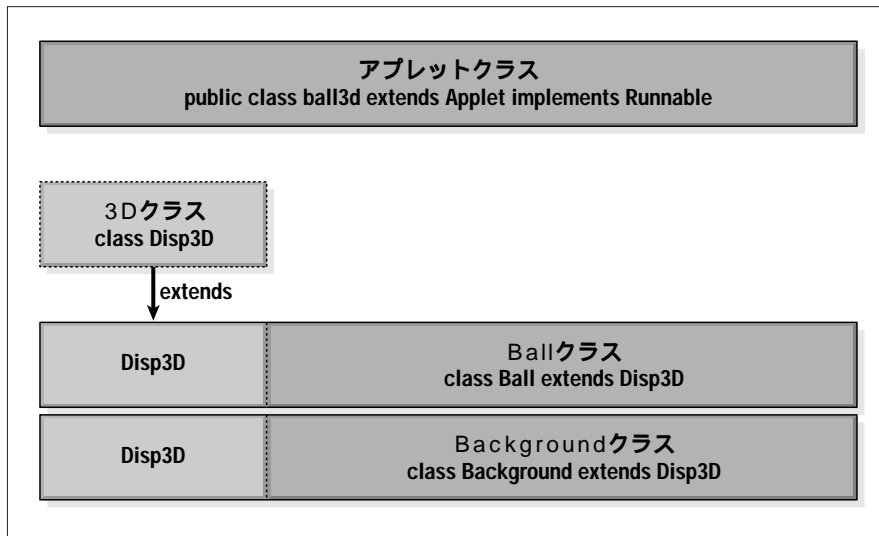


図5 クラス構成 (ball3d.java)

リスト3-4 ball3d.java (Background Class)

```

//-----
// Background Class
//-----
class Background extends Disp3D {

    Background(int w, int h) {
        super(w,h);
    }

    public void disp(Graphics g) {
        int ax[] = new int[4];
        int ay[] = new int[4];

        g.setColor(Color.blue);
        g.fillRect(0, 0, width, height);

        g.setColor(Color.cyan);
        ax[0] = d2x(-500,-500,0);
        ay[0] = d2y(-500,-500,0);
        ax[1] = d2x(-500,-500,500);
        ay[1] = d2y(-500,-500,500);
        ax[2] = d2x(-500, 500,500);
        ay[2] = d2y(-500, 500,500);
        ax[3] = d2x(-500, 500,0);
        ay[3] = d2y(-500, 500,0);
        g.fillPolygon(ax,ay,4);

        ax[0] = d2x( 500,-500,0);
        ay[0] = d2y( 500,-500,0);
        ax[1] = d2x( 500,-500,500);
        ay[1] = d2y( 500,-500,500);
        ax[2] = d2x( 500, 500,500);
        ay[2] = d2y( 500, 500,500);
        ax[3] = d2x( 500, 500,0);
        ay[3] = d2y( 500, 500,0);
        g.fillPolygon(ax,ay,4);

        g.setColor(Color.black);
        ax[0] = d2x(-500,-500,500);
        ay[0] = d2y(-500,-500,500);
        ax[1] = d2x( 500,-500,500);
        ay[1] = d2y( 500,-500,500);
        ax[2] = d2x( 500, 500,500);
        ay[2] = d2y( 500, 500,500);
        ax[3] = d2x(-500, 500,500);
        ay[3] = d2y(-500, 500,500);
        g.fillPolygon(ax,ay,4);
    }
}
  
```


プログラミング工房

ソフトウェアによってハードウェア（機械）をコントロールするのは楽しいものだ。いまは夏真っ盛り。そこで、今回はこれまでの連載とはちょっと路線を変更して、エアコンのリモコンを改造し、Linuxからエアコンのオン・オフを行うためのプログラムを作成してみることにしよう。

第9回 ハードウェアのコントロール

文：藤沢敏喜

Text: Toshiki Fujisawa

暑い。蝉の声をうるさく感じる夏本番である。この季節、外から帰ってきてムツとする部屋に入るのには本当に嫌なものだ。また、暑くて寝苦しい夜は、エアコンを快適な温度にコントロールできる部屋で眠りたい。そこで、今月は「夏休みの工作」としてPCからエアコンを操作するプログラムを作成してみたい。ハードウェアのアクセスを行うプログラミングは難しいのでは、と思う方が多いかもしれないが、そのようなプログラムを組むのが初めての人でも理解でき、実際にやってみることができる簡単なプログラム例から実験してみることにしよう。

なにはともあれ、まずは実験

プログラミングをして一番楽しいのは、ハードウェアをアクセスするプログラムを書いているときではないかと筆者は思っている。自分の書いたプログラムによって、機械が命令どおりに動いたときの気持ちは最高である。

しかし、ハードウェアのアクセス実験となると、半田ごてやテスター、そしてオシロスコープなどが必要になる場合もあり、その用意を考えるだけでめげてしまう。そこで、ここではそのような道具を準備しなくても、I/Oポート入力の実験ができる方法を考えてみた。用意するものは、ゼムクリップとパラレルプリンタポートがついているPC/AT互換機だけである。最初にリスト1のプログラムを、Linux上で、



```
$ gcc -O i-test.c
```

というコマンドでコンパイルして、「a.out」を作成する。なお、I/O命令を使用するために、「-O」オプションの指定が必須であることに注意する必要がある。

そして、ルートユーザーになってから、

```
# ./a.out      ( 付録CD-ROMの場合「make 1」でもよい)
```

として、このプログラムを実行すると、画面には「7f」という文字が1秒ごとに表示されるはずである。

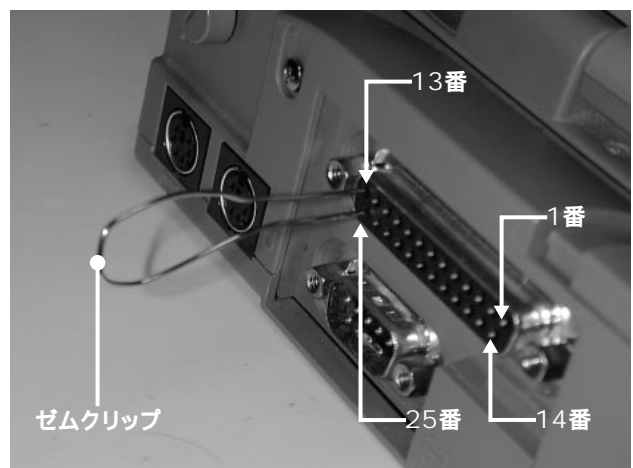


写真1 ゼムクリップを曲げて、13番ピンと25番ピンを直結する

次に、ゼムクリップを写真1のように曲げて、プリンタポートの左端にある上下のピン（13番ピンと25番ピン）に差し込むと表示が「6f」に変化し、クリップをはずすと「7f」が表示に戻るはずだ。

なお、プリンタのコネクタは写真にあるように、上のほうが広がっているように取り付けられていることがほとんどであるが、上下逆になっている場合もあるかもしれない。差し込む場合にはピン番号（小さく刻印してある）をよく確認してからクリップを差し込もう。

うまくいかない場合は、BIOSセットアップのプリンタの設定を見直してみよう。最近のPCでは、パラレルポートの設定が「EPP」や「ECP」などになっているものもあるが、このプログラムを動作させるためには、いわゆる「標準モード」（Normal）に設定する必要がある。また、BIOSセットアップでI/Oアドレスを設定できるPCも多いが、その場合は、アドレスとして0x378を選択する。これらの設定を見直しても、うまく動作しない場合は、プリンタの設定ガイドなどのドキュメントを参照し、プリンタから正しく印字できるかどうかを調べてみるとよいだろう。

ハードウェアを扱ううえでの注意

ここではゼムクリップを使用したけど、無理に奥まで差し込んだりすると、プリンタのコネクタ部分を破壊する恐れがある。大事なPCを壊したくない場合は、ゼムクリップの代わりに柔らかい電線を使うとよいだろう。また、プリンタ延長ケーブルを使って、そのケーブルの先にゼムクリップを差せば、より安全である。

さらに、静電気でプリンタポートが破壊されることもあるので、体から静電気を十分に放電し、ゼムクリップを絶縁物で巻いて直接触らないように注意する必要がある。

なんだか、製品の取り扱い説明書にある注意書きのようになってしまったが、実際のところは静電気防止のための対策がそれなりにしてあるはずだ。筆者の経験によれば、出力回路がショートするような過激なことをしなければ、そう簡単に壊れるわけではないので、あまり神経質になることもないかもしれない。

しかしながら、どんなへまをやっても壊れることがない

リスト1 プリンタポートのデータを読むサンプルプログラム

```

/* === list1.txt ===== */
#include <stdio.h>
#include <unistd.h>
#define IO_ADR 0x378

/*-----*/
#ifdef __FreeBSD__
#include <fcntl.h>
#include <machine/cpufunc.h>
#define IO_ENABLE() (open("/dev/io", O_RDWR, 0777) == -1)
#endif
/*-----*/
#ifdef linux
#include <sys/io.h>
#define IO_ENABLE() ioperm(IO_ADR, 3, 1)
#endif
/*-----*/
int main(void)
{
    if (IO_ENABLE()) {
        printf("Can't enable io\n");
        return 1;
    }
    for(;;) {
        printf("%02x\n", inb(IO_ADR+1));
        sleep(1);
    }
}
/*-----*/

```

プリンタポートのI/Oアドレス

I/Oアクセスできるようにする

2桁の16進数で表示

プリンタポートを読む

1秒間待つ

ソフトウェアとは違って、ハードウェアを扱う場合には、それなりの配慮が必要であることは確かである。

CPUからのI/Oアクセス

PCには、プリンタポートやシリアルポートなどさまざまなI/Oポートが存在しているので、CPUからそれらのポートを扱うためには、どのI/Oポートをアクセスすべきかを識別する必要がある。

この識別をするために用いるのがI/Oポートを示す番号であるが、この番号の付け方はコンピュータの構成により2種類ある。ひとつは、I/Oポートのレジスタをメモリ空間の一部として割り当てる方法である。つまり、CPUはメモリにアクセスするのと同じ命令を用いて、I/Oポートのレジスタを読み書きすることになる（メモリマップドI/O）。もうひとつは、I/Oポートのレジスタをメモリ空間とは別のI/O空間として割り当てる方法（I/OマップドI/O）で、IntelのPentiumプロセッサを使ったシステムが、この方式でI/Oをアクセスしている。

この方式は、CPUにI/Oアクセスをする特別な命令が必要になり、その命令を使ってI/O空間をアクセスしなければならない。Pentiumプロセッサの場合、バイト単位でI/Oにアクセスする命令として、inb(in byte)とoutb(out byte)が用意されている。アセンブラ（GNUのas）で、

```
inb %dx, %al
```

と記述すると、DXレジスタ（16ビット）に入っているI/Oアドレスで示されるI/Oレジスタのデータを、AL（8ビット）レジスタに得ることができる。また、

```
outb %al, %dx
```

と記述すれば、ALレジスタに入っている8ビットのデータをDXレジスタで示すI/Oアドレスに出力できる。

C言語からのI/Oアクセス

前述のように、I/Oにアクセスするためには、in命令やout命令を使用するが、C言語からアクセスする場合は、何らかの形でこれらの命令を使えるように、IN関数、OUT関数をアセンブラで記述するか、コンパイラに用意されているマクロなどを用いる必要がある。その際には、OSやコンパイラによって引数や使い方が違うので注意が必要である。たとえば、I/Oアドレスが0x379番地（“0x”は16進数であることを示す）のポートのデータを読み出したり、0x378番地に0x5Aというデータを書くためには、OSによって次のような記述の違いがある。

1. DOSの場合

```
data = inportb(0x378);
outportb( 0x378, 0x5A );
```

2. FreeBSDの場合

```
data = inb(0x378);
outb( 0x378, 0x5A );
```

3. Linux (RedHat 6.2J) の場合

```
data = inb(0x378);
outb( 0x5A, 0x378 );
```

上記のように、FreeBSDやDOSとは違って、Linuxではoutbの引数の順番が逆である。FreeBSDやLinuxなどのUNIX系OSでは、多くのプログラムはあまり変更なく移植可能なことが多いが、ハードウェアアクセスの部分については、OSによってかなり異なる。

Column

LinuxのI/Oポート プログラミングミニハウツー

LinuxでI/Oポートのプログラミングを行うときに参考になる文書として、「Linux Programin mini HowTo」がある。これは、川島浩さんによって和訳もされている。この文書には、

- ・ C言語でのIN/OUT命令の使用法
- ・ ioperm(2)関数の使用法
- ・ トラブルシューティング
- ・ /dev/portを使ったアクセス方法
- ・ 時間待ちについて
- ・ パラレルポートの詳細仕様とその使用法

- ・ ゲーム（ジョイスティック）ポートの使用法

といった内容が書かれている。特に、この記事で使うプリンタポートの情報はとても詳しくたいへん参考になるので、ぜひ一度目を通してほしい。

I/O アクセス許可

I/Oポートが無制限にアクセスできるということは、そのコンピュータの資源を自由自在に扱えるということである。たとえば、IDEハードディスクを操作するI/Oポートのアクセスができると、ハードディスクのすべてのファイルを読んだり書いたりすることが可能だ。

セキュリティの概念がまったくないDOSなどでは問題にならないが、多数のユーザーが使う可能性のあるUNIX系のOSでは重大な問題となる。たとえば、shellを解放しているプロバイダで使われているサーバが、一般ユーザーのI/Oアクセスを許していると、悪意のあるユーザーはそのサーバを簡単に落としてしまうことができる。

このため、由緒正しいUNIX系のOSでは、I/Oアクセスをカーネルモードで動作している場合だけに限って許可していることが多かった。しかしながら、PC/AT互換機とともに発達してきたLinuxの場合、安直にI/Oをアクセ

スしたいという要求が強かったせいか、あるおまじないを唱えると、それ以降はユーザープロセスからでもI/Oアクセスが可能となるようなしくみが用意された。

このしくみを実現するシステムコールは*ioperm*と呼ばれる。たとえば、0x378から0x37bまでの4バイトのI/Oポートアクセスを許可したい場合は、

```
#define IO_PERMIT_ON    1
#define IO_PERMIT_OFF  0
ioperm(0x378, 4, IO_PERMIT_ON)
```

というように使用する。

この関数はroot権限でなければ使用できないという制限はあるが、伝統的なUNIXでは邪道ともいえる方法である。そのため、BSDユーザーのなかには、*ioperm*に違和感を感じる人も少なからずいたようである。

このような事情から、Linuxが生まれて間もないころは、この*ioperm*の存在が、BSDとLinuxとの大きな思想の違

リスト2 プリントポートにデータを出力するサンプルプログラム

```
/* === list2.txt ===== */
#include <stdio.h>
#include <unistd.h>
#define IO_ADR 0x378

/*-----*/
#ifdef __FreeBSD__
#include <fcntl.h>
#include <machine/cpufunc.h>
#define OUT(adr,data) outb(adr,data)
#define IO_ENABLE() (open("/dev/io",O_RDWR,0777) == -1)
#endif
/*-----*/
#ifdef linux
#include <sys/io.h>
#define OUT(adr,data) outb(data,adr)
#define IO_ENABLE() ioperm(IO_ADR,3,1)
#endif
/*-----*/
int main(void)
{
    if ( IO_ENABLE() ){
        printf("Can't enable io\n");
        return 1;
    }
    OUT(IO_ADR, 0x01);
    sleep(1);
    OUT(IO_ADR, 0x00);
    return 0;
}
/*-----*/
```

プリントポートのI/Oアドレス

I/Oアクセスできるようにする

プリントポートのbit0をHIGHにする

プリントポートのbit0をLOWにする

いであると言われることも多かった。

ところが、「便利だからいいじゃん」という声には勝てなかったのか、FreeBSD 1.0からは、“ /dev/io ” というデバイスが用意された。このデバイスをオープンすると、すべてのI/Oアクセスが許可されるという、Linuxよりもおらかな方法が採用されるようになったのである。ただし、FreeBSDでは、 /dev/ppiX というパラレルポート専用のデバイスが用意されている。これは、セキュリティの観点からは、I/Oを直接触るよりも優れた方法である。

出力の実験

今月号の最終目的であるエアコンのコントロールには、I/Oポートに出力を行わなくてはならない、そこでまずは、プリンタポートへの出力実験を行ってみる。必要なものは、

- LED
- 100 オーム程度の抵抗

である。LEDと抵抗を直列につなぎ（指でねじって接続）写真2のように接続する。LEDには極性があるので、グランド側と電源側を正しく接続する。

なお、抵抗は電源側にあってもよいし、グランド側にあってもよい。用意ができればリスト2のプログラムを、先ほどの入力実験のときと同じようにしてコンパイルする。そしてこのプログラムを実行すると、LEDが1秒間点灯して消えるはずである。

うまくいかない場合は、LEDの極性が逆かもしれないので、反対にして接続してみよう。また、プリンタポートの端子は表1にあるようなI/Oアドレスと各ビットに割り当

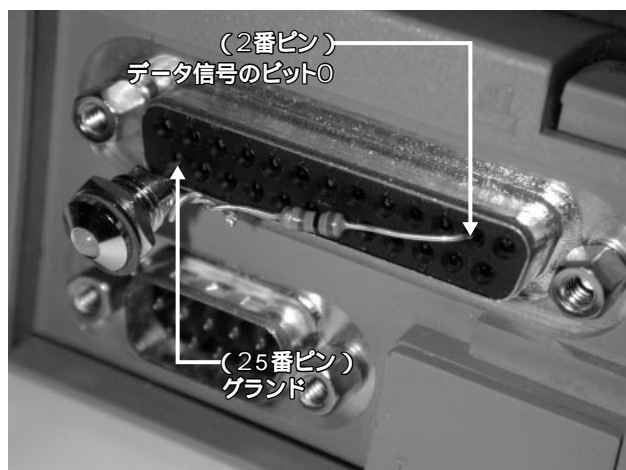


写真2 LEDと抵抗をつなぎ、2番ピンと25番ピンをつなぐ

てられているので、ほかの端子も試してみてもよいだろう。

エアコンのコントロールへの応用

さて、ここでエアコンの電源をON・OFFする方法を考えてみる。電源を入れたり、切ったりするのに最初に思いつく方法は、家庭用AC100V電源そのものをコントロールすることであろう。

しかしながら、エアコンはかなりの大電力を使う機器であり、数十アンペアの電流をコントロールするのは難しいばかりでなく、コントロールする装置の作り方がまずいと、場合によっては火事になることもありえる。また、最近のエアコンは、AC電源を接続しただけでは動作しないようになっているものも多いようだ。このため、エアコンに付属しているリモコンを使って、エアコンの電源をコントロールするという、簡単で安全な方法を今回は採用した。

赤外線方式のリモコンであれば、プリンタポートに接続した赤外LEDを特定のタイミングで光らせる方法が考えられる。しかし、通信フォーマットを調べるのが面倒なので、今回はリモコン自体を改造することにする。改造した場合、当然のことながらメーカーの保証は受けられなくなるが、万一壊したとしても、リモコンを新規に購入すれば済むので気が楽だ。

さて、リモコンの電源ボタンを押す方法であるが、まず考えられるのは、機械的に押すことであろう。ソニーのAIBOを持っている人は、AIBOをPCでコントロールして、足でリモコンのボタンを押すというのもおもしろい。

ピン番号	方向	信号の名称	I/Oアドレス	ビット
1	出力	STROBE	BASE+2	bit0
2	出力	D0	BASE+0	bit0
3	出力	D1	BASE+0	bit1
4	出力	D2	BASE+0	bit2
5	出力	D3	BASE+0	bit3
6	出力	D4	BASE+0	bit4
7	出力	D5	BASE+0	bit5
8	出力	D6	BASE+0	bit6
9	出力	D7	BASE+0	bit7
10	入力	ACK	BASE+1	bit6
11	入力	BUSY	BASE+1	bit7
12	入力	PE	BASE+1	bit5
13	入力	SLCT	BASE+1	bit4
14	出力	AUTO_FD_XT	BASE+2	bit1
15	入力	ERROR	BASE+1	bit3
16	出力	INIT	BASE+2	bit2
17	出力	SLCT_IN	BASE+2	bit3
18 ~ 25		GND	-	-

標準プリンタポートの場合、BASE=0x378

表1 プリンタポートのピン番号とポートの割り当て

次に考えられるのは、電氣的にボタンを押したことにする方法である。最近のリモコンではボタンが導電性のゴムになっていて、写真3のように、基板にそのゴムが押し付けられると通電するようになっているものがほとんどである。したがって、このパターンになんらかの信号を与えればボタンが押されたことになる。しかしながら、このパターンはマトリクス状になっていて、片方がチップの出力、片方が入力となっているので、回路構成をそれなりに調べて、それに合った信号を作成しなければならない。

そこで今回は、安直にリレーを使うことにした。リレーを使えば、極性を気にすることなく、ゴムスイッチを押すのとほとんど等価な結果が得られるからである。

今回使ったリモコンで、電源ボタン部分のゴムスイッチが接触する2つの接点のパターンを追いかけると、ゴムスイッチとは反対側の基板から信号を取り出すのが楽だった。そこで、写真4のようにしてリモコンの外に出すことにした。

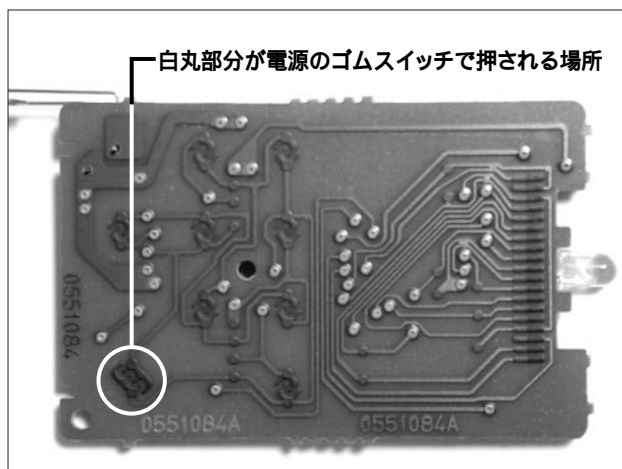


写真3 リモコンに内蔵された基板

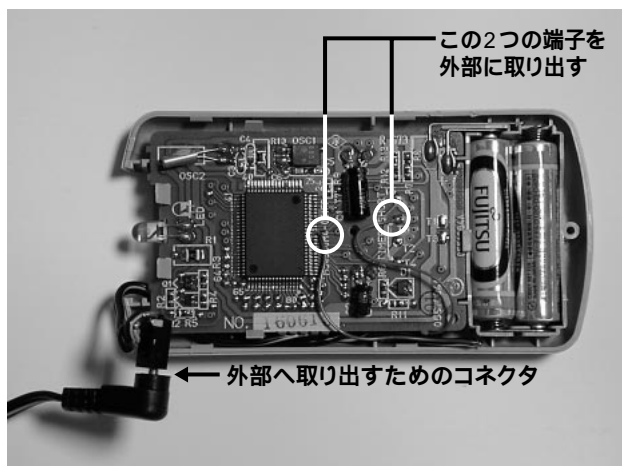


写真4 外部へ回路を取り出すための改造を施したリモコン

制御するのに使ったのは図1に示す回路で、NPNとPNPのトランジスタを使ってリレーを駆動している（リレーとは電磁石を内蔵したスイッチであり、電磁石に電流を流して、鉄のスイッチ切片を引き付けることにより、スイッチのON・OFFをするもの）。

プリンタポートからの出力がHIGH、つまり5Vになると、NPNトランジスタのベース（B）に電圧がかかり、ベース電流が流れて、NPNトランジスタがON状態になり、コレクタ（C）からエミッタ（E）へ電流を流せるようになる。

NPNのトランジスタがONになると、回路図で上のほうにあるPNPトランジスタのベース（B）側がグランド側に引っ張られてPNPトランジスタがON状態になり、コレクタ（C）からエミッタ（E）側へ電流が流せるようになる。つまり、2つの単3電池からの3Vの電圧がリレーに与えられるというわけである。リレーに電流が流れると、電磁石に出力側の鉄の切片が引き寄せられ、リモコンのボタンへと接続された2つの端子がショート状態になり、目的が達成される。

なお、この回路は前に別の目的のために、あり合わせの部品で作ったものであるため、今回の目的には多少冗長な構成になっている。プリンタポートは比較的電流を多く流せるので、抵抗をうまく選べばNPNのトランジスタ1個でも駆動できるはずである。この回路を使って、実際に制作したのが写真5である。プリンタケーブルはプリンタ側部分を切断し、必要な2本の線だけを適当なコネクタを介して基板へと接続してある。

一方、リレーの切片側はリモコンへと接続されているが、リモコンから線が出ているとじゃまなこともあるため、手元にあったコネクタを介して接続してある。

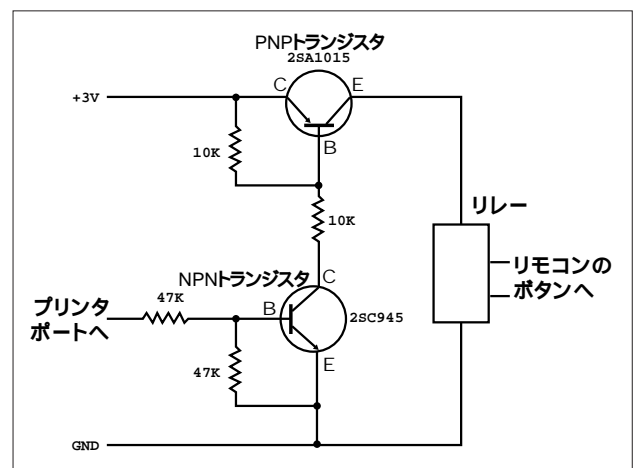


図1 リレーを制御するための回路図

インターネットからの エアコン無人コントロール

このハードウェアを動作させるソフトウェアは、LEDを光らせるのに使ったものと同じである。すなわち、先ほどのプログラムを実行するとエアコンの電源がONになり、もう一度実行すると電源がOFFになる。

筆者の住んでいる地域では、今月から定額制のIP接続サービスが使えるようになる。したがって、インターネットからtelnetで自宅のマシンにログインして、このプログラムを動作させれば、どこからでもエアコンがコントロールできるようになるはずだ。

常時接続していなくても、定期的に自動PPP接続をしてメールを取りに行き、そのメールを元にエアコンをコントロールする方法もあるだろうし、WebのCGIでキックするなど、やり方は無数に考えられる。UNIX系のOSでエアコンをコントロールする醍醐味は、やはりインターネットと連携したときにあるといえるだろう。

自動エアコン制御

UNIX系のOSは、24時間連続で運転させることが常識であり、特定の時間に特定のプログラムを実行することが

簡単にできるようになっている。

たとえば、今回作成したairconを/usr/local/bin/に置いて、/etc/crontabに、

```
50 1,5 * * * root /usr/local/bin/aircon
```

```
10 2,6 * * * root /usr/local/bin/aircon
```

という記述をすると、真夜中の1時50分と2時10分、そして5時50分と6時10分に/usr/local/bin/airconが実行される。つまり、1時50分～2時10分の間と5時50分～6時10分の間だけエアコンの電源がONになることになる。

なお、ゲームポートなどのA/D変換ができるポートに電子温度計を接続すれば、室温やCPUの表面温度を監視しながらのコントロールも可能になるだろう。

家電製品には、エアコン以外にもインターネットからコントロールすると便利なものがけっこうある。

インターネットから炊飯器がコントロールできれば、帰宅の時間が変わっても、i-modeの携帯電話で炊き上がりの時間を変更することも可能になるはずだ。

もちろん家電だけでなく、おもちゃを動かすなどの応用例は無限にあるし、I/Oアクセスに熟練すれば、カーネルのデバイスドライバを書くことも可能になる。この機会に、ハードウェアをコントロールする楽しみを味わってほしい

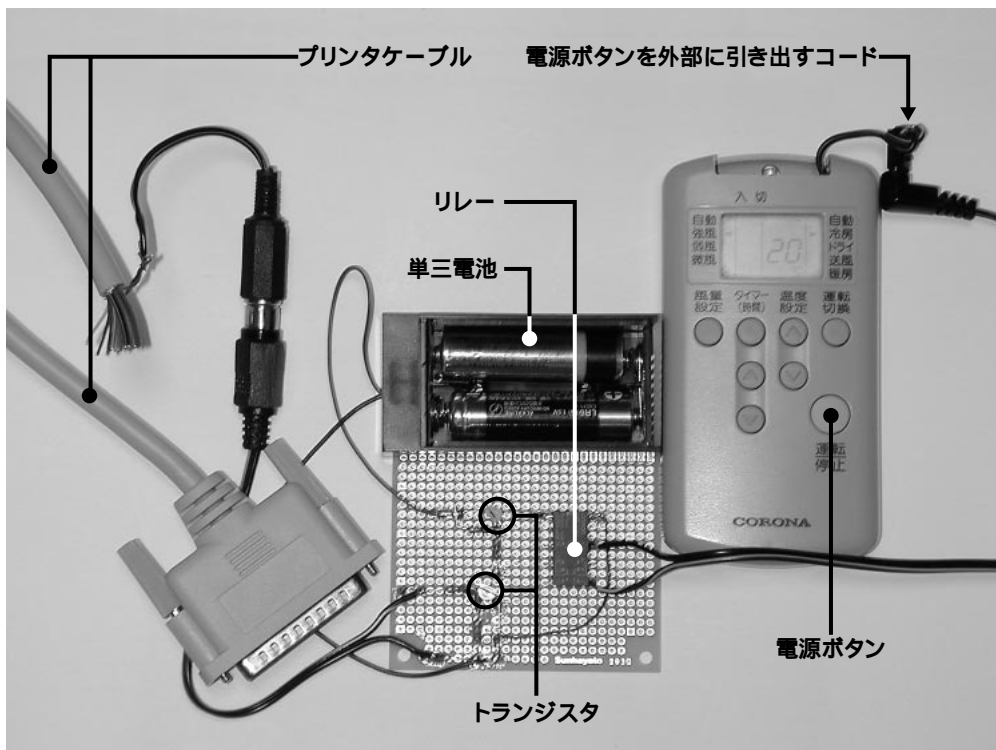


写真5 完成したリモコンと制御用の回路

ステップアップC言語

機械語

C言語は低水準言語に近い高級言語であるといわれる。低水準言語とは、機械語やそれと1対1に対応するアセンブラ言語である。機械語とは、メモリに配置される特定のビットパターンのデータであり、メモリからデータバスへそのビットパターンを出力し、それをCPUが読み取って実行する。

18年ほど前に筆者が趣味で組み立てた6809(8ビットのCPU)を使ったコンピュータには、データバスに直結した8個のLEDと8個のスイッチが接続してあった。

当時は8インチのフロッピードライブがジャンクでさえ数十万円もしていたので、プログラムは、このスイッチで入力することが多かったのである。6809の場合、ジャンプ命令は16進数で7Eであり、ビットパターンにすると、

```
0 1 1 1 1 1 1 0
```

であった。このジャンプ命令を入力するには、8個のデータバススイッチのうち両端の2つを下に倒せば(Lowにする)よいことになる。そして、ジャンプするアドレスの上位のアドレスと下位のアドレスをビットパターンに変換し、8個のスイッチでそれぞれ入力すると、初めて特定のアドレスにジャンプすることをコンピュータに指示することができたのである。

当時は加算命令や乗算命令など、かなり多くの命令をビットパターンで覚えていたことを懐かしく思い出すが、命令自体は人間が覚えるには難しすぎるため、今はもう完全に忘れてしまった。

このように人間が理解するのが難しい、ビットパターンの羅列が機械語と呼ばれるものなのである。

機械語とアセンブラ

機械語に対し、人間が理解しやすいように表したものがアセンブラ言語である。

Pentiumプロセッサ用の32ビットアセンブラ(たとえば、LinuxのGNU as)では、

```
jmp L1
```

L1:

```
jmp L1
```

といったプログラムを記述できるが、これをfoo.sとしてファイルに保存し、「as -a foo.s」を実行すると、

```
1 0000 EB00          jmp L1
2
3 0002 EBFE          jmp L1
```

という表示が得られる。ここで、0000と0002はアドレスであり、EBがjmp命令に対応する機械語である。00とFEはこのジャンプ命令の飛び先で、次の命令からの相対アドレスだ。つまり、EB00はその次のL1ラベルへ飛ぶことになる。一方、80以上は負の相対アドレス、つまり-2を示しているため、EBFEもまたL1にジャンプすることを意味している。

このように、アセンブラは、“jmp L1”というような人間にわかりやすい文字列を、“EBFE”というように機械語に自動的に変換してくれる。

つまり、アセンブラを使うことにより、EBがジャンプ命令に相当するということを手間が記憶する必要がなくなるので、たいへん便利なのだ。

アセンブラとC言語

アセンブラの出現により、人間は意味不明なビットパターンと格闘する必要がなくなった。たとえば、プリンタポートのI/Oアドレスである0x378(10進数で888)に0x01を出力するには、

```
B001 BA78030000 EE
```

という意味不明な機械語でなく、

```
movb $1,%al
movl $888,%edx
outb %al,%dx
```

という記述をすればよい。しかしながら、アセンブラで記述する場合には、「out命令がDXレジスタのアドレスにALレジスタの内容を出力することしかできないため、あらかじめALレジスタに1を、DXレジスタに888(0x378)を入れておかなければならない」といった、Pentiumプロセッサ命令の制限を知っておく必要がある。

これをもっと楽にしようとしたのがC言語であり、上記は本文で説明したCのプログラムでは、

```
OUT(0x378, 0x01)
```

と記述できる。

gccでは「-S」オプションを付けると、アセンブラのソースコードが出力できるので、

```
OUT(0x378, 0x01)
```

が実際に、

```
movb $1,%al
movl $888,%edx
outb %al,%dx
```

というアセンブラ言語に翻訳されるかどうかを、実際に試してみしてほしい。

面倒でも必要なアセンブラ

I/Oアクセスを伴う場合は、高速な応答が要求されることが多く、Cで記述しては間に合わず、アセンブラを使わざるを得ないこともよくある。また、ブートローダのようにプログラムサイズが極端に制限されている場合も、アセンブラが活躍する。

なお、コンパイラもソフトウェアであり、必ずバグが存在する。筆者はつい先日、某組み込みCPU用Cコンパイラで、構造体のメモリ配置に関するバグに遭遇し、アセンブラコードを出力して、リンカのバグであることを確認することができた。

C言語などの高級言語が発達した現在でも、やはり基本は機械語であり、アセンブラを理解することは重要である。

Ruby で行こう

もう2000年も後半です。去年の後半から正月までずっと盛り上がっていた2000年問題(Y2K)の記憶も薄れてきたのではないのでしょうか? あれほど騒いだのはいったいなんだったのか、という感じです。今回はそのような問題の根幹となった「時刻と暦」について考えます。

第9回 タイム・アフター・タイム

文: 赤松智也

Text: Tomoya Akamatsu

2000年問題とは、ひと言でいうと「西暦の年号を2桁で表現しているシステムは、繰り上がりが起きる2000年に、いきなり過去(1900年)に戻ってしまう」というものでした。

2000年問題が最初に発生したのは、なんと1970年代なのだそうです。30年ローンを扱うプログラムが、2000年以後に終了するローンを正常に扱えなかったのだと聞いています。そんなに昔からわかっていたことが、あれだけ重大視されるのはどういうことかと思うのですが、見積りが甘いというかなんというか.....。

映画やドラマのような大きな問題は起きなかったものの、実際に日付が「19100年」と表示されるといったささいな問題はあちこちで発生したようです。

19100年という表示は、以下のような状況で発生しました。

```
printf "19%d\n", year
```

ここで、yearに「西暦 - 1900」の数字が入っていると、2000年が19100年になってしまいます。UNIXでは、localtime(3)関数が返すtm構造体が年を「西暦 - 1900」で表現するので、

```
printf "%d\n", year+1900
```

とすべきところを、思わず"19"を頭に付けてしまうことが結構あったようです。Rubyでは、年は実際の西暦年で扱

いますから、このような問題とは無縁ですけど。

Y10K

人類はなんとか2000年問題を乗り切りました。しかし、人類には別の問題が待ちかまえているようです。それは「Y10K」です。つまり、年号は4桁であると信じ込んだ人が、年データを格納する領域を4桁しかとらず、西暦10000年にもう一度同様の問題が発生するというものです。

え、そのころには人類は生きていないって?

まあ、そうかもしれません。それでなくても、8000年後には現代のようなコンピュータとソフトウェアは使われていないかもしれません。しかし、考えてみれば「このソフトウェアは2000年が来る前にリプレースされるはずだ」という思い込みが、2000年問題の大騒ぎを呼んだわけです。今から8000年後の心配をするのも、まったくの無駄ではないかもしれません。おそらくは無駄でしょうけどね。(笑)

2038年問題

しかし、本当の「次の問題」はそんな未来にではなく、すぐ近くに待っています。それは、「2038年問題」です。聞いたことがありますか? この問題には、Rubyも無縁ではられません。

2038年問題がなんであるかを解説するには、少々技術的

な話をしなければなりません。

UNIXに限らず、多くのOSでは時刻を「過去のある時点」からの経過時間（秒）で表現しています。人間が「年・月・日」の組み合わせで時刻を表現するのは対照的です。UNIXの場合、過去のある時点とは1970年1月1日午前0時（UTC）です。

たとえば、Rubyの誕生日（1993年2月24日）の日本時間の正午は、コンピュータにとっては「730522800」という数字で表現されます。Rubyを使えば、以下のようにして確かめられます。

```
p Time::local(1993,2,24,12,00).to_i
# => 730522800
```

問題は、コンピュータが扱う整数の大きさにはふつう限界がある、ということです。整数のサイズとして、32ビットを用いているシステムでは表現できる整数は、最大で2147483647、最小で-2147483648です。21億強というとかなり大きいようですが、実際には全人類の数も表現できないほど小さい数です。このさほど大きくない整数で秒数を数えた場合、前述の1970年1月1日午前0時（UTC）から数え始めると、限界が来るのは2038年1月19日3時14分7秒（UTC）です。そこから先は、2000年問題と同様に過去に逆戻りです。

38年後といえはざいぶん先の話です。そのときまでには、すべてのCPUとOSは64ビット化されているでしょうか？ そう願いたいものです。64ビット化されていれば問題はざいぶん先送りできます。西暦292277026596年12月4日15時30分7秒（UTC）までは大丈夫のようです。この未来の日付は、以下のようにして確認しました。

```
require 'date2'
last = (1<<63)-1 # 最大の符号付き64ビット整数
puts Date::new3(1970,1,1)+(last/(24*60*60))
# => 292277026596-12-04
p (last%(24*60*60))/3600
# => 15
p (last%(24*60*60)%3600/60)
# => 30
p (last%(24*60*60)%3600%60)
# => 7
```

整数サイズに限界のないRubyだからこそ、こんなに楽

に計算ができるのだと思います。西暦2922億年ってのはちょっと想像を越えますね。2038年までに、コンピュータとOSの64ビット化が完了することを望みます。ゲーム機でさえ、64ビットとか128ビットを名乗る昨今、こんな心配は不要でしょうか。

Time クラス

もう少しRubyらしい話もしましょう。ここで、時刻を司るTimeクラスの機能についてまとめておきます。Timeクラスのメソッドの概要は表1のようになっています。

RubyのTimeオブジェクトは、Cライブラリの時刻関係の関数と構造体をきれいにクラスにまとめているので、非常に使いやすいのです。整数値を、時刻を表現すると解釈するのではなく、時刻を時刻のまま扱えるのがありがたいところです。最もありがたいのは、Ruby全体で時刻を表現する部分に、Timeオブジェクトが使われていることです。ですから、

```
p File.mtime("ChangeLog")
# => Wed Jul 05 13:18:18 JST 2000
```

というふうに的確に表示されます。これがもし、以下のようにならば、直感的にはいつなのかわかりません。

```
p File.mtime("ChangeLog")
# => 962770698
```

地球は丸い

海外旅行をしたことはありますか？ 海外へ電話したことは？ そういうときに気になるのは時差です。地球は丸いので、朝は東から西へと地球の上を移動します。結果として日本はヨーロッパよりも8時間、アメリカよりは15時間程度早く朝を迎えることとなります。

ですから、同じ「朝8時」と言っても、それがどこの時間での朝8時なのかでまったく違う時間を指すこととなります。したがって、「日本時間の朝8時」というように、どこの時間なのかを明確にする必要があります。

しかし、それではいろいろな場所の時間を比較する場合に、お互いの時差を計算する必要があります。それは面倒

なので、国際的にひとつの時間に揃えることができます。それが「協定世界時 (Coordinated Universal Time = UTC)」です。グリニッジ標準時間 (Greenwich Mean Time = GMT) と呼ばれることもあります。Ruby (そしてUNIX) の時刻モデルは、ローカルタイムと協定世界時を扱うことのできるモデルです。逆に言うと、第3のタイムゾーンの時刻は簡単に扱うことはできません。たとえばこんな感じです。

```
# 本誌の発売日正午 (ローカルタイム)
t1 = Time::local(2000,8,8,12)
p t1
# => Tue Aug 08 12:00:00 JST 2000

# 本誌の発売日正午 (UTC)
t2 = Time::gm(2000,8,8,12)
p t2
# => Tue Aug 08 12:00:00 GMT 2000
```

Time オブジェクトの時刻表現をローカルタイムで行うか、協定世界時で行うかの設定も可能です。

```
p t1
# => Tue Aug 08 12:00:00 JST 2000

# 協定世界時に設定
t1.gmtime
p t1
# => Tue Aug 08 03:00:00 GMT 2000

# ローカルタイムに設定を戻す
t1.localtime
p t1
# => Tue Aug 08 12:00:00 JST 2000
```

なお、Coordinated Universal Timeの略なのになぜUTCと呼ぶかと疑問に思いませんでしたか？ どうも英語とフランス語が対立して名称は英語、略語の語順はフランス語と痛み分けしたらしいという話らしいです。

サマータイム

時差もさることながら、もうひとつ、日本にない問題が

あります。「サマータイム」です。サマータイムは、別名 Daylight Saving Time といって、夏のある時期、時計を1時間早めることで朝の明るい時間を有効に活用しようというルールです。アメリカでは、4月の最初の日曜日から10月の最後の日曜日までがサマータイムになります。

では、サマータイムの実際を見てみましょう。ご存じの

スーパークラス	
Object	
インクルードしているモジュール	
Comparable	
クラスメソッド	
Time::at(time[,usec])	経過秒から Time オブジェクトを生成する
Time::gm(year[,mon, day,hout,min,sec])	Time オブジェクト (協定世界時) を生成する
Time::local(year[,mo n,day,hout,min,sec,u sec])	Time オブジェクト (ローカルタイム) を生成する
Time::now	現在の Time オブジェクトを生成する
Time::times	自身のプロセスとその子プロセスが消費したユーザー / システム CPU 時間の積算を返す
インスタンスメソッド	
time + n	n 秒後の Time オブジェクトを返す
time - time	2つの時刻の差を秒数 (Float) で返す
time - n	n 秒前の Time オブジェクトを返す
time > time	時刻を比較する
time >= time	
time < time	
time <= time	
time <=> time	
time.asctime	時刻を表現する文字列を返す
time.ctime	
time.day	日を返す
time.mday	
time.gmtime	タイムゾーンを協定世界時に設定する
time.gmt?	タイムゾーンが協定世界時に設定されているときに真を返す
time.sec	time のそれぞれの部分を返す
time.min	
time.hour	
time.mday	
time.day	
time.mon	
time.month	
time.year	
time.zone	
time.isdst	
time.localtime	以後の時刻表示をローカルタイムで行う。自身自身を返す
time.strftime(fmt)	フォーマットに従った時刻表示を行う
time.to_f	time を秒単位の Float として返す
time.to_i	time を秒単位の整数として返す
time.tv_sec	epoch からの秒数を整数で返す
time.tv_usec	時刻の μ 秒部を返す
time.usec	

表1 Time クラス

ように日本にはサマータイムがありませんし、UTCにもサマータイムはありませんから、タイムゾーンを日本に設定している限り、サマータイムにお目にかかることはありません。そこで、タイムゾーンを強制的にアメリカ東部時間に設定してみましょう。タイムゾーンの設定は環境変数ひとつで行えます。

```
$ export TZ=US/Eastern
```

これでアメリカ東部時間になりました。

```
$ date
Tue Aug 08 09:45:00 EDT 2000
```

タイムゾーンがEDTになっています。これはEastern Daylight saving Timeの略なんだそうです。サマータイムが適用されない時期は、この部分がEST (Eastern Standard Time) になります。タイムゾーンを元に戻すには、

```
$ unset TZ
```

とします。

では、サマータイムがどのように始まるかをみてみましょう。米国の2000年のサマータイム開始日は4月2日です。以下のスクリプトで確認できます。

```
# 4月2日の始まりの時刻
t = Time::local(2000,4,2)
4.times do

  p t

  t += 60*60
end
```

このプログラムの出力は以下のようになります。

```
Sun Apr 02 00:00:00 EST 2000
Sun Apr 02 01:00:00 EST 2000
Sun Apr 02 03:00:00 EDT 2000
Sun Apr 02 04:00:00 EDT 2000
```

あれ？2時がありませんね。そうか、こういうふうにして1時間早めるのか。知りませんでした。時間がジャンプした後から、ESTがEDTに変わっているのもわかります。

中には、以下のような出力になった人もいるでしょう。

```
Sat Apr 01 23:00:00 EST 2000
Sun Apr 02 00:00:00 EST 2000
Sun Apr 02 01:00:00 EST 2000
Sun Apr 02 03:00:00 EDT 2000
```

実は、Ruby 1.4.4以前のバージョンには、サマータイムの扱いにバグがあります。この出力はバグのせいです。Rubyは日本製なので、長らくこのバグに気づく人がいなかったんですね。

サマータイム期間の終わりも見てみましょう。2000年の10月最終日曜日は10月29日ですから、上のプログラムの日付を変更するだけで対応できます。

```
t = Time::local(2000,10,29)
```

結果は以下のとおりです。

```
Sun Oct 29 00:00:00 EDT 2000
Sun Oct 29 01:00:00 EDT 2000
Sun Oct 29 01:00:00 EST 2000
Sun Oct 29 02:00:00 EST 2000
```

ふーん、午前1時が2回あるんですね。勉強になるなあ。私自身はサマータイムを経験したことがないのですが、1年に2度もすべての時計を1時間ずらすというのはなかなか面倒そうです。実は日本でも戦後すぐの時期にしばらく導入されていたことがありますが、生活のリズムが狂うと不評で、すぐにとりやめになったそうです。今でもときどき景気対策とか省エネだとかでサマータイム導入がとりざたされますが、プログラマーとしては絶対にやめてほしいですね。Rubyにあったのと同じようなバグが大量に発生しそうです。Y2Kに続く新たな社会問題かもしれません。

時刻と日付

このようにTimeクラスはなかなか使い勝手の良いクラスなのですが、時間に関するあらゆることをカバーするには少々弱点があります。最大の弱点はなんといっても「表

現できる時刻の範囲に制限がある」ことでしょう。

Timeクラスのオブジェクトは、1970年1月1日から2038年1月19日までの範囲だけを表現します。簡単な話、1970年より前に生まれた人は自分の誕生日すら表現できないわけです。

Timeクラスは時刻を秒単位（より正確にはマイクロ秒単位）で表現できますが、場合によってはもっと大ざっぱに日単位での表現でかまわないというケースもたくさんあるでしょう。

Rubyには、そのようなケースのためのクラスも用意されています。それはDateクラスです。Dateクラスは日単位での表現を用いるため、十分過去から十分未来までを表現できます。ちょっとした実験を行ったところ、西暦10兆年（！）までは計算できました。そこから先は日付計算の誤差で日付がずれてしましますが、まあ、宇宙ができてからでさえ、せいぜい150億年しかたっていないのに、そんな未来のことを扱うことはないでしょう。

date と date2

実は歴史的な事情から、Rubyには2つのDateクラスがあります。requireするライブラリ名の違いから、ここではdateとdate2と呼びましょう。この2つのクラスは以下の点で異なっています。

1. dateではDateオブジェクトの生成方法は「年・月・日」を指定する方法だけですが、date2では「ユリウス日を指定」、「年と通算日を指定」、「年・月・日を指定」の3種類の方法が提供されています。
2. date2はユリウス日（紀元前4712年1月1日を基準とする日数）を基準日として持ちますが、dateは年月日のままデータを持ちます。dateの日付計算の基準は紀元1年1月1日です。
3. date、date2の双方とも改暦（ユリウス暦からグレゴリオ暦への変更）をサポートしています。しかし、dateはイギリスでの改暦にしか対応していません。date2は任意の改暦に対応しています。
4. 改暦だけでなく、date2のほうが機能が多彩です。たとえばdate2には、ある日からある日までのあいだ繰り返すイテレータ（downto、step、upto）があります。
5. 暦計算のアルゴリズム

すでに述べたように、dateとdate2では基準にする日が

異なります（紀元前4712年1月1日と紀元1年1月1日）。dateは紀元前の計算に対応していないようです。また、dateは歴史的にバグが多かったのも事実です。最近はもうなくなったようですが（原稿執筆直後に、1.5系のdate.rbはdate2.rbの統合が行われました。APIも基本的にdate2のものに揃えられ、ただDate::newだけはDate::3の別名に変更されました）。

dateはRubyの最も古いプログラムのひとつです。ちょっと古すぎるかもしれないので、一般的な使用には、より新しいdate2のほうが良いかもしれません。

Dateクラスを使った例として、日数計算があります。

```
require 'date2'

# 今日はRuby誕生何日目?
printf "今日はRuby誕生 %d 日目\n", Date::today-Date::new3(1993,2,24)
```

出力は、たとえば、

```
今日はRuby誕生 2723 日目
```

となります。Rubyが誕生5千日目を迎えるのは、

```
require 'date2'
printf "Ruby誕生5千日目は %s です\n",
Date::new3(1993,2,24)+5000
```

とすれば求められます。結果は以下のとおりです。

```
Ruby誕生5千日目は 2006-11-03 です
```

改暦

Dateクラスの説明に「改暦」という言葉がありました。これはなんだかわかりますか？これは、西洋における旧暦（ユリウス暦）から新暦（グレゴリオ暦）への切り替えのことです。ややこしいのは、この改暦のタイミングが国によって違うことです。

たとえば、イギリスでは1752年9月に行われました。UNIXのcalコマンドはイギリスの改暦に対応しています。実際に見てみましょう。

```
% cal 9 1752
      September 1752
 S M Tu W Th F S
          1  2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

おやおや、なんだか変ですね。2日から14日まで日付が飛んでいます。これは改暦、つまり暦の切り替えに伴う日付のジャンプです。

主な国の暦の切り替えを表2に示します。

dateのほうは、calにならって改暦はイギリス式に固定ですが、date2のほうは任意の改暦日を選択できます。Ruby付属のサンプルプログラム「cal.rb」は、calコマンドのRuby版ですが、date2の機能を活かして、各国の改暦日に対応しています（ISO3166の国コードで指定する）。

```
% ruby sample/cal.rb -c es 10 1582
% ruby sample/cal.rb -c gr 3 1924
```

などを試してみてください。試した後は中身を見てもよいでしょう。

イースター

「イースター」をご存じですか？復活祭とも言います。仏教国（？）の日本ではあまりなじみがありませんが、イエス・キリストの復活をお祝いする日です。イエス・キリストの誕生をお祝いするクリスマスは、すっかりキリスト教色を失ったものの、それでも日本に定着していますし、ここ数年ハロウィーンも話題になっていますが、イースターはさっぱりです。

だいたい、イースターがいつなのかわかりにくいというのが問題なのかもしれません。クリスマスは12月25日、ハロウィーンは10月31日と毎年決まっていますが、イ

国	切り替え日
ドイツ	1700年03月01日
スペイン	1582年10月15日
フランス	1582年12月20日
イギリス	1752年09月14日
ギリシャ	1924年03月23日
イタリア	1582年10月15日
ロシア	1918年02月14日
アメリカ	1752年09月14日

表2 改暦のタイミング

スターは「あるルール」に従って毎年変わります。

このルールは、聖書に書いてあるイエス・キリストの復活した日にちなんで、以下のように決められています。

春分の日後の、最初の満月の後の、最初の日曜日

まん中の満月がくせもので、これのおかげでカレンダーを見ただけでは、いつがイースターなのかわかりません。せいぜい3月の終わりから4月の後半にかけてのどれかの日曜日というくらいまでです。

しかし、ちょっとした公式を使うと、いつがイースターかを計算できます。サンプルプログラムのgoodfriday.rbを使うと、今年のイースターの日と、それを基に計算できるほかのキリスト教の祭日の日付を出力してくれます。goodfriday.rbの出力を図1に示します。こんなにあるんですね。

では、goodfriday.rbを改造して、今後10年のイースターの日付を出力するプログラムを書いてみましょう（リスト1）。これさえあれば、イースターの日付に悩むことはありません。

関数easter()はgoodfriday.rbからそのままコピーして来ました。残念ながら、私にはこの式でなぜその年のイースターが求められるのか、さっぱりわかりません。やはり、付け焼き刃の暦の知識では立ち打ちできないようです。後

```
2000-02-20 Septuagesima Sunday
2000-02-27 Sexagesima Sunday
2000-03-05 Quinquagesima Sunday (Shrove Sunday)
2000-03-06 Shrove Monday
2000-03-07 Shrove Tuesday
2000-03-08 Ash Wednesday
2000-03-12 Quadragesima Sunday
2000-04-02 Mothering Sunday
2000-04-09 Passion Sunday
2000-04-16 Palm Sunday
2000-04-20 Maunday Thursday
2000-04-21 Good Friday
2000-04-22 Easter Eve
2000-04-23 Easter Day
2000-04-24 Easter Monday
2000-04-30 Low Sunday
2000-05-28 Rogation Sunday
2000-06-01 Ascension Day (Holy Thursday)
2000-06-04 Sunday after Ascension Day
2000-06-11 Pentecost (Whitsunday)
2000-06-12 Whitmonday
2000-06-18 Trinity Sunday
2000-06-22 Corpus Christi (Thursday after Trinity)
```

図1 goodfriday.rbの出力

は10回ループを回して、計算したイースターの日を出力するだけです。

easter.rbの出力は以下のとおりです。

```
2000-04-23
2001-04-15
2002-03-31
2003-04-20
2004-04-11
2005-03-27
2006-04-16
2007-04-08
2008-03-23
2009-04-12
```

parsedate

日付の書き方にはいろいろとあるものです。まあ、日本式の「平成12年8月8日」を解釈するのはちょっと難しそうな気がしますが（日本語の場合、解釈の前にエンコーディングという難関がある）、英語っぽい表現に限っても、

- **ctime(3)、asctime(3)の返すもの**

```
Mon Dec 31 23:59:59 2001
```

- **date(1)の出力**

```
Mon Dec 31 23:59:59 JST 2001
```

- **RFC 822 で定義されたもの**

```
Mon, 31 Dec 2001 23:59:59 JST
```

- **RFC850 で定義されたもの**

```
Monday, 31-Dec-01 23:59:59 JST
```

- **ISO8601 の表現**

```
2001-12-31 23:59:59
```

のような、さまざまなバリエーションがあります。これらを「できるだけ善意に」解釈しようというライブラリが `parsedate` です。 `parsedate` ライブラリは、 `ParseDate::parsedate` というモジュール関数を提供します。 `parsedate` は与えられた文字列の中の「日付っぽいパ

ターン」を解釈して、「年、月、日、時、分、秒、タイムゾーン、曜日」の配列を返します。わからなかった部分には、 `nil` が入ります。上であげた例を解析させてみましょう（リスト2）。

プログラム `pdtest.rb` の出力は以下のとおりです。

```
[2001, 12, 31, 23, 59, 59, nil, 1]
[2001, 12, 31, 23, 59, 59, "JST", 1]
[2001, 12, 31, 23, 59, 59, "JST", 1]
[1, 12, 31, 23, 59, 59, "JST", 1]
[2001, 12, 31, 23, 59, 59, nil, nil]
```

結果を見るとRFC850で定義されたパターンだけ、西暦1年だと見なされています。 `parsedate` は、デフォルトでは下2桁の年号をサポートしません。本当に西暦1年を期待する場合もあるでしょうから、ライブラリがでしゃばらないのが正しい方針なのだと思います。よくある「69以下なら2000を、70以上なら1900を加える」という規則の適用には、第2引数に真を与えます。具体的には、 `p ParseDate::parsedate(line, true)` のようにします。結果は以下のとおりです。

リスト1 easter.rb

```
#!/usr/bin/env ruby

# based on goodfriday.rb: Written by Tadayoshi Funaba
1998

require 'date2'

def easter(y)
  g = (y % 19) + 1
  c = (y / 100) + 1
  x = (3 * c / 4) - 12
  z = ((8 * c + 5) / 25) - 5
  d = (5 * y / 4) - x - 10
  e = (11 * g + 20 + z - x) % 30
  e += 1 if e == 25 and g > 11 or e == 24
  n = 44 - e
  n += 30 if n < 21
  n = n + 7 - ((d + n) % 7)
  if n <= 31 then [y, 3, n] else [y, 4, n - 31] end
end

year = Time::now.year
10.times do |i|
  puts Date::new3(*easter(year+i))
end
```

```
[2001, 12, 31, 23, 59, 59, nil, 1]
[2001, 12, 31, 23, 59, 59, "JST", 1]
[2001, 12, 31, 23, 59, 59, "JST", 1]
[2001, 12, 31, 23, 59, 59, "JST", 1]
[2001, 12, 31, 23, 59, 59, nil, nil]
```

日付の解析については、最近のUNIXには、`strptime(3)`という関数があります。これは期待するフォーマットを指定して日付などの時間情報を取り出すという、ちょうど`strftime(3)`の逆を行う関数です。Rubyには、`strptime(3)`相当は標準添付されていません。

しかし、`date2.rb`などの一次配布パッケージである`date2-1.14.tar.gz`には、`strptime.rb`が含まれています。このパッケージには、今回紹介した`date2.rb`、`parsedate.rb`（実際にRubyに添付されているのはパッケージ中の

リスト2 `pdtest.rb`

```
require 'parsedate'

while line = DATA.gets()
  p ParseDate::parsedate(line)
end
__END__
Mon Dec 31 23:59:59 2001
Mon Dec 31 23:59:59 JST 2001
Mon, 31 Dec 2001 23:59:59 JST
Monday, 31-Dec-01 23:59:59 JST
2001-12-31 23:59:59
```

`parsedate3.rb`）、`cal.rb`、`goodfriday.rb`が含まれているだけでなく、日付と暦に関する興味深いプログラムが入っています。このパッケージは以下から取得できます。

<http://www.kt.rim.or.jp/tadf/date2-1.14.tar.gz>

まとめと謝辞

時刻も暦も奥深いものです。今月分の手稿を書き始めたときには、ごく一部としてとりあげるつもりだったのですが、1回分まるまる時間と暦で使い切ってしまいました。今回は時間と暦に関するうんちくばかりで、Rubyの話が妙に少なかったような気がします.....。

今回の話（特に暦関係）は、Ruby添付の`date`、`date2`、`parsedate`、`cal.rb`、`goodfriday.rb`の各プログラム、および以下のWebページを参考にさせていただきました。

<http://www.kt.rim.or.jp/tadf/programming-and-calendar.html>

特に、Webページはプログラミングと暦の関係について非常によくまとめてあります。これらプログラムおよびWebページはすべて、ふなばただよしさんの手によるものです。心からお礼を申し上げます。

Column

今月のRuby 1.5

先月号のこのコラムで「今ごろ1.6がリリースされているはず」と書きましたが、予想どおり（おいおい）6月末には1.6.0はリリースされませんでした。ぎりぎりで見つかった一貫性のない仕様の修正などが理由のようです。

やはり、どうしても原稿を執筆してから印刷されるまでにタイムラグのある雑誌連載では、変化の激しい最新の情報には追隨するのは難しいですね。ともあれ、1.5系の仕様はほぼ固まったようですから（原稿執筆時現在）、いくらなんでも今月号が書店に並ぶころには、リリースされているのではないのでしょうか。

7月前半までに、1.5系に行われた主な変更は以下のとおりです。

- `/`、`%`、`divmod`の挙動が明確化された
Fixnum、Bignum、Float、Rationalで、`/`、`%`、`divmod`の挙動が微妙に食い違っていました。統一されました。

- `modulo`が導入された
以前からある`remainder`（剰余）と対になる`modulo`（法）が導入されました。挙動は`%`と同じです。

- `date`ライブラリが`date2`と同じアルゴリズムを使うもの（ふなばさん作）に差し替えられた

- `rand`関数の引数が省略できるようになった

- `Time::local`、`Time::gm`でマイクロ秒が指定できるようになった

- `Time`のメソッド名でGMTの代わりにUTCも使えるようになった（`Time::utc`、`Time#utc`、`Time#utc?`）

- こまごまとしたバグが修正された

仕様のフリーズ宣言が出てから1ヵ月以上経つ気がするんですが、それでもぎりぎりまで仕様が変化するのが、なんとなくRubyらしいですね（不謹慎ですが）。

Webサーバ構築術(最終回)

本連載で、ひと通りApacheの使い方と解説、運用の方法を書いてきたが、最終回はApache自体の運用と管理を解説したい。これで、Webサイトとして最低限必要なことや、やらなければならないことがたいていはいはできるはずだ。

Apacheの管理

文：中島昌彦

Text：Masahiko Nakajima

ログのローテートと解析

ApacheをRPMパッケージでインストールしているときは、cronによってlogrotateが動き、ログファイルを週間単位で切り出して、圧縮保存してくれているはずだ。ログファイルはしっかりとメンテナンスをしていないと、あっという間に巨大化する。しかも、Webサイトは突然アクセス数が増えることがある。dfを見て、ログ保存先のファイルシステムの空き容量が十分にあるからと思った翌日に、突然アクセス数が増えて、ファイルシステムフルになることある。そこまで極端な例は少ないが、何もせずにログファイルを放置しておく、知らないうちに巨大化していることは間違いない。

ログファイルは/var/log/httpdディレクトリに置かれて、logrotateで1週間単位で切り出される。そして、過去ログは過去1カ月(4週間)分がバックアップ保存される。すなわち4週間を過ぎたログは捨てられてしまう。

また、Apacheをソースからコンパイルしている場合には、Apacheのログは/usr/local/apache/logs以下に保存される。そのままではlogrotateによるログの切り出しは働かないので、1つのファイルにどんどんログが貯まっていくことになる。

ログを保存している/var/logや/usr/local/apache/logsディレクトリの容量が十分でないと、たとえ1カ月分のログとしても、ファイルシステムがフルになる可能性もある。RPMパッケージでインストールしようと、ソースからインストールしようとログのメンテナンスは必ずしないといけないわけだ。

さらに話をややこしくすることとして、Apacheのログは他のサーバログと異なり、解析することでマーケティングデータとなることが挙げられる。会社なら新製品が出ればアクセス数が増えるであろうし、学校なら入試が近くなればアクセス数が増える。また、ビジターがどこから来ているかを調べることで、どういう人が自分のサイトに

興味を持っているのかもわかる。単純にログをローテートするだけでなく、ローテートしたあとには、きちんと解析処理しておくことが必要だ。

logを切り出し、ログ解析ソフトに渡す準備をする

Apacheをソースからインストールしている場合のログファイルは、/usr/local/apache/logsがデフォルトのログディレクトリとなる。httpd.confの設定を変えていなければ、そのディレクトリに、access_log、error_logという名前がApacheのアクセスログファイルが作られている。

貯まったログファイルを切り出すには、access_logとerror_logの2つを単純に別のファイル名、もしくはディレクトリにmvするのだが、書き込みファイルを移動したまま放置すると、このあとのアクセスもしくはエラーログの書き込みが行われなくなってしまう。そこで、ログファイルを移動したあとに、Apacheを再起動してログファイルへの再open処理をするといった手順

を踏めばいい。

シェルスクリプトとしてはlogcut.sh (リスト1-1) のようなもので、このスクリプトをcronとして定期的に動かせばいい。変数basedirにはApacheのログファイルディレクトリを指定する。そして、basedirで指定したディレクトリの下に、historyディレクトリを作っておく。

logcut.shが、/home/rootディレクトリにあるとすると、cronに登録する手順は、

```
# chmod +x /home/root/logcut.sh
# crontab -e
```

という作業だ。エディタが起動するので下記の1行を追加する。

```
0 7 * * * /home/root/logcut.sh
```

この場合は毎朝7時にlogcut.shが動くが、毎週月曜日の朝7時に動くようにするなら、

```
0 7 * * 1 /home/root/logcut.sh
```

とする。Webサーバのアクセス頻度によって違うだろうが、毎日または1週間に1回程度動かしておけば十分だ。ただし、注意点としてcronで動作させる実行時間が密接に関係する。通常のHTMLファイルのコンテンツだけなら、いつログを切り出そうと問題ない。し

かし、CGIが動いていて、apachectl restartのタイミングと同時にCGIがファイルをオープンして書き込もうとしたとき、CGIが書き出したファイルが安全である保証がない。このため、CGI側でログカット時間の前後数秒はファイル書き込みを抑制するという工夫をしたり、ログカットをする時間をできる限りアクセスの少ない曜日、時間帯にセットするという配慮が必要だ。たいていは、月曜日の朝となることだろう。

Webalizerで ログを解析する

ログを切り出したからといって、そのままほったらかしているとマーケティングデータの集計がとれない。どのサイトから自分のWebサイトを閲覧しにきているのか、さらには、何時ぐらにもっともアクセスが集中するのか、そういったログファイル解析をして、自分の会社の製品や学校のWebサーバ運用に役立てよう。

切り出したログファイルは、そのままログ解析ソフトでログ解析をしてしまいたい。今回は、フリーなログ解析ソフトの中でもメジャーなAnalogではなく、Webalizerを使ってみる。あまりたいした理由ではないが、Webalizerのほうがセットアップする項目が少なく、見た目がAnalogよりグラフィカルになっている。細かな解析目的ではなく、どんなサイトからどの

くらいアクセスしているのかを手軽に見ようとするならば、Webalizerのほうが見栄えがいい。

Webalizerは、<http://www.mrunix.net/webalizer/download.html>から入手できる。しかも、各種プラットフォーム用のバイナリもあるので、バイナリを入手してインストールしてしまうと手軽だ。Red Hat系ならば、Linux ELF - Intel x86(Red Hat 5.x X ftp://ftp.mrunix.net/pub/webalizer/webalizer-1.30-04-linuxelf-rh50-bin.tgz) を入手し、

```
# mkdir webalizer-1.30-04
# mv webalizer-1.30-04-linuxelf-rh50-bin.tgz webalizer-1.30-04
# cd webalizer-1.30-04
# tar xvfz webalizer-1.30-04linuxelf-rh50-bin.tgz
```

の手順で展開していく。webalizerがカレントディレクトリにできているので、

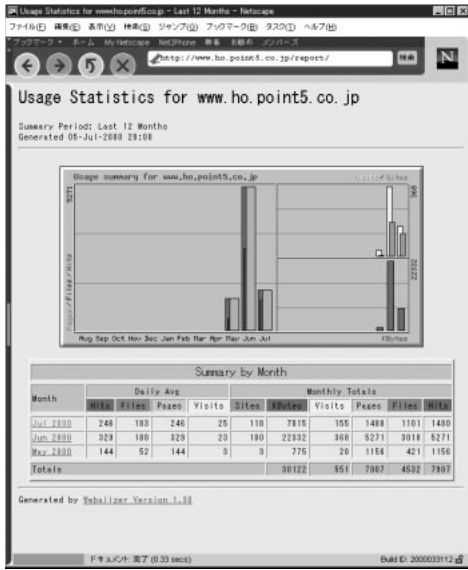
```
# mkdir /usr/local/apache/htdocs/webalizer
# ./webalizer -o /usr/local/apache/htdocs/webalizer /usr/local/apache/logs/history/access_log
```

とすることで、-oで指定したディレクトリに、ログファイル/usr/local/apache/logs/access_logを解析した結果が出力される。今回は、リスト1-1のlogcut.shによって、/usr/local/apache/logs/history/に過去のログが保存されているので、webalizerに解析させるログファイルを/usr/local/apache/logs/history/access_logを指定している。これにより、常に切り出した直後のログを解析できていることになる。

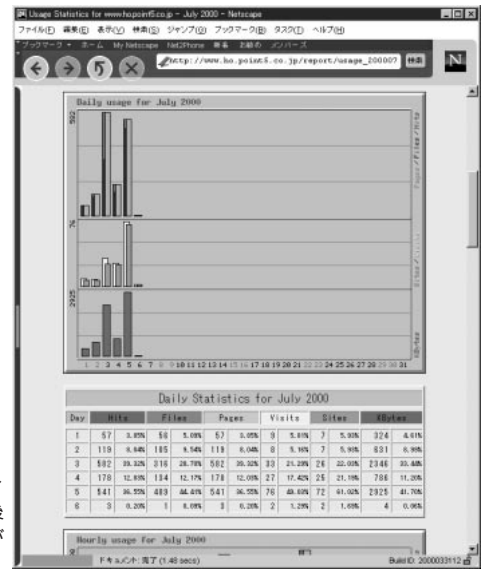
リスト1-1 logcut.sh

```
#!/bin/sh
basedir='/usr/local/apache/logs'
apachectl='/usr/local/apache/bin/apachectl'
mv='/bin/mv'

$mv $basedir/access_log $basedir/history/
$mv $basedir/error_log $basedir/history/
$apachectl restart
```



画面1 Webalizerのレポート画面
月単位でアクセス数がわかる。



画面2 Webalizerの特定月のレポート
日ごとのヒット数とアクセス者数、サイト数がわかる。このレポートページの前後には、ドメイン単位でのアクセス元などが求められている。

解析した結果は、Webブラウザで見ることができる(画面1、画面2)。この場合は、ドキュメントディレクトリが/usr/local/apache/htdocsなので、URLとしてhttp://www.ho.point5.co.jp/webalizer/というように、/webalizer/を指定する。ただし、こうすると自分で閲覧するだけに限らず、ほかの人からもアクセスできてしまうので、/usr/local/apache/htdocs/webalizer/のディレクトリに.htaccessを置いて、内部ネットワークからしかアクセスできないようにアクセス制限を行う仕組みにしておく必要がある。

なお、webalizerはグラフィックの作成にgdというグラフィックライブラリを利用している。うまく画像が出力できないときは、gdをインストールしておくといい。gdは、http://www.boutell.com/gd/http/gd-1.8.3.tar.gzで入手できる。インストール手順は、以下の通りである。

```
# tar xvfz gd-1.8.3.tar.gz
# cd gd-1.8.3
# make
# make install
```

正しく画像が出力されたとして、ここでひとつ落とし穴がある。たとえば週に1回ログファイルを切り出している場合、Webalizerで解析した段階で、前回の解析結果が最新のログファイルを解析した結果に置き換わる。つまり、過去の集計分は保持されない。そこで、ログファイルを切り出して解析していく場合、過去の記録を保持する設定にして、アクセスログ解析結果に、最新のログ情報を追加して表示するようにする。

Webalizerを展開したディレクトリ中には、sample.confというファイルが存在している。sample.confを/etc/webalizer.confにmvまたはcpする。これが最初の作業だ。次に/etc/webalizer.confを開き、Incrementalという設定行を探し出す。Incrementalは、過

去の記録に最新のログ記録を追加するかどうかという設定で、この行のコメントを外して、yesにすることで、過去の記録を保持したまま新しいログ分の記録を上書きする設定となる。該当行は次のようになるはずだ。

```
Incremental yes
```

これで、毎回切り出したログファイルをWebalizerを通して解析する仕組みができあがる。

ところで、historyディレクトリに移動したログファイルは、頻繁に利用するものではない。ディスク容量を有効に使うためにgzipで圧縮したり、テープやCD-Rなどにバックアップして削除するといい。Webalizerでログ解析を

```
リスト1-2 logcutnew.sh

webalizer='/usr/local/webalizer'
outdir='/usr/local/apache/htdocs/webalizer'
logdir='/usr/local/apache/logs/history/access_log'
extdate='date date +%y%m%d'
gzip='/usr/bin/gzip'

$webalizer -o $outdir $logdir
$mv $logdir $logdir.$extdate
$gzip -9 $logdir.$extdate
```

してから圧縮処理するまでの一連のスク립トは、リスト1-2のようなものだ。これをリスト1-1の行末に加えておくと、ログの切り出しから解析、圧縮までのひと通りの作業が実行される。

あとは、ファイルシステムの空き容量を見ながら、1カ月とか、半年に一度というタイミングで、圧縮した過去ログを定期的にバックアップして削除すればいい。

帯域の監視も管理者の重要な仕事

Webマスターの仕事は、アクセス数の把握だけではない。アクセス数と同様に大切なものは、Webサーバがどれだけのネットワーク帯域を利用しているかを認識しておくことだ。必ずしもWebマスター=ネットワーク管理者ではないので、全体のネットワーク帯域内でWebサーバがどれだけの帯域を使っているかどうかまではわからないことだろう。しかし、Webサーバが使っている帯域を知っていて損はない。

たとえば、アクセスがほとんどないような時間帯に、自分の管理するWebサーバにアクセスしてもレスポンスが遅いというようなことがあったとする。Webサーバが利用している帯域を知っていれば、それはサーバまでの経路の問題なのか、Webサーバへアクセスが集中していて接続回線帯域を使い切っているのかを判断できる。

MRTGを使ったWebサーバの帯域監視

サーバのネットワーク状態を監視するツールとしてMRTG (Multi Router Traffic Grapher) がある。MRTG自体はメジャーなソフトで、さまざまな場所でこのソフトは使われている。

SNMP (Simple Network Manage

ment Protocol) を使ってネットワーク負荷をグラフ化するMRTGを使い、Webサーバの公開側につながっているNICカードのトラフィック量を見られるようにしておくと、Webサーバがどのくらいの帯域を利用しているか、ある程度判定できる。

必要なソフトは、SNMPサーバと、MRTGだ。SNMPサーバはucd-snmptを使う。これは、<http://ucd-snmpt.ucdavis.edu/>から入手できる。なお、最近のRed Hat系のディストリビューションには、ucd-snmptがRPMパッケージで用意されているので、最新バージョンではないが、それを利用するのもいいだろう。

また、MRTGは<http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/pub/mrtg-2.8.12.tar.gz>を参照してほしい。

ucd-snmptのインストールは、展開してからconfigureをして、make、

make installをするだけだ (リスト2-1)。make installがうまくいけば、snmptデーモンが/usr/local/sbin以下に作られるので、

```
# /usr/local/sbin/snmpt
```

としてsnmptを起動する。動作確認が済んだら、/etc/rc.d/rc.localの最後にsnmptを動かすようにコマンド行を加えておくといい。

ucd-snmptと比べてちょっと設定がややこしくなるものがMRTGだ。MRTGのセットアップは、リスト2-2のように行う。

configureをするときにはgdが必要となる。もしそうしたメッセージが出るようなら、Webalizerのときと同じように、gdを組み込んでおく。

もう一点注意する部分は、makeのあとの作業だ。makeまでは他のソフトウェアと同じような手順となるが、

リスト2-1 ucd-snmptのインストール

```
# tar xvfz ucd-snmpt-4.1.2.tar.gz
# cd ucd-snmpt-4.1.2
# ./configure
(システムコンタクト情報を入力)
# make
# make install
```

リスト2-2 MRTGのインストール

```
# tar xvfz mrtg-2.8.12.tar.gz
# cd mrtg-2.8.12
# ./configure
# make
# cp -r run /usr/local/mrtg
# cd /usr/local/mrtg
# ./cfgmaker public@www.ho.point5.co.jp >mrtg.cfg
# vi mrtg.cfg
(WorkDir: /usr/local/apache/htdocs/mrtgを追加)
# mkdir /usr/local/apache/htdocs/mrtg
# ./mrtg mrtg.cfg
# ./mrtg mrtg.cfg
# ./mrtg mrtg.cfg
```

3回繰り返す

makeが終わった段階で、カレントディレクトリにrunという名前バイナリが出力される。それをそっくり/usr/local/mrtgに移動したり、複写したあとに、各種の設定をする。

MRTGの設定といっても、cfgmakerコマンドを使い、監視するホスト名を指定するだけだ。ここでは、public@www.ho.point5.co.jpというようにコミュニティ名@ホスト名でSNMPサーバにアクセスする先を指定する。適宜、自分の環境に合わせてホスト名を指定するといひ。コミュニティ名は、セキュリティのための識別子で、32文字までの任意の英数字を指定する。publicは、snmpdのデフォルトの設定だ。

デフォルトのままでは危険だと思う場合は、snmpdのソースディレクトリ中に、EXAMPLE.confがあるので、これを/usr/local/share/snmpd.confにcpし、com2secで指定しているコミュニティ名を変更する。それに応じて先ほどのpublicの名前を変えればいひ。

話を元に戻そう。MRTGのcfgmakerを動かすと、mrtg.cfgがカレントディレクトリに作られる。ただし、このままでは正しく動作しないため、コンフィグレーションファイルの行頭に手を加える。

```
WorkDir: /usr/local/apache/htdocs/mrtg
```

というように、WorkDir:行を加えておく。MRTGは、WorkDirで指定するディレクトリに、トラフィックレポートを書き出す。そして、Webブラウザを使ってその情報を読み出す仕組みになっている。Apacheのドキュメントディレクトリ以下のどこかのディレクトリを指定しておくといひ。

正しくトラフィックレポートが出力

されるようなら、mrtgのトラフィック解析を5分ごとに実行するようにcronで指定する。rootユーザーでcrontab -eとしてエディタを起動し、

```
* /5 * * * * /usr/local/mrtg/mrtg /usr/local/mrtg/mrtg.cfg
```

と1行追加しておけば、5分ごとにMRTGによるトラフィック解析が実行される。

MRTGは、そのネットワークカードに対して、入ってくるデータ量(IN)と出て行く量(OUT)をグラフとして見られるものだ。INが緑色、OUTは青で表示される(画面3)。Webサーバの場合、OUT側のデータ量のほうが大きいはずなので、OUT側が契約しているネットワーク帯域に近くなると、帯域いっぱい利用しているということになる。テレホーダイタイム近くになり、帯域の6~7割を利用しているような場合は、閲覧者に対して満足な帯域を提供できていない。回線速度を考え直さなければいけな状態だと考えるといひ。

Apacheのサービス断を検知する

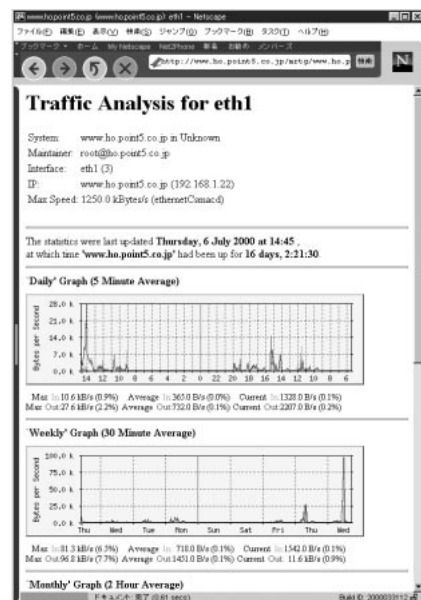
Webマスターが常に困るもののひとつが、サーバのサービス断だ。CGIプログラムやDBのクラッシュによるシステムダウンをはじめとして、ネットワーク経路のダウン、サーバのハードウェア的な故障など、さまざまな要因がWebマスターを襲う。

こうしたサーバダウンは、めったに起きることはない。起きることがないだけに、何かしらの形で常時監視して、問題が生じたら即対処できるような形にしておく必要がある。そもそもSNMPもそのひとつの方法だが、今回

はまったく別の観点から、可能な限り完全にサーバの状態を監視する方法を考えてみる。

できる限り完全な監視をするには、Webサーバを置いているネットワークとはまったく別のネットワークから、定期的に目的のWebサーバへHTTPリクエストを送るようにすればいひ。つまり、ごく普通のユーザー同様に、Webサーバにアクセスできたかどうかで、サービスが正しく動作しているかどうかを検証するのだ。

それをスクリプトにしたものが、reporter.pl(リスト3)である。これをcronで5分に一度とか、10分に一度といったタイミングで動かすと、wgetを使って指定したWebサーバにアクセスを試み、指定時間内にレスポンスが返ってくれば、サーバが正常に動作していると判断する。読み出しファイルとして、ゼロバイトのgif画像を指定しておけば、ネットワーク帯域を著しく消費することもないし、アクセス数を



画面3 MRTGのレポート

5分単位、30分単位、2時間単位、1日というように、グラフ表示される。青が出ていったバケット。緑が入ってきたバケット数。通常、Webサーバであると、青が緑より高く、青と緑がほぼ比例した波になる。この画面はテストサーバなので、かなり違った動きになっている。

いたずらに増やすこともなくなる。

あらかじめ必要なプログラムは、wget程度で、ほかはごく標準的なものを使っている。使い方は、スクリプト中の@hostという変数に、アクセス先のURLを指定しておく。timeover変数には、タイムアウトとして処理する秒数を指定する。たとえば10と指定すれば、10秒以内に返事が返ってこない場合にはサーバダウンと認識して、指定ユーザーにサーバがダウンしたというメールを送信する。

touser変数は、メールの送信先ユーザーだ。最近では携帯電話で電子メールを受信できるので、携帯電話の電子メ

ールアドレスを指定しておくといい。最後のfromuser変数は、Fromフィールドに記述するユーザー名を指定しておく。

このreporter.plというスクリプトは、基本的にどこに置いてかまわない。Webサーバと同じでもいいのだが、できればWebサーバが構築されているネットワークとは別のネットワークサーバで動かしたい。また、そのネットワークはグローバルIPの振られた、正しいネットワークであることがより望ましい。ルーティング経路として、一般の閲覧者同様に、必ず外のネットワークを通して、Webサーバの外側の

Ethernetポートを通してアクセスするほうが正しい結果を返すはずだ。もしプライベートIPアドレスを振った組織内部の別サーバから動かした場合、外側のEthernetカードが不調で動かない場合や、外側のデフォルトルータの故障でパケットが届かないようなときに、正しく判定ができない。

Webマスターの仕事とは

本連載も今回で最後を迎える。自分なりのWebマスターのイメージは、しっかりとサーバが管理でき、やりたいことをサーバ上で実現する商売だと思っている。

セキュリティも守りつつ、また、24時間、365日のあいだ運用できるべきだと思っている。会社が長期の休暇だからと、Webサーバも電源を落としてしまうようでは、Webマスターの仕事とはいえない。

インターネットという存在がクローズアップされ始めたころ、誰もがWebサーバ自体がITの中心的な存在になるとは思っていなかったに違いない。それが、今や電子メールとならんで代表的なインターネットサービスのひとつとなってしまった。もはや、Webマスターはシステム全体のrootが片手間に管理できる商売ではなくなったような気がする。そういう意味では、今求められているWebマスターは、コンテンツ面も理解して、CGIも書き、サーバも構築するという、Apacheサーバの管理が得意な管理者というイメージだろうか。

ところが、現実にはそういったWebマスターは少ない。少しでも多くの人々が、そうしたWebマスターに近づいてくれることを願っている。

リスト3 reporter.pl

```
#!/usr/bin/perl
@host=('http://www.ho.point5.co.jp/', 'http://gw.ho.point5.co.jp/');
$timeover=1;
$touser='kkuge@point5.co.jp';
$fromuser='SERVER_REPORTER<kkuge@point5.co.jp>';
$sendmail='/usr/sbin/sendmail';
$date='/bin/date';
$wget='/usr/local/bin/wget';

foreach (@host) {
    &downcheck("$_")
}
exit;

sub downcheck {
    local $rtncode=system ("$wget -q -O /dev/null -t 1 -T $timeover $_[0]");
    if ($rtncode) {
        &callmail ($_[0]);
    }
}

sub callmail {
    local $time=`$date +%H:%M`;
    chomp $time;
    open (SENDMAIL,"| $sendmail -t");
    print SENDMAIL<<MSGEND;

From: $fromuser
To: $touser
Subject: SERVER FAILED ($time

CAN'T CONNECT TO $_[0] AT $time
MSGEND

    close SENDMAIL;
}
```

Linux ファイルシステムの現在と未来

次世代ファイルシステムのひとつ、ReiserFSの名前を最近よく聞くようになってきた。SuSE以外にも、いくつかの最新ディストリビューションで利用可能になってきている。そこで今回は現時点のReiserFSの性能を検証してみた。

最終回 ReiserFSを使ってみる

文：長岡モイチ

Text : Moichi Nagaoka

ReiserFSとは

第1回でも少し触れたと思うが、ReiserFSはHans Reiser氏が中心となって開発を進めているLinuxのファイルシステムだ。ReiserFSの開発にはSuSEがメインスポンサーとしてついており、同社のSuSE Linuxでは早くからReiserFSがサポートされている。

またReiserFSの機能ごとに異なるスポンサーがついており、現在のところジャーナリングの部分にはmp3.comが、squid cacheの最適化にはintegratedlinux.comが、HSM (Hierarchical Storage Manager)の部分にはBigStorageがスポンサーとして名前を連ねている。ホームページは<http://devlinux.com/projects/reiserfs/>なので、興味のある人はアクセスしてみたい。

ReiserFSはファイルシステムの構造に全面的にバランス木を用いた革新的なファイルシステムであり、現在も開発途中である。しかしながら、パフォーマンスはすでにext2に代表される固

定ブロックサイズ構造のファイルシステムに匹敵するといわれている。また、ReiserFSにはジャーナリング機能も備わっており、システムクラッシュした際のfsck時間が大幅に短縮できることも売りのひとつだ。

最近ReiserFSをサポートするディストリビューションも増えてきて、ReiserFSも実用レベルまで達してきているといえよう。ReiserFSの名前は雑誌、Webページ、ネットニュースではよく聞くけれど、実際のところどうかと興味のある読者も多いと思う。そこで、今回は現時点のReiserFSの機能、性能について焦点を当て、実際にどのくらい使えるか検証してみたいと思う。

ところで、ReiserFSは現在、Linuxカーネルに対するパッチとして配布されている。したがって、ReiserFSを使用したい場合はReiserFSをサポートしているディストリビューションを選択するか、自分でカーネルソースにパッチを当ててコンパイルするしかない。今回の検証には、本誌8月号付録のフ

ランス製LinuxディストリビューションのMandrake 7.1の環境を使用した。ReiserFSのバージョンは3.5.19となる。なお検証環境として、ABIT BP6のDual Celeron 466MHzにIBM製25GBバイトIDEハードディスクをUltraDMA/33のインターフェイスに接続してテストを行った。

ReiserFSのI/Oパフォーマンス

まずファイルシステムの基本性能を調べるために、ベンチマークソフトのbonnie++を実行してみた。結果は表1にまとめたので詳細はそちらを参照して欲しい。このテストでは、200Mバイトのファイルに対するバイト単位、ブロック単位でのシーケンシャルリード/ライト、ランダムシークを行っている。また、ディレクトリ操作に関するテストも行っており、ディレクトリに50K (50 × 1024) 個のファイルを作成した場合の性能も計っている。

bonnie++テストの結果からは、現時点のReiserFSはext2と比較してファ

ファイルI/Oに関してはほぼ同等の性能が出ていると思われる。そして、ディレクトリ操作のテストではすべての項目でext2を大きく引き離し、ReiserFSのパフォーマンスの高さを表している。ディレクトリへのファイル作成の速度は約50倍、ファイル名の検索は約100倍という違いは、実際に運用するにあたってかなり大きいと思う。たとえば、ディレクトリ内に大量のファイルを作成するメールサーバやニュースサーバの、大幅なパフォーマンスアップにつながることを期待できる。ただし、bonnie++のテストではmkfsした直後の状態で行ったので、ディスク消費が進んだ状態でも結果が同じかは正直不安なところではある。

また、bonnie++のディレクトリ操作テストではReiserFSのパフォーマンスの高さが結果として表れたが、この結果はReiserFSが採用しているバランス木構造がファイル名の検索に大きく貢献していると思われる。bonnie++のディレクトリ操作テストで作成される各ファイルのサイズは0バイトである（つまり、creatしただけでwriteしていない）が、ReiserFSではディレクトリエントリ、i-nodeデータ、ファイルデータなどのファイルオブジェクトはすべてバランス木上のitemとして組み込まれているので、一般的な環境、たとえばニュースサーバのようにファイル

サイズが1Kバイト~10Kバイト程度のファイルが、ひとつのディレクトリに大量に作成される場合は結果が異なる可能性がある。

そこで、ファイルシステムの消費量増加に伴う速度低下があるか調べてみた。テスト方法として、Mandrakeの/usr/src/linux-2.2.15以下、つまりカーネルのソースツリーまるごとを新規作成したサブディレクトリにコピーする処理を行わせてみた。テスト方法はそれぞれ1Gバイトのパーティションに、77Mバイトのカーネルソースツリーを10個の異なるサブディレクトリにコピーするというものである。ディスクの消費量は約770Mバイトになるはずであり、1Gバイトのパーティションのうち80パーセント弱を占有することになる。

テストの結果は、ext2の7分35秒に対し、ReiserFSが7分25秒となり、ext2とReiserFSでは大きな差がつかなかった。bonnie++のベンチマークの結果とあわせると、ファイルI/Oに関しては通常の使用においてはReiserFSはext2と比べても遜色がないといえるだろう。

あとはReiserFSを実際に使用する前に運用時のディスクI/Oパターンと類似の負荷をかけ性能劣化がないかチェックして、大丈夫ならばReiserFSを使うことができるだろう。

mkfsとfsckの時間

大容量ディスクが安い最近では、1パーティションが10Gバイトを超えるような構成で使いたいと思う読者もいることだろう。特にマルチメディア系の用途やVMwareの仮想ディスク用など、大容量パーティションの需要は増えている。ところで実際に大容量パーティションを使用すると感じるのが、mkfsとfsckの時間の長さだ。

25Gバイトのハードディスクをext2でmkfsした場合、1分40秒近くかかってしまった。fsckはファイルシステムが空の状態でも1分34秒かかった（テストにはUltraDMA/33のインターフェイスを使い、hdparmでDMAなどのパラメータをチューニング済みである）。

mkfsは当然パーティションサイズが大きければそれだけ時間がかかるし、fsckはファイルシステムの消費が進めばそれだけ時間がかかるので、十分消費しているファイルシステムをfsckすると5分以上かかることは大いにあり得る。

つまり、大容量ディスクを細かなパーティションに分割して使用する場合は問題がないが、ひとつの巨大なパーティションとして使用する場合は、ext2ではだんだんつらくなってきている状況である。ディスクの大容量化、低価格化は今後も進み、サーバ用途では

ファイルI/Oの結果			シーケンシャル書き込み				シーケンシャル読み込み				ランダムシーク		
	バイト単位		ブロック単位		再書き込み		バイト単位		ブロック単位				
転送量 (MB)	K/sec	%CP	K/sec	%CP	K/sec	%CP	K/sec	%CP	K/sec	%CP	K/sec	%CP	
ext2	200	4206	77	5294	12	2048	36	3493	88	4523	77	23.0	1
ReiserFS	200	4128	99	5258	36	1937	33	3707	94	4440	69	22.7	1

ディレクトリ操作の結果				シーケンシャル作成				ランダム作成					
	Create		Read		Delete		Create		Read		Delete		
ファイル数	/sed	%CP	/sed	%CP	/sed	%CP	/sed	%CP	/sed	%CP	/sed	%CP	
ext2	50K	86	99	261	100	3523	98	87	99	283	99	144	53
ReiserFS	50K	4364	100	29451	100	6434	100	4365	100	27827	100	2293	58

表1 bonnie++ベンチマーク結果

K/sec = 1秒あたりの転送量(Kバイト)。数値が大きいほど性能が高い /sed = 1秒あたりの回数。数値が大きいほど性能が高い %CP = CPU使用量。数値が低いほどよい

ext2では正直やってられない状況のほ
 ず。ReiserFSは、そんな問題に対する
 ひとつの答えになるのだろうか。mkfs
 とfsckについて調べてみた。

mkreiserfsによる ファイルシステム作成

ReiserFSはmkreiserfsコマンドによ
 ってファイルシステムを作成する。そ
 こで、25Gバイトのハードディスクを1
 パーティションとしてmkreiserfsを実
 行してみた。mkreiserfsを実行すると
 ファイルシステムの管理領域の作成は
 2、3秒で終了し、あとはsyncを行っ
 ているメッセージが出力される。

結果、なんと25Gバイトのファイル
 システム作成にかかった時間はたった
 の10秒であった。いろいろパーティシ
 ョンのサイズを変更して試してみたこ
 ころ、すべて10秒前後で完了した。ど
 うやらReiserFSはlog領域ほかの32M
 バイト程のファイルシステム管理領域
 以外はタッチしていないようだ。

ext2の場合はブロックグループに分
 け、ブロックグループごとに管理領域
 を書き込む必要があるので、ファイル
 システムのサイズが大きくなればそれ
 だけ時間がかかるが、ReiserFSでは最
 初に予約する32Mバイトだけでよいら
 しい。これはファイルシステム内のオブ
 ジェクトを、バランス木を用いて管理
 するReiserFSならではのアドバンテ
 ージではないかと思われる。

ちなみに、管理領域として32Mバイ
 ト必要とするということは、100Mバイ
 トほどのファイルシステムではデータ
 領域として使用できる割合が低くなる
 ので、好ましくないということもある。
 ただし、ReiserFSはext2のように
 固定ブロックサイズでデータを管理し
 ていないので、小さなファイルを大量
 に作成する場合は、ファイルシステム

のフラグメンテーションの関係でext2
 よりも多く作成できる可能性もある。
 小さなファイルを作成した場合の未使
 用領域の割合については、あとで検証
 してみよう。

ジャーナリングの効果

さて、次はfsckだ。ext2ではきちん
 とshutdownを行えばファイルシステム
 がクリーンな状態になるので、ブート
 時のfsckは行われない。しかし、何回
 もブートしていたり前回のブートから
 日にちが経過している場合は、fsckが
 走ってしまう。

ReiserFSはジャーナリングファイル
 システムである。大容量ディスクを使
 っている読者ならば誰もがブート時の
 fsck やmountにいらいらしていること
 と思うが、ReiserFSを使えばジャーナ
 リングのおかげでそんな悩みはなくな
 るはず。ということで、/usr/src/
 linux-2.2.15のカーネルのソースツリー
 をコピーさせ、コピーしている最中に
 リセットボタンを押すという荒業に対
 してReiserFSは期待に応えてくれるか
 テストを行ってみた。

結果は期待通り、ReiserFSのmount
 はほぼ10秒で完了した。このときリプ
 レイしたトランザクションは1つだけ
 であった。またfsckの時間は常に0秒
 である。ReiserFSにはfsckは必要なく、
 mount時に実行されるlogのリプレイ
 だけだからだ。実際、Mandrakeの
 fsck.reiserfsは/bin/trueのシンボリック
 リンクであり、fsck時には何も処理
 を行っていない。ちなみに、ReiserFS
 はmount時にlog領域に残っているトラ
 ンザクションの数と処理した時間を出
 力するので、dmesgなどで確認するこ
 とができる。

テストの結果から、ReiserFSのジャー

ナリング機能は期待通りであり、大
 容量パーティションには不可欠である
 といえるほどその効果を発揮してくれ
 た。ふだんfsckのプログレスバーにい
 らいらしている人ならば、ジャーナリ
 グを使用したらきっと感動するはずだ。

ディスク使用率

先ほど少し触れた、小さなファイル
 を大量に作成した場合のディスクの使
 用率について調べてみよう。まずddで
 512バイトのファイルを100個作成した
 ときのディスク使用率から求めてみた。
 ext2の場合、使用率についてはブロッ
 クサイズから割り出せる。デフォルト
 の4Kバイトブロックの場合は1/8であ
 り、1Kバイトブロックで1/2となる。
 ReiserFSの場合、実際にテストしたと
 ころ64Kバイトの増加となり、ディスク
 使用率はおよそ78%の好成績となっ
 た。

次に、5Kバイトのddの場合を考え
 る。4Kバイトブロックのext2では、デ
 ィスク使用率は5/8となり、固定ブロッ
 クサイズのファイルシステムには不利
 な状況だ。対するReiserFSは、これ
 も同じく約5/8となった。

また、カーネルソースをコピーした
 あとのdfを見比べてみると、ext2は766M
 バイト消費したのに対し、ReiserFSは
 予約領域32Mバイトを足しても712M
 バイトしか消費していなかった。

これらのテスト結果から、ReiserFS
 のディスク使用率はext2よりも良いと
 いうことがいえるだろう。

期待を抱かせる完成度

さて、ひと通りReiserFSを検証して
 みて思った感想は、「結構使えるかも」
 という期待感だった。従来の固定ブロッ
 クサイズではなく、バランス木を用

いた新しいファイルシステムということで安定性について不安があったが、テストで使用してみた限りではそんな感じはなく、ext2と同じような感覚で使用できた。

また、ジャーナリングの機能は大容量ディスクユーザーにとっては大変魅力的であり、これだけでもext2からReiserFSに移行する価値はあるかもしれない。

mkfsが10秒でできることも用途によっては有難いだろう。たとえば大容量パーティションをテンポラリに使う場合などは、ファイルをrmする操作よりもmkfsし直したほうが早いだろう。ほかにもいろいろ使い方がありそうだ。

このようにテストした限りにおいては、ReiserFSにはルートファイルシステムとしては使用できないという機能上の弱点以外には、性能上の目立った弱点は見つからなかった。もちろん、長い間実際に使用しているわけではないので、運用して判明する問題点もあるだろう。しかしながら、テストの結果、またテスト中にファイル操作をした経験からは、ReiserFSは安定しているといえる。ジャーナリング機能を持たせたいディストリビューションは、今後ReiserFSを積極的に採用していきだろう。

ReiserFS、そしてLinux ファイルシステムの将来

ReiserFSはLinuxの次世代ファイルシステムのひとつとして、また現時点では一般ユーザーが気軽に使用できる唯一のジャーナリングファイルシステムとして位置付けられる。このファイルシステムは2.3カーネルのときにすでにレベルの内容だったため、当初Linux2.4に加わると思われていたが、紆余曲折があり統合は開発版カーネルである2.5

のツリーにまで遅れる見込みだ。

話は変わるが、最近のlinux-kernelメーリングリストでReiserFSの開発者であるHans Reiser氏の投稿から始まった長いFlame Warがあった。このFlame Warが現在のReiserFSの状況を少なからず表していると思う。

ここではその内容は詳しく説明しないが、話の発端はLinux 2.4にReiserFSが含まれないことに対するReiser氏の不満が爆発し、件の投稿となった。Reiser氏の不満はAlan Cox氏にぶつけられ、以降ext3の作者などたくさんの人を巻き込む大きなスレッドとなった。興味のある人はhttp://kt.linuxcare.com/kernel-traffic/kt20000710_75.ep1#1を読んでみて欲しい。

筆者はふだんこのメーリングリストは斜め読みしかしないのだが、話題がタイムリーだったためこのスレッドだけはまじめに読んでみた。ここで繰り広げられた各人の感情的な内容はさておき、Linuxファイルシステムの将来像に関する技術的な話も話題に上ったので、ReiserFSの立場も含めて少し紹介したい。

次世代Linuxファイルシステムは、XFSやJFSなどのベンダーの移植プロジェクトによるものを除くとReiserFSとext3の2種類が有名だ。ext3はext2を拡張しジャーナリング機能も備えたファイルシステムであり、よくReiserFSと対比されて取り上げられる。スレッドで出てきた話から、どうやらAlan Cox氏とext3の開発者であるStephen C. Tweedie氏は、Linuxカーネルのファイルシステムにジャーナリング用の共通レイヤを実装しようとしているらしい(チェックしてないが実は前から議論はあったかもしれない)。

ジャーナリングはカーネルの仮想メモリサブシステム、特にバッファキャ

ッシュサブシステムと密接な関係を持つ必要がある。Alan Cox氏らはジャーナリングファイルシステムとバッファキャッシュのあいだに共通したインターフェイスを実装することを考えている。

一方、ReiserFSのコードは現在のカーネルアーキテクチャと密接に結びついている。したがって、Alan Cox氏はReiserFSを2.4カーネルに統合するよりも、まず共通レイヤをカーネルに実装してからReiserFSにこれを使う実装にコード変更するほうがよいと思ったようだ。

話の流れから推測すると、ジャーナリング用の共通レイヤはext3のジャーナリングレイヤがもともになるようだ。この話が素直に発展していくとなると、次世代ファイルシステムはすべてこのジャーナリングレイヤを使用することになる。また、ジャーナリングファイルシステムとしてext3がもっとも早くLinuxカーネルツリーに統合される可能性もある。もっとも、ext3はいまだバージョン0.2なので、この推測に関しては自信がない。

この政治的ともいえる理由により、ReiserFSは現在の完成度とは裏腹にLinuxカーネルに統合されるのは少し後になりそうだ。さらに今回のFlame War事件により、ReiserFS側はLinux開発者コミュニティに対して大きな傷跡を残してしまったと思われることが心配だ。

以上、ReiserFSを通してLinuxファイルシステムの今後について少し話をさせてもらったが、いちLinuxユーザーとしては、ReiserFSをはじめとする次世代Linuxファイルシステムが早くカーネルツリーに統合され、ext2が今のminixファイルシステムのように過去のものになることを心から願っている。

賢く使うUNIX

これであなたもスマートなUNIX使い！

プロセスの表示と制御

Linuxの標準シェルであるbashのコマンドラインを中心として、Linuxの便利な使い方について紹介していく本連載。今回は、実行中のプログラムの実体であるプロセスについて取り上げ、ps / top / killなどのコマンドやbashが備えるジョブ制御機能について説明した後、対話的にプロセスの終了を行うスクリプトを作成する。

今月のお題

指定した文字列をコマンドに含むプロセスを選択して終了するスクリプトを作成する

「プロセス」とは、メモリに読み込まれて実行可能な状態にあるプログラムのことだ。このように、メモリ上のプログラムにわざわざ名前を付け、ファイルシステム上の実行ファイルと区別しているのは、Linuxを含むUNIX系OSが複数のプログラムを同時に実行できるマルチタスクOSであるからにほかならない。

UNIX系OSでは、絶えず複数のプロセスが同時に実行されている。一般ユーザーでも、コマンドラインの最後に「&」をつけることで、指定したコマンドのプロセスをバックグラウンドで実行できる。また、パイプ「|」を使って複数のコマンドを接続した場合には、各コマンドのプロセスが同時に実行される。

各プロセスには、それぞれ「プロセスID」と呼ばれる数値が割り振られている。このプロセスIDは、プロセスの停止や終了を行う際に、対象となるプロセスの特定に使われる。同じコマンドを複数実行していても、それぞれのプロセスIDが異なるため区別できるのだ。

プロセスの一覧を表示するpsやtop、プロセスの停止・終了などを指示するkillといったコマンドを使うと、無反応になったプロセスを安全に終了したり、指定したプロセスを停止・再開したりできる。さらに、bashなどのシェルに



Illustration : Manami Kato

文：大池 浩一
Text : Kouichi Ooike

は「ジョブ制御」と呼ばれる機能が用意されており、一連の仕事をしている複数のプロセスをまとめて扱うことが可能だ。なお、こうしたコンソールベースでの操作は、telnetなどを利用して接続した外部端末からでも行えるため、たとえLinuxを動かしているパソコンのキーボードが効かなくなったり、画面に何も表示されなくなった場合でも復旧が可能だ。

前半では、プロセスを表示するps / topの使い方や、bashのジョブ制御、プロセスを停止・終了させるkillの使い方を説明しながら、プロセスの状態やシグナルによる制御について簡単に触れる。後半の「今月のお題」では、指定した文字列をコマンド名に含むプロセスを一覧表示し、ユーザーが対話的に選択したプロセスだけを終了するスクリプトを作成する。

なお、今回はあくまでユーザーからの視点でプロセスの表示や終了方法などを説明している。プロセスの生成や終了、実行に関するシステム的な話は、本誌2000年1、2月号の「Linux日記」で行われている。プロセスの状態遷移やシグナルによる割り込みなどの詳細を知りたい方はそちらを参照されたい。

プロセス情報を一覧表示する ps

UNIX系OSでは、ユーザーが実行したコマンド以外にも、各種デーモン（サーバーなど）や、cronシステム（本誌1999年12月号の本連載を参照）により定時に起動されるコマンド、Xウィンドウ上で動作するウィンドウマネージャなど、さまざまなプロセスが同時に実行されている。

こうしたプロセスのプロセスIDやコマンド名などの情報を出力するコマンドとして、psやtopが用意されている。まずは基本的なpsから説明しよう。Linuxのpsには3系統のオプションが混在しており（コラム参照）、以下ではBSD形式のオプションを利用する。あらかじめ、

```
$ export PS_PERSONALITY=bsd
```

としておこう（これを忘れると、psの出力する情報が異なったものになる）。

オプションを付けずにpsを実行すると、現在の端末でユーザーが実行したプロセスの一覧が表示される。

```
$ ps
  PID TTY          STAT       TIME COMMAND
 1027 ttyp0        S            0:00 bash
 1167 ttyp0        R            0:00 ps
```

先頭行はヘッダで、2行目以降が実際のプロセスの情報だ。左から順に、プロセスID（PID）、制御端末名（TTY）、プロセスの状態（STAT）、消費したCPU時間（TIME）、コマンド名（COMMAND）が表示される。

psのオプションとしては以下の3つがよく使われる。これらを組み合わせて使うことも可能だ。

a...他のユーザーのプロセスも表示

u...ユーザー名や開始時間などの情報を表示

x...他の制御端末で実行中のプロセスや、制御端末を持たない（デーモンなどの）プロセスも表示

たとえば、「ps x」とすると、そのユーザーが所有する全プロセスが表示される。さらに詳しい情報を得るには「ps ux」とすればいい。同様に、「ps ax」とすると、他のユーザー所有のプロセスも含めてすべてのプロセスが表示され、「ps aux」ではさらに詳しい情報が得られる。

なお、psの出力は、端末の桁数を超える部分を自動的に切り捨ててしまう。各プロセスの長いコマンドラインを正確に知りたい場合は、wオプションを2つ指定して、

```
$ ps auxww
```

とする必要がある。

数多くのプロセス情報が出力されて、目的のプロセスを見つけにくい場合は、psの出力をパイプでgrepに接続し、コマンド名などで検索すればいい。たとえば、

```
$ ps auxww | grep bash
```

とすると、「bash」を含む行だけが出力される。なお、そのままだと検索を行うプロセスも検索結果に含まれてしまう。この問題を避けるには、

```
$ ps auxww | grep '[ -]bash'
```

のように検索パターンを少し変更して、「bash」か「-bash」を検索するようにすればいい。

Column

psのオプションについて

初心者ユーザーにとって面倒なことに、Linuxで使われているps（バージョン2以降）には、3系統のオプションが混在している。そのうち2つは、UNIXの2つの系統「BSD」と「System V」にそれぞれ由来する。両方で使われているpsは、オプションや表示内容に互換性がないのだ。

慣れ親しんだオプションを忘れて新しいものを覚えるのは大変なので、Linuxのpsではオプションの前に「-」があるかどうかで、BSD形式とSystem V形式のオプションを区別するように作られている。「ps aux」のように「-」を付けない場合はBSD形式として扱われ、「ps -ef」のように「-」を付けた場合はSystem V（正確にはUNIX98スタンダード）形式として扱われるのだ。もし、「BSD形式のオプションしか使わない」

という場合には、あらかじめ「export PS_PERSONALITY=bsd」としておくといい。オプションを付けない場合の表示や、「-」を付けたオプションもすべてBSD形式として扱われるようになる。

ちなみに、残りのオプションの系統は、「ps --version」のように「--」に続けて指定するGNU形式の長いオプションで、ふだん使うことはほとんどない。

プロセス情報をリアルタイムに表示する top

psには各フィールドの値に基づいて出力をソートするオプションも用意されており、消費したCPU時間の多い順などでも表示できる。しかし、実際にはこれらのオプションを使う機会はほとんどない。別のコマンドtopを使うと、同じことが簡単に行えるからだ。

topは、プロセスの情報をリアルタイムに表示するコマンドだ(画面1)。上部には、システムの負荷を表わす「ロードアベレージ」や状態別のプロセス数、CPU・メモリ・スワップの状態といったシステム情報が表示され、その下に各プロセスの詳しい情報が表示される。また、これらの情報は4秒ごとに表示が更新され、プロセス情報はCPU使用率(%CPU)の高い順にソートされる。表示間隔やソートの基準となる項目は変更可能だ。

表示されるプロセス情報は、psをuオプション付きで実行した場合と似ている。左から順に、プロセスID(PID)、所有ユーザー名(USER)、優先度(PRI)、nice値(NI)、合計サイズ(SIZE)、物理メモリ消費量(RSS)、共有メモリ量(SHARE)、状態(STATE)、ライブラリページ数(LIB)、CPU使用率(%CPU)、物理メモリ使用率(%MEM)、消費したCPU時間(TIME)、コマンド名(COMMAND)だ。

通常、topは「対話的実行モード」で実行されるので、さまざまな設定をキー操作で変更できる(表1)。たとえば、最初は全プロセスが一覧表示されているが、これを特定のユーザーのプロセスだけに変更するには、uキーを押してユーザー名を入力すればいい。

また、メモリを大量に消費しているプロセスを探すには、Mキー(大文字)を押して物理メモリ使用率の高い順にソートする。各プロセスのコマンドラインを知りたい場合は、

画面1 数秒ごとにプロセス情報が更新される

cキーを押してコマンド名のみの表示からコマンドライン表示に切り替えよう。後述のkillと同様に、問題のあるプロセスを停止・終了させることも可能だ。

このほか、fキーを押すと、画面に表示される項目を対話的に追加・削除できるし、oキーで表示順序を変更することもできる。変更した設定を次回も有効にしたい場合は、Wキー(大文字)を押して、現在の設定をファイル(~/toprc)に保存しておくといい。

psとtopはどちらも一長一短があるので、grepにより検索するなど、プロセス情報をさらに加工する場合はps、プロセス情報をリアルタイムに眺めたり、ソートしたりしたい場合はtopといった具合に、要求に応じて両者を使い分けるとよいだろう。また、最近では「gPS」や「qps」など、Xウィンドウ上でtopと同様の情報を表示するアプリケーションも何種類か作られている。これらを使うと、マウス操作によりプロセス情報のソートや、プロセスの停止・終了処理などが可能だ。

プロセスとジョブ

bashなどのシェルでは、コマンドラインの最後に「&」をつけることで、指定したコマンドのプロセスをバックグラウンドジョブとして実行できる。「ジョブ」とは、簡単にいえば、シェル上で実行されたコマンドラインにより生成されるプロセスのグループのことだ。

スペース	情報を即時更新
Ctrl-L	画面の再描画
h/?	ヘルプを表示
k	プロセスをkill(終了・停止など)
i	アイドル・ゾンビプロセスの表示(トグル)
n/#	表示するプロセス数を指定
q	topを終了
r	プロセスをrenice(優先度変更)
S	CPU時間に対する累積モード(トグル)
s	表示間隔の秒数を指定
u	指定ユーザーの所有するプロセスに限定
f/F	フィールドの追加・削除
o/O	フィールドの並び替え
l	ロードアベレージの表示(トグル)
m	メモリ情報の表示(トグル)
t	概要情報の表示(トグル)
c	コマンドラインの表示(トグル)
N	プロセスIDでソート(数値順)
A	プロセス開始時間でソート(新しいもの優先)
P	CPU使用率でソート(高いもの優先)
M	物理メモリ使用率でソート(高いもの優先)
T	CPU時間でソート(多いもの優先)
W	現在の設定をファイル(/.toprc)に保存

表1 top実行中のキー操作

たとえば、テキストファイル hoge.txt の行数や文字数などを調べるために、「wc hoge.txt > hoge.dat」をバックグラウンドで実行すると、

```
$ wc hoge.txt > hoge.dat &
[1] 921
```

などと数字が表示された後、bashのプロンプトに戻る。大カッコ内の「1」はbashが割り当てたジョブ番号、その後の「921」はプロセスIDだ。ジョブ番号「1」は、現在シェル上で他のバックグラウンドプロセスが実行されていないことを意味する。他のジョブが実行中の場合は、2、3、...と順番にジョブ番号が割り当てられる。

その後、wcの実行が完了すると、bashは、

```
[1]+ Done wc hoge.txt > hoge.dat
$
```

のように、ジョブ番号の入ったメッセージをプロンプトの直前に表示する。

現在実行中のジョブの一覧は、組み込みコマンドのjobsで表示できる。

```
$ jobs
[1] Running emacs &
[2]- Stopped LANG=C man ps
[3]+ Running wc hoge.txt > hoge.dat &
```

OSが割り当てるプロセスIDがあるのに、bashがわざわざ別の番号を割り当てるのはなぜだろうか。ひとつは、次節で説明するkillなどのコマンドで、わかりやすい指定方法を用意するためだ。プロセスIDはシステム全体で一意でなくてはならないため、値が何桁もあつたりして結構覚えにくい。これに対し、ジョブ番号は現在シェル上で実行中のプロセスだけを区別すればよいので、1から始まる連続した値を割り当てられるのだ。

もうひとつの理由は、複数のプロセスを1つのジョブとしてまとめて扱えることだ。同じ「プロセスグループ」に属するプロセス、たとえばパイプで接続された複数のコマンドのプロセスは1つのジョブになる。killの引数としてジョブ番号を使えば、こうした複数のプロセスを一度に停止・終了できる。

bashのジョブ制御機能を使う

bashには、バックグラウンドジョブをフォアグラウンドジョブ（「&」をつけずに実行した状態）に変更したり、逆にフォアグラウンドジョブを後からバックグラウンドジョブに変更するといった、便利なジョブ制御機能が用意されている。

バックグラウンドジョブをフォアグラウンドジョブに変更するには、組み込みコマンドのfgを利用する。ジョブが1つの場合は引数なしで実行すればいい。複数のジョブがある場合は、対象となるジョブのジョブ番号を「%ジョブ番号」として引数で指定する。たとえば、先ほどのjobsの出力の状態、

```
$ fg %2
```

とすると、「LANG=C man ps」がフォアグラウンドジョブとなる。その結果、画面にpsの英語版マニュアルが表示され、キー入力を受け付けるようになる。

逆に、フォアグラウンドジョブをバックグラウンドジョブに変更するには、まずCtrl-Zキーを押してフォアグラウンドジョブを停止させる。

画面には、

```
[4]+ Stopped find -name '*.d' > list
$
```

のように、ジョブ番号とコマンドラインが表示されてbashのプロンプトに戻る。

このジョブをバックグラウンドで再開するには、組み込みコマンドのbgを利用する。ジョブを停止した直後ならば、そのまま、

```
$ bg
[4]+ find -name '*.d' > list
```

とすればいい。停止したジョブが複数ある場合は、fgと同様に、「%ジョブ番号」を引数で指定して対象となるジョブを指定する。

プロセスを停止・終了する kill

UNIX系OSでは、プロセスの停止や終了にプロセス間通信の一種である「シグナル」が使われる。シグナルの一覧は、bashの組み込みコマンドのkillを、

```
$ kill -l
```

と-lオプション付きで実行すると表示される。30種類を超えるシグナルが用意されているが、実際に使う機会があるシグナルは8種類ほどだ(表2)。

一部のシグナルは、端末で特定のキーを押すことで発生する。たとえば、「&」を付けずに実行したフォアグラウンドプロセスを終了するには、Ctrl-Cキーを押せばいい(Emacsなど一部アプリを除く)。これは、Ctrl-Cキーを押すとINTシグナルが発生し、それを受け取ったプロセスが終了するからだ。同様に、フォアグラウンドプロセスの実行を停止するには、TSTPシグナルを発生させるCtrl-Zキーを押せばいい。

一方、バックグラウンドで実行中のプロセスはこうしたキー入力を受け付けない。そこで、bashの組み込みコマンドkillを使ってプロセスに各種シグナルを送り、プロセスの停止や終了を行うという方法が用いられる。よく、「プロセスを殺す」とか「なかなか死なないプロセス」などと物騒な言いかたをするのは、このkillというコマンド名に由来する。

killの使い方は簡単で、終了したいプロセスのプロセスIDを引数に指定すればいい。たとえば、プロセスID「965」のプロセスを終了させるために、

```
$ kill 965
[4]+ Terminated          find -name '*.d' > list
```

とすると、プロセスのジョブ番号やコマンドラインを表示して終了する。ただし、一般ユーザーの場合、終了できるのは自分が所有するプロセスに限られる。このため、他のユーザーの所有するプロセスや、各種デーモンのプロセスを終了させたい場合は、スーパーユーザー(root)になって作業する必要がある。

引数には、プロセスIDだけでなく、「%」を付けたジョブ番号も指定できる。先ほどの例ならば、

```
$ kill %4
```

とすればいい。ジョブ番号のほうがプロセスIDより覚えやすいので、通常はこちらを使うことが多いだろう。また、パイプ「|」を使ったコマンドラインのように、ジョブが複数のプロセスで構成されている場合は、それらを一度に終了できるという利点もある。

通常、killはプロセスを終了させるTERMシグナルを送る。その他のシグナルを送るには、killのオプションとして「-」に続けてシグナル名(またはシグナルに対応する数字)を指定すればいい。

たとえば、プロセスを停止させるSTOPシグナルを送るには、-STOPオプションを指定して、

```
$ kill -STOP %4
[4]+ Stopped (signal)    find -name '*.d' > list
```

とする。STOPシグナルは無視できないシグナルなので、これを受け取ったプロセスは問答無用で停止する。逆に、停止したプロセスを再開させる場合は、killを-CONTオプション付きで実行して、プロセスの実行を再開させるCONTシグナルを送ればいい。この方法で、キー入力を受け付けないバックグラウンドプロセスを自由に停止・再開させることができる。

最強のKILLシグナルは最後の手段

プロセスを終了させる数種類のシグナルの中でも、TERMシグナルは弱い部類で、シグナルを「トラップ」して対応する処理を行うことができるよう設計されている。たとえ

INT	割り込み (INterrupt) Ctrl-Cキーで発生。プロセスは終了。
QUIT	終了 Ctrl-¥キーで発生。プロセスはcore + 終了。
HUP	ハングアップ (Hang UP) 端末の回線切れで発生。プロセスは終了。
TERM	ターミネート (TERMinate) killのデフォルト。プロセスは終了。
KILL	キル プロセスは終了 (無視できない)。
TSTP	端末停止 (Terminal SToP) Ctrl-Zキーで発生。プロセスは停止。
STOP	停止 プロセスは停止 (無視できない)。
CONT	再開 (CONTinue) 停止したプロセスの実行を再開。

表2 killやキー操作でよく使われるシグナル

ば、作業用の一時ファイルを削除したり、編集中のファイルを保存するなどの「後始末」を行ってから終了できるわけだ。

しかし、プロセスの状態によっては、TERMシグナルでは終了できない(なかなか「死なない」)こともある。その場合は、より強い終了シグナルである「QUITシグナル」や「HUPシグナル」を送ってみよう。

```
$ kill -QUIT %4
```

などとすればいい。たいていは、これらのシグナルを送ることでプロセスを終了できるはずだ。

それでも終了しない場合は、最後の手段としてKILLシグナルを送る。KILLシグナルは、プロセス側でトラップできないシグナルなので、(一部のシステムプロセスを除いて)どんなプロセスでも問答無用で終了できるのだ。もちろん、そのプロセスを削除する権限のある所有者か、スーパーユーザーで作業する必要がある。

```
$ kill -KILL %4
```

```
[4]+ Killed find -name '*.d' > list
```

最強のKILLシグナルを経験すると、通常のkillで使われるTERMシグナルなどを試さずに、どんな場合でもKILLシグナルを使いたくなるかも知れない。しかし、KILLシグナルを受け取ったプロセスは「後始末」なしに終了してしまうため、一時ファイルが削除されないまま残るなどの問題がある。あくまで、通常のTERMシグナルで終了できなかつ

た場合の最後の手段として使ったほうがいい。

ところで、こうした操作は、telnetなどを利用して接続した外部端末からでも行えるので、キーボードが効かなくなったり、画面に何も表示されなくなった場合でも復旧が可能だ。telnet接続した端末で、「悪さをしている」プロセスのプロセスIDをpsで調べ、そのプロセスをkillで終了させればいい。通常のkillで死なない場合はKILLシグナルの出番だ。

KILLシグナルでも死なないプロセス

KILLシグナルでも終了できないケースは、いくつか考えられる。ひとつは、「init」(プロセスID「1」のプロセス)などのシステムプロセスの場合だ。これらはシャットダウン直前まで動き続ける必要があるため、KILLシグナルでも終了しないようになっている。

一方、システムプロセス以外のプロセスがKILLシグナルで終了できないのは、特殊なケースに限られる。たとえば、psで表示されるプロセスの状態が「Z」(ゾンビ)になっている場合がこれに相当する。ゾンビ状態のプロセスは「すでに死んでいる」ため、さらに殺すことはできない。通常は、そのプロセスを起動した親プロセスが自動的に成仏させる。このほか、プロセスの状態が「D」(ディスク)、つまりディスクの入出力の完了を待っている状態では、KILLシグナルを送ってもすぐには終了せず、入出力が完了した時点で終了する。

Column

コマンド名を指定して終了させるには

psでプロセスIDを調べ、同じ番号をkillの引数に書くのは面倒なので、たいていの人は「killの引数にコマンド名を直接書けたらなあ」と思ったことがあるはずだ。実は、bashの組み込みコマンドのkillでは、「%コマンド名」という形式でジョブを指定できる。たとえば、バックグラウンドジョブとして実行中のfindを終了させるには、「kill %find」とすればいい。ただし、この方法は

ジョブ番号と同様、現在の端末から実行したコマンドに対してのみ有効だ。また、同じコマンドを複数実行している場合には、「bash: kill: ambiguous job spec: コマンド名」と警告され、それらを同時に(あるいは選択したものだけ)終了したりはできない。

外部からのシグナル(特にKILLシグナル)によるプロセスの終了は、正常終了時の処理が行われない可能性があるため、想定外のプロセスを誤って終了してしまわないようにこうした仕様になっているわけだが、

面倒に感じる人は思いのほか多かつたらしく、killallというコマンドが別途用意されている。

killallの使い方は簡単で、引数でコマンド名を指定すればいい。同じコマンドのプロセスが複数ある場合には、それらすべてに同じシグナルが送られる(-iオプションでプロセスの対話的な選択も可能)。また、デフォルトではTERMシグナルが送られるが、killと同様に「-シグナル名」オプションにより、その他のシグナルも送信可能だ。

今月のお題



指定した文字列をコマンドに含むプロセスを選択して終了するスクリプトを作成する

後半は毎回ひとつのテーマに絞り、それを実現する方法を説明する。今回のお題は、

指定した文字列をコマンドに含むプロセスを一覧表示し、選択したものを終了するスクリプト killsel を作成する

というもの。プロセスIDやジョブ番号を指定しなくてはならないkillと違って、コマンド名の一部を指定するだけでいい。コマンド名で終了できる点では、killall(コラムを参照)を-iオプション付きで実行した場合に似ているが、最初にプロセスの一覧を表示する点異なる。

killselの基本的な処理の流れは、

- (1) ps でプロセス情報を出力する
- (2) grep で指定した文字列を含むものだけに絞る
- (3) 得られた候補を一覧表示して選択を待つ
- (4) 選択されたプロセスをkillで終了する

となる。以下では、まず(1)(2)の処理について説明し、続いて(3)(4)の処理を追加してスクリプトkillselを完成させることにしよう。

psの出力をgrepで絞り込む

psには、表示する情報の項目をユーザーが設定できるoオプションが用意されている。たとえば、

```
$ ps o pid,command
```

とすると、プロセスIDとコマンドラインの内容だけが表示される。項目を示すコード(「pid」など)については、psのマニュアルを参照されたい。

今回のスクリプトでは、ユーザーの所有する全プロセスに対して、プロセスID、制御端末名、コマンド名の3つを表示することにする。対応するコマンドラインは、

```
$ ps xho pid, tty, comm
```

となる。なお、hオプションは、1行目のヘッダを非表示にするためのものだ。

psの出力を、指定した文字列を含むものだけに絞り込むには、本連載ではおなじみの検索コマンドgrepを利用する。たとえば、「gnome」という文字列を含むプロセス情報だけ表示するには、

```
$ ps xho pid, tty, comm | grep 'gnome'
```

とすればいい。

ただし、psの出力にはプロセスIDと制御端末名も含まれるため、このままではその部分と一致してしまう場合もある。たとえば「tty」を含むコマンドを検索しようとしても、制御端末を持つ全プロセスが表示されてしまうのだ。この問題を解消するには、プロセスIDと制御端末名の表示に使われる行頭から15文字分の文字列を読み飛ばせばいい。具体的には、

```
$ ps xho pid, tty, comm | grep -E '^.{15,}tty'
```

などとする。

検索パターン前半の正規表現「`^.{15,}`」は、行の先頭から15文字以上の任意の文字列にマッチする。このため、後半の「tty」はそれ以降の文字列に対してしかマッチできないのだ。なお、このような{}を使った文字数指定は拡張正規表現と呼ばれ、grepでは-Eオプションを指定する必要がある(egrepでは使えないので注意されたい)。

実際のスクリプト前半部分では、上記のコマンドラインで得られるプロセス情報を、コマンド置換\$()を使って1行ずつ配列の要素に格納する。この部分のプロトタイプは以下ようになる。

```
1: IFS='
2: '
3: psarray=( $(ps xho pid, tty, comm | grep -E
   ".{15,}$1") )
```

まず、1、2行目では、環境変数IFSの内容（初期設定はスペース、タブ、改行）を改行のみに再設定している。IFSは、コマンド置換の結果を配列に格納する際の区切り文字として使われるため、スペースが含まれる初期設定のままだとプロセスIDや制御端末名、コマンド名がそれぞれ別の配列要素になってしまうからだ。続いて、3行目でプロセス情報が出力され、スクリプトの最初の引数（\$1）で指定した文字列をコマンド名に含むものだけが、コマンド置換により配列psarrayの要素として格納される。なお、\$1が空の場合は検索文字列が「^.{15,}」になり、すべてのプロセス情報がマッチする。

メニュー表示はselectを使う

配列psarrayに格納されたプロセス情報を番号付きのメニューとして一覧表示し、入力を促すプロンプトを表示して、入力があるとメニューから抜けるという処理は、先月号の本連載にも登場したselect制御構造で実現できる。この制御構造を使うと、上記のような処理をわずか数行の記述で行えるのだ。

後半部分のプロトタイプは以下ようになる。

```
1: PS3='which process? '
2: select selection in "${psarray[@]}"; do
3:   if [ -n "$selection" ]; then
4:     echo "killing $selection"
5:     kill $(( ${selection:0:5} ))
6:     break
7:   else
8:     echo "Invalid Number" > /dev/stderr
9:   fi
10: done
```

1行目でselect制御構造の選択時に表示されるプロンプトを環境変数PS3に設定し、2行目からselect制御構造が始まる。まず、psarrayの全要素を列挙した「"\${psarray[@]}"」がリストとして渡され、配列の各要素に番号が付いたメニューが表示される。ユーザーが番号を入力すると、対応する要素がシェル変数selectionに格納され、do～doneが実行されてメニュー表示に戻る。基本的にはこの繰り返しだ（Ctrl-Cキーで中断可能）。

do～doneの処理は、まず3行目でselectionの内容が空でないか調べる。正しい範囲の番号を選択した場合はselectionにプロセス情報が格納されるので、当然空ではな

い。そこで、4行目でselectionの内容を表示して、5行目でselectionから抜き出したプロセスIDを引数としてkillを実行し、6行目のbreakでselectから抜ける。一方、入力が範囲外の場合はselectionの内容が空なので、8行目でエラーメッセージを出力してメニュー表示に戻る。

スクリプトに関するプロセスを一覧から取り除く

前半と後半のプロトタイプを組み合わせると、スクリプトとは一応完成するが、ひとつ問題がある。プロセス情報の一覧表示に、このスクリプトに関するプロセスが含まれてしまうのだ。具体的には、スクリプト本体、コマンド置換のために実行されるbashのサブシェル、サブシェル内で実行されるpsとgrepの4つのプロセスだ。

これらを表示しないようにするには、psで出力されるプロセス情報の末尾4行を取り除けばいい。この修正を加えたスクリプトkillselをリスト1に示す。修正部分は、4行目のコマンド置換内のコマンドラインで、psとgrepの間に、入力行を逆順に出力する「tac」と、7行目以降を出力する「tail +7」がパイプ（|）を接続している。

実は、「入力の末尾N行を取り除く」コマンドは用意されていないので、「入力行を逆順に出力して、先頭のN行を取り除く」ことで代用しているのだ。tacとtailのプロセスも取り除く対象に追加されるので、tailの引数は（4行分取り除く+5ではなく）6行分取り除く+7になることに注意されたい。また、tacにより、メニューはプロセスIDの大きいものから順に表示される。

リスト1 スクリプトkillsel

```
1: #!/bin/bash2
2: IFS='
3: '
4: psarray=$(ps xho pid, tty, comm | tac | tail +7 |
grep -E "^.{15,}$1")
5: PS3='which process? '
6: select selection in "${psarray[@]}"; do
7:   if [ -n "$selection" ]; then
8:     echo "killing $selection"
9:     kill $(( ${selection:0:5} ))
10:    break
11:   else
12:     echo "Invalid Number" > /dev/stderr
13:   fi
14: done
```

初心者から脱出

Emacs はじめました

第7回 カスタマイズへの階段

Emacs のカスタマイズ性の高さといったら、それはもう折り紙つき。それというのも、Emacs が Lisp (elisp) で書いてあってマクロを追加したり、Emacs の一部を書き換えたりできるおかげなのですが、いきなり Lisp を理解するのはいささか敷居が高いのも事実。でも、Lisp がわからなくて、手の届く範囲で自分の道具を使いやすくすることはできるんです。

文：佐々木太良
Text : Taroh Sasaki

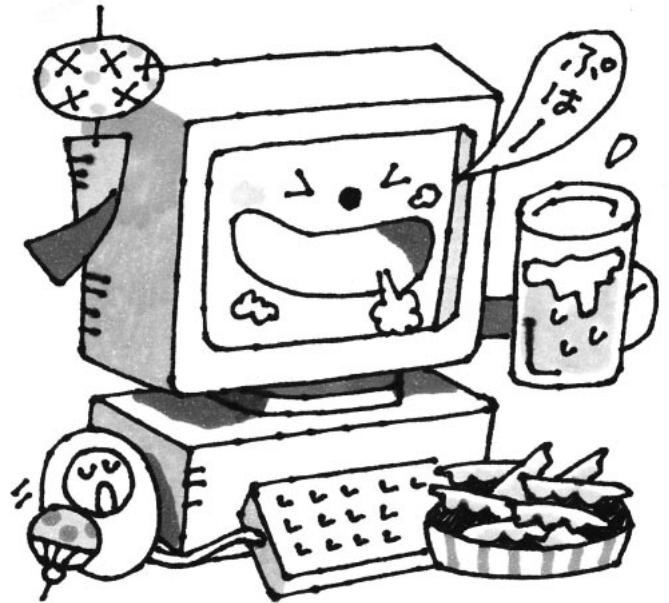


Illustration : Manami Kato

はじめに

皆さん、暑さにめげず Emacs してますか～？

「Emacs 上で動くアプリケーション」シリーズはちょっとお休み。今回は、Emacs のカスタマイズに挑戦します。とはいっても、実は今までにもアプリケーションを設定するとき、知らず知らずのうちにやっているんですよ。今回は、そここのところを正面から取り組んでみます。

Emacs のカスタマイズ

Emacs に新たな機能を追加したり、既存の機能をカスタマイズ（好みに合わせて変更）するには、

自分の趣味に合わせて自力で機能を変更する

Emacs Lisp (elisp) で書かれたアプリケーション（マクロ）を手に入れてインストールする
まず、外部のプログラムを持ってきてインストールする。
Emacs からそのプログラムを呼び出すようなマクロをインストールする（この部分は2番目の方法と同じ）

のような方法があります。

また、こうしてインストールした機能は、

自分だけ楽しく使用する

このコンピュータを使っているみんなに自分の趣味を押し付け……もとい、便利な機能が使えるようにする

のように設定できます。

RPM などのパッケージでは、インストールから設定までの手順は自動化されているのでおまかせにできます。でも、こうした仕組みを知っているのと知らないのでは大違い。自分の好みに合わせて手を加えたり、パッケージ化されていないちょっとしたマクロを自分の手でインストールすることができれば、Emacs ライフも楽しさ倍増というものです。

Emacs と Lisp

Emacs が「Lisp」で書かれていて「マクロ」でカスタマイズできることは、もうご存じでしょう。では Lisp とは、そしてマクロとは、何なのでしょう。Lisp というのは、カッコが多くて目がちかちかするプログラミング言語です……冗談だと言いたいところですが、Emacs のカスタマイズをしていて左と右のカッコの数が合わず、ほんとうに泣きそうになることがあります。

Lisp は、プログラムが書かれた順番に実行される手続き型の C 言語などとはちょっと変わっています。えらく乱暴に言ってしまうと、リストというデータ構造を処理するこ

とでなにかを達成し、プログラム自身もリスト構造で表わされているのです。

Emacsは、主要部分（Lisp言語を解釈する部分など）のみがC言語で書かれ、その上に乗っているエディタの細々した機能はすべてLispで書かれています。

マクロを使ってみよう

まずは「他人が作ってくれたマクロを持ってきて使う」方法から説明しましょう。

最近のEmacsのパッケージには、有名な「便利マクロ」は標準で入っています。ディストリビューションでも、あらかじめ用意してくれていたり、パッケージであとから導入できることが多くなっています。しかし、ここで述べるやり方は、友だちが作ってくれたりネット上で流れていたちょっとしたマクロを自分の手でインストールする場合に役立つものです。でも、お仕着せの環境を使うにしても、マクロが動作する原理は知っておいてムダにはなりません。

マクロファイルはどこにある

マクロは、.el（elispファイル）または.elcという拡張子がついたファイルに保存されています。.elファイルのほうはテキストエディタで読み書きできます（コラム「.elと.elc」参照）。

これらをどこに置いたらよいかというと、Emacsのロードパスのどこかです。シェルの環境変数などをセットしたことがあればもうわかりでしょう。Emacsはマクロの実行を指定されると、ロードパスに列挙されているディレクトリを順番に探してマクロをロードします。ロードパスはload-pathというLisp変数に格納されているので、**コラム「設定内容を確認する」**の方法で確かめてみてください。

さて、マクロの置き場所は、このロードパス上のどこでもよいのですが（ちょっとしたマクロなら、/usr/local/emacs/site-lisp/などがよいでしょう）、新しいディレクトリを作ってロードパスに追加することもできます。たとえば、自分専用のマクロを /elisp/ディレクトリにまとめて置くのなら、.emacsに、

```
(setq load-path
      (cons "~/elisp" load-path))
```

と書いておけばよいのです。

さて、実行

こうしたマクロを手動で実行するときには、M-x load-libraryとします。たとえば、モード行に時刻とCPU使用率を表示してくれるdisplay-timeマクロ（display-time.el）は、多くのディストリビューションに付属しています。これを実行する場合は、

Column

.elと.elc

C言語などで書いたプログラムは、いったんコンピュータが直接実行できる機械語にコンパイル（変換）しておいたものを実行するので、高速に動作します。しかしちょっとしたことなら、人間が書いたプログラムをコンピュータが解釈しながら実行してくれるインタプリタのほうが、コンパイルの手間もなく便利です。

Lispのプログラムはインタプリタ言語なので、思いついた機能をマクロで作成して追加し、試して気に入ったらすぐにふだん使いのEmacsに導入できるという気軽さがあります。しかしインタプリタ言語には、実行効率がいささか悪いという面もあります。そこでEmacsでは、バイトコンパイルという手法が使われています。Lispのマクロ(.elという拡張子がついていることが多い)は一種のテキストファイルで、これがEmacsに与えられるとLisp解釈エンジンが一字一句解釈することになっています。この解釈が終わってあとは実行するだけ、という状態を保存しておけば、ある程度実行は速くな

ると考えられます。牛でいえば、反芻したあとの牧草を保存しておけば消化が速い、といえれば分かりやすいでしょうか。え、ちょっと汚ない？ 失礼。

この解釈終了後の中間状態が、.elcという拡張子がついたファイルです。エディタで開いてみればわかりますが、これは元のマクロとは似ても似つかぬバイナリファイルです。

Emacsのアプリケーションパッケージは、元の.elファイルがなく、.elcファイルのみで配布されることもあります。またソースパッケージ(SRPM)などの場合は、インストール中にバイトコンパイルが自動で行われます。

手動で.elファイルから.elcファイルを生成する手順は、

1. Emacsを立ち上げる
2. M-x byte-compile-fileを実行する
3. ファイル名(.....el)をたずねられるので入力する

となります。

Load library: display-time

と入力します(画面1)。“el”は省略できます。同様に、ロードパス上にないマクロはM-x load-file で実行可能です。でもこのときは、マクロの所在ディレクトリも入力しないとけません(補完がききます)。

カスタマイズってどこをいじるの?

毎回使う機能なら、初期設定で起動直後から使えたほうがいいにきています。Emacsの初期設定ファイルは2種類あります。自分だけ(あるいは特定のユーザーだけ)の操作環境を変えたいのなら、自分(その人)のホームディレクトリにある.emacs(/.emacs)を変更します。

Linux コンピュータの全ユーザーも同様にしたいければ、/usr/local/emacs/site-lisp/site-start.el というファイルをまったく同じようにいじることになります。Emacsは次のように動作します。

1. 起動時にプログラムのデフォルトの設定で初期化する
2. システム全体の初期設定ファイルを読む
3. 個人の初期設定ファイルを読む
- (4. シェルの環境変数から設定を読む)
- (5. コマンドラインオプションで設定を変更する)

UNIXの各種ツールの初期設定ファイルと同様ですね。会社や学校の研究室の共用マシンなら、システム管理者がシステム全体の設定ファイルを使いやすく手入れをすれば、全ユーザーがメリットを享受できます。また、その設定が気に入らなければ、ユーザーは個人用の設定ファイルを使って別の設定に上書き可能です。

個人用のLinuxマシンなら「管理者=ユーザー」ですから、どちらでもよいように思われるかもしれませんが、でも、システムの設定はrootユーザーで操作し、ふだんは一般ユーザーとして仕事をしましょう。一般ユーザーの.emacsだけをカリカリにカスタマイズしてしまうと、rootでの操作で「あれっ?」ということになります(伝統的なUNIXの教えに従うと「rootがEmacsを使うなんてけしからん」ということになりそうですが.....)。

ディストリビューションによっては、ユーザー環境のテンプレートファイル/etc/skel/.xemacs.elが用意してあって、adduserでユーザーアカウントを作成するときにホームディレクトリにコピーされ、その後Emacsを起動する



画面1 display-timeマクロでモード行に表示されたタイマーとCPU使用率

ときに /.xemacs.elが参照されます。この場合でも、site-start.elをLispディレクトリ(/usr/lib/xemacs-*/lisp/など、細々とした.el、.elcファイルがたくさんある)に置けば、システム共通の設定として使われます。進境著しいEmacsの設定をリアルタイムにユーザーの環境へ反映させるには、むしろsite-start.elのほうが優れているでしょう。

ここで、.emacsではなく.[x]emacs.elという名前を使っているのは、将来ここに自分勝手(?)な設定を書きすぎて、バイトコンパイルしなければいけなくなったら、Emacsが.emacs.elcも読んでくれるからです。

そんなわけで、以降の.emacsの説明はsite-start.el、.[x]emacs.el[c]にも同様に当てはまります。

まずは、ディストリビューション付属の初期設定ファイルを覗くことから始めましょう。各社がくふうをこらして環境をセットしているわけですから、この手の設定ファイルがいちばんの見どころなのです。

初期設定のあれこれ

.emacsやsite-start.elもelispマクロなので、本格的に初期設定をいじるには、elispを学ばないといけません。それでも、単純な変更ならある程度はパターン化することができます。

マクロをロードする その1

前述のM-x load-libraryを.emacsのなかで実行し、起動時にマクロを自動的にロードします。

```
(load-library "prom-mew")
```

一般的に、M-x hogehogeという操作が可能だったとすれば、.emacsのなかに“(hogehoge)”と書くこともできます。

マクロをロードする その2

これまでの連載を読み返すと、新たにマクロを追加したときに、上記とは異なる方法で起動するものがあることに気づきます。たとえばMewの場合、.emacsに、

```
(autoload 'mew "mew" nil t)
```

....

と書きましたが、これはM-x mew (mewは“ ’ ”の後ろの名前)が実行されたときに、mewというマクロ(“ mew ”で指定するファイル名)をロードして実行します。実は、mew マクロにはこのツールのスタートアップ方法が書いてあります。nilの部分には、helpで出てくる説明を入れることができます。この方法は、パッケージ化されている巨大なツールによく見られます。

マクロをロードする その3

Wanderlustなどは、autoloadするファイルがあまりにも多いため、ロードパス上に1個、wanderlust-startup.elというファイルを置いて、.emacsに、

```
(require 'wanderlust-startup)
```

と書いています。wanderlust-startup.elのなかでやっていることはその2の方法とあまり変わりません。

また、最近のXEmacs21のパッケージ化のやり方では、サイトパッケージのディレクトリ(/usr/lib/xemacs/site-packages/lisp)に、Emacsのアプリケーションパッケージごとにelisp マクロをまとめて置くようになってます。

変数をセットする

「Lispのプログラミングをすっとばしといて何抜かすか」とお怒りになる向きも、まあまあ。

Emacsやあまたある便利なマクロのなかには、Lisp変数に設定されている値によって振る舞いを変え「定食方式カスタマイズ」を可能にしているものがあります。

たとえば.emacsに、

```
(setq tab-width 4)
```

のような行を入れておくと、Emacsは変数tab-widthを参照してタブストップを4カラムごとに変更します。つまり、

TAB キーを押すと4カラムごとにカーソルが移動していきます。ツールによっては「~という変数をセットして使ってね」(セットしないと動かないよ)などというものもあるので、これはぜひ覚えておきたいところです。Wanderlustの場合は.emacsに、

```
(setq wl-summary-highlight nil)
```

```
(setq wl-summary-highlight t)
```

と書いてハイライトを指定できます。“ t ” “ nil ” は真・偽を表す論理値ですが、あまり気にせずに使ってください。

Column

設定内容を確認する

Emacsは細々した設定をいろいろ読み込むので、設定内容を確認するためにいちいち各種のマクロを覗いていたのではたまりません。もうすこしよい方法は、カスタマイズメニューを使うことですが、設定の多くはLisp変数にセットされているので、たいがいは変数の内容を表示させればすみます。

変数の内容を確認する方法は2つあります。まず1つめは、Emacsのヘルプ機能を使って目当ての変数の中身を表示させる方法です。ヘルプのキー操作C-h(ディストリビューションによってはC-Hに割り当ててある)に続いて、“v”を入力すると変数を覗き見るモードになります。Describe variable:のプロンプトに続けて、load-path(補完もきくので l o a TAB p TAB で結構です)と答えて RET キーを押すと、ウィンドウが半分に分れて、* Help *という名前のウィンドウに変数の内容と簡単な説明(ない場合もある)が表示されます。

```
load-path's value is ("/usr/local/share\
/mule/site-lisp/apel" "/usr/local/share\
```

```
/mule/19.34/site-lisp/emu" "/usr/local/\
....
```

Documentation:

```
*List of directories to search for files
to load. Each element is a string
(directory name) or nil (try default
directory). ...
```

もうひとつの方法は、Lispの対話機能を使って直接、変数を表示させてしまう方法です(コラム「ちょっと便利な実験環境」参照)。* scratch *バッファがあるときは、たんに、

```
load-path[C-j]
("/usr/local/share/mule/site-lisp/apel"\
"/usr/local/share/mule/19.34/site-lisp\
/emu" "/usr/local/share/mule/site-lisp/\
flim" ...(略)
```

のように表示されます。

文字列をセットするときは、

```
(setq tex-command "jlatex")
```

のようにします。複数の値を列挙したリスト（配列のようなもの）をセットするときは、

```
(setq mew-mail-domain-list '("taroh.org" "ay.nu"))
```

と書きます。要素が1個でも' (...) は省略できません。使用する局面はあまりないのですが、ツールの使用中に変数の値を変えるのなら M-x set-variable で可能です。

たとえば fill-mode（5月号に登場）の行幅を変えるには、M-x set-variable の “ Set-variable: ” のプロンプトに続けて文字数を答えます。よく使いそうな変数にはたいていショートカットが用意されていて、fill-mode には C-u 数字 C-x f が使えます。

フック

「きっかけを用意してあげるから、そのときにあなたのマクロを実行したげるよん」というツールもあります。「きっかけ」でなにかを呼び出すことをフックといいます。

たとえば、Mew のドラフトモードに入るとき、C-c C-i で出だしの「ちわ、佐々木です」という文章を自動的に挿入する方法を紹介したことがありましたが（7月号のリス

ト1、2参照）これは Mew がドラフトモードに入るのを「きっかけ」にして、出だしを挿入するマクロを実行しているのです。

フックの中身は、真偽値や文字列ではなく関数であることに注意してください。おおむね、

```
(add-hook 'フックの名前 '出来あいの関数名)
```

または、

```
(add-hook 'フックの名前
  (function (lambda ()
    やりたいこと
  ))
)
```

という形になります。

メニューを活用する

ここまで苦勞して Emacs をカスタマイズする方法を見してきましたが、Emacs20 や XEmacs21 では、メニューから Emacs の基本機能をカスタマイズできます（石投げないでね！）。残念ながら、この機能だけでは Emacs のカスタマイズすべてはできないし、これまで説明してきた基本的なことは知っているべきですが、時間のあるときにカス

Column

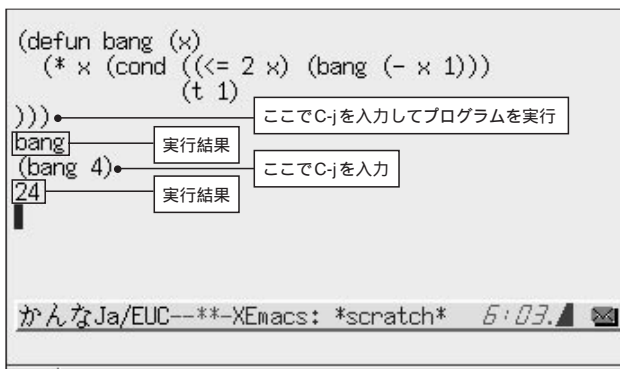
ちょっと便利な実験環境

カスタマイズに慣れてくると、elisp プログラマーでなくても M-H v で変数の中身を見たり（M-H f で関数の仕様を確かめたり）しながらマクロの変更や継ぎ足しをするようになってきます。そんなとき、ちょっとした elisp の書式を試してみたいことがあります。

このためのバッファが、おなじみの *scratch* なのです。今まではたんなる「落書き帳」として説明してきましたが、じつはこのバッファ、モードが（Lisp Interaction）となっていて、入力されたものを elisp コマンドとみなして実行できるようになっています。年輩の方には、BASIC マシンと同じノリだといえればわかりやすいかもしれません（BASIC もインタープリタですよ）。

scratch バッファを開いて、右の画面のように、elisp の文を実行できます。改行は、キーワードや文字列の途中でなければ、好きなところでできます。通常のバッファだと、C-j は C-m と同様に改行として働きますが、（Lisp Interaction）なるバッファでは、実行が C-j、改行が C-m だからです。

この方法を応用すると、Emacs を使って Lisp 言語のお勉強ができます……というのは筆者もちょくちょくやっていることなのですが、このたびこれを発展させた書籍が刊行されました。著者は Mew の作者、山本和彦さんです（『リスト遊び』山本和彦著、アスキー刊）。Emacs マクロプログラマーになろうというよりも、純粋に言語としての Lisp を学びたい人向けですが、いずれ Emacs のマクロを書くという人は押さえておきたい一冊です。



タイムズメニューをついていると思わぬ便利項目を発見してちょっと得した気分になります。

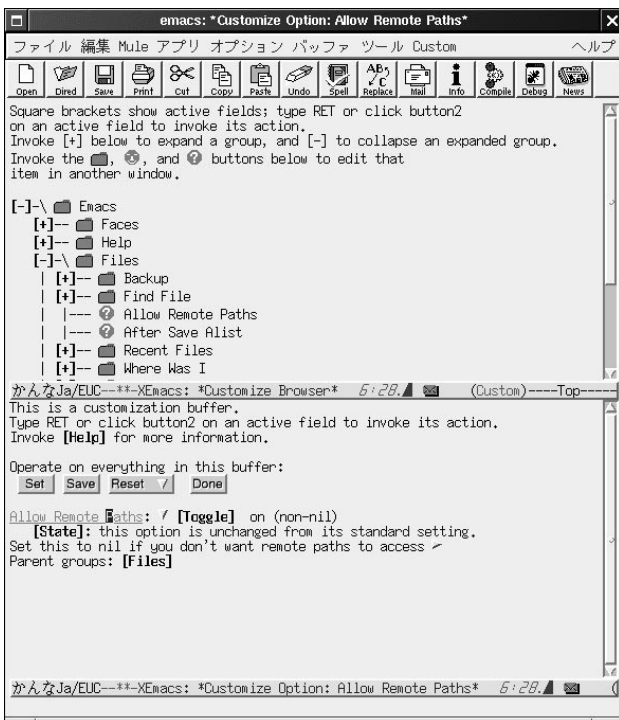
M-x customize-browseを実行すると、なにやらフォルダの中にフォルダやいろいろなマークの木構造が現れます(画面2)。あれね、どこかで見たような画面ですね。前回のWanderlustのフォルダバッファに似ています。

このバッファ(*Customize browser*)は、Emacsというフォルダをルート(頂点)として、似たようなカスタマイズ項目をフォルダにまとめて名前をつけてあります。フォルダについている[+][-]印を、マウスの右ボタン(ボタン3)でクリックすれば開けたり閉めたりできます。

末端(Wanderlustでいえばメッセージ)の項目には[?]マーク(XEmacs21では顔マーク)がついています。カスタマイズ項目を右ボタンでつくと、各項目のカスタマイズができるバッファが開きます。この中では選択できるものが、[...]になっていたり色違いになっています。

- [Toggle] 中ボタン(両ボタン同時押し・ボタン2)でクリックするたびに状態が変わる
- [Value Menu] 中ボタンでメニューがドラッグできる
- (数字) 書き換えると値がセットできる

なにか項目をカスタマイズしたら、[Save] ボタンを押してセーブしましょう。詳しくは説明しませんが、セーブ



画面2 カスタマイズメニュー (XEmacs21)

したあとで.emacsを見てみると、最後のほうに、

```
(custom-set-variables
 '(next-line-add-newlines t)
 '(delete-key-deletes-forward t)
 '(tab-width 4)
 '(kill-whole-line t))
```

のような行が追加されているのがわかります。前のほうに手作業で書いておいた同じ変数の設定は、メニューからの設定で上書きされることとなります。

Emacs19では、メニューによる簡単なカスタマイズとはいきませんが、この原型になったM-x edit-optionsという機能があるので、Emacs19でカスタマイズに挑戦したい人は、ひととおり見ておくといいでしょう。

便利なカスタマイズたち

ショートカットキー

カスタマイズしていちばんよく作るのは、なんといっても短縮キー操作です。Emacsは、よく使う“M-x elisp関数”の操作に短縮キー操作が用意されていないことがしばしばあります。もしかして、はじめから自分でカスタマイズすることを前提にこうした関数が作られているのかなあ(もっとも、すぐに「連想しやすくして」「空いてる」キー操作がほとんどないという問題に直面します(笑))。

筆者の場合、行番号にジャンプするM-x goto-lineを、C-c Gに割り当てて使っています。このためには、

```
(global-set-key "\C-cG" 'goto-line)
```

とします。global-set-keyというのは、現在のモード(テキストを編集中心だとかPerlのソースだとか)を問わずある関数の実行をあるキー操作に割り当てるものです。このほかに便利なキー割り当てをいくつか紹介しておきましょう。ただし、これが身につけてしまうと他人のEmacs環境は使えなくなるし、後輩に伝授するときにはちょっと不便になります(マクロごと伝授するのが正しいでしょう)。

```
(global-set-key "\C-c%" 'vip-paren-match)
```

C-c %で対応するカッコにジャンプする(.emacsに(load "vip")の記述が必要、Emacs19で使用可能)


```
(global-set-key "\C-xt" 'toroku-region)
```

C-x t で選択した範囲を Wnn の辞書に登録する (5月号で登場)

ウィンドウサイズとフォント (Emacs20 / XEmacs21) デフォルトのままのウィンドウサイズとフォントでは気に入らないことがあります。変数 (リスト) `default-frame-alist` を以下のように変えてみましょう。

```
(setq default-frame-alist (
  width 80
  height 25
  cursor-color "white"
  font "-*-fixed-medium-r-normal--16-*-*-c-80-iso8859-1"
))
```

筆者は日本語のほかにも中国語 (Big5) やハングル語のフォントを使うことがあるので、これらを併記してもサイズが見合う 16 ドットフォント (上記の指定では jis16 が使われる) を指定しています。

このほか、以下の変数 (リスト) に同様の書き方ができます。

```
initial-frame-alist
```

起動時のフレームの設定

```
minibuffer-frame-alist
```

独立ミニバッファのフレーム設定

```
default-x-frame-alist
```

X 使用時のフレームの設定

```
default-tty-frame-alist
```

tty 端末 (TELNET など) 使用時のウィンドウサイズの設定

それぞれのリストのなかで指定できるキーと値については、C-h v で表示される変数の説明を見てください。

XEmacs の見栄え!?

これは、Emacs というよりは X のカスタマイズですが、Emacs20 と XEmacs21 では、好みの背景にすることができます。ファイル `/.Xresources` に、

```
Emacs*EmacsFrame.default.attributeBackground: brown
```

```
Emacs*EmacsFrame.default.attributeForeground: turquoise
```

の 2 行を書いて、X サーバを再起動 (いったんログアウトして再ログイン) してみてください。この指定は、前述の `default-x-frame-alist` でも指定できるはずなのですが、バージョンによってはうまくいかないこともあるようです。

長時間使う場合は、ハデハデにするよりは、白系に黒系 (black や white、gray10 ~ gray90 といった名前で色を指定) を組み合わせるのが見やすいようです。上記の `default` を `isearch` に変えて、インクリメンタル検索中のサーチ文字列をカラー化するのが実用的な使い方でしょう。

さらになんと、画像 (.xpm 形式) を貼り付けることもできます。好きな画像 (`/lib/hoge.xpm` としましょう) を用意して、上記を、

```
Emacs*EmacsFrame.default.attributeBackgroundPixmap:~/lib/hoge.xpm
```

としてみてください.....文字が見えにくいよ、といわれそうですが (笑)。

小物たち (Emacs 19/20/21)

```
(display-time)
```

モード行に時刻と CPU 使用率を表示してくれます。

```
(setq next-line-add-newlines nil)
```

t の場合はバッファの終端で `next-line` (C-n) を押したときに次々行を追加してくれますが、nil の場合は止まってくれます。

```
(kill-whole-line nil)
```

t の場合は行頭で `kill-line` (C-k) を押したときに行全体を削除して次の行が繰り上がってきますが、nil の場合は (行中で `kill-line` したとき同様) 1 アクション目で行末まで削除、2 アクション目でその空行を削除、という動作になります。

Emacs 19 と Emacs 20 (XEmacs 21) で動作を変えたいいろいろ便利なマクロを手に入れても、持っている Emacs のバージョンと合わなくて、Emacs 19 では動くけど Emacs 20 や XEmacs 21 ではダメ、などという場合があります。Vine Linux などのように 2 つの Emacs が付属し

ている場合は、使用する Emacs アプリケーションによって立ち上げる Emacs を変えることもあるでしょう。また、Wanderlust などのように、それぞれのバージョンごとに別のマクロが用意されている場合もあります。

このような場合、まずマクロのロードパスを変えてやる必要があります。幸い、最近のディストリビューションでは、Emacs19 のマクロは /usr/local/mule/19.34/... 以下に、XEmacs21 用のマクロは /usr/local/xemacs-21.1.6/... 以下に、などと分けることが多いようです。これを調べて、Emacs19 用のマクロと XEmacs21 用のマクロをそれぞれのディレクトリにうまく配置すれば、mule + Emacs19 を起動したとたん XEmacs21 用のマクロを読もうとして失敗する、などということはありません。

問題は、ホームディレクトリに1つしか置けない .emacs です。それでも、ちょっとした Lisp の条件判断を使えば、実行している Emacs のバージョンを自動判別して個別の操作ができます。

```
(cond ((string-match "^19" emacs-version)
      Emacs19 系のときに実行されるべきこと
      (いくつかも)
) ((string-match "^20" emacs-version)
  Emacs20 系のときに実行されるべきこと
  (いくつかも)
))
```

実はこの小技は、(はじめから分けておけばよいはずの) site-start.el に対しても有効です。2つの異なる場所にある site-start.el をそれぞれ編集していると、どちらのバージョンにも共通のカスタマイズ (Wnn のキー配置の変更など) を2回書かなければいけません。そんなときは一方の site-start.el から他方の site-start.el にシンボリックリンクを張って (同一のファイルとして) それに対して上のような変更を加えればよいでしょう。

このほか X を使用しているときには、以下の技も使用できます。

Emacs19 の場合

```
(cond
  ((and window-system (string-match "^19" emacs-version))
    ここに中身
  ))
```

Emacs20 以上の場合

```
(cond
  ((>= (string-to-int emacs-version) 20)
    ここに中身
  ))
```

.emacs のデバッグ

カスタマイズにはまってくると、.emacs にいろいろなものを書きすぎて、ある日突然動かなくなってしまうことがあります。どんなプログラミング言語でもデバッグに王道はないといわれますが、起動直後のミニバッファの警告や、なにか文字を入力したあとの * Warnings-Show * バッファの表示 (Emacs20、XEmacs21 の場合) を見ると、原因の見当がつくこともあります。筆者は、以下のような超原始的な方法をよく使います。

1. すべての行の先頭に ";" をつけてコメントにする (設定ファイルのない生 Emacs 状態に戻る)
2. たいがい追加したアプリケーションごとにマクロがブロックになっているので、そのブロックごとに ";" を外して (C-x r k で矩形削除が便利) 再起動する
3. 動かなくなった行に間違いあり

おわりに

Emacs のカスタマイズへの道は、けして Lisp の達人になることばかりではありません (便利なツールを作りたい、という場合は別ですが)。人のカスタマイズをおりに触れて見て回ることもお勧めです。ディストリビューションに付属している気合いの入った設定ファイルは参考になるでしょう。まず、よく使う操作や不便に思ったことを、知っている Lisp の範囲で何とかしてみることはじめてみましょう。面白いマクロができたら、もちろんここで取り上げさせていただきますので、ご意見とともども taroh@taroh.org までお気軽にメールをお送りください。

では来月まで Happy Hacking!!

Linux 日記

第12回 名前解決(5)

さて、手回し計算器とDNSの解説もいよいよ大詰め。今回は、手回し計算器の構造、そしてDNSでの逆引き、メール配送と名前解決の関係について解説します。

文： 榊 正憲

Text : Masanori Sakaki



先月、タイガー計算器の使い方を簡単に紹介した。せっくなので今回は構造を紹介しようと思う。製品によって構造上の差違はあるが、基本的な構造は似たようなものだ(と思う)。以下の解説は、筆者が持っている物に基づいている(写真とその説明を付録CD-ROMに収録したので、もし興味があれば参照してほしい)。

チェックレジスタ

計算に使用する数値はチェックレジスタで指定する。我が家の計算器では、このレジスタは10桁である。各桁ごとに0から9までの10ポジションのスライドレバーがあり、レバー位置に応じて表示窓に数字が表示される。このレバーは、ハンドルで回転するドラムに連動しており、ドラム内部のカム機構で、プロダクトレジスタの歯車を進める9個の歯を制御する。チェックレジスタのドラムに付いている歯は、ハンドルの回転角度に応じて、飛び出したり引っ込んだりしている。プロ

ダクトレジスタの歯車と噛み合う位置にあるとき、飛び出していればプロダクトレジスタの値が増減し、飛び出していなければ変化しない。歯が出たり入ったりするタイミングは、スライドレバーのポジションで決まる。スライドレバーが0なら歯はずっと引っ込んだまま、9ならすべての歯がプロダクトレジスタの歯車と噛み合う。加算を行うか、減算を行うかは、ハンドルの回転方向によって決まる。この構造により、ハンドル1回転の操作で、各桁ごとに0から9までの加減算を行えるのだ。

プロダクトレジスタ

ハンドルの回転により、チェックレジスタの内容がプロダクトレジスタに累積的に加減算される。プロダクトレジスタの中核となる部分は10歯の歯車で、各歯車の回転角に応じて、表示窓に0から9の数字が表示される。乗算の桁数の関係から、プロダクトレジスタの桁数はチェックレジスタの桁数の倍ある。

加減算処理は、プロダクトレジスタの歯車と、チェックレジスタのドラムに設けられた歯によって処理される。ハンドルの回転により、プロダクトレジスタの各桁はチェックレジスタで指定された0から9の範囲で増減するが、9から0に増えたとき、あるいは0から9に減じたときに、上位桁を±1だけ増減する必要がある。マルチプライヤレジスタは、9から0に変わると同時に上位桁を進めるという単純な機構で実現しているが、チェックレジスタとプロダクトレジスタの加減算にはこの方法は使えない。上位桁の加減算も同時に行われているからだ。そのため、チェックレジスタとプロダクトレジスタは、各桁ごとの加減算と桁上げ処理を別々に行っている。

ハンドルをある程度回すと、チェックレジスタのところで説明したように、プロダクトレジスタの各桁に対して加減算が行われる。ハンドルが特定の角度で重くなるのを避けるために、加減算処理は桁ごとに多少タイミングをず



キャンセルレバー

各レジスタを0にリセットするには、各レジスタの横にあるレバーを操作する。このレバーは各桁のギヤを0の位置まで動かし、リセットを行う。

バックトランスファ

計算式によっては（連続した乗算など）プロダクトレジスタの値をチェックレジスタにセットしたいことがある。これを行うには、まずチェックレジスタをクリアし、次にバックトランスファレバーを操作しながらプロダクトレジスタをクリアする。チェックレジスタの置数レバーには、ふだんはどことも噛み合っていない歯が付いている。バックトランスファレバーを操作すると、シフト機構が1桁の半分だけずれ、プロダクトレジスタの歯車（加減算用の中間の歯車）がこの歯に噛み合うようになる。この状態でプロダクトレジスタのキャンセルレバーを操作するとどうなるか？ プロダクトレジスタの各桁が0に戻るまで、チェックレジスタの置数レバーが進められるのだ（すなわち、Dynamic RAMやコアメモリと同じく、破壊読み出しということになる）。機械式レジスタは、データ転送だけでも一苦労なのだ。

その他

タイガー計算器は固定小数点計算を行うので、小数点位置は人間が管理しなければならない。そのため、スライド式のマーカがあるのだが、我が家の計算器は、チェックレジスタ用のマーカが欠落している。

現在のコンピュータソフトウェアだろうが、機械式計算器だろうが、想定していない操作にある程度耐えなければならない。たとえばプログラムであれば、ダイアログボックスのクリック

できないボタンを無効にするといった処理である。タイガー計算器も同様だ。ハンドルがホームポジション以外の位置にあるときは、各操作レバーなどを動かすことはできない。また、ハンドルを一方に回し始めたら、途中で向きを変えることはできない。計算が歯車機構で行われているので、不当な操作は計算間違いや機器の破損を引き起こす可能性があり、なんとしても避けなければならないからだ。

そのための処理は、すべてカムやレバーを使ったロック機構によって実現されている。ロック機構が有効になると、実際に各レバー類が動かなくなるのである。計算器内部には、用途がよくわからない部品がいくつもあるが、その多くはこういったインターロックのための機構である。

さて、昔のハイテク製品の話はこれぐらいにしておいて、DNSの話の続きだ。今回は逆引きから説明を始めよう。

逆引き

前回まで解説してきたことは、基本的にドメイン名からIPアドレスを求めるといった処理である。これに対して、

IPアドレスからドメイン名を求めるということもできる。netstatなどのネットワークツールは、システムが使っているIPアドレスなどを表示するが、IPアドレスからドメイン名を求めることにより、名前を使って出力することが可能になる。DNSはこのようなアドレスの解決もサポートしている。IPアドレスからドメイン名を求め、「逆引き」という（ドメイン名からIPアドレスを求める処理は「正引き」である）。

逆引きは192.1.1.16といったIPアドレスから、www.ascii.co.jpといったドメイン名を検索する処理である。DNSは、ドメイン名からIPアドレスを求める階層化検索のメカニズムをほぼそのまま使って逆引きを実現している。IPアドレスは32ビットのアドレスであるが、その配分はある程度階層的なものだ。適当なビット配分により、上位がネットワークを識別し、下位がそのネットワーク中のホストを識別するからである。

ドメイン名は右が上位である。それに対して、アドレスは左が上位である。これをDNSの環境にはめ込むために、アドレスを逆順に並べる。192.1.2.0 / 24というネットワークであれば、

Column

特殊なIPアドレス

IPでは、アドレスのホスト部の各ビットがすべて0、すべて1のアドレスは特殊な扱いを受け、個々のノードに割り当てることができない。すべて0のアドレスは、そのネットワークそのものを示すために使われる。クラスC（24ビットネットマスク）の192.1.2.0というアドレスは、特定のノードを識別するアドレスではなく、ネットワーク部が192.1.2であるクラスCネットワークを識別するために使われる。すべて1のアドレスはブロード

キャストアドレスである。たとえば192.1.2.255というアドレスは、ネットワーク192.1.2.0上のすべてのノードを宛先とするアドレスである。

ネットワーク部とホスト部の分割がオクテット境界にならないようなネットマスク値の場合も同様である。たとえば28ビットネットマスク（FFFFFFFF0h）の場合、192.1.4.16（C0010410h）はネットワークを識別するアドレスであり、そしてこのネットワークのブロードキャストアドレスは192.1.4.31（C001041Fh）になる。

リスト1 named.confと逆引きゾーンデータベースファイル

```
zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "localhost.rev.db";
};
zone "2.1.192.IN-ADDR.ARPA" {
    type master;
    file "linux-mag.rev.db";
};
```

2.1.192と並べる。これにより、ピリオドで区切られた文字列を階層的に処理するというDNSの分散データベースのメカニズムを使うことが可能になる。

インターネットドメイン名では、逆引きのためにarpa.というトップレベルドメイン名が定義されている。そのサブドメインに、インターネットアドレスという意味のin-addr.という名前を付け、それに先ほどの逆順のアドレスを組み合わせる。たとえば、192.1.2.0というネットワークの逆引きを行うためのドメイン名は、2.1.192.in-addr.arpaになる。そして、この先頭に、4オクテット目のホストアドレスを付ければよい。

逆引きデータベースファイルもnamed.confで指定する(リスト1)。0.0.127.IN-ADDR.ARPAは、ループバックアドレス(127.0.0.1)の逆引きのためのドメインで、どのようなマシンでも共通である。このゾーンには「1」というエントリがあり、localhost、loopbackといった名前を返すように定義する。

逆引きデータベースは、PTRレコードを使って登録する。これによりIPアドレスからドメイン名を得るレコードであるということがわかる。2.1.192.IN-ADDR.ARPAゾーンのデータベースはリスト2のようになる。SOA、NSレコードは正引きの場合と同じである。そして、Aレコードの代わりにPTRレコードを使って、数値に対してドメイン名を割り当てている。数値にはゾーン名が付加されるので、たとえばnsの1

リスト2 linux-mag.rev.dbの例

```
; SOAレコード
@ IN SOA ns.linux-mag.ascii.co.jp. postmaster.ascii.co.jp. (
    ;
    ;
);
; ネームサーバレコード
IN NS ns.linux-mag.ascii.co.jp.
IN NS ns.ascii.co.jp.
IN NS ns.provider.ne.jp.
; ホストレコード
1 IN PTR ns.linux-mag.ascii.co.jp.
3 IN PTR mail.linux-mag.ascii.co.jp.
4 IN PTR db-svr.linux-mag.ascii.co.jp.
5 IN PTR www.linux-mag.ascii.co.jp.
```

は、1.2.1.192.IN-ADDR.ARPAになる。これにより、192.1.2.1の解決を求めると、ns.linux-mag.ascii.co.jpが得られる。

半端なネットマスクの場合の逆引き
たとえばクラスCアドレス(192.1.2.0/24など)を1つ持ち、そのアドレス範囲全体の逆引きデータベースを作るのは簡単である。2.1.192.in-addr.arpaゾーンのデータベースに、0から255までのPTRレコードを並べるだけでいい。では、ネットマスクが28ビットだったら(オクテット境界と一致しないネットマスク値のとき)どうすればいいか? 具体的なネットワークアドレスは、192.1.2.0、192.1.2.16、192.1.2.32とようになる。

プロバイダとの契約によっては、こ

の程度のグローバルアドレスしか割り当ててもらえないことが多い。実際、筆者のOCN契約も28ビットのネットマスクで、使用可能なグローバルアドレスは16(実際にホストに割り当てられるのは14)しかない。

もっとも簡単な方法は、実際のネットマスクビット数に関わらず、オクテット単位でin-addr.arpaゾーンを管理してしまうというというものだ。たとえば192.1.2.0というアドレス空間を、16人の契約者に16アドレスずつ割り当てているとしよう。この16のネットワークの逆引きは、すべてプロバイダ側の2.1.192.in-addr.arpaゾーンで行えばいい(図2)。この場合、逆引きデータベースの管理はプロバイダ側で行うことになるので、変更などがあつた場合

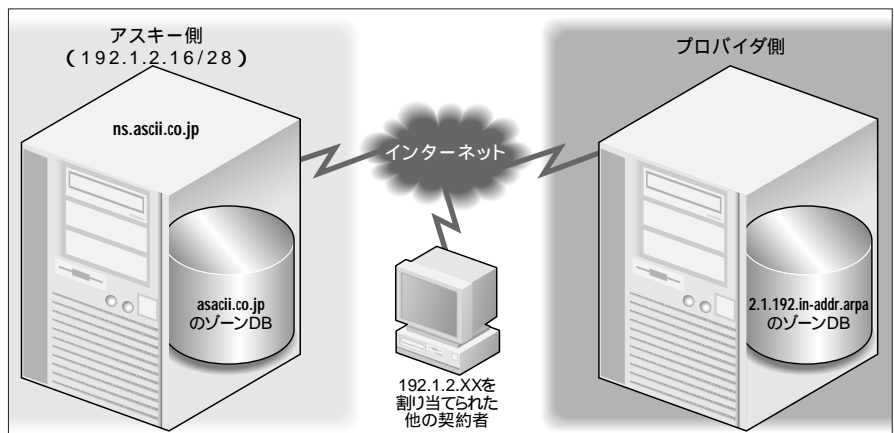


図2 逆引きゾーンだけプロバイダで管理



は、プロバイダ側のデータベースを修正しなければならない。Webブラウザなどを使ってオンラインで変更できるようなインターフェイスが必要になるだろう（プロバイダに変更依頼をメールで送るという方法もあるが）

できれば、16.2.1.192.in-addr.arpaというようなゾーンを契約者側のネームサーバで管理したいところだ。ところが、ここまで説明してきたネームサーバのロジックから考えると、IPアドレスの第4オクテットまで含んだゾーンにPTRレコードを登録しても、うまく機能するとは思えない。たとえば192.1.2.17というアドレス（17という名前のPTRレコード）をこのゾーンに登録したら、192.1.2.16.17というアドレスになってしまいそうだ。しかし、うまく解決する見事なトリックがあるのだ。

筆者がOCN用のネームサーバの設定を行ったとき、これには悩んだ。何に悩んだかという、とてもうまく働きそうにないこのような設定をするようにOCN側が指示しており、しかもそれが正しく機能するからである。nslookupなどでOCN側のサーバ設定を見ても、どうなっているのかよくわからない。『DNS & BIND 第3版』を読み返してやっとヒントをつかんだ。

長くなるのでここにはネタばらしは

Column

普通はMXを登録する

たとえばuser@mail-svr.provider.ne.jpという形式のメールアドレスであれば、mail-svr.provider.ne.jpというホストをメールサーバとして稼働させることにより、MXレコードを登録しなくても、とりあえずメールを配送できる。しかしこのような形式であったとしても、普通はmail-svrを指すMXレコードを用意している。つまり、mail-svr.provider

.ne.jpというドメインに対して、メールエクスチェンジャmail-svr.provider.ne.jpを定義するのである。冗長に思うかもしれないが、そうでもないのだ。まず、MXを使用することで、バックアップサーバを登録できる。もう1つの理由は、メール配信の際に、常にMXを参照するメールサーバがある（送信先サーバの判定にMXのみを使用し、メールアドレスのドメイン名への接続を試みないという設定のメールサーバ）からである。

書かない。もし知りたければ、『DNS & BIND 第3版』の255ページあたりを見してみよう。ヒントはCNAMEレコードの活用である。

電子メールとMXレコード

本連載のDNSの解説の最初のほうで、メールアドレスの話をしたのを覚えているだろうか？ メールアドレスには2種類ある。メールアドレスのドメイン部にメールサーバのホスト名まで含んでいるタイプのものと（user@mail-svr.provider.ne.jpといった形式）、ドメイン部にメールサーバを含んでいないもの（user@company.co.jpの形式）である。ユーザーが使っているメールアプリケーション（クライアント）からメールを受信したメールサーバは、

宛先アドレスのドメイン部から、中継先のメールサーバを判定する。メールアドレスに相手側のメールサーバのドメイン名が指定されていれば、何も悩むことはない。そのドメイン名に対応するアドレスを持つホストに中継すればいい（図3）。

しかし、メールアドレスにメールサーバのドメイン名が含まれていない場合は、どのようにして転送先のメールサーバを調べるのだろうか？ たとえば、user@ascii.co.jpというメールアドレスにメールを送る場合を考えてみよう。ascii.co.jpドメインのメールサーバは、mail.ascii.co.jpとする。つまり、user@ascii.co.jpという宛先にメールを送るメールサーバは、mail.ascii.co.jpのIPアドレスを調べ、そのアドレスに接

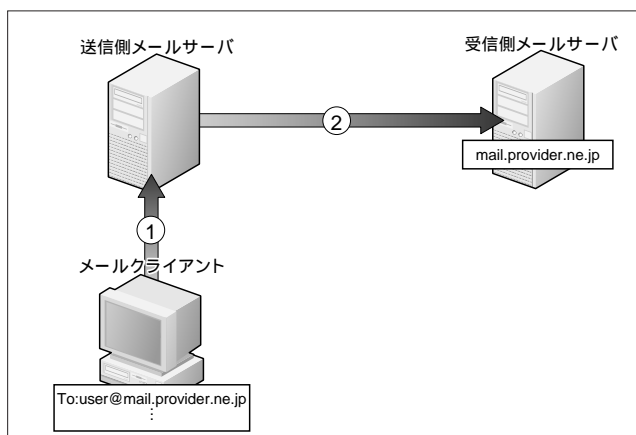


図3 メールサーバ名を含むメールアドレスに対するメール送信

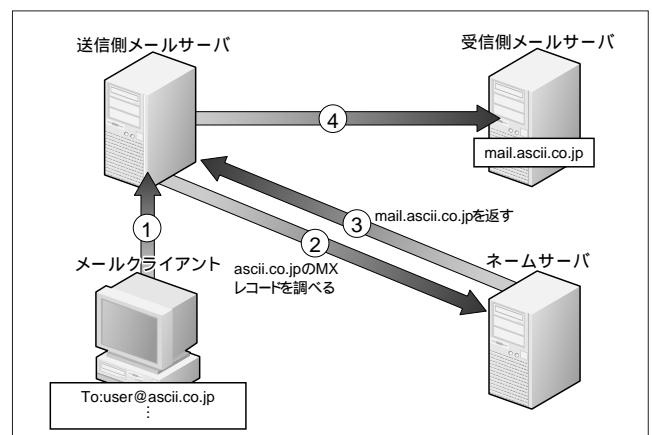


図4 MXレコードを使ったメール送信

続し、メールを送らなければならないのである。これを行うには、何らかの方法で、ascii.co.jpというドメイン名とmail.ascii.co.jpというホスト名を関連付けなければならない。これを行うのが、DNSのMXレコードである(図4)。

メールサーバは、メールアドレスのドメイン部(ascii.co.jp)をメールサーバ名として解釈する前に、まず、この名前に対応するMXというレコードを検索する(リスト3)。このリクエストに対し、ネームサーバは、MXレコードを検索して、メールエクスチェンジャ情報を返す。

送信側メールサーバは、MXレコードで指定されたドメイン名(これは実在のホストである)を受信側メールサーバとして解釈する。そして、この名前をネームサーバに問い合わせ、得られたIPアドレスを使って相手に接続する。MXが登録されていない場合は、メールアドレスのドメイン部を受信側メールサーバとして解釈し、接続を行う(MXレコードが登録されていない場合はエラーとするメールサーバもある)。

このようなMXレコードとメールサーバの連携により、メールアドレスのドメイン部とメールサーバ名を対応づけることができる。画面1の例では、

ascii.co.jpドメインのメールサーバはascii.co.jpのサブドメイン名、つまり、アスキーの組織内のホスト名であるが、メールエクスチェンジャとしてまったく別のドメイン名を指定することもできる。たとえば、ascii.co.jpドメインに対して、mail0.provider.ne.jpというメールエクスチェンジャを指定することもできる。このようにすることにより、アスキー宛のメールはすべてプロバイダのmail0というメールサーバに送られるようになる。

プロバイダが提供するサービスに、user@会社名.co.jpというメールアドレスを使ったメールサービスの提供(もちろん、メールサーバはプロバイダが用意する)があるが、これはこのよう

な仕組みを使っているのである。

このような仕掛けを使う場合は、メールサーバの設定をいろいろいじり、自身のドメイン名(mail0.provider.ne.jpなど)とは関係なく、user@ascii.co.jpというアドレスのメールを送受できるようにしなければならない。

あるドメイン名に対して、メールエクスチェンジャは複数指定することができる。複数のメールエクスチェンジャを用意する場合は、それぞれに優先順位を指定することができる。この場合、メールサーバは優先度の高いサーバに接続する。優先順位が同じなら、ランダムに選択する。これにより、メール処理の負荷分散、バックアップサーバの設置などが可能になる。

リスト3 ゾーンデータベースファイルでMXレコードを定義する

```
@      IN SOA  ns1.ascii.co.jp. postmaster.ascii.co.jp. (
      :
      IN MX  10 mail0.provider.ne.jp.
```

```
$ nslookup
> set type=MX      メールエクスチェンジャレコードを指定
> ascii.co.jp.    ascii.co.jpドメインのMXレコードを調べる
      :
ascii.co.jp      preference = 10, mail exchanger = ascnm2.ascii.co.jp
ascii.co.jp      preference = 10, mail exchanger = ascnm1.ascii.co.jp
      :
```

画面1 MXレコードの検索

Column

UUCPメールとDNS

今ではあまり使われていないが、UUCP(UNIX to UNIX CoPy)を使ったメールのバッチ配信という仕組みがある。UUCPは、UNIXホスト間をモデムなどで接続し、ファイルのコピー、コマンドのリモート実行などを行う一群のツールだが、通常はメール、ニュースの配信に使われる。

UUCPを使ってメール配信を行っているサイトは、たとえIPリーチャブルではなくても、

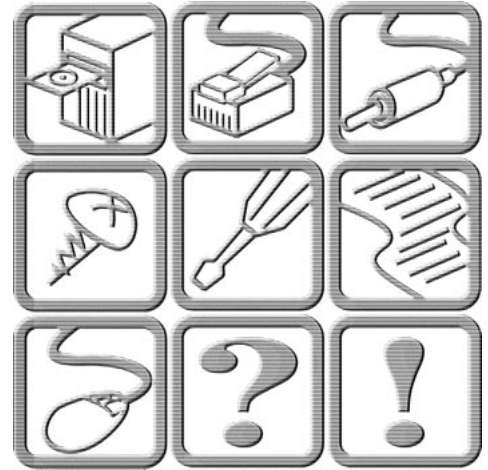
つまりインターネットに直接つながっていても、インターネットドメイン名を使ったアドレスでメール配信が可能である。これもMXレコードのおかげだ。たとえば、takobeya.ascii.co.jpというドメインがascii.co.jpとUUCPでメールをやり取りしているとしよう。writer@takobeya.ascii.co.jpといったユーザー宛のメールのために、ascii.co.jpゾーンのネームサーバに以下のようなMXレコードを用意する。

```
takobeya IN MX 10 uucp-svr.ascii.co.jp.
```

これにより、takobeya.ascii.co.jp宛のメールは、アスキー社内のUUCPサーバホストに送られる。uucp-svrのメールサーバは、宛て先を解析し、UUCP送信用のプールにこのメールを投入する。そしてtakobeyaとUUCP接続を行ったときに、このメールはtakobeya側に転送され、takobeya側のメールサーバで適宜処理されることになる(こういった配信規則なども定義できるから、sendmail.cfは記述が大変なのである)。

Try & Try

[trái] [trái]



ネットワーク - フィルタリング機能 前夜

文: 政久忠由

Text: Tadayoshi Masahisa

今回はよいよLinux 2.4の新しいIPパケットフィルタリング機能を見ていきたいと思うのだが、一応ネットワークの基礎的な部分も押さえておく必要があるので、フィルタリングの一手手前のルーティング、必要な構成とツールあたりまでを紹介しておくことにする。というわけで核心部分は次回の予定。



ネットワーク



ネットワーク機能は今のコンピュータやOSには不可欠な機能だ。最低限インターネットには接続できなければならないし、PC同士、携帯端末などとも通信する必要がある。

コンピュータ分野に限定するとネットワークは2つ以上のホスト(コンピュータとか適当な機器)で構成され、互いに通信、情報のやり取りができる状態を表す。

僕たちの身近なネットワークはLANやインターネットであるが、これらは通常、TCP/IP、そしてイーサネットやISDN、PPPなどさまざまなプロトコル(通信規約)の組み合わせによって成り立っている。OSやCPUが異なる場合、一般に同一のバイナリプログラムを実行することはできないが、大型ホストコンピュータやスーパーコンピュータとPCやPDAの間であってもプロトコルさえ合えばデータ/情報をやり取りすることはできるのだ。

インターネットはTCP/IPで構成された世界中を接続するネットワークだ。またLANはインターネットが

TCP/IPで構成されていることもあって基本的には同じ技術を流用することがほとんどで、物理的な部分はイーサネットが一般的となっている。ちなみにインターネットの物理的な部分は、利用者に近いところではLANで使用されているイーサネット、ISDNなど、バックボーンと呼ばれているところでは1対1の通信ではなく複数の通信データのあるポイントからあるポイントにまとめて一気に運ぶためにATMやフレームリレーなど伝送媒体の帯域を最大限活かす転送効率のよいプロトコルが用いられている。



ルーティング



インターネットにしるLANにしるネットワークは単一ではなく複数のネットワークが接続された状態にある。通常、各接続ポイントには複数のネットワークが接続されていて、どのネットワークを経由していくか選択するようになっている。この作業をルーティング(経路制御)と呼ぶ。

ある情報がひとつのホストからもうひとつのホストに到着するまで、たとえば自宅(日本)からフランスのルーブル美術館のホームページを見るには複数のネットワークを経由する必要があるし、またその経路の組み合わせも複数存在する。

最も効率が良い経路は距離的に短い直通ルートだが、自宅からルーブル美術館にケーブルを通して人なんているわけがない。ルーティング(経路制御)では通常、経由

するネットワークが少なくネットワークの帯域の広い経路を選択して転送するようになっている。自宅から情報を送出する段階で経路を限定指定することもできるが一般には成り行き任せである。基本的に経由するネットワークが少ないということは距離的に短いことが予想できるが、これだけに依存すると回線が細い場合があるのでネットワークの帯域に応じて重み付け（格付け）をして総合評価で最適な経路を決定するようになっている。



IP プロトコル



実際に通信を行うには各ホストを特定する必要がある。これにはMACアドレスと呼ばれるハードウェアアドレスが用いられる。しかしインターネットのような巨大なネットワークをカテゴリ化されていないハードウェアアドレスで管理するのは無理だ（ハードウェアアドレスはハードウェアの製造メーカーが機器に対して付けるもので利用者にとってはランダムなものとなる）。そのハードウェアアドレスで管理するとなると各ルーティングポイントですべてのアドレスを集約する必要があるが数1000万件のアドレス管理なんて無謀なのである。そこで中間管理レイヤとしてIPプロトコルが用いられているのだ（IPはインターネットプロトコルの略）。

IPプロトコルによって使用されるIPアドレスは企業や団体などに対して割り当てられる。これによりあるIPアドレスのまとまりの所在が特定できルーティング情報として利用できるのだ。なおIPアドレスとハードウェアアドレスの対応は各団体が適当に決定してもかまわない。



TCP / UDP



TCP / UDPはそれぞれデータを転送するためのプロトコルで、TCPでは信頼性を、UDPでは能率を提供している。アプリケーションプログラムは目的に応じて選択できるようになっている。TCPではデータの転送を行う前に接続を確立する作業が行われ、またデータの到着を随時確認するようになっているので相手の存在が必ず確認されておりデータがロスしてもこのプロトコルレベルで対応することができる。

一方UDPでは相手の存在を確認することなくデータを送信するので転送の手間は少なく済むが、プロトコルレベルで到着の確認も行われないので本当に正確に到着しているかどうかはわからない。

TCP

```
apollo.telnet > ceres.4309: P 31334:31545(211) ack 0 win 32693 (DF)
```

```
ceres.4309 > apollo.telnet: . ack 31545 win 16676 (DF)
```

```
apollo.telnet > ceres.4309: P 31545:31756(211) ack 0 win 32693 (DF)
```

UDP

```
athena.bootpc > 255.255.255.255.bootps: secs:2560 flags:0x8000 C:athena [|bootp]
```

```
athena.bootps > 255.255.255.255.bootpc: flags:0x8000 C:athena [|bootp]
```

アプリケーションプログラムでは、通常データのロスは死活問題なので信頼性の高いTCPを用いることが多い。HTTP、FTP、TELNET、SMTP、NNTPなどほとんどの通信はTCPを使用している。UDPはデータが少々ロスしようとかまわない分野で使用されている。動画や音声などのストリーム転送だ。また接続を確立しなくてもデータを送信できることから、同時に複数のホストに対して送信するブロードキャストやマルチキャストなどでも使用される。DHCPやDNSで利用されている。なお前回扱ったNFSはUDPを使用していた。データロスが生じてはならないファイル転送でUDPが使用されているわけだが、これはLANなどの物理的に信頼性の高いネットワークで用いられるということと、アプリケーションレベルでデータの到着の確認や再配送を行ってもネットワーク上の転送効率を向上させたいという理由からだ。最近は物理ネットワーク自体の性能が上がったのでNFSでTCPを使用する動きもある（Linuxでも実装されつつある）。



ICMP



ICMPはネットワーク情報を送るために使用される。一般のデータを転送する目的には使用できない。単純にpingコマンドで使用されているプロトコルといったほうがわかりやすいと思う。相手の存在を確認したり経路速度を計測したりする目的で使用される。

```
# ping athena
```

```
PING athena (192.168.0.2) from 192.168.0.13 : 56(84) bytes of data.
```

```
64 bytes from 192.168.0.2: icmp_seq=0 ttl=128 time=0.6 ms
```

```
64 bytes from 192.168.0.2: icmp_seq=1 ttl=128 time=0.3 ms
```

```
--- athena ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.4/0.6 ms
```



ARP / RARP



ARPはIPアドレスからハードウェアアドレスを見つけ出すアドレス解決プロトコル、RARPはその逆のハードウェアアドレスからIPアドレスを探す逆アドレス解決プロトコルだ。先ほど述べたようにTCP/IPネットワークではIPアドレスによってマシンを特定するわけだが、実際にデータを転送するにはハードウェアアドレスが不可欠だ。ハードウェアアドレスは各ネットワークセグメントの段階でIPアドレスを基にARPによって最終的な対応が調べられるようになっている。

```
arp who-has gateway tell apollon
arp reply gateway is-at 0:a0:de:xx:xx:xx
arp who-has athena tell apollon
arp reply athena is-at 0:c0:26:xx:xx:xx
```

ARPの問い合わせはブロードキャストとしてネットワーク全体に伝播され該当するホストからの返答を待つことになる。IPアドレスは32ビット（IPv4の場合）で構成されているが、イーサネットのハードウェアアドレスは48ビットで構成されている。IPアドレスとハードウェアアドレスの対応はhostsファイルのような形（/etc/ethersなど）で静的に設定することも可能だが、最近はほとんどそのようなことはせずARPによって動的に解決するようになっている。ARPによって解決された対応は各ホストで適当な時間キャッシュされるようになっていて余計なトラフィックを軽減している。キャッシュしておく時間は各OSの設定によりまちまちである。頻繁にハードウェアを変更しないネットワークであればキャッシュ時間を長くするのもひとつの手だ。キャッシュ内容はarpコマンドで確認でき、また対応を静的に指定することも可能だ。以下ではアドレスにホスト名が表示されているがこれは見栄えをよくするために、保持されているARPキャッシュではIPアドレスとなっている。

```
# arp
Address HWtype HWaddress      Flags Mask  Iface
athena  ether  00:C0:26:xx:xx:xx  C      eth0
```

```
gateway ether 00:A0:DE:xx:xx:xx  C      eth0
ceres   ether 00:90:CC:xx:xx:xx  C      eth0
```

RARPは現在ではほとんど使用されない。その昔ディスクレスネットワークブートマシンなどで使用されていたが、今ではそれはBOOTPやDHCPに取って代わられたので必要ないのである。



ネットワークの構成



各コンピュータ（ホスト）のネットワークを構成する際に設定する項目は、最低限IPアドレスとネットマスクである。そして必要に応じてデフォルトゲートウェイのアドレスやDNSサーバ、そしてホスト名、ドメイン名を設定すると思う。

IPアドレスはそのホストを特定するための一意なアドレスであることはこれまでの説明でわかると思うが、ネットマスクは何なのだろう？ 例としてIPアドレスに192.168.0.11、ネットマスクに255.255.255.0を指定した場合を想定する。

ネットマスク

ネットマスクの値はIPアドレスをネットワーク部とホスト部に分けて処理するために使用される。例ではネットワーク部が192.168.0、ホスト部が11となる（実際には2進数で考えなければならないが）、これらは各ネットワークセグメントを特定し、ルーティングをどのように行うかを指定するために利用される。ルーティングで各IPアドレスを個別に処理するとしたらあまりに面倒だし、各セグメントにランダムなIPアドレスが格納されたとしたらできなくはないけど管理の破綻は免れないだろう。そこで各セグメントをネットマスクによってネットワーク部を特定する方法で効率よく管理できるようにしているのだ。またホスト部はそのセグメント内で使用できるIPアドレスの数を限定することになる。例では0から255までの256のIPアドレスが利用可能だ（ホスト部のビットが全部0の0はマルチキャスト、全部1の255はブロードキャストに使用されることが通例となっているので実質は254個）。だから1セグメント内に254台以上のホスト機器を格納するにはホスト部を拡大する必要がある。これにはネットマスクを1ビット単位で左にずらしていけばよい。1ビット左にずらすことで約2倍のホスト部が得られることになる。ネットマスクを255.255.254.0にするとホスト部は192.168.0.0 ~

192.168.1.255、実質510個のIPアドレスがホスト機器で使用できるようになる。ネットマスクは左からどの位置までのビットが1であるかを表すものなので、ネットマスク255.255.255.0は192.168.0.0/24、255.255.254.0は192.168.0.0/23という表記でも表される。

ネットマスクは同一セグメントでは基本的に統一しなければならないが、場合によっては問題なく通信できてしまうこともある。たとえば192.168.1.2/24と192.168.1.10/23という設定を行ったとしても192.168.1.10から192.168.1.2へは問題なく通信することができる。それはネットワーク部が同一であると判定できるからだ。しかし192.168.1.10が192.168.0.11に通信しようとするとう問題が生じる。192.168.0.11はセグメントが異なるのでルーティングしないと通信できないのだが、192.168.1.10は同一ネットワークと判断し、ARPで解決しようとするのだ。でも解決できるわけないので通信できないということになる。

デフォルトゲートウェイ

デフォルトゲートウェイは各セグメント内に存在しないホストへのパケットをとりあえず送る先のホストのアドレスだ。通常デフォルトゲートウェイはルーティング情報に基づき適当なネットワークにそのパケットを転送することになる。

各コンピュータ(ホスト)はIPアドレスを得て通信を開始しようとするとき、そのアドレスが先ほどの自ネットワーク部の値と比較して同じであればARPでハードウェアアドレスを探し、そのアドレス宛に送信を行う、しかし異なる場合は自身が保有するルーティング情報を参照して適当なホストを見つける。そしてそのハードウェアアドレスを探して送信を行うようになっている。デフォルトゲートウェイは自身が保有するルーティング情報で解決できないときにとりあえずお任せするホストのことなのだ。

そのため通常デフォルトゲートウェイには実際に経路を接続しているルータを指定する。

直接通信できるのはセグメント内のハードウェアアドレスが直接解決できる相手だけで、それ以外の相手とはゲートウェイやルータと呼ばれるホスト経由でやり取りされるわけだが、デフォルトゲートウェイは必ずしも同一セグメントのホストである必要はない。ただしその場合はデフォルトゲートウェイにたどり着けるルーティング情報を各ホストが保有していなければならない。でも普通は各ホストの面倒をひとつひとつ見ていくのは無駄なので、デフォルトゲートウェイにネットワーク全体の状況をきちんと把握

している各セグメントに接したルータを指定するのだ。近頃はルータ同士で動的にルーティング情報をやり取りし、静的に指定しなくても最適な最新の情報を保持できるようになっているのでほとんどルーティング調整の手間はかからないようになっている。

なお各コンピュータのルーティング情報(テーブル)はrouteコマンドで表示/変更できるようになっている。

```
# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.0.0 * 255.255.255.0 U 0 0 0 eth0
default gateway 0.0.0.0 UG 0 0 0 eth0
```

このデフォルトゲートウェイは何も通常のホスト(コンピュータ)に限ったことではなく、ルータにも必ずといっていいほど設定されている。インターネットを構成しているネットワークを接続しているルーティングポイントでも、直接もしくは間接的に隣接するネットワークのネットワークアドレスとネットマスクの組み合わせによるネットワーク部とそれを処理可能なゲートウェイを示す上記のようなルーティング情報(テーブル)を基にどの経路に転送するかを決定するのだが、解決できない場合にはデフォルトゲートウェイに指定されたアドレス(普通は上位のルーティングポイント)にとりあえず転送するのである。

グローバルIPアドレスはNICなどで管理されていて日本ではJPNICなどが管理しているわけだが、上位のルーティングポイントではその割り当て状況によって経路を示している。たとえばOCNの専用線接続で8個のIPアドレスを割り当ててもらったとしても、インターネット全体でそのためのルーティング情報がやり取りされることはない。インターネット全体では日本のJPNIC管轄IPアドレス、そしてOCNに割り当てられたブロックの一部としてOCNの上位ルーティングポイントに到着し、そこからOCN管理のネットワーク部を参照してネットワークセグメントを特定して最終的な隣接ルーティングポイントまでパケットが配信されるようになっている。逆に割り当てられた一意のグローバルIPアドレスといえども勝手に別のポイントに接続したのでは、パケットは決してルーティングされない。



Linux マシンをルータに



ルータは2つ以上のネットワークを接続し適切に経路制

御を行う機器の総称だ。

Linux マシンをルータ (IP ルータ) にするには2つ以上のネットワークインターフェイスを装備するだけだ。イーサネット2枚でもよいし、イーサネットとPPPの組み合わせでもよい。カーネル構成も特に変更する必要はない。必要なのはカーネルのパラメータを1カ所変更するだけだ。

```
# cat /proc/sys/net/ipv4/ip_forward
0
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

このip_forwardはデフォルトでは0で、0以外を設定すると各インターフェイス間でパケットのフォワード、橋渡しを行うようになる。

単純に2つのネットワークセグメントを接続しただけであれば、ルーティング情報などは特別変更、追加する必要はない。一応マシンの状況を確認しておく。

```
# ifconfig
eth0  Link encap:Ethernet  HWaddr 00:C0:26:A5:02:B7
      inet addr:192.168.0.13 Bcast:192.168.0.255 Mask:255.255.255.0
eth1  Link encap:Ethernet  HWaddr 00:00:C0:63:68:81
      inet addr:192.168.1.2 Bcast:192.168.1.255 Mask:255.255.255.0
# route
Destination Gateway  Genmask         Flags Metric Ref    Use Iface
192.168.1.2 *        255.255.255.255 UH   0      0      0 eth1
192.168.1.0 *        255.255.255.0   U    0      0      0 eth1
192.168.0.0 *        255.255.255.0   U    0      0      0 eth0
default   gateway  0.0.0.0         UG   0      0      0 eth0
```

この状態でそれぞれのセグメントの適当なマシンのデフォルトゲートウェイ、もしくはルーティング情報をこのLinuxマシンのアドレス、192.168.0.13と192.168.1.2にしてみよう。

```
# route add 192.168.1.0 netmask 255.255.255.0
192.168.0.13
# route add default gw 192.168.1.2
```

セグメント192.168.0のコンピュータではルーティング情報を追加し192.168.1ネットワークへのパケットは192.168.0.13に送信する設定、セグメント192.168.1のコンピュータではデフォルトゲートウェイとして192.168.1.2を

指定する設定を行った。

挙動を確かめるために設定をひとつ終わるたびにpingコマンドでテストを行ってもよいだろう。片方のコンピュータにだけ設定した状態ではリクエストメッセージはターゲットに到着するものの返答が迷子になる状態などが観察できる。

#あるターゲットに対し異なるセグメントから行ったよ

```
# ping 192.168.0.11
PING 192.168.0.11 (192.168.0.11) from 192.168.0.13 : 56(84)
bytes of data.
64 bytes from 192.168.0.11: icmp_seq=0 ttl=128 time=0.3 ms
# ping 192.168.0.11
PING 192.168.0.11 (192.168.0.11) from 192.168.1.10 : 56(84)
bytes of data.
64 bytes from 192.168.0.11: icmp_seq=0 ttl=127 time=0.6 ms
```

このテストではTTLの値と要した時間に注目してほしい。TTLは経路を通過するたびに1ずつカウントが減られていくホップカウントと呼ばれるもので、ルーティングされた回数、通過したネットワークセグメントの数がわかる。ここではルーティングされた回数は1回なので128から127へ1減っている。また時間は2倍を要していることがわかる。まあ当然の結果だ。

次に先ほどの後者のテストのネットワーク上でのやり取りを見てみることにしよう。

```
arp who-has 192.168.1.2 tell 192.168.1.10
arp reply 192.168.1.2 is-at 0:0:c0:63:68:81
192.168.1.10 > 192.168.0.11: icmp: echo request (DF)
192.168.0.11 > 192.168.1.10: icmp: echo reply (DF)
```

通信に先立ちデフォルトゲートウェイのハードウェアアドレスをARPで探し、ICMPのエコーリクエストを行っている。このログでは表示していないが、実際のパケットではIPアドレスは192.168.0.11だが、イーサネットレベルのパケット(フレーム)に付けられたあて先ハードウェアアドレスはデフォルトゲートウェイのものだ。

ルーティング先のネットワークではルータがターゲットのハードウェアアドレスを解決しパケットを転送している。

```
arp who-has 192.168.0.11 tell 192.168.0.13
arp reply 192.168.0.11 is-at 0:a0:de:8:c4:d3
192.168.1.10 > 192.168.0.11: icmp: echo request (DF)
```

```
192.168.0.11 > 192.168.1.10: icmp: echo reply (DF)
```



IP パケットフィルタリング



IPパケットフィルタリングはルーティングの中間で行うパケットに対する処理ということになる。各インターフェイスに入ってくる時、出て行くときなどのそれぞれのタイミングで処理を行うことができる。2つのネットワークを接続している場合は少なくとも4カ所のチェックポイントがあることになる。詳細は次回説明するとして下準備を少しだけ紹介しておく。

特に変更を加えなくてもルーティングはできるようになったわけだが、次のIPパケットフィルタリングやNAT、IPマスカレードの段階へ移るにはカーネルの再構成やツールを入手しなければならない場合がある。一応ここではカーネルは2.4.0-test1以降を前提としており、カーネル構成で有効にする項目は以下のとおりだ。IP: Netfilter Configurationではとりあえず全部モジュールとしてコンパイルしておく。あとコンフィギュレーションツールのiptablesは<http://netfilter.filewatcher.org/iptables-1.1.0.tar.bz2>などから入手してコンパイルしておこう。

```
Networking options --->
[*] Network packet filtering (replaces ipchains)
    IP: Netfilter Configuration --->
<M> Connection tracking (required for masq/NAT)
(以下省略)
```

すぐに試してみたいと思うので、PPP接続を内部ネットワーク全体で利用するためのごく簡単な設定（IPマスカレード）を紹介しておく。

```
# modprobe ip_tables
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

DNSやデフォルトゲートウェイ、ルーティング情報の設定などは別途行う必要があるが、これでこのコンピュータのPPP0インターフェイスのIPアドレス（IPマスカレード機能）を利用して内部ネットワークはインターネットなどを利用できるようになる。内部ネットワークはプライベートIPアドレスを使用していれば、インターネットなどの外

部ネットワークから侵入されることはあまり考えなくてもよい（途中のルーティングでそんなアドレス宛のパケットは捨てられる）が万全を期するために以下の設定も行う。

```
# iptables -A INPUT -i ppp0 -m state --state NEW,INVALID -j DROP
# iptables -A FORWARD -i ppp0 0 -m state --state NEW,INVALID -j DROP
もしくは
# iptables -A INPUT -i ppp0 -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A FORWARD -i ppp0 -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT -i ppp0 -j DROP
# iptables -A FORWARD -i ppp0 -j DROP
```

これは通信の状態によるフィルタリングだ。これでPPP0インターフェイスに対するINPUTとFORWARD要請の不正な接続要求を排除することができる。前者ではstateで指定したNEW（外部を発端とする接続）とINVALID（不正な接続）を捨てている。後者ではESTABLISHEDとRELATEDという内部を発端とする外部接続通信のみ通過を許可し、それ以外は許可しないという設定だ。後者のほうがより安全なんだけど、中にはこの設定だと問題になる接続もあるかもしれないので注意すること（FTPで問題になる可能性がある）。なおNEWやESTABLISHEDはTCPのみで使用される通信を確立する際のSYN、SYN/ACK、ACKというフラグを意味する。すなわちNEWというのはSYNフラグのみが付いているパケット、ESTABLISHEDはそれ以外、すでに通信が確立されていることを表している。

Windowsネットワークがある場合、その関連パケットを外に出したくないので以下の設定も行っておこう。内側のネットワークに入ってくる段階でTCP、UDPのそれぞれdport、sportの両方でNetBIOS関連137～139を破棄してしまうわけだ。FORWARDだけでも大丈夫だが、必要に応じてINPUTにも同様のフィルタを登録してもよい。

```
# iptables -A FORWARD -i eth0 -p TCP --sport 137:139 -j DROP
# iptables -A FORWARD -i eth0 -p TCP --dport 137:139 -j DROP
# iptables -A FORWARD -i eth0 -p UDP --sport 137:139 -j DROP
# iptables -A FORWARD -i eth0 -p UDP --dport 137:139 -j DROP
# iptables -A INPUT -i eth0 -p TCP --sport 137:139 -j DROP
```


Linux Garbage Collection

目からウロコの用語辞典

文：しのはらひろあき
Text：Hiroaki Shinohara

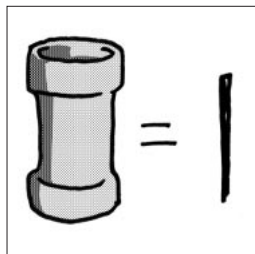
第5回

- 【パイプ】(ばいぶ)
- 【フィルタ】(ふいるた)
- 【head】(へっど)
- 【tail】(ている)

パイプ

【ばいぶ】

プロセス同士でデータのやりとりをするためのしくみ。bashなどのシェル上では、「パイプ記号」の「|」を使うことでコマンド間でパイプが確立され、データのやりとりが可能になる。その「管」のような役割が、現実世界の「パイプ」によく似ているため、この名前と呼ばれる。よくわからない人が使うと、ときどきエラーを起こして詰まるのも現実のパイプと同じ。



パイプ(そのまんま)

フィルタ

【ふいるた】

標準入力やファイルからデータを読み込んで、なんらかの加工をし、そのデータを標準出力や指定ファイルに書き出すよう作られた、単純なプログラムのこと。複雑な作業も、小さなフィルタプログラムを「パイプ」で連携させて処理させることができる。Linuxでは、この「フィルタ文化」が長く受け継がれてきた。そのため、/binや/usr/binには「head」

や「tail」など、細かい処理を行う専用のフィルタプログラムが多数用意されている。ちなみに、極端に多機能なフィルタプログラムは、よく押し売りに来る浄水フィルタと一緒にあまり好まれない。

～ Linux 劇場「セールスマンは2度biffを鳴らす」～

私は美代子。28歳。毎朝主人を送り出したあとは、家事とLinuxにはげむ、ごくふつうの専業主婦。働かなくても生活が保証されている主婦って、他人からはディスク容量にもCPUパワーにも困らない幸せな生活を送っていると思われがちだけれど、でも何かものたりない...
...。男がやってきたのは、そんな漠然とした不満に悶々としていたある昼下がりのことでした。



セールスマンは2度biffを鳴らす
利奈楠企画

ピンポン。

「あら、誰かしら。はーい、今出ます～」

ガチャ。

そこにはスーツ姿の若い男が。ちょっとハンサムな.....。ふだん、男といえば主人ばかりの私は、思わずときめいてしまったんです。それが間違いの元。気を許したすきに、男はずかずかと私のホームディレクトリに上がりこんできました。彼は、最近よく噂に聞くフィルタ販売のセールスマンだったのです。

「いやー、奥さん。今日はいいフィルタが入ったんで、ぜひご紹介したいと思ひましてお伺いしたんですよー。これひとつで奥さんのLinuxライフもグレードアップ間違いなし！なにせ、あのLinus Torvaldsが開発したというオープンソースなフィルタ！品質は折り紙つきです。それが今なら月々

たったの4980円ですよー」

「あの……。うち、フィルタはPerlで自作と決めていますので……」

「そんなこといわずに～。奥さん。一度試せば、その機能に思わず病みつき！ 間違いなしですよー」

白い歯と小麦色の肌の、魅力的な男です。ああ、一度くらい、こんな男にファイルシステムをマウントされてみるのも……。などと妄想にふけていると、なんと彼は勝手にわたしのLinux Boxをまさぐって、インストール作業を始めてしまったのです！

「じゃあ、インストールしてしまいますよー。ほらっ！」

「ああっ！ そんな、いきなり……」

「見てください。標準入力から読み込んだテキスト各行のナンバリングも、逆順ソートもオプションひとつで自由自在！ しかも、この多彩な機能を実現しているのが、たったひとつのシェルスクリプトなんです。どうです？ 欲しくなってきましたでしょう」

「あああ……」

男の巧みな話術とキーボードさばきに、私はしだいに恍惚感をおぼえはじめていました。でも、そのとき、私を愛してくれている誠実な夫の笑顔が脳裏に。はっ、と我にかえった私は……。

「ダメっ！ イヤですっ！ やめてくださいっ！」

「へへへ。口ではイヤと言っている、奥さんのコマンドラインは正直だねえ。ほうら、bashがどんどん入力を補完していったるじゃありませんか……」

「いやっ、そんなウソよ！」

などといいながらも、男の指摘の正しさに私はいらだっていました。意志とは裏腹に、Linuxはファイル探索パスへ男のフィルタをsetしていきます。ああ、このまま落ちていくのかしら……。

ガチャ。

「ただいま……。アッ、おまえは誰だ！ 勝手に上がりこんで美代子のLinuxに何してるんだっ！」

「あなたっ！ 助けてっ！ この人がっ！」

「へっへっへ。これはこれは。美代子さんのご主人ですか」

「妻にこんなことを……。出ていけっ！ さもないとrootを呼ぶぞっ！」

「いいんですか？ そんなことをいって。いま私が帰ったら、このページはネタ切れで埋まらなくなってしまいますよ！ なにせ筆者は、“フィルタ”なんて難しいテーマで書くことを編集者と約束しておいて、まったく筆が進まないことに困りきっているんですから」

「うーん、それはまずいな……」

(続かない)

head

【へっど】

入力されたテキストデータから先頭数行だけを読んで出力する、代表的なフィルタプログラム。なんだかわからない巨大なファイルや、いちいちぜんぶ読むのが面倒なファイルの中身をちょっと確認するのに便利。このコーナー担当の編集者は、筆者の原稿をheadで読んでいるらしい。文章の後半に行くにしたがって間違いが増えるのは、実はこのためである(編注:ウソです。wcで行数だけ確認しています)。

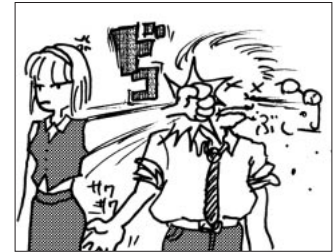
例) .cshrcの先頭10行を読む

```
% cat .cshrc | head -10
```

tail

【ている】

任意のファイルを尻(テイル)からなめまわすように操作する、セクハラフィルタプログラム。headの逆。だったらなぜ「foot」じゃないのかというツッコミは、Linuxの世界では御法度となっている。



テイル返し

例) .cshrcの最後の10行を読む

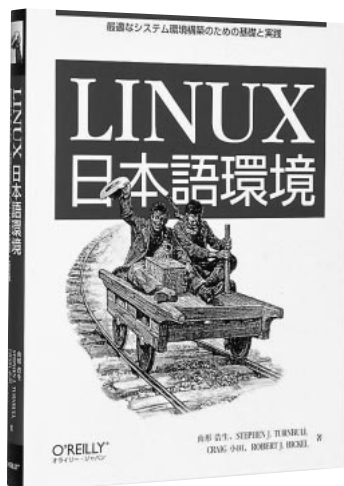
```
% cat .cshrc | tail -10
```

「-f」オプションをつけると待機状態になって、対象ファイルの末尾に新たなデータがつけ加えられるたびに読みとって出力する。これを応用して下記のようにメールスプールを読むとbiffコマンドの代わりになる。

```
% tail -f /var/spool/mail/foobar &
```

超便利！ 筆者のオリジナルアイデアなので、みだりにマネしないこと(編注:本当か?)

Books



Linux 日本語環境

山形浩生 / Stephen J. Turnbull / Craig 小田 / Robert J. Bickel 著
 オライリー・ジャパン
 B5判変形 / 376ページ

本体価格 3600円

先月号に続いて、Linuxにおける日本語環境に関する1冊。日本語を母国語とする人がコンピュータシステムを使いこなそうとすると、至極当然ではあるけれど、そのシステムでの日本語の扱いというのは避けては通れない問題だ。最近の日本語ディストリビューションは、デフォルトの設定でも十分に日本語を処理できるようになっているが、フォントを追加したり、日本語入力システムを変更したりする際には、ユーザーの側で設定しなければならない部分も多い。

本書は、文字コードセットの国際標準化、ディストリビューションでの日本語化の取り組みといったベーシックなトピックから、X Window Systemの設定、ユーティリティやアプリケーションでの日本語の使用まで幅広く取り上げている。特に日本語が使用できるツールやアプリケーションの紹介に多くのページを割いていて、実用度の高い内容となっている。

Linuxシステムコール

塚越一雄 著
 技術評論社

A5判 / 336ページ

本体価格 2480円

システムコールとは、ファイルやデバイスへの入出力操作、プロセスの制御といったカーネルの機能を利用するためのプログラミングインターフェイスで、UNIXのユーザープログラムからこれらの機能にアクセスするためには欠かせないものだ。C言語のライブラリ関数の内部でもシステムコールが呼び出されている。つまり、UNIXでプログラミングをするにあたっては、システムコールを理解しておくことが望ましい。

システムコールにはUNIXのバージョンごとに方言があり、たとえばLinuxとFreeBSDでは互換性がない。このため、C言語の解説書などでシステムコールについて触れられている場合でも、必要としている知識が得られないことがある。本書はLinuxのシステムコールに的を絞って、その基礎を解説しており、Linuxのプログラマーがシステムコールについて学ぶにはうってつけの教科書だ。



GIMP エッセンシャルテクニック

向井領治 / 古川康弘 共著
 オーム社

B5判変形 / 336ページ / CD-ROM1枚付き

本体価格 2800円

GIMPはLinuxで使える高機能なフォトタッチソフトというだけでなく、Linuxのアプリケーションソフトウェアを代表する存在だ。「まだまだ使えるアプリケーションが少ない」というLinuxに対する評価もある中、ワープロや表計算ではなく、ある意味特殊なグラフィックス処理の分野で、これだけ高機能なフリーソフトウェアが開発されたのはなんだか不思議な感じもする。もちろん使う側からすれば、優れたソフトは大歓迎である。

さて、せっかくの優れた機能も使いこなせなければ意味がないというわけで、本書の紹介である。各機能の使い方だけでなく、サンプル画像を紹介しながら、それぞれの機能を使いこなすうえでの注意点を説明するなど、「勘所」を押えた内容になっており、これからGIMPを使い始めようというユーザーにお勧めの1冊だ。対象となるGIMPのバージョンは、1.1.18。CD-ROMにはUNIX (Linux) 版とWin32 (Windows) 版のGIMPが収録されている。



インターネット事件+事故レポート1999 アクトンファイル2000

アクトンファイル作成委員会 編

サイビズ

B6判変形 / 160ページ

本体価格 980円

インターネットの急速な普及に比例するように、それに関連する事件もまた増え続けている。クラッキングやコンピュータウイルスによる被害など、ネットワーク上で発生するものはもちろん、ネットワークを利用した詐欺、違法販売、プライバシー侵害・名誉毀損といった犯罪行為など……。今年に入ってから、佐賀県のバスジャック少年による掲示板への犯行予告ともとれる書き込みなどがあり、「犯罪の影にインターネットあり」という様相さえ呈してきているのが現状だ。

本書はタイトルのとおり、1999年に発生したインターネット関連の主な事件・事故をまとめたレポートである。それぞれの事件の概要と背後関係が簡潔かつ客観的にまとめられていて、キーワード解説もかなり充実している。情報を整理して頭に入れておきたい向きには有用だ。少々不謹慎かもしれないが、読みものとしても楽しめる1冊である。

UNIXコマンド例文事典 sh/bash編



OfficeMASA 著

技術評論社

A5判 / 304ページ

本体価格 2180円

mule/viスーパーリファレンス



伊東真人 / 小巻賢二郎
田谷文彦 / 前田雄一郎 著

ソフトバンク
パブリッシング

A5判 / 336ページ

本体価格 2200円

UNIXセキュリティツール



Seth T. Ross 著
トップスタジオ 訳
坂井 順行 監修

翔泳社

B5判変形 / 272ページ

本体価格 2800円

LinuxのためのXFree86入門



Aron Hsiao 著
コスモプラネット 訳

アスキー

B5判 / 424ページ

本体価格 3800円

読者の声

俺にも
いわせろ!

夏真っ盛り!! 海に山に繰り出しましょう! 白い雲、青い海、力強い緑をたたえた山。いいですね~。現実はず。白い肌、青い顔、未編集原稿の山。しくしく。とりあえず、ビールと枝豆っ。

8月号特集1へのお便り

「Linuxレスキュー 疑問・難問・200問」はなかなかためになりました。ちょっとしたことが、案外わからなかったりするので、いい内容でした。

(福井県 服部昌博さん)

「Linuxレスキュー」はとても参考になりました。今後ともよい誌面づくりを期待しています。

(兵庫県 高峰 章さん)

「Linuxレスキュー」には助けられました。パッケージ付属マニュアルだけではわからないことも多いのです。LinuxはSolarisやHP-UXとはちょっと違うことがわかり、有益でした。

(東京都 島立隆男さん)

④8月号の特集1には、たくさんの反響をいただきました。担当した編集部員たちも喜んでおります。この特集では、編集部員たちも知らなかったことがポロポロと出てきて、「な~るほど」とか、「そうだったのかぁ」という声飛び交っていました。

今後も、お役に立てる記事づくりを心が

けてまいります。

ワイヤレス時代はじまる

無線LANの特集がよかったです。実践的でとてもわかりやすかった。有線LANとの速度比較を踏まえると、無線LAN導入の参考になります。

(三重県 坂倉孝明さん)

⑤担当は、自分の部屋と階下の間で無線LANを使用しています。1世代前の転送速度が2Mbpsの製品なのですが、インターネット接続の回線速度がボトルネックになるため、Webを見たりする分には遅さを感じません。さすがに数百Mバイトのファイル転送は恐くて試せませんが、とても便利です。

11Mbpsの製品も、出そろってきましたね。価格も2Mbpsのころより安いのでちょっとくやしい気もします(笑)。

ネットワークは1日にしてならず?!

何につけてもネットワークの設定が難しい。スタンドアロンな頭だからだろうか。そもそも私の頭はシングルタスクなのだけれども。

(大阪府 松本友宏さん)

⑥ネットワーク周りは設定が複雑ですよね。ある程度、TCP/IPに関する知識がないとドキュメントを理解するのもひと苦労です。しかも、サーバプログラムごとに設

定ファイルの書式が違う……。担当はネットワーク関係の設定で何度徹夜をしたかわかりません。

苦労はしても、うまく動いたときの嬉しさはたまりません。また、各プロトコルが巧妙な原理で動作しているのには感心しっぱなしです。

Rogueで今夜も眠れない

今回のRogueの記事はよかったです。おかげでかなり睡眠時間が削られます。

(愛知県 今枝卓也さん)

Rogueのコーナー、よかったです。Rogueはやりたくても、よくわからなくて……。BNNの『Rogueハンドブック』も探したことがあるのですが、すでに絶版でした。Webで見られるようになるとは知りませんでした。

(東京都 野尻 佐智子さん)

⑦RogueやNetHackには不思議な魅力がありますね。Rogueをプレイしていて、また遊んでいると言われたら、viの練習だといいわけすればいいのだ、とか、C(ケンタウロス)を見たら急いで逃げろとか、いろいろな話が伝わっています。

名作と呼ばれるものは、時代を超えて楽しめるものなのですね。

ご要望もお待ちしております

Netscape Navigator、XEmacsなど

の日本語フォント設定についての解説を希望します(特にTrueType)

(北海道 須賀英夫さん)

連載を単行本にして出してください。

(千葉県 李明哲さん)

④連載の書籍、ムック化も進めてまいります。Linux magazine booksシリーズの第1弾として、『PC UNIXの日本語処理と設定』(本体価格1700円)が発売されました。本書は、Linux magazineで連載していた、「Linuxの日本語環境」をもとに、加筆・修正したものです。また、ネットワーク関連の特集、連載をまとめたムック『Linuxネットワークサーバ構築ガイド』(本体価格1800円)を8月28日に発売する予定ですので、こちらもよろしく願いたします。

コンサバのパッケージ

Kondara MNU/Linux Server 1.1の堅いパッケージを見ていると、暴走族が足を洗ってしまったように感じられて、ちょっと淋しいです。

(神奈川県 藤原干城さん)

④デスクトップ用途向けのKondara MNU/Linux 1.0 / 1.1が白を基調として、Kondaraペンギンをドーンとあしらっていたのに対し、Kondara MNU/Linux Server 1.1はマットブラックの渋いパッケージで登場したので驚いた方も多いのではないのでしょうか。実は、パッケージの裏面にこっそりとKondaraペンギン君がいます。ネクタイを締めて、憂鬱そうな顔をしているのが笑えます。

暑さに負けるな!

クーラーのない自分の部屋にいると、

PCより先に頭が熱暴走しそうです。

(埼玉県 室井武雄さん)

④今年は猛暑です。暑くて頭が腐りそうだとよく言ったものだな、なんてバカなことを考えていたら、担当の自宅マシンがトンでしまいました。冷却不足で、マザーボードが摂氏50度、CPUは摂氏65度にもなっていたのです。仕方がないので、ケースやファンを新調したのですが、部屋の温度が摂氏40度にもなるので効果が上がりませんでした。

結局、雨戸を閉めて、エアコンの24時間運転に踏み切ってしまいました。なんか間違っているような気もしますが、暑くて頭が回らないので、いいことにします(^_^;;;

アンインストールはドウする?

RPM形式ばかりでソフトウェアをインストールしているので、tar.gz、zip形式で配布されているソフトウェアをインストールすると、アンインストールする方法がわかりません。

(福岡県 SAMSUN BAHARINさん)

④このような形式で配布されているソフトウェアの中には、アンインストールもできるようにMakefileを作っているものもあります。この場合は、rootユーザーになって、make uninstallなどとすれば、インストールされたファイルを削除してくれます。lessなどで、Makefileをのぞいてみてください。

Makefileがアンインストールを考慮していないときは、インストールされたファイルをひとつひとつ手で削除するしかありません。

make -n installとすると、make installで実行される内容が表示されますの

で、ファイルのインストール先がわかるでしょう。

ほげほげ遠藤さん感染す!

どーも、ほげほげ遠藤です。8月号は「ほげ」の特集、誠にありがとうございました。編集部のみなさんも、特1担当の方も、ご苦労様でした。掲載リンク先を見てびっくりこきました! 「ほげ」を探求している方がいらっしゃるのですね。なるほど、大変奥が深く、すごいひとことです。あまりの「ほげ」の偉大さに、自分のペンネームに使おうなんて、とんでもない。またペンネーム考えます。そんな折、私はいつの間にか「ほげ症候群」に感染していました。編集部の方々は何人感染しましたか? 感染率が高いとの噂ですのでご注意ください。それでは.....。

(山形県 ホゲホゲ遠藤さん)

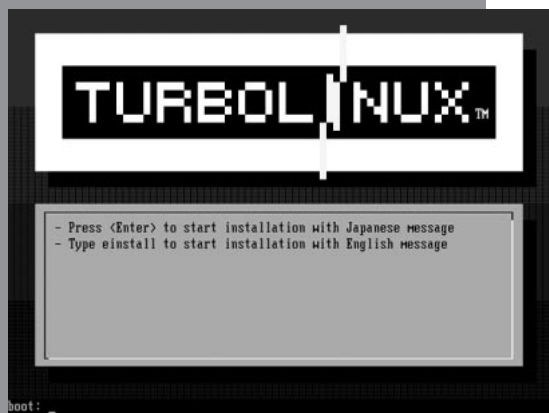
④編集部には、こういう話が好きな人間が3人ほどいます。もちろん3人とも感染しました。そのうちのひとり、8月号の特集で「ほげ」の謎に迫った編集部員(にゃー@編集部)は、ときどき人を担ぐことがあるので注意が必要なのです。先日、数人の編集部員で、ターボリナックス ジャパンさんにいただいたペンギンぬいぐるみは、なぜ首が灰色なんだろうと話していると、「それはアゴシタハイイロペンギン。英語では通称グレイジョウと呼ばれている種類のものだ」と解説してくれました。もちろん、真っ赤なウソでした。ホゲホゲ遠藤さん、新たな謎を発見したら、ぜひお知らせください。ナイスな新ペンネームにも期待しております。

担当は今からビアガーデンへいきます。では来月またお会いしましょう。

付録CD-ROMに収録した

TurboLinux Server 日本語版6.1 のインストール

本誌付録CD-ROM収録のTurboLinux Server 日本語版6.1はフリー版です。製品版に含まれる商用ソフトは含まれていません。また、製品版を販売しているターボリナックス ジャパンからサポートを受けることはできません。



インストーラの起動

インストールするマシンがCD-ROMから起動できる場合は、付録CD-ROM1からブートします。CD-ROMから起動できない場合は、WindowsかDOSでインストールフロッピーを作成します。作成手順は次の通りです。フォーマット済みのフロッピーディスクをAドライブに入れ、CD-ROMドライブの¥DOSUTILS¥BOOT.BATを実行します。同様に、¥DOSUTILS¥EXTRAHW.BATを実行すれば追加デバイスディスクを、¥DOSUTILS¥PCMCIA.BATを実行すればPCMCIAサポートディスクを、それぞれ作成できます。

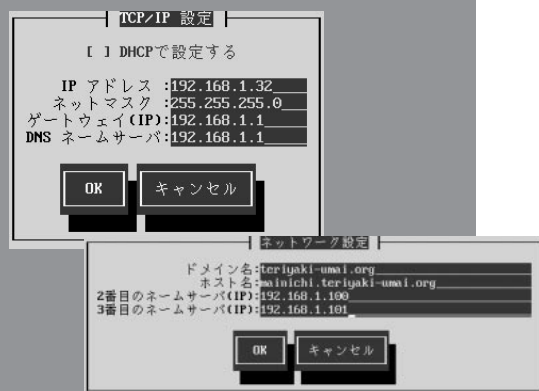
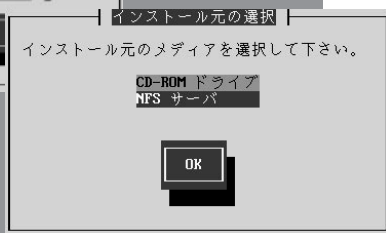
付録CD-ROM1、あるいは作成したブートフロッピーからマシンを起動すると左のような画面になります。画面下部の「boot:」プロンプトに対し、Enterを入力するとインストールが始まります。日本語のフォントが化けてしまうときは、「boot:」プロンプトに対し、「eininstall」と入力すると、英語表示でインストーラが起動しますので、試してみてください。



キーボード、インストール元メディアの選択

インストーラが起動すると、最初にキーボードの種類を聞いてきます。日本語106 / 109キーボードを利用しているなら「jp106」を選びます。

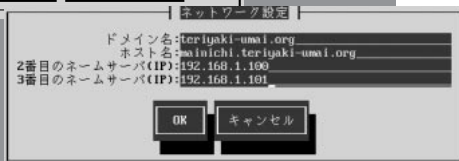
次に、インストール元となるメディアを選択します。ここでは、付録CD-ROMを利用するので「CD-ROMドライブ」を選択します。



ネットワークの設定

ここでは、TCP/IPの設定を行います。環境に合わせ、IPアドレスやネットマスクなどの情報を入力します。DHCPサービスを利用できる場合は、スペースキーで「DHCPで設定する」にチェックをすれば、マシンの起動時にIPアドレスなどを自動設定できるようになります。

ネットワーク設定の画面では、ドメイン名やホスト名を設定します。「DHCPで設定する」を選択したときは、DHCPサーバから取得した情報が表示されますので、内容を確認しましょう。

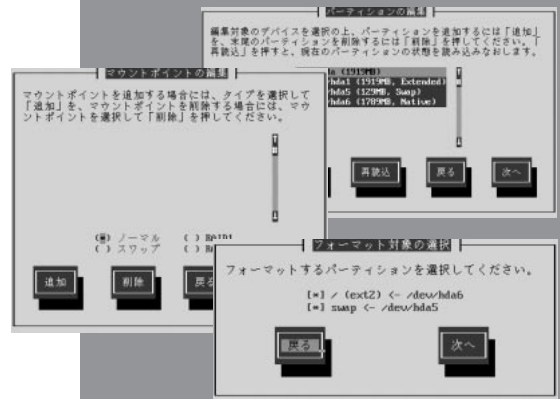


ハードディスクの設定

Linuxをインストールするハードディスクの設定を行います。モード選択の画面では、パーティションの設定を自分で行うか、インストーラに任せるかを選択します。ここでは「マニュアルモード」を選んだものとして解説をします。

最低でもルートパーティションと、スワップパーティションの2つが必要になりますので、「追加」を押してこれらのパーティションを作成します。スワップパーティションは128Mバイトほどでよいでしょう。

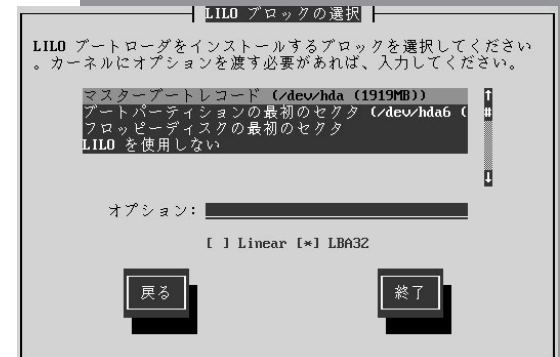
パーティションを作成したら、マウントポイントの編集を行います。「() ノーマル」にチェックしてから「追加」を押して、作成したパーティションをどのマウントポイントに割り振るかを設定します。スワップパーティションは、「() スワップ」にチェックを入れて追加します。設定が済むと、パーティションのフォーマットを行います。フォーマットしたくないパーティションがある場合はチェックを外します。



LILOブロックの選択

Linuxで標準的に使われているブートローダのLILO (Linux LOader) の設定を行います。Linux専用機にするなら「マスターブートレコード」を選びます。DOSやWindows 95/98とデュアルブートにするなら「ブートパーティションの最初のセクタ」を選びます。インストールするマシンのBIOSがLBAモードをサポートしているなら「[] LBA32」にチェックを入れておきます。これによって、Linuxのカーネルイメージがハードディスクの1024シリンダ以降に置かれていてもブートできるようになります。

ファイルシステムを作成したら、インストールするパッケージ内容の設定に進みます。



インストールタイプの設定

インストールするパッケージを選びます。サーバの利用目的別にいくつかのインストールタイプが用意されています。また、個別のパッケージを自分で選ぶことも可能です。F1キーを押すと、各パッケージタイプの説明が表示されます。

ここでは、すべてのパッケージがインストールされる「オールインワン」を選んだものとします。この場合、およそ850Mバイトのディスクスペースが必要になります。

パッケージのインストール状況は棒グラフとして表示されますので、インストールが済むまでしばらく待ちましょう。



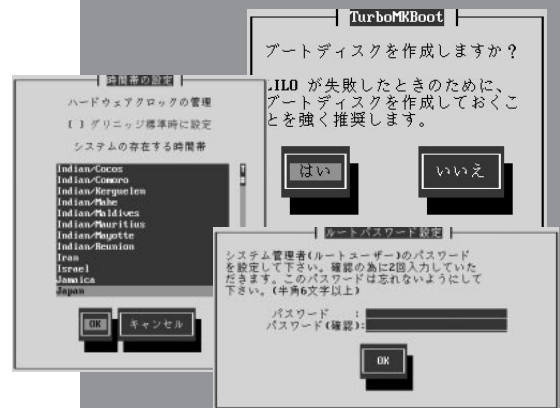
ブートディスク作成、タイムゾーン、パスワードの設定

パッケージのインストールが終わると、ブートディスクの作成画面になります。LILOからのブートに失敗したときなどは、ブートディスクが必要になりますので、作成することを強くお勧めします。

次に、タイムゾーンの設定を行います。日本国内で使う場合は「Japan」を選びます。

続いて、rootユーザーのパスワードを設定します。半角6文字以上のパスワードを、2つの設定欄それぞれに入力します。セキュリティ保持のため、パスワードの入力時にエコーバックはまったく行われません。慎重に入力してください。

ここまでで、基本システムのインストールが終わりました。システムがシャットダウンされますので、CD-ROMやフロッピーディスクを抜き、Enterキーを押してマシンを再起動します。リポート後に、X Window Systemの設定を行います。





X Window Systemの設定 (キーボード、マウス)

マシンが再起動すると、最初に拡張インストール画面となり、追加CD-ROMを持っているかどうかたずねてきます。付録CD-ROMでインストールしている場合は、「いいえ」を選びます。

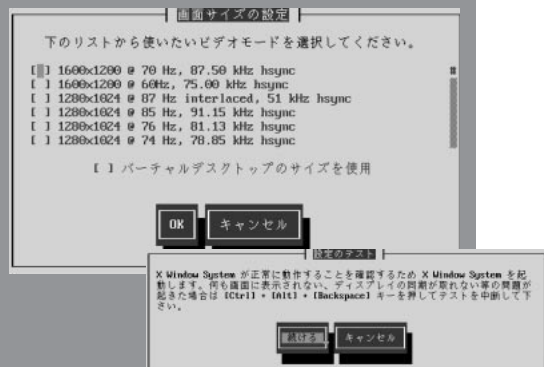
ここから、X Window Systemの設定が始まります。まず、キーボードとマウスの種類を選びます。2ボタンのマウスを使っている場合は「[] 3ボタンのエミュレーション」にチェックをすると、左右ボタンを同時に押すことで、3ボタンマウスの中ボタンと同様の働きをさせられるようになります。



グラフィックスカード、ディスプレイの選択

ここでは、グラフィックスカードの自動認識をすることがたずねられます。よほど古いか、特殊なカード以外は自動認識が可能ですので、「はい」を選びます。左のように、認識結果が表示されますので内容を確認しましょう。正しく認識されている場合は、検出データ画面で「検出値で設定」を選びます。正しくなければ「手動設定」を選んで設定しなおします。ビデオメモリのサイズなどが判定できなかったときは、設定画面になりますので、手動で設定します。

グラフィックスカードの設定が済んだら、ディスプレイの設定です。メニューに使用しているディスプレイがあれば、それを選びます。メニューにない場合は「カスタムディスプレイ」を選び、ディスプレイのマニュアルを参考にして水平/垂直同期周波数を入力します。ディスプレイが追従できない数値を設定すると、故障の原因となることがありますので気をつけてください。



ビデオモードの選択

X Window Systemで利用する画面色数、画面解像度を設定します。まず、画面色数を選びます。次に利用したいビデオモード(解像度とリフレッシュレート)を選びます。複数のビデオモードを選ぶことも可能です。この場合、Ctrl+Alt+プラス(マイナス)キーでモードを切り替えて使うことができます。

デフォルトとなるビデオモードと、画面フォントの解像度を選んだあとで、設定に基づいてX Window Systemを起動します。「Next」を押すと、ビデオモードが切り替わります。すべてのビデオモードを確認したら「Quit」を押してテストを終了します。画面に何も表示されないときは、Ctrl+Alt+Back Spaceキーを押せば中断することができます。



ログイン方法、ウィンドウマネージャ、サービスの選択

Linux起動時に、コンソール画面を表示する「テキストログイン」と、X Window Systemを起動してログインを待つ「グラフィカルログイン」からログイン方法を選びます。次に、ウィンドウマネージャを選択します。デフォルトではGNOME環境が選ばれています。

最後に、システムの起動とともに開始するサービス(デーモン)を選びます。必要のないデーモンは起動しないようにしましょう。

以上でインストールが終わりまりました。インストーラが終了すると、ログイン画面になります。