

NEWS EXPRESS

Distribution

Software

Hardware

Headline

Event

目指せ！Xプログラマー!?

米Intelが、写真のようなインターネット端末「Dot.Station」を6月22日に発表した。Webブラウザ、電子メール、電話機能を持っており、年内にはECやオンライントレードなどのサービスプロバイダから提供されるというから、Webやメールを利用したい非PCユーザーがターゲットだ。たぶんDot.Station自体は無料かそれに近い金額で販売されると予想される。

CPUにCeleronが使われ、ハードディスクを内蔵するというから、中身は普通のPCとたいして変わらないはずだ。アプリケーションのバージョンアップなどはネットワーク経由で行われるという。そして、OSにはLinuxが採用されている。

実物に触ったわけではないが、X上で動作する専用のアプリケーションプログラムが立ち上がり、ユーザーインターフェイスを司っていると思われる。

ところで、こういう製品が増えると、そのアプリケーションを作るエンジニアが必要になってくる。X Window Systemのプログラミング技術が要求されるようになるのだ。

Linux (UNIX)の世界では、従来(現在も?)はコマンド使いが偉かったのだが、今後はXプログラマーがもてはやされる時代になるかもしれない。Windowsアプリケーションを増やすため、マイクロソフトはVisual BASICやVisual C++といったGUI中心の開発環境を整えてきたように、現在まだ貧弱なLinux上のGUI開発環境が今後急速に進歩していくことだろう。



Hardware

発売日

2000年6月12日

インターネット・アプライアンスサーバ
PowerAppシリーズURL <http://www.dell.com/jp/>

デルコンピュータは、高性能アプライアンスサーバ「PowerApp (パワーアップ)」シリーズを6月12日より発売した。高さ1U (4.4cm) サイズのWebサーバ向け「PowerApp.web 100」とインターネット・キャッシュサーバ向け「PowerApp.cache 100」、同2U (8.8cm) サイズの「PowerApp.cache 200」がある。CPUはPentium (600 / 650 / 700 / 733MHz) ハードディスクはSCSIインターフェイス、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。

カスタマイズされたRed Hat Linux 6.2 + Apacheを搭載し、「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。

PowerAppシリーズは、Webホスティング、サイトへのアクセス負荷を軽減するインターネット・キャッシング、あるいは負荷を分散するロード・バランシングなど、特定のサービスを提供する単機能サーバで、あらかじめ組み込まれたOSとアプリケーションにより、その機能を提供するようにデザインされている。

そのため、特に複数台のサーバに対して設定や管理を行う必要のあるISPやASPなどが、導入時の設定やシステムの管理・保守が容易に行えるように設計されていて、デル独自のKick Startユーティリティを使い、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。



PowerApp.web 100

発売	デルコンピュータ株式会社
TEL	044-556-6190
価格	20万8000円～

Software

発売日

2000年6月7日

Linuxサーバで稼動するeCRMソフトウェア
HDE Customers Care 1.0URL <http://www.hde.co.jp/>

ホライズン・デジタル・エンタープライズは、eCRM (Internet-based Customer Relationship Management) ソフトウェア「HDE Customers Care 1.0」(以下Customers Care)を6月7日より発売した。

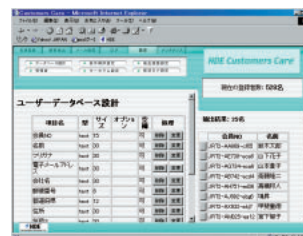
Customers Careは、電子メールを利用したマーケティング機能を持った顧客データベースシステムで、顧客の属性と顧客の電子メールアドレスを統合管理できるインターフェイスを備え、時間指定メール配信、特定の属性に一致する顧客に対する一斉メール配信などの機能を持っている。

また、Webサーバに顧客登録フォームを用意し

て、インターネット上で顧客登録システムとして利用することができ、顧客の属性の登録は、Webブラウザで自由に追加 / 設定が可能である。

対応ディストリビューションは、Red Hat Linux 6.2、TurboLinux Server6.1 (予定) で、顧客データベースにはPostgreSQL、WebサーバにはApacheとPHPを利用する。

Customers Careの価格は、データベース1件、3000レコード、10フィールドまでのLite版が9万8000円、5件、10000レコード、30フィールドまでのStandard版が49万8000円、それ以上はEnterprise版となりオープンプライズである。



発売	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
価格	9万8000円～

Software

発売日

2000年7月14日

スプレッドシートベースのGUI構築ツール
WINGZ v2.5J for LinuxURL <http://www.ifour.co.jp/>

アイフォーは、スプレッドシートベースのGUI構築ツール「WINGZ v2.5J for Linux」を7月14日より発売する。WINGZは、Windows、Macintosh、UNIXといったマルチプラットフォームで動作し、今回Linuxに対応する。価格は1万2000円。サポート・ライセンスを分離することで低価格化を実現した。初年度1万本の出荷を予定している。

WINGZは、強力な機能をもったHyperScriptによって、GUIベースのアプリケーションを簡単に作成することが可能。ビジネス・算術・統計・データベース・文字列・制御などの充実した関数、ダイアログボックスをGUIベースで設計することで、

対応するスクリプトを自動生成するスクリプティング・ツール 32768行×32768列の広大なワークシート データベース機能や行列・シミュレーション・解析計算機能 入力・検索・使用ファイルの履歴機能 マウスによる立体グラフの視点変更などが可能といった特徴がある。

なお、Oracle / Informixデータベースにネイティブに接続し、データの問い合わせ・更新・削除、データベース・表・列の制御などをWINGZから直接行うことができ、リレーショナル・データベースのフロントエンドとして利用できるWINGZ-DataLink v2.0Jをバンドルしている。



Hardware

発売日

2000年6月16日

超低価格Alpha 21264A搭載エントリーモデル
VT-Alpha6 Sシリーズ

URL <http://www.v-t.co.jp/>

ビジュアルテクノロジーは、最新の64ビットCPU Alpha 21264Aを搭載したエントリーモデル「VT-Alpha6 Sシリーズ」を6月16日より発売した。

VT-Alpha6 Sシリーズは、CPUにAlpha 21264A 600MHz（キャッシュ2Mバイト）、64MバイトECC SDRAM、20GバイトIDEハードディスク、3.5インチフロッピー、48倍速CD-ROM、グラフィックスカード（8Mバイト）、10/100BASEネットワークカード、日本語109キーボード、3ボタンPS/2マウスを搭載して29万7500円である。OSに

発売	ビジュアルテクノロジー株式会社
TEL	03-3434-0367
価格	29万7500円～

Red Hat Linux 6.2 Deluxe Edition for Alpha（Kondara MNU/Linuxも対応予定）をプリインストールした場合には、作業費およびOSメディア代として3万4000円の費用が別途かかる。

なお、CPUは700MHzクロックも選択可能となっている。

開発環境として、Linux OS専用にチューニングされたCompaq Fortran / C for Linux Alphaが用意されている。



Hardware

発売日

2000年7月1日

Linux動作確認済み、低価格56K Fax/Modem
REX-PCI56

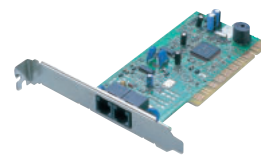
URL <http://www.rexpccard.co.jp/>

ラトックシステムは、最大56Kbpsの高速ダウンロードを実現したPCIバス用FAXモデムを発売した。国際標準規格ITU-T V.90およびK56flexに対応し、プロバイダの通信方式に合わせて自動的に認識する。価格は3800円。5年間の長期保証がつい

発売	ラトックシステム株式会社
TEL	06-6633-6766
価格	3800円

ている。

対応OSは、Windowsだが、同社でLinuxでの動作確認を行っている。動作確認されているディストリビューションは、TurboLinux、Red Hat Linux、Kondara MNU/Linux。



Software

発売日

2000年6月5日

統合型データ解析環境
S version 4.4

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

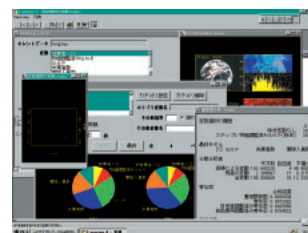
株式会社アイザックは、統合型データ解析環境「S version 4.4」を6月5日より発売した。発見的手法に基づくデータ分析やグラフ作成、Webアプリケーションの開発を支援するソフトウェアで、version 4.4から64ビットプロセッサに対応し、巨大なデータセットの処理が可能になった。

Windows 98 / 2000、Linux、各種のUNIXシステムに対応していて、UNIXやWindows上でのGUI機

発売	株式会社アイザック
TEL	03-3406-1645
価格	18万円～

能、Oracle8iへの対応といった多くの機能強化が行われている。

製品のライセンス価格は、対象プラットフォームにより異なっており、PC-UNIX版（Linux / FreeBSD / Solaris for x86対応）が、大学向け9万円、商用向け18万円、年間サポート料金が4万8000円となっている。なお、非商用の個人ユーザーは大学向けと同価格。



Software

発売日

2000年5月23日

多変量解析も可能な統計ソフト
STAT for Linux/BSD

URL <http://www.personal-media.co.jp/>

パーソナルメディアは、統計パッケージソフト「STAT for Linux/BSD」を5月23日より発売した。従来UNIX-EWS（エンジニアリングワークステーション）用として販売してきたSTATを、Linux / FreeBSD用に移植を行ったもので、基本統計から多変量解析までを網羅した16種類の統計機能を、対話形式で処理できる統計解析ソフトウェア。

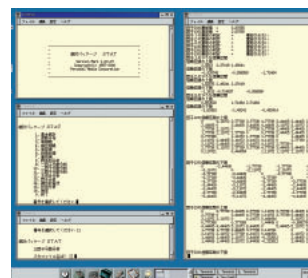
STATがサポートする機能は、基本統計、度数分布、単相関、順位相関、単回帰、段階回帰、重回帰、時系列分析、一因子分散分析、二因子分散分析、三因子分散分析、主成分分析、因子分析、線

発売	パーソナルメディア株式会社
TEL	03-5702-7858
価格	9万8000円

形計画法、判別分析、検定の16種類。

各統計機能はシェルスクリプトから利用可能、取り扱える変数やサンプル値は実質的に無制限、英語表示機能、欠損値処理や出力ファイルの保存、グラフ / 表計算ソフトとのデータ連係が可能といった特徴がある。

Red Hat Linux 6.1、Vine Linux 2.0のほか主要なLinuxおよびFreeBSD 3.3で動作する。インストールに必要なディスク容量は30Mバイト、日本語マニュアル「コンピュータ統計処理入門」が付属し、価格は9万8000円となっている。





「VMware Express for Linux」をバンドルした「TurboLinux Workstation日本語版6.0」発売開始

2000年6月23日

ターボリナックス ジャパンは、2000年4月7日に発売した「TurboLinux Workstation日本語版6.0」に、新たに「VMware Express for Linux」と、ユーティリティソフト集である「ボーナスCD」をバンドルしたパッケージの発売を開始した。

価格は従来のままの1万2800円。サポート体系やそのほかのバンドルソフトウェアについては変更はない。

ターボリナックス ジャパン

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

Mozilla M16をリリース

2000年6月17日

Mozilla Organization (Mozilla.org) が開発するオープンソースのWebブラウザ最新版「Mozilla Milestone 16」(M16) がリリースされた。Mozilla.orgは、ある程度安定した開発中のMozillaを、「Milestone」としてほぼ1月おきにリリースしている。16回目となるM16では、見た目を定義する「スキン」の切り替えや入力のオートコンプリート(ただしインラインのみ)、IRC (Internet Relay Chat) クライアント「Chatzilla」などを新たにサポートした。ベータ版であるため動作



はや遅いが、かなり安定していて、日本語のページを含むほとんどのWebページを正しく表示することが可能だ。

Mozilla.orgは、米Netscape Communicationsが自社のブラウザのソースコードを公開し、誰にでも開発/利用を可能にしたことから、開発を積極的に進めるために設立された。Netscape Communicationsが4月6日にリリースした「Netscape 6 Preview Release 1」は、Mozilla M14の成果を元に作られている。

Mozilla M16はソースコードおよびバイナリ形式で公開されている。Windows、MacOS、Linux、OpenVMS、FreeBSD、SolarisバイナリはMozilla.orgのFTPサイトおよびそのミラーサイトからダウンロードできる。ファイルサイズは、ソースコードが23Mバイト、バイナリは各OSともほぼ7Mバイト。

Mozilla M16リリースノート

(<http://www.mozilla.org/projects/seamonyk/release-notes/m16.html>)

KDE 2.0へのベータ版「KDE 1.91」が公開

2000年6月16日

KDE Teamは6月14日、KDE 2.0リリースの前段階となるベータ版「KDE 1.91」(コードネーム「Kleopatra」)をリリースした。KDE 1.91は、ノルウェー Troll TechのGUIツールキット「Qt 2.1」をベースに、オフィススイート「KOffice」や管理ツール「kadmin」、ゲーム、ユーティリティなどを含んでいる。

KDE 2.0は次の機能をサポートし、KDE 1.xと比べて大きく進歩している。

- ・クライアント間通信を行なうDCOP (Desktop COmmunication Protocol)
- ・非ブロッキングなGUIを可能にするI/Oライブラリ「KIO」
- ・Windowsのように、アプリケーションを他のアプリケーションに埋め込むことを可能にする「KParts」
- ・メニューやツールバーなどのGUIを、XMLにより作成可能にする「XML GUI」
- ・HTML 4.0や JavaScript、CSS-2、SSL、Netscape Communicatorのプラ

ゲインをサポートするレンダリングエンジン「KHTML」

KDE 2.0の公開は9月に予定されている。

KDE 1.91のソースコードはftp.kde.orgからダウンロード可能だ。バイナリパッケージは、現在のところCaldera OpenLinux 2.4とSuSE 6.4に対応したRPMパッケージが公開されている。

KDEホームページ

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

米Intel、IA-64 LinuxのSDKをリリース

2000年6月14日

米Intelと米Hewlett-Packardは6月13日、IA-64用Linuxアプリケーション開発キットをリリースした。キットにはIA-64 Linuxをシミュレートする「IA-64 Linux Simulator」のほか、カーネルやドキュメントなどが含まれる。IA-64はIntelの次世代CPUアーキテクチャの名称で、第1世代の製品として今年後半に「Itanium」の出荷が予定されている。

IA-64 Linux Simulatorは IA-32(Pentium)やCeleronのような現行のx86アーキテクチャ)でIA-64の命令セットをシミュレートするソフトウェアで、Linuxカーネル2.2で動作する。命令の並列実行やマルチプロセッシングをサポートしないという制限はあるものの、実際にIA-64のコードを実行することが可能だ。

今回のキットのリリースは、IA-64でLinuxをサポートしようという一連の動きの最新のものの。以前から、Itanium出荷と同時にLinuxをサポートしようとさまざまな企業が準備を進めている。たとえばLinuxカーネルの移植は、IntelやHewlett-Packardのほか、多くのLinuxディストリビュータが参加する「Trillian Project」が進めており、すでにベータ版をリリースしている。Red HatはIA-64 Linuxディストリビューションのアルファ版をリリースしている。これらの企業の努力によって、IA-64 Linuxが早期に移動し始めるのは確実といえるだろう。

開発キットのダウンロード

(<http://developer.intel.com/design/ia-64/linux.htm>)

米IBM、Linuxサポートを大幅に拡大

2000年6月13日

米IBMは6月12日、ハイエンドシステムからノートPCまでのLinuxサポートをさらに強化する発表を行なった。

発表内容は、ThinkPadへのLinuxプリインストールのほか、「RS/6000」対応Linuxディストリビューションの出荷、WebSphereのLinux対応、「DB2」と「WebSphere」をパッケージ化した「Small Business Pack for Linux」を特別価格で提供することなどである。

まず、RS/6000に対応したディストリビューションだが、今月中にドイツのSuSEから出荷が予定されている。価格は60日間インストールサポート付きで49.95USドル。また、IBMは2001年後半に登場する同社の次期「POWER4」プロセッサでもLinuxをサポートしていくことを明言した。

WebSphereのLinux対応版については7月出荷を予定しており、価格はStandard Editionが795USドル、Advanced Editionが7500USドル。対応ディストリビューションはStandard EditionがCaldera OpenLinuxとRed Hat Linux、Advanced EditionはRed Hat Linuxに対応する。

さらに、期間限定のプロモーション製品として、「Small Business Pack for Linux」が用意される。これは、IBMのDB2とLotus Domino、WebSphereをパッケージにしたもので、価格は499USドル。プロモーション期間は6月13日から12月13日まで。同製品のユーザー数は、サーバ1台あたり100ユーザーまでに制限されている。

IBMはすでに、PCサーバ「Netfinity」でLinuxをサポートしているほか、メインフレーム「S/390」へのLinuxの移植を支援している。

米IBM

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

Linuxカーネル2.2.16がリリース

2000年6月9日

Linuxカーネル2.2系列を管理するAlan

Cox氏は、Linuxカーネル2.2.16をリリースした。主な変更点はセキュリティホール修正で、アップグレードが「強く推奨」されている。

問題のセキュリティホールは、カーナビリティを扱うコードのバグにより、任意のコードを実行できるユーザーがsetcap(2)システムコールを通してroot権限を奪取できるという重大なもの(SecurityFocusのバグ報告)。それ以外にも、ELFローダやreadv/writev関数、シグナルスタックに存在するバグなどが2.2.16で修正されている。

サポートするアーキテクチャはx86とAlpha、PowerPC、SPARC、S/390の各プラットフォーム。MIPSやARM、M680x0への移植版はメインのカーネルツリーに統合中のため、それぞれのプラットフォーム特有のカーネルツリーをダウンロードするべきとのこと。

Linuxカーネル2.2.16のダウンロード

(<ftp://ftp.jp.kernel.org/pub/linux/kernel/v2.2/linux-2.2.16.tar.gz>)

米Sun Microsystems、Java開発ツール「Java 2 SDK, Standard Edition v1.3.0 for Linux」ベータ版を公開

2000年6月6日

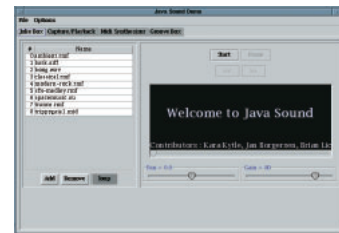
米Sun Microsystemsは、パフォーマンス改善のための「Java HotSpot」テクノロジーなどを搭載した、Java開発ツール「Java 2 SDK, Standard Edition v1.3.0 for Linux」のベータ版を公開した。

「Java Sound」のデモ、オーディオやMIDIファイルなどを再生する「Juke Box」や、ソフトウェアシンセサイザなどが含まれる。

このベータ版は、JDC(Java Developer Connection)の会員を対象に、JDCのページ(<http://www.sun.com/>)から無償でダウンロードできる。JDCは、名前やメールアドレスなどの登録だけで誰でも会員になれる。会費は無料。ファイル形式はRPM形式とtar.gzの形式のもので用意されている。Linux版のファイルサイズは約40Mバイト。

米Sun Microsystems

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



米Linuxcare、Linuxを「IA-64」に移植する「Trillian Project」へ参加

2000年6月1日

Linuxやオープンソースソフトウェアへの総合的なサポートで知られる米Linuxcareが、Linuxを「IA-64」に移植するプロジェクト「Trillian Project」へ参加することを発表した。

同プロジェクトへの参加は、IA-64上で動作するLinuxをサポートすることにより、企業でのLinuxの採用を促進するという目的があるからとのこと。

さらに、同社のエンジニアがプロジェクトに参加することで、LinuxのIA-64上への移植が加速され、また、IA-64上で動作するLinuxが現実使用されるようになったときに、より良いサポートをすることができるのもメリットだという。

米Linuxcare

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

IA-64 Linux Project

(<http://www.linuxia64.org/>)

レッドハット、日本シグナスソリューションズとの合併を完了

2000年6月1日

レッドハットは、6月1日、日本シグナスソリューションズとの合併を完了した。これは、米国で1999年11月に米Red Hatが米Cygnus Solutionsの買収を発表、2000年の1月に合併が完了したのを受けてのもの。日本国内でも同様の体制が整ったことになる。

今回の合併で、日本シグナスソリューションズは、レッドハットの社屋内に移転、6月1日より、正式にレッドハットとしての業務を開始した。日本シグナスソリューションズの提供していた製品、サービスは、引き続きレッドハットから提供される。また、日本シグナスソリューションズのWebサイトのURL

(<http://www.cygnus.co.jp/>) は、当面の間は引き続き利用できるという。

両社の合併が完了したことにより、組み込み機器から、デスクトップ、サーバソリューションまでを、オープンソースでサポートする企業が国内に誕生したことになる。

また、今後の展開についてレッドハットに質問したところ、現在のところ製品やサービスについて具体的な計画はなく、これから検討していくとのこと。

レッドハット

(<http://www.redhat.com/jp/>)

Windows エミュレータ「Wine」のロードマップ発表

2000年5月31日

「Wine project」は、Windows エミュレータ「Wine 1.0」リリースに向けてのロードマップを発表した。これは、「Wine」の開発情報を提供するWebサイト「Wine Development HQ」の「Wine Weekly News #45」上で発表されたもの。

Windows エミュレータ「Wine」は、カナダCorelの「WordPerfect」や、米IBMの「IBM NetObjects TopPage for Linux (日本では『ホームページ・ビルダー』)」などでも使用されており、オフィシャルなリリースである「1.0」の登場は、1993年に開発が開始されて以来、長らく待ち望まれている。

ロードマップの内容だが、具体的な日程などは含まれておらず、バージョン「1.0」のリリースへの条件が主な内容となっている。

ロードマップでは、ベータ版として公開されるバージョン「0.9」が、一般に公開する前に解決しなければならない問題として、以下を挙げている。

- ・ドキュメントの整備
- ・Wine APIの整理
- ・カナダのCorelによる「Wine」に対する改善のマージ
- ・Wine上のWindowsアプリケーションが、ほかのWindowsアプリケーションを起動した場合のプロセスの管理方法の改善

「Wine」のアプリケーションウィンドウの管理方法の改善

リリース予定は、以下のようになっている。

- ・上記の問題が解決され、「Wine」の一般公開に必要な機能がそろった時点でフィーチャーフリーズし、ベータバージョン「0.9」としてリリース
- ・十分にバグフィックスし、安定した時点で「1.0」としてリリース

また、「Wine 1.0」のリリースにより、同プロジェクトが終了するわけではなく、安定性の向上や、さらなる機能の追加を行っていくという。そして、新しいプロジェクトとして、「GNOME」、「KDE」などのデスクトップ環境との統合なども考えているという。

Wine Development HQ

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

「一太郎Ark for Java」ソースコード公開開始!

2000年5月31日

ジャストシステムは、5月31日正午、「一太郎Ark for Java (以下、一太郎Ark)」のソースコードを公開した。公開されたソースコードは、「一太郎Ark」本体のソースコード、APIドキュメント、標準プラグインの3つ。すべて、Javaのクラスライブラリのアーカイブ形式であるjarファイルとなっている。

ダウンロードは「一太郎Ark DevelopersSite」から可能。

今回のソースコードの公開は、「一太郎Ark」のプラグインの開発促進やパッチの開発、公開が主な目的となっている。また、気になるライセンスだが、オープンソースライセンスではなく、ジャストシステムが定めた独自のもので、名称は「一太郎Ark デベロッパーライセンス」となっている。

「一太郎Ark デベロッパーライセンス」の主な内容は以下のとおりだが、詳しくは「一太郎Ark DevelopersSite」にて全文を確認されたい。

- ・開発者が新たに開発したプラグインは自由に公開、配布可能
- ・オリジナルのソースコードからのバイナリコードの作成/使用は、パッチおよび

プラグインの動作試験あるいは評価目的でのみ可能。

- ・それ以外は「一太郎Ark」の正規ライセンスを購入しなければならない
- ・オリジナルのソースコードの再配布はどのような形であろうとできない
- ・オリジナルのソースコードに対する修正はパッチ(差分)でのみ公開、配布可能。また、ジャストシステムは、そのパッチを自由に利用できる

また、同社は、ソースコードの公開とともに「一太郎Ark」開発者向けのメーリングリスト「Ark Developers」の運営を開始した。参加方法など詳しくは「一太郎Ark DevelopersSite」を参照のこと。

一太郎Ark DevelopersSite

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

「VMware Beta 2.0.1 for Linux」配布開始

2000年5月29日

米VMwareは、「VMware Beta 2.0.1 for Linux」の配布を開始した。バグフィックスが主な内容となっている。ベータ版とはいえ、2.0以降アップデート版が配布されるのは今回が初めて。

VMwareは同社の主力製品で、Windows NT/2000、Linux上で動作し、それらのOS(ホストOS)上に仮想マシンを作りあげることができるソフトウェア。ユーザーはその仮想マシン上で別のOS(ゲストOS)を動作させることができ、それにより1台のマシン上で複数のOSを使うことが可能となる。

前バージョンからの主な変更点は以下のとおり。

- ・ホストOSとして「Red Hat Linux 6.2」、「SuSE Linux 6.4」、「OpenLinux 2.4」をサポート
 - ・Raw Disk関連のバグフィックス
 - ・ゲストOSのWindows 95/98が起動しないことがあるバグをフィックス
- ダウンロードは同社のWebサイトから可能。ファイルサイズは約6Mバイト。

ダウンロードページ

(<http://www.vmware.com/download/downloadlinux.html>)

Distribution

新着ディストリビューション

Kondara MNU/Linux Server 1.1

常に最先端をいくKondara MNU/Linux。そのサーバ版のリリースだ。簡単かつ確実にパッケージやLinuxシステムの管理を行うmphツール群、ジャーナリング機能を備えた注目のファイルシステムReiserFSのサポートなど、Kondaraならではの最新技術へのアプローチを探ってみる。

LASER5 Linux 6.2

完全なるWindowsリプレース。新しいLASER5 Linuxは、そう感じさせてくれるほど完成度の高い日本語ディストリビューションだ。日本語変換プログラムのATOKやGUIメーラのSylphed、さらには評価版ながらMS Office互換のThinkFree Officeなどの収録により、デスクトップ環境としては最高クラスの出来栄だ。

Linux Mandrake 7.1

目指すは最速ディストリビューション。PGCCによりビルドされたパッケージ群、インストール時に選べるReiserFSやXFree86 4.0などの収録は、神速を誇るKondara MNU/Linuxを凌駕するかのような内容だ。MandrakeとKondara、最速の栄冠を手にするのは果たしてどちらだろうか。

プロサーバ Linux ver3.0

とにかく手間いらず。[Enter]キーをポンポンと叩くだけで終了するインストール、初心者ユーザーも迷わず操作できるGUIのサーバ管理ツールにより、簡単にネットワークサーバが構築できる。Debianのパッケージ管理方式の採用も合わせて、完全なるメンテナンスフリーを目指す。

Kondara MNU/Linux Server 1.1

Kondara MNU/Linux Server 1.1 (以下Kondaraサーバ)はKondara Projectが開発するサーバ用のディストリビューションで、独自管理ツール群mph、ジャーナリング機能を備えたファイルシステムReiserFS、次世代ネットワークプロトコルIPv6などをサポートしているのが特徴だ。Intel x86アーキテクチャとAlpha上で動作し、基本システムは、カーネルにバージョン2.2.15、ライブラリにglibc2.1.3、XにXFree86 3.3.6をそれぞれ採用している。

パッケージの販売はデジタルファクトリが行い、インストールやXの初期設定などが90日間5件まで無償でサポートされるほか、サーバ用ディストリビューションらしく、詳細なネットワーク設定や、サーバの構築などの有償サポートも行っている。

統合管理ツールmph

KondaraサーバはLinuxをGUIで管理できるmphというツール群(画面1、2)が大きな特徴だ。Kondaraサーバに含まれるmphは表1のとおりだが、この中から代表的なものを見てみよう。

mph-admin

表1にあげたmph管理ツール群を統合するためのツールで、ユーザーはこのmph-adminを起動することで、mph-kernel(画面1)やmph-sendmail(画面2)といった個別の管理ツールを利用できる。

mph-kernel

カーネル再構築はソースが公開されているLinuxならではの作業で、必要な機能だけを取り込んだ特製カーネルが作成できるというメリットがある。その反面、初心者ユーザーには敷居が高い作業であることもまた事実だ。そこで、mph-kernelを使えば、主な機能を選択することで、RPMパッケージ化された新しいカーネルが作成できる。ユーザーは古いカーネルと新しいカーネルを簡単に入れ替えられるというわけだ。

mph-sendmail

sendmailはUNIX系OSで定番のメール配送アプリケーションだが、sendmailがもつ高い柔軟性を享受するには、難解な設定ファイルとして知られる

sendmail.cfの編集が必要だ。このmph-sendmailはsendmailすべての設定をするわけではなく、最も基本的な、受信するドメインと、転送を許可するドメインの2つを設定するものだ。

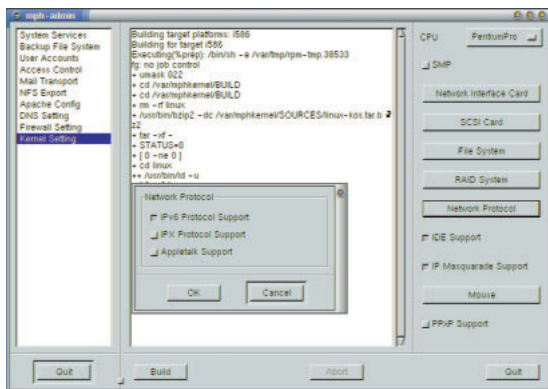
mph-get

ほかのmphツール群がGUIで操作するのに対し、mph-getだけはコンソール上で操作するアプリケーションだ。このmph-getは、事前に設定しておいたFTPサイトから最新のパッケージをダウンロードして、インストールまで行う優れものだ。Kondaraユーザーはmph-getを使うことにより、Kondaraサーバを最新状態に保つことができる。

これまでのLinuxは、設定ファイルの散在によるシステム管理の難しさが指摘されていた。しかし、これらmph管理ツール群を使えば、初級Linux管理者の悩みはあっさりと解決されることだろう。

インストール時に選べるReiserFS

Kondaraサーバは、インストール時

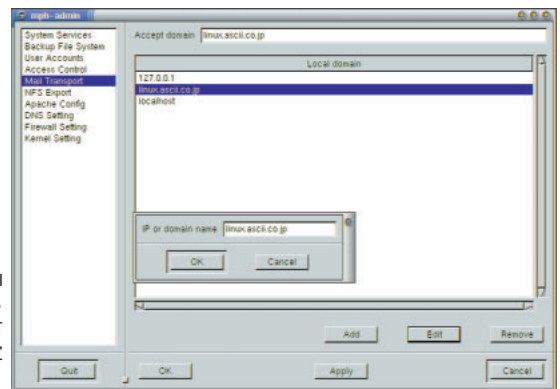


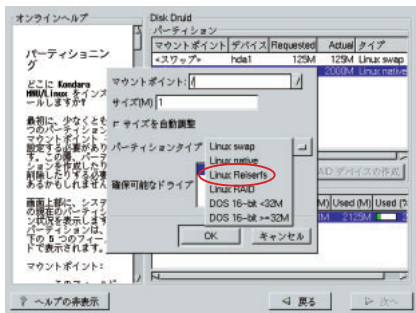
画面1 mph-kernel

これまでLinux初心者にはやや敷居の高かったカーネルの再構築作業が、このツールにより簡単に行える。

画面2 mph-sendmail

メールサーバソフトsendmailの基本的な設定、「受信するドメイン」と「転送を許可するドメイン」の2つを設定する。





画面3 ReiserFSが選択できるインストーラ

Red Hat系ディストリビューションのユーザーにはおなじみのインストーラ画面だが、プルダウンメニューには「Linux Reiserfs」という見慣れない項目がある。これを選択すると、ジャーナリング機能を備えたファイルシステムReiserFSが使用可能になる。

にジャーナリング機能を備えたReiserFSを選択できる（画面3）。Linuxが不慮の事故でシャットダウンした場合、Linuxはその直後の起動時にファイルシステムの整合性をチェックする。その際、現在Linux標準のext2だと、大容量ディスクの整合性をチェックするのに長い時間を要してしまう。ところが、ジャーナリング機能をもつReiserFSは、データの変更内容を逐一記録しているため、不正なシャットダウンが起こったときでも、その記録をもとにファイルシステムの整合性をチェックするので、起動時間を大幅に短縮できるというわけだ。大容量ディスクへのext2ファイルシステムの使用に限界を感じているユーザーにとって、ReiserFSの採用は非常にうれしい新機能だろう。



次世代ネットワーク環境

Kondaraサーバは、さまざまな次世代ネットワーク技術をサポートしている。

IPv6 (Internet Protocol version 6) インターネット利用者が急増している現在、IPアドレスは「水より貴重な資源」と言われるくらい不足している状態である。現在インターネットで標

パッケージ名	機能
mph-get	最新パッケージのダウンロードからインストールまでを行う
mph-admin	mph管理ツール群を統合する
mph-apache	WebサーバApacheを設定する
mph-backup	ちょっとしたバックアップに使用する
mph-dns	ネームサーバを設定する
mph-host	インターネットサービスのアクセス制御を設定する
mph-ipfwadm	ファイヤウォールを設定する
mph-kernel	カーネル再構築からRPMパッケージの作成までを行う
mph-nfs	ネットワークファイルシステムを設定する
mph-sendmail	基本的なメールサーバの設定を行う
mph-sysv	サービスのランレベルを設定する
mph-user	ユーザー管理を行う

表1 GUI管理ツール群mphの機能一覧

Kondaraサーバの大きな特徴である、GUI管理ツール群のmph。Linux管理に不慣れなユーザーは、これらのツールを使うことで、簡単にサーバの設定ができるだろう。mph-getのみコンソール上で操作する。

準のIPv4が32ビットアドレス空間を使うのに対し、IPv6では128ビットのアドレス空間が使えるようになる。これにより、莫大なIPアドレスが確保できるようになるため、次世代のプロトコルとして一部で導入が始まっている。

VPN (Virtual Private Network)

VPNは、暗号技術を使うことで、公共のインターネット上に仮想的な組織内LANを構築する技術だ。Kondaraサーバは、このVPNを使うためのCIPEというモジュールをサポートしている。

NetLOCK 3.0J

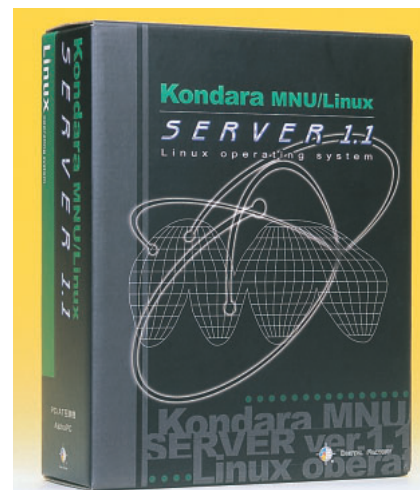
今回発売されたKondaraサーバには収録されていないものの、Kondaraを販売しているデジタルファクトリは、Netlock Technologies, Incが開発するNetLOCK 3.0Jを、今後はKondaraサーバにあらかじめ組み込んだ状態で販売することを決めている。このNetLOCK 3.0Jは、IP層を丸ごと暗号化するIPSecという技術に準拠したセキュリティ製品だ。ユーザーはKondara上でサーバ用のNetLOCK Managerを走らせ、WindowsやMac OSなど複数のOSに対応したクライアント用のNetLOCK Agentを使うことで、セキュリティの確保されたネットワー

環境をKondaraサーバのインストール直後から利用できるというわけだ。



大規模サーバの運用に最適

このように、KondaraサーバはReiserFS、IPv6、VPNといった先進機能をいち早く取り入れており、Raw I/Oや4Gバイトメモリのサポートとあわせて、大規模なデータベースシステムや、高度なセキュリティが求められるネットワーク環境構築に、特に威力を発揮するディストリビューションといえるだろう。



製品名	Kondara MNU/Linux Server 1.1
価格	2万4800円 1万9800円 (アカデミック版)
問い合わせ先	デジタルファクトリ株式会社 06-6882-5850 http://www.digitalfactory.co.jp/

LASER5 Linux 6.2

LASER5 Linux 6.2 (以下LASER5 Linux)は、レーザーファイブ株式会社がRed Hat Linux 6.2をベースにして開発するディストリビューションで、収録される商用ソフトが異なる2種類のパッケージが存在する。LASER5 Linuxは、多数の商用アプリケーション(表1)を収録して、同社より6月30日から発売されている。

Linuxの基本システムは、カーネルにバージョン2.2.14、ライブラリにglibc2.1.3、XにXFree86 3.3.6が採用されており、製品版ユーザーは、電話、FAX、Webページおよび電子メールにより、インストールからX Window System起動までについてのサポートを受けられる。

ここまで来たLinux デスクトップ環境

これまでのLinuxは、Windows環境と比べてオフィスツールの不在による、デスクトップ環境の弱さが指摘されていた。しかし、LASER5 Linuxは、今回新たにMicrosoft Office互換のオフィス用アプリケーションThinkFree Office(画面1)を収録して、日本語の文書作成環境を整えている。日本で

はオフィス用アプリケーションとして、Microsoft Officeが広く普及していることを考えると、ThinkFree Officeは、今後Linuxのデスクトップ市場でのシェア拡大を、大きくあと押しする存在になるだろう。

このほかにも、Windows環境で定評のあるATOK(Develに収録)や、UNIX環境でよく使われるWnn6を、日本語変換プログラムとしてバンドルしており、GTK+で開発されている、注目のGUIメーラSylpheedもあわせて使えば、Windows環境をリプレースできるほどのデスクトップ環境となるだろう。



LinuxでもJBuilder

JBuilderはWindows上で定評のあるJava統合開発環境で、LASER5 Linuxには同製品のエンタープライズ版となるJBuilder 3.5 Foundationがバンドルされている。

Linux上でJBuilderを使うには、JBuilder本体のほかにJDKが必要だ。しかし、JBuilder、JDKともにファイルサイズが非常に大きいため、ダイヤルアップ環境のユーザーがダウンロードするのは難しいだろう。そこで

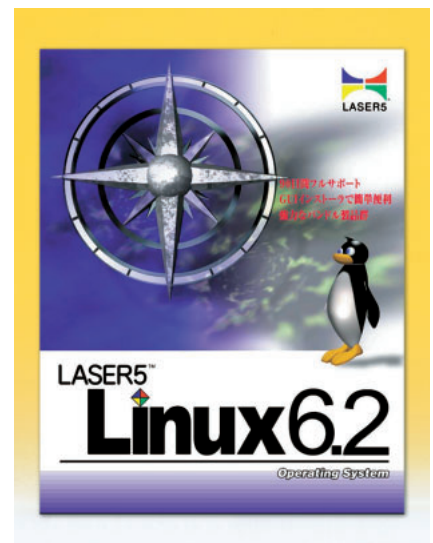


画面1 Microsoft Office互換のオフィスツール ThinkFree Office

Windows環境で作成したExcelファイルは問題なく開けた。画面でもわかるように、.docや.xls形式のファイルに対応している。

LASER5 Linux 6.2 Develは、JBuilderの使用に必要なファイルをすべて収録して、ユーザーがインプライズのサイトでライセンス(無料)を取得すれば、その日からJavaの開発が始められるように配慮しているというわけだ。

文書作成とJava統合開発環境、新しいLASER5 Linuxは、どこまでもWindows環境に迫る出来栄である。



製品名 LASER5 Linux 6.2
 価格 9800円
 1万4800円(Devel)
 問い合わせ先 レーザーファイブ株式会社
 03-5818-6626
 http://www.laser5.co.jp/

製品名	製品概要
Wnn6 version 3.01	日本語かな漢字変換システム
eWnn for LASER5 Linux	英和辞書引きソフト
翻訳魂 ver1.0	翻訳ソフト
DynaFont 和文 5書体	日本語商用フォント
Shade for Linux 体験版	3Dグラフィックツール
ThinkFree Office	Microsoft Office互換アプリケーション
以下はDevelのみ収録	
ATOK12 SE R.2 for Linux	日本語かな漢字変換システム
System Commander Lite	複数のOSを安全に利用
JBuilder 3.5 Foundation	Java統合開発環境
JDK 1.2.2	Java開発環境

表1 収録される主な商用アプリケーション
 LASER5 Linux 6.2の製品版は2種類あり、収録する商用アプリケーションが異なる。

Linux Mandrake 7.1

Linux Mandrake 7.1(以下Mandrake)は、フランスのMandrakeSoft社 (<http://www.linux-mandrake.com/>)が開発するディストリビューションで、XFree86 4.0、ジャーナリング機能を備えたReiserFSなど、最新機能をサポートしているのが特徴だ。基本システムは、カーネルにバージョン2.2.15、XにXFree86 3.3.6 (XFree86 4.0もインストール時に選択可)、ライブラリにglibc2.1.3を採用している。収録されるパッケージはPentiumプロセッサに最適化されており、意欲的に最新パッケージを収録するスタイルなど、日本で人気のディストリビューションKondara MNU/Linuxに通じるものがある。



XFree86 4.0が選べる インストーラ

ヨーロッパでBetter Red Hat Linux (改良されたRed Hat Linux)として評価の高いMandrakeは、フランスというお国柄を反映してかインストーラ (画面1) がとてもお洒落にデザインさ

れている。もちろん外見がお洒落だけでなく、インストール時に、ジャーナリング機能を備えた注目のファイルシステムReiserFSを選択できるほか、XFree86の最新バージョン4.0も選択できるなど機能面でも万全だ。このほかにも、ブートローダにGrubを採用したことで、ハードディスクの1024シリンダー外に配置してあるカーネルからの起動も可能にしている。



独自管理ツールDrakConf

Mandrakeは、GUIで操作できる独自のLinux管理ツールDrakConf (画面2) が特徴だ。各ディストリビューションによる独自管理ツールの開発は最近の流行りだが、このDrakConfは、ユーザー管理やネットワークなどの一般的な設定はもちろん、RPMパッケージの管理 (画面2) セキュリティレベル、ハードウェア、X Windows Systemなどを直感的に設定できるため、数ある独自ツールの中でも非常に完成度が高い部類といえるだろう。

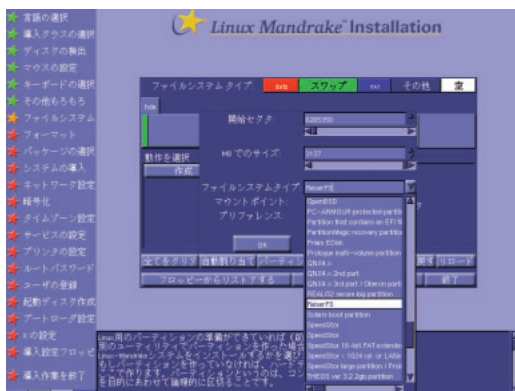


日本語が整備されれば最強か

そんなMandrakeも、日本語化に關しては他のディストリビューションに若干見劣りする。日本語対応のターミナルソフトrxvtや、メニューが日本語化されたKDEにより、ある程度は日本語に対応しているものの、インストール時に日本語キーボードが選択できなかったり、収録されているキーボードの定義ファイルにバグがあったりと、日本語化の優先度はそれほど高くないようだ。

しかしながら、容易なインストールやセキュリティレベルの設定など、初心者、ハイエンドユーザーのどちらにも対応できるレベルなので、これから本格的な日本語対応が進めば最強のディストリビューションとなるだろう。

本誌付録CD-ROMに収録されているのは、Linux Mandrake 7.1の<FTP版>です。付録CD-ROMによるインストール中、「Extension CD」や「Application CD」の有無をたずねる場面がありますが、これらは含まれていません。また、MandrakeSoft社からサポートを受けることもできません。



画面1 お洒落なインストーラ

フランス産のディストリビューションらしく、インストーラは非常に洒落たデザインだ。画面のように、注目のファイルシステムReiserFSがパーティション作成時に選択できる。



画面2 独自管理ツールDrakConf

Mandrakeの独自管理ツールDrakConfからRPMパッケージの管理ツールを起動したところ。ディストリビューションごとの独自管理ツールの開発は最近の流行りだ。

プロサーバ Linux ver3.0

プロサーバ Linux ver3.0 (以下プロサーバ) は、Debian GNU/Linux 2.2 (potato)をベースにするディストリビューションで、独自のGUI管理ツール(画面1)での簡単サーバ管理が特徴だ。カーネルはバージョン2.2.14、ライブラリはglibc2.1.3、XはXFree86 3.3.6を基本システムとして採用している。プロサーバはギデオンが開発し、富士マグネディスクが販売している。ユーザーは、インストールから管理メニューの起動まで、最大2件まで無償サポートを受けられる(3件以上、サポート範囲外の内容については有償)。

Enterキーを叩くだけのインストーラ

プロサーバのインストールはEnterキーを数回叩くだけだ。CD-ROMなどからインストーラを起動すると、パーティション作成から、適当な領域配分まで自動で行い、ユーザーは、インストーラに示されたパーティション情報を確認するだけだ。次の場面で起動用

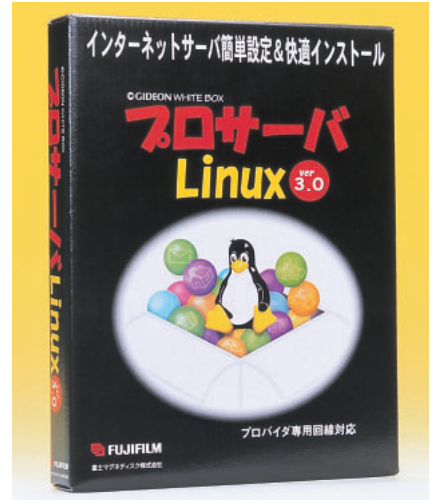
フロッピーディスクの作成を行うと、パッケージのインストールが始まり、Xも設定することなくLinuxの起動後から使用可能となる。パーティション作成や、インストールパッケージの選択が可能なカスタムインストールも用意されているが、標準タイプのインストールだとEnterを5回叩くだけで終了する。



GUIで簡単サーバ管理

プロサーバの最大の特徴は、独自に開発されたGUIサーバ管理ツール(画面1)だ。このツールはXと同時に起動し、ユーザーはこのツールで、sendmailを使ったメールサーバや、Windowsマシンとファイル共有するためのSambaなどを、グラフィカルな画面での直感的操作で設定できる。そして、設定内容をフロッピーディスクへ保存すれば、誤った設定変更をしても、設定変更前の状態へ簡単に戻すことができる。

また、プロサーバはDebianをベースにしているため、インターネットさえ



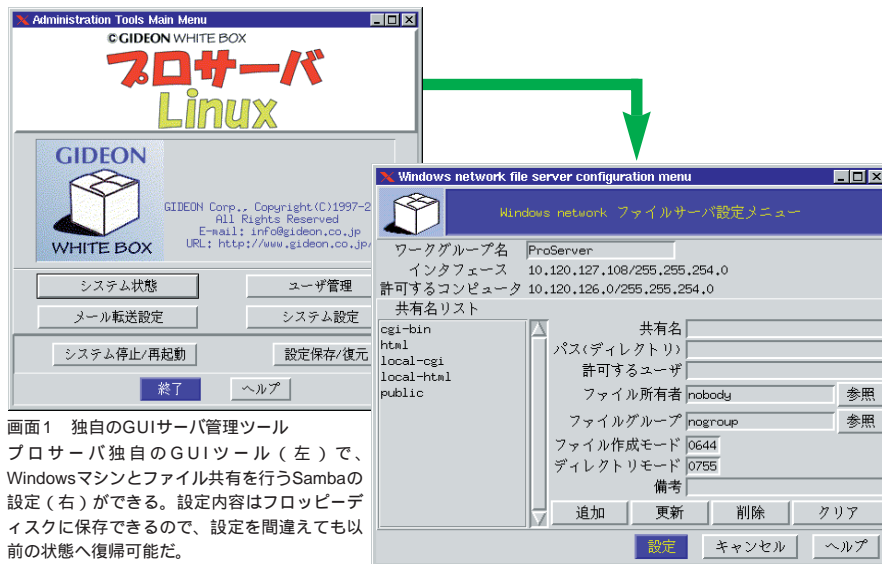
製品名 プロサーバ Linux ver3.0
 価格 1万9800円
 問い合わせ先 富士マグネディスク株式会社
 0424-81-8222
<http://www.fujifilm.co.jp/fmd/>

利用できれば、aptというパッケージ管理ツールを使い、Linuxを最新の状態に保つことができる。このaptはFTPやHTTPプロトコルを利用するDebianの管理ツールで、あらかじめFTPサイトやWebサイト(マシンがファイアウォール内部にある場合はプロキシサーバも)を設定しておけば、コマンド一発で、パッケージ間の依存関係をこわすことなく、最新のパッケージをインストールしてくれる優れたツールだ。



管理者不在のサイトにお勧め

このように、プロサーバはサーバの導入から運用まで、手間がかからないように細かい面まで配慮されているので、「Linuxに詳しい管理者はいないけど、Linuxを使ってファイルサーバやプリントサーバを簡単に構築したい」というユーザーにお勧めのディストリビューションだ。



画面1 独自のGUIサーバ管理ツール
 プロサーバ独自のGUIツール(左)で、Windowsマシンとファイル共有を行うSambaの設定(右)ができる。設定内容はフロッピーディスクに保存できるので、設定を間違えても以前の状態へ復帰可能だ。

Distribution ▶▶▶

▶ ターボリナックス、大規模並列計算用ソフト「enFuzion 6.0」をリリース

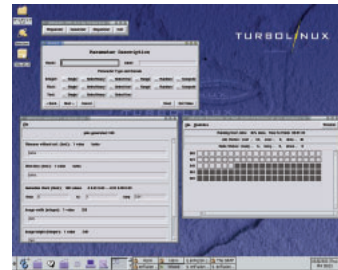
ターボリナックス ジャパンは、Linux、UNIX、および Windows NT上で動作する大規模クラスタリング用ソフト「enFuzion 6.0」を6月30日に発売した。価格は35万円（5ノード版）から。

これまでの大規模数値計算といえば、スーパーコンピュータが主なプラットフォームであったが、このenFuzion 6.0は、複数台のマシンに分散して処理を行うことにより、大規模計算のプラットフォームをコストのかかるスーパーコンピュータから、安価なPCやワークステーションへの置き換えを目指すソフトウェアである。

科学技術用途に限らず、超多変数データ解析を行う金融シミュレーションや、ヒトゲノム解析、店頭POSシステムから扱

い出した大量のデータ処理、顧客動向把握や新製品開発に有効なデータを引き出すデータマイニングといったさまざまな分野に応用できる。

enFuzion 6.0は、特別なプログラミングが不要で、既存のアプリケーションをそのまま性能向上されられるのが特徴だ。



ターボリナックス ジャパン株式会社

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

▶ Linux Mandrake 7.1/Alphaベータ版公開

仏MandrakeSoftは、ジャーナリング機能を備えたファイルシステムReiserFSや、メジャーバージョンアップされたXFree86 4.0などの先進機能をサポートしたディストリビューションLinux Mandrake 7.1のAlphaベータ版を公開した。

すでに正式版がリリースされているIntelアーキテクチャに加えて、Alphaプラットフォームをサポートした。Alphaマシンが活躍する科学技術分野では、商用UNIXの使用が一般的であったが、現在はソースコードが公開されているLinuxを採用するケー

スが増えている。

しかしながら、AlphaアーキテクチャをサポートするLinuxディストリビューションはそれほど多くないのが現状で、先進機能をサポートしつつ、簡単に導入できるLinux Mandrake 7.1のAlphaアーキテクチャサポートは、Alphaユーザーにとって非常にうれしいニュースとなるだろう。

なお、SPARC版もベータテスト中である。

MandrakeSoft社 (<http://www.linux-mandrake.com/>)

▶ レッドハット、HancornWord日本語版のRed Hat Linux独占バンドルを発表

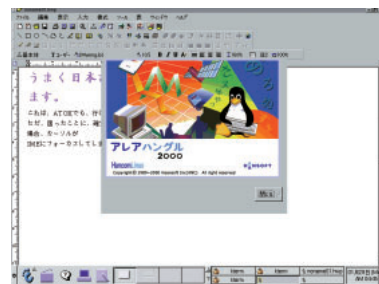
レッドハットは、韓国のハンコムリナックスと戦略的な提携契約を5月23日に締結し、今後順次出荷されるRed Hat Linuxに、HancornWord日本語版を独占的にバンドルして供給を開始する予定であると発表した。

HancornWordは、5月に行なわれたLinuxWorld Expo/Tokyo 2000で、アレアハングル2000として展示されていた製品で、中国語、ハングル、英語、日本語の4か国語に対応し、Microsoft Wordや、一太郎とのファイル互換性を持っている。

HancornWordはWindows環境エミュレータ「Wine」が使用されている。

すでにLinux用ワープロを中国と韓国でリリースしているハ

ンコムリナックスは、日本法人を設立し、本格的に日本に進出する。また、今後発売を予定しているワープロ以外のオフィス製品もレッドハットと共同でマーケティング活動を行っていく。



ハンコムリナックス (<http://www.hancom.com/jp/>)

▶ Slackware 7.1リリース

Slackware Linux Projectは、Slackware 7.1を6月24日にリリースした。1999年10月29日にリリースされた前バージョン7.0から約7か月ぶりのリリースである。

Slackware 7.1は、基本コンポーネントにカーネル2.2.16、glibc 2.1.3、PCMCIAカードサービス3.1.16、XFree86 3.3.6が採用されている。

今回よりパッケージが増えたため、ISOのCDイメージがinstallとcontribの2枚に増えた。contribには、XFree86 4.0、KDE 1.90/1.91、gcc 2.7.2.3/2.95.2などが収められている。また、ハードディスクのUltraDMA/66転送が有効となるパッチを当てたカーネルも用意されている。

Slackware Linux Project (<http://www.slackware.com/>)

Products

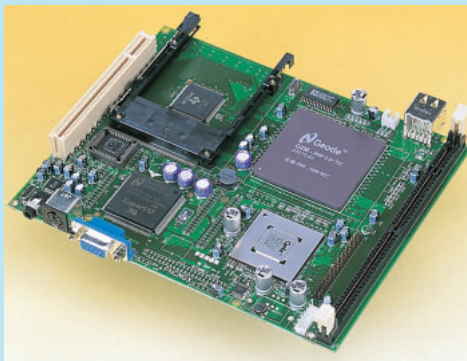
38

x86互換CPU搭載の小型マザーボード
PNCMGX ボードPC

40

ネットワークトラブルを解決するLAN監視/解析ソフトウェア
Packet Warehouse

x86互換CPU搭載の小型マザーボード



PNCMGX ボードPC

インターネット常時接続が身近になってきて、24時間電源を入れておくようなIPルータは低消費電力で音が静かなことが望まれている、この小型マザーボードを使ってLinuxを動かせば、自由にカスタマイズが可能なIPルータを実現できるだろう。

製品名	PNCMGX
価格	4万5000円(1~5台注文時)
問い合わせ先	株式会社ピノー TEL 03-5464-1851 http://www.pinon-pc.co.jp/

PNCMGXは、ピノーが設計/開発したボードPCで、145mm x 180mmとA5サイズより少し小さいマザーボード上に、CPUとグラフィックスやI/Oコントローラ、オーディオといったx86互換機の基本機能を搭載している。



小型マザーに
PCの機能を凝縮

CPUは、ナショナル セミコンダクター社のGeode GXMという、2Dグラフィックアクセラレータ、SDRAMインターフェイス、PCIバスコントローラ機能を統合したPentiumMMX互換

マイクロプロセッサが使われていて、クロックは266MHzで動作させている。基本的な機能は、サウスブリッジのCyril (現VIA Technologies) CX5530と、SuperI/Oと呼ばれるPC97317コントローラの3チップで構成している。

グラフィックスは、ビデオメモリにメインメモリを利用するUMA (Unified Memory Architecture) 機能が使われ、そのためメモリコントロール回路を内蔵しているCPUの内部の回路で、2Dハードウェアアクセラレーションを実現している。

メインメモリには、168ピンのDIMM

スロットが1つ用意されていて、通常のPC用と同じSDRAMが使用できる。フラッシュROMに、General Software社のEmbedded BIOSとROM-DOSが搭載されている。

そのほかのI/Oポートは、表1にあるようにPCに必要な機能をほぼ全部備えている。ただし、ボード面積が小さいためPC標準のコネクタが用意されているのは、VGA出力とUSBだけで、マウスとキーボードは、1つのPS/2ポートからスプリッターケーブルで2本に分けて接続する。ハードディスクやフロッピーなどは、基板上に付けられて

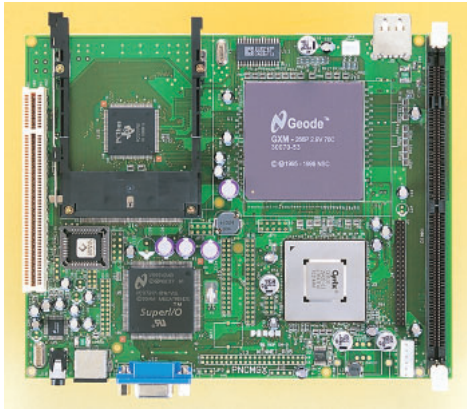


写真1 PNCMGXマザーボード
統合型x86互換CPUのGeodeを採用し、グラフィックス、サウンド、PCカード、USBなどをオンボードに搭載している。

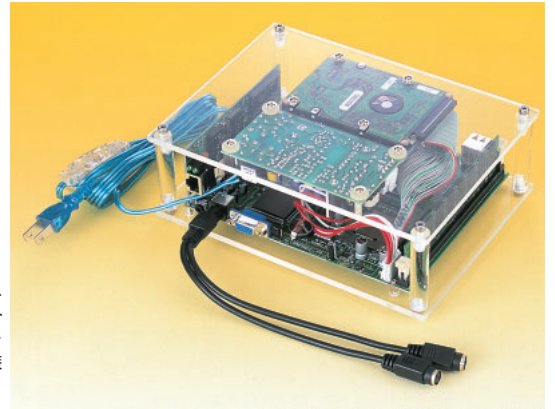


写真2 PNCMGX組み立て例
PNCMGXと2.5インチハードディスク、電源を、アクリル板を使って組み立てた。PCIバスにネットワークカード、メモリは128MバイトのDIMMを装着している。

いるピンコネクタに接続することになるので専用ケーブルが必要である。

接続ケーブルは、ハードディスク用2種、フロッピー用、PS/2用があり、4本セットで5600円となっている。



Linuxでの動作は?

このPNCMGXに、128MバイトのSDRAMとネットワークカードを装着し、2.5インチハードディスクと電源を、アクリル板で作った簡易ケースに組み込んだのが写真2である。これは試用機として製作されたもので販売されていない。実際にはケース、電源などを自分で用意する必要がある。

試用機にキーボード、マウス、モニタを接続し電源を入れると、まず画面1のように表示され、ちょっと驚いた。BIOSは同社でチューニングが可能なので、起動時のビットマップ画像は簡単に変更できるそうだ。専用機に組み



画面1 PNCMGXの電源ON直後に表示される画面

込んで使う場合など初期画面をカスタマイズできるのは便利だろう。

先ほどの画面はすぐに消え、通常のBIOS画面になり“LILO boot:”のメッセージが出て、ハードディスクにインストールされているRed Hat Linux 6.2Jが起動し、ログイン画面になる。当たり前だが、普通のPCを使っているのと何も変わりはない。

CPUの性能を見るために、ベンチマークプログラムnbenchを計ってみたところ、MEMORY INDEXが0.379、INTEGERが0.501、FLOATING-POINTが0.542という値であった。この数値はAMD K6/233MHzを1としたときの相対値で、半分ぐらいの性能だ。

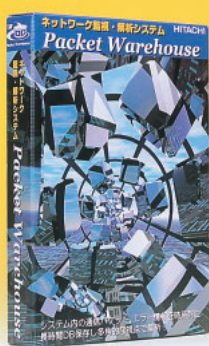
今後、CPUのスピードアップとローパワー化を計るために、Geode GX1への変更が予定されている。

PCIスロットとPCカードスロットが、それぞれひとつずつ用意されているのは嬉しい点だ。ネットワークカードやSCSIコントローラはPCIスロット用の安価なものを利用できるし、デジカメなどで使われるスマートメディアやコンパクトフラッシュ用インターフェイスカード、無線LANカードのようにPCカードで提供されているものも使えるのである。

その拡張性の高さを生かして、インターネット端末やコンパクトサーバとして利用できるだろう。

型番	PNCMGX
CPU	Geode GXM (PentiumMMX互換) 266MHz
ROM (Flash)	1Mバイト
RAM	168ピンDIMMスロット×1 (16 ~ 128Mバイト対応)
グラフィックス	CPU内蔵2Dアクセラレータ (VGA互換) 640 × 480 (1677万色) ~ 1280 × 1024 (256色) 1/1.5M/2M/4Mバイト (UMAによりメインメモリを使用)
CRTインターフェイス	標準VGAコネクタ
ハードディスクインターフェイス	IDEインターフェイス (UltraDMA/33) × 1ポート
フロッピーインターフェイス	1ポート
シリアルポート	RS-232C × 2ポート (ドライバなし)
パラレルポート	1ポート
USBポート	2ポート
オーディオ	AC97オーディオコーデック、スピーカーアウト
PCIスロット	1スロット
PCカードインターフェイス	1スロット (Type II)
キーボード / マウス	PS/2ポート × 2 (スプリッターケーブル使用)
基板サイズ	145mm × 180mm (突起物を除く)
消費電流	2.5A (DC 5V、Geode 2.9V版使用時)
消費電力	15W以下、平均12W (Geode 2.9V版使用時)

表1 PNCMGX ボードPCの主な仕様



Packet Warehouse

遅いとかつながらないといったLANのトラブルは、なかなか原因がわからないことも多い。Packet Warehouseは、LANに流れるパケットを常に監視し、ハードディスクに記録しておくので、収集したデータをじっくり解析することが可能だ。

製品名	Packet Warehouse
価格	35万円（パケットレコーダ） 14万8000円（解析View）
問い合わせ先	日立プロセスコンピュータエンジニアリング株式会社 TEL 0294-53-5111 http://www.hitachi.co.jp/hipro/

現在普及している10BASE-Tや100BASE-TXイーサネット（LAN）は、CSMA/CDという通信方式が使われている。複数のノードが接続されているLAN上では、複数の通信を同時に行うことができないため、まず、通信が行われているかどうかを確認し、空いていたら送信する。しかし、通信待ちのマシンが複数あった場合、同時に送信を始めてしまうと、データはケーブル上で衝突し、正しくデータを送れない。そこで、衝突を検出したマシンはランダムな時間待ってから再送することで、連続して衝突しないように工夫されている。

多くのマシンが接続されているLANでは、トラフィックが増えると衝突する可能性が高くなり、実際の

データ転送に使える時間が減るので、速度が低下したり、つながらないといった症状が現れる。また、マシンの故障や設定ミスによって、不要な通信パケットを送り続けていることもあるかもしれない。このようなトラブルは、LAN上に流れるデータを解析して、原因を追及する必要がある。



LAN上に流れるデータを モニタするパケットレコーダ

PCからサーバのファイルへアクセスするときや、Webブラウザを使ってWebサーバにアクセスするときなど、PCIはLAN上に相手サーバのアドレスを指定して信号を出すわけだが、同一のLAN上にあるすべてのマシンへ信号が流れるため、その信号をモニタリングして、パケット情報とし

て保存しておくのが、パケットレコーダの役割である。

Packet Warehouseは、日立プロセスコンピュータエンジニアリングが開発したLAN監視／解析ソフトウェアで、パケットを収集する「パケットレコーダ」と収集した情報を解析する「解析View」の2つのソフトウェアで構成されている。

パケットレコーダは、LinuxをOSに使った専用マシンとして動作する。PCのスペックは、Pentium 350MHz以上のCPU、メモリ64Mバイト以上が推奨されていて、システムディスク用に2Gバイト以上のIDEハードディスク、パケット収集用にアダプテックのSCSIカードAHA-2940UWと、それにつながるUltraSCSI（またはUltraWide SCSI）のハードディスク（最小2Gバイト）という構成だ。

データの解析は別のPCで行うので、それと接続するネットワークカード

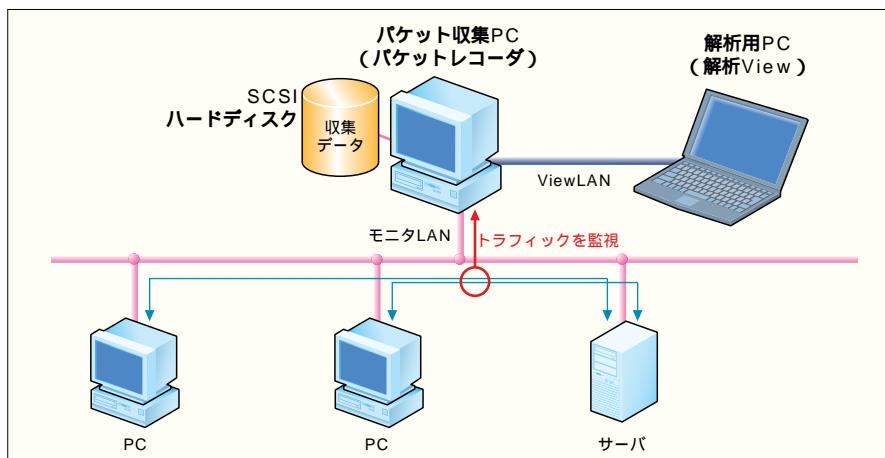
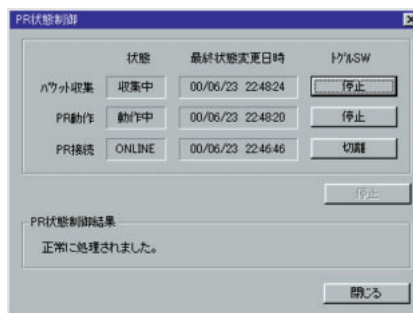


図1 Packet Warehouseシステム接続図



画面1 パケットレコーダ状態制御画面
ViewPCからパケットレコーダを管理することが可能。

のほかに、モニタLAN用のネットワークカードも必要である。モニタLAN用カードを追加することで、1台のパケットレコーダで2回線までのLANをモニタすることが可能だ。モニタLAN用ネットワークカードとしてIntel PRO/100+の動作が確認されていて、パケットデータを確実に取得するために推奨されている。

パケットレコーダのインストーラは、システム用IDEハードディスクのパーティションをすべて消去し、Red Hat Linux 6.0とパケットレコーダソフトを同時にインストールするようになっている。その後、ネットワークなどの設定ファイルを書き換え、パケット収集用SCSIハードディスクのパーティションを、fdiskコマンドで2Gバイト単位で区切っていくとセットアップは終了である。

解析View用PCで操作が行える

解析Viewは、Windows 95/98/NT4.0で動作するソフトウェアであり、ふだん使用しているPCにインストールしてもかまわない。原因を追及するために現場に近いところで使用するには、ノートPCなどを利用すると便利だろう。Windows上でsetup.exeを実行することでインスト

ールできる。

図1のように、パケットレコーダのモニタLAN側ネットワークカードにモニタしたいLANを接続し、View LAN側ネットワークカードにView用PCを接続する。パケットレコーダPC単体でデータを収集しておいて、あとでView用PCと接続して解析することも可能である。

さて、View用PC上でWindowsメニューからPwhViewを起動してみよう。まず行うのはパケットレコーダのIPアドレスを設定し、アタッチという作業で接続する。アタッチすることパケットレコーダの制御が行えるので、PR動作、パケット収集のボタンを押して開始する(画面1)。モニタLAN上に流れているデータが、パケットレコーダにどんどん取り込まれていく。

パケットレコーダで収集したデータは、プロトコル解析を行って表形式の画面で見ることができ、その1つを選んでデコード表示を行うと、さらに詳しく表示される(画面2)。ここでは、ポート番号80を使用してWebサーバと通信していることがわかる。

また、トラフィック統計解析を行うと、一定時間内に流れたパケットの量がグラフで表示され、LANの利

用状況の変化を一目で把握できる(画面3)。縦軸は、パケット数をバイト数や利用率に変更して見ることも可能だ。

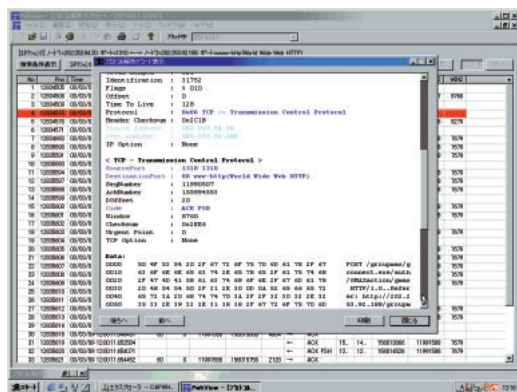
そのほか、モニタLAN上の変化がイベントログに記録され、未定義プロトコルやIPアドレス重複の検出、エラー発生、しきい値を超えた利用率の変化などがわかる。

ハードディスクへ保存し、じっくり解析

通常、こういった解析には、LANプロトコルアナライザが使われることが多いが、記憶容量が少なく全データを記録することはできない。そのため条件(フィルタ、TRAP)を設定して、その条件に合った前後のデータだけを見ることになる。

しかし、予想した条件以外の原因や、再現性がない突発的なエラーの場合、理由がわからないことが多い。そういう時には、LANに流れる全データを長時間にわたって監視/保存/解析することができるPacket Warehouseが役立つであろう。

LANトラブルの原因がわかれば、衝突を避けるためにLANを分割したり、スイッチングHUBを導入するなどの最適な対策が行える。すなわち無駄な出費も削れるわけだ。

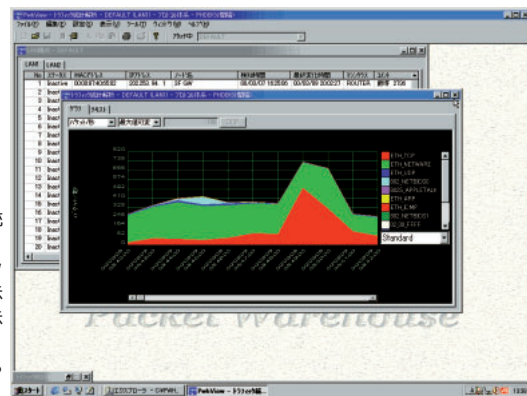


画面2 プロトコル解析デコード表示

取り込まれているデータを表の中から選択すると、パケットを解析してプロトコルやデータ内容を見やすく表示する。

画面3 トラフィック統計解析

単位時間あたりのトラフィック量をグラフで表示した。リアルタイム表示も可能で、パケット数、バイト数、利用率を見ることができる。



Linuxレスキュー

『疑問・難問』



『200問』

Linuxを使いたい！使ったけれどわからない！使っているうちにいろいろ困ったー！誰もが一度は悩む疑問・難問。そんな悩めるアナタを必ず救う処方箋を200問用意しました！これを読めば問題解決！目からウロコ間違いなし。明日からはぐっすり眠れます。

文：編集部 + 竹内充彦
Text: Linux magazine + Michihiko Takeuchi
illustration: Aki



基礎



Question 1



Linuxをインストールしたいのですが、どれくらいのディスクスペースを用意すればいいでしょうか

A. 選択するインストールオプションやディストリビューションによっても大きな違いがあります。最近のRed Hat系ディストリビューションの場合、フルセットでインストールすると1~1.4Gバイト程度を使用します。ユーザーが使用するデータファイル用のディスク領域と合わせて、とりあえずは、2~3Gバイトほど用意しておけば大丈夫です。

Question 2



PC-9801にLinuxをインストールしたいのですが、友人には「ムリだ」と言われました。本当にだめなんでしょうか？

A. こじまみつひろ氏を中心に日本で開発されているPlamo Linuxは、PC-98x1シリーズをサポートしています。Plamoでは、もともと京大マイコンクラブが開発したLinux/98というカーネルを採用しています。最新バージョンのPlamo 2.0には、Linux/98 2.2系の最新版が含まれていますので、AT互換機とほぼ同等の環境でLinuxが利用できるでしょう。

Question 3



テキストログインのときに、タイプミスしてBackSpaceキーを押すと“ ^H ”という変な文字が入力されます。キーで戻ろうとすると“ ^[[D ”となって困ってます

A. 初めて経験すると確かに少しあせってしまいますが、心配することはありません。ログインプロンプトの状態では、DeleteキーにBackspaceキーの機能が割り当てられた状態になっています。Deleteキーで戻って再入力すれば大丈夫です。

Question 4



PCの電源を切りたいんだけど、いきなりスイッチを押して大丈夫ですか？

A. いきなり電源を切ってしまうと、ディスクに保存されていないデータが消えてしまったり、二度と起動しなくなったりする可能性があります。電源を切る前に必ずシャットダウン操作が必要です。次のコマンドだけは覚えて

おきましょう。

```
# shutdown -h now
```

オプションの-hを-rにすると、システムの終了ではなく、再起動が行われます。

Question 5



システムが反応しません。リセットしていいですか？

A. キーボードがロックアップして使えない場合は、ネットワークログインを試してみましょう。それでもだめなときは、リセットボタンを押す、あるいは電源を切るしかありません。この場合、しばらく待ってハードディスクのアクセスランプが点灯していないことを確認してください。ディスクキャッシュに残っているデータも、一定の時間がたつとハードディスクに書き込まれるので、こうすることで、ディスクデータの破損を最小限に抑えることができます。

次回の起動時に、ファイルシステムに破損がないかチェックされます。チェックの結果、問題がなければ通常どおり起動します。

Question 6



あまりにもコマンドが多すぎて使い方がわかりません

A. Linux (UNIX) には、非常に多くのコマンドがあり、しかもそれぞれのコマンドにたくさんのオプションがあります。そのすべてを覚えている人はいません(と思われる)。実際には、頻繁に使用するコマンドはそれほど多くありません。lsやcpのようによく使うコマンドを解説した入門書が数多く出版されていますので、1冊購入することをお勧めします。

あまり使わないコマンドやオプション、初めて使うコマンドの場合は、調べながら使うということになります。Linuxには、manというオンラインマニュアルシステムが用意されています。“man <コマンド名>”と入力すれば、コマンドの使い方とオプションの説明が表示されます。多くのコマンドは英語のmanページしかなく、決してわかりやすくはありませんが、コマンドを使いこなす手がかりにはなります。

Question 7



rootで使い続けるのはよくないと聞きました。ほんとはですか？

A. rootユーザーはシステムに関するあらゆる権限を持っていますから、システムクラッシュ、ファイルの



破壊などが容易に行えます。たとえば、“rm -rf *”を実行すると、それが誤った操作でも、カレントディレクトリ以下にあるファイルとディレクトリをすべて削除できてしまうのです（試さないほうが身のためです）。これは、企業などの開かれた環境において、離席時のセキュリティが保てないといった問題にもつながります。

このような問題を避けるために、一般ユーザーを作っておき、ふだんはそのユーザーで作業するようにしましょう。rootユーザーの権限が必要なときだけ、suコマンドを利用して、一時的にrootユーザーとなり、作業が終わったらexitして、すぐに一般ユーザーに戻ることをお勧めします。



Question 8

一般ユーザーはどうやって作るの？

A. useraddコマンドを使ってコマンドラインから作成するのが基本となります。コマンド構文はやや複雑ですが、覚えておくと便利です。また、Linuxconfやkuserなどのツールを使って作成することも可能です（システム構成によっては使えないこともあります）。TurboLinuxのように独自のコマンド（turbousercfg）が用意されているディストリビューションもあります。



Question 9

WindowsではInternet Explorerを使っています。馴れているのでLinuxでも使いたいのですが、メニューにありません

A. Internet ExplorerはWindowsに標準で添付されていて、スタイルシートやXMLなどに対応した豊富な機能を持つ優れたWebブラウザですが、残念ながらLinux版は現在のところありません（Microsoftでは、SunのSolarisとHPのHP-UXに対応したバージョンはリリースしていません）。

このことからわかるように、Windowsのアプリケーションは、そのままではLinux上では利用できません。ただし、

方法がないわけではありません。米国のVMware社が開発した、その名もズバリVMware for Linuxを使えば、Linux上でWindowsアプリケーションを利用できます。というより、Windowsそのものが動作するのです。詳しくは同社のWebサイト（<http://www.vmware.com/>）をチェックしてみてください。



Question 10

しばらくほうっておくとコンソール画面がブランク画面になってしまいます。いちいち復帰するのは面倒なので、できればずっと表示したいのですが……

A. Linuxではコンソールのブランク画面への切り替えがデフォルトで設定されています。通常のコンソールであれば、

```
# setterm -blank 0
```

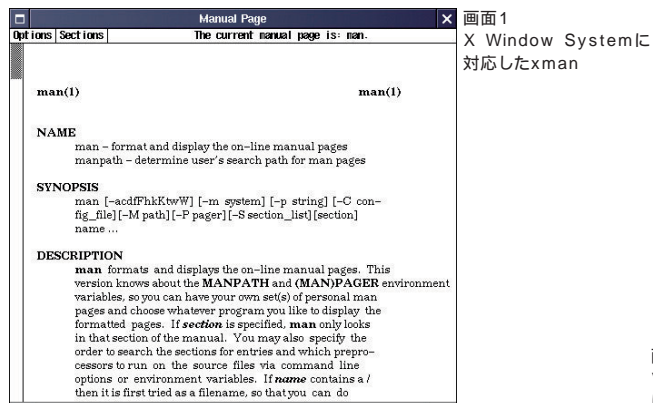
とすればこの設定はオフされます。そのままでは、システムをリブートすると設定がクリアされてしまいますので、/etc/rc.d/rc.localファイルに、このコマンドを追加しておきましょう。これで、起動時に設定が有効になります。

日本語表示に対応したkonを使っている場合は、/etc/kon.cfgを変更する必要があります。

```
SaveTime:
    5
```

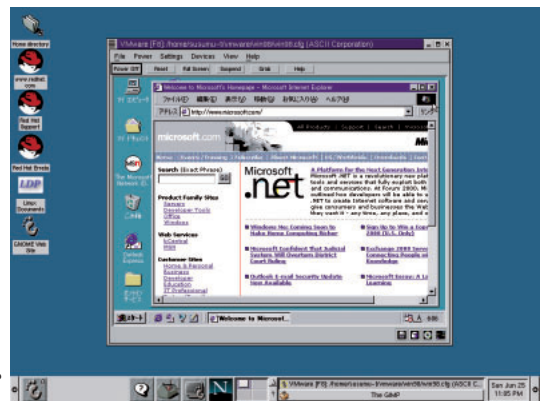
kon.cfgにある上記の行の“5”を“0”とすれば、ブランクスクリーンの設定がオフになります。

X Window Systemには、xlockやxscreensaverといったスクリーンセーブプログラムがあり、たいいていのディストリビューションではデフォルトでオンになっています。“xset s off”コマンドを実行すれば、スクリーンセーブ機能を無効にすることができます。



画面1
X Window Systemに
対応したxman

画面2
VMwareを使えばIEも
ほらこのとおり



ディストリビューション



Question 11



ディストリビューションとはなんですか？

A. 厳密な意味でのLinuxは「カーネル」と呼ばれるコア部分（システムの基本サービスを提供する）だけを指します。そのカーネルにライブラリ、基本コマンド、アプリケーションなど、さまざまなソフトウェアを組み合わせ、インストールするとLinuxシステムが動作するようにしたものを「ディストリビューション」といいます。

Question 12



ディストリビューションは何種類あるのでしょうか？

A. 日本語に対応しているディストリビューションだけで10種類以上あります。海外のものも含めると100種類は超えているでしょう。また、Red Hat Linuxのように、スタンダード/デラックス/プロフェッショナルというように複数のパッケージが販売されていたり、AlphaやPowerPCといったCPU用や、各国語版など、同じディストリビューターから複数の製品が販売されていたりします。

Question 13



それぞれのディストリビューションはどこが違うのですか？

A. インストーラやハードウェアの設定ツールなどを、ディストリビューターが独自に開発していたり、カスタマイズしていたりするので違いがわかるでしょう。標準としているウィンドウマネージャや日本語化への取り組み方にも、個性が現れるところです。

製品として提供されているものには、日本語TrueTypeフォントや商用かな漢字変換ソフト、ブートマネージャなどの商用ソフトウェアが含まれていることが多いようです。また、電話、FAX、メールなどで受けられるサポートの内容もディストリビューターによって異なります。サポート体制のしっかりしたRed Hat Linuxや、TurboLinux Server、Kondara MNU/Linux Serverなどのサーバ用途に特化した

ディストリビューションも提供されています。

ディストリビューションについての情報は、<http://www.linux.or.jp/distributions/index.html>（日本語）や<http://www.linux.org/dist/index.html>（英語）、各ディストリビューションのWebサイトに掲載されています。

Question 14



カーネルやglibcのバージョンの違いは気にしなくてもいいの？

A. ベースとなるLinuxカーネルのバージョンは、なるべく新しい（数字が大きい）ほうが、いろいろ改良されていてよいでしょう。カーネル2.0.xと2.2.xや、glibc 2.0と2.1といったメジャーバージョンが違う場合には、大きく機能が変更（向上）されています。

しかし、たとえばカーネル2.2.13と2.2.14と2.2.15のあいだにそんなに大きな違いはありません。ハードウェア用ドライバの改良やセキュリティホール改善といった、一般ユーザーにとっては微妙な点が違うだけなので、ディストリビューションを比較するときには気にしなくてもよいでしょう。

それよりも、ディストリビューターがWebページで公表しているセキュリティ情報などをチェックして、アップデートすることが肝心です。

Question 15



パッケージとはなんですか？

A. あるひとつのアプリケーションに関連するプログラムや設定ファイル、ドキュメントなどを*.rpmや*.debといったファイルにひとまとめにしたもので、パッケージマネージャを利用することでインストールやアンインストールが正しく行えるようになっていきます。ソースプログラムや、バイナリを配布するのに都合がよいため、Linuxで広く使われています。

Question 16



FTP版と製品版との違いはなんですか？

A. どのようなものを無料で配布するかという方針は、ディストリビューションによって異なりますが、大まかにいって製品版から商用アプリケーションやサポートなどを除いたものがFTP版となっている場合が多いようです。ただし、フリーでFTPサイトからダウンロードできるものであっても、機能が限定されているというわけではなく、Linuxシステムが持っている特徴などは製品版と変わりはありません。



Question 17

RPMとはなんですか？

A. RPM (Red Hat Package Manager) は、Red Hat Softwareによって開発されたパッケージ管理方式のことです。数多くのディストリビューションによって採用されており、インストール時に記録しておくデータベースを元に、パッケージ間の依存関係のチェックが行われるため、ソフトウェアが動作しない環境ではインストールできない(エラーになる)ようになっています。

ソフトウェアのインストール、アップグレード、アンインストールは、rpmコマンドによって容易に行うことができます。rpmコマンドを利用する際、最低限必要なオプションは次のとおりです。

```
# rpm -i (パッケージのインストール)
# rpm -U (パッケージのアップグレード)
# rpm -e (パッケージのアンインストール)
# rpm --help (rpmコマンドの簡単なヘルプ)
```

また、コマンドラインが苦手な初心者ユーザーは、rpmコマンドを利用しなくても、glintやGNOMEのGnoRPM、KDEのkpackageといったGUIツールを利用するのもよいでしょう。

Question 18

rpmファイルなら、RPM系のどのディストリビューションでもインストールできますか？

A. 他のディストリビューション用のバイナリRPMはインストールしないほうがよいでしょう。ソースレベルでは互換性を持っていても、ディレクトリ構成が異なっていたり、ライブラリのバージョンが異なっていることがあるからです。

ソースRPMを利用して、rpm --rebuildコマンドで、そのディストリビューション用のバイナリRPMを再構成してみると、うまくいくことがあります。

Question 19

PlamoやSlackwareでRPMパッケージを使うには？

A. バイナリで配布されているRPMパッケージを、Plamo LinuxやSlackwareで利用することはほとんど不可能です。これらのディストリビューションではライブラリのバージョンやインストールされている環境がRPMで

管理されていないため、依存関係のエラーが解決できません。

ソースのRPMパッケージからなら、tar + gz形式に変換し、コンパイルすることでインストールが可能です。具体的には、rpm2tgzまたは、rpm2targzというプログラムで、それぞれ.tgzまたは.tar.gzに変換し、tar xvzf ファイル名と行って展開したソースプログラムを、コンパイル(make)します。しかし、ほとんどのLinux用プログラムは、RPM以外に.tar.gzや.tgz形式でもソースコードが配布されているので、そちらを利用したほうがよいでしょう。

Question 20

debってなに？

A. debはDebian GNU/Linuxによって開発されたパッケージ管理方式です。強力なパッケージ管理能力が特徴で、Storm LinuxやCorel LINUXなどもdeb方式を採用しています。deb方式のパッケージは、dpkgコマンドによって利用することができますが、dpkgのインターフェイスとしてdselectやaptなどを使うと便利です。

また、依存関係のチェックなどを行わないため厳密にはパッケージ管理方式とは呼べませんが、SlackwareやPlamo Linuxでは、バイナリにインストールシェルスクリプトを加えてtar.gz形式でまとめたファイルを利用して、pkgtoolコマンドなどでインストール/アンインストールを行えるようになっています。

Question 21

RPMとdebでは互換性はあるのですか？

A. それぞれは別の管理方式ですので互換性はありません。ただし、alien(<http://kitenet.net/programs/alien/>)というコンバートプログラムによって、RPMとdeb(dpkg)とtgzのフォーマットを相互に変換することができます。また、KDEのパッケージ管理コマンド「kpackage」はRPMとdebの両方をサポートしています。

Question 22

日本語ディストリビューションは、どこが「日本語」なんですか？

A. 日本語化が必要になる要素として、マルチバイト対応のライブラリ、インストーラ、マルチバイト対応コマンド、アプリケーション、ドキュメント(manも含む)環境定義ファイルの用意などがあります。これらを総合的にまとめて日本語が使えるようになっています。

ハードウェア



Question 23



私の使っているPCでLinuxは動作するの？

A. Linuxのインストーラやカーネルがハードウェアを自動認識して、必要なドライバを組み込みますので、Windowsが動作するPCならほとんどの場合動作します。

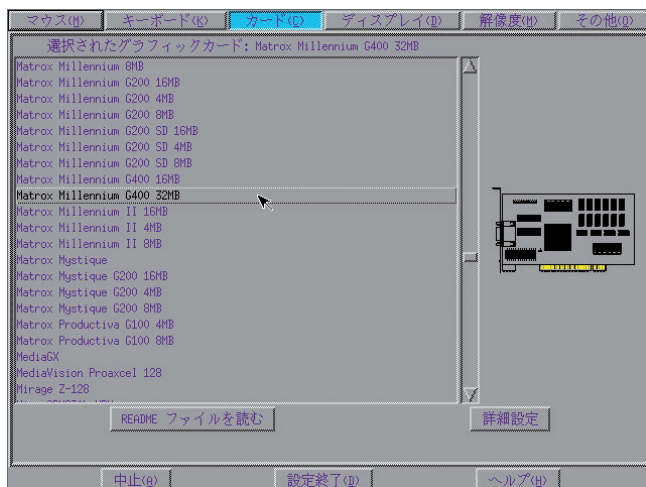
ただし、Linux用のドライバはすべてのハードウェアに対応しているわけではありません。また、ハードウェアメーカーがLinuxでの動作を検証していないため、インストール中にハングアップしてしまうこともあります。特にサウンドカードや内蔵モデムは、Linuxで使えない可能性が高いデバイスです。

Question 24



Linuxで動作するパーツなのかを調べるには？

A. パーツの場合には、まず製品と使用しているチップの型番を調べ、レッドハット (<http://www.redhat.com/jp/>)、ターボリナックス (<http://www.turbolinux.co.jp/>)、レーザーファイブ (<http://www.laser5.co.jp/>) といったディストリビューションのWebサイトで、動作確認情報を探すとよいでしょう。そのパーツメー



画面3 XFree86Setupのグラフィックスカード選択画面

カーのWebサイトにLinuxへの対応状況が書かれている場合もあります。

開発中のドライバやアプリケーションを使ってもよいのなら、LinuxLinks.com (<http://www.linuxlinks.com/>) から探してみましょう。そういうドライバは、ひょっとするとカーネルの再構築まで行う必要があるかもしれません。

そのほかには、gooやAltaVistaなどのロボット系Web検索サイトや、Linux関係のメーリングリストで問い合わせるということが考えられます。

Question 25



グラフィックスカードをインストーラが正しく認識しません

A. Red Hat、Vine、Kondara、LASER5のグラフィカルインストーラが正しく表示されない場合には、インストールの最初に「boot:」と表示されたところで“text”と入力し、テキスト画面でインストールしましょう。

Linuxではグラフィックス出力に、フリーのXサーバであるXFree86を採用しています。最新ディストリビューションのほとんどは、XFree86 3.3.6を採用していて、Voodoo3やMillennium G400、RIVA TNT2、i810、GeForce256などもサポートされています。

インストーラのX Window Systemの設定で、グラフィックスカードの自動認識に失敗して正しく表示できなかった場合は、Xの設定はキャンセルして、インストール後に手動で再度設定するとうまくいくことがあります。セットアップはXF86Setupコマンドで行います(画面3)。Red Hat Linux 6.2JなどではXF86Setupはインストールされていないので、CD-ROMからXFree86-VGA16-*、XFree86-XF86Setup-*というRPMファイルをインストールします。ディストリビューションによっては、Xconfiguratorコマンドやturboxcfgコマンドを使う場合もあります。

お使いのグラフィックスカードがXFree86に対応しているかどうかは、マニュアルを読むか、それぞれのディストリビューションやXFree86.org (<http://www.xfree86.org/>) のWebサイトで調べてください。

Question 26



内蔵モデムが使えません

A. 最近のノートPCに採用されている内蔵モデムは、DSPを使いソフトウェアによってモデム機能を実現しているいわゆる「Winmodem」というものです。Winmodemはソフトウェアでモデム機能が実現されていま



すので、基本的にWindows上でしか使用できません。

しかし、Linux上でこのWinmodemを使えるようにドライバを開発する「linmodem」というプロジェクト (<http://www.linmodems.org/>) が立ち上がっています。

Lucent社のチップを使ったWinmodemのLinux用ドライバに関して、バイナリパッケージがLucent社からリリースされているほか、PCCHIPS社から、PC-TEL (<http://www.pcchips.com.tw/driver/Linux/MODEM/>) というドライバが提供されています。「LT Modem」 (<http://www.close.u-net.com/lmodem.html>) というプロジェクトも活動しています。

Question 27



サウンドカードの設定はどうするのですか？

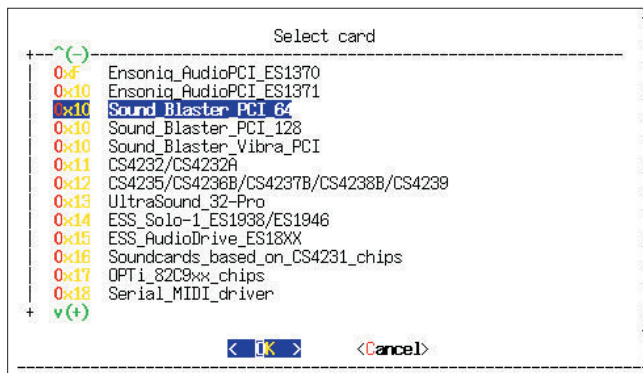
A. Linuxのカーネルには、フリーのサウンドドライバが含まれています。しかしサウンドカードによっては、インストーラで自動的に検出されない場合があります。そのときには手動で設定します。

Red Hat系のLinuxでは、sndconfigコマンドで設定します。root権限で、まず/usr/sbin/sndconfigを実行してください。またTurboLinuxでは、turbosoundcfgコマンドで設定します。

lsmodコマンドで、soundモジュールが組み込まれているか確認しておきましょう。

なお、設定された結果は、/etc/conf.modulesファイルに記録されます。ISAバスのサウンドカードなどで、I/OアドレスやIRQ、DMAポートなどが合っていないために動作しない場合には、conf.modulesファイルを正しい値にエディタで変更します。修正したら、

```
# /etc/rc.d/init.d/sound restart
```



画面4 alsacnfのサウンドカード選択画面

と入力し、サウンドドライバを再起動します。

CreativeのSound Blaster (Live!を除く) やESS系サウンドチップ、YAMAHAのFM音源チップOPL3などを使用しているのならば、カーネル標準のドライバで動作します。

最近では、ALSA (<http://www.alsa-project.org/>) のドライバ (alsasound) を組み込んでいるディストリビューションが多くなっています。alsaconfを実行することで、対話的にサウンドカードを設定することができます (画面4)。

Sound Blaster互換製品やYAMAHAのYMF724、YMF744などを使用している場合には、商用のOpen Sound System (<http://www.opensound.com/>) ドライバを組み込む必要があります。

Question 28



プリンタに出力したいのですが？

A. インストール時にプリンタの設定を行わなかった場合には、Red Hat LinuxやLASER5 Linuxでは、GNOMEのメニューからコントロールパネルを開き、Printtoolを使用して、プリンタの設定を行うことができます (画面5)。TurboLinuxでは、turboprintcfgを使用します。

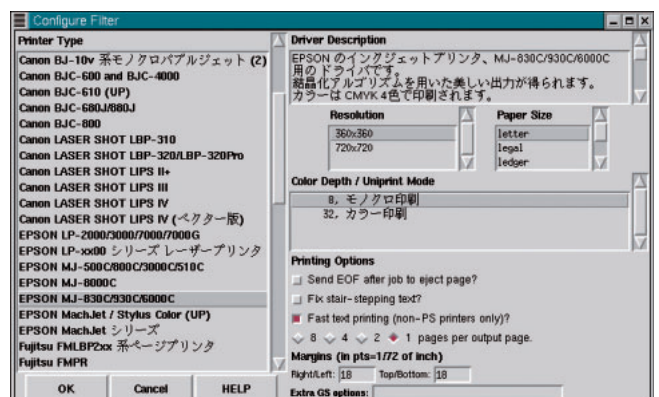
パラレルポートに接続したローカルプリンタや、LANで接続されたネットワークプリンタはもちろんのこと、Windowsマシンで共有プリンタに設定したプリンタにもSMBプロトコルでプリントアウトすることができます。

Question 29



UltraDMA/66対応のハードディスクは使えますか？

A. UltraDMA/66をサポートしているIDEコントローラは何種類もありますが、対応はまちまちのようです。TurboLinux 6.0では、PROMISEのUltra/66やHPT366をサポートしていますが、ディストリビューションによっては、



画面5 Red Hat系ディストリビューションに採用されているPrinttool

インストーラがそれらを認識しないため使えないこともあります。その場合には、UltraDMA/33対応のコントローラで接続することになります。

VIAやINTELのUltraDMA/66対応のチップセット用のドライバは、最新カーネルや、開発版カーネル2.3で実験的にサポートされているので、カーネルを再構築することで利用できる可能性があります。具体的なインストールの方法は、The Linux Ultra-DMA Mini-Howto(<http://www.gee.kyoto-u.ac.jp/LDP/HOWTO/mini/Ultra-DMA.html>)などを参考にするとよいでしょう。



Question 30

ハードディスクのアクセスが遅いような気がします

A. IDEハードディスクのアクセス設定が、PIO転送になっている場合があります。ハードディスクとマザーボードがDMA転送に対応しているのなら、DMA転送に設定することで高速にアクセスできる可能性があります。

下記のようにhdparmコマンドで調べて「using_dma = 0 (off)」と表示されたらDMA転送は使われていないことがわかります。

```
# hdparm -t -d /dev/hda
```

もし、DMAがオフの場合には、

```
# hdparm -t -d 1 /dev/hda
```

とDMAをオンにテストしてみます。エラーが出力されずに転送時間が速くなったなら、/etc/rc.d/rc.localに、「hdparm -d 1 /dev/hda」を加えてブート時に設定するように変更します。



Question 31

64Mバイト以上のメモリが認識されません

A. 現在のディストリビューションでは、64Mバイト以上のメモリを自動的認識するようになっていますが、マザーボードによっては64Mバイトとなってしまうことがあります。その場合は、ブート時にカーネルにオプションをつけることでメモリの容量を指定することができます。たとえば、128Mバイトのメモリを指定するにはLILOと表示された時点で、

```
LILO: linux mem=0x8000000
```

または、

```
LILO: linux mem=128M
```

というように指定します。

毎回ブート時に指定するのは面倒なので、/etc/lilo.confにオプションを書いておきましょう(リスト1)。lilo.confを書き換えたら、/sbin/liloコマンドを実行し、ブートセクタのliloに反映させます。なお、起動後に認識されたメモリ容量を表示するには、freeコマンドを使います。



Question 32

SCSIカードはなにが使えますか？

A. SCSIホストアダプタは、業界標準ともいえるアダプタ社をはじめとして多くのメーカーの製品に対応しています。有力なディストリビューションのWebサイトには、そのディストリビューションが対応しているSCSIホストアダプタのリストが掲載されています。



Question 33

ハードディスクのRAID対応は？

A. COMPAQ SMART2やMylex DAC960といったRAID専用コントローラを使うことができます。RAIDサブシステムのように、ホスト側からは1台のハードディスクドライブに見せるタイプのものは、そのままでもちろん使えます。

カーネル2.2からソフトウェアRAIDが実用的に動作するようになりました。ほとんどのディストリビューションが、そのままソフトウェアRAIDに対応しています。また、Red Hat系のグラフィカルインストーラ(Anaconda)では、インストール時にRAIDデバイスを作成することができ、簡単にソフトウェアRAIDを実現できるようになっています(画

リスト1 メモリ容量を128Mバイトに設定する場合の/etc/lilo.confの例

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz
    label=linux
    root=/dev/hda1
    initrd=/boot/initrd
    append="mem=128M"
    read-only
```

追加



面6) ソフトウェアRAIDに関する情報は、The Software-RAID HOWTO (<http://www.linux.or.jp/JF/JFdocs/The-Software-RAID-HOWTO.html>) を参照してください。



Question 34

USBデバイスは使えますか?

A. LinuxのUSBサポートは、バージョン2.2.7からカーネルのソースに含まれていますが、開発途上のコードのため実際には、Corel LINUXやMandrakeのように海外のディストリビューションで組み込まれている以外、ほとんどの日本語ディストリビューションではサポートされていません。

そのため、USBデバイスはカーネル2.4が正式にリリースされるまで待ったほうがよいでしょう。それまでは、USB接続のマウスやキーボードはPS/2変換コネクタなどを利用して、PS/2デバイスとして接続することをお勧めします。

カーネル2.4を採用したディストリビューションでは、USB接続のデジカメ、プリンタ、スキャナやZIPドライブなどのストレージデバイスもサポートされることでしょう。LinuxのUSBサポートに関する情報は「Linux USB」(<http://www.linux-usb.org/>) を参照してください。



Question 35

IrDAは使えますか?

A. IrDA (赤外線ポート) のサポートは、2.2系カーネルから含まれるようになってきました。シリアルポートのエミュレーションを行うプロトコルであるIrCOMMは安定して動作しているようです。IrCOMMを使うと、IrDA対応のISDN公衆電話やIrDAポートが装備されている携帯電

話とやり取りができるようになります。また、IrDAのプロトコルのひとつであるIrOBEXを使うことで、PalmIIIとデータ交換ができるようになるとドキュメントに記述があります。詳しくは「The Linux IrDA Project」(<http://www.cs.uit.no/linux-irda/>) を参照してください。



Question 36

CD-R/RW、DVD-ROM、DVD-RAM、MOドライブを接続したいのですが

A. これらのドライブにはSCSI接続とIDE接続の2種類ありますが、基本的にどちらでも利用できます。ただし、Linuxで利用できるほとんどの書き込みプログラムは、SCSI接続を前提としているので、IDE接続のドライブの場合にはLinuxのSCSIエミュレーション機能 (ide-scsiドライバ) を利用する必要があります。

DVD-ROMドライブは、CD-ROMドライブと同様に接続し、CD-ROMドライブの代わりに使うことができます。DVD-ROMはUDF (Universal Disk Format) で記録されていますので、LinuxにUDFドライバを組み込むことで、データを読み込むことが可能です。Kondara MNU/Linux 1.1では、標準でUDFドライバモジュールがサポートされています。DVD-ROMをマウントする時に“-t udf” オプションを付けてください。



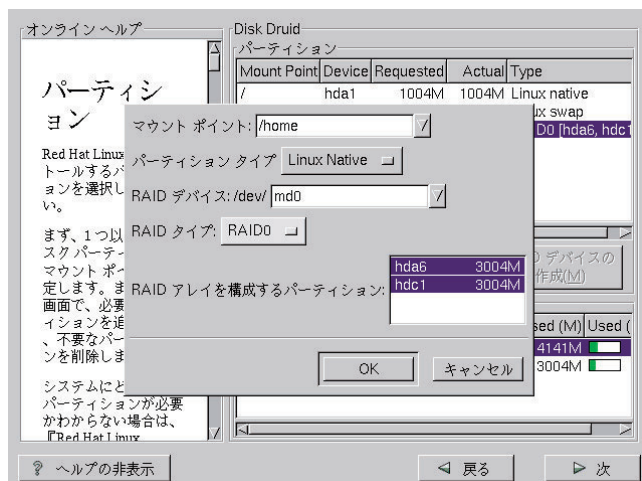
Question 37

CD-R/RWドライブにデータをバックアップするには?

A. LinuxでCD-R/RWにデータを書き込む方法は、大きく3つの手順に分けることができます。まず、適当なディレクトリにCD-R/RWへ書き込むためのファイルを全部コピーしておきます。次に、mkisofsコマンドで、先ほど用意したディレクトリから、ハードディスク上に仮想ファイルシステムのイメージを作成します。念のため、そのファイルシステムをマウントして、内容を確認しておくといでしょう。最後に、そのファイルシステムのイメージを、cdrecordコマンドでCD-R/RWに書き込みます。

コマンドラインでは操作が面倒だという人は、X Window System上で動作するX-CD-Roastを使うとGUIで操作することができます。X-CD-Roastは内部でmkisofsやcdrecordを呼び出しています。

詳しい情報は、CD-Writing HOWTO(<http://www.linux.or.jp/JF/JFdocs/CD-Writing-HOWTO.html>) をご覧ください。



画面6 ソフトウェアRAIDに対応しているグラフィカルインストーラ

システム起動



Question 38

LinuxのインストールCD-ROMから起動できない



A. 古いマシンや古いCD-ROMドライブだと、CD-ROMからのブートはサポートされていません。マザーボードによっては、BIOSをアップデートすればCD-ROMブートが可能になるかもしれません。SCSI接続のCD-ROMドライブの場合、SCSIアダプタもCD-ROMブートに対応していなければなりません。

また、マザーボードのBIOS設定で、CD-ROMの起動順位をハードディスクより先にしておかないと、ハードディスクにあるOSが起動してしまいますので注意しましょう。

CD-ROMブートができない場合、インストールCD-ROMの中に、インストールフロッピーのイメージファイルと、そのイメージをフロッピーディスクに書き込むDOSプログラム (rawrite.exe) が入っているはず。これらを使い、DOSかWindowsでインストールフロッピーを作成して、フロッピーからインストーラを起動しましょう。

Question 39

LILLOプロンプトがLIで止まって起動しないのですが



A. ブート時のLILLOというメッセージは、起動のフェーズによって1文字ずつ表示しているため、どの文字まで表示されたかによって原因を突き止めるヒントになっています。LIで停止したならば、LILLO (Linux LOader) ブートローダがLinuxカーネル (たとえば/boot/vmlinuz) の場所を見つけられない状態です。あらかじめ作成しておいた緊急用の起動フロッピーを使用しましょう。

Linuxをインストール直後に起こった場合には、インストールしたハードディスクのパーティションの設定が、LILLO設定のところで問題があった可能性が高いでしょう。

いままで動作していたシステムで、LILLOの表示途中で止まる場合、ハードディスク内のパーティションを変更したり、カーネル再構築やliloプログラムによってファイルを書き換

えたことが原因で、カーネル (vmlinuz) を見つけられなくなったと考えられます。

LILLOではなく、BIOSの制限を受けないブートマネージャを使う、あるいはフロッピーディスクからカーネルを読み出して起動するという回避方法もあります。

Question 40

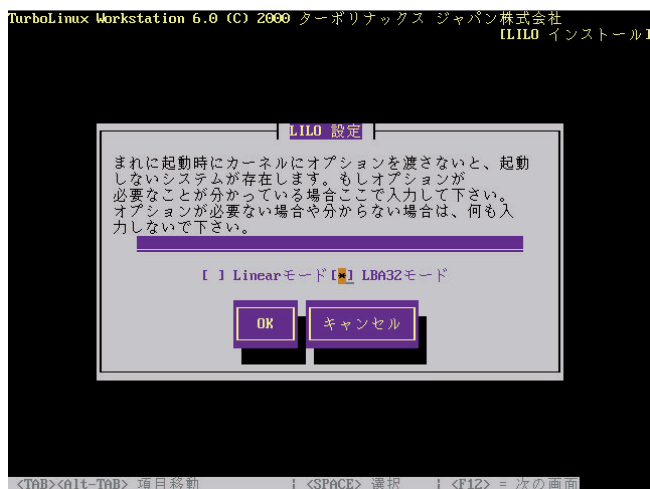
1024シリンダを超えるパーティションにはインストールできないのですか？



A. LILLOブートローダは、BIOSを使用してハードディスクからLinuxカーネルを読み込んでいます。そのBIOSのInt13Hファンクションには1024シリンダ以上へアクセスできないという制限があり、カーネルが1024シリンダ以降の部分に配置されているとLinuxを起動できません。そのため、インストーラが“LILLOをインストールできません”といったエラーメッセージを出すようになっています。

この問題を解決するには、カーネルを1023シリンダ以内に置くか、LILLOを使わずにブートするしかありません。Red Hat Linuxなどでは、カーネルを/bootディレクトリに置くので、ハードディスクの前のほう (1023シリンダ以内) に10~15Mバイト程度のパーティションを作り、これを/bootにマウントすれば問題を回避できます。

しかし、最近のBIOSでは、拡張Int13Hファンクションで32ビットのセクタアドレスが利用できるようになりました。LILLOのバージョン21-3以上なら、2048Gバイトまでのハードディスクをサポートしています。LILLOのバージョンは、“/sbin/lilo -V” とすれば表示されますが、LILLO version 21-3以上を使用している日本語ディストリビューションは、



画面7 TurboLinux Workstation 6.0のインストーラ (LILLO設定) ハードディスクの8.4Gバイト (1024シリンダ) を超えるパーティションにインストールすることが可能なLBA32モードをサポートしている。



2000年6月現在でTurboLinux Workstation 6.0だけのようです。

また、TurboLinux 6.0のインストーラでは、LILO設定(画面7)に「LBA32モード」というオプションが選べるようになり、それをチェックすれば、1024シリンダを超えるパーティションにインストールが可能です。

Question 41



LILOの起動時に、Windowsをデフォルトにしたい

A. 1台のPCにWindowsとLinuxをインストールしてマルチブート環境を構築している場合、“LILO boot:”プロンプトで、linux、またはdosといったラベルを入力することでOSを切り替えられるように設定できます。このとき、単にEnterキーを押すとふつうはLinuxが起動しますが、Windowsをデフォルトにしたいならば、/etc/lilo.confに、“default = 起動したいOSのラベル名”を記述して、/sbin/liloと実行します。

Question 42



起動フロッピーを作りたい

A. ハードディスクからLinuxを起動できなくなったときなどに活躍するのが起動フロッピーです。

ディストリビューションによっては、起動フロッピーを作るためのシェルスクリプトが用意されているものもありますが、ここでは標準的なコマンドを使い、シンプルな起動フロッピーを作る方法を紹介합니다。

```
# fdformat /dev/fd0H1440
# dd if=/boot/vmlinuz of=/dev/fd0
# rdev /dev/fd0 /dev/hda1
```

このフロッピーで起動すると、カーネルをフロッピーディスクから読み出し、/dev/hda1のパーティションを/(ルート)ファイルシステムとしてマウントします。

しかし、カーネルに渡すオプションが入力できないので、ランレベルの指定などはできません。そこで、今度は、オプションを指定できる、もう少し便利な起動フロッピーを作ってみます。このフロッピーには、LILOとカーネルイメージを入れ、フロッピー内のカーネルとハードディスクのカーネルを選べるようにします。

下準備として、フロッピーディスクに書き込むLILOのために、設定ファイル/etc/lilo.flopを作ります(リスト2)。

/etc/lilo.confをコピーして雛形にすると簡単です。

次に、起動フロッピーを作成します。最初にフォーマットし、ext2ファイルシステムを作成します。

```
# fdformat /dev/fd0H1440
# mke2fs /dev/fd0
```

作成したディスクをマウントし、ハードディスクからチェインローダとカーネルをコピーします。

```
# mount -t ext2 /dev/fd0 /mnt/floppy
# cp -p /boot/chain.b /mnt/floppy
# cp -p /boot/vmlinuz-2.2.13 /mnt/floppy
```

最後に、フロッピーディスクにLILOをインストールし、アンマウントすればできあがりです。

```
# lilo -C /etc/lilo.flop
# umount /mnt/floppy
```

この起動フロッピーでブートすると、LILOの“boot:”プロンプトが表示されるので、ここで“linux”と入力すればハードディスクのカーネルが、“rescue”と入力すればフロッピーディスクのカーネルが起動します。さらに、“rescue 1”などと入力すればランレベルの指定も可能です。

ただし、ブートメディアのドライバをモジュールにしてい

リスト2 /etc/lilo.flopの例

```
boot=/dev/fd0
#LILOはフロッピーに置く
map=/mnt/floppy/lilo-map
prompt
timeout=50
default=linux

image=/boot/vmlinuz-2.2.13
#ハードディスクのカーネル
label=linux
read-only
root=/dev/hda1

image=/mnt/floppy/vmlinuz-2.2.13
#フロッピーディスクのカーネル
label=rescue
read-only
root=/dev/hda1
```

る場合は、これらの方法で起動フロッピーを作ることはできません。ブートに必要なドライバを組み込んだカーネルを作る必要があります。



Question 43

ランレベルってなに？

A. Linuxには、ランレベルという複数の動作モードがあり、0~6のランレベルによって、機能の範囲を変えられるようにしています。たとえば、Red Hat LinuxやTurboLinuxでは、表1のようになっています。

通常のテキストコンソール起動時は、ランレベル3で動作し、グラフィカルログインはランレベル5で動作しているというわけです。Linuxは、マルチユーザーOSなので、同時に複数のユーザーで利用できますが、管理者がディスククラッシュを復旧する際などは、ほかのユーザーやデーモンがファイルを読み書きすると非常にマズいこととなります。このようなときには、ほかのユーザーやデーモンを排除するためにシングルユーザーモードで作業をします。

起動時のデフォルトランレベルは、`/etc/inittab`で設定されています。このファイル中に、`id:3:initdefault`と書かれていればデフォルトのランレベルは3になります。これを書き換えるか、起動時にLILOの“boot:”プロンプトで“linux 5”のようにすれば、ランレベルを指定することができます。

一方、起動してからランレベルを切り替えるのがinitコマンドです。たとえば、“init 6”を実行すると、すぐにランレベル6となり、Linuxが再起動します。



Question 44

デーモン (daemon) とはなんですか？

A. ゲームのように、ユーザーと対話しながら表舞台で活躍するアプリケーションとは違い、人目につかないところでひっそりと仕事をする、「縁の下の力持ちプログラム」の総称です。“ps ax”コマンドを実行するとわかるように、Linuxでは数多くのプログラムがデーモンとして動作しています。たとえば、仮想記憶を実現するkswapd、ログを集めてファイルに記録するklogdやsyslogd、各種のネッ

0	停止
1	シングルユーザーモード
2	マルチユーザーモード (ネットワーク機能なし)
3	フルマルチユーザーモード (テキスト)
4	未使用
5	マルチユーザーモード (X Window System)
6	再起動

表1 ランレベルによるLinuxの機能

トワークサーバプログラムなどはデーモンとして動作しています。

起動されたデーモンは、すぐに活動するわけではなく、特定のリクエストを待ち、リクエストを受信するとそれに対する処理を行います。そして処理が済むと、次のリクエストが来るまでまた休むのです。たとえば、Webサーバのhttpdは、ユーザーから接続されるまで休んでいて、接続されると活動を始めます。また、決められた時刻や一定時間ごとにプログラムを起動するcronも代表的なデーモンです。



Question 45

システムサービス (デーモン) の起動、停止の方法を教えてください

A. システムの起動時に、自動的に起動するシステムサービスの設定はツールを利用するのが便利です。TurboLinuxではturboserviceコマンド、Red Hat Linuxならntsysvコマンドが利用できます。また、X Window Systemで動作するLinuxconfやksysvコマンドなどでも設定できます。

すぐにデーモンを起動したいときは、Red Hat Linuxなどの場合、`/etc/rc.d/init.d`ディレクトリ以下にデーモンの起動スクリプトが用意されていますので、起動したいデーモン名のファイルに、引数としてstartをつけて実行します。たとえば、httpd (Webサーバ) を起動したいなら、

```
# /etc/rc.d/init.d/httpd start
```

とすると起動されます。引数にはstopとrestartも使え、それぞれ停止と再起動を行います。

一方、Slackwareなどでは`/etc/rc.d/`ディレクトリに`rc.httpd`のようなファイルが用意されていますので、それを実行してデーモンを起動します。デーモンを止めたい時には、“ps ax”を実行して、止めたいデーモンプロセスのPID (プロセスID) を調べ、killコマンドを使います。



Question 46

ときどき起動にもものすごく時間がかかるのはなぜ？

A. Linuxでは、fsckというファイルシステムのチェックプログラムが使われています。これは、Windowsのスキャンディスクのようなものです。そして、Linuxを一定の回数 (たとえば20回) 起動すると、fsckが自動実行されます (正確にはファイルシステムのマウント回数)。また、前回の終了時に正しくシャットダウンされていない場合も実



行されるようになっていきます。チェックするパーティションが小さければすぐに済みますが、パーティションが大きくなるとそれなりに時間がかかるようになります。

また、メール配送プログラムのSendmailは、起動する際にDNS (Domain Name System) サーバに接続しようとして、このときにDNSサーバが見つからないとSendmailがタイムアウトするまで数分間待たされることになります。不要なサービス (デーモンプログラム) は止めておくといでしょう。

Question 47



rootのパスワードを忘れてしまった!

A. rootユーザーのパスワードを忘れてしまうなどということは、管理者として失格といえるでしょう。かといって、紙に書いてディスプレイの横に貼っておくなどというのはもってのほかです。しかしながら、rootユーザーのパスワードをどうしても思い出せない場合は、Linuxをシングルユーザーモードで起動することで再設定することができます。

さて、rootのパスワードがわからないと、シャットダウンもできないことが多いでしょう。この場合、まずsyncコマンドを実行して、ディスクキャッシュに残っているデータをディスクに書き込みます。次にCtrl + Alt + Deleteキーを押して終了します。

シングルユーザーモードで起動するには、Linuxの起動時のLILOプロンプトで、

```
LILO boot: linux single
```

または、

```
LILO boot: linux 1
```

と入力します。この“linux”は起動するOSを選ぶためのラベルで、環境によっては違うかもしれません。その場合は、“boot:”プロンプトに対してTabキーを押すと、登録されているラベルが一覧表示されるので、その中からLinuxを起動するものを選びます。

シングルユーザーモードで起動すると、パスワードを入力しなくてもシェルのコマンド画面になり、rootユーザーとしてログインされた状態になります。passwdコマンドで新しいパスワードを設定します。

Question 48



シングルユーザーモードでの起動を禁止したい

A. Linuxをシングルユーザーモードで起動できれば、コンソールが前であれば誰でも容易にrootユーザーを乗っ取ることができるということになります。これはセキュリティ上での大きな問題になります。そこで、LILOにパスワードを設定して、シングルユーザーモードでの起動の際にパスワードが必要になるように設定するとよいでしょう。

LILOの起動時にオプションが指定された場合に、パスワードを要求するように設定するには、/etc/lilo.confを編集し、“label=linux”と書かれている行のあとに次の2行を追加します。

```
restricted
password=LILOパスワード
```

「LILOパスワード」には、設定するパスワードを指定しません。編集が終了したら、他のユーザーがパスワードを見れないように、lilo.confのパーミッションを変更します。

```
# chmod 600 /etc/lilo.conf
```

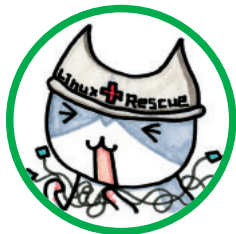
パーミッションの変更を忘れないようにしてください。最後に、/sbin/liloとして変更をliloに反映させます。これで、LILOにオプションを指定して起動しようとする場合に、パスワードが必要になります。

```
LILO boot : linux single
Password :
```

もし、Linux起動時に必ずパスワードが必要になるようにしたい場合は、「restricted」の行を削除してください。しかし、この場合はリモートブートができなくなるという弊害もあります。

LILOのパスワード設定は完全なセキュリティではありません。フロッピーディスクからブートしてファイルシステムをマウントされたり、LILOを書き換えられたりする可能性もあります。より高いセキュリティを望むのであれば、BIOSでフロッピーディスクからの起動を禁止し、BIOSのパスワードを設定するなどの対策が必要です。

シェル



Question 49



シェルってなんですか？

A. コマンドラインに渡された文字列を解釈して、その命令をシステムに渡すものです。Bourneシェル (sh)、Cシェル (csh)、Kornシェル (ksh)、tcsh、bash、zshなどの種類があります。

Question 50



それぞれのシェルはどんな特徴がありますか？

A. シェルは、sh系とcsh系に大別することができます。shはBourneシェル、Bシェルともいわれます。shは、UNIXでは標準的な存在ながら、非常にプリミティブな機能しか持っていません。

一方cshは、ジョブ制御やエイリアスなどshにない機能を持っています。また、C言語に似た文法でスクリプトを記述できるという特徴を持っています。

これらの2つのシェルを中心に拡張されたものがいくつかあります。sh系では、Kornシェル (ksh) やbashなどです。

一方のcsh系には、tcshがあります。tcshは、Cシェルの機能に加えて、コマンド行編集が改善されています。

Linuxで標準的に利用されているbash (Bourne Again Shell) は、shとの互換性だけでなく、cshやKornシェルの機能も取り込んでいます。

Question 51



自分が使っているシェルを知りたい

A. コマンドラインから、

```
$ echo $SHELL
```

と入力します。Linuxではほとんどの場合、bashを使用しているでしょう。

Question 52



シェルを変更したい

A. コマンドラインから、chshコマンドを実行することでシェルを変更できます。さらに、LinuxconfやturbousercfgなどのGUIツールからも変更可能です。

Question 53



.bash_profile、.bashrc、.bash_logoutは、どのような働きをするの？

A. 各ユーザーのホームディレクトリにあるこれらのファイルは、bashの環境変数の定義やプログラムを書いておきます。.bash_profileはログインする際に読み込まれます。.bash_loginや.profileなども同様ですが、ホームディレクトリから、.bash_profile、.bash_login、.profileという順番で探していきます。特別な理由がない限り、.bash_profileを利用すればよいでしょう。

.bashrcはシェルが起動されるたびに読み込まれます。ただし、通常は、ログインシェルとサブシェルで実行できるよう、.bashrcを.bash_profileの中で読み込むような設定になっています。

なお、起動時に/etc/profileがあれば、このファイルが一番先に読み込まれます。

.bash_logoutは、ログアウト時に読み込まれて実行されるファイルです。

Question 54



lsしても、(ドット)で始まるファイルが表示されません

A. 標準ではlsコマンドは、先頭に「.」(ドット)のついたファイル(ドットファイル)は表示しません。表示させたいときは、次のように、-a (all) オプションつきで実行します。

```
$ ls -a
```

Question 55



.bash_profileの変更が有効になりません

A. bash_profileはログイン時に一度しか読み込まれません。変更した設定を有効にしたいなら、

```
$ source ~/.bash_profile
```

として、再度読み込ませることで、変更が有効になります。



Question 56



変数の設定でexportするものとしがないものがあるのですが

A. 「変数名=値」として設定する変数は、「シェル変数」と呼ばれる変数です。これは、シェルスクリプトの値の保存や、組み込み変数と呼ばれるbashが参照する値を設定するためのものです。一方、「export 変数名=値」のように設定する変数は、「環境変数」と呼ばれる変数で、サブプロセスにも伝わります。EDITORなどは環境変数に定義すべき変数です。

Question 57



文字列は'と"のどちらで囲めばいいのでしょうか？

A. 単なる文字列をコマンドに渡すだけなら、どちらでも構いません。しかし、変数などをコマンドに渡すときに'（シングルクォート）と"（ダブルクォート）では、その挙動は異なります。シングルクォートでは変数が展開されず、ダブルクォートでは変数が展開されます。

Question 58



パスを簡単に追加したい

A. bashを使っている場合、パスを設定するには、

```
$ export PATH=<path1>:<path2>
```

としますが、以前に設定されていたパスは、新しく設定したパスで上書きされ、以前のパスは消えてしまいます。以前に設定されているパスに、新たにパスを追加したい場合、全てのパスを入力するのは非常に面倒です。そこで、単にパスを追加するのであれば、環境変数PATHを利用して新しいパスのみを追加できます。

```
$ export PATH=~:/bin:$PATH
```

\$PATHとすることで、現在のPATHを展開することができます。上の例の場合、新しいパスの後ろに、以前に設定されていたPATHが追加されることになります。

ただし、このようにして設定したパスは、設定したシェルでのみ有効になります。また、シェルを終了したりログアウトしてしまえば設定されたパスは消えてしまいます。よく使うパスであれば、.bash_loginや、.bashrcなどに記述したほうがよいでしょう。

Question 59



指定時間にプログラムを実行したい

A. バックアップや、コンパイルなどを行う場合は、誰も使っていない夜間などに行いたいものです。このような場合は、atコマンドを使います。atコマンドを使用すれば、指定時間に自動的にプログラムを起動することができます。atコマンドを使用するには、atdデーモンが起動している必要があります。

atコマンドの書式は、

```
at -f <プログラム名> <実行時間>
```

のように指定します。たとえば、makeしたい場合は、makeを実行するシェルスクリプト（ここでは、atmake）を作成し、次のように指定します。

```
$ at -f atmake 4:00
```

```
warning: commands will be executed using /bin/sh
```

```
job 11 at 2000-06-22 04:00
```

上の例では、4時0分になるとatmakeが実行されます。最下行の「job」の後ろの数値は、ジョブナンバーです。実行されるプログラムはこのナンバーによって管理されます。なお、直接プログラムを指定することもできますが、atコマンドを実行したディレクトリがカレントディレクトリになるので、シェルスクリプトを記述したほうが確実です。

指定できる時間には、次のような形式のものがあります。

HH:MM	23:30	23時30分に実行
HHam	11am	午前11時に実行
HHpm	11pm	午後11時に実行
MM/DD/YY DD.MM.YY	09/01/00	2000年9月1日に実行

また、「now + minutes、hours、days、weeks」形式でも指定できます。nowを使って指定する場合は、

```
$ at -f atmake now + 3 hour
```

というように指定します。上の例では、3時間後に実行するように指定しています。そのほか、「2am today」（今日の午前2時）「11pm tomorrow」（明日の午後11時）といった

指定もできます。

atコマンドで予約されたコマンドは、

```
$ atq
```

とすることでみることができます。このとき表示されるのはジョブナンバーです。

予約されているコマンドを削除するには、

```
$ atrm 11
```

とします。指定する数値はジョブナンバーです。

atコマンドで実行されたコマンドが結果などを標準出力に表示した場合は、その内容が設定したユーザーにメールで送信されます。



Question 60

定期的にプログラムを実行したい

A. 定期的なバックアップや定期的にディスクのエラーチェックを行いたい場合は、cronコマンドを使用します。cronコマンドはatコマンドに似ていますが、atコマンドは一度しか実行されないのに対し、cronコマンドは定期的に繰り返し実行することができます。毎日、毎週、毎月行うバックアップなどに特に向いています。cronコマンドを使用するには、crondデーモンが起動している必要があります。

定期的に起動されるプログラムは、crondデーモンによって管理されています。実行する処理を登録するには、crontabコマンドを使用します。crontabコマンドを使用する場合は、登録するプログラムを記述した指定ファイルを作成します。指定ファイルは次のような書式で作成します。

```
* * * * * <command>
```

最初の5カラムまでは順に、分、時、日、月、曜日となっています。「*」を指定した場合は、その項目を指定していないということになります。曜日は、日曜日が0、月曜日が1.....土曜日が6として指定します（7も日曜日）。複数の条件を指定した場合は、「,」で指定します。たとえば、毎月1日と15日というような場合は、「1,15」とします。

実行するコマンドは<command>に指定します。たとえば、毎月1日と15日の午前2時30分に/usr/local/backup実行したい場合は、

```
30 2 1,15 * * /usr/local/backup
```

という指定ファイル（ここではcrontab）を作成し、

```
$ crontab crontab
```

として実行します。atコマンドと同様に、標準出力はメールで送信されます。新たにジョブを登録すると、以前に登録されていたジョブは上書きされます。複数のジョブを登録する場合は、2行目以降に記述していきます。

登録されているジョブを確認するには、「-l」オプションを付けます。

```
$ crontab -l
```

登録されているジョブを削除するには、「-r」オプションを付けます。また、「-e」オプションを付けて起動すれば、直接登録しているジョブを編集することができます。

rootユーザーであれば、他のユーザーのジョブを確認することができます。この場合は、ユーザー名を指定します。

```
$ crontab -u linuxuser -l
```



Question 61

suコマンドを実行してrootになったのに、管理者コマンドが使えない

A. オプションをつけずにsuコマンドを実行すると、直接rootユーザーとしてログインしたときとは違い、もとのユーザーの環境を引き継いだままroot権限を持ちます。このため、管理者コマンドが置かれた/sbinや/usr/sbinなどにはパスが通っていません。管理者コマンドを利用する場合は、suコマンドに-(ハイフン)オプションをつけてrootユーザーになります。



Question 62

長いコマンドやオプションを入力するのが面倒なんです

A. よく使う長いコマンドやいくつかのオプションを必ず要とするコマンドは、別名(エイリアス)を定義することで短くすることができます。たとえば、「mount /mnt/cdrom」といったよく使うコマンドを、「mtcd」とすることで簡単に入力することができます。別名を定義するには、

```
# alias mtcd='mount /mnt/cdrom'
```



とします。これで、「mtd」と入力すると、「mount /mnt/cdrom」が実行されます。また、「umount /mnt/cdrom;eject」と、2つのコマンドを1つのコマンドとして登録させることもできます。cshを使う場合は、=ではなく、スペースで区切ります。

オプションを指定しない場合は、現在設定してある別名が表示されます。また、設定してある別名を削除するには、

```
$ unalias mtd
```

とします。よく使うエイリアスであれば、.bash_profileなどに記述しておくとういでしょう。



Question 63

バックグラウンドでプログラムを実行したい

A. UNIXは優れたマルチタスクOSですので、いつでも複数のプログラムを実行することができます。1つ1つのプログラムをプロセスといい、ユーザーと対話しているプログラムをフォアグラウンドプロセス、同時に動いているプログラムをバックグラウンドプロセスといいます。バックグラウンドで動いているプロセスは、まさに裏で動いているプログラムという感覚がしっくりします。

プログラムをバックグラウンドで実行するには、コマンド名の後ろに&（アンド）を付けます。たとえば、

```
$ make &
```

というように実行します。&を付けて実行されたプログラムは、バックグラウンドプロセスとして実行されます。コンソール用に作成されたプログラムで、ユーザーと対話が必要ないように設計されたプログラムはバックグラウンドで実行するには向いていません。ただし、ktermなど、X Windows System上のプログラムは、ウィンドウに表示されるのでむしろバックグラウンドでの実行に向いています。

バックグラウンドで実行しているプロセスを確認するには、

```
$ jobs
```

```
[1]  Running                  make &
```

と入力します。[]で囲まれた数値がジョブナンバーです。

バックグラウンドで実行するプログラムが標準出力に結果を表示するような場合は、フォアグラウンドプロセスの表示に影響を与えてしまうため、ファイルにリダイレクトします。



Question 64

プロセスを一時停止したい

A. 実行中のプロセスを一時停止するには、CTRL+Zキーを押します。このとき、プロセスはサスペンド状態となり一時停止します。プロセスの状態を確認するには、jobsコマンドを使います。

再びプロセスの実行を再開したい場合は、

```
$ fg <ジョブナンバー>
```

と入力します。また、停止中のプロセスをバックグラウンドで実行したい場合は、

```
$ bg <ジョブナンバー>
```

と入力します。fg、bgコマンドともに、ジョブナンバーを指定しない場合は、一番最後に一時停止したプロセスを指定したことになります。



Question 65

パスワードを変更したい

A. パスワードを変更するには、passwdコマンドを使用します。passwdコマンドは対話式になっていて、現在のパスワードと、確認のために新しいパスワードを2回入力します。

```
$ passwd
```

```
Changing password for linuxuser
```

```
(current) UNIX password:
```

```
New UNIX password:
```

```
Retype new UNIX password:
```

```
passwd: all authentication tokens updated successfully
```

「(current) UNIX password:」に現在のパスワード（変更する前の古いパスワード）、「New UNIX password:」と、「Retype new UNIX password:」に新しいパスワードを入力します。パスワードが違っている場合は変更することができません。なお、入力したパスワードは、セキュリティのために表示されませんが、タイプしたとおりに入力されます。大文字小文字は区別されますので、CAPSやNumLockの状態に注意しましょう。パスワードに含めることができる文字

は、英数字と一部の記号のみです。

また、ルートユーザーであれば、ユーザーのパスワードを変更することもできます。この場合は、passwdのオプションにユーザー名を指定します。

```
# passwd linuxuser
```

ルートユーザーがユーザーのパスワードを変更する場合は、現在のパスワードを入力する必要はありません。新しいパスワードを2回入力します。



Question 66

プロンプトを変更したい

A. プロンプトとは、コマンドを入力する場所を示すもので、通常は\$（一般ユーザー）や#（ルートユーザー）などで示されます。プロンプトを変更する方法は、使っているシェルによって異なります。Linuxで標準的に使われているbashでは、PS1というシェル変数に設定されています。

MS-DOSのようなプロンプトにしたい場合であれば、次のように設定します。

```
$ PS1="\w>"
```

¥a	ASCIIコードのベル文字（07）
¥d	曜日日月フォーマットの日付
¥e	ASCIIコードのエスケープ文字
¥h	マシン名（ホスト名）
¥H	ホスト名
¥n	改行
¥r	復帰
¥s	シェル名
¥t	HH:MM:SSフォーマットの時間（24時間制）
¥T	HH:MM:SSフォーマットの時間（12時間制）
¥@	HH:MMフォーマットの時間（am/pm時間制）
¥u	ユーザー名
¥v	bashのバージョン
¥V	bashのバージョンとパッチレベル
¥w	カレントディレクトリ（/から表示）
¥W	カレントディレクトリ
¥!	入力するコマンドの履歴番号
¥#	入力するコマンドのコマンド番号
¥n	8進数nnnに対応する文字
¥¥	¥記号（バックスラッシュ）
¥[表示されない文字列の開始。端末制御シーケンスを入れる場合に使用
¥]	表示されない文字列の終了

表2 シェルに指定できるエスケープシーケンス

プロンプトにはエスケープシーケンスと呼ばれる特殊な文字を埋め込むことができます。デフォルト状態で表示されるユーザー名やディレクトリ名はエスケープシーケンスを使用して表示しています。bashで使用できるエスケープシーケンスには表2のようなものがあります。たとえば、「ユーザー名@マシン名:ディレクトリ名\$」というような形式のプロンプトにしたい場合は、次のように指定します。

```
$ PS1="\u@\h:\w$ "
linuxuser@lmtpc06:/usr/local$
```

また、プロンプトに色を付けたい場合は、端末制御シーケンスを使用して次のように指定します。

```
$ PS1="\[\033[32m\][\u@\h \W]$ \[\033[0m\] "
[yoshi-u@lmpc23 yoshi-u]$
```

このように設定したプロンプトは、設定したコンソールや端末エミュレータにのみ反映されます。また、ログオフすれば元に戻ってしまいます。継続してプロンプトを変更したい場合は、ユーザーディレクトリの.bashrcに設定します。



Question 67

コマンドの結果などが長すぎてスクロールしてしまいます

A. コマンドの多くは、結果などを標準出力としてコンソール画面に表示しますが、1画面に収まりきれないほどの情報が表示されることがよくあります。たとえば、「ps aux」などを実行した場合は、1画面に収まりきれないでしょう。

標準出力に表示されるテキストは、パイプ（|）を使ってリダイレクトすることで、別のプログラムに渡せます。

たとえば、1画面ごとに表示をポーズする、moreコマンドやlessコマンドにリダイレクトすれば、1画面ごとにポーズしながら表示させることができます。

```
$ ps aux | more
```

```
$ ps aux | less
```

表示されたテキストは、スペースキーで1画面ごとに、Enterキーで1行ずつスクロールさせることができます。lessコマンドでは、逆方向にもスクロールさせることができます。逆方向にスクロールするにはBキーを押します。more、less



コマンドともに、/`<文字列>`とすることで、任意の文字列を検索することもできます。

**Question 68**

任意の文字列を含むファイルを探したい

A. 任意の文字列を含むファイルを探したい場合は、grepコマンドを使用します。grepコマンドには、探したい文字列と、検索するファイルを指定します。検索するファイルには、ワイルドカードが使用できます。

```
$ grep hogehoge *
```

上の例では、「hogehoge」という文字列を含むファイルを検索しています。また、「-i」オプションを付けると、大文字小文字を区別しません。

もし、特定のディレクトリ以下を検索したい場合は、次のように指定します。

```
$ grep -i -r foo /usr/doc
```

上の例では、「/usr/doc」ディレクトリ以下にある、大文字小文字を問わず、「foo」という文字列を含むファイルを検索します。

**Question 69**

任意のファイルがどこにあるのか探したい

A. あるファイルがどのディレクトリにあるのか探すには、いくつかの方法があります。よく使われるのは、locateコマンド（実際には、slocateへのシンボリックリンク）です。locateコマンドは、あらかじめ作られたディレクトリ構造のデータベースを参照して、任意のファイルを検索します。たとえば、「hogehoge」という名前のファイルを検索するには、

```
$ locate hogehoge
```

とします。locateコマンドはデータベースを使用しているために非常に高速ですが、データベースが更新された後に作成したファイルなどは検索できません。データベースを更新するには、rootユーザー権限で「locate -u」とします。

なお、ワイルドカード（*?）も利用できます。

その他の方法として、findコマンドも利用できます。

```
$ find / -name "hogehoge"
```

上の例では、ファイル「hogehoge」をディスクすべてに対して検索しています。「/」の代わりに、特定のディレクトリを指定することもできます。この場合は、指定したディレクトリ以下を検索することになります。ただし、findコマンドを実行したユーザーがパーミッションを持っていないディレクトリ以下は検索することができません。

なお、locate、findコマンドとも、ワイルドカードが利用できます。

また、プログラムやmanファイルを探すのであれば、whereisコマンドを使うこともできます。

**Question 70**

ファイル名などを簡単に入力したい

A. 長いファイル名などを入力する場合、途中まで入力すれば、残りは自動的に補完して、ファイル名を完成させることができます。たとえば、「packages.rpm」というファイル名を入力したい場合、「pack」まで入力したあとに、TABキーを押すと、「packages.rpm」に補完されます（bashの場合）。ただし、「packages.rpm」の他に、「pack～」で始まるファイルがある場合は、補完されません。この場合は、もう一度TABキーを押すと、「pack～」で始まるファイル名の一覧が表示されます。他のファイルと重複しない部分まで入力し、再びTABキーを押すことで、ファイル名を補完することができます。また、ワイルドカードを利用して、「*.rpm」などという使い方もできます。

ファイル名の補完は、「ファイル名の一部は覚えているけど完全な名前は忘れちゃった」というときにも便利です。

**Question 71**

同じ名前の別のプログラムが実行されてしまいます

A. 違うディレクトリに、同じ名前のプログラムが存在する場合、どちらのプログラムが実行されるかは重要な問題です。確実に目的のプログラムを実行する場合は、ルートからのフルパスで指定するか、相対パスで指定します。

```
$ /bin/ps
```

フルパスで指定

```
$ ../bin/ps
```

相対パスで指定

これは、カレントディレクトリにあるプログラムも例外ではなく、「./」に続けてプログラム名を指定します。

環境変数PATHに指定したディレクトリのプログラムは、ディレクトリの位置を指定しなくても実行できます。この場合は、先に指定した（前にある）ディレクトリから順に検索されます。このため、同一の名前のプログラムがあった場合は、PATHで先に指定したディレクトリにあるプログラムが実行されます。たとえば、次のようにパスを設定した場合、

```
$ export PATH=/bin:/usr/bin:/usr/local/bin:/usr/bin
/X11:/usr/X11R6/bin
```

「/bin」「/usr/bin」「/usr/local/bin」……という順に検索されます。このため、あまり使わないコマンドがあるディレクトリほど後ろに設定したほうが効率が良くなります。

もし、同じ名前の違うプログラムを実行させたい場合は、「パスの順序を変える」「aliasで定義する」「シンボリックリンクを作成する」「相対/絶対パスで指定する」などの方法を使います。また、実際にどのプログラムが実行されるのかを調べるには、

```
$ which hogehoge
```

とします。



Question 72

以前のディレクトリにすぐ戻りたい

A. ディレクトリを移動して作業をしたあと、前のディレクトリに戻りたい場合は、pushdコマンドを使用してディレクトリを移動します。

たとえば、「/tmp」ディレクトリに移動する場合は、

```
$ pushd /tmp
```

とします。これで、移動する前のディレクトリが保存されます。元のディレクトリに戻る場合は、「popd」と入力します。

「pushd」と入力するのが面倒であれば、「alias cd='pushd'」として、cdコマンドをpushdコマンドに置き換えてしまうとよいでしょう。なお、現在保存されているディレクトリを調べるには、dirsコマンドを使います。

```
$ dirs
```

pushd、popd、dirsコマンドは、シェルの内部コマンドです。



Question 73

ホームディレクトリにすぐ移動したい

A. 自分のホームディレクトリに戻りたいということはよくあります。自分のホームディレクトリは、「cd」とだけ入力することで移動することができます。また、ホームディレクトリ自体は「~」（チルダ）で表すことができます。たとえば、「~/hogehoge」とすることで、いつでも自分のホームディレクトリのファイルにアクセスすることができます。



Question 74

ディスクを圧迫している大きなファイルを探したい

A. ディスクの中にあるファイルを探し出す方法はいくつかあります。lsコマンドのオプション「-lRsk」を使えば大きなファイルを探し出すことができます。

```
$ ls -slRk / | sort -nr | less
```

上の例ではlsコマンドオプションに、「ディスクの割り当て量を最初に表示(s)、長い形式(l)、サブディレクトリを再帰的に表示(R)、サイズをキロバイト単位(k)」を指定しています。出力されたlsコマンドの結果をリダイレクトし、sortコマンドで「数値としてソート(n)、大きい順に表示(r)」としてソートし、lessコマンドで表示させています。

また、特定の大きさ以上のファイルを探すには、

```
$ find / -size +10000k
```

とします。上の例では10000kバイト以上のファイルを検索しています。



Question 75

ディレクトリの階層構造を知りたい

A. ディレクトリの階層構造を表示させるには、treeコマンドを使用します。

```
$ tree
```

ディレクトリを指定すると、そのディレクトリ以下の階層を表示することができます。



Question 76

画面をクリアしたい

A. コンソール画面をクリアしたい場合は、clearコマンドを使います。

```
$ clear
```

Question 77

LinuxコマンドとDOSコマンドとの違いは
なんですか

A. LinuxのコマンドとDOSのコマンドは非常に似ています。これは、もともとDOSがUNIXをお手本に作成されたからです。階層ディレクトリ構造などもUNIXをお手本にしています。しかし、一部は微妙に異なっています。たとえば、ディレクトリの区切り文字は、DOSでは「¥」（バックslash）ですが、Linuxでは「/」です。

よく使うコマンドも若干違いますので、表3を参照してください。

Question 78

ログファイルの新しい部分だけを読みたい

A. ログファイルのような大きなファイルの最初の部分や終わりの部分だけを見たい場合は、head、tailコマンドを使うと便利です。headコマンドは、テキストファイルの先頭部分、tailコマンドは終わりの部分を表示します。

```
$ head /var/log/messages
```

```
$ tail /var/log/messages
```

表示する行数を指定したい場合は、行数を指定します。

```
$ tail -20 /var/log/messages
```

上の例では、ファイルの終わりの20行分を表示します。

Question 79

自動的にログアウトしたい

A. bashシェルには、一定時間使用していないと自動的にログアウトする機能があります。この機能を利用すれば、うっかりログアウトすることを忘れてしまうようなことはありません。自動ログアウトを設定するには、シェル変数「TMOUT」を設定します。設定する値は秒です。

```
$ TMOUT=600
```

上の例では、10分間（600秒）シェルのプロンプトが表示されたままになっていると自動的にログアウトするように設定しています。bashrcなどに記述しておくといでしょう。特に、rootユーザーが必ずログアウトするようにしておく安心です。ただし、bashプロンプトが表示された状態で指定時間が経過すると、サスペンド中のプロセスなどがあったとしても自動的にログアウトしてしまうので注意してください。「TMOUT=0」とするとログアウトしません。

Question 80

他のユーザーがログインしているかどうかを調べたい

A. シャットダウンやリブートする場合には、他のユーザーがログインしているかどうかを確認してから行うほうがよいでしょう。他のユーザーがログインしているかどうかを調べる場合は、

```
$ who
```

```
yoshi-u pts/0 Jun 21 17:46
```

```
root pts/1 Jun 28 22:09
```

とします。表示される日付は、ログインした日時です。上の例では、2人のユーザーがログインしています。

また、各ユーザーがどのような処理をしているかを調べるには、wコマンドを使用します。wコマンドでは、ログインしているユーザーがどのような処理をしているのか、どのくらいCPUパワーを占有しているのかが表示されます。

DOS	Linux	機能
attrib	chmod	ファイルの属性の変更
cd	cd	ディレクトリの移動
cls	clear	画面のクリア
copy	cp	ファイルのコピー
del	rm	ファイルの削除
dir	ls	ディレクトリの表示
echo	echo	文字列の表示
help	man	マニュアルの表示
mkdir(md)	mkdir	ディレクトリの作成
more	more	テキスト表示時のポーズ
move	mv	ファイルの移動（ファイル名の変更）
rename (ren)	-	ファイル名の変更
rmdir(rd)	rmdir	ディレクトリの削除
sort	sort	ソート
tree	tree	ディレクトリの階層構造の表示
type	cat	テキストファイルの表示

表3 DOSコマンドとLinuxコマンドの比較

ディスク・ファイル



Question 81



テキストファイルを読みたいのですが

A.

catコマンドを使います。

```
$ cat test.txt
```

とすれば、カレントディレクトリの“test.txt”の内容が表示されます。catは猫のことではなく、“CA Tenate”(連結する)という単語に由来したものです。catコマンドは、複数のファイルをつなげてコンソール(標準出力)に表示するものですが、実際には上のようにファイルを1つだけ指定して使うこともよくあります。

Question 82



ディレクトリを丸ごと削除したいのですが

A.

ディレクトリの削除は、rmdirコマンドで行いますが、rmdirは中身が空のディレクトリしか削除できないので、何段階も作られたディレクトリを消す場合は、めんどろです。

あるディレクトリ以下のファイルやサブディレクトリをまとめて削除するには、“-r”オプションを付けたrm(ReMove)コマンドを利用します。

```
$ ls -FR
hoge.txt  unused/
```

```
unused:
test.txt  test.bak
```

```
$ rm -r unused
```

```
$ ls -FR
hoge.txt
```

このように、“rm -r”とすることで、ディレクトリを中

身ごと消すことができます。しかしLinuxでは、いったん消したファイルは復活できません。間違えて“rm -r”をしてしまうと、貴重なデータを丸ごと失うことになります。そのディレクトリ以下のファイルやサブディレクトリが、本当に必要ないと確認できる場合にだけ、“rm -r”を用いるようにしましょう。

Question 83



フロッピーディスクをフォーマットしたいのですが

A.

rootユーザーになっておいて、

```
# fdformat /dev/fd0
# mkfs -t msdos /dev/fd0
```

とすることで、一般的な2HDのフロッピーディスクをFAT形式でフォーマットできます。

フロッピーディスクやハードディスクを利用するためには、まず一定のサイズの領域(セクタ)に切り分けます(初期化)。続いてOSが利用する管理用の領域を作成します(ファイルシステムの作成)。通常は、この作業を合わせて「フォーマットする」と呼んでいます。

Windowsでフロッピーディスクをフォーマットする際でも、メッセージをよく見ていると、この2つの作業を行っていることがわかります。

Linuxでは、初期化には“fdformat”、ファイルシステムの作成には“mkfs”と別のコマンドを用います。mkfsコマンドは、“-t”オプションで指定したファイルシステムを作成します。Linuxの標準ファイルシステムはext2ですが、フロッピーディスクの場合はFAT形式にして、Windowsマシンでも読み書きできるように、“-t msdos”と指定しておきましょう。

Question 84



ファイル名が“.tar.gz”で終わっているファイルは、何ですか？

A.

関連のあるファイルをひとまとめにした「アーカイブファイル」と呼ばれるファイルです。

プログラムを配布する際には、ソース、ドキュメント、データなど多数のファイルが必要になります。これらをひとつにまとめたものが、アーカイブファイルです。たいていは、さらに圧縮をかけてファイルサイズを小さくしてあります。

Windowsでは、拡張子が“.zip”のZIP形式や、“.lzh”のLHA形式がよく用いられています。

UNIXでは、複数のファイルをひとまとめにするtarコマン



ド、圧縮のためのgzipコマンドを組み合わせるアーカイブファイルを作成するのが一般的です。tarコマンドでひとまとめでしたファイルには、“.tar”という拡張子を付けるのが一般的です。また、gzipコマンドで圧縮したファイルは、“.gz”が名前の末尾に付けられます。こうして“.tar.gz”という拡張子を持ったファイルができあがります。

gzipコマンドが一般的になる以前は、compressというコマンドがファイルの圧縮に用いられていました。compressで圧縮したファイルは、名前の末尾に“.Z”が付けられるので、比較的古いアーカイブファイルには“.tar.Z”という拡張子を持つものがあります。

gzipがデフォルトで用いるLZ77という圧縮アルゴリズムには、特許的に不透明な部分があります。そのため最近では、別のアルゴリズムを用いたbzip2という圧縮コマンドが使われるようになりつつあります。たとえばLinuxのカーネルソースは、gzip版とbzip2版の両方が用意されています。本誌でカーネルソースを収録する際には、より圧縮率の高いbzip2版を使用しています。



Question 85

アーカイブファイルはどうやって元に戻すの？

A. tarコマンドと圧縮コマンド（bzip2、gzipなど）を組み合わせる用います。

圧縮 / 解凍コマンドであるgzipは、tarコマンドと合わせて使うことが多いため、tarコマンドで“z”オプションを指定することにより、自動的にgzipを呼び出すことが可能です。tarコマンドには多くのオプションがありますが、

“f” 続けてファイル名を指定

“t” リスト表示

“x” 解凍

“z” gzipコマンドを呼び出す

“v” 実行中の情報を表示

くらいを覚えておけば、ひととおり利用できます。

```
$ ls
hoge-3.14.tar.gz
$ tar tzf hoge-3.14.tar.gz
hoge-3.14/
hoge-3.14/hoge
:
```

```
:
```

```
$ tar xzf hoge-3.14.tar.gz
$ ls -F
hoge-3.14/ hoge-3.14.tar.gz
```

上記のように、拡張子が“.tar.gz”のファイルの内容を確認するには、

```
$ tar tzf ファイル名
```

解凍するには、

```
$ tar xzf ファイル名
```

です。bzip2コマンドを用いて圧縮した、拡張子が“.tar.bz2”のアーカイブファイルを扱うには、bzip2コマンドを用います。画面のように“-c”を指定することで、bzip2コマンドの出力を直接tarコマンドに渡すことが可能になります。また“-k”を指定しておかないと、アーカイブファイルが消えてしまいます。



Question 86

ほかのマシンで作ったアーカイブを解凍したら、ファイルのパーミッションが変わってしまいました

A. ユーザーIDが異なっているからです。UNIX系のOSでは、ユーザーIDでユーザーを識別しています。ユーザーIDが違えば、たとえ名前が同じでも別のユーザーとして扱われます。ユーザーIDは、/etc/passwdファイルに以下の形式で記録されています。

```
tux-lgo:x:1000:1000::/home/tux-lgo:/bin/bash
```

「:」で区切られた3つめのフィールドがユーザーIDです。LAN環境では、/etc/passwdファイルを共通化するなど

リスト3 拡張子が“.tar.bz2”のファイルの解凍方法

```
$ ls
foobar-2.718.tar.bz2
$ bzip2 -dkc foobar-2.718.tar.bz2 | tar t
foobar-2.718/
foobar-2.718/foobar
:
:
$ bzip2 -dkc foobar-2.718.tar.bz2 | tar x
$ ls -F
foobar-2.718/ foobar-2.718.tar.bz2
```


して、マシン間でユーザーIDとユーザー名を一致させておくといいでしょ。



Question 87

Windowsとテキストファイルをやりとりすると改行がおかしいのですが？

A. DOS / WindowsとUNIX (Linux) では、テキストファイルの改行コードが異なるからです。

テキストファイルの行末には、改行コードがあります。DOS / Windowsでは、「CR」 + 「LF」の2バイトで、改行を表しているのに対して、UNIX系OSでは、「LF」1バイトで表します。このためLinuxで作成したテキストファイルを、Windowsマシンの「メモ帳」で見ようとすると、うまく改行されません (画面8)。

Windows用のテキストエディタでも、秀丸 (シェアウェア) やWZ EDITOR (商用) は、改行コードを自動判別して、保存時に変換することも可能です。またLinuxのEmacs、viなどの最新バージョンでは改行コードを判別可能です。

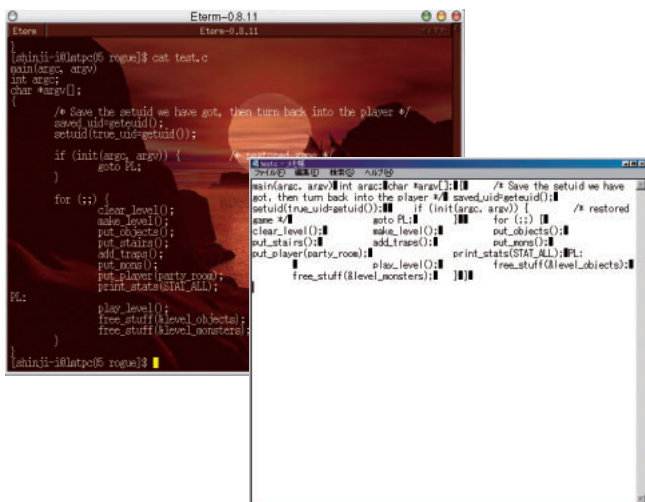


Question 88

コマンドでファイルをゴミ箱に捨てるには？

A. デスクトップ環境であるKDEには、ゴミ箱が備わっています。ファイルマネージャからのドラッグ&ドロップでファイルを捨てたり、メニューから「空にする」ことができるなど、MacOSやWindowsのゴミ箱とほぼ同様の機能を持っています。

KDEのゴミ箱の実体は、カレントディレクトリの /Desktop/Trash/ というフォルダです。たとえばカレントディレクトリのtest.txtをゴミ箱に捨てるには、



画面8 Linuxではきちんと読めたテキストが、Windowsの「メモ帳」では改行されない

```
$ mv ./test.txt ~/Desktop/Trash/
```

のようにします。また、

```
#!/bin/sh
mv $* ~/Desktop/Trash/
```

という内容のスクリプトを「remove」という名で保存し、実行属性を付けてパスの通ったディレクトリに置けば、

```
$ remove ./test.txt
```

のようにして、ファイルを誤って消さないようゴミ箱に捨てることもできます。



Question 89

ファイル名を変更するコマンド (rename) はないの？

A. UNIX系のOSでは、mvコマンドを用いてファイル名を変更します。

```
$ mv foo.txt bar.txt
```

カレントディレクトリの「foo.txt」を「bar.txt」に変更するには、上のようにmvコマンドを使います。また、

```
$ ls -F
bar.txt  subdir/
$ mv bar.txt subdir
$ ls subdir
bar.txt
```

このようにmvコマンドを使うと、ファイルをディレクトリ間で移動させることができます。mvは、ディレクトリ名まで含めた名前の変更を行うコマンドと考えればよいでしょう。



Question 90

ハードリンクとソフトリンクはどう違うの？

A. UNIXのファイルシステムは、ファイルのサイズや日付、パーミッションなどの情報をi-nodeというデータ構造内に収めています。各i-nodeには番号が付けられており、その番号とファイル名の対応表をディレクトリと呼びます。ハードリンクは、同一のi-nodeを複数のファイル名で参照することです。効率的には優れていますが、異なるパー



ディレクトリ間のハードリンクはできません。

ソフトリンクはシンボリックリンクとも呼ばれ、ファイルの中身をファイルへのパスとして認識するものです。MacOSの「エイリアス」、Windowsの「ショートカット」と同じ考え方のものです。リンクファイル 実体のファイルとアクセスされるので、効率は高くないうえに、実体のファイルが削除されてしまってリンクが途切れることもあります。ファイルシステム間にまたがったリンクを張ることが可能です。

Question 91



ファイルの終わりは、どうやって指定するの？

A. ファイルサイズで判断します。Linuxには、以前DOSにあったようなファイル終端記号「EOF」はありません。Linux上で「EOF」がファイルの中間に入っているようなテキストファイルを作成し、Windowsマシンに持っていくと、「EOF」以下が見えません。

Question 92



dfでディスクを見ると100%を超えているんですが？

A. rootユーザー用の予約領域があるからです。

```
$ df
Filesystem 1k-blocks    Used Available Use% Mounted on
/dev/hda5  1976492  1209872    666216   64% /
/dev/hda8  8428164  3427684   4572352   43% /home
```

dfの出力を見ると、ブロック数から計算した使用率よりも、少し大きい値が表示されていることがわかります。これは、各パーティションごとに一定の割合でrootユーザー専用の予約領域があるからです。よってrootユーザー権限でたくさんディスク領域を消費すると、計算上の使用率が100%を超えることがあります。

予約領域のサイズは、tune2fsコマンドによって変更可能ですが、通常は変える必要はありません。

Question 93



40Gバイトのハードディスクをつないだのですが、33Gバイトと認識されてしまいます

A. ハードウェア上の制約で、33.8Gバイト以上のハードディスクは、正しく認識されないことがあります。

歴史的に、IDEハードディスクは、サイズの増加にBIOSの規格決定が追いつかないということを繰り返しています。

つい最近まで、起動カーネルのイメージをディスクの先頭8.4Gバイト以内に置かねばならなかったのも、BIOSが認識できるディスクのサイズが8.4Gバイトに制限されていたからです。

33.8Gバイト以上のハードディスクに対応しているカーネルは、2.2.14以降、2.3.21以降です。またマザーボードのBIOSもなるべく新しいものにアップグレードしておく必要があります。

Question 94



2つのファイルの差違を調べたい

A. diffコマンドを使います。ソースファイルをアップロードしたときのように、一部分だけが異なる2つのファイルと比較するのに適しています。書式は以下のとおりです。オプションによって、数種類の出力形式を選ぶことができます。

```
diff [オプション] file1 file2
```

ソースで提供されているプログラムに対して、「パッチ」と呼ばれる修正が配布されることがあります。パッチの実体は、新旧のソースファイル間でdiffを取ったものです。

Question 95



あるディレクトリ以下のファイルをすべて調べするには？

A. findを使います。findは、ファイル名、パーミッション、作成時刻などさまざまな条件を組み合わせて、ファイルを選択できるコマンドです。

たとえば、「/etc以下のディレクトリで、rootユーザーだけに読み書きが許されているファイル」をすべて表示するには、以下のように指定します。

```
# find /etc -perm 600 -user root
```

デフォルトでは、指定した条件に合ったファイル名をフルパスで出力しますが、さらにコマンドを指定して、各ファイルを処理することも可能です。

Question 96



ログファイルやバックアップファイルを削除したい

A. findを使って、削除したいファイルを見つけて削除しましょう。

ログファイルやバックアップファイルのように、決まった拡張子を持ったファイルを探し出すには、findが有効です。たとえばホームディレクトリ以下にあるログファイル（拡張子“.log”）をすべて削除するには、

```
$ find ~ -name "*.log" -exec rm {} \;
```

と指定します。また、バックアップファイル（拡張子が“.bak”か、“ ”で終わる）をすべて削除するには、

```
$ find ~ \( -name "*.bak" -o -name "*~" \) -exec rm {} \;
```

と指定します。

Question 97

特定の文字列を含むファイルを探したい

A. grepを使います。たとえばカレントディレクトリにあるCのソースファイル（拡張子が“.c”）から、“test”という文字列を探すには、

```
$ grep 'test' *.c
```

のように指定します。

grepは、“Global Regular Expression Print”の略であり、正規表現（Regular Expression）と呼ばれる、より柔軟に文字列を指定が可能です。たとえば、「test1、test2、...test9」といった文字列を探すには、

```
$ grep 'test[1-9]' *.c
```

のように指定します。

Question 98

起動時にプログラムを自動的に実行するには？

A. /etc/rc.d/rc.localにコマンドを書き加えましょう。Linux起動時には、多くの設定ファイルが読み込まれ、環境が構築されていきます。/etc/rc.d/rc.localは、ほかの設定の終了後に読み込まれる設定ファイルで、全ユーザー共通のカスタマイズは、このファイルで設定するのがいいでしょう。

たとえば、起動時に/tmp以下をクリーンアップするには、以下の1行を末尾に加えます。

```
rm -rf /tmp/*
```

また、IDEハードディスクの転送モードなども、rc.localで設定するのに適しています。UltraDMA/33のように高速なDMA転送をサポートしているチップセットでは、

```
hdparm -d1 /dev/hda
```

としておくことで、DMA転送を利用できます。

Question 99

FTPするとファイルが化けるんですが？

A. 転送モードをバイナリにしましょう。FTPプロトコルには、アスキー/バイナリの2種類の転送モードがあります。アスキーモードでは、「CR」+「LF」というデータをやりとりすると、「CR」を削除するようになっています。これはテキストの改行コードが、DOS/Windowsでは、「CR」+「LF」なのに対して、UNIXでは「LF」のみであるという違いを吸収するための仕組みです。

FTPクライアントは、転送するファイルを自動的に判別して、アスキー/バイナリモードを切り替えるようになっていますが、時には判別を失敗することがあります。その場合、プログラムや画像データなどバイナリのファイルに含まれている「CR」+「LF」というデータから「CR」を削除してしまうので、内容が元のファイルと異なってしまう、つまり「化けて」しまいます。

テキストファイルの改行コード処理は、エディタなどでも行えますので、FTPでファイル転送をする場合には、常にバイナリモードで行うほうがいいでしょう。

Question 100

フロッピーディスクやCD-ROMをマウントするってなに？

A. WindowsやDOSとは違い、Linuxにはドライブレーンという概念はなく、すべてのファイルがひとつの木構造に収まっています。すなわち、/（ルート）ディレクトリの下にすべてのディレクトリ、サブディレクトリ、ファイルがあります。

フロッピーディスクやCD-ROMを使うときは、この木構造の一部にこれらをくっつけて（マウント）使います。マウントするにはmountコマンドを使います。例として、/mnt/cdromというディレクトリ（これをマウントポイントという）にCD-ROMをマウントしてみましょう。



```
# mount -t iso9660 -r /dev/cdrom /mnt/cdrom
```

-t iso9660というオプションは、CD-ROMで使われるISO-9660というファイルシステムを使うという指定です。

-rは、読み出し専用でマウントするというオプションです。マウントしたあとで、“ls /mnt/cdrom”と入力すると、CD-ROMの内容が表示されます。

CD-ROMを取り出す時には木構造から外す（アンマウント）作業が必要です。これにはumountコマンドを使います。

```
# umount /mnt/cdrom
```

このようにして、ひとつの木構造を保ったまま、リムーバブルメディアを利用できるのです。

リムーバブルメディア以外にも、ハードディスクをマウントすることももちろん可能です。実際に、Linuxが起動するときには、ハードディスクのパーティションを/（ルート）にマウントします。環境によっては、/bootや/homeなどを別々のパーティションにマウントしていることもあります。



Question 101

rootユーザーしかマウントできないの？

A. ほとんどのディストリビューションでは、rootユーザーにしかマウント/アンマウントを許可していません。

一般のユーザーにもCD-ROMのマウント/アンマウントを許可する場合は、/etc/fstabファイルをエディタで開き、CD-ROMデバイスの行の第4フィールドに“user”、または“users”を加えます。たとえば、第4フィールドが“noauto,ro”になっているなら、“noauto,ro,user”のようにします。フロッピーディスクなどでも同じ要領で設定できます。

“user”を加えた場合は、誰でもマウントできますが、そのデバイスをマウントしたユーザー以外はアンマウントができなくなります。一方、“users”を加えた場合は、ほかのユーザーでもアンマウントできるようになります。

安全のため、ハードディスクのマウント/アンマウントは、rootユーザーだけが行えるようにします。



Question 102

アンマウントできないのはなぜ？

A. umountコマンドやejectコマンドを使おうとすると“device is busy”と表示されてアンマウントできな

いことがあります。このようなときは、アンマウントしようとしているファイルシステムを利用しているユーザーやプログラムがあるということです。たとえば、/mnt/cdromにCD-ROMをマウントしているときに、カレントディレクトリが/mnt/cdrom/fooなどになっているとアンマウントできません。



Question 103

CD-ROMを自動的にマウントできませんか？

A. automountデーモンを使ってCD-ROMなどを自動的にマウントすることができます。/misc/cdにCD-ROMデバイスを自動マウントする設定を説明します。

エディタで/etc/auto.masterを開き次のように記述します。

```
/misc /etc/auto.misc --timeout 60
```

続いて、エディタで/etc/auto.miscを開いて次のように記述します。

```
cd -fstype=iso9660,ro :/dev/cdrom
```

2つのファイルを用意したら、automountデーモンを起動します。Red Hat LinuxやTurboLinuxでは、/etc/rc.d/init.d/autofs startを実行すれば起動します。この設定例では、CD-ROMをドライブに入れるとすぐに/misc/cdにマウントされ、その後、60秒間使われないと自動的にアンマウントされます。そして、またアクセスされると自動的にマウントします。



Question 104

ディスクの使用状況を知るには？

A. dfコマンドを実行すると、各パーティションの利用状況が表示されます。

次の例では、3つのパーティションが、それぞれ/、/boot、/varにマウントされていることと、パーティションごとの利用している容量、全体容量と利用率がわかります。

```
# df
Filesystem 1k-blocks  Used  Available Use% Mounted on
/dev/hda7   5644996  973560  4384680  18% /
/dev/hda1    23302    4013    18086  18% /boot
/dev/hda6   401548  152060  249488  38% /var
```


またduコマンドは、ディレクトリごとのファイル容量を表示します。

```
# du /usr/man/
9956    /usr/man/man3
9364    /usr/man/man1
      :
8       /usr/man/man9
2832    /usr/man/mann
28344   /usr/man
```

数字は、各ディレクトリのファイル容量です。デフォルトではKバイト単位となるので、/usr/man以下に全部で28344Kバイトのファイルがあることがわかります。



Question 105

Linuxでは最大でどのくらいの大きさのファイルが作れますか？

A. Linuxで標準的に使われているext2ファイルシステムでは、1つのファイルの最大サイズは2Gバイトという制限があります。



Question 106

128Mバイトのメモリを搭載しています。スワップはどれくらいが適当ですか？

A. 利用するプログラムの種類、数、負荷によって最適なスワップの大きさは変わるので、これが正解というものはありません。一般には、64~127Mバイトもあれば十分です。古いカーネルでは、1パーティションで128Mバイト以上のスワップスペースは利用できません。いずれにしても、このスワップを使い切るようであれば、パフォーマンスは著しく低下していますから、素直にメモリを増設しましょう。



Question 107

消したファイルを復活できますか？

A. いいえ、できません。rmコマンドに-iオプションをつけると、ファイルを消す前に、本当に消すかシステムが確認してきますので、活用しましょう。



Question 108

DOSのフロッピーディスクを読み書きできますか？

A. DOSのフロッピーディスクは、LinuxではDOSファイルシステムとして認識されますが、カーネルがDOSファイルシステムをサポートするようにコンパイルされ

ていないと利用できません。利用しているカーネルがサポートするファイルシステムは、/proc/filesystemsに書かれています。

次の例では、DOSのフロッピーディスクを/mnt/floppyにマウントしています。

```
# mount -t msdos /dev/fd0 /mnt/floppy
```

取り出す前には、アンマウントしてデータがすべて書き込まれるのを確認します。アンマウントしないで取り出してしまうとデータが破損します。

```
# umount /mnt/floppy
```

ディストリビューションによっては、DOSのフロッピーディスクを使うためのmtoolsというツールが収録されています。これは、UNIX系のOSでDOSのフロッピーディスクを使うためのコマンドラインツール集です。マウントすることなく、ファイルのコピーやフォーマットができます。

使っているディストリビューションに収録されていないければ、<http://mtools.linux.lu/>から入手することができます。



Question 109

Macintoshのフロッピーを読み書きできますか？

A. MacOSのHFSファイルシステム用モジュールをインストールすれば、可能です。

デフォルトで、HFS用モジュールを組み込んだカーネルを用意しているPC用ディストリビューションはほとんどありません。しかし、多くのディストリビューションで、HFS用のローダブルモジュールが用意されていますので、以下の手順でインストールします。

```
# insmod hfs
```

HFSファイルシステムを使えるようになったら、マウントします。

```
# mount -t hfs /dev/fd0 /mnt/floppy
```

ただし、Macintoshの2DDフロッピーディスク(800Kバイトフォーマット)はPCで利用できません。これはハードウェアの制限なので、ソフトウェアではどうしようもありません。



Question 110



Windowsパーティションを
読み書きできますか？

A. カーネルがvfatファイルシステムをサポートしていれば、Windows 95/98のパーティションを利用できます。しかし、そのままでは日本語のファイル名を扱えませんが、パッチを当ててカーネルを再構築する必要があります。このパッチは、小柳雅明さんが作成したものを、川口浩さん、松嶋 明宏さんが改良したものです。最新版は松嶋さんのWebページ (<http://www.jaist.ac.jp/amatsus/>) で公開されています。

ディストリビューションによっては、vfatファイルシステムで日本語のファイル名を扱えるカーネルを採用しているものもあります。

Question 111



ディスクを増設して/homeディレクトリの
容量を増やしたいのですが？

A. 新しいパーティションを適当なところにマウントして、/homeからシンボリックリンクを張ります。

まず、増設するディスクを接続してLinuxを起動します。増設したディスクを/dev/hdbとすると、次のようにfdiskコマンドを起動して必要な容量のLinuxパーティションを作ります。

```
# fdisk /dev/hdb
```

ここでは、新しいパーティション/dev/hdb1を作ったとして説明を続けます。このパーティションにext2ファイルシステムを構築します。

```
# mke2fs /dev/hdb1
```

利用する前に、fsckを実行してファイルシステムを検査します。

```
# fsck /dev/hdb1
```

ファイルシステムに問題がなければ、/disk2というマウントポイントを作り、/dev/hdb1を/disk2にマウントします。

```
# mkdir /disk2
```

```
# mount -t ext2 /dev/hdb1 /disk2
```

次に、/homeの内容を/disk2にコピーします。

```
# cd /disk2
```

```
# tar -cf - -C / home | tar -xvpf -
```

この例では、カレントディレクトリを/disk2に移し、tarコマンドを使って/にあるhomeディレクトリ以下を/disk2にコピーしています。コピーが終わったら、/homeの中にあるディレクトリとファイルを消し、/homeから/disk2/homeへシンボリックリンクを張ります。

```
# rm -rf /home
```

```
# ln -s /disk2/home /home
```

最後に、/etc/fstabに次の行を追加して、起動時に/dev/hdb1がマウントされるようにします。

```
/dev/hdb1 /disk2 ext2 default 1 1
```

Question 112



Linuxのパーティションが
削除できないのですが？

A. Linuxのfdiskなどで作ったパーティションには、DOSやWindowsのFDISKでは消せないものがあります。Linuxのインストールディスクで起動して、Linuxのfdiskを使って消しましょう。

Question 113



Linuxを削除したけど、LILOが消えません。
どうしたら消せますか？

A. MBR (マスターブートレコード) に居座ったLILOを消すには、DOSかWindows 95/98のコマンドラインから“FDISK /MBR”を実行します。これで、MBRが新しいものに書き込まれるのでLILOを消すことができます。Windows NT / 2000しかないときは、Windowsの起動ディスクでブートして [修復セットアップ] を行います。

Question 114



Windows 2000のパーティションはマウントで
きますか？

A. 残念ながら、現時点ではWindows 2000のNTFS 5はLinuxでマウントすることはできません。なお、Windows NT 4.0のNTFSはマウントできます。

日本語



Question 115

Linuxで使われている漢字コードは？

A. Linuxでは通常、日本語EUCが使われています。WindowsやMacintoshで作られた日本語のテキストはシフトJIS (MS漢字コード) になっています。またメールの本文にはISO2022-JP (JIS) コードが使われています。

Question 116

日本語が表示されないのですが

A. 環境変数LANGが設定されていないために正しく表示できないことがあります。その場合には “ export LANG=ja_JP.ujis ” (LANG=jaや、LANG=ja_JP.eucJPが正しいディストリビューションもある) とします。逆に英語表示にするなら “ export LANG=C ” とします。

表示が正しくできるようになったら、.bash_profileにその設定を記述しておけば次回からも有効になります。

Question 117

日本語ロケールってなんですか？

A. ロケール (locale) とは言語環境を指定するもので、プログラムはあらかじめ用意されているロケールデータベースを参照することによって、その国の言語や文化に合わせた動作をすることができます。漢字などの文字コードの違いはもちろんですが、プログラム内部では単純な数値として計算したものを、表示する時にその国に合わせて出力することもあります。たとえば、通貨を “ US \$ 49.98 ” や “ 980円 ” のように表示したり、日付を “ 6/20/2000 ” や “ 平成12年6月20日 ” のように表記したりすることがあります。

日本語に対応したLinuxでは通常、環境変数LANGやLC_ALLに “ ja_JP.ujis ” (日本語EUC) を指定します。しかし「Linuxにおける日本語ロケールに関する指針 (1999年12月15日)」の中で、“ ja_JP.eucJP ” (日本語EUC) を使うことが推奨されたため、最近のディストリビューションで

は “ ja_JP.eucJP ” が使われています。なお、Kondara Projectでは、今後EUCからUTF-8コードセットに移行していくことを表明しています。

Question 118

かな漢字変換システムを切り替えるには？

A. Kondara MNU/LinuxやVine Linuxでは、Xのグラフィカルログインで、日本語入力XIM (X Input Method) を選ぶことができます。また、LASER5 Linuxなら、GNOMEメニューから [システム] [日本語入力システム切り替え] を選ぶとXIM Switcherが起動し、変換サーバを切り替えることができます。

なお、コンソールからコマンドで切り替える場合には、Vine Linuxはsetimeコマンド、Kondara MNU/Linuxはsdrコマンドを使用します。

Question 119

コンソールでログインしたときに日本語表示をするには？

A. コンソールでログインすると英語モードになっているために日本語が表示できません。ほとんどの日本語Linuxディストリビューションに含まれている、kon (漢字コンソールエミュレータ) プログラムを実行すると、日本語が表示できるようになります。

Red Hat Linux日本語版のkonで、manやlessコマンドの時にエラーになる場合、

```
export TERM=vt100
```

として環境変数TERMを変更すればよいでしょう。

なお、ログイン後に自動的にkonを起動するためには、そのユーザーのホームディレクトリにある .bash_profileに、以下のプログラムを挿入します。

```
if [ "$TERM" = "linux" -a "$KONFLAG" = "" ] ; then
    export KONFLAG=true
    exec kon
fi
```

また、Kondara MNU/Linuxには、konが含まれていないので、カーネルをフレームバッファ対応 (Kondara 1.1なら、/boot/vmlinuz-2.2.14-5k4fb) に変更し、jfbtermプログラムを使用します。



Question 120

日本語が文字化けして読めません

A. 元のファイルの漢字コードが日本語EUCでないことが考えられます。WindowsやMacintoshのファイルは、シフトJISなので、それらのファイルを単にcatなどで表示すると文字化けしてしまいます。

Windowsからtelnetで接続する時には、漢字コードにEUCを使うようにtelnet (Tera Termなど) プログラムを設定します。

Question 121

X上のターミナルウィンドウで日本語が表示できません

A. ktermやrxvtを使っていますか? xtermでは日本語は表示できません。ktermを起動するには、コマンドラインからktermと入力するか、日本語化されたGNOMEのメニューなどから [ユーティリティ] [日本語端末] を選びます。ファイルマネージャで /usr/X11R6/bin ディレクトリを表示し、ktermのアイコンをマウスでドラッグ&ドロップすれば、デスクトップや下段のパネル上にショートカットを作ることができるので、便利に使えるでしょう。

Question 122

画面がぐちゃぐちゃになってしまったのですが

A. konやX上のターミナル画面やtelnetなどで、誤ってバイナリファイルや、異なる漢字コードのファイルなどを表示してしまった場合に、画面9のようにぐちゃぐちゃになってしまうことがあります。このような場合は、次のように入力することで画面をリセットすることができます。

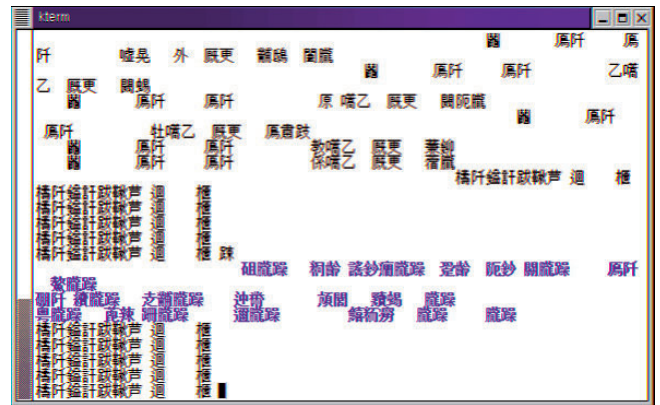
```
reset
```

もちろん、画面上で文字化けしてしまっているので、タイプした文字が読めませんが、間違えないようにタイプします。

上記の方法で画面が戻らない場合は、次のように入力します。

```
echo ^[c
```

ただし、“^[]” はエスケープという特殊な文字のため、キーボードからそのまま入力できるわけではありません。上記のように入力するには、“echo”、スペースキーとタイプした



画面9 バイナリファイルを表示してしまったところ

あと、CTRL + V、ESCキー、“c” と入力し、ENTERキーを押します。

Question 123

日本語のファイル名は使えますか?

A. Windowsに慣れたユーザーには、日本語のファイル名を利用したいという方も多いでしょう。結論からいうと、「使えるが、使わないほうがいい」という回答になります。日本語ファイルそのものは、cat、cp、mvといった各種コマンドは対応しており、問題なく利用することができます。また、bashならファイル名補完も行うことができます。以前は、うまく表示できなかったlsコマンドも、最近のGNU fileutilsのものがインストールされていれば問題なく表示できます。

しかし、GUIでファイルを扱うことが多いWindowsと異なり、CUIでの作業が多いLinuxでは、ファイル名入力のたびごとにかな漢字変換をオン/オフするのはかなり煩雑な作業で、あまりお勧めはできません。

Question 124

Windowsで作られたフロッピーの日本語ファイル名が表示されません

A. Windowsで作られたフロッピーのディレクトリを表示するために、mdirコマンド (MS-DOSフロッピーをLinuxからマウントせずに扱えるツール) を使っても日本語に対応していないため、日本語ファイル名がXXXXと表示されます。フロッピーを、/mnt/floppyディレクトリにVFATファイルシステムとしてマウントしましょう。

```
# mount -t vfat /dev/fd0 /mnt/floppy
```

とすれば、lsコマンドなどで日本語のファイル名として扱え

ようになります。なお、フロッピーをイジェクトする前に必ずumount /dev/fd0としてください。

ただし、カーネルのVFATファイルシステムドライバに日本語対応のパッチが当たっていない場合には、正しく日本語表示できません。日本語ディストリビューション以外を使用しているのなら、ファイル名には英数字だけを使うほうがよいでしょう。



Question 125

シフトJISのファイルをEUCに変換したいのですが

A. nkf (ネットワーク漢字コード変換フィルタ) という漢字コード変換コマンドがありますので、それを使ってみましょう。たとえば、シフトJISで書かれたsample.txtというファイルを日本語EUCにするには、

```
% nkf -e sample.txt
```

と行います。nkfは標準入力から受け取って、変換した結果を標準出力にアウトプットできますので、パイプ(|)を使ってほかのプログラムとの入出力をつないでいくことができます。



Question 126

日本語のmanが表示されません

A. Vine LinuxやLASER5 Linuxなどでは、日本語のmanページを表示するためのプログラムとしてjmanが用意されていますので、“jman <コマンド名>”と実行します。

jmanがないディストリビューションでは、環境変数LANGの設定を確認しましょう。“env | grep LANG”と入力して“LANG=ja_JP.ujis”が表示されないなら、“export LANG=ja_JP.ujis”を設定します(LANG=jaやLANG=ja_JP.eucJPが正しいディストリビューションもある)。

なお、それでも日本語表示されない場合は、日本語のmanページがインストールされていない可能性もあるので、/usr/manディレクトリに、ja_JP.ujisディレクトリなどがあるかを確認します。



Question 127

日本語の入力をするには?

A. ほとんどの日本語ディストリビューションにはかな漢字変換システム(ATOKやWnn6、Cannaなど)

とXのインプットメソッド(kinput2)がバンドルされており、しかもすぐに入力ができるように設定されています。たとえば、X Window SystemでCannaを使って日本語を入力するには、Shift + スペースキーを同時に押すことで、かな漢字変換モードに入ります。入力した単語はスペースキーで変換/次候補、Enterキーで確定します。

Shift + スペースキーを押しても変換モードにならない場合には、“ps ax”で表示される実行中のプロセスをチェックします。jserver、cannaserver、kinput2といったプログラムが起動されていないか、日本語入力のできない端末(xtermなど)で利用しようとしている可能性がありますので、確認してみてください。



Question 128

フリーで利用できるLinuxのかな漢字変換システムにはどのようなものがありますか?

A. フリーで利用できるものにはWnn4、FreeWnn、Canna、SKK、sj3、T-Codeなどがあり、多くの日本語対応ディストリビューションのFTP版には、CannaやFreeWnnが収録されています。

・Wnn4

Wnn(うんぬ)は、京都大学、オムロン、アステックによって共同開発された、クライアント/サーバ方式のかな漢字変換システムです。かな漢字変換サーバは「jserver」が担当し、フロントエンドプロセッサ「uum」、Xのインプットメソッド「kinput2」「Xwnmo」、マルチリンガルエディタ「mule(Emacs)」などから利用します。Wnn4は、1989年にリリースされたもので、各種ヨーロッパ言語、日本語、中国語(簡体字、繁体字)、韓国語に対応した変換サーバ(日本語変換サーバ部分はjserver)と各言語用の辞書が含まれており、マルチリンガル入力システムという特徴を持っています。なお、Wnn4.2をGPLに即した形に再整備したFreeWnn 1.0が、FreeWnnプロジェクト(<http://www.freewnn.org/>)によって1999年4月に公開されました。

・Canna

Canna(かな)は、NECによって開発されたクライアント/サーバ方式のかな漢字変換システムです(<http://www.nec.co.jp/japanese/product/computer/soft/canna/>)。かな漢字変換サーバは「cannaserver」が担当し、フロントエンドプロセッサ「canuum」、Xのインプットメソッド「kinput2」、マルチリンガルエディタ「mule」などから利用



します。どのような環境からでも統一的なインターフェイスによって入力ができるという特徴を持っています。また、カスタマイズファイルである.cannaも一元化されています。

・SKK

SKKは、現京都大学の佐藤雅彦氏によって開発されたかな漢字変換システムです (<http://skk.kuis.kyoto-u.ac.jp/skk/>)。文法解析をいっさい行わず、漢字変換部分や送り仮名の始まりを、ユーザーが大文字で入力することにより指定するという、変わった特徴を持っています。一見、面倒な印象を持つかもしれませんが、ユーザー自身が変換を完全にコントロールできるため、ストレスが少なく、根強い人気があります。開発当初は、EmacsLispだけで記述されたものでしたが、現在は共有辞書にアクセスするサーバ「skkserv」も作られており、こちらも利用できるようになっています。また、フロントエンドプロセッサ「skkfep」やXの入ットメソッド「skkinput」なども開発が進められています。



Question 129

LinuxにはWindows並の変換精度のかな漢字変換システムはないのですか？

A. かな漢字変換システムの変換精度は、とても開発に手間がかかるものであり、フリーで開発されることが多いUNIXのかな漢字システムは、メーカーが開発している製品ほどの変換精度がなく、UNIXの弱点といわれていました。しかし、状況は好転し、次に述べるような商用の変換システムがLinuxにバンドルされるようになっており、変換精度もWindowsに引けをとらないレベルに達しています。

・ATOK12 SE for Linux

ATOK12 SE for Linuxは、ジャストシステムによって開発されているかな漢字変換システムで、WindowsやMacintosh環境で人気の高いATOKのLinux版です。ただし、単体販売されずOEM提供だけとなっているので、利用するには、バンドルされているディストリビューション製品を購入するしかありません。なお、R.2という最新バージョンになって使い勝手が向上しています。

・Wnn6 for Linux

Wnn6 for Linuxは、オムロンソフトウェアによって販売されているクライアント/サーバ方式の商用かな漢字システムです。最新バージョンは3.01です。フリーで配布されていたWnn4から変換効率やカスタマイズ機能が強化されていま

す。具体的には、Wnn4に文節区切りを判断させるアルゴリズムを追加して、文節区切りの間違いによる誤変換を減らしたことや、「FI関係辞書」という用例を記述した辞書によって、同音異義語の選択を最適化した「FI変換」と呼ばれる機能、さらに、ユーザーの確定情報を学習しておく「FI学習」という機能が追加されたことにより、変換効率がWnn4よりも飛躍的に良くなっています。

・VJE-Delta Ver.3.0 for Linux/BSD

VJE-Deltaは、ボックスから販売されているLinuxおよびFreeBSDで動作する日本語入力システムです。MS-DOSの時代から連文節変換を実現していたVJEの最新版で、豊富な辞書と最新の文法解析技術、AI変換アルゴリズムによって最適な日本語変換を実現しています。また、日本語ワードプロセッサVJE-Penがフリーソフトとして添付されています。



Question 130

日本語関連の便利なツールがあれば教えてください

A. 漢字かな混じり文をひらがな/カタカナ/ローマ字のテキストに変換するKAKASI (KANji KANA Simple Inverter) というソフトウェアがあります。テキストファイル中に読めない漢字がある場合や、漢字にふりがなをつける場合などに便利に利用できるでしょう。

また、日本語テキストを全文検索するシステム「Namazu」 (<http://www.namazu.org/>) があります。コマンドラインやX上のクライアント、Webサーバ用のCGIも用意されています。



Question 131

Li18nuxとはなんですか？

A. Linux Internationalization Initiative (リ눅クス国際化企画) を略したもので、Linuxのシステムやアプリケーションなどを国際化するために集まったボランティアベースのワーキンググループです。ここでいう国際化というのは、「複数の国や地域の言語や習慣に対応させること」を指します。

Internationalization (国際化) の先頭と末尾のアルファベットの間の文字数を数字で表記しています。またL10Nは、「Localization」を意味し、「ある特定の国や地域の言語や習慣に対応させること」を指します。さらにM17Nは、「Multilingualization」を意味し、「国際化を言語だけの観点からとらえた」ものです。I18NとM17Nは、よく混同している人もいますので、注意してください。

X Window System



Question 132



X Window Systemとはなんですか？

A. UNIXで標準的に使われているウィンドウシステムです。元々、MIT（マサチューセッツ工科大学）とDECによって始められたAthenaプロジェクトによって開発されたもので、参考にした「W」という名のウィンドウシステムの次という意味から「X」という名がつけられています。最新のバージョンはX11R6.4で、このバージョンよりThe Open Groupが開発し、X.Orgが管理を行っています。

Xの特徴は数多くあり、詳細は専門書に譲りますが、「クライアント/サーバシステム」とすることで柔軟なシステム構成となっていること、ネットワーク経由の接続を可能にした「ネットワーク透過性」などが挙げられます。

Question 133



XFree86とX Window Systemはどう違うのですか？

A. 元々XFree86は、X Window Systemをi386系のCPU上で動作するUNIX用に移植したものです。したがって、XFree86は「X Window Systemの一形態」ということができるでしょう。最新バージョンは、X11R6.4をベースにしたXFree86 4.0で、デバイス依存部を動的に組み込む「ロードダブルサーバ」や複数のディスプレイへの描画を可能にした「マルチヘッド対応」などが実現されています。

Question 134



X-TTとはなんですか？

A. X-TTは、TrueTypeフォントのレンダリングエンジンであるFreeTypeを利用して、TrueTypeフォントを扱えるようにしたXサーバのことです。このX-TTを利用すると、ギザギザのないきれいなフォントで表示することができます。

XFree86 4.0では、X-TrueTypeまたは、xfsftのどちらかを組み込めばTrueTypeフォントが使用できます。

Question 135



ウィンドウマネージャとは？

A. ウィンドウマネージャは、ウィンドウの移動、サイズ変更、装飾などを受け持っている特殊なXクライアントのことです。特殊といわれるのは、1つのXサーバ上に1つしか動作しないことや、ユーザー操作に応じてほかのウィンドウの管理を行う点が、ほかのXクライアントと異なるからです。このウィンドウマネージャを変更することによって、さまざまな見栄えや操作感（Look & Feelといいます）を変更することができます。ウィンドウマネージャには、twm、fvwm2、qvwmm、AfterStep、Window Maker、Enlightenmentなど数多くの種類があります。

Question 136



ウィンドウマネージャを変更したいのですが

A. 多くのディストリビューションではswitchdeck-gnomeというツールで変更することができます。TurboLinuxでは、turbowmcfgというツールで変更します。

Question 137



KDEやGNOMEはウィンドウマネージャじゃないの？

A. 結論からいうと、KDEやGNOMEはウィンドウマネージャではありません。というよりも、ウィンドウマネージャを土台に、アプリケーションや各種設定ツールまで提供し、統合的な環境を提供することを目的としたものです。そのため、「統合デスクトップ環境」と呼ばれることもあります。ちなみに、KDEのウィンドウマネージャはkwmであり、GNOMEのウィンドウマネージャはEnlightenmentやSawfish、Window Makerなど複数あります。

Question 138



MotifやQtとかGTK+ってなんですか？

A. 説明の前に、まずXのアプリケーションプログラム（Xクライアント）の仕組みを簡単に説明しましょう。Xクライアントは、図1に示すような構成となっています。最下層にあるのが、基本的な機能をもった「Xlib」です。Xlibはあまりにも基本的な機能しか持っていないため、ボタンひとつ描画するにもとても手間がかかってしまいます。そこで、Xlibの上にプログラマがより使いやすい上位ライブラリが用意されています。これが、「ツールキット」です。

「ツールキット」は、さらにツールキットの核となる機能



を提供する「イントリンシクス」(Intrinsics)と、それにかぶせて「見た目」を提供する部品の集まりである「ウィジェットライブラリ」(Widget Library)の2つに分けることができます。Xでは、Xlibまでを標準機能と定義しており、ツールキットは自由に選択できるようになっています。これは、アプリケーションの見せ方は定義しない、というXの設計思想が反映された結果です。

Motifは、OSF (Open Software Foundation) によって提供されているツールキット (正確にはウィジェットライブラリ) で、有料です。SolarisやHP-UXといった商用UNIXのデスクトップ環境として利用されているCDEは、このMotifをベースに作られています。

GTK+ (GIMP Tool Kit+) は、元々高機能の画像処理ソフトウェア「Gimp」を記述する目的で設計されたツールキットで、Motifに似たLook & Feelを提供しています。GTK+は、LGPLにしたがって配布されているフリーなソフトウェアで、統合デスクトップ環境「GNOME」でも利用されています。

Qtは、ノルウェーの Troll Tech社によって開発されているツールキットで、Windowsともソースレベルの互換性を持つという特徴があります。以前のバージョンのQtは商用で開発を行う際は有料となっていました。Qt-2.0からはQPLという新しいライセンスになり、実質的にフリーなソフトウェアといえるでしょう。Qtは、統合デスクトップ環境「KDE」やOpenLinuxのインストーラ「LIZARD」などで利用されています。



Question 139

リモートマシンからローカルマシンにXアプリケーションを表示させたいのですが

A. まずは、ローカルマシン(local)のXサーバに他のマシン(remote)からの接続を許可する設定をします。

```
local$ xhost remote
```

さらに細かい制御を行いたい場合はxauthコマンドがありますので、詳細はマニュアルを参照してください。次に、リモートマシンに描画させたいマシンの情報を教えます。具体的には、次の手順で環境変数DISPLAYを設定します。

```
remote$ export DISPLAY=local:0.0
```

0.0は、複数のXサーバと複数のスクリーンを特定するための

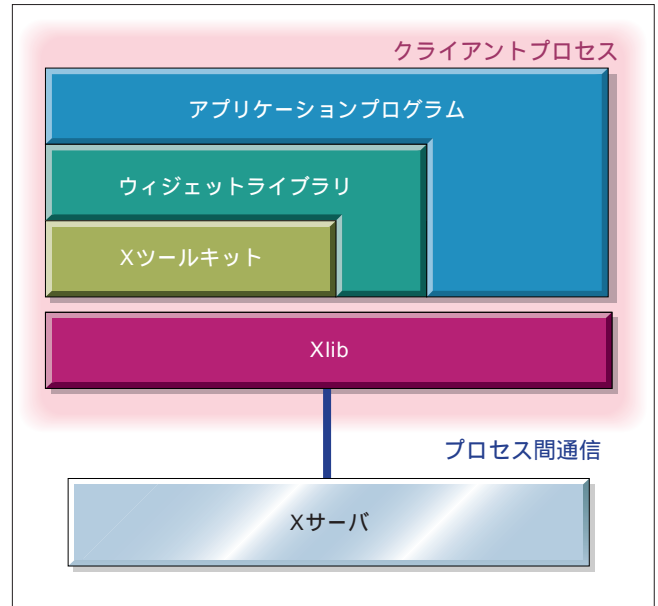


図1 Xクライアントのソフトウェア構成

番号ですが、ほとんどの場合このままで問題はありません。あとは、表示させたいアプリケーションをリモートマシン側で起動させるだけで、ローカルマシンに表示されるようになります。なお、シェルスクリプトxonが利用できれば、環境変数の指定がいらす便利に利用できます。



Question 140

X Window Systemが反応しなくなった

A. Ctrl + Alt + BackSpaceキーを押せば、X Window Systemは強制終了されます。



Question 141

テキストログインをグラフィカルログインに変更したいのですが

A. ランレベルでこのログイン方法が決まっています。(現在のランレベルはrunlevelコマンドでわかります)。ディストリビューションによっても異なりますが、多くのディストリビューションは、テキストログインの場合はランレベルが3になっているはず。そこで、rootユーザーになってから、標準のランレベルを指定してある/etc/inittabをエディタで開いて、

```
id:3:initdefault:
```

という行の「3」を「5」に変更してリポートすると、次回からグラフィカルログインとなるはず。一時的にログイン方法を変更するだけなら、「init 5」と実行します。また、

起動時のLILOプロンプトから、

```
LILO boot: linux 5
```

として、引数にランレベルの5を指定すれば、Xが起動し、グラフィカルログインとなります。

Question 142

グラフィカルログインから抜きたいのですが

A. ランレベルを変更するか、仮想コンソールを利用します (Ctrl+Alt+F1 ~ F6)。ファンクションキーのそれぞれが仮想コンソール1~6に対応しています。Ctrl+Alt+F7がグラフィカル画面です。

Question 143

Xconfiguratorを実行するとブラックアウトしてしまいます

A. Xconfiguratorで設定に失敗して、マシンを起動してもなにも表示されない場合は、別のマシンからtelnetで接続し、そのマシンをリブートします。その後、仮想コンソールなどを使い、コマンドラインベースでX環境を設定できる、xf86configコマンド (XFree 3.3.6以前ではXF86Setupコマンド) を使います。

Question 144

一時的にXの画面解像度を変更したい

A. XFree86では、ドライバとビデオカードとディスプレイが許せば、異なる解像度のモードを複数持つことができます。複数の解像度が登録されていれば、Ctrl + Alt + テンキーの +、- キーで解像度を変更することができます。あれっ、切り替わらないぞ! と思ったら、

リスト4 /etc/X11/XF86Config (2ボタンマウス) の設定

```
Section "Pointer"
...
    Emulate3Buttons          3ボタンエミュレーションを使用
    Emulate3Timeout          80 同時押しの誤差を80ミリ秒
...
EndSection
```

リスト5 /etc/X11/XF86Config (3ボタンマウス) の設定

```
Section "Pointer"
...
    Buttons 3                3番目のボタンを使う
...
EndSection
```

BIOS設定でテンキーが使えなくなっていたなんて失敗もあるので注意しましょう。

Question 145

マウスの真ん中ボタンを有効にしたいのですが?

A. GNOMEやX対応アプリケーションには、マウスの真ん中のボタンでしかできない操作があります。もともとX Window Systemは3ボタンマウスを想定して設計されているため、2ボタンマウスで使う場合は工夫が必要です。

2ボタンマウスで真ん中のボタンを使えるようにするには、3ボタンエミュレーション機能を使いましょう。/etc/X11/XF86Configの「Pointer」セクションに3ボタンエミュレーションを使うように指定します (リスト4)。3ボタンエミュレーション機能では、左右のボタンを同時に押すことで、真ん中ボタンを押したことになります。タイミングが難しいぞなんてビビることはありません。左右のボタンが押されたタイミングの誤差を、ミリ秒単位で指定することである程度のズレを吸収してくれます。

ホイールマウスならホイール自体がボタンとしてクリック可能になっているので、ちゃんとした3ボタンマウスとして使用可能です。ボタンの数を正しく指定すればOKです。この場合は「Pointer」セクションにリスト5のように指定します。

Question 146

Netscape Navigatorでマウスのホイールを使いたい

A. XFree86 3.3.2以上を使っているなら、比較的簡単にホイールマウスを使うことができます。XFree86の設定でホイールの動きをイベントとして割り当てればOKです。もし、手持ちのマウスがMicrosoft社のIntelliMouseでPS/2コネクタ接続ならば、/etc/X11/XF86Configの「Pointer」セクションをリスト6のように編集します。リスト6には、ロジクール社 (米Logitech社) のMouseMan+の設定もコメントにして示していますので参考にしてください。これでボタンイベントの割り振りは、左ボタン=1、真ん中ボタン=2、右ボタン=3、ホイールを上回転=4、下回転=5となります。実は、GNOMEには、このホイールのイベントに対するハンドラが用意されており、gEditやファイルマネージャなどのGNOMEアプリケーションでは、これだけで画面のスクロールにホイールが使えるようになります。

しかし、GimpやNetscape Navigatorなど、X Window Systemジェネリックなアプリケーションは、そのままでは対応できません。そうした未対応アプリケーションでは個別



に設定が必要になります。ここではNetscape Navigatorの設定について紹介しておきましょう。マウスのイベントをNetscape Navigatorの各機能に割り付けます。「/.Xdefaults」の最後にリスト7を追加してください。これでNetscape Navigatorでホイール機能が使えるようになります。

また、他のアプリケーションのホイール機能の割り付けについては、下記のサイトに詳しく解説されているので参考にするといいでしょう。

<http://www-sop.inria.fr/koala/colas/mouse-wheel-scroll/>



Question 147

i810チップセットでXFree86を使うときの注意点は？

A. XFree86で、i810チップセットに統合されているビデオ機能を使おうとしてもそのままではXが起動しません。実は、agpgart.oというモジュールを組み込む必要があるのです。

現在組み込まれているモジュールは/sbin/lsmmodコマンドで確認することができます。もし、agpgart組み込まれていなければ試しに組み込んでみましょう。

リスト6 PS/2接続のIntelliMouseの場合の変更点

```
Section "Pointer"
    ...
    Protocol "imps/2"          「PS/2」を変更する
#   Protocol "MouseManPlusPS/2" MouseMan+の場合

    ZAxisMapping 4 5          ホイールをボタン4、5として拾う
    ...
EndSection
```

```
# /sbin/modprobe agpgart
```

これでstartxコマンドを実行すれば、X Window Systemが起動するはずですが、問題なく起動できれば、システム起動時にモジュールを読み込むように設定しておきましょう。「/etc/conf.modules」にagpgartモジュールを組み込むように指示します。

```
alias char-major-10-175 agpgart
```

ただし、i810チップセットはXFree86 3.3.6以降でのみサポートされています。

リスト7 /.Xdefaultsに追加する内容nfの例

```
!## NETSCAPE
Netscape*drawingArea.translations: #replace \
<Btn1Down>: ArmLink() \n\
<Btn2Down>: ArmLink() \n\
~Shift<Btn1Up>: ActivateLink() \n\
~Shift<Btn2Up>: ActivateLink(new-window) \
                DisarmLink() \n\
Shift<Btn1Up>: ActivateLink(save-only) \
                DisarmLink() \n\
Shift<Btn2Up>: ActivateLink(save-only) \
                DisarmLink() \n\
<Btn1Motion>: DisarmLinkIfMoved() \n\
<Btn2Motion>: DisarmLinkIfMoved() \n\
<Btn3Motion>: DisarmLinkIfMoved() \n\
<Motion>: DescribeLink() \n\
<Btn3Down>: xfeDoPopup() \n\
<Btn3Up>: ActivatePopup() \n\
Ctrl<Btn4Down>: PageUp() \n\
Ctrl<Btn5Down>: PageDown() \n\
Shift<Btn4Down>: LineUp() \n\
Shift<Btn5Down>: LineDown() \n\
None<Btn4Down>: LineUp()LineUp()LineUp()
                LineUp()LineUp()LineUp()
None<Btn5Down>: LineDown()LineDown()LineDown()
                LineDown()LineDown()LineDown()
Alt<Btn4Down>: xfeDoCommand(forward) \n\
Alt<Btn5Down>: xfeDoCommand(back) \n
```

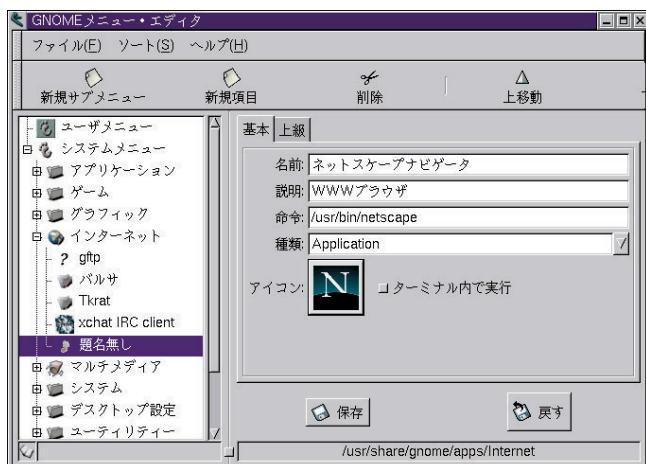
Ctrlキーでページ単位のスクロール
Ctrlキーでページ単位のスクロール
Shiftキーで1行ずつのスクロール
Shiftキーで1行ずつのスクロール
ホイールで6行単位スクロール
ホイールで6行単位スクロール
上と同じ
上と同じ
Altキーと同時に「進む・戻る」
Altキーと同時に「進む・戻る」

Question 148

GNOMEのメインメニューにプログラムを登録するには？

A. 「GNOMEメニュー・エディタ」(以下メニューエディタ)を使って自由に登録削除することができます。メニューエディタは、GNOMEのパネルにある足跡アイコンをクリックしてメニューから[デスクトップ設定]-[メニューエディタ]を選択して起動します。

GNOMEのメニューはシステムメニューとユーザーメニューに分かれています。システムメニューは全ユーザーに共通のメニューで、root権限でないとは編集できません。ユーザーメニューは、ユーザー独自のメニューでユーザーごとに編集することができます。メニューエディタのツールバーから[新規作成]ボタンをクリックします。メインメニューの下に[題名無し]という項目が作成されます。[基本]タブに、メニューに載せる名前、実行したいコマンドのパス名を入力します。アイコン部分をクリックするとアイコンを変更することができます(画面10)。X Window System対応アプリケーションでない場合は、[ターミナル内で実行]オプションをクリックしてオンにしておきます。



画面10 GNOMEメニュー・エディタ



画面12 GNOMEコントロールセンター

Question 149

GNOMEパネルにランチャを登録するには？

A. GNOMEパネルにアプリケーションランチャを登録するにはいくつかの方法があります。もし、登録したいコマンドがシステムメニューやユーザーメニューにすでに登録されているなら、メニューからパネルにドラッグ&ドロップするだけでランチャとして登録することができます。

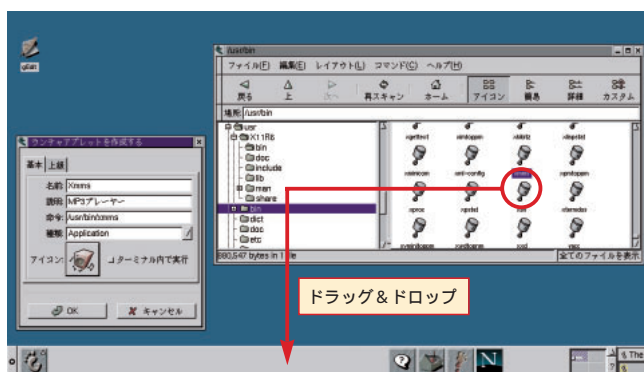
メニューにないアプリケーションをランチャとして登録するには、登録したいプログラムファイルを、ファイルマネージャからパネルヘドラッグ&ドロップします。すると、[ランチャアプレットを作成する]ダイアログボックスが表示されるので、必要事項を入力します(画面11)。

Question 150

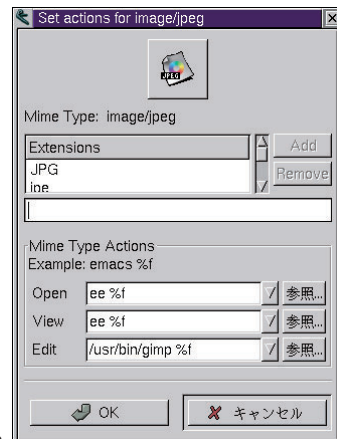
データファイルとアプリケーションの関連付けはどうやるの？

A. データファイルの形式と起動するアプリケーションの関連付けは、自由に変更することができます。

システムメニューの[デスクトップ設定]-[GNOME Control Center]を選択すると、GNOMEコントロールセ



画面11 ドラッグ&ドロップしてランチャに登録



画面13 起動アプリケーションを指定する



ンターが起動します。ウィンドウ内左側のツリーにある「MIME型」をクリックして選択すると、データ型とアプリケーションの関連付けが指定できます(画面12)。

右側に表示されるMIME型と拡張子の一覧から、アプリケーションと関係付けたい項目を選び、[編集]ボタンをクリックします。ダイアログボックスが表示されるので、Open、View、Editの3つのアクションにそれぞれ対応するアプリケーションを指定します(画面13)。ここで指定したアプリケーションは、ファイルマネージャでファイルアイコンを右クリックしたときに表示されるメニューの[開く]、[閲覧]、[編集]を選択すると起動します。

ファイルアイコンをダブルクリックした場合は、Openで指定したアプリケーションが実行されます。

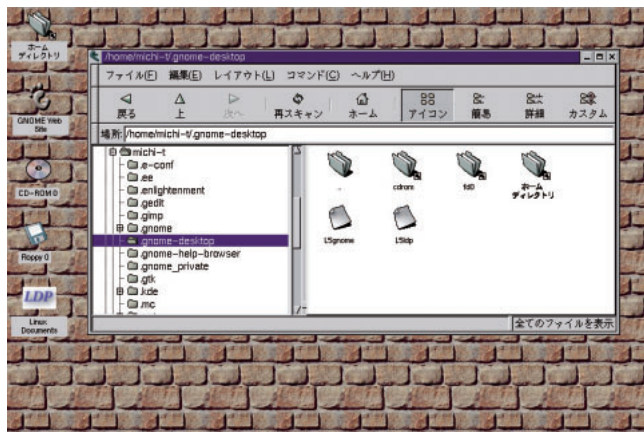


Question 151

GNOMEのデスクトップに置いたファイルはどこに保存されるの？

A. デスクトップに置いたファイルは、「/home/username/.gnome-desktop」というフォルダの下にあります。つまり自分のホームディレクトリの下に隠しディレクトリがあり、その中に保存されているのです。これは通常のディレクトリ表示では見えません。lsコマンドに「-a」オプションを付けて実行するか、GNOMEのファイルマネージャなら[編集]-[設定]を選択し、[設定]ダイアログボックスの[ファイル表示]タブで[隠しファイルを見る]ボタンをオンにします。そうすれば、「/home/username/.gnome-desktop」フォルダが見えるようになり、中身にアクセスできます(画面14)。

このディレクトリにファイルをコピーすれば、デスクトップにファイルをコピーできます。また、このディレクトリのファイルを削除すれば、デスクトップのファイルを削除できます。もちろん、ディレクトリを作ることもできます。



画面14 ~/.gnome-desktopにあるデスクトップのファイル



Question 152

ファイルマネージャのファイルアイコンの意味は？

A. ファイルマネージャに表示されるファイルアイコンはファイルの種類(拡張子やモード)に応じて決められています。主なアイコンは表4の通りです。また、アイコンとファイルの種類の関連付けは、拡張子やファイルのパーミッションによって決められているため、必ずしもこの通りの内容ではありません。たとえば、マニュアルは拡張子が数字であるという前提に関係付けていますが、ログファイルの一部は拡張子が数字になっているため、マニュアルのアイコン表示になっていたりします。



Question 153

デスクトップに壁紙を貼りたい！

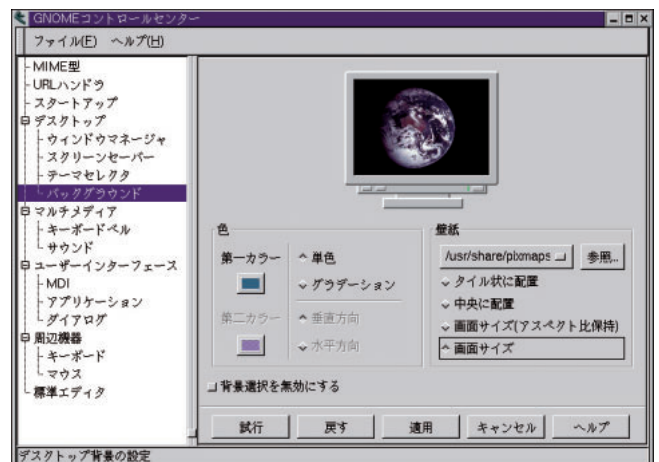
A. GNOMEのデスクトップに壁紙を貼るには、GNOMEコントロールセンターを使います。左のツリーから[デスクトップ]-[バックグラウンド]を選択すると、右側が壁紙を指定する画面に切り替わります(画面15)。JPEG、GIF、PNGなどのデータ形式が貼り付け可能です。



Question 154

デスクトップアイコンのデザインを変更したい

A. デスクトップアイコンを右クリックして、メニューから[プロパティ]を選択します。[プロパティ]ダイアログボックスの[アイコン]タブを表示させ、アイコンボタンをクリックすると、アイコンファイルの選択画面になります(画面16)。また、アイコンを自作することもできます。GIMPなどのツールを使い、オリジナルアイコンを作



画面15 壁紙の設定












アイコン	意味
	ディレクトリ
	ディレクトリ (移動または読み出し禁止)
	コマンド、実行形式ファイル
	テキストファイル (.txt)
	プログラムのソース (.c, .cc, .h)
	Webファイル (.html)
	画像ファイル (データ形式はアイコンにラベルで表示)
	ベクトル画像ファイル (データ形式はアイコンにラベルで表示)
	動画ファイル (データ形式はアイコンにラベルで表示)
	音楽ファイル (データ形式はアイコンにラベルで説明)
	マニュアル (実際には、拡張子が数字のファイル)
	RPM形式のアーカイブ (.rpm)
	アーカイブファイル (.tar, .tgzほか)
	ブロックデバイスファイル (/devフォルダにある)
	キャラクタデバイスファイル (/devフォルダにある)
	コアファイル (異常終了時のメモリの内容、デバッグ時などに使う)
	その他のファイル

表4 主なアイコンと意味 (LASER5)

成し、先ほどのアイコン選択画面の [参照] ボタンでそのファイルを指定します。

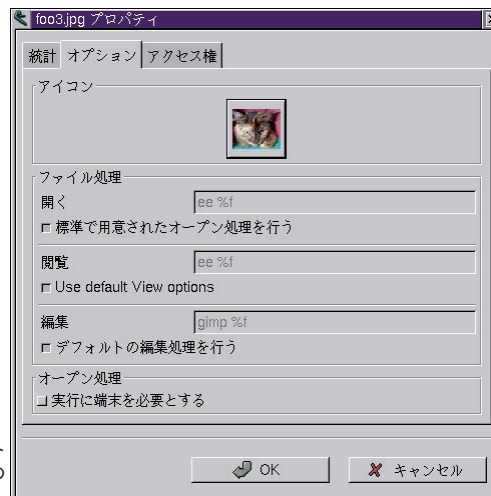
Question 155

ログインパネルから誰でもシャットダウンできるのは困る!

A. 多くのディストリビューションでは、GNOMEのログインパネルにシャットダウンメニューが表示されています。ログインパネルのシャットダウンメニューを表示させないようにするには、/etc/X11/gdm/gdm.confというファイルを書き換えます。

[system] セクションの「ShutdownMenu=1」となっている部分の「1」を削除します。

```
[system]
ShutdownMenu=
```



画面16 デスクトップのアイコンの設定

こうしておけばログインパネルにシャットダウンメニューが表示されなくなります。シャットダウンするには、rootでログインしてGNOMEのメニューでシャットダウンするか、shutdownコマンドを使うことになります。

Question 156

GNOMEでKDE用のユーティリティを使いたい

A. GNOMEと双壁をなすデスクトップ環境にKDEがあります。KDEはGNOMEに比べ、GUI化されたユーティリティが充実しています。GNOMEのままで、KDEのユーティリティが使えたら便利です。GNOMEもKDEも、最近では互いのメニューを参照できるようになっています。もし、両方ともインストールされていれば、GNOMEのメニューの「KDEメニュー」からKDEのツールを利用することができます。

Question 157

プログラム起動時にウィンドウの大きさを指定したい

A. 多くのX Window System対応アプリケーションでは、起動時のウィンドウサイズを指定する場合、「-geometry」オプションが有効です。たとえば、Netscape Navigatorを、800×600ピクセルのウィンドウで起動したければ、コマンドラインから、

```
$ netscape -geometry 800x600
```

と指定します。ただし、「-geometry」オプションはアプリケーションごとに実装されているため、必ずしもすべてのアプリケーションについて、この方法で指定できるわけではあ



りません。また、ktermでは「-geometry」オプションの数値は「1行の文字数」×「行数」で指定します。

なお、ほとんどの場合、サイズに続けてウィンドウの表示位置座標も指定できます。たとえば、ktermを80字×40行の大きさで、デスクトップの左上から右に60ピクセル、下に100ピクセルの位置に表示させたいければ、

```
$ kterm -geometry 80x40+60+100
```

と指定します。「-60-100」と指定すると、デスクトップの右下から左へ60ピクセル上に100ピクセルの位置になります。

**Question 158**

Xのログインパネル(xdm)表示の時のデスクトップに壁紙を貼りたい!

A. ログインパネル表示中のバックグラウンドも変更することができます。X起動時に実行されるプログラムは、/etc/X11/xdm/Xsetup_0というファイルに記述されています。このファイルに、

```
/usr/bin/xsri -scale -keep-aspect ファイル名
```

のように記述します。「-scale」オプションを指定すると、画像が画面いっぱいに表示されるよう拡大縮小されます。このとき、「-keep-aspect」オプションを指定すれば縦横の比率が保たれます。「-geometry」や、「-center-horizontal(左右中央に貼る)」、「-center-vertical(天地中央に貼る)」オプションで、貼り付け位置を指定することもできます。ファイルはJPEG、GIF、PNG形式などが貼り付け可能です。

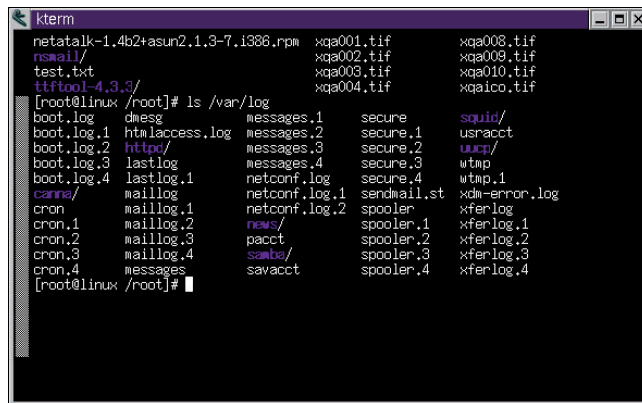
**Question 159**

ktermでスクロールバーを出すには?

A. ktermを使っていて、以前のコマンドの結果をたどって見たいというときにスクロールバーがあれば便利なのですが、ktermを普通に起動してもスクロールバーは表示されません。「-sb」オプションを付けて起動しましょう。

```
$ kterm -sb &
```

と指定します。左ボタンで下へ、右ボタンで上へスクロールし、真ん中ボタンでスライダをつかんで動かすことができます。また、スクロールの範囲はデフォルトでは64行になっていますが、「-sl」オプションで行数を指定することもできます。



画面17 スクロールバーのついたkterm

```
$ kterm -sb -sl 100 &
```

上のように指定すれば、100行分たどってスクロールすることができます。しかし、GNOMEなどでランチャアイコンをクリックして起動する場合、オプションを付けることができません。デフォルトで、スクロールバーを付けるようにできないのでしょうか。それには、「/.Xdefault」ファイルに以下のように記述します。

```
KTerm*scrollBar: True
```

```
KTerm*saveLines: 100
```

こうしておけば、ランチャから起動しても別のktermのコマンドラインから起動しても必ずスクロールバーが付きます。

**Question 160**

Xでスクリーンショットをとりたいのですが

A. これは使用しているディストリビューションやウィンドウマネージャなどにも左右されるので一概にはいえませんが、xv、Gimp、ksnapshotなどで全画面、ウィンドウ単位などで画面をとることが可能です。また、Xに標準添付されているxwdコマンドを利用すれば、コマンドラインからも画面をとることができます。

**Question 161**

WindowsでX Window Systemは使えないのですか?

A. X Window Systemは、クライアント/サーバとして動作するように設計されているため、ネットワークを経由して他のマシンからX用のソフトを使うことができます。Windowsでも、ASTEC-XのようなWindows用のXサーバソフトを使えば、X用のソフトを使うことができます。

ネットワーク



Question 162



TCP/IPってなんですか？

A. コンピュータなどでデータ通信を行う際には、どのような方法で通信を行うかを取り決めておく必要があります。この取り決めをプロトコルといい、TCP/IP (Transmission Control Protocol / Internet Protocol) もそのひとつです。TCP/IPは、1970年代の半ばにアメリカ国防総省によって作られたARPANETというネットワークのために開発されました。ARPANETはインターネットの原型となったネットワークです。そのため、インターネットでもTCP/IPを使用しています。

TCP/IPには、特定のハードウェアベンダーに依存しないという特色があり、インターネットの普及ともあいまって、LinuxやUNIXはもちろん、WindowsやMacintoshでも標準的に使われるプロトコルになっています。

Question 163



ネットワークカードが認識されない

A. まず、ドライバがちゃんと組み込まれているのか、ifconfigコマンドで確認しましょう。正常に動作している場合は画面18のようになります。しかし「eth0」の行から7行が表示されずに、「lo」の行から始まっている場合には、ネットワーク機能は正常に動作していません。

PCIバスの10BASE-T / 100BASE-TX対応ネットワークカードの場合、普通はインストーラの自動検出によって認識されますが、もしも自動認識されなかった場合には、手動での設定が必要になります。

ディストリビューションをインストールした直後のカーネルでは、ハードウェアを自動的に検出し、必要なドライバをモジュールとして組み込んでいます。手動でモジュールを読み込むには、

```
# modprobe <ドライバ名>
```

という書式で、ドライバを設定します。I/Oアドレスや10BASE-T / 100BASE-TXの切り替えの指示は、ドライバ名のあとに「eth0= <パラメータ>」として指定します。

また、ISAバスのネットワークカードでは、割り込みやDMAアドレスがほかのカードと衝突している可能性があります。その場合にはpnpdumpコマンドで状況を調べ、isapnpコマンドでカードの使用アドレスを変更して、衝突を回避する必要があります。

Question 164



インストール時のIPアドレスの設定は、手動設定、BOOTP、DHCPのどれを選べばいいの？

A. BOOTP、DHCPは、ともにIPアドレスを動的に割り当てる仕組みのことで、BOOTPサーバやDHCPサーバがある環境なら、IPアドレスの設定でBOOTP、DHCPを選ぶと、起動時にBOOTP、DHCPサーバから設定すべきIPアドレスを受け取って自動的に設定します。

DHCPサーバがない場合は、IPアドレスを直接指定します。専用線などで常時インターネットに接続する場合を除き、あとで説明するプライベートアドレスの中から適当な値を指定すればよいでしょう。

Question 165



pingを実行しても相手のホストから返事がありません

A. まず自分のハードウェアがきちんと動作しているのか、ifconfigコマンドで確認します(画面18)。特に2行目に注目しましょう。「inet addr:」がIPアドレス、「Bcast:」がブロードキャスト、「Mask:」がネットマスクをそれぞれ表しています。

次に、どこに原因があるのかを調べるため、近くのネット

```
# /sbin/ifconfig
eth0 Link encap:Ethernet HWaddr 00:48:E8:EB:92:44
      inet addr:10.0.0.9 Bcast:10.255.255.255 Mask:255.0.0.0
      UP BROADCAST RUNNING MTU:1500 Metric:1
      RX packets:313 errors:0 dropped:0 overruns:0 frame:0
      TX packets:95 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xd800

lo   Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:3924 Metric:1
      RX packets:23 errors:0 dropped:0 overruns:0 frame:0
      TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
```

画面18 /sbin/ifconfigでネットワークカードの状態を見る



ワークから順番にpingコマンドで試験していきます。最初はホスト名を指定せずにIPアドレスを直接入力します。

- (1) 同一セグメント内のホスト
- (2) ゲートウェイ
- (3) 別セグメントのホスト

IPアドレスでpingコマンドを実行してみて返事があるようならば、次にホスト名を指定しましょう。ホスト名を指定するとダメなようならDNSの名前解決ができていないと考えられますので、ネームサーバに問い合わせるコマンドで確認します。nslookup <ホスト名> を実行して正しいIPアドレスに設定されているか調べます (画面19)。



Question 166

「unkonwn host」となってしまうのですが

A. ホスト名からIPアドレスへの変換ができていない可能性があります。IPアドレスがわかっている場合には、IPアドレスを直接指定してみるとうまくいくことがあります。

```
# nslookup <ホスト名>
```

と入力してみてください (画面19)。

最初のServerが正しく表示されない場合には、DNSサーバに接続できていません。/etc/resolv.confファイルの「nameserver」行にDNSサーバが登録されていますので、内容を確認してください。

「can't find <ホスト名>: Non-existent host/domain」と表示された場合は、そのホスト名はDNSに登録されていません。/etc/hostsファイルに、ホスト名とIPアドレスを登録すれば解決します。

nslookupコマンドでは、ホスト名の代わりにIPアドレスを指定すると、DNSの逆引きが行われ、ホスト名が表示されます。

```
# nslookup www.ascii.co.jp
Server: <DNSサーバ名>
Address: <DNSサーバのIPアドレス>

Name: at2.ascii.co.jp
Address: 210.140.231.23
Aliases: www.ascii.co.jp
```

画面19 nslookupでDNSサービスの動作を調べる



Question 167

ホスト名はなんでもいいの？

A. ローカルネットワークの中で重複していなければ自由な名前を付けて構いません。ただし、そのホストをインターネットにつなぐ場合は、英文字以外を使うと問題が起こる可能性があります。また、大文字と小文字が区別されることを期待してはいけません。インターネットには、さまざまな種類のシステムが接続されているため、英文字以外の記号が扱えなかったり、大文字と小文字の区別ができない環境があるかもしれないからです。

また、利用者のことを考えて、あまり長い名前や綴りが複雑すぎる名前も避けましょう。品のない名前もお勧めできません。以上のようなホスト名の付け方のガイドラインがRFC 1178として示されています。



Question 168

IPアドレスは自由につけてもいいの？

A. インターネットでは、機器ごとに世界唯一のIPアドレスを使う必要があり、プロバイダから指定されたIPアドレスだけが使用を許されます。しかし、現在主に使われているIPバージョン4ではおよそ43億個のアドレスしかありませんので、世界中の機器すべてにユニークなアドレスを付けることはできません。そこで、インターネットにつながない閉じたネットワークで自由に利用できるプライベートアドレスが用意されています。家庭内LANなどでは次に示すプライベートアドレスの中から適当なクラスを選び、その範囲内のアドレスを使いましょう。

クラスA 10.0.0.0 ~ 10.255.255.255

クラスB 176.16.0.0 ~ 172.31.255.255

クラスC 192.168.0.0 ~ 192.168.255.255



Question 169

MACアドレスってなんですか？

A. MACアドレス (Media Access Controlアドレス) とは、イーサネット機器に付けられている48ビット長のアドレスです。これは、製品のベンダーが出荷時に付けるもので、1台1台の機器に固有のもので、イーサネットでは、通信相手の特定にはこのMACアドレスが使われます。イーサネットでTCP/IPを利用する際には、相手のIPアドレスがわかっても、MACアドレスがわからなければ通信がで

きません。このため、ARP (Address Resolution Protocol) によってIPアドレスとMACアドレスの対応を取っています。

Linuxで、イーサネットカードのMACアドレスを調べるにはifconfigコマンドを実行します。

Question 170

ゲートウェイアドレスってなんですか？

A. ゲートウェイとは、ネットワーク間でデータを転送するシステムです。ローカルネットワーク (同一セグメント) につながれた2台のマシンは直接通信できます。しかし、インターネットのようにさまざまなネットワークがつながれている先にあるマシンと通信する場合は直接通信することができないので、ゲートウェイにデータを渡して転送してもらうのです。このゲートウェイのIPアドレスが「ゲートウェイアドレス」です。ゲートウェイはルータだったりゲートウェイ機能を持ったコンピュータだったりします。

データを受け取ったゲートウェイが送り先のマシンに直接つながっていなければ、送り先により近い別のゲートウェイに渡します。こうして順次データが転送されて目的のマシンに到達するわけです (図2)。

では、データを渡すべきゲートウェイはどのように選ばれるのでしょうか？ TCP/IPで通信するコンピュータやルータは、送り先のIPアドレスに応じて適切なゲートウェイを指定するルーティング情報を持っています。この情報を参照してデータの送信先を決めるのです。ルーティング情報はrouteコマンドを使って静的に設定するか、RIP (Routing Information Protocol) で動的に設定します。

家庭内LANなど、小規模なLANでは特別なルーティング情報を設定する必要はなく、デフォルト (既定の) ゲートウェイだけを設定すれば十分でしょう。ISDNルータやIPマス

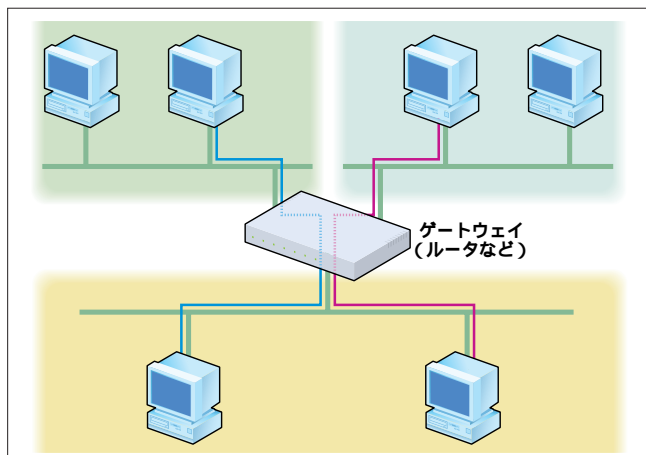


図2 ゲートウェイはネットワーク間をつなぐシステムだ

カレードを行うLinuxサーバを介してインターネットに接続している場合は、そのIPアドレスをデフォルトゲートウェイアドレスとして設定します。

設定されているルーティング情報は、netstat -rを実行することで見ることができます。

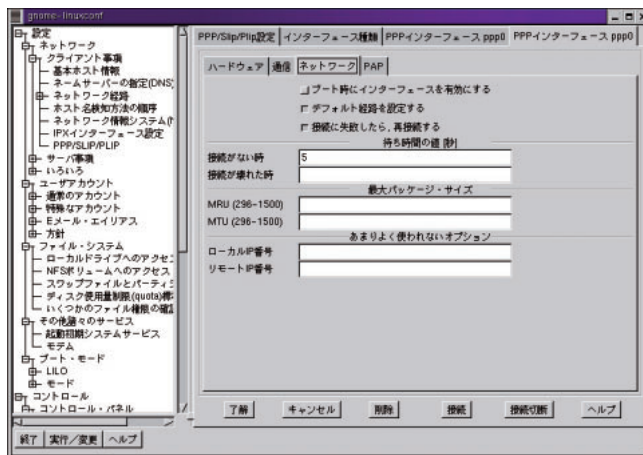
Question 171

サブネットマスクってなんですか？

A. 32ビットの長さを持つIPアドレスは、ネットワーク部とホスト部という2つの部分に分けることができます。通常のアドレスとして使われるIPアドレスは、ネットワーク部が8ビットのクラスA、16ビットのクラスB、24ビットのクラスCが定義されており、それぞれ1677万7214台、6万5534台、254台のマシンを同一セグメントに接続できます。

従来は、このクラスに従ってネットワークセグメントを分けていたのですが、最小のクラスC単位でIPアドレスを割り当てても256個のIPアドレスを消費します。10台ずつの小さなセグメントを10個作り、それぞれのセグメントを相互につないで運用したいときなどは、マシンが100台しかなくても2560個ものIPアドレスを用意しなくてはなりません。これでは無駄も多く、容易にセグメントの分割をすることができないので、クラスよりも柔軟にネットワーク部とホスト部に分けられる「サブネット」という概念が導入されました。

サブネットとは、クラス別のネットワークアドレスに加え、ホスト部のアドレスもネットワーク部の一部とみなして、1つのセグメントに接続できるマシンの台数を調整するものです。IPアドレスのうち、どの部分がネットワーク部でどの部分がホスト部であるかを示すのがサブネットマスクです。たとえば、IPアドレスの25ビット目までをネットワーク部とするなら、サブネットマスクは255.255.255.128となります。こ



画面20 LinuxconfでPPPの設定を行う



のサブネットには126台のマシンを接続することができます。サブネットによって柔軟なセグメント分割ができるのです。

家庭内LANなど、小規模で閉じたネットワークに接続する場合はクラスC（サブネットマスクは255.255.255.0）のプライベートアドレス（192.168.1.0～192.168.1.255）を使えばよいでしょう。

Question 172



ISDNルータを使う設定は？

A. Linux側の設定は、LANに接続する時と特に変わりはありません。デフォルトゲートウェイとしてISDNルータを指定するだけです。ただし不要な発呼を行わないように、マニュアルを参照してISDNルータを設定しましょう。

Question 173



モデム/TAを使う設定は？

A. 電話回線を通じてインターネットサービスプロバイダに接続するには、PPP（ポイント・ツー・ポイントプロトコル）を使います。LinuxではPPP機能を提供するプログラムとして、pppdまたはPPoPがよく使われています。pppdの設定はLinuxconfで簡単に行えるようになっています（画面20）。

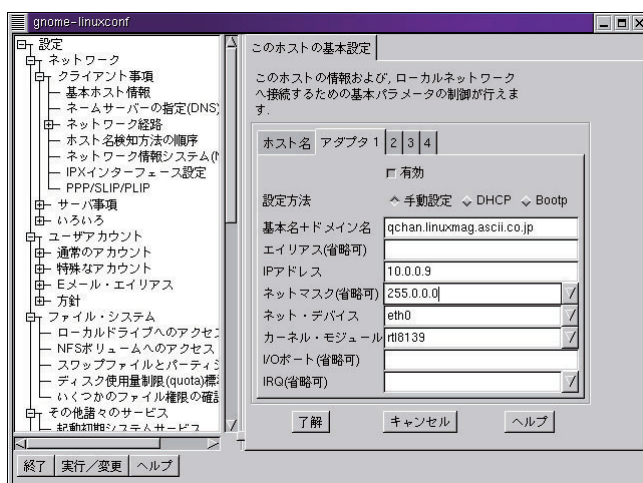
なお、PPP接続のための設定ファイルは、/etc/pppディレクトリ以下にあります。

Question 174



IPアドレスやホスト名を変更したいが、どこを変えればいい？

A. Red Hat系のディストリビューションなら、Linux confユーティリティを使うのが簡単です（画面21）。



画面21 Linuxconfでネットワークの設定を変更する

TurboLinuxなら、turbonetcfgユーティリティを使いましょう。

設定ファイルをエディタで変更するなら、ホスト名は/etc/sysconfig/networkファイルの「HOSTNAME =」を変更します（リスト8）。IPアドレスは/etc/sysconfig/network-scripts/ifcfg-eth0の「IPADDR =」を変更します（リスト9）。そのほかのネットワークパラメータも同様に変更できます。

設定を変更したら、

```
# /sbin/ifdown eth0    (停止)
# /sbin/ifup eth0      (起動)
```

として、ネットワークカードを停止/再起動してください。また、

```
# /etc/rc.d/init.d/network restart
```

を実行してネットワークを再起動することもできます。

Question 175



IPマスカレードとは？

A. 最近のISDNルータには標準的についでいる機能で、内部ネットワークのIPアドレスと外部ネットワークのIPアドレスを相互に変換する機能です。NAT（Network Address Transration）とも呼ばれます。この機能によって、インターネットに接続するための回線やIPアドレスが1つしかなくても、内部のネットワークにあるマシンで共有し、同時にインターネットが利用できるようになります。

内部/外部両方のネットワークに接続されたLinuxマシンでIPマスカレードを動作させれば、同様の機能を持たせるこ

リスト8 /etc/sysconfig/networkの例

```
NETWORKING=yes
FORWARD_IPV4=false
HOSTNAME=qchan.linuxmag.ascii.co.jp
GATEWAY=10.0.0.1
GATEWAYDEV=eth0
```

リスト9 /etc/sysconfig/network-scripts/ifcfg-eth0の例

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
IPADDR=10.0.0.9
NETMASK=255.255.255.0
```

とができます。このLinuxマシンは、内部のネットワークから出されるパケットのアドレスを外部のIPアドレスに置き換えて外部に中継し、外部からのパケットには逆の置き換えを行って内部に中継します。そのため、外部から見えるのはIPマスカレードを導入した1台だけになり、セキュリティを高めることが可能です。外部との通信は、ISDNや電話回線を使う場合と、LANや専用線を使う場合の両方があります。

Linux IP Masquerade mini HOW TO (<http://www.linux.or.jp/JF/JFdocs/IP-Masquerade.html>) が参考になるでしょう。



Question 176

ネットワークカードを2枚挿していますが、1枚しか認識されません

A. ネットワークカードのドライバは、モジュールとして組み込む場合と、カーネルに組み込む場合の2通りがあります。モジュール組み込みの時には、`/etc/conf.modules`ファイルに1枚目 (`eth0`) のカードの設定が「`alias eth0 <モジュール名>`」のように書いてあるはずですが。

テストのために、2枚目のカードを、

```
# modprobe eth1 <モジュール名>
```

と入力して手で組み込んでみましょう。正しくロードされたかは、`lsmod`コマンドでモジュール名が表示されたことで確認できます。この時のモジュール名は、2枚目のネットワークカードに対応するドライバ名を指定します。

次に、`linuxconf`プログラムなどで`eth1`のネットワークドライバを設定し、`ifconfig`コマンドで、`eth0`と`eth1`の設定を確認します。うまく動作するようなら、2枚目のカードの設定を`/etc/conf.modules`ファイルに、

```
alias eth1 <モジュール名>
```

のように書き加えます。

カーネルに組み込む場合は、`/usr/src/linux`ディレクトリで、`make xconfig` (コンソール用は`make menuconfig`) を実行して、使用しているネットワークカードの項目を「Y」に設定し、カーネルを再構築します。モジュールではないので、`/etc/conf.modules`ファイルの変更は不要です。

使っているネットワークカードのドライバ名を調べるには、Linux Ethernet-Howto (<http://www.linux.or.jp/JF/JFdocs/Ethernet-HOWTO.html>) を読むとよいでしょう。



Question 177

2台のマシンでDNSサーバは必要ですか？

A. DNS (Domain Name System) サーバとは、ホスト名とIPアドレスの対応を相互に変換 (名前解決) するサーバです。インターネットに接続されたDNSは分散データベースシステムとして動作しており、DNSサーバ同士で通信することで数多くあるホストの名前を解決しています。

数台からなる小規模のLANでDNSサーバを使うことも可能ですが、DNSを利用せずに`/etc/hosts`ファイルを使うほうが簡単です。`/etc/hosts`ファイルには、ホスト名とIPアドレスの対応を1行に1組ずつ記述します (リスト10)。

次に、`/etc/host.conf`ファイルを参照して、

```
order hosts,bind
```

と書かれていることを確認します。これは、名前解決の際に、まず`/etc/hosts`ファイルを参照し、解決できないときは`bind` (DNS) を参照することを意味します。



Question 178

Windowsマシンとファイル/プリンタを共有したい

A. Sambaというユーティリティを使うと、LinuxマシンをWindowsマシンのファイルサーバや、プリンタサーバとして使うことができます。Red Hat系のLinuxでは、Sambaの設定ファイルは、`/etc/smb.conf`です。初期設定ではほとんど「;」でコメントアウトされていますので必要な部分を修正します「[]」で囲まれた名前が共有名となっています。プリンタは[`printers`]セクションで設定します。

なお、Sambaに接続する時にユーザー認証が行われるため、あらかじめパスワードを設定しておく必要があります。パスワードファイルはLinuxで使われる`/etc/passwd`ではなく、`/etc/smbpasswd`ファイルを使います。`/etc/passwd`から`/etc/smbpasswd`ファイルを作成するスクリプトが用意されているので、次の手順で設定しましょう。

リスト10 /etc/hostsの例

```
127.0.0.1    localhost.localdomain localhost
10.0.0.1    lmgate.linuxmag.ascii.co.jp lmgate
10.0.0.9    qchan.linuxmag.ascii.co.jp qchan
10.0.0.10   pchan.linuxmag.ascii.co.jp pchan
```



Linuxレスキュー 『疑問・難問・200問』

```
# cat /etc/passwd | mksmbpasswd.sh > /etc/smbpasswd
# chmod 600 /etc/smbpasswd
```

chmodコマンドで/etc/smbpasswdファイルをroot以外がアクセスできないようにしておきます。

なお、smb.confを変更したあとに、

```
# /etc/rc.d/init.d/smb restart
```

としてSambaを再起動すると、設定変更が反映されます。

Samba 2.xから、SWATというWebブラウザベースで作る、Samba設定ツールがあります。これを使うとSambaの各種設定が楽に行えます (画面22)。

Sambaについての情報は、日本Sambaユーザ会のWebページ (<http://www.samba.gr.jp/>) をご覧ください。



Question 179

Macintoshとファイル/プリンタを共有したい

A. netatalkというユーティリティプログラムを使うと、LinuxマシンをMacintoshのファイルサーバとして使えるようになります。netatalkについての情報は、netatalkのWebページ (<http://www.umich.edu/~rsug/netatalk/>) をご覧ください。



Question 180

Webサーバを起動したい

A. ほとんどのLinuxディストリビューションには、WebサーバソフトApacheが含まれています。“ps

ax”でプロセスの実行状態を見て、httpdという行が何行も表示されるようなら、すでにApacheは起動されています。

設定ファイルは/etc/httpd/conf (あるいは/usr/local/apache/conf) ディレクトリに、そしてHTMLファイルは、/home/httpd/html (あるいは/usr/local/apache/htdocs) ディレクトリにあります。試しに、Netscapeを起動してみましょう。URLに、localhostと入力することでローカルマシンにアクセスしてみましょう (画面23)。



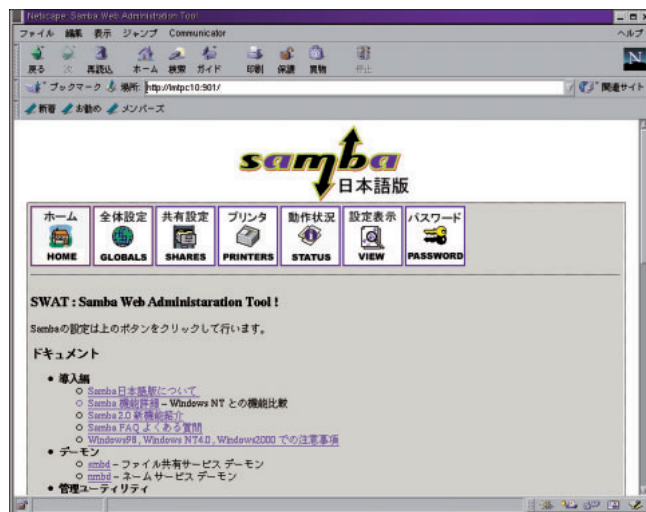
Question 181

rootでtelnetやFTPログインができない

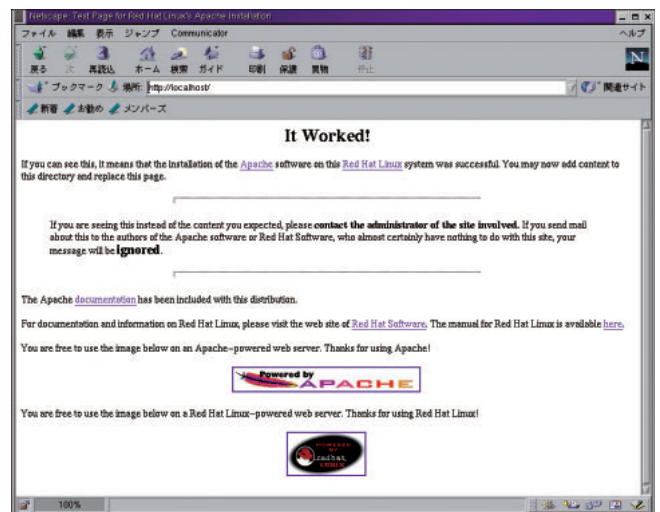
A. telnetやFTPでは、ユーザー認証の際にパスワードをテキスト (平文) のままで回線に流すため、通常はrootユーザーのtelnetやFTPは禁止されています。

どうしてもというのであれば、/etc/ftpusersファイルからrootと書かれた行を消すことでFTPログインが可能になります。また、/etc/securettyファイルを削除すればtelnetも可能になります。しかし、セキュリティが極めて脆弱になりますので、できる限りSSH (Secure Shell) を使いましょう。SSHは、ユーザー認証と通信内容を暗号化します。これにより、リモートログインや、ホスト間でのファイルコピーを安全に行えます。ただし、商用利用する場合は有償のライセンスを受ける必要があります。

詳しくは、SSHのWebページ (<http://www.ssh.org/>) を参照してください。また、SSHをフリーで実装するOpenSSHプロジェクト (<http://www.openssh.com/>) もあります。



画面22 SWATを使えばWebブラウザでSambaの設定ができる



画面23 自ホスト (<http://localhost/>) にアクセスしてApacheの動作を確認

miscellaneous (その他もろもろ)



Question 182

新しくハードディスクを追加したのでチェックをかけたかと思っています。Linuxではどうすればいいですか？

A. 新しいハードディスクにファイルシステムを作成したときに、念のためチェックしておくのは好ましいことです。このチェックにはfsckコマンドを使います。

```
# fsck -aC /dev/パーティション
```

と実行すると、チェックが開始され進行状況が表示されます。エラーがあった場合は自動的に修復されます。

ルート(/)ファイルシステム以外は、umountコマンドでマウントを解除してfsckを実行します。ルートファイルシステムにfsckを実行するときは、シングルユーザーモードで行うようにしてください。



Question 183

起動時にエラーが出ているのですが、速すぎて読み取れません

A. 起動後のログインプロンプトの状態、Shift + PageUPキーを押すと画面をスクロールアップすることができます。Shift + PageDownキーでスクロールダウンも可能です。ほかのキーを押すとログインプロンプトに復帰

```
md.c: sizeof(md_p_super_t) = 4096
Partition check:
 hda: hda1 hda2 < hda5 >
 autodetecting RAID arrays
 autorun ...
... autorun DONE.
UFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 64k freed
INIT: version 2.78 booting
Welcome to LASEM5 Linux
Press 'I' to enter interactive startup.
Mounting proc filesystem [ OK ]
Configuring kernel parameters [ OK ]
Setting clock : Mon Jun 26 17:29:13 JST 2000 [ OK ]
Loading default keymap [ OK ]
Activating swap partitions [ OK ]
Setting hostname hoge [ OK ]
Checking root filesystem
/dev/hda1: clean, 21477/76960 files, 56730/153712 blocks [ OK ]
Remounting root filesystem in read-write mode [ OK ]
Finding module dependencies [ OK ]
Checking filesystems [ OK ]
```

画面24 Shift + PageUp/Downを使えば起動時のメッセージをゆっくり見られる

できますが、Deleteキー（バックスペース操作が割り当てられたキー）で戻るのが余計なキーシーケンスが表示されないのでスマートです。

起動後のデフォルトがグラフィカルログインになっている場合は、ブート時の画面情報がクリアされてしまうためShift + PageUpによる操作はできません。デフォルトのランレベルをテキストログインに変更して再起動してください。少し乱暴な方法ですが、起動メッセージが表示されている際に、Ctrl + Sでスクロールをストップさせることも可能です（再開はCtrl + Q）。ストップした状態で、Shift + PageUpキーを使ってスクロールアップします。



Question 184

コンソールの状態でも複数の画面が使えるってホント？

A. Linuxには仮想コンソールという機能があります。コンソール画面で、Alt + ファンクションキーを押すと、それぞれ別のコンソール画面に切り替えることができます。ディストリビューションによって多少異なりますが、普通はF1 ~ F6までの6画面が使えます。

仮想コンソールは、X Window System環境からも利用できます。この場合、Ctrl + Alt + ファンクションキーで画面を切り替えます。ほとんどのディストリビューションでは、Alt + F7でX Window Systemの画面に戻ります。



Question 185

コマンドを実行しても“Permission denied”と表示されて何も起こりません。どうして？

A. Linux (UNIX) では「パーミッション」という属性を使ってファイルやディレクトリに対する操作を制限しています。表示されたメッセージは「そのユーザーに許可されていない操作を行おうとした」ことを表しています。

コマンドを実行できない場合のもっとも一般的な解決法

```
kterm [Linux magazine]$mkdir /etc/MyFavorite
mkdir: cannot make directory '/etc/MyFavorite': Permission denied
[Linux magazine]$su -
Password:
[Linux magazine]#mkdir /etc/MyFavorite
[Linux magazine]#ls -ld /etc/MyFavorite/
drwxr-xr-x 2 root root 4096 Jun 27 09:33 /etc/MyFavorite/
[Linux magazine]#
```

画面25 suの使用例



は、suコマンドを使った方法です。引数を付けずにsuコマンドを実行するとrootユーザーとしての権限が与えられます（rootユーザーのパスワードを知っていることが前提となります）。rootユーザーは、すべてのコマンドに対して実行パーミッションを持っていますから、コマンドを実行できるようになるはずですが。

Question 186

カレントディレクトリにあるプログラムが実行できません。デフォルトではカレントにパスが通らないのですか？

A. LinuxやUNIXではDOSとは違い、カレントディレクトリにパスを通さないのが普通です。これはセキュリティを確保するためです。

たとえば、悪意を持った人がトロイの木馬やクラックプログラムを/rootにlsという名前で置いたとします。このとき、rootユーザーのPATH変数の先頭にカレントディレクトリが含まれていたらどうなるでしょう。rootユーザーがログインして、ホームディレクトリ(/root)でlsを実行すると、本当のlsではなく、悪者が置いた悪のプログラムが実行されてしまいます。

このようなことが起こらないよう、カレントディレクトリにパスを通してはいけません。カレントディレクトリにあるプログラムが安全であることが確認できていれば、/(ルート)ディレクトリからの絶対パスでプログラムを指定するか、

```
# ./hoge
```

のように、.(カレントディレクトリ)からの相対パスを指定してプログラムを実行します。

また、/usr/local/binなどに一般のユーザーが実行ファイルを置けるようにしているなら、rootユーザーのPATH変数に設定するディレクトリの順序にも気をつけてください。できれば、これらのディレクトリにはパスを通さないほうが安全です。

Question 187

coreという名前の巨大なファイルが知らない間にできてしまいました。削除しても大丈夫でしょうか？

A. coreファイルは、プログラムが正しく終了できなかった際に、そのときのメモリの状態を記録しておくために作成される特殊なファイルです。内容を解析し、エラー時のシステムの状態を知ることで、プログラムの修正などに利用します。一般のユーザーは内容を見ても理解できませんから、ディスク容量が不足している場合などには、削除し

てしまってもかまいません。

coreファイルの最大サイズを0に制限して、作成されないように設定することもできます。bashの場合は、“ulimit -c 0”コマンドをユーザーのホームディレクトリにある.bashrcファイルに記述しておけば、coreファイルは作成されません。

Question 188

バックアップとして、-hostsというファイルを作成したのですが、消せなくなって困っています

A. このようにハイフン(-)で始まる名前のファイルを削除しようとして、普通にrmコマンドを実行しても、rmコマンドがファイル名をオプションと勘違いしてエラーになってしまいます。

```
# rm -hosts
rm: invalid option -- 1
```

ファイル名の前に./をつけて相対パスでファイル名を指定すれば、削除することができます。

```
# rm ./-hosts
```

また、ハイフンを重ねると、それ以降にオプションはないという意味になり、“-hosts”がファイル名と解釈されます。

```
# rm -- -hosts
```

名前にスペースが含まれているファイルについても触れておきましょう。この場合、ダブルクォーテーションまたはシングルクォーテーションマークでファイル名を囲めば削除できます。

```
# rm "blank file"
```

Question 189

?マークの入った、得体の知れないファイルがあります。これも削除できますか？

A. Windowsなどから日本語の名前のついたファイルでFTPでLinuxマシンに転送すると、シフトJISが表示できないLinux上では化けてしまいます。たとえば、“業務報告”というファイルは“????????”というように見えます。ktermなど日本語の表示できるターミナル上で、

```
# ls | nkf -e
```

として、漢字コードをEUCに変換すれば本当のファイル名を見ることができます。そこで、この情報とワイルドカードを組み合わせて、慎重に消すことになります。“業務報告”は8バイトですので、1バイトのキャラクタ1個にマッチするワイルドカード“?”を8個並べれば消すことができます。同じディレクトリに8バイトの長さの名前を持つファイルがあると、すべてマッチしてしまうので、rmコマンドには必ず-iオプションをつけて対話形式で使いましょう。

```
# rm -i ??????????
```

また、削除したいファイルが固有の拡張子をもっていれば、ワイルドカードを利用することもできます。たとえば“?y?[?W??.xls”などと化けてしまった“ページ数.xls”というファイルがあって、同じディレクトリにxlsという拡張子を持つ別のファイルがなければ、

```
# rm -i *.xls
```

として消すことが可能です。念のため、rmには-iオプションをつけておきましょう。



Question 190

アスタリスクを入力したはずなのに、ダブルクォーテーションが入力されてしまいます

A. 日本語キーボードを使っているのに、Linuxの設定が英語キーボードになっているのでしょう。

キーボードの設定は、コンソール環境とX Window Systemで別々に行う必要があります。コンソール環境では、loadkeysコマンドに、引数としてキーマップファイルを指定することでキー配置を設定しています。キーマップファイルには、キーボードごとのキー配置が記述されています。

LASER5 Linux 6.0やTurboLinux 6.0では、/etc/sysconfig/keyboardというファイルで、キーマップファイルを指定します。たとえば、このファイルに、“KEYTABLE="jp106"”と記述すると、次回起動時から日本語106キーボードのキー配置になります。また、Slackware 7.0では、/etc/rc.d/rc.keymapのなかでキーマップファイルを指定しています。

どのようなキーマップファイルがあるのかは、キーマップファイルのある/usr/lib/kbd/keymaps/i386/qwerty/ディ

レクトリを見ればわかります。

X Window Systemでは、/etc/X11/XF86Configというファイルの中で指定します。この中の“Section keyboard”で“XkbModel "jp106"”と、“XkbLayout "jp"”を記述すれば、日本語106キーボードの設定になります。



Question 191

Caps Lockキーと左Ctrlキーの機能を入れ替えたい

A. ひとつ前の質問と同様に、コンソール環境とXの環境を別々に設定する必要があります。ここでは、TurboLinux 6.0とLASER5 Linuxを例に説明します。

コンソール環境では、jp106などのキーマップファイルをもとにして、Caps Lockキーと左Ctrlキーを入れ替えたキーマップファイルを作り、これを使うように設定をします。

作業は次のようになります。まず、キーマップファイルのあるディレクトリをカレントにします。

```
# cd /usr/lib/kbd/keymaps/i386/qwerty
```

もともとなるjp106.map.gzをjp106_CC.map.gzにコピーします(LASER5 Linuxでは、以下“map”を“kmap”に読み替えてください)。

```
# cp jp106.map.gz jp106_CC.map.gz
```

gzipで圧縮されたjp106_CC.map.gzを展開します

```
# gunzip jp106_CC.map.gz
```

展開されたjp106_CC.mapをエディタで開き、次の2カ所を書き換えます。

```
keycode 29 = Control          Caps_Lock
keycode 58 = Caps_Lock       Control
```

エディタを終了して、gzipで圧縮します。

```
# gzip -9 jp106_CC.map
```

新しいキーマップファイルができましたので、/etc/sysconfig/keyboardで、キーマップファイルの指定を“jp106”から“jp106_CC”に変更します。変更したら、キーマップ



ファイルの変更をシステムに反映させます。

```
# /etc/rc.d/init.d/keytable restart
```

これで、コンソール環境ではCaps Lockキーと左Ctrlキーの機能が入れ替わりました。

X環境での設定は比較的簡単です。エディタで、/etc/X11/XF86Configを開き、“Section "Keyboard"”の中に、“XkbOptions "ctrl:swapcaps"”を追加します。このあと、Xサーバを再起動すれば、Caps Lockキーと左Ctrlキーの機能が入れ替わります。



Question 192

システムを起動するたびに時刻が狂ってしまいます。そのたびにdateコマンドで合わせているのに.....なぜ？

A. システムの時刻には、ハードウェアクロックによるものと、オペレーティングシステムが使用するものの2種類があります。Linuxは起動時にハードウェアクロックから日付と時刻に関する情報を取得し、これをシステムクロックとして以後の処理で使用します。

dateコマンドはシステムクロックを設定するコマンドです。dateでいくら時刻を合わせても、ハードウェアクロックが正しい時刻に設定されていないければ、起動のたびにシステムクロックも狂ってしまうのです。

ハードウェアクロックを操作するコマンドはhwclockです。オプションなしで実行すると、ハードウェアクロックの現在の時刻が表示されます。時刻を設定するコマンドラインは次のとおりです。

```
# hwclock --set --date="MM/DD/YY hh:mm:ss"
```

dateコマンドの使い方も確認しておきましょう。日付と時刻を設定する場合のコマンドラインを示します。

```
# date MMDDhhmm
```

hwclockかdateのどちらかで正しい時刻をセットして、hwclockの同期オプションを使ってシステムクロックとハードウェアクロックを合わせます。

```
# hwclock --systohc
```

```
# hwclock --hctosys
```

Question 193



システムの時刻を正確に合わせたい

A. PCがハードウェアクロックとして使っている時計は、それほど正確なものではありません。厳密に複数のマシンで時刻を同期させたい場合などには、NTP (Network Time Protocol) を使います。Linuxで利用できるNTPツールとしては、xntp3 (<http://www.eecis.udel.edu/ntp/>) があります。ダイヤルアップ環境なら、xntp3に含まれるNTPクライアントntpdでNTPサーバにつないで時刻を同期させることができます。常時接続の環境なら、xntpdデーモンを走らせておけば、ほとんど誤差なく同期をとることができます。

また、Linux上でxntpdを動作させておけば、そのマシンがNTPサーバとなります。LANでつながったWindowsマシンで“桜時計”などのNTPクライアントを実行すれば、このマシンも正確な時刻に同期させることができます。

xntp3は、NTP Ver.3に対応しており、最新版はNTP Ver.4に対応したntpになります。現状では互換性の問題が発生しないようにxntp3を使ったほうがよいかもかもしれません。



Question 194

メールに添付されて送られてきたWordやExcelのファイルを読んだり、印刷したりできますか？

A. WordとExcelのファイルのどちらも、そのままLinuxで扱うことはできません。これらのファイルをHTMLに変換するツールがありますので、利用してみてもいいでしょうか？ 複雑な文章を完全に変換するのはムリですが、HTMLに変換しておけば、Netscape Navigator (Communicator) などのWebブラウザで読むことも、印刷することも可能です。WordやExcelのファイルから、テキストを抽出するツールもあります。

```
mwordveiw
```

Word 8のファイルをHTMLに変換するツールです。現在はwvWareというツールに移行しています。wvWareはVine Linux 2.0のVinePlusというツール集に含まれています。一次配布元である<http://www.wvware.com/>からもダウンロードできます。

```
xlHtml
```

Excel 95以降のファイルをHTMLに変換するツールです。一次配布元は<http://www.xlhtml.org/>です。

デ変研DocCat

データ変換研究所 (<http://www.dehenken.co.jp/>) が発売しているツールです (価格は4600円)。対象ファイルからテキストのみを抽出します。95以降のWord/Excelだけでなく、97以降のPowerPoint、Version9以降の一太郎に対応しています。

また、ApplixwareやStarOffice (オフィススイート)、Gnumeric (スプレッドシート) といったLinux用アプリケーションでも処理できます。この場合は、アプリケーション独自のファイル形式に変換しますから、もとのファイルにより近い形で扱うことができます。ただし、日本語が扱えない、サポートされるWord/Excelのバージョンに限られる、といった問題点があります。



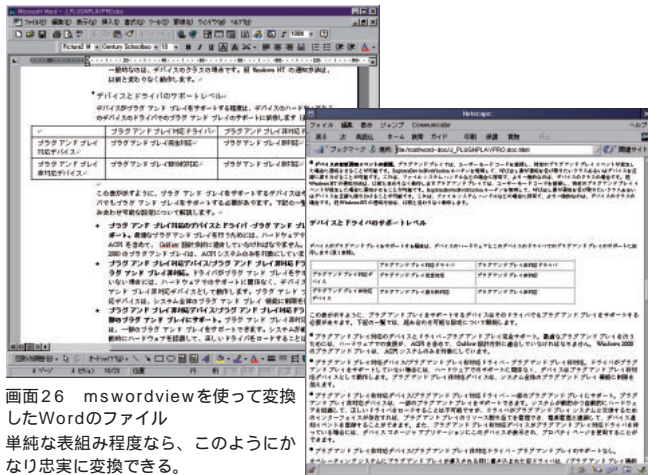
Question 195

Netscape NavigatorのブックマークをWindowsとLinuxで同じものにできればなあ、と考えています。可能ですか？

A. ブックマークファイルはそれぞれ、Windowsでは¥Program Files¥Netscape¥Users¥<ユーザー名>¥bookmark.htm、Linuxではユーザーのホームディレクトリのnetscape/bookmarks.htmlに保存されています。どちらもHTMLファイルですから、互換性の問題はありませぬ。どちらかのファイルを上書きコピーすれば、ブックマークを移行することができます。

ネットワーク上にsambaで設定された共有ディレクトリがあれば、そこにブックマークファイルを置いて、インポートして共有することも可能です。

ブックマークの項目名に日本語を使っている場合には、注意が必要です。文字コードの関係で文字化けしてしまうのです。nkfプログラムなどで、あらかじめ文字コードを変換しておきましょう。



画面26 mswordviewを使って変換したWordのファイル
単純な表組み程度なら、このようになり忠実に変換できる。



Question 196

/tmpはなんのためにあるの？

A. /tmpはシステムが一時的なファイルを作成するために使用するディレクトリです。一時的な作業スペースとして使用することができますが、Linuxでは/var/tmpを使うほうがよいでしょう。/var/tmpはviなどのコマンドが一時的なファイルを作成する場所でもあります。/usr/tmpも同様ですが、通常は/var/tmpへリンクされています。

また/tmpには、どのユーザーからもファイル操作が可能なパーミッション設定になっています。そのため、複数のユーザーでマシンを共有している場合などに、ほかのユーザーとのデータの受け渡し場所として適しています。このことは同時に、参照されても構わないようなファイルだけを置くようにしたほうがよいということも意味します。重要なファイルは/tmpに置かないようにしましょう。



Question 197

Linuxって著作権フリーなんでしょうか？ コピーして、人にあげてもだいじょうぶですか？

A. Linuxカーネルの著作権は、Linus Torvaldsさんが保持しています。また、カーネルの一部はコードを書いた開発者が著作権を持ちます。ですから、PDS (Public Domain Software = 著作権を放棄したソフトウェア) ではありません。

カーネルの配布はGPLに基づいて行われています。GPLとは、FSF (Free Software Foundation) によって策定されたライセンスです。GPLでは複製・変更・配布が許されていますが、再配布を制限することは禁止されており、また、ソースコードが入手できることを保証しなければなりません。公式な文書は、英語で書かれた原文 (<http://www.gnu.org/copyleft/gpl.html>) のみですが、英語が苦手な方には、日本語訳 (<ftp://ftp.sra.co.jp/pub/gnu/local-fix/GPL2-j/gpl.text.gz>) も参考になるでしょう。

カーネル以外のプログラムの配布条件は、それぞれのプログラムのライセンスに従います。GPLに従っているものは、再配布しても問題ありません。商用フォントや商用ソフトウェアは再配布できませんので気をつけましょう。



Question 198

Linuxのマスコットはなぜペンギンなの？

A. 1996年の初頭、linux-kernelメーリングリストで、Linuxにはどのようなロゴやマスコットが合うかが議



論されていました。ほかのOSロゴのパロディがよいとか、サメやワシといった強くて気高いものがよいという意見が数多く出されましたが、Linusさんがペンギンが好きだと発言したことで、ペンギンに決まりました。そして、ペンギンロゴコンテストが開かれ、Larry EwingさんがGimpで描いたTuxがロゴとして採用されたのです。

この数年前、Linusさんは、オーストラリアの動物園で小さなペンギンにかまれ、“ペンギン中毒”に感染してしまったのです。これに感染すると、ペンギンのことばかりを考え、ペンギンラブ状態に陥るのだそうです。Linuxにマスコットが必要だということになったとき、ペンギン中毒のLinusさんの頭のなかにまず浮かんだのがペンギンだったというわけです。Tuxをロゴキャラクターに決めたのは、面白みを持ったキャラクタだからとのことで、これはLinuxにとっても不可欠の要素なのだそうです。

Tuxに関しては、Steve BakerさんのWebページ(<http://www.woodsoup.org/sbaker/tux/doc/>)に詳しく説明されています。



Question 199

コンピュータ関係の雑誌や書籍でコンピュータ名やユーザー名によく使われる「hoge」ってなんですか？ 気になって夜も眠れません。教えてください

A. hogeに関してはさまざまな由来が語られています。が、英語圏におけるプログラム構文やネットワーク構成の説明で、用例として使われる「foo」、「bar」の日本語訳という説が有力です。foo、bar自体も、海兵隊のスラング説、アメリカンコミック起源説、中国語あるいはイディッシュ起源説などがあり、語源がはっきりしていません。特に意味を持っていない、あたり障りがない、といった意味合いで「ほげ(=hoge)」を訳語としたものと思われまます。hogeに関する学術的な報告(<http://www.selab.tutkie.tut.ac.jp/yoshida/hoge.html/>)もなされており、これなどはhogeの浸透を物語る一例といえるでしょう。



Question 200

コンソールで作業するとき、たまには気分を変えて違ったテキストカラーを使ってみたいのですが、可能でしょうか

A. コンソールの色は、settermコマンドを使って変更できます。-inversescreenオプションを付けて実行するとデフォルトの白黒が反転されます。また、-foregroundと-backgroundオプションで色を指定することも可能です。たとえば、テキストの色を青、背景色を白にするコマンドラインは以下のようになります。

```
# setterm -clear all -foreground blue -background
white -store
```

テキストと背景に同じ色を指定すると、ちょっと困った状態になるので気をつけてください。

X環境のコンソールの場合、使っているコンソールエミュレータによって設定が異なります。ここでは、もっともポピュラーな日本語コンソールエミュレータであるktermを例に説明します。ktermでは、ctrlキーを押しながらマウスの両方のボタン(3ボタンマウスの場合はまん中のボタン)を押すとメニューが表示されます。メニューから画面を反転表示するオプションを選ぶと、現在設定するテキストの色と背景色が反転します。

色の設定は起動コマンドで指定できます。“-fg <色>”がテキストの色、“-gb <色>”が背景色の指定です。ウィンドウマネージャのメニューやデスクトップのショートカットアイコンから起動している場合は、それぞれにこのオプションを追加する必要があります。



Question 201

Linuxは何に使えばいいのでしょうか？

A. 多少CPUが遅かったり、メモリが少ないIPCでも、X Window Systemを使う必要のないネットワークサーバとしてなら、Linuxで十分に利用できます。新たにPCを購入したりして、あまり使われなくなったPCがあれば、Linuxをインストールしてホームサーバの構築に挑戦してみてもどうでしょうか。ネットワークはLinuxの得意分野のひとつですので、充実した環境を手に入れることができます。

もちろん、それなりの経験やスキルは必要となりますが、それらを身につけていくにつれて、できることがどんどん広がるのもLinuxのおもしろいところです。

```
kterm
Parts (C) 1996 By Greg J. Badros <gjb@cs.duke.edu>
! You may use this file as specified under the GNU General Public License
!
! =====
! xterm (and friends)
!
XTerm*highlightSelection: true
! Uncomment this to use color for the bold attribute
XTerm*VT100*colorBDMode: off
! XTerm*VT100*colorBD: blue
! Uncomment this to use color for underline attribute
XTerm*VT100*colorULMode: on
XTerm*VT100*underline: off
XTerm*VT100*colorUL: magenta
! Uncomment this to display the scrollbar
XTerm*scrollBar: true
!
! This resource specifies whether or not to ignore the 'alternate screen'
! of applications such as vi. When it is on, these applications will restore
! the contents of the screen when they are exited to what they were before
! they were started. When it is off, the contents of vi will remain on the
! screen after the program is quit.
XTerm*VT100*titleInhibit: true
! Defaults: unmodified, euc-jp: line 1
```

画面27 ktermの色を変更すればviもemacs風に？

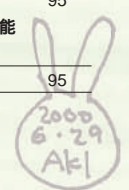
Question Index

NO. セクション	Question	ページ
1 基礎	Linuxをインストールしたいのですが、どれくらいのディスクスペースを用意すればいいのでしょうか	44
2	PC-9801にLinuxをインストールしたいのですが、友人には「ムリだ」と言われました。本当にだめなのでしょうか	44
3	テキストログインのときに、タイプミスしてBackSpaceキーを押すと“^H”という変な文字が入力されます。キーで戻ろうとすると“^[D”となって困ってます	44
4	PCの電源を切りたいんだけど、いきなりスイッチを押して大丈夫ですか？	44
5	システムが反応しません。リセットしていいですか？	44
6	あまりにもコマンドが多すぎて使い方がわかりません	44
7	rootで使い続けるのはよくないと聞きました。ほんとはどうですか？	44
8	一般ユーザーはどうやって作るの？	45
9	WindowsではInternet Explorerを使っています。馴れているのでLinuxでも使いたいのですが、メニューにありません	45
10	しばらくほっておくとコンソール画面がブランク画面になってしまいます。いちいち復帰するのは面倒なので、できればずっと表示したいのですが.....	45
11 ディストリビューション	ディストリビューションとはなんですか？	46
12	ディストリビューションは何種類あるのでしょうか？	46
13	それぞれのディストリビューションはどこが違うのですか？	46
14	カーネルやglibcのバージョンの違いは気にしなくていいの？	46
15	パッケージとはなんですか？	46
16	FTP版と製品版との違いはなんですか？	46
17	RPMとはなんですか？	47
18	rpmファイルなら、RPM系のどのディストリビューションでもインストールできますか？	47
19	PlamoやSlackwareでRPMパッケージを使うには？	47
20	debってなに？	47
21	RPMとdebでは互換性はあるのですか？	47
22	日本語ディストリビューションは、どこが「日本語」なんですか？	47
23 ハードウェア	私の使っているPCでLinuxは動作するの？	48
24	Linuxで動作するパーツなのかを調べるには？	48
25	グラフィックスカードをインストーラが正しく認識しません	48
26	内蔵モデムが使えません	48
27	サウンドカードの設定はどうするのですか？	49
28	プリンタに出力したいのですが？	49
29	UltraDMA/66対応のハードディスクは使えますか？	49
30	ハードディスクのアクセスが遅いような気がします	50
31	64Mバイト以上のメモリが認識されません	50
32	SCSIカードはなにが使えますか？	50
33	ハードディスクのRAID対応は？	50
34	USBデバイスは使えますか？	51
35	IrDAは使えますか？	51
36	CD-R/RW、DVD-ROM、DVD-RAM、MOドライブを接続したいのですが	51
37	CD-R/RWドライブにデータをバックアップするには？	51
38 システム起動	LinuxのインストールCD-ROMから起動できない	52
39	LILOプロンプトがLで止まって起動しないのですが	52
40	1024シリンダを超えるパーティションにはインストールできないのですか？	52
41	LILOの起動時に、Windowsをデフォルトにしたい	53
42	起動フロッピーを作りたい	53
43	ランベールってなに？	54
44	デーモン (daemon) とはなんですか？	54
45	システムサービス (デーモン) の起動、停止の方法を教えてください	54
46	ときどき起動にもすごく時間がかかるのはなぜ？	54
47	rootのパスワードを忘れてしまった！	55
48	シングルユーザーモードでの起動を禁止したい	55

NO. セクション	Question	ページ
49 シェル	シェルってなんですか？	56
50	それぞれのシェルはどんな特徴がありますか？	56
51	自分が使っているシェルを知りたい	56
52	シェルを変更したい	56
53	.bash_profile、.bashrc、.bash_logoutは、どのような動きをするの？	56
54	lsしても (ドット) で始まるファイルが表示されません	56
55	.bash_profileの変更が有効になりません	56
56	変数の設定でexportするものとしなものがあるのですが	57
57	文字列は"と"のどちらで囲めばいいのでしょうか？	57
58	パスを簡単に追加したい	57
59	指定時間にプログラムを実行したい	57
60	定期的にプログラムを実行したい	58
61	suコマンドを実行してrootになったのに、管理者コマンドが使えない	58
62	長いコマンドやオプションを入力するのが面倒なんです	58
63	バックグラウンドでプログラムを実行したい	59
64	プロセスを一時停止したい	59
65	パスワードを変更したい	59
66	プロンプトを変更したい	60
67	コマンドの結果などが長すぎてスクロールしてしまいます	60
68	任意の文字列を含むファイルを探したい	61
69	任意のファイルがどこにあるのが探したい	61
70	ファイル名などを簡単に入力したい	61
71	同じ名前の別のプログラムが実行されてしまいます	61
72	以前のディレクトリにすぐ戻りたい	62
73	ホームディレクトリにすぐ移動したい	62
74	ディスクを圧迫している大きなファイルを探したい	62
75	ディレクトリの階層構造を知りたい	62
76	画面をクリアしたい	63
77	LinuxコマンドとDOSコマンドとの違いはなんですか？	63
78	ログファイルの新しい部分だけを読みたい	63
79	自動的にログアウトしたい	63
80	他のユーザーがログインしているかどうかを調べたい	63
81	Linuxでは最大でどのくらいの大きさのファイルが作れますか？	63
82 ディスク・ファイル	テキストファイルを読みたいのですが	64
83	ディレクトリを丸ごと削除したいのですが	64
84	フロッピーディスクをフォーマットしたいのですが	64
85	ファイル名が ".tar.gz" で終わっているファイルは、何ですか？	64
86	アーカイブファイルはどうやって元に戻すの？	65
87	ほかのマシンで作ったアーカイブを解凍したら、ファイルのパーミッションが変わってしまいました	65
88	Windowsとテキストファイルをやりとりすると改行がおかしいのですが？	66
89	コマンドでファイルをゴミ箱に捨てるには？	66
90	ファイル名を変更するコマンド (rename) はないの？	66
91	ハードリンクとソフトリンクはどう違うの？	66
92	ファイルの終わりは、どうやって指定するの？	67
93	dfでディスクを見ると100%を超えているんですが？	67
94	40Gバイトのハードディスクをつないだのですが、33Gバイトと認識されてしまいます	67
95	2つのファイルの差を調べたい	67
96	あるディレクトリ以下のファイルすべてを調べるには？	67
97	ログファイルやバックアップファイルを削除したい	67
98	特定の文字列を含むファイルを探したい	68
99	起動時にプログラムを自動的に実行するには？	68
100	FTPするとファイルが化けるんですが？	68
101	フロッピーディスクやCD-ROMをマウント	68
102	rootユーザーしかマウントできないの？	69
103	アンマウントできないのはなぜ？	69
104	CD-ROMを自動的にマウントできませんか？	69
105	ディスクの使用状況を知るには？	69
106	Linuxでは最大でどのくらいの大きさのファイルが作れますか？	70

NO.	セクション	Question	ページ
107		128Mバイトのメモリを搭載しています。スワップはどれくらいが適当ですか？	70
108		消したファイルを復活できますか？	70
109		DOSのフロッピーディスクを読み書きできますか？	70
110		Macintoshのフロッピーを読み書きできますか？	70
111		Windowsパーティションを読み書きできますか？	71
112		ディスクを増設して/homeディレクトリの容量を増やしたいのですが？	71
113		Linuxのパーティションが削除できないのですが？	71
114		Linuxを削除したけど、LILOが消えません。どうしたら消せますか？	71
115	日本語	Linuxで使われている漢字コードは？	72
116		日本語が表示されないのですが	72
117		日本語クエールってなんですか？	72
118		かな漢字変換システムを切り替えるには？	72
119		コンソールでログインしたときに日本語表示をするには？	72
120		日本語が文字化けして読めません	73
121		X上のターミナルウィンドウで日本語が表示できません	73
122		画面がぐちゃぐちゃになってしまったのですが	73
123		日本語のファイル名は使えますか？	73
124		Windowsで作られたフロッピーの日本語ファイル名が表示されません	73
125		シフトJISのファイルをEUCに変換したいのですが	74
126		日本語のmanが表示されません	74
127		日本語の入力するには？	74
128		フリーで利用できるLinuxのかな漢字変換システムにはどのようなものがありますか？	74
129		LinuxにはWindows並の変換精度のかな漢字変換システムはないのですか？	75
130		日本語関連の便利なツールがあれば教えてください	75
131		Li18nuxとはなんですか？	75
132	X Window System	X Window Systemとはなんですか？	76
133		XFree86とX Window Systemはどう違うのですか？	76
134		X-TTとはなんですか？	76
135		ウィンドウマネージャとは？	76
136		ウィンドウマネージャを変更したいのですが	76
137		KDEやGNOMEはウィンドウマネージャじゃないの？	76
138		MotifやQtとかGTK+ってなんですか？	76
139		リモートマシンからローカルマシンにXアプリケーションを表示させたいのですが	77
140		X Window Systemが反応しなくなった	77
141		テキストログインをグラフィカルログインに変更したいのですが	77
142		グラフィカルログインから抜きたいのですが	78
143		Xconfiguratorを実行するとブラックアウトしてしまいます	78
144		一時的にXの画面解像度を変更したい	78
145		マウスの真ん中ボタンを有効にしたいのですが？	78
146		Netscape Navigatorでマウスのホイールを使いたい	78
147		i810チップセットでXFree86を使うときの注意点は？	79
148		GNOMEのメインメニューにプログラムを登録するには？	80
149		GNOMEパネルにランチャを登録するには？	80
150		データファイルとアプリケーションの関連付けはどうやるの？	80
151		GNOMEのデスクトップに置いたファイルはどこに保存されるの？	81
152		ファイルマネージャのファイルアイコンの意味は？	81
153		デスクトップに壁紙を貼りたい！	81
154		デスクトップアイコンのデザインを変更したい	81
155		ログインパネルから誰でもシャットダウンできるのは困る！	82
156		GNOMEでKDE用のクーティリティを使いたい	82
157		プログラム起動時にウィンドウの大きさを指定したい	82
158		Xのログインパネル(xdm)表示の時のデスクトップに壁紙を貼りたい！	83
159		ktermでスクロールバーを出すには？	83

NO.	セクション	Question	ページ
160		Xでスクリーンショットをとりたいのですが	83
161		WindowsでX Window Systemは使えないのですか？	83
162	ネットワーク	TCP/IPってなんですか？	84
163		ネットワークカードが認識されない	84
164		インストール時のIPアドレスの設定は、手動設定、BOOTP、DHCPのどれを選ばいいの？	84
165		pingを実行しても相手のホストから返事がありません	84
166		「unkonwn host」となるのですが	85
167		ホスト名はなんでもいいの？	85
168		IPアドレスは自由に付けてもいいの？	85
169		MACアドレスってなんですか？	85
170		ゲートウェイアドレスってなんですか？	86
171		サブネットマスクってなんですか？	86
172		ISDNルータを使う設定は？	87
173		モデム/TAを使う設定は？	87
174		IPアドレスやホスト名を変更したいが、どこを変えればいい？	87
175		IPマスカレードとは？	87
176		ネットワークカードを2枚挿していますが、1枚しか認識されません	88
177		2台のマシンでDNSサーバは必要ですか？	88
178		Windowsマシンとファイル/プリンタを共有したい	88
179		Macintoshとファイル/プリンタを共有したい	89
180		Webサーバを起動したい	89
181		rootでtelnetやFTPログインができない	89
182	misc	新しくハードディスクを追加したのでチェックをかけたいと思っています。Linuxではどうすればいいのですか？	90
183		起動時にエラーが出ているのですが、速すぎて読み取れません	90
184		コンソールの状態でも複数の画面が使えるってホント？	90
185		コマンドを実行しても“Permission denied”と表示されて何も起こりません。どうして？	90
186		カレントディレクトリにあるプログラムが実行できません。デフォルトではカレントにパスが通らないのですか？	91
187		coreという名前の巨大なファイルが知らない間にできてしまいました。削除しても大丈夫でしょうか？	91
188		バックアップとして、-hostsというファイルを作成したのですが、消せなくなって困っています	91
189		?マークの入った、得体の知らないファイルがあります。これも削除できますか？	91
190		アスタリスクを入力したはずなのに、ダブルクォーテーションが入力されてしまいます	92
191		Caps Lockキーと左Ctrlキーの機能を入れ替えたい	92
192		システムを起動するたびに時刻が狂ってしまいます。そのたびにdateコマンドで合わせているのに.....なぜ？	93
193		システムの時刻を正確に合わせたい	93
194		メールに添付されて送られてきたWordやExcelのファイルを読んだり、印刷したりできますか？	93
195		Netscape NavigatorのブックマークをWindowsとLinuxで同じものにできればなあ、と考えています。可能ですか？	94
196		/tmpは何のためにあるの？	94
197		Linuxって著作権フリーなんでしょか？ コピーして、人にあげても大丈夫ですか？	94
198		Linuxのマスコットはなぜペンギンなの？	94
199		コンピュータ関係の雑誌や書籍でコンピュータ名やユーザー名によく使われる「hoge」ってなんですか？ 気になって夜も眠れません。教えてください	95
200		コンソールで作業するとき、たまには気分を変えて違ったテキストカラーを使ってみたいのですが、可能でしょうか	95
201		Linuxは何に使えばいいのでしょうか？	95



常時接続サービス百花撩乱

インターネットサーバでキメる!

IP接続サービス、CATVインターネット、xDSLを使い倒そう

photo : Takashi Shinohara



Part 1

いま、常時接続が熱い！

Netfinity 1000

文：しのはらひろあき / 編集部
Text : Hiroaki Shinohara / Linux magazine

毎月一定額の料金を支払うだけで、常時インターネットを利用できる。1、2年前まで、そんな「常時接続」はほとんどの人にとって別世界の話だった。個人レベルで常時接続環境を導入しようと思えば、NTTの「OCNエコノミー」か、OCNを足回りに利用した他通信事業者の接続サービスしか選択肢がなく、しかも月額3万円以上の高嶺の花。気軽に利用できるものではなかった。毎月2万円を超える電話料金の請求書に顔を青くしながらも、OCN導入にこの足を踏んでいたパワーユーザーも少なくないはずだ。

しかし、今年になって個人でも利用できる常時接続環境が急速に現実味をおびてきた。CATVインターネットへの期待の高まり、規制緩和や業界からの働きかけによるADSLの首都圏試験サービスの実現に加え、5月からはNTT「IP接続サービス」の試験サービスエリアが拡大。誰もが恩恵を受けられるとはいかないが、月数千円でつなぎ放題という時代がすぐそこまでやってきているということは、もはや疑う余地がない。

インターネットへの常時接続環境が普及すれば、メールのチェックも気がねなくできるし1日中チャットすることもできる。だが、Linuxユーザーなら、常時接続がもたらす多くの可能性を敏感に感じとっているのではないだろうか。そこで、それぞれの接続サービスのあらましとともに、これらがもたらすLinuxの新たな使いこなしの道をさぐってみよう。



いま、個人ユーザーにも手の届く常時接続として注目されているサービスは、主に3つある。

IP接続サービス

従来からあるISDN回線を利用して定額料金での利用を実現したNTTのサービスだ。昨年11月から首都圏など地域限定で試験サービスが提供されてきたが、7月より正式名称を「フレッツ・アイ (FLET'S Isdn)」として本格始動することになった。

IP接続サービスの中核は、NTTが都道府県単位で構築する「地域IP網」。このネットワークに、ユーザーがISDN回線経由で、プロバイダが専用線で接続することで相互の通信がとりもたれ、インターネットへのアクセスが可能になる(図1)。ユーザーからみると、地域IP網へは通常のアクセスポイントと同様にダイヤルアップで接続するが、

回線はIP網への接続のみに利用されるため、電話料金を接続時間に応じて支払う必要はない。また、アクセス用電話番号も利用者個別に割り当てられるので“話中”の心配もない。

IP接続の最大のメリットは、既存の設備、ソフトウェアにほとんど手を加えずに移行できるという点だ。すでにISDN回線でダイヤルアップ接続を行う環境があるなら、接続先電話番号をNTTから交付された専用番号に、接続用アカウント名を「従来アカウント名@プロバイダ名」のように変更するだけで利用できる(radiusを使用したローミング接続だと思えばよい)。Linuxマシンをダイヤルアップルータがわりにしていたとしても、対応は容易だ。

いっぽうで大きなデメリットといえるのが、ISDNのBチャンネル1本にしかな対応していないこと。このため、速度は64kbpsが上限となっている。NTTへの月額利用料4500円(タイプ1)+プロバイダ料金月額300~3000円というコストは、理想論をいえば少々

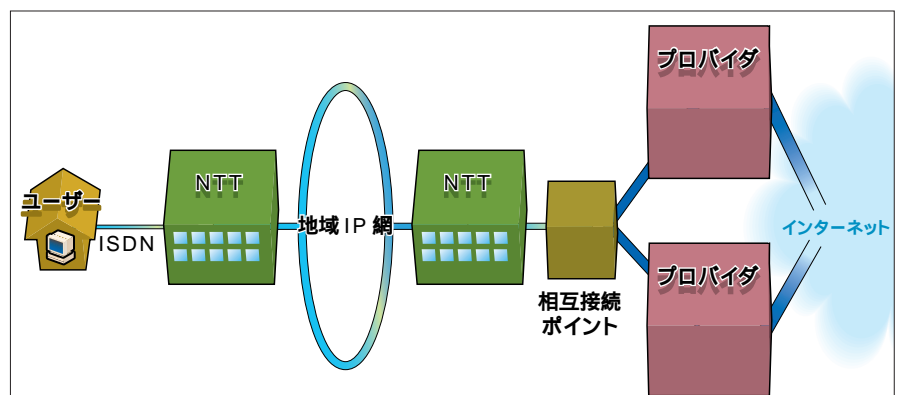


図1 IP接続サービス
利用者は、ISDN回線を経由して接続した地域IP網を介し、プロバイダにアクセスする。企業のプライベートネットワーク構築用プランもある。

高い感もある。とはいえ、8月には主要都市に提供地域を拡大、2000年度中に全国ISDN利用世帯の8割をカバーするというNTTの計画を信じれば、いまもっとも現実的な常時接続の選択肢であることは間違いのないだろう。

【NTT東日本 IP接続サービス】

<http://www.ntt-east.co.jp/teigaku/>

【NTT西日本 IP接続サービス】

<http://www.ntt-west.co.jp/ipnet/ip/>

CATVインターネット

メガビットクラスの大容量通信にも対応する広帯域インフラとして注目の

ケーブルテレビ（CATV）を利用したインターネット接続サービス。

PCは、イーサネットをつなげた“ケーブルモデム”を介してCATV回線と接続される（図2）。IPマスカレードなどを使用しての回線共有は技術的には容易だが、事業者によっては許可していなかったり、クライアント追加ごとに料金加算を科している場合もあるので、契約時に約款を確認したい。

日本におけるCATVは、テレビ難聴地域対策で利用されてきた。また、かつては行政による規制もあり、各局のサービスエリアが狭く事業規模も小さかったため、設備投資が難しく、インターネットアクセスラインとして普

及する足かせとなってきた。

サービス内容は事業者によってまちまちで、従量課金だったり、月額6000円で上限128kbpsなど、価格、速度ともにふるわないサービスが多かったのはこのため。とはいえ、プロバイダによるアウトソーシングの普及、J-COMやタイトスなど大手を中心にした合従連衡で、その問題も解決していきそうだ。ただ、居住地域でCATV業者がサービスしていなければ当然利用できないこと、集合住宅へのCATV回線引き込みが難しいことなど、導入が運に左右されてしまうという厳しい現実もある。

ADSL

既存の銅線アナログ電話回線で、音声帯域以外の周波数を使用して高速データ通信を実現するサービス。CATVインターネット同様、メガビットクラスの通信が可能だ。Asymmetric Digital Subscriber Lineの略で、この名は局設備からユーザーに向けた下り方向のデータ転送速度が、逆の上り方向よりも高速に設定されている“非対称型”であるところからきている。

ADSLは、アナログ電話回線の両端、すなわち電話局と加入者宅の双方に“ADSLモデム”を設置することで、データ通信を行う（図3）。とにかく電話線さえ引いてあれば導入できるということで、推進を求める声が強い。ただ、回線速度は回線品質に依存すること、日本ではISDN回線との干渉など検証すべき点が多い。そのため、今のところNTTのメタル線解放に沿って、各プロバイダが試験サービスを行っている段階。通信速度は東京めたりっく通信の場合で下り640kbps / 上り250kbps、利用料金は月額6300円（タイプ1-2）。他のサービスに比べると高速で安価な点が魅力だ。

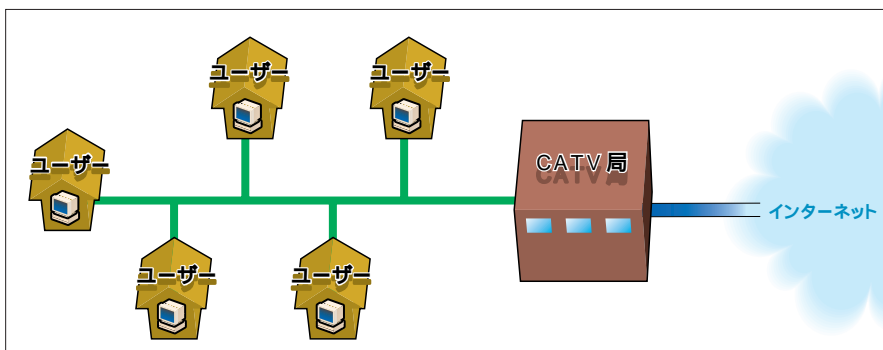


図2 CATVインターネット

CATV網をインターネットアクセスラインとして利用する。回線を利用者全員で共有する必要があるため、占有者がいると速度が遅くなる。

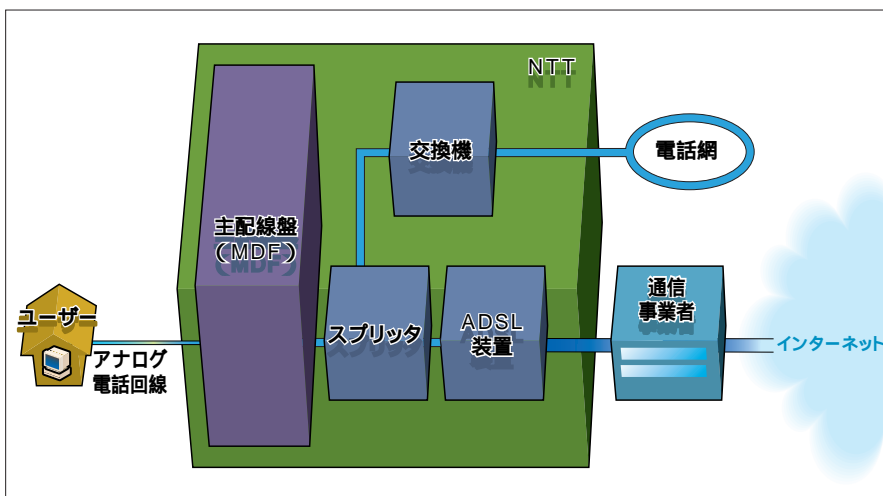


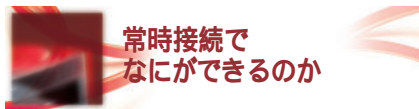
図3 ADSL

既存のアナログ電話回線に、デジタルデータを乗せる。電話、FAXなどは従来どおり使用できる。

夢のような常時接続サービスだが、現状では試験提供されている地域がまだまだ狭い。また、収容局から離れると伝送容量性能が悪化し、利用が困難な場合もあることには注意が必要だ。とはいえ、国内大手のプロバイダもADSL試験サービス参入を発表するなど普及への動きは急だ。東京めたりっく通信は、2000年中に東京、大阪を中心にサービス地域を拡大するようなので期待したい。

【NTT東日本 ADSL接続サービス】
<http://www.ntt-east.co.jp/senyo/ADSL/>

【東京めたりっく通信】
<http://www.metallic.co.jp/>



手の届くところまできた常時接続サービスだが、導入するとはたしてなにができるのか。メールチェックやWebアクセスが心おきなくできるのがまずひとつ。Linux利用者ならスクリプトを組んでインターネット上の情報を自動収集する、といったことがしやすくなるのもうれしい。もうひとつ、期待されるのが「サーバ公開用途」だ。自宅でWebサーバやメールサーバを立ち上げ、インターネットに公開する。もし実現できれば、従来は唯一の選択肢だったOCNエコノミーに比べ、非常に

安上がりですむ。しかし、それぞれのサービスの提供機能の違いから、こういった用途には一定の制限がかかることもある点に注意しておきたい。

大きな問題となるのが、利用時に付与されるIPアドレスの種別だ。OCNエコノミーの場合、契約ごとに数個のグローバルIPアドレスが付与される。グローバルアドレスなら、ドメイン取得、サーバ公開にはなんの不都合もない。しかし、CATVインターネット、ADSLサービスの場合、接続する端末にプライベートアドレスしか付与されないことがある。この場合、サーバ公開は事実上不可能だ。CATVでは、特にあてはまるケースが多い。ADSLでも、たとえば東京めたりっく通信の標準的プラン（A1・月5500円）はプライベートアドレス使用だったりする。

付与されるのはグローバルアドレスだが、接続ごとに異なるアドレス（非固定のアドレス）を割り振る事業者もある。この場合、サーバを公開することは可能ながら、独自ドメインでの運営はまずムリ。CATV、ADSLでは、この点をよく確認しておきたい。

微妙な存在なのがIP接続サービスで、一般的なダイヤルアップサービス同様、接続ごとに非固定グローバルアドレスが付与される（試験サービスに参加するプロバイダには、この形態が必須とされていた）。ここまでは特に変わったことではないが、実は、IP接続サービス対応プロバイダの中には他業

者との差別化のため、固定グローバルアドレスを割り振ることをウリにしているところも登場しているのだ。「VC-Net」（<http://www.vc-net.ne.jp/>）がそれで、特殊な例とはいえ、うまく活用すればサーバ公開はもちろん、独自ドメインでの運営も可能だ。回線の安定性から考えると万人にお勧めできるものではないが、腕試しに“専用線ごっこ”を楽しむには最適かもしれない。本特集「Part 2」の解説を参考にしつつ、公開サーバ構築にトライしてみるといいだろう。表1に各常時接続サービスでなにができるかをまとめた。



常時接続環境は便利で楽しい面ばかりではない。最近は、自動ツールでポートスキャンや既知のセキュリティホールをチェックを大量のホストに対してしかけ、穴のあるサーバが見つかる不正アクセスをするといった手口があたりまえになっている。ただつなげているだけだからとセキュリティホールを放置しておくのは非常に危険だ。さらに、踏み台として他のホストの攻撃に利用されたり、メールの第三者不正中継に使われたりすれば、コトはあなただけの問題ではない。インターネットのほかの利用者にも迷惑をかけることになる。安全で快適な常時接続ライフのために、「Part 3」の記事を参考にセキュリティ向上をはかろう。

サービス名	OCNエコノミー	IP接続サービス	CATVインターネット	ADSL
付与されるIPアドレス種別	固定グローバル	非固定または固定	主にプライベートまたは非固定グローバル	主に固定グローバルまたはプライベート
インターネットへのアクセス				
Webサーバの外部への公開		(IPアドレス直接指定時)	(グローバルアドレスなら直接指定で)	(グローバルアドレスの場合)
独自ドメインの運用		x (固定アドレスなら不可能ではない)	x	(グローバルアドレスの場合)
メールサーバの立ち上げ		x (固定アドレスなら不可能ではない)	x	(グローバルアドレスの場合)

表1 常時接続サービスでなにができるか



Linuxサーバの接続方法

常時接続サービスを利用するからには、LANで接続されたクライアントマシンからもインターネットを利用できるようにしたい。ところが、ISP (Internet Service Provider) から貸与されるIPアドレスは数が限られており、その数を超えるPCで同時に接続することはできない。

この問題を解決するのがLinuxのIPマスカレード機能やISDNルータのNAT(Network Address Translation)機能だ。これらを使えば、LAN内のPCでインターネット接続回線を共有し、同時にインターネットを利用できるようになる。

それでは、Linuxサーバおよび、クライアントマシンをどのように接続すればいいのかについて考えてみよう。

LinuxサーバでIPマスカレード

CATVインターネット、ADSLを使う場合、あるいはISDN回線をTA (ターミナルアダプタ) で利用する場合はLinuxサーバでIPマスカレード機能を

使う。また、ISDNルータで接続するが、NAT機能を使わない場合も同様だ。

TA (ターミナルアダプタ) を利用する場合は、TAをLinuxサーバにつないでPPPでISPに接続し、イーサネットを介して他のPCから共有する。

CATVのケーブルモデム、ADSLモデム、ISDNルータで接続する場合は、Linuxサーバにイーサネットカードを2枚挿し、1つをモデム、ISDNルータに、もう一方をLANに接続する。

この接続方式は、LinuxサーバがインターネットとLANの間に設置される形となる(図4-A)。この場合、Linuxサーバが直接インターネットにつながるため、インターネット向けのサービスがしやすい反面、サーバのセキュリティ確保には気をつける必要がある。

ISDNルータのNATを利用する

フレッツ・アイを利用する場合は、通常のISDN接続と同じようにISDNルータのNAT機能を利用することができる。LANで接続されたPCでWebのブラウズなどを行うだけであれば、設定も簡単なこの接続方式が便利だ。また、Webキャッシュサーバのように、LAN

内のPCにだけサービスするサーバを立てる場合にもお勧めできる。

この接続形式は、インターネットとLinuxサーバの間にISDNルータが入ることになる(図4-B)。インターネットに公開するサーバを立てる場合は、ISDNルータにそのための設定をする必要がある。

本特集では、インターネットに公開するサーバがテーマなので、LinuxサーバでIPマスカレードを行う接続方式を中心に解説する。



IPマスカレードの動作原理

最初に、IPマスカレードの動作原理を把握しておこう。図5を見てほしい。LANでは192.168.1.0 / 24 (192.168.1.0 ~ 192.168.1.255) というプライベートIPアドレスのネットワークを使用しており、Linuxサーバは192.168.1.1というIPアドレスを持つネットワークインターフェイスeth0でこのネットワークにつながっている。同時に、Linuxサーバのeth1というネットワークインターフェイスには、ISPからaaa.bbb.ccc.ddd (実際は数字) というグローバルIPアドレスを付与されている。Linuxサーバは、2つのIPアドレスを持ち、LANとインターネットという2つのネットワークにつながっているわけだ。

さて、このとき192.168.1.2というプライベートIPアドレスを持つクライアントマシンがインターネットに接続したいとしよう。このマシンは直接インターネットにつながっているわけではないので、そのままではインターネットを利用できない。そこで、LinuxサーバでIPマスカレードが使えるようにし、クライアントマシンのデフォルトゲートウェイとしてLinuxサーバを指定する。すると、Linuxサーバがパケ

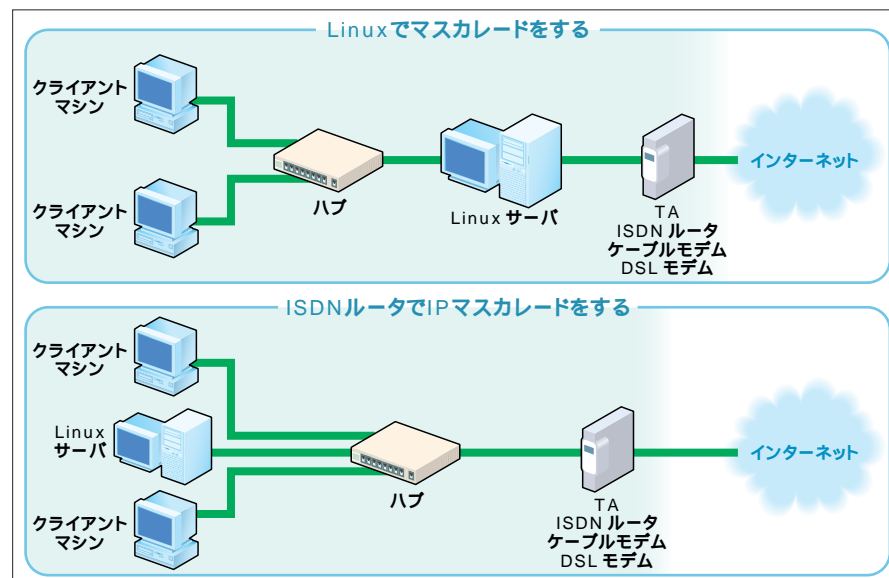


図4 LinuxでIPマスカレードする場合とISDNルータでNATを使う場合

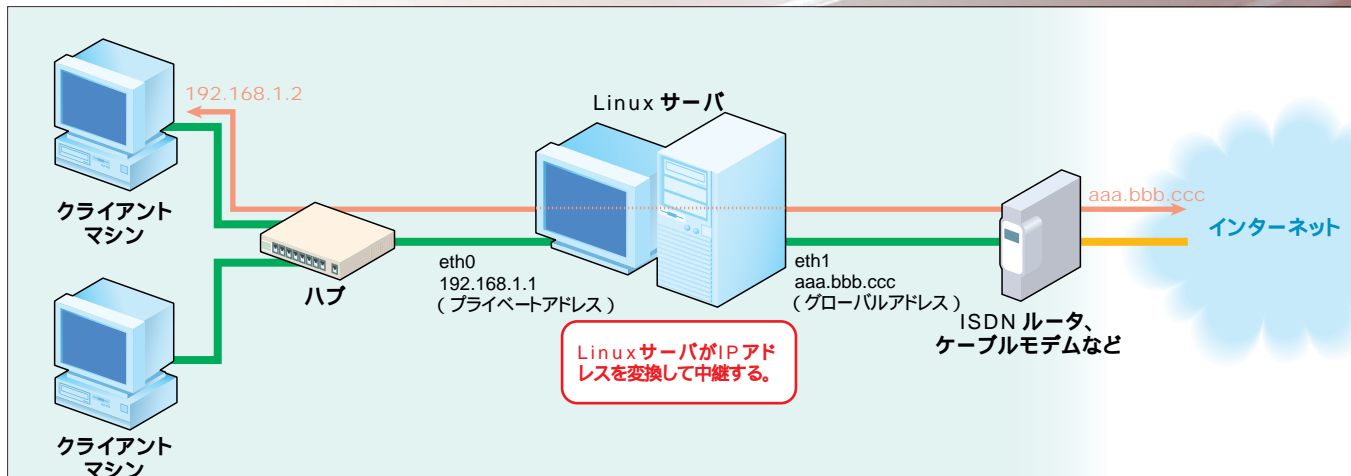


図5 IPマスカレードの仕組み

ット内の192.168.1.2というIPアドレスをaaa.bbb.ccc.dddに付け替えてインターネットへ中継する。インターネット側から送られたパケットは、aaa.bbb.ccc.dddを192.168.1.2に付け替えてクライアントマシンに送られる。この結果、クライアントマシンは、透過的にインターネットを利用できる。これがIPマスカレードの動作概要だ。

IPマスカレードの設定

それでは、LinuxサーバでIPマスカレードを使えるように設定しよう。ほ

とんどのディストリビューションでは、IPマスカレードを利用できるカーネルがインストールされている。ここでは、カーネルのバージョンが2.2.xであることを前提に解説する。

まず、IP転送を可能にする必要がある。Red Hat系のディストリビューションでは、/etc/sysconfig/networkファイルに記述されているFORWARD_IPV4 = falseのfalseをtrueに書き換える。それ以外のディストリビューションでは、/etc/rc.d/rc.localに“echo"1" > /proc/sys/net/ipv4/ip_forward”という行を追加する。Linuxサーバを再起動すれば準備は完了だ。あとはIP転送のポリシーを設定すればよい。

ポリシーの設定にはipchainsコマンドを使う。rootになって、リスト1のようにシェルから入力してみよう。1行目ですべてのパケットを転送しないようにしている。これはセキュリティ確保のために必須だ。続く2行目で、ソース（送信元）アドレスが192.168.1.0/24、ディスティネーション（送信先）アドレスが任意のパケットについては

アドレス変換をして外部と接続する設定だ。これでLAN内のクライアントマシンからもWebやtelnetのインターネット接続ができるようになる。ipchains -Lを実行すると、設定したポリシーを確認することができる。

ところが、そのままではFTPやIRCなどを利用することはできない。これらは個別に対応したモジュールが必要だ。モジュールは、/lib/modules/2.2.* /ipv4以下に置かれているので、必要なものをインストールする。たとえば、FTPを利用するには、

```
# modprobe ip_masq_ftp
```

を実行しておく。モジュールのインストールとipchainsの設定は、/etc/rc.d/rc.localに記述しておけばLinuxの起動時に自動実行される。

ここではIPマスカレードに関する最低限の設定を説明したので、セキュリティについてはほとんど考慮していない。本特集のPart 3を参考にして、より安全な転送ポリシーを設定してほしい。

Column

イーサネットカードの2枚挿し

多くのディストリビューションは、インストール時に複数のイーサネットカードを検出しないので注意しよう。1枚のカードしか認識されなかったときは、Linuxのインストール後に手動で設定する必要がある。設定には、Red Hat系のディストリビューションではLinuxconfやnetconf、TurboLinuxではturbonetcfgなどのツールを使うとよいだろう。

また、使用するイーサネットカードのドライバをモジュールで用意するか、カーネルに組み込んでおくことを忘れないようにしよう。

リスト1 IP転送ポリシーの例

```
# ipchains -P forward DENY
# ipchains -A forward -s 192.168.1.0/24 -d 0.0.0.0/0 -j MASQ
```

Part 2

インターネットサーバの設定

Netfinity 1000

文：山岸典将

Text : Norimasa Yamagishi

Part 2では、常時接続環境を前提としたインターネットサーバを構築するための、各種サーバアプリケーションについて

解説する。もちろん、すべてのサーバを立てる必要はない。必要なサーバから順に試してみよう。

IPアドレスの割り当てに関しても、動的に割り当てる方法と、ある固定のインターフェイス（具体的にはイーサネットカードのMACアドレス）に特定のIPアドレスを静的に割り当てる方法が選択できる。

DHCPサーバ `dhcpd`

LANを組むとなると、いくつものマシンにネットワーク情報を設定することになる。すべてのマシンに重複しないIPアドレスを正確に割り振ってやらなければいけないし、それ以外の項目については何度も同じことを設定する必要がある。これはかなり面倒な作業だ。そんな悩みを解決してくれるのがDHCPだ。

DHCP (Dynamic Host Configuration Protocol) だ。これにより、個別のクライアントでネットワーク情報を設定する必要がなくなる。すなわち、サーバ上でネットワーク情報を一元管理できるようになる。

ノートパソコンを会社と家の両方で使う場合など、ネットワークの頻繁な変更にも簡単に対応できる。DHCPで取得した情報はクライアントが自由に解放、更新することができるのだ。なお、DHCPは、クライアントのIPアドレスはもちろん、ゲートウェイアドレス、ネットマスク、タイムサーバアドレスなど、さまざまな情報を自動配布できる。



dhcpdのインストール

Linuxで動作する代表的なDHCPサーバアプリケーションがdhcpdだ。Red Hat系のディストリビューションであれば、RPMパッケージをインストールすればよいだろう。Red Hat Linux 6.2Jではインストール用のCD-ROMにdhcpdのRPMパッケージが入っている。

```
# mnt /mnt/cdrom
# rpm -ihv /mnt/cdrom/RedHat/RPMS/dhcp-2.0.5.i386.rpm
```

TurboLinuxでは、dhcpdは標準でインストールされる。



dhcpdの設定

dhcpdの設定ファイルは/etc/dhcpd.confだ。もっとも、rpmコマンドでは設定ファイルの作成などはしてくれないので、自分で1から作らなければならない。これは最初からdhcpdがインストールされるTurboLinuxでもRed Hat Linuxでも同じだ。dhcpd.confでは、表1のようなキーを使って設定を行う。ここでは、リスト1のような/etc/

DHCPとは

ネットワークに関する情報を持ったサーバが、クライアントのリクエストに応じて、ネットワークの設定情報を渡すというしくみを実現するのが

リスト1 /etc/dhcpd.confの設定例

```
option domain-name "linux-hoge.or.jp";
option domain-name-server 203.139.160.74, 210.160.90.114;
option router 192.168.1.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
default-lease-time 86400;
max-lease-time 259200;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.250;
    host levin {
        hardware ethernet 00:90:97:51:EA:C6;
        fixed-address 192.168.1.11;
    }
    host trueno {
        hardware ethernet 01:23:45:67:AB:Cd;
        fixed-address 192.168.1.12;
    }
}
```


dhcpd.confを作ったとして、これを例に解説しよう。

optionで始まる行は、DHCPクライアントに渡す情報とその値を指定している。よく使われるオプションは表2に示したとおりだ。ただし、これらの情報はDHCPクライアントが対応していなければ渡すことはできない。たとえば、time-serversという情報をWindows 98に渡そうとしても、このオプションに対応しないWindows 98では無視されてしまう。

default-lease-timeは、クライアントに渡すIPアドレスなどの資源貸し出し時間を秒単位で指定する。クライアントからのDHCPリクエストに貸し出し時間に関する要求が含まれなければ、ここで指定した時間だけ資源を貸し出す。この時間が過ぎると、クライアントは再びDHCPリクエストを行わなければならない。この例では86400秒、すなわち1日となっている。

max-lease-timeは、DHCPリクエストで資源の貸し出し時間も要求してきた場合の、最大貸し出し時間を秒単位で指定する。時間が過ぎた場合のクライアントの動作はdefault-lease-timeの場合と同じだ。この例では259200秒、すなわち3日となっている。

subnetで始まる{ }で囲まれたブロックの内部には、運用するサブネットごとに設定を書く。リスト1では、192.168.1.0/24というサブネットで割り

```

kterm
[root@mirage /root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:D0:09:3E:F2:84
          inet addr:210.160.90.117  Bcast:210.160.90.127  Mask:255.255.255.240
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2094 errors:0 dropped:0 overruns:0 frame:0
          TX packets:334 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:4 Base address:0xc000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:97 errors:0 dropped:0 overruns:0 frame:0
          TX packets:97 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

[root@mirage /root]#

```

画面1 eth0の「HWaddr」がMACアドレスだ

当てることのできるIPアドレスをrangeで指定している。この例では、動的に割り当てるIPアドレスは192.168.1.100 ~ 192.168.1.250までという意味になる。

さらに、hostブロック内でIPアドレスを固定して割り当てるホストを指定している。ここではイーサネットカードのMACアドレスが00:90:97:51:EA:C6というマシンに、192.168.1.11というアドレスを割り当て、名前はlevinとしている。複数のホストを指定して固定割り当てを行うことができる。なお、各行の最後に“ ; ”をつけるのを忘れないようにしよう。

このように、dhcpd.confでは、共通する情報は1回だけ書けばよく、入力間違いといったことは起こりにくい。

また、IPアドレスの動的割り当ては、同じアドレスを割り当ててしまうことがないため、DHCPを使うとネットワーク情報の管理が容易になるだろう。

dhcpd.confの設定が終わったら、dhcpdが配布したアドレスを記録しておくためのファイル/etc/dhcpd.leasesを作成しよう。

```
# touch /etc/dhcpd.leases
```

touchコマンドで中身が空のファイルを作る。dhcpdは、/etc/dhcpd.leasesを新たに作ってはくれないのだ。

ファイルを作成したら、次のように入力してDHCPサーバを起動しよう。

```
# /etc/rc.d/init.d/dhcpd start
```

キー	値
option	DHCPクライアントに渡す情報と値
default-lease-time	デフォルトのDHCP貸し出し時間
max-lease-time	最大のDHCP貸し出し時間
subnet	サブネットによる範囲の限定
range	貸し出すことのできるIPアドレス
host	固定のIPアドレスを割り当てる場合のマシン名
hardware ethernet	イーサネットのMACアドレス
fixed-address	固定IPアドレス

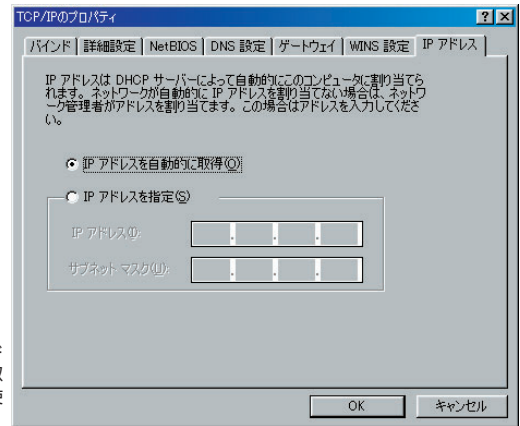
表1 dhcpd.confのキー

オプション	説明
domain-name	ドメイン名
domain-name-server	DNSサーバのIPアドレス
router	ルータのアドレス (ゲートウェイアドレス)
subnet-mask	サブネットマスク
broadcast-address	ブロードキャストアドレス
time-offset	UTCからのオフセット
time-servers	タイムサーバのIPアドレス
netbios-name-server	NetBIOSネームサーバのIPアドレス
pop-server	POPサーバのIPアドレス

表2 よく使われるオプション



画面2
winipcfgでは「アダプタアドレス」がMACアドレス



画面3 「IPアドレスを自動的に取得」がDHCPを使う設定だ

IPアドレスの静的割り当て

前述のように、MACアドレスに対して静的なIPアドレスを割り当てることができる。Linuxマシンのように、telnetでログインしたり、ネットワークアクセスすることが多いマシンは、静的なIPアドレスを割り当てよう。動的割り当てでは、ホスト名やIPアドレスが変わってしまうので、かえって不便になる。

LinuxマシンのMACアドレスを調べるにはifconfigコマンドを使う(画面1)。Windows 98ではwinipcfgコマンドで(画面2)、Windows NT / Windows 2000ならipconfigコマンドで調べることができる。

DHCPクライアントの設定

DHCPを使うための設定を、クライアントマシンのOSごとに説明しよう。

Windows 98の場合

コントロールパネルのネットワークで「TCP/IP <ネットワークカード名>」をダブルクリックし「TCP/IPのプロパティ」を開く。「IPアドレス」タブの中の「IPアドレスを自動的に取得」を選択し「OK」ボタンを押す。そして再起動すればDHCPを使う設定は完了だ(画面3)。

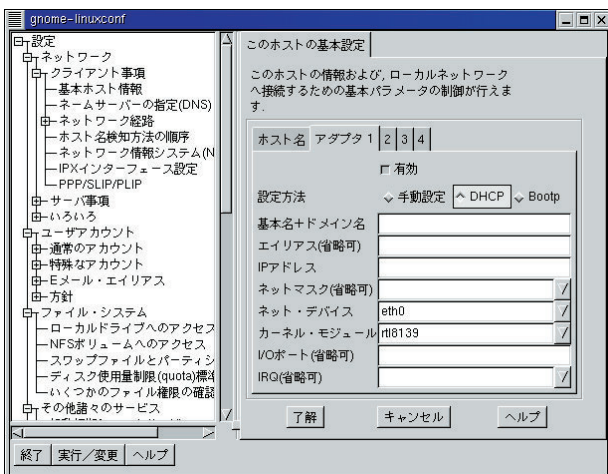
Linuxマシンの場合

Red Hat系の場合にはLinuxconfの

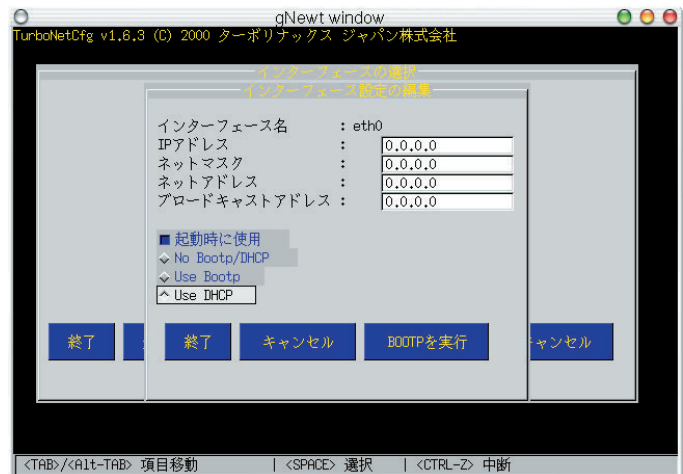
「基本ホスト情報」でDHCPを利用するように設定できる(画面4)。TurboLinuxの場合にはturbonetcfgの「ネットワークインターフェース」「編集」でDHCPを利用するように設定しよう(画面5)。

DHCPクライアントの設定が完了したら、実際にDHCPによりIPアドレスを受け取ることができるか試してみよう。Windows 98なら winipcfgで、Linuxならifconfigで、現在設定されているIPアドレスがわかる。

正常に動作しているようだったら、Linuxconfやntsysvコマンド、TurboLinuxならturboserviceコマンドを使ってdhcpdが自動起動するように設定しよう。



画面4 Red Hat系ではLinuxconfでネットワークの設定ができる



画面5 TurboLinuxではturbonetcfgで設定

プロキシサーバ Squid

プロキシサーバ (proxy) とは代理サーバという意味で、あるクライアントからの依頼を受け、そのクライアントの代わりに接続先サーバにアクセスし、クライアントとサーバの接続の中継を行う。ファイアウォールを構築する場合には、ちょうど壁となる部分に内部と外部のどちらにもアクセスできるプロキシサーバを置き、WebやFTPのアクセスを中継させることが多い。

プロキシサーバには、一度アクセスしたデータを保存しておき、再度同じデータへのアクセス要求があった場合には、保存しておいたデータを提供するキャッシュ機能を備えているものが多い。これにより、LAN内部から外部の同じサイトへのアクセスが何回か発生した場合、プロキシサーバ内部に保

存されているデータを再利用することができる。通常、内部ネットワークは外部への接続速度に比べて高速なので、応答も速くなり、外部とつながる回線への負荷も軽減される (図1)。

常時接続にした場合、ファイアウォールを設置し、プロキシサーバを使うことでセキュリティを向上させるのは必須といってもよい。また、そうでない場合もキャッシュ機能を利用することで、より高速なインターネットアクセスが可能となるだろう。

Squidは、Webキャッシュサーバとしての能力に優れているプロキシサーバだ。



Red Hat系のディストリビューショ

ンであれば、RPMを利用すればよいだろう。Red Hat Linux 6.2Jではインストール用のCD-ROMにSquidのRPMパッケージが入っている。rpmコマンドを使い、次のようにしてインストールしよう。

```
# mnt /mnt/cdrom
# rpm -ihv /mnt/cdrom/RedHat/RPMS/squid-2.3.STABLE1-5.i386.rpm
```

TurboLinuxでは、CD-ROMにはSquidのRPMパッケージが入っていないが、TurboLinuxのFTPサイトに用意されているので、次に示すURLのファイルを入手しよう。

```
ftp://turbolinux.co.jp/pub/TurboLinux/TurboContrib/RPMS/squid-2.3.STABLE2-2.i386.rpm
```

また、Squidのオフィシャルサイト (<http://www.squid-cache.org/>) から、ソースコードを入手してmakeすることも可能だ (画面6)。



Squidの設定は、/etc/squid/squid.confというファイルで行う。設定項目は非常に多いが、ほとんどの項目につ

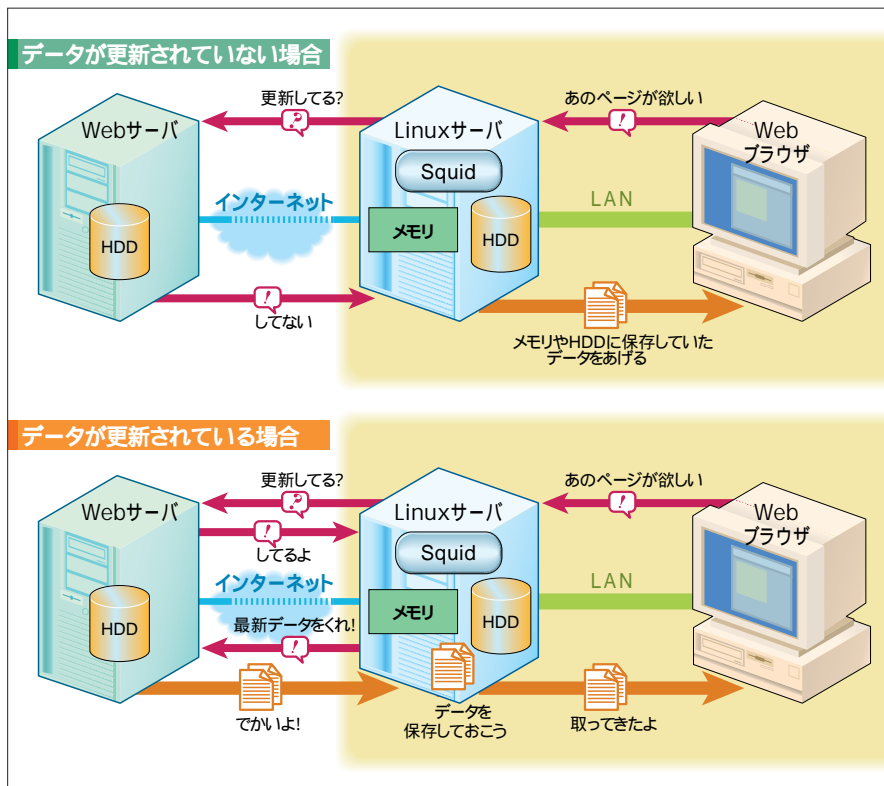


図1 キャッシュサーバSquidの動作



画面6 Squidのオフィシャルサイト

いては初期設定でも問題ない。

ここでは、設定しておきたい重要な項目に関して、説明しよう。#で始まる行はコメント行として無視される。なお、設定ファイル中には、コメント行として各設定項目の説明も書かれているので参考にしよう。

http_port

Squidに接続するためのポートを設定する。Squidは、ここで設定したポートを使ってクライアントからの接続要求を待つ。Squidを利用するには、Webブラウザなどでこのポートを指定する。初期設定は3128だが、Webプロキシサーバは8080や10080を使うことも多い。

```
http_port 8080
```

acl

Squidの利用を許可する対象のホストやポートのリストを定義する。この

あと説明するhttp_accessなど、アクセス制限の設定で利用する。

インストールしたままでは、localhostすなわち、Squidが動作するサーバ自身からしか利用できないようになっているので、aclとhttp_accessの追加設定は必須となる。

書式は以下のとおり。

```
acl aclname acltype <リスト>
```

aclnameにはリストの名前を自由に指定できる。http_accessなどで設定するアクセス制限はリスト名で行うことになるので、わかりやすい名前をつけるとよいだろう。

acltypeにはそのリストで定義する内容の種別を指定する。指定が必須といえるのが、アクセス元となるホストを指定するsrcだ。ほかにポートを指定するportや、利用できる時間を指定するtimeなども用意されている。

ホストは、IPアドレスのほか、ネッ

トワークアドレスとネットマスクにより指定することもできる。IPアドレスを直接指定する場合は、各リストに1つずつしか指定できないので、複数のホストからSquidを利用する場合は、ネットワークアドレスで指定するほうがよいだろう。

Squidを利用するホストのリストを作成して、既存のaclの行の下に追加しよう。

リスト2の例は、OCNエコノミーなどで利用する場合だ。mynetというリスト名で、IPアドレスが210.160.90.112～210.160.90.127までのホストを定義している。

なお、IPマスカレードなどを使い、プライベートアドレスに192.168.1.1などを使用している場合は、以下のように指定をする。

```
acl mynet src 192.168.1.0/255.255.255.0
```

http_access

httpプロトコルによる接続を許可するかどうかを、aclで設定したリストにもとづいて指定する。

```
http_access allow mynet
```

この行を追加すると、mynetという名前のリストにあるホストが、Squidを利用できるようになる。なお、「http_access deny all」という行がすでにある場合は、その行より前に追加する必要がある。

ftp_user

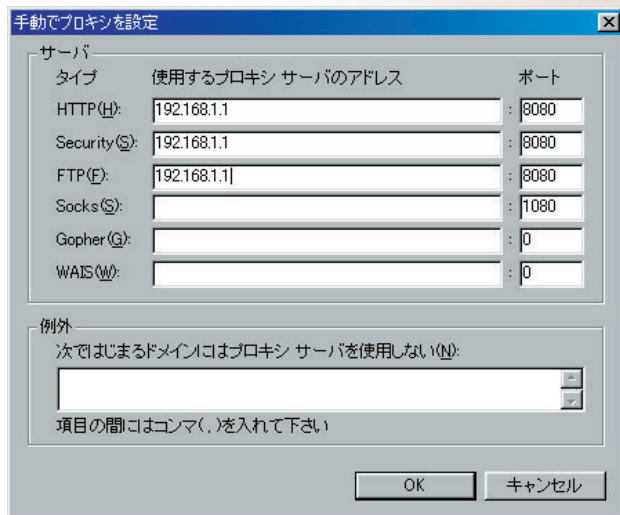
Squidを通して、anonymous FTPを行うときに相手のFTPサーバに対して通知されるメールアドレスを指定する。FTPサーバによってはこの項目がキチンと設定されていないと、接続を

リスト2 /etc/squid/squid.confの例

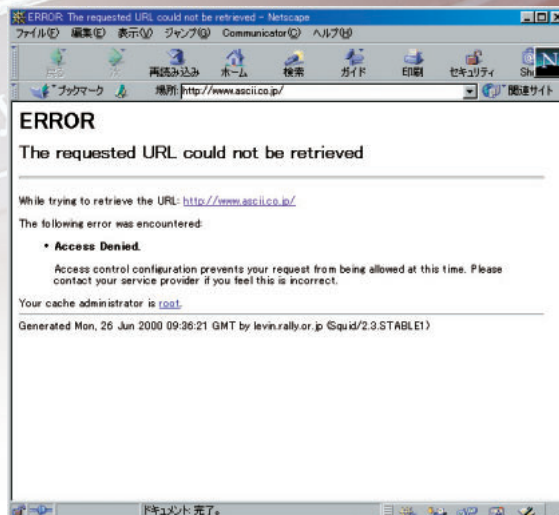
```
http_port 8080
:
ftp_user wwwuser@linuxmag.rally.or.jp
:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
acl mynet src 210.160.90.112/255.255.255.240
:
http_access allow localhost
http_access allow mynet
http_access deny all
:
icp_access allow all
miss_access allow all
```

このように許可したいホストのリストを追加する

mynetリストで指定されているホストのアクセスを許可する



画面7 Netscape Navigatorでのプロキシの設定例



画面8 アクセス制限に問題がある場合のエラー画面

拒否される場合がある。

```
ftp_user wwwuser@rally.or.jp
```

設定が完了したら、Squidを起動して、テストしてみよう。



Squidの起動は/etc/rc.d/init.d/squidというコマンドで行う。startという引数で起動、stopで停止、restartで設定ファイルを読み直して再起動させることが可能だ。

```
# /etc/rc.d/init.d/squid start
Starting squid: squid
```

Squidが起動したら、Webブラウザにプロキシの設定を行い、テストを試みよう。

プロキシサーバの名前はIPアドレスでもホスト名でもよいが、DNSの設定を行っていない場合は、IPアドレスで指定するしかないだろう。ポートはhttp_portで指定した値を記入する(画面7)。

Webブラウズができない場合は、

Squidの設定がブラウザの設定に間違いがある。http_portの値が間違っている場合には、タイムアウトしてしまい、まったく接続できない。aclかhttp_accessの記述が間違っている場合には、画面8のようにSquidのエラーメッセージが表示されるはずだ。squid.confを修正して再起動しよう。

```
# /etc/rc.d/init.d/squid restart
```

Squidは、RPMでインストールされただけでは自動的に起動されない。Linuxconfやntsysvコマンド(TurboLinuxならturboservice)などで自動起動するように設定しよう。



Squidを使い始めて間もないうちは、キャッシュにデータがないので、今ま

でも遅く感じることもあるかもしれない。しかし、使い込むほどに、よくアクセスするサイトのデータがたまり、効果が出てくるはずだ。

squid.confを変更することにより、メモリやハードディスク上のキャッシュ容量、キャッシュを保存しておく期間などを細かく設定することができる(表3)。たとえば、メモリ上にキャッシュする量を増やせば、レスポンスがよくなる。ただし、システムに搭載したメモリ量が少なく逆効果となってしまう。これらの値の最適値は、利用形態によって変わってくるので、いろいろと試して自分の環境に最適な値を見つけてほしい。

また、Squidは別のSquidサーバと階層構造を構築することもできる。ある程度規模が大きいネットワークではこの階層構造の構築も、Webアクセスの高速化には有用な手段だろう。

項目	説明
cache_mem	メモリ上にキャッシュするサイズ
cache_dir	キャッシュを保存するディレクトリとその容量
maximum_object_size	キャッシュするオブジェクトの最大サイズ
ipcache_size	DNSのキャッシュの数
dns_children	DNSキャッシュの子プロセスの数
reference_age	キャッシュしたオブジェクトを保存する期間

表3 squid.confのパフォーマンスに関係する設定

ネームサーバ BIND

インターネット上のコンピュータがほかのコンピュータに接続するときは、「210.140.231.23」のようにIPアドレスで相手を識別する。しかし、IPアドレスは人間にはわかりにくいので、「www.ascii.co.jp」のようなホスト名という、人間にとってわかりやすい名前をつけているのだ。このホスト名と、IPアドレスの変換をするのが、DNS (Domain Name System) だ (図2)。DNSは、DNSサーバ同士が互いにデータをやりとりする分散データベースシステムとして動作している。

インターネットに接続する場合、プロバイダがDNSサーバを提供してくれるので、外部からLANへのアクセスがない場合は必須のシステムではない。しかし、自分のコンピュータをWebサーバや、メールサーバとして公開する場合には、外部のコンピュータがLAN内部のコンピュータを参照する必要があるため必須となる。

また、自分のコンピュータを外部に公開しない場合でも、LANの中にDNSの情報をキャッシュしておくようにすると、外部のネットワークに対して問い合わせる頻度が減るので、名前解決

を高速に行えるようになる。

LinuxやUNIXでは、標準的なDNSサーバとして、BINDのバージョン8が使われることが多い。



Red Hat Linux 6.2Jではサーバインストールを選択すると、BINDがインストールされた状態になっている。Turbo Linux 6.0Jでもやはり、BINDはインストールされる。もし、BINDが入らないようなインストールをした場合には、bind-8.2.2_P5-9.i386.rpmとbind-utils-8.2.2_P5-9.i386.rpmといったRPMパッケージをインストールしよう。

BINDの設定は、複数のファイルにより行う。ただし、すべてについて解説をすると、1冊の本になってしまうほどBINDの設定は奥が深いものだ。DNSについて興味があるようなら、本誌で連載中の「Linux日記」を熟読されることをお勧めする (より本格的に知りたいなら、オライリージャパンの『DNS & BIND 第3版』を読むしかないだろう)。

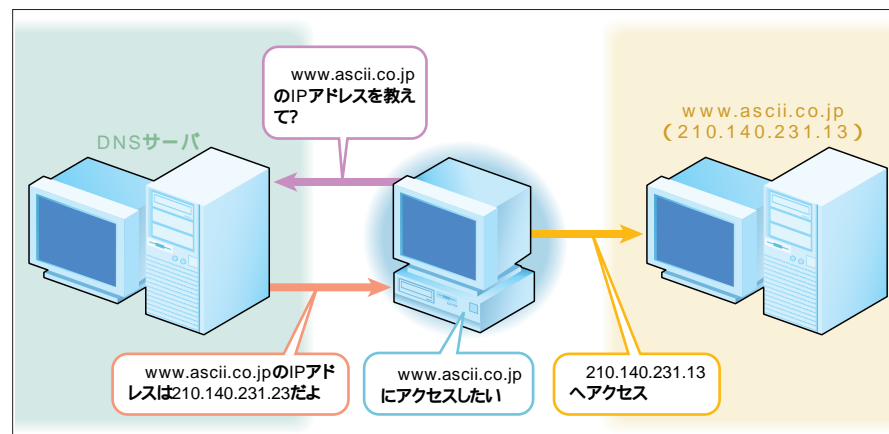


図2 DNSによる名前解決

今回は実用本位ということで、DNS情報をキャッシュするのみの、キャッシュ専用DNSサーバの設定について説明し、その後で自分のドメインの情報を提供するDNSサーバの設定について簡単に説明しよう。



外部ネットワークのDNSサーバを利用している場合には、キャッシュ専用DNSサーバ (キャッシュサーバ) を設定すると、外部のDNSサーバに問い合わせた情報がキャッシュサーバに保存され、2回目以降の名前解決を高速なLAN内部でできるようになる。キャッシュサーバの設定には難しい知識もそれほど必要がなく、効果が大きいので、お勧めのサーバ機能だ。

Red Hat Linux 6.2JでインストールされるBINDは、すでにキャッシュサーバ向けに設定されている。実際、そのままでもキャッシュサーバとして動作するが、動作効率をよくするため、/etc/named.confをリスト3のように書き換えよう。DNSサーバのIPアドレスに関しては、それぞれの環境に合わせて読み替えてほしい。

さらに、DNSサーバとして自分自身を使うように、/etc/resolv.confを次のように書き換える。

```
nameserver 127.0.0.1
```

named.confとresolv.confを書き換えたなら、BINDの実行ファイル/usr/sbin/namedを起動してみよう。

起動したら、まずnslookupで適当なホスト名が解決できるかを調べる。解決できたら再度nslookupで同じホスト名を調べてみよう。2回目にはキャッシュを使用しているので、「Non-autho

ritative answer:」というメッセージが表示されるはずだ。

```
# nslookup www.ascii.co.jp
Server: localhost
Address: 127.0.0.1

Name: at2.ascii.co.jp
Address: 210.140.231.23
Aliases: www.ascii.co.jp
```

```
# nslookup www.ascii.co.jp
Server: localhost
Address: 127.0.0.1

Non-authoritative answer:
Name: at2.ascii.co.jp
Address: 210.140.231.23
Aliases: www.ascii.co.jp
```

正常に動作することを確認したら、LinuxconfやturboServiceなどのコマンドを使い、BINDが自動起動するよ

うに設定しよう。



OCNエコノミーなどの常時接続環境では、独自ドメインを取得して運用することができる。この場合、自分のドメイン内のマシン情報を外部に提供するためにはプライマリとなるDNSサーバを設置する必要がある。

プライマリDNSサーバの設定は、キャッシュ専用DNSサーバの設定に比べるとかなり複雑になる。ここでは、表4に示すようなOCNエコノミー環境を仮定して、BINDの設定を説明する。OCNを申し込むと、表4のような設定情報をもらえるので、異なる部分から自分の環境に置き換えて設定ファイルをつくってほしい。

接続方法	OCNエコノミー
BINDのバージョン	BIND 8
ドメイン名	lm-hoge.or.jp
プライマリDNSサーバ	ns.lm-hoge.or.jp (192.168.69.32)
セカンダリDNSサーバ	ns-tk012.ocn.ad.jp (203.139.160.74)
ネットワークアドレス	192.168.69.32
サブネットマスク	255.255.255.240

表4 OCNエコノミーでのネットワーク環境例

内容	ファイル名
動作設定ファイル	/etc/named.conf
ルートキャッシュファイル	/var/named/named.ca
ローカルホスト逆引きファイル	/var/named/0.0.127.in-addr.arpa
正引きデータベースファイル	/var/named/db.lm-hoge.or.jp.zone
逆引きデータベースファイル	/var/named/db.lm-hoge.or.jp.rev

表5 BIND 8の設定ファイル

リスト3 /etc/named.confの例

```
options {
    directory "/var/named";
    /*
     * If there is a firewall between you and nameservers you want
     * to talk to, you might need to uncomment the query-source
     * directive below. Previous versions of BIND always asked
     * questions using port 53, but BIND 8.1 uses an unprivileged
     * port by default.
     */
    // query-source address * port 53;
    forward first;
    forwarders {
        203.139.161.74;
        203.139.160.74;
    };
};
zone "." {
    type hint;
    file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
```

本来のプライマリDNSサーバ

本来のセカンダリDNSサーバ

この5行を加える

設定ファイルの概要

BIND 8には、表5に示す設定ファイルがある。

DNSの名前解決には、正引きと逆引きが存在する。正引きとは、ホスト名からIPアドレスを解決することで、逆引きとは、IPアドレスからホスト名を解決することだ。DNSサーバはどちらの問い合わせに対しても回答できるように両方のデータベースを持つ必要があるのだ。

BINDのデータファイルは以下のよう構造になっている。

<名前> <クラス> <レコード種別>
<パラメータ>

<名前> は、解決対象となるデータだ。<クラス> はデータのクラスを表すが、現在ではインターネットを表すIN以外はほとんど使われない。<レコード種別> には、そのドメインで最も信頼できる情報であることを示すSOA、ネームサーバを示すNS、正引

きデータであることを示すA、逆引きデータであることを示すPTRなどがある。<パラメータ> は、<レコード種別> に対応するデータである。

動作設定ファイル

DNSサーバの全体的な設定は/etc/named.confで行う。基本的には、各設定ファイルの名前および、ファイルの位置が主な内容となっている。C++言語と同様に、//で始まる行および、/*と*/で囲まれた部分はコメントになる。

ここでは、リスト4を例として、各セクションごとに番号を振って説明しよう。

1. named.conf以外の設定ファイルが置かれるディレクトリを指定する
2. ルートキャッシュファイルを指定する
3. localhostの逆引きデータベースファイル名を指定する
4. ドメイン名を指定する
5. 正引きデータベースファイルを指定

する

6. ネットワークアドレスを逆にした値に「.in-addr.arpa」を加えて指定する
7. 逆引きデータベースファイルを指定する

ルートキャッシュファイル

インターネットすべてのネームサーバ管理階層の最上位には、複数のルートネームサーバが設置されている。ルートキャッシュファイルnamed.caには、ルートネームサーバのIPアドレスが記入されている。このファイルは、最初から用意されているので、そのまま使えばよいだろう。

最新のものはftp://rs.internic.net/domain/named.rootにあるので、必要ならばFTPで取ってくればよい。

ローカルホスト逆引きファイル

localhostの逆引きデータベースファイル0.0.127.in-addr.arpaは、DNSサーバ自身を定義するアドレスを指定する(リスト5)。

これ以降に説明する設定ファイルに共通の、気をつけてほしい点がひとつある。それはホスト名の後には必ず「.」を付けるということだ。これを忘れたがために、DNSサーバが正常に動作しないというトラブルはかなり多い。

先ほどと同じように、リスト中の番号に沿って説明する。

1. DNSサーバの名前を指定する。さらにメールサーバの管理者のアドレスを@マークの代わりに.を使って指定する
2. Serial: このデータが新しいものかどうかを判断するためのシリアル値を指定する
3. NS: DNSサーバの名前を指定する

リスト4 /etc/named.confの例

```
options {
    directory "/var/named"; _____ 1
};

zone "." {
    type hint;
    file "named.ca"; _____ 2
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "0.0.127.in-addr.arpa"; _____ 3
};

zone "lm-hoge.or.jp" { _____ 4
    type master;
    file "db.lm-hoge.or.jp.zone"; _____ 5
};

zone "32.69.168.192.in-addr.arpa" { _____ 6
    type master;
    file "db.lm-hoge.or.jp.rev"; _____ 7
};
```

2.のシリアル値は、セカンダリDNSサーバがプライマリサーバからデータをコピーする(ゾーン転送)際に使われる。セカンダリDNSサーバは、プライマリサーバのシリアル値が以前のものより大きければゾーン転送を行う。逆にいえば、ファイルを書き換えてもシリアル値が新しくないと、データの変更がセカンダリDNSサーバに反映されないので注意しよう。シリアル値は、1から順に付けていってもよいのだが、変更日の年月日に更新回数を付けると確実だ。

正引きデータベースファイル

正引きデータベースファイル/var/named/db.lm-hoge.or.jp.zoneは、ホスト名からIPアドレスを調べるためのデータベースだ(リスト6)。単純なホスト名とIPアドレスの対応のほか、ホストの別名も定義できる。各項目を順に説明しよう。

1. DNSサーバの名前を指定する。さらにメールサーバの管理者のアドレスを指定する
2. Serial: このデータが新しいものかどうかを判断するためのシリアル値を指定する
3. NS: プライマリDNSサーバの名前を指定する
4. NS: セカンダリDNSサーバの名前を指定する
5. MX: メールサーバの名前を指定する。
6. A: 正引きのためのホスト情報を指定する
7. CNAME: 別名を指定する

6.のAレコードは、

<ホスト名>. IN A <IPアドレス>

という形で、参照させたいホストを登録する。

7.のCNAMEは、woody.lm-hoge.or.jpというホストをwww.lm-hoge.or.jp

jpという名前でも参照できるようにしている。

逆引きデータベースファイル

逆引きデータベースファイル/var/named/db.lm-hoge.or.jp.revはIPアドレスからホスト名を調べるためのデータベースだ(リスト7)。

1. DNSサーバの名前を指定する。さらにメールサーバの管理者のアドレスを指定する
2. Serial: このデータが新しいものかどうかを判断するためのシリアル値を指定する
3. NS: プライマリDNSサーバの名前を指定する
4. NS: セカンダリDNSサーバの名前を指定する
5. PTR: 逆引きのための自分のドメインのネットワーク名を指定する
6. PTR: 逆引きのためのホスト情報を指定する

リスト5 /var/named/0.0.127.in-addr.arpaの例

```
0.0.127.in-addr.arpa. IN SOA ns.lm-hoge.or.jp. pm.ns.lm-hoge.or.jp. ( _____ 1
20000428 ; Serial _____ 2
10800 ; Refresh
3600 ; Retry
604800 ; Expire
86400 ) ; Minimum
IN NS ns.lm-hoge.or.jp. _____ 3
1.0.0.127.in.addr.arpa. IN PTR localhost.
```

リスト6 /var/named/db.lm-hoge.or.jp.zoneの例

```
@ IN SOA ns.lm-hoge.or.jp. pm.ns.lm-hoge.or.jp. ( _____ 1
20000428 ; Serial _____ 2
10800 ; Refresh
3600 ; Retry
604800 ; Expire
86400 ) ; Minimum
IN NS ns.lm-hoge.or.jp. _____ 3
IN NS ns-tk012.ocn.ad.jp. _____ 4
IN MX 10 mail.lm-hoge.or.jp. _____ 5
localhost IN A 127.0.0.1
rt.lm-hoge.or.jp. IN A 192.168.69.33
ns.lm-hoge.or.jp. IN A 192.168.69.34 _____ 6
woody.lm-hoge.or.jp. IN A 192.168.69.35
www.lm-hoge.or.jp. IN CNAME woody.lm-hoge.or.jp. _____ 7
```


6.のPTRレコードは、

<IPアドレスを逆順にした数字>. IN PTR
<ホスト名>
という形式で記述する。

以上のファイルを作成して、/etc/named.confで指定したディレクトリ(ここでの例では/var/named)に配置すれば、BINDの設定は終了となる。

DNSサーバを設定したら、/etc/resolv.confも変更しておこう。resolv.confではそのマシンのドメインと、参照すべきDNSサーバを指定する。今回の例では、次のように記述しておけばよい。

```
domain lm-hoge.or.jp
nameserver 192.168.69.32
nameserver 203.139.160.74
```



BINDの動作テスト

すべての設定が終わったら、BINDの実行ファイル/usr/sbin/namedを起動してみよう。起動したら、設定したホスト名が解決できるかをnslookupコマンドで調べてみよう(画面9)。正常に動作していれば、名前解決に使われたDNSサーバと解決結果が表示され

る。さらに、逆引きや設定したすべてのホスト名について正常に解決できるかを調べれば完璧だ。

正常に解決できない場合は、まず/var/log/messagesを見てみよう。設定を間違えてエラーが起きていれば、このファイルに出力されているはずだ。特に多い間違いは、ホスト名の「.」のつけ忘れと、シリアル値の変更忘れがある。

BINDが正常に動作することを確認したら、Linuxのブート時に自動起動するように設定しよう。自動起動の設定方法は、キャッシュ専用DNSサーバの場合と同じだ。最初にも書いたよ

うに、DNSは世界中に分散したデータベースシステムだ。1台のDNSサーバがすべてのデータを持つわけではなく、自分の担当するドメインの情報以外はほかのDNSサーバに問い合わせる。このように、DNSサーバ同士は、互いに連携しながらインターネット上にある膨大なホストの名前を解決しているのだ。問い合わせによって得られた情報は、一定期間キャッシュされるため、サーバの設定を変更しても、その内容がほかのサーバに伝搬されるにはしばらく時間がかかるので、運用する際には注意してほしい。

正引きをチェック

```
# nslookup ns.lm-hoge.or.jp
Server: ns.lm-hoge.or.jp
Address: 192.168.69.32
Aliases: 32.69.168.192.in-addr.arpa

Name: ns.lm-hoge.or.jp
Address: 192.168.69.32
```

逆引きをチェック

```
# nslookup 192.168.69.35
Server: ns.lm-hoge.or.jp
Address: 192.168.69.32
Aliases: 32.69.168.192.in-addr.arpa

Name: woody.lm-hoge.or.jp
Address: 192.168.69.35
Aliases: 35.69.168.192.in-addr.arpa
```

画面9 nslookupでDNSの設定をチェック

リスト7 /var/named/db.lm-hoge.or.jp.revの例

```
32.69.168.192.in-addr.arpa. IN SOA ns.lm-hoge.or.jp. pm.ns.lm-hoge.or.jp. ( 1
20000428 ; Serial 2
10800 ; Refresh
3600 ; Retry
604800 ; Expire
86400 ) ; Minimum

IN NS ns.lm-hoge.or.jp. 3
IN NS ns-tk012.ocn.ad.jp. 4
IN PTR lm-hoge.or.jp. 5
33.69.168.192 IN PTR rt.lm-hoge.or.jp. 6
34.69.168.192 IN PTR ns.lm-hoge.or.jp. 6
35.69.168.192 IN PTR woody.lm-hoge.or.jp. 6
```

メールサーバ Sendmail&qpopper

常時接続が可能になり、独自ドメインを取得した場合には、まず最初に使いたいのがメールだろう。

メールを利用するためには主に、メールを配送するためのプログラムであるMTA (Mail Transfer Agent) と、メールを読み書きするためのプログラムMUA (Mail User Agent) が必要になる。さらに、メールサーバとメールを読み書きするマシン (MUAが動作するマシン) が異なる場合には、メールサーバからメールを取ってくるために必要なPOPサーバが必要になる。MTAとPOPサーバを使ったメール配送の仕組みは図3のようになる。この中で、メールサーバに必要なプログラムはMTAとPOPサーバだ。

最近では、MTAとしてqmailというプログラムもよく使われるようになってきたが、今回は標準的なSendmailをMTAとして使用する。しかし、Sendmailというプログラムの設定は非常に難しい。そのため、Sendmailの設定ファイルを作成するための優れたプログラム、CFを利用することにしよう。

また、POPサーバはqpopperというプログラムを使うことにする。



Red Hat Linux 6.2Jでは、インストールタイプをサーバにした場合はSendmail-8.9.3-20が標準でインストールされる。TurboLinuxでインストールされるのはSendmail-8.9.3-16だが、問題になるような差異はないだろう。利用中のLinuxマシンにSendmailがインストールされていない場合は、インストール用のCD-ROMからインストー

ルする。また、最新版の情報は、sendmail.orgのWebサイト (http://www.sendmail.org) から入手しよう (画面10)。

Sendmailがインストールされているといっても、まったく設定はされていないので、設定ツールのCFをインストールし、これを用いてSendmailの設定ファイルを作成する。

CFのインストール

CFはRed Hat Linux 6.2JのCD-ROMには含まれていないので、Ring

Server ProjectのFTPサーバなどからダウンロードする必要がある。Ring Server Projectのサーバから入手するならば、ftp://ftp.ring.gr.jp/pub/net/mail/CF/以下にある3つのファイルを取ってこよう。

CF-3.7Wp12.tar.gz

CF-3.7Wp12-smtppcheck.patch1

CF-3.7Wp12-smtppcheck.patch2

TurboLinuxの場合は、/usr/srcの下にCFがインストールされている。ただし、2つのパッチは当たっていないので、これらを手に入れて当てたほうがよ

画面10 sendmail.orgのWebサイトで最新情報をチェック

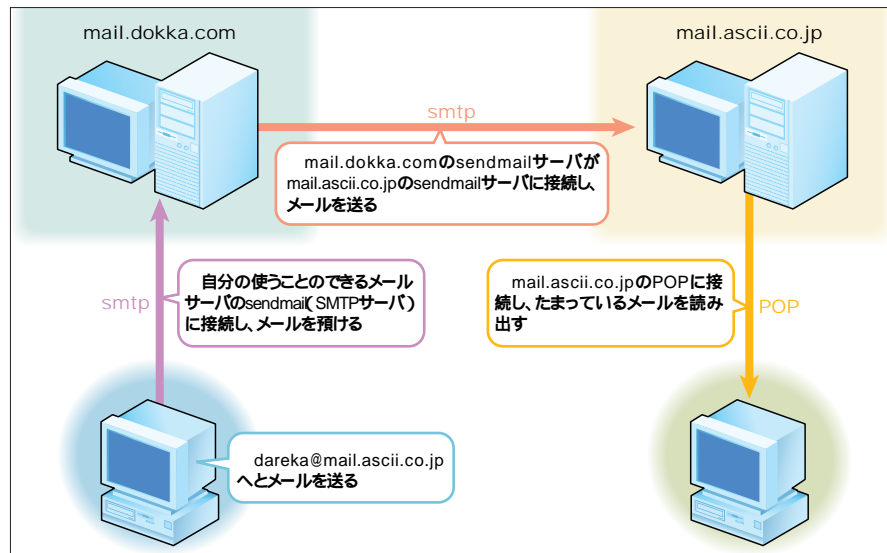
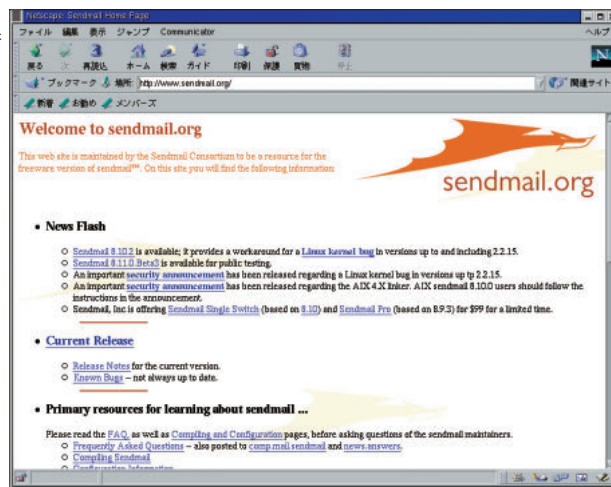


図3 メール転送の仕組み

いだらう。

CF-3.7Wp12.tar.gzを展開し、2つのパッチを当ててmakeする。これでCFのインストールは完了だ。

```
$ tar zxvf CF-3.7Wp12.tar.gz
$ cd CF-3.7Wp12/Master
$ patch < ../../CF-3.7Wp12-smtp
check.patch1
$patch < ../../CF-3.7Wp12-smtp
check.patch2
$ cd ..
$ make cleantools
$ make tools
$ make samples
```

CFの設定

次に、Standardsディレクトリの中にあるSendmail V.8用のサンプルファイルをコピーして、自分の環境に合う

ようにエディタで修正する。

```
$ cp Standards/sendmail-v8.def
sendmail.def
$ vi sendmail.def
```

この定義ファイルにある項目のうち、動作させるのに最低限必要な設定はOS_TYPEだが、そのほかに、外部のユーザーに勝手にメールサーバを使われないような設定も必ずしておこう(不正リレーの防止)。

また、メールサーバがmail.dirt.gr.jpという名前の場合に、メールアドレスとして、nor@mail.dirt.gr.jpではなくnor@dirt.gr.jpを使いたい場合もCFで設定する。

定義ファイルの“#”で始まる行はコメントとなる。リスト8の例を参考に、設定に必要な行だけコメント

を外して書き換えるとよいだろう。

sendmail.cfの作成

sendmail.defの編集が終わったら、Sendmailの設定ファイルsendmail.cfを作成しよう。sendmail.cfの作成にはmakeコマンドを使う。

```
$ make sendmail.cf
```

それでは、できあがったsendmail.cfをテストしよう。次のようにして、Sendmailの実行ファイル/usr/sbin/sendmailを起動する。

```
# /usr/sbin/sendmail -bt -C ./sendmail.cf
```

-btオプションはテストモードで起動することを意味する。-Cオプションは設定ファイルの指定だ。

テストモードで起動すると、“>”というプロンプトが表示され、入力待ちとなる。ここで、/parseコマンドに続けてメールアドレスを入力すると、そのアドレスをSendmailがどのように解釈するかを見ることができる。

画面11を参考に、各自でテストをしてほしい。特に注目すべきなのは最終行だ。サイト内部宛であれば、mailerがlocalでuserはユーザー名だけ。サイト外部宛であればmailerはsmtp、hostはhost名、userはユーザー名とホスト名となるはずだ。これが間違っている場合は設定をやり直す必要がある。

テストモードの終了はCtrlキー + Dをタイプすればよい。テストが終了したら、できあがったsendmail.cfを/etcディレクトリにコピーしよう。そしてSendmailを再起動すればメールの配送部分については完成だ。

リスト8 sendmail.defの変更例

```
OS_TYPE=linux-redhat
:
FROM_ADDRESS = '$m'
:
ACCEPT_ADDRESS = '$m'
:
MAIL_RELAY_RESTRICTION=yes
LOCAL_HOST_IPADDR='192.168.69.35 192.168.69.36'
LOCAL_HOST_DOMAIN=lm-hoge.or.jp
```

メールを送るときに“ユーザー名@ドメイン名”というアドレスで送る

“ユーザー名@ドメイン名”宛のメールも受け取る

メールサーバの利用に制限をかける

メールサーバを使うドメインを指定する

メールサーバを使うマシンのIPアドレスを指定する

```
# /usr/sbin/sendmail -bt -C ./sendmail.cf
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> /perse root@mail.lm-hoge.or.jp
Cracked address = $g
:
:
mailer local, user root
> /parse nor@dirt.gr.jp
Cracked address = $g
:
:
mailer smtp, host dirt.gr.jp., user nor@dirt.gr.jp
```

サイト内部の配送チェック

内部の配送方法

サイト外部の配送チェック

外部の配送方法

画面11 sendmail.cfのテスト


```
# cp sendmail.cf /etc
# /etc/rc.d/init.d/sendmail restart
```

TurboLinuxではSendmailは最初から動いているわけではないので、上の方法ではsendmailが使えるようにはならない。turboserviceなどで、自動起動に設定してやる必要がある。

sendmailがキチンと動き出しても、CFの設定ファイルであるsendmail.defは消去せずに残しておこう。Sendmailの設定を変更したいと思ったときに、sendmail.cfを見ただけでは、現在の設定がどうなっているのか理解できない恐れがある。実際にsendmail.cfの中身を見れば、いかにsendmail.cfの内容が難解で、CFによる設定が楽なのかがわかるはずだ。



qpopperの設定

Sendmailを設定し、メールを送ることはできるようになった。次はメールを受け取り、読むことができるようにしましょう。

もちろん、メールサーバ自体にログインし、サーバ上のプログラムを使ってメールを読むのなら、特別なプログラムはいらないのだが、Windowsで使い慣れたメールクライアントを使いたいという要求も多いだろう。その場合はPOPサーバを立ち上げて、クライアントからメールを読み出せるようにしなければならない。

今回は、暗号化パスワード認証にも対応したPOPサーバ、qpopperをインストールしてみよう。

qpopperのインストール

qpopperは、Red Hat Linux、TurboLinuxのどちらにも含まれていない。そのため、配布元である

QUALCOMM社のFTPサーバからソースを取ってきて、コンパイルすることになる。ソースファイルは次のURLに置かれている。

```
ftp://ftp.qualcomm.com/eudora/servers/unix/popper/
```

ファイルを入手したら、次の手順でコンパイル、インストールをしよう。

```
$ tar zxvf qpopper3.0.2.tar.gz
$ cd qpopper3.0.2
$ ./configure --enable-apop
--enable-specialauth
--with-popuid=mail
$ make
$ su
Password: <パスワード>
# install -m755 popper/popper /usr/sbin/
```

次に、クライアントがPOPプロトコルでアクセスしてきたら実行ファイルpopperが起動するように、/etc/inetd.confを修正する。行頭が「#pop-3」になっている行を探し、行頭の#を取り、リスト9のように修正すればよい。

修正できたら、inetdを再起動する。

```
# /etc/rc.d/init.d/inet restart
```

これで、POP3プロトコルにより、メールサーバからメールを読み出すことができるようになった。



APOPOPの利用

qpopperは、パスワードを暗号化し

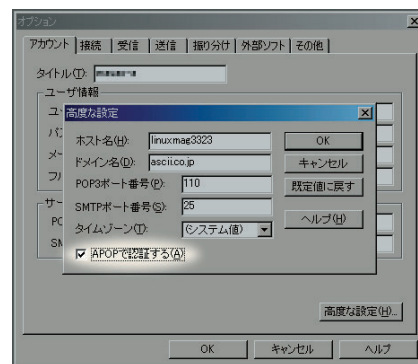
て送るAPOPという方法にも対応している。APOPを使うにはメールクライアントソフトもAPOPに対応している必要がある。たとえば、Windowsで動作するAL-Mail32はAPOPに対応しており、オプションでAPOPを使うかどうかを設定できる(画面12)。

APOPを使うには、popperとともに作られた、popauthというプログラムでパスワードを設定する必要がある。次のようにしてpopauthをインストールしておこう。

```
# cp popper/popauth /usr/bin
# chmod 4755 /usr/bin/popauth
# chown mail /usr/bin/popauth
# chgrp mail /usr/bin/popauth
```

パスワードの設定は、各ユーザーがログインして、popauthコマンドを使うのが確実だろう。

```
$ popauth
Changing only APOP password for nor
New password: <パスワード>
Retype new password: <パスワード>
```



画面12 AL-Mail32でAPOP認証を使うように設定する

リスト9 /etc/inetd.confを修正する

```
pop-3 stream tcp nowait root /usr/sbin/tcpd popper -s
```

Webサーバ Apache

常時接続環境になれば、自分のところでWebサーバを運用してみたいという誘惑にかられるのは、当然のことだろう。自前のWebサーバを持てば、次のような環境が手に入るのだ。

- ・データ容量は好きだけ使える。
- ・サーバの設定（だけでなく使うサーバそのものも）を自由にできる。
- ・CGI、SSI、PHP、PostgreSQLなど使い放題。

もっとも、「ドメイン取っていないから」という人や「固定IPアドレスじゃないからなあ」という人もいるかもしれない。そんな場合でも、今はWebページの転送サービスもある。常時接続環境とLinuxの組み合わせなんて、「Webサーバを立ち上げましょう」といっているようなものだ。

ただし、プロバイダによっては、サーバの運用を禁止しているところもあるので、約款を確認してほしい。サーバを運用しても大丈夫かどうか、よくわからなければプロバイダに問い合わせよう。

Apacheの起動

ほとんどのディストリビューションでは、WebサーバのApacheが自動的にインストールされ、しかも起動して

いる。試しにそのマシンに、Webブラウザでアクセスしてみよう。テストページが見えるはずだ（画面13）。テストページが表示されない場合は、ネットワークそのものの設定が間違っている可能性もある。もっとも、Apacheがインストールされていない場合もある。そのときは、RPMパッケージからApacheをインストールしよう。

Apacheの初期設定

Apacheは、バージョン1.3になったときに設定ファイルの見直しが図られ、以前のバージョンでは3つに分かれていた設定ファイルが/etc/httpd/conf/httpd.confに集約された。しかし、ディストリビューションによっては、以前のバージョンと同じように、次の3つのファイルに設定が分かれている。

```
/etc/httpd/conf/httpd.conf
/etc/httpd/conf/srm.conf
/etc/httpd/conf/access.conf
```

その場合でも、設定の書式は同じなので、設定項目ごとにgrepして該当するファイルを書き換えよう。

まず、初期設定のうちでも、重要なディレクトリ構成について説明する。htmlファイルが置かれるのは、基準となるディレクトリ（DocumentRootと

いう）だ。たいていのディストリビューションでは、/home/httpd/htmlが指定されている。たとえば、Webサーバがwww.lm-hoge.or.jpだとすると、http://www.lm-hoge.or.jp/で参照されるのは/home/httpd/htmlディレクトリとなる。

CGIを使用するためのディレクトリとして、/home/httpd/cgi-binが用意されている。このディレクトリに置かれた実行ファイルは、CGIプログラムとしてWebブラウザ経由でアクセスすることができる。Apache関連ディレクトリの初期設定を表7にまとめた。

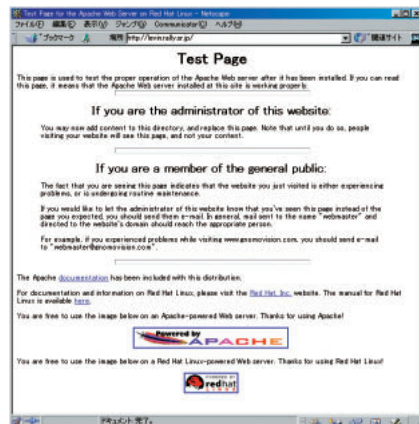
設定ファイルの中に、サーバの管理者を示すServerAdminという項目がある。初期設定では、root@localhostになっているので、環境に合わせて書き換えよう。

また、webmaster宛てにメールが来ることも考え、次のように設定してメールが自分に届くようにしておくのもよいだろう。

```
# vi /etc/aliases
```

以下のような行を追加する。

```
webmaster: <自分のログイン名>
```



画面13 Apacheのテストページ

実際のディレクトリ	アクセス時のディレクトリ	説明
/home/httpd/html	/	htmlファイルなどの置き場所
/home/httpd/icons	/icons/	icon画像ファイルの置き場所
/home/httpd/cgi-bin	/cgi-bin/	CGIファイルの置き場所
/home/foo/public_html	/foo/	ユーザーfooのhtmlファイルなどの置き場所
/usr/sbin/httpd		実行ファイル
/etc/httpd		設定ファイル
/var/log/httpd		ログ

表7 ディレクトリの初期設定（Red Hat Linux 6.2J）

ファイルを書き換えたら、newaliases
コマンドを実行して、変更したメール
のエイリアスを有効にしておく。

```
# newaliases
```

最初にしておきたいこと

ドキュメントを置くべき/home/httpd
以下のディレクトリはすべて、rootの
所有でパーミッション（属性）も
owner（root）のみが書き込めるよう
になっている。したがって、データ
を変更するには、rootになって作業す
ることになる。Webのデータをいじるた
めだけに、すべての権限を持つスーパ
ーユーザー（root）になるというのは、
かなり危険な行為といえる。そこで、
rootでなくても、Webデータを書き換
えられるようにしよう。

まず、Webデータの操作を目的とし
た、webadminグループを作る。そし
て、自分をそのグループのメンバーに
してしまおう。

```
# vi /etc/group
```

vigrコマンドを使うのもよい。以下の
ような行を追加する。

```
webadmin:x:1001:<自分のログイン名>
```

次に、/home/httpd以下のディレク
トリについて、グループをwebadmin
に変え、グループパーミッションも開
けておく。

```
# chgrp webadmin /home/httpd/*
# chmod g+w /home/httpd/*
```

これで、グループwebadminに所属す
る人はWebデータを自由に改変でき
ようになる。

```
$ ls -al ~/public_html
total 12
drwxrwxr-x  2 nor    nor    4096 Jun 27 20:52 .
drwx---  3 nor    nor    4096 Jun 27 20:55 ..
-rw-rw-r--  1 nor    nor    34 Jun 27 20:52 index.html
```

ここが-xでなければアクセスできない

画面14 nobodyがホームディレクトリ(..)にアクセスできない

ユーザー用ディレクトリ

初期設定では、ユーザー用のWeb用
ディレクトリはpublic_htmlのはずだ。
しかし、ホームディレクトリに
public_htmlというディレクトリを作
り、その中にドキュメントを置いたの
にまったくアクセスできないというこ
とがある。この場合、ディレクトリも
しくはファイルのパーミッションが違
っている可能性が高い。Apacheは、
初期設定ではnobodyの権限で動く。
このため、ユーザーのホームディレク
トリとpublic_htmlディレクトリ、さら
にファイルのパーミッションがnobody
に対して読めるようになっていないと、
アクセスすることはできない。

lsコマンドで確認してみよう（画面
14）。ホームディレクトリのパーミッ
ションがotherに対して実行不可になっ
ているので、実行属性をつける。

```
$ chmod o+x ~
```

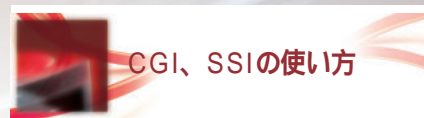
これで、index.htmlにアクセスでき
るようになったはずだ。

Apacheの再起動

Apacheの設定ファイルを書き換え
たら、次のようにしてApacheを再起
動する必要がある。

```
# /etc/rc.d/init.d/httpd restart
```

restartの代わりにstart、stopを用い
れば、起動、停止になる。



プロバイダが用意するWebサービス
では、自由にCGIやSSIが使えないこ
とが多い。そのため、自由にできるWeb
サーバを持った今こそ、CGIやSSIを使
いまくりたいという人は多いだろう。

CGIを使う

前述の通り、ApacheではCGIを使
うためのディレクトリとして/home/
httpd/cgi-binが用意されている。それ
以外のディレクトリでCGIを使うには、
設定ファイルhttpd.confにCGIとして
扱う拡張子を設定すればよい。

```
Addtype application/x-httpd-cgi .cgi
```

もしくは、以下のように設定する。

```
AddHandler cgi-script .cgi
```

ただし、これだけではCGIが使える
ようにはならない。さらにCGIプログ
ラムを置くディレクトリにおいて、ア
クセス制御によってCGIを有効にする
必要がある。これについてもhttpd.
confに設定する。

```
<Directory /home/* /public_html/
cgi-bin>
    Options ExecCGI
</Directory>
```


この設定では、各ユーザーのホームディレクトリにあるpublic_html/cgi-binディレクトリにあるCGIプログラムを動作させることができるようになる。

SSIを使う

現バージョンのApacheでは、/home/httpd/htmlディレクトリにおいて、SSIが有効になっている。ただし、ユーザー用のWeb用ディレクトリに関しては、SSIが有効になっていない。これを有効にするには以下のような記述をhttpd.confに追加する。

```
<Directory /home/*public_html>
    Options +Includes
</Directory>
```



ログファイルの分析

自分でWebサーバを構築した場合、当然すべてのアクセスログを手に入れることができる。これは貴重な財産だ。しかし、ログファイルをそのままがめても、全体は把握できない。そこで、分析ツールの出番となる。ここで紹介するのはanalogというツールだ。ログファイルから、月別、曜日別、時刻別、ファイル別などのアクセス分析をしてくれる。分析の結果はHTMLファイルに出力されるので、Webブラウザで見ることができる(画面15)。

analogのWebページ

<http://www.analog.cx/>



常時接続を利用したWebサーバの公開

OCNエコノミーなどでドメインを取得し、固定グローバルIPでWebサーバを公開するのが、一番楽な方法だろう。

ただ、そうでなくてもグローバルIPを持ってさえすれば、Webサーバの公開は可能だ。無料のWebページスペースにトップページを置き、自前サーバへリンクを貼っておけばよいのだ。巨大なデータや、バックグラウンドでデータベース処理をする必要があるページなどは、自分のサーバに置くといい使い方もある。

プロバイダなどのWebページスペースに、リスト10のようなHTMLファイルを用意すれば、特定のIPアドレスへ

ジャンプさせることができる。LinuxサーバのIPアドレスを監視し、変更があったらIPアドレス部分を書き換えたファイルを再度アップロードする。そういった処理をすべて自動的にすることも、Linuxの機能を使えば比較的簡単だろう。

それぞれの環境で工夫して、Webサーバを使ってほしい。最初はLAN内にサーバを設置して、自作したWebページのテストなどに活用するのもよいだろう。

The screenshot shows the 'Request Report' section of the analog web server statistics. It lists files with at least 20 requests, sorted by the number of requests. The data is as follows:

reqs	%bytes	last time	file
107300	9.56%	27/Jun/00 21:55	/
30900	10.41%	28/Jun/00 00:41	/links.html
29700	4.14%	27/Jun/00 21:20	/rallu/result/
25600	11.42%	27/Jun/00 16:25	/rallu/sch/
18000	3.52%	27/Jun/00 21:19	/whatsnew.html
15100	0.66%	27/Jun/00 20:04	/rallu/
13500	5.31%	27/Jun/00 18:49	/event.html
12100	0.62%	27/Jun/00 20:37	/jmcchiba/rallu/
10200	0.55%	27/Jun/00 21:32	/jmcchiba/rallu/result/
9600	7.33%	27/Jun/00 08:30	/guest.html
8700	4.61%	27/Jun/00 18:48	/jmcchiba/rallu/result/00acua.rbac/
7300	0.50%	27/Jun/00 18:21	/hor/
6900	3.19%	27/Jun/00 18:50	/rallu/result/00bc_hotstaff/
5700	2.13%	27/Jun/00 22:32	/rallu/result/00k_artas/
4700	2.51%	27/Jun/00 01:51	/kikurin/19881108/nazo/nazo.html
4700	2.10%	27/Jun/00 19:09	/dirt/sch/
4400	0.24%	21/Jun/00 00:20	/jmcchiba/rallu/point/
4300	2.05%	28/Jun/00 00:03	/rallu/result/99k_beng.html
4100	0.41%	13/Jun/00 22:25	/kikurin/20000604/20000604.html
4000	0.84%	26/Jun/00 22:46	/kikurin/19880517/19880517-2.html
4000	0.51%	26/Jun/00 00:45	/kikurin/19880517/19880517-1.html
3000	0.89%	27/Jun/00 18:48	/rallu/result/97necost.html

画面15 analogによるログ解析

リスト10 000.000.000.000というサイトにジャンプする

```
<html>
<meta http-equiv='refresh' content="0"; URL=http://000.000.000.000/">
</html>
```

FTPサーバ

wu-ftpd

FTPはインターネット上でのファイル転送の手段として古くから使われてきたアプリケーションだ。現在のようにWebが普及する前は、プログラムやデータ、ドキュメントもanonymous(匿名)FTPという手段で提供されるのが普通だったのだ。今までに、数多くのFTPサーバプログラムが作られてきたが、Linuxで標準的に使われているのがwu-ftpdだ。wu-ftpdは、BSD UNIXのソースをもとに、ワシントン大学セントルイス校で開発されたFTPサーバプログラムで、現在でも改良が続けられている。

FTPは過去のものか

Web(HTTP)でのファイル転送が可能で現在の読者の中には、FTPの存在意義を疑問視する方もいるかもしれない。

しかし、今なおFTPはファイル転送の手段として欠かせないものである。その理由のひとつに、FTPによるファイル転送の確実性があげられる。HTTPはその原理上、ファイル転送が正常に行われたかどうかを判断する手段がない。それに対し、FTPはファイル転送の失敗を検出できる。HTTPは、Webページのデータ転送を目的としているため、転送の確実性に重きをおいていないのだ。重要なファイルを転送する場合には、確実でない転送は無意味だといえるだろう。

また、Red Hat LinuxやTurboLinuxなどのディストリビューションそのものがFTPで提供されていることから、FTPが巨大なファイルを確実に転送するのに向いていることがわかるだ

ろう。Webページでもファイルの転送にはHTTPではなくFTPを使っているところも多いのだ。

このほかに、FTPではファイルを受け取るだけではなく、ファイルを送ることもできる。もちろん、HTTPでもそれなりのプログラムを使えばファイルを送ることはできるが、やはり安定性は低いし、ブラウザや環境によって転送に制限があることもある。

常時接続環境であれば、外出先から自宅のサーバにFTPでログインして、必要なファイルを取り出すことも可能だ。このように、FTPは、確実にファイル転送するための手段として確固たる地位を築いているのだ。

anonymous FTPとは

anonymous FTPとは読んで字のごとく、anonymous(匿名)で、すなわち、そのコンピュータにアカウントがないユーザーでもファイル転送を行える手段だ。

anonymous FTPサーバを利用する場合には、ログインIDとして、「anonymous」あるいは「ftp」を入力し、パスワードとして、自分の電子メールアドレスを入力することでログインできる。

anonymous FTPサーバでは、匿名ユーザーには特定のディレクトリ以下しか見せない。不特定多数に対して配布したいファイルをそれらのディレクトリに置いておけば、自分のマシン内のファイルをFTPで安全に公開することができるようになる。

仮に、ドメインを取得していなかったとしても、固定したグローバルIPア

ドレスを使うことができるならば、anonymous FTPサーバを設置して、ファイルを公開するのは有効だろう。

また、匿名ユーザーがサーバに対してファイルを転送するようにもできる。メールの添付などでは送りたくない巨大なファイルのやりとりに活用できるだろう。ただし、匿名ユーザーがサーバに送ったファイルは、匿名ユーザーからは見えないように設定しよう。さもないと、あなたのサーバが違法データの交換所にされてしまうかもしれない。

wu-ftpdの特徴

Red Hat LinuxやTurboLinuxでは、FTPサーバとしてwu-ftpdを採用している。wu-ftpdは世界中でもっとも使われているFTPサーバのひとつだ。安定性も高く、広く普及していることから関連情報も入手しやすい。さらに、anonymous FTPサーバとして使いやすいのも特色といえよう。

anonymous FTPサーバとして使う場合には、wu-ftpdの持つ次の機能が役に立つだろう。

- ・ログ記録
- ・アクセス制御
- ・メッセージ表示

また、ファイルを転送するとき、データ圧縮をしたり、tarによるアーカイビングをサポートしているのも嬉しい。この機能は、初期設定で有効になっている。たとえば、foo.barというファイルがあった場合に「get foo.bar.gz」と指定すれば、foo.barをgzipで圧縮したfoo.bar.gzというファイルを転送する。ディレクトリに対して「.tar」をつけて指定すれば、ディレクトリ全体

をtarでまとめて転送する。



wu-ftpは、Linuxのインストールと同時にインストールされるディストリビューションが多い。このようにしてインストールされたwu-ftpは、inetdにより、FTPの接続要求があったときに起動する設定になっている。

TurboLinux 6.0の場合は、標準のインストール状態では、/etc/inetd.confのftpの行がコメントアウトされているため、wu-ftpを利用できない。そこで、コメントを示す「#」を取る必要がある(リスト11)。TurboLinux 6.0では、セキュリティを確保するため、inetdから起動されるプログラムへのアクセスが禁止されている。このため、/etc/hosts.allow、/etc/hosts.denyというファイルも編集しなければならない。これらのファイルの内容については、本特集のPart 3で解説しているので、そちらを参照してほしい。設定が終わったら、次のように入力してinetd

を再起動する必要がある。

```
# /etc/rc.d/init.d/inet restart
```

さて、いよいよwu-ftpの設定を説明する。wu-ftpの設定ファイルは/etcディレクトリに置かれている。この中でも、重要と思われるファイルについて解説しよう。

ftppusers

/etc/ftppusersではFTPサーバへの接続を拒否したいユーザー名を指定する。初期設定では、一般的に危険だと思われるユーザー名が並んでいる(リスト12)。

ftppaccess

/etc/ftppaccessは、wu-ftpの動作の詳細を指定するファイルだ。ftppaccessの初期設定(リスト13)をもとに、各設定項目を説明するが、通常のプライベートなFTPサーバとして利用するなら、初期設定のままでも問題はないだろう。ただし、インターネットで公開する

FTPサーバとして不特定多数のアクセスがある場合には、これらの項目を変更追加したほうがよいかもしれない。

• class

利用者名と接続元ホスト名の組を定義する。同時接続数などは、ここで定義したクラス名で制御される。初期設定では、allというクラス名が定義してある。allは、ユーザーとして、アカウントのあるユーザー、ゲストユーザー、匿名ユーザーが含まれ、接続元ホスト名はすべてのホストになっている。

• email

このホストの管理者のメールアドレス。この項目は書き直すべきだろう。

• loginfails

何回ログインに失敗したら接続を切るかを指定する。初期設定の場合は、5回ログインに失敗したら接続を切る。

• readme

READMEファイルの情報の表示方法を指定する。初期設定では、ログインしたときとディレクトリを移ったときに、そのディレクトリのREADMEで始まるファイルの内容を表示する。

リスト11 /etc/inetd.confのftp行にあるコメントマーク(#)を削除

```
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
```

リスト12 ftppusersの初期設定

```
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

リスト13 ftppaccessの初期設定

```
class all real,guest,anonymous *
email root@localhost
loginfails 5
readme README* login
readme README* cwd=*
message /welcome.msg login
message .message cwd=*
compress yes all
tar yes all
chmod no guest,anonymous
delete no guest,anonymous
overwrite no guest,anonymous
rename no guest,anonymous
log transfers anonymous,real inbound,outbound
shutdown /etc/shutmsg
passwd-check rfc822 warn
```


- message

表示するメッセージファイルの名前を指定する。初期設定では、ログインしたときにはルートディレクトリのwelcome.msgを、ディレクトリを移ったときには、そのディレクトリにある.messageというファイルの内容を表示する。

- compress

ファイル転送時に圧縮を許可するかどうかを指定する。初期設定では、クラスがallのユーザーに対し圧縮転送を許可する。

- tar

ファイル転送時にtarでまとめて転送することを許可するかどうかを指定する。初期設定では、クラスがallのユーザーに対して許可する。

- chmod、delete、overwrite、rename

それぞれのコマンドについて許可するかどうかを指定する。初期設定では、ゲストと匿名ユーザーにはchmod、delete、overwrite、renameを許可しない。

- log

ログファイルの記録内容を指定する。初期設定では、匿名ユーザーとアカウントのあるユーザーのファイル転送内容を記録する。

- shutdown

ここに指定したファイルが存在する場合、ファイルの内容に従いFTPサービスを停止する。このファイルはftpshutコマンドを使用して作成することができる。

- passwd-check

anonymous FTPにおけるパスワードのチェック方法を指定する。初期設定の場合は、パスワードがメールアドレスとしてRFC822に準拠してなければ警告を出す。

このほかに、重要と思われる項目は以下のものだ。

- limit

最大利用ユーザー数や利用時刻の制限を行う。以下のように指定すれば、匿名ユーザーは10人までに制限され、それを超えた場合には/etc/ftp/msg.overというファイルの内容を表示して接続を切る。

```
limit anonymous 10 Any /etc/ftp/msg.over
```

- deny

アクセスを禁止するアドレスを指定する。以下のように指定すれば、210.160.90で始まるIPアドレスからの接続を拒否し、msg.denyというファイルの内容を表示する。

```
deny 210.160.90.* /etc/ftp/msg.deny
```

- upload

特定のディレクトリに対する匿名ユーザーの書き込みを制御する。以下のように指定すれば、匿名ユーザーは/home/ftp/incomingのみに書き込むことができる。そして、書き込んだファイルは所有者root、グループftpadmin、属性が0660となる。またincomingディレクトリでのサブディレクトリ作成を許可する。

```
upload /home/ftp/ * no
```

```
upload /home/ftp/ /incoming yes
root ftpadmin 0660 dirs
```

- ftpconversions

/etc/ftpconversionsでは、圧縮転送やtarによるアーカイブ転送で呼び出される圧縮プログラムなどの設定をする。ただし、使用するプログラムは、/home/ftp/binディレクトリに存在していなければならない。ファイルの記述内容は、決してわかりやすいとはいえないが、どのようなファイル名を指定すると、どのプログラムが使われるかは判断できるだろう(リスト14)。通常は変更しなくてよいだろう。

このほかにも、いくつかの設定ファイルや項目で、FTPサーバのきめ細かい制御をすることができる。興味がある方はぜひドキュメントを読んで、いろいろ試してほしい。

リスト14 ftpconversionsの初期設定

```
:.Z: : :/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
: : :.Z:/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
.gz: : :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
: : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
: : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
: : :.tar.Z:/bin/tar -c -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
: : :.tar.gz:/bin/tar -c -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
```

Part 3

常時接続時代のセキュリティ対策

Netfinity 1000

文：山岸典将

Text : Norimasa Yamagishi

「常時接続を導入すれば世界はバラ色」というところに水を差すようだが、1つキチンと考えておかなければいけない問題がある。それは、セキュリティ対策だ。

攻撃の種類には大きく分けて2つある。ホストに侵入し、その資源を盗用する「利用型」と、システムの停止を狙う「破壊型」だ。「たとえ攻撃を受けても、大事な情報なんてないし、それほど困らない」などと思っている人も多いだろう。しかし、そんな気楽なことをいってられないのが現実だ。資源を利用するというのは、内部のデータを利用するだけではない。そのコンピュータを使って、ほかのコンピュータに攻撃をかけるというのがクラッカーの手口だ。このようなパターンを俗に「踏み台に使う」という。マシンが踏み台にされてしまったら、あなた自身が加害者として扱われてしまうかもしれないのだ。自分の身は自分で守るしかない。

セキュリティ対策の基本

セキュリティ対策としてはどのようなものがあるのだろうか。もちろん、本来はキチンとしたファイアウォールを立ち上げて運用するのが望ましい。しかし、現実には、個人や、小規模なネットワークでそこまでやるのは難しい。できる範囲でベストな対策を講じよう。

基本的なセキュリティ対策のポイントは次の3点だ。

1. 入口を締める
2. 監視を怠らない
3. 情報を集める



入口を締める

まず、侵入可能な経路をできる限り少なくしよう。外部との接点となっている部分はなんだろうか。TAやモデムを使っている場合には、それらがつながっているマシン、ルータを使っている場合にはルータそのものだ。

侵入経路を少なくするには、パケット（ネットワーク上を流れる信号）の種類によるフィルタリングを行い、外部から入ってくるパケットをできる限り少なくすることが大切だ。

ルータでのセキュリティ

ルータでインターネットに接続している場合、ルータの持つIPマスカレード機能を利用すれば、LAN内のネットワークを丸ごと外部ネットワークから隠すことができる。もっとも、これをするるとLANの中でサーバを立ち上げて、そのままでは外部から参照できない。

LAN内のサーバを外部に公開する場合は、まずルータでパケットフィルタリングを行おう。最近のルータであれば、パケットフィルタ機能を持っているはずだ。ここで、必要ないパケットを制限してしまえば、不要なパケットが入ってくることはないので、攻撃はある程度せき止めることができるはずだ（画面1）。また、公開するサーバ自

身でもパケットフィルタリングをすることが望ましい。

ゲートウェイマシンのセキュリティ

インターネットとLANの間にLinuxマシンを設置し、IPマスカレードを行っている場合など、Linuxマシンをゲートウェイにしているならパケットフィルタリングは必須だ。IPマスカレードを利用すれば、内部のLANは外部から参照できなくなるが、ゲートウェイであるLinuxマシンは外部にさらされている。したがって、このマシン自身のセキュリティが非常に重要になる。

不要なサービスは動かさない

不要なサービスはすべて止めてしまおう。使ってもいないサービスのセキュリティホールを突かれて、システムに侵入されてしまうほどバカバカしいことはない。

個人でLinuxを使っている場合は、1台のマシンでX Window Systemを使い、すべてのサーバを兼ね、インターネットに直接接続している場合もあるだろう。そのような場合でも、実際に動いているサービスを調べ、本当に必要なサービス以外はひとつひとつ止めていくといった心掛けが大事だ。セキュリティを高めるという作業は、こつこつと穴を塞いでいく地道な作業なのだ。

パケットフィルタリング

Linuxカーネル 2.2では、ipchainsというコマンドでパケットフィルタリングの設定を行う。IPマスカレードの設定

でも利用するコマンドだ。ipchainsは、送信元や発信先のIPアドレスや、プロトコルの種類、利用するインターフェイスなどによって、パケットのフィルタを細かく制御できる。

では、フィルタリングの設定をしよう。ここでは、LANで192.168.1.0 / 24 (192.168.1.0 ~ 192.168.1.255) というプライベートアドレスを利用しており、Linuxマシンのネットワークインターフェイスeth0がインターネットにつながっているものとして話を進める。

最低限、設定しておきたいのは、LAN内部から発信したように見せかけて、外部から侵入してくるパケットの遮断だ。このあと解説するtcp_wrappersというプログラムでは、接続元のIPアドレスをもとにアクセスの制御を行う。これを使って、LAN内のIPアドレスからのアクセスだけを許可していても、そのIPアドレスを詐称して外部からアクセスされては意味がない。そこで、rootユーザーになって、リスト1のように入力しよう。これは、ソースアドレスに192.168.1.0 ~ 192.168.1.255というIPアドレスを持ち、インターフェイスeth0から入ってくるパケットを拒否するという設定になる。この例をもとに、各スイッチを解説しよう。

- A input
ipchainsには、input、output、forwardという3つのチェーンがあり、それぞれパケットを受け取ったとき、送信するとき、転送するときにチェックされる。初期設定では、これらのチェーンには何も設定されていないので、パケットのチェックはまったく行われない。リスト1の - A inputは、input、すなわち受け取ったパケットに対するチェーンにルールを追加 (Add) するという意味だ。

- i eth0
チェックの対象となるインターフェイスは、eth0だという指定。TAやモデムでインターネットにつないでいるならppp0などになる。

- s 192.168.1.0/24
発信元 (ソース) IPアドレスが192.168.1.0 ~ 192.168.1.255であるパケットを対象とする。IPアドレスだけではなく、ポート番号も指定できる。192.168.1.0/24 137:139と書けば、このIPアドレスでポート番号が137 ~ 139のパケットが対象となる。

- d 0/0
送信先 (デスティネーション) IPアドレスは指定しない。これで、送信先はすべてのIPアドレスが対象となる。

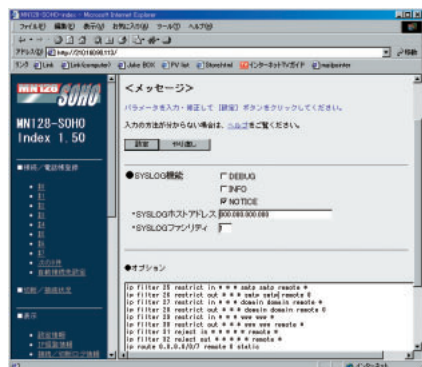
- j DENY
設定した条件に合うパケットを拒否することを意味する。DENYのほかに、REJECT、ACCEPTなどが設定できる。詳しくはmanを参照のこと。

このほか、- pオプションでプロトコルを指定することも可能だ。各チェーンには、- Aオプションで複数のルールを追加することができる。各チェーンに設定されているルールは、ipchains - Lを実行して確認することができる (画面2)。

サービスへのアクセス制御
サーバが提供するサービスには、必要なマシン以外はアクセスできないようにしたい。inetdから起動されるサービスへのアクセス制御をするのがtcp_wrappersだ。

tcp_wrappersの本体はtcpdというプログラムだ。あるサービスへの要求が来ると、まずtcpdが起動される。tcpdは、設定ファイル/etc/hosts.allow、/etc/hosts.denyを順に調べ、設定内容にしたがってサービス要求に応えるかどうかを決定する。/etc/hosts.allowには、接続を許可するホストを、/etc/hosts.denyには拒否するホストをそれぞれ記述する。

tcp_wrappersはLinuxのほとんどのディストリビューションにおいて、標



画面1 ルータでのパケットフィルタリング設定

```
リスト1 内部のIPアドレスを詐称する外部パケットを遮断する

# ipchains -A input -i eth0 -s 192.168.1.0/24 -d 0/0 -j DENY

# ipchains -L
Chain input (policy ACCEPT):
target prot opt source destination ports
DENY all ----- 192.168.1.0/24 anywhere n/a
Chain forward (policy DENY):
Chain output (policy ACCEPT):
```

画面2 パケットフィルタリングの設定を確認

準で利用されている。その初期設定はディストリビューションによって異なり、TurboLinux 6.0では、すべてのサービスに対するアクセスを禁止している。そのため、TurboLinux 6.0をインストールした直後の状態では、inetdで起動するサービス (telnet、ftp、smtp など) はほかのホストから利用できない (リスト2)。

/etc/hosts.allow と /etc/hosts.deny の書式は次の通り。

```
< service list > : < host list >
< : command >
```

< service list > は、ルールを適用するサーバ名だ。カンマで区切って複数のサーバを指定することもできる。また、ALL はすべてのサーバを指定する。< host list > は、ホスト名の指定だ。カンマで区切り、複数のホストを指定することもできるが、192.168.1 とすれば、192.168.1.0 ~ 192.168.1.255 という IP アドレスのホストを指定したことになる。また、.ascii.co.jp として、

ascii.co.jp ドメインのホストすべてを指定することもできる。ALL は、すべてのホストを意味する。command は省略可能なオプションで、条件に合ったホストからのアクセス時に実行するコマンドをフルパスで指定する。

運用の際には、/etc/hosts.deny に ALL:ALL を指定しておき、接続を許可するサービスとホストだけを /etc/hosts.allow に追加するのがよいだろう。たとえば、/etc/hosts.allow に、

```
ALL : 127.0.0.1, 192.168.1
```

と記述しておけば、Linux サーバ自身 (127.0.0.1) と、IP アドレスが 192.168.1.0 ~ 192.168.1.255 のホストからのみすべてのサービスを利用できる。

前述のとおり、tcp_wrappers は ipchains によるパケットフィルタリングと組み合わせて使うことを強くお勧めする。

弱点を調べる

とりあえず思いつく対策は行った。

リスト2 TurboLinuxのtcp_wrappers制御ファイル

```
$ cat /etc/hosts.allow
#
# hosts.allow This file describes the names of the hosts which are
#             allowed to use the local INET services, as decided
#             by the '/usr/sbin/tcpd' server.
#
# See man hosts_access(5) for more information

ALL : 127.0.0.1 ————— すべてのプロトコルに対して、127.0.0.1からの接続を許可する

$ cat /etc/hosts.deny
#
# hosts.deny This file describes the names of the hosts which are
#            *not* allowed to use the local INET services, as
#            decided
#            by the '/usr/sbin/tcpd' server.
#
# See man hosts_access(5) for more information.

ALL : ALL ————— すべてのプロトコルに対して、すべてからの接続を禁止する
```

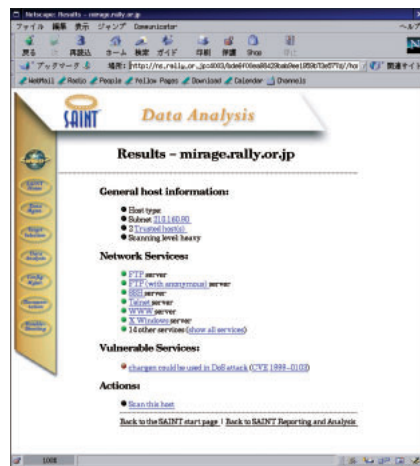
ローカルホスト以外のすべてのホストへのサービスを禁止している

しかし、人間は間違いをおかす動物だ。そこで、自分のサイトの弱点を調べるためのツールを使おう。自分自身に対してポートスキャンをかけたり、攻撃可能なセキュリティホールがないかを調査し、結果をレポートするツールだ。このようなツールで最初にリリースされたのはSATANというプログラムで、ほかのホストの弱点を調べることもできるため、一時は賛否両論があったが、現在ではいくつかの弱点検査ツールが存在する (画面3)。

これらのツールで、自分のLinuxマシンにセキュリティ上の弱点がないか確認しておこう。いうまでもないことだが、間違ってもこのようなツールでよそのホストを調査しないこと。攻撃と受け取られることは間違いない。



ログの監視は、トラブルを予防するための基本だ。すでに侵入されてしまっている場合、クラッカーはその痕跡を消そうとするため、ログにはなにも残っていないかもしれない。しかし、侵入の前段階では、侵入を試みた様子がログに残っていることも多い。また、侵入以外のトラブルについても、ログ



画面3 Perlで動作する弱点検査ツール SAINT (<http://www.wwdsi.com/saint/>)

記録されることがあるので、各種ログの監視は怠らないようにしたい(表1)。

もっとも、すべてのログを入念に監視することは、時間的にも困難なことが多い。そのような場合には、ログの監視をするツールを使おう。そのようなツールとして有名なものにSwatchがある(ftp://ftp.stanford.edu/general/security-tools/swatch/)。Swatchは、Perlで書かれたログ監視ツールで、設定された文字列がログに現れると、メ

ールを送ったり、特定のプログラムを実行したりできる。決して完璧なもの

ではないが、警報装置としては、かなり役に立つだろう。

ファイル	説明
/var/log/lastlog	ユーザーが最後にログインした時刻のログ
/var/run/utmp	ログインしているユーザーのログ
wtmp	ログインの記録を保持する
boot.log	起動時のメッセージ
messages	メールを除くすべてのログ
spooler	newsやUUCPのログ
secure	認証関連のログ
cron	cronのログ
maillog	sendmailのログ
xftplg	FTPのログ

表1 チェックすべきログ

情報を集める

セキュリティには完璧という言葉はない。どのような対策をしたとしても、破られる可能性はなくなる。そのため、常に最新の情報を収集し、あらたな脅威に備えることが重要だ。セキュリティに関する情報は、Webページやメーリングリストで得るのが一番早いだろう。

国内外でセキュリティ関連の情報を扱う著名なサイトを紹介する。

・国内

JPCERT/CC (画面4)

<http://www.jpCERT.or.jp/>

IPAセキュリティセンターホームページ

<http://www.ipa.go.jp/security/>

NetSecurity

<https://www.netsecurity.ne.jp/>

・海外

CERT

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

SecurityFocus (画面5)

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

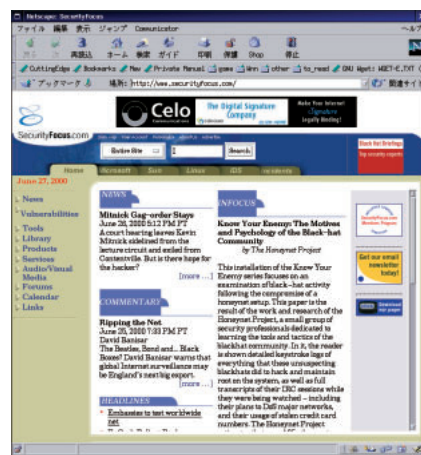
万一、被害にあってしまったら、JPCERT/CCなどを通じて情報を公開することをぜひお願いしたい。情報の共有によって対策が進み、被害は広がらずにすむはずだ。

ディストリビューターの情報

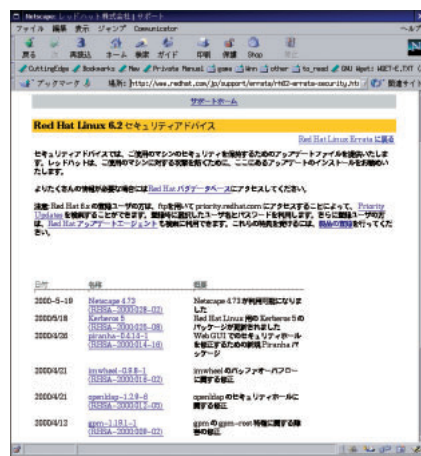
ディストリビューターは、各種の情報とともに、リリース後に判明した問題を修正するアップデートパッケージを配布している。定期的にWebサイトを訪れて、チェックを怠らないようにしてほしい(画面6)。



画面4 JPCERT/CC 不正アクセスに関する情報を受け付けている。



画面5 SecurityFocus 豊富なニュースとツールライブラリがある。



画面6 レッドハットのセキュリティアドバイスのページ



家じゅうどこでもLinux 無線LANで送る ワイヤレスな日々

文: にゃー@編集部
Text: Nyaa@Linux magazine
Photo: Shuichi Mito (Dee)

6月号の自作PC特集で激安PCを完成させるという任務を完遂(?)したあと、立派なLinuxerになるべく日々精進していた私のもとに、あらたなる指令が届けられた。

「最近結構、無線LANも安くなってきたし、802.11bに対応した11Mbpsモデルが出揃ってきたんで、Linuxで動くかどうか調べといて。あんまお金かけられないんで、機材はメーカーさんから借りてね。」

以上が指令の概要である。こうして、「ワイヤレスな」生活が始まったのであった。おりしもNETWORLD + INTEROP 2000 TOKYOが開催されており、各メーカー様に貸出しをお願いするには好都合である。いざ行かん幕張へ!

⚡ 機器を調達するの巻

と、はりきっていたのではあるが、初日に寝坊するは、2日目は健康診断だは、3日目は強風で京葉線が止まっちゃうはで、結局行けなかったのであった。そうこうしているうちに、お目付け役兼渉外担当である副編集長から「メルコさんとアイコムさんにはお願いしといたから、あとはコレガさんとブラネックスさんに連絡してね」と言われ、なんだかんだしているうちに表1の4機種が揃ったのであった(各機種については137ページを参照)。

機材が揃ったところで、それぞれのマニュアルに目を通す。接続形態として、2つのモードがあるようだ。

インフラストラクチャモード
「アクセスポイント」と呼ばれる基地局を介して通信する方式で、アクセスポイントをハブにつなげば有線ネットワークとの接続も簡単に行える。

アドホックモード
無線LANカードを装着したPC間で、ピアツーピア接続を行うモード。PC間でデータのやり取りをするだけなら、無線LANカードがPCの台数分だけあればよい。

悲しいことに、編集部にころがっていたノートPCは1台だけであった(Panasonic Let's note AN-L1。型番はAN-L1T515J5。ちと古め)。PCカードスロットのあるデスクトップマシンもない。しかたがないので、ひとまずインフラストラクチャモードをテストしてみることにした(なんだか、行きあたりばったりだな)。

⚡ アクセスポイントを設定するの巻

Webブラウザを使って設定できて、設定項目も少ない(マニュアルで確認済)ことから、まずはコレガのAP-11を設置することにした。決して楽をしようというのではなく、今回はあくまで「無線LANカードがLinuxで動作するのか?」がメインテーマなので、ヨ、と言い訳しているうちに設定終了。

ハッキリいって設定は簡単であった。アクセスポイントを既存のLAN環境にケーブル接続。デフォルトで「192.168.0.230」というIPアドレスが設定されているので、任意のクライアントから

タイプ	製品名	メーカー
PCMCIA LANカード	WAVEMASTER SL-1100	アイコム
	Wireless LAN PCC-11	コレガ
	GeoWave GW-NS110	ブラネックス
	AIRCONNECT WLI-PCM-L11	メルコ
アクセスポイント	WAVEMASTER AP-3	アイコム
	Wireless LAN AP-11	コレガ
	AirStation WLA-L11	メルコ

表1 今回のテストに使用した11Mbps無線LAN機器

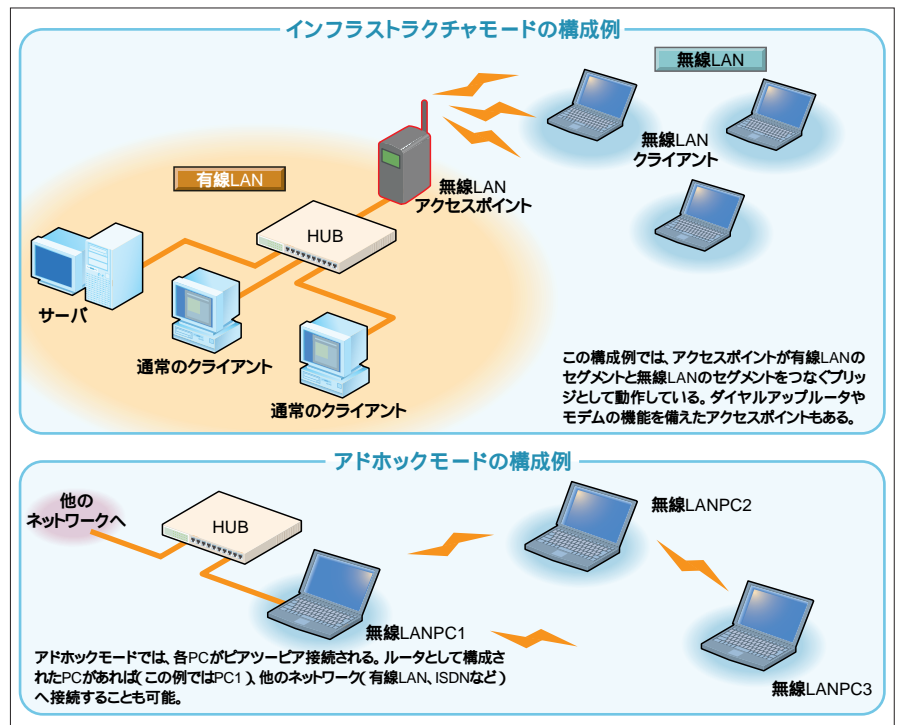


図1 無線LANの接続形態



画面1 AP-11の設定画面
必要な設定はこの画面だけで行える。機能はシンプルだが、設定がラク。

Webブラウザを使ってアクセス。そして、これまたデフォルトで設定されているパスワードを使ってrootユーザーとしてログインすると、あら不思議。設定用のWebページがオープンされるのであった。どうやらというか、当然そうだろうというか、アクセスポイントなるものはWebサーバとして動作しているのであった。

画面1の設定項目のうち、[SSID] はアクセスポイントと接続される無線LANクライアントとの間でのみ有効なグループIDの指定だ(ESSIDという場合もある)。異なるSSIDが設定されたクライアントは、そのアクセスポイントに接続できないようになっている。いわば、もっとも基本的なセキュリティ機構である。[IP設定] の各項目は、



写真1 コレガ Wireless LAN AP-11

ネットワークの構成に合わせて適宜設定すればいい。AP-11はDHCPのクライアント機能を持たないので、この画面での設定が必須である。

[WEP]はWired Equivalent Privacyの略で、IEEE802.11で規定されているオプションの暗号化仕様。この設定画面では、適当なキーワードを入力して[キーの生成] ボタンをクリックするとキーコードが自動生成されるようになっている。指定した暗号化キーは、接続するクライアントのすべてに設定する必要がある。ただし、ここでもやはり「まずは接続することが重要」と言い訳しつつ、設定を省略した私ではあった。

⚡ 大問題なり！の巻

アクセスポイントの設定を終えて一息ついたところで、今回最大級ともいえる災厄が振りかかった(やや誇張)。

ぬわんと、コレガの無線LANカードPCC-11がAL-N1のPCカードスロットに挿さらないのである。どうもコネクタの切り欠きの部分がひっかかっているようなのだ。「ぬおおお」と強引にはめてみたが、なにやらしっくりこなうえに、抜いてみると切り欠きの部分がチョッピリ削れているではないか！(コレガさんごめんなさい)。

まさかこのような事態に陥るとは、考えてみれば、PCカードについてあまりに無知であった。しかし嘆いてばかりも



写真2 コレガ Wireless LAN PCC-11
省電力設計も装着できなければ意味がない。スロットの確認を忘れずに！

いられないので、雑誌、書籍、Webサイト、ほかの編集部員などなど、あらゆる情報源を駆使して情報を収集したところ、以下のような事実が判明した。

形状は同じでも、コネクタの仕様が異なる2種類のPCカードが存在する。ひとつは16ビット幅のバスを使用するタイプで、「16ビットPCカード」と呼ばれることもある。

もうひとつは「CardBus」と呼ばれる規格をサポートしたタイプで、32ビット幅のバスを持つ。

CardBusを規定している「PC Card Standard」では、より電力消費の少ない3.3V仕様も規定されていてゲー。

どうやら5Vと3.3Vの仕様の違いが原因であるようだ。確かに、PCC-11の製品パッケージには「PCMCIA 3.3V仕様」とハッキリ印刷されている。3.3V仕様のカードは、CardBus対応のスロットにしか装着できないのだ。AL-N1は少し古いモデルのノートPCなのでCardBusに未対応だったんである。うかつであった。これで、ほかのカードもダメだとシャレにならない。あわてて試してみることにした……。

メルコのWLI-PCM-L11とプラネックスのGW-NS110はOKであった(アイコムはSL-1100は挿さらず)。ふーやれやれ。ひと安心である。ひと安心ついでに、よくよく見ると、装着できる



写真3 メルコ WLI-PCM-L11
こちらは5V仕様になっている。古いノートPCで使えるのはこのタイプ。

ものできないものでは切り欠きの部分の高さがずいぶん違う(写真4)。購入時には手持ちのPCのカードスロットを忘れずに確認しておこう。

カードサービスを ゲットするの巻

スロットに装着するだけでずいぶん手間取ったが、(CardBus対応のノートPCは別途入手するとして)無線LANカードの設定に進もう。

と思ったのだが、PCカードに無知であることを自覚したので、無線LANの設定をする前に、ここは謙虚にLinuxでのPCカードのサポート状況を調べることにした。

LinuxのPCMCIAカードサービス(pcmcia-cs)の配布元であるLinux PCMCIA Information Page(<http://pcmcia.sourceforge.org/>)に、最新のサポート状況がアップされている。バージョンごとの変更点(changes)を見ると、バージョン3.3.9で無線LANのドライバモジュールが追加され、以降バージョン3.1.12、3.1.14でも関連する変更が加えられているようだ(推定。英語は苦手)。6月7日付けの最新のサポート状況も確認してみたところ(supported cards)、4機種のうちメルコのWLI-PCM-L11のみが記載されていた。

AL-N1にインストールしたVine Linux 2.0では、PCMCIAカードサービスのバージョンは3.1.10である。サポート状況を見るとやはり新しいバージョンのほうがよさそう、つーことで、最新のバージョン3.1.16をダウンロードしておく(FTPサイトのURLは<ftp://pcmcia-cs.sourceforge.net/pub/pcmcia-cs/>)。

PCMCIAカードサービスはソースコードでしか配布されていない。バイナリパッケージ野郎である私などは、思

わず「バイナリで配布してよ～」とお願いしたくなったが、このサービスで使用されるモジュール類はカーネルの設定に依存する部分があるため、ディストリビューションや環境によってはソースからコンパイルしないと正しく動作しないらしい。コンパイル時にはカーネルモジュールが作成されるため、カーネルのソースコードも必須である。このあたりの詳しい情報については『Linux PCMCIA HOWTO』を参照しよう(JFによる日本語訳が<http://www.linux.or.jp/JF/JFdocs/PCMCIA-HOWTO.html/>にある)。

というわけでコンパイルである。なにもビビることはない。ガーっと気合を入れてやればきっと大丈夫だ。まずはカーネルのソースとコンパイラがインストールされているかを確認。カーネルソースは通常、/usr/src/linux以下にインストールされている。ない場合は、rpmコマンドなどを使ってインストールする必要がある。

```
# rpm -ihv Kernel-headers-2.2.14-1v16.i386.rpm
```

```
# rpm -ihv Kernel-source-2.2.14-1v16.i386.rpm
```

コンパイラに関しても同様で、glibc-devel、make、cpp、egcs、egcs-c++

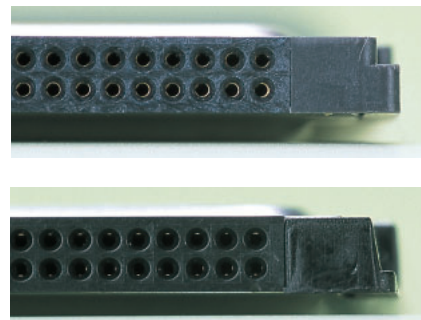


写真4 3.3V仕様と5V仕様では切り欠きの高さが違う3.3V仕様(上)のほうが高くなっており、5V仕様のスロットには装着できないようになっているのだ。

などのパッケージがインストールされているか確認しておこう。

次に、任意のディレクトリ(/usr/srcなど)でPCMCIAカードサービスのソースファイルを展開。

```
# tar -xzf pcmcia-cs-3.1.16.tar.gz
```

このコマンドを実行したディレクトリにpcmcia-cs-3.1.16ディレクトリが作成される。続いて、以下のコマンドラインを実行してコンパイルする。

```
# cd pcmcia-cs-3.1.16
```

```
# ./Configure
```

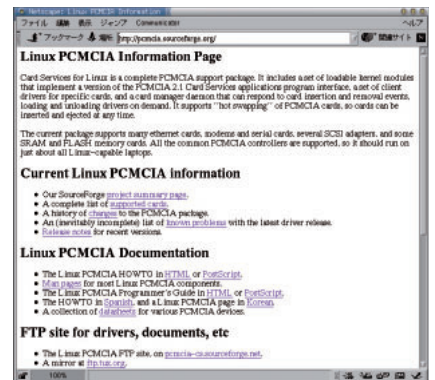
```
# make all
```

```
# make install
```

“./Configure”の実行時にいくつかオプションを聞かれたが、なんとなく勘で応えておいた。やや不安を感じつつ次のステップに移ろう(その後の調査ではデフォルトのオプションでも問題なく動作した)。なお、PCMCIAカードサービスがきちんとバージョンアップされているか不安なときは、

```
# cardmgr -V
```

とすれば確認できる。“version 3.1.16”と表示されればマル。



画面2 Linux PCMCIA Information

やっとな無線LANの設定ですよの巻

ようやく無線LANカードの設定にこぎつけたな。まずは、さきほどサポートが確認されたメルコのWLI-PCM-L11を使ってテストを開始しよう。

とりあえず何も設定していない状態で電源を投入。ここでまたもや不測の事態が！ PCMCIAサービスが起動するタイミングで「ズラズララ～」と大量のエラーが表示され、あげくには“Kernal Panic”などという不穏なメッセージとともに停止してしまったではないか！ “Killing interrupt Handler...”と表示されているので、IRQの競合が原因なのだろう。前出の『Linux PCMCIA HOWTO』によると「使っていそうなシステムリソースはカードサービスでは使用しない」はずなのに……なぜだ。

そのままリブートすると再びカーネル君がパニックを起こすので、インタラクティブモードで起動して(ブートメッセージに“Press to enter 'I' Interactive Startup”と表示されたところで“I”を入力)PCMCIAサービスをスキップ

する。PCMCIAサービスが使用するIRQ、I/Oポート、メモリセグメントなどの設定ファイルは、/etc/pcmciaにあるconfig.optsだ。AL-N1のマニュアルがなかったので、動物的勘を頼りに、IRQの使用を制限する“exclude irq 5”オプションを追加した。

でもってPCMCIAサービスを起動。起動スクリプトは/etc/rc.d/init.d/pcmciaである。このスクリプトにstartオプションを付けて実行する。

```
# /etc/rc.d/init.d/pcmcia start
```

今度はサービスの起動に合わせて「ピッ、ピッ」とビープ音がして、無事カードが認識された。カードの状態はリスト1の各コマンドで確認できる。

catで表示しているstabファイルはディストリビューションによっては違うパス(“lib”ではなく“state”)にあるので注意。このファイルには使用中のドライバやデバイスの名前などが記録されている。cardctlはPCMCIAサービスのユーティリティで、カードの構成(configオプション)や認識情報(identオプション)の確認が可能だ。

stabの内容からわかるように、無線LANカードのドライバ(wvlan_cs)がバインドされているデバイスは“wvlan0”である。ifconfigコマンドを実行してみたが、wvlan0は動作していない。ifconfigコマンドにオプションを付けてこのデバイスのインターフェイスを起動してみる。

```
# ifconfig wlan0 up
```

ここで、もう一度オプションなしでifconfigを実行したところ、wvlan0はアクティブになっているがIPアドレスがゲットできていない。Red Hat系ディストリビューションではpumpコマンドを使ってDHCPによるIPアドレスの割り当てを行う。やってみよう。

```
# pump -i wlan0
```

またまたifconfigを実行したところ、めでたくIPアドレスが割り当てられた。pingでの確認でも問題なし(ワイ)。うれしくなってきたので、AL-N1を持ち歩いて接続状況を確認してみることにしようではないか！

どこまで電波なんだろうの巻

ここまで読んで、違和感を覚えたあなたはエライ(私ごときにほめられて



写真5 メルコ Air Station WLA-L11

リスト1 無線LANカードの認識

```
# cat /var/lib/pcmcia/stab
Socket 0: MELCO WLI-PCM-L11
0      network wlan_cs      0      wlan0
Socket 1: empty

# cardctl config 0
Vcc 5.0V Vpp1 0.0V Vpp2 0.0V
interface type is "memory and I/O"
irq 3 [exclusive] [level]
function 0:
  config base 0x03e0
  option 0x41
  io 0x0800-0x083f [16bit]

# cardctl ident 0
product info: "MELCO", "WLI-PCM-L11", "Version 01.01", ""
manfid: 0x0156, 0x0002
function: 6 (network)
```

も別に嬉しくもないだろうけど)。そう、何を隠そうこの段階ではコレガのアクセスポイントにメルコの無線LANカードという組み合わせで動作しているのである。IEEE802.11bという同じ規格なんだから、あたりまえといえばあたりまえなんだけど、メーカーさんではサポートしてくれないだろうし、結構微妙なところであったりする。

まあ、普通の10BASEや100BASEのネットワークカードで「ハブのメーカーが違うので動作しません」ということはないの、無線LANでも相互運用性はしっかり確保されていくはずだ。

少し話がそれたが、接続限界テストに移ろう。テストに際して、アクセスポイントをメルコのWLA-L11に交換した。もちろん接続性に問題はない。

図2に示すとおりLinux magazine編集部のあるフロアは結構広い。図中のaの付近が当編集部である。a地点はマイシートなので、すでに接続は確認済みだ。pingを打ちながら、b地点から

順にウロウロと歩き回って接続確認を行った。

bは間仕切りでパーティション分けされたスペース。ここもOk。cは打ち合わせ用のスペースで、やはりパーティション分けされている。ここでつながれば、打ち合わせ中にこっそりWebサイトを見ることだって可能になる。「おお～」つながっているではないか。よしよし。fはトイレ。無事つながったが、個室に入って扉を閉めるとさすがに通信が切れた。むむ「ウンチョスをしながらWebサーフィン」という野望がついえたな。残念である。

結局つながらなかったのは、f地点のトイレの個室の中と、j地点のみであった。h地点はやや通信状況が不安定で、つながったりつながらなかったりといった感じだ。ここは個人用のロッカーとして使っている棚がいくつも並んでいるので、電波状況があまりよくないのだろう。i地点では問題なくつながったことから、距離ではなく遮蔽物が原

因のようだ。

それにしても、結構電波が届くものである。特にd、e、f地点は、オフィスとはドア以外完全に壁で区切られていて、アクセスポイントとの間に遮蔽物も多い。この状況で通信できるのであれば、オフィス環境での使用もまったく問題ない(んじゃない?)。今回は実験できなかったが、これなら普通の住宅も大丈夫なはず。少し気になるのは階が違う場合くらいだろうか。

11Mbpsの無線LANでは、電波状況に応じて伝送速度が1Mbps、2Mbps、5.5Mbps、11Mbpsの各モードで自動的に切り替わるらしい。そこで、実際に伝送速度が切り替わっているのかも確認してみることにした。チェックにはiwconfigというツールが必要だ。このツールは、前出のLinux PCMCIA Information Pageからダウンロードできる(ftp://pcmcia-cs.sourceforge.net/pub/pcmcia-cs/cntrib/)。今回は最新のバージョン20を使用した。

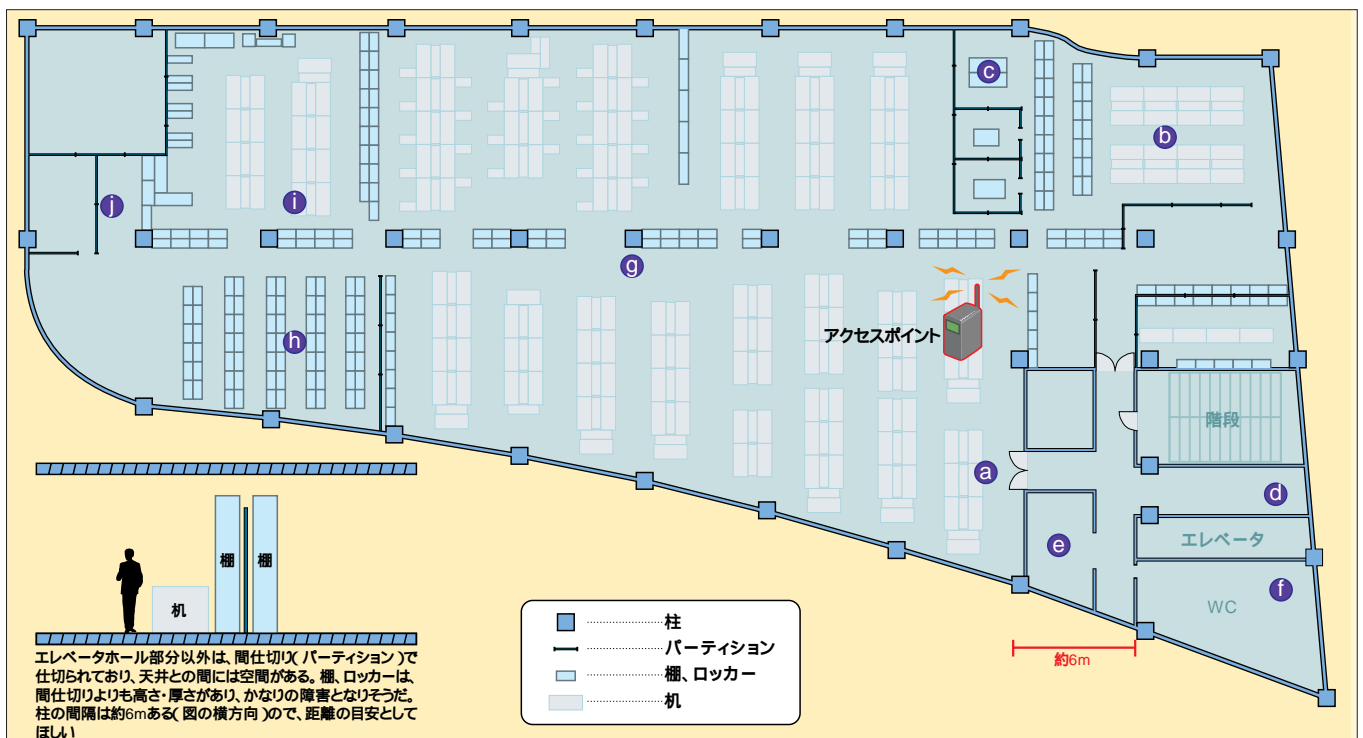


図2 接続テストを行ったオフィスのレイアウト (アスキー本社。初公開?)

リスト2はiwconfigの実行例だ。このようにiwconfigは、さまざまな通信状況の確認が行える優れたものだ。そこで、伝送速度のモードだけでなく、リンクの品質（Link quality）と電波の強さ（Signal level）もチェックしてみることにした。

図2のうち4つの地点をピックアップしてテストした結果が表2である。距離、遮蔽物などによる通信状況に応じて伝送速度が確かに切り替わっていることがわかる（複数回iwconfigを実行し、途中でモードが切り替わった場合は両方を併記してある）。テストを行っていくうち、アンテナの向きによってもモードが切り替わることに気づいた。アンテナに指向性があるようだ。g地点で、アンテナをアクセスポイントに向けた場合と、逆向きの場合の両方を確認した結果もあわせて表2に示す。

⚡ アドホックな関係の巻

数日間、うれしげにワイヤレスっているうちにアドホックモードの実験を行える環境が整った。今月号の特集2担当から私物のノートPC（Compaq ARMADA 4130T）を強奪したのである。すぐさまRed Hatの6.2Jをインストール。さっそくテスト開始だ。

このマシンもやはりチト古めであるため、CardBus対応のロットを持っていない。強奪しといてなんだが「チエツ」て感じた。プラネックスのGW-NS110は装着できるので、こちらの動作確認を兼ねてアドホックモードを試すことにしよう。

GW-N110は、PCMCIAカードサービスで標準サポートされていないので設定が必要になる。まず、/etc/

pcmcia/にあるconfigファイルとconfig.optsファイルにリスト3の内容を追加。configの設定内容はカードの認識情報で、“cardctl ident”で確認できる。config.optsのほうに追加したのは、アドホックモードに関する設定だ。“port_type=3”がアドホックモードで動作することを示している。“channel=1”は通信チャンネル、“network_name=TEST_LAN”はESSIDの指定である。

次に、同じディレクトリにあるnetwork.optsファイルにネットワーク情報を設定する。今回のテストでは、2台のマシンをアドホックモードで動かすので、IPアドレス、サブネットマスクなどを指定し、DHCPなどのIPアドレスの自動取得を無効にした（リスト4）。

PCMCIAサービスを再起動したところ、無事にカードが認識され、各ネットワーク情報も設定されているのだが、iwconfigで確認すると2Mbpsモードで動作していると表示された。これではせっかくの性能が台無しなので、config.optsに“transmit_rate=5”オプションを追加した。これで11Mbpsモードで動作するはずだ。iwconfigで確認。“Bit Rate=11Mb/s”と表示される。グッドだ。

1台だけではさびしいだろうから、AL-N1のWLI-PCM-L11も設定してあげよう。こちらは、カードの設定は完了しているので、config.optsにアドホックモードの指定を追加するだけでいい。AL-N1のPCMCIAサービスを再起動してpingしてみる。成功。順調であるな。こちらはデフォルトの設定で、11Mbpsモードで動作した。

アクセスポイントを使ったインフラストラクチャモードと同じく、アドホックモードでも接続状況の確認を行った。図2のアクセスポイントの位置に

地点	伝送モード	Link quality	Signal level
a	11Mbps	21/92	-65dBm
b	11Mbps/5.5Mbps	15/92	-73dBm
e	5.5Mbps/2Mbps	14/92	-75dBm
g	5.5Mbps/2Mbps	8/92	-82dBm
g*	11Mbps/5.5Mbps	15/92	-74dBm

表2 動作状況のテスト結果

* アクセスポイントの方向にアンテナを向けた場合

リスト2 iwconfigの実行例

```
# iwconfig wlan0
wlan0 IEEE 802.11-DS ESSID:"TEST_LAN" Nickname:"HERMES I"
      Frequency:2.412GHz Sensitivity:1/3 Mode:Managed
      Access Point: 00:60:1D:21:D1:DF
      Bit Rate:11Mb/s RTS thr:off Fragment thr:off
      Encryption key:off
      Power Management:off
      Link quality:19/92 Signal level:-70 dBm Noise level:-89 dBm
      Rx invalid nwid:0 invalid crypt:0 invalid misc:0
```

リスト3 アドホックモードの設定例

```
[/etc/pcmcia/configへの追加設定]
# TEST for PLANEX GW-NS110
card "PLANEX GeoWave/GW-NS110"
  manfid 0xd601, 0x0002
  bind "wlan_cs"

[/etc/pcmcia/config.optsへの追加設定]
# OPTION for wlan_cs
module "wlan_cs" opts "port_type=3 channel=1 network_name=TEST_LAN"
```


Compaq ARMADAを置き、各地点で接続を確認してみる。表3に示すように、インフラストラクチャモードとほぼ同様な結果となった。電波の出力自体は変わらないのだろうか。ちょっと意外な結果である。確認中に、WLI-PCM-L11は自動的に伝送速度のモードが切り替わっていることがわかった。この結果も表3に掲載しておく。

⚡ CardBusはどうなんなの巻

数日後、副編@渉外担当にレンタルをお願いしていたNECのLaVie NX (型番はLT2344A) が到着した。もちろんCardBus対応スロット対応マシン(変?)である。「間に合わなければ2機種だけでいいな」とすっかり油断していたのだが、来ちゃったものはしかたがない。テストするとしよう。

まず、コレガのPCC-11をおそろおそろ挿し込んでみた。おおっ大丈夫だ。きちんと装着できるぞ。逆に、5V仕様のカードをCardBusスロットに装着することもできた(動作もOK)。さて、挿さったのはいいが、はたして動くんであろうか…。結論から言うと、コレガのPCC-11、アイコム製のSL-1100とも問題なく動作した。リスト5に示したのが追加した設定である。

PCMCIAカードサービスの無線LANカード用ドライバモジュールには以下



写真6 プラネックス GW-SN110



写真7 アイコム WAVEMASTER SL-1100

の5種類があり、各ドライバごとに対応しているカードが異なる。

```
wvlan_cs
airo_cs
netwave_cs
ray_cs
wavelan_cs
```

それぞれのドライバでサポートされるカードの情報は、PCMCIAカードサービスのソースコードディレクトリにあるSUPPORTED.CARDSファイルに記載されている(このファイルの最新版がLinux PCMCIA Information Pageの“supported cards”ページ)。基本的には/etc/pcmcia/configに、カ

ードの認識名とバインドするドライバを記述すればいいだけだ。リスト5の“card”で始まる行が認識名、“bind”で始まる行がドライバの指定である。“manfid”は製造元を識別するID。カードの認識に関連する情報は、前述したように、“cardctl ident”でわかる。

今回はすべての機種がwvlan_csで動作した。「SUPPORTED.CARDSに記載されていないぞ。困ったなあ」という場合でも、各ドライバを順に試していけば(“bind”先を適当に変えてみる)運が良ければ動作する可能性があるはず。これは無線LANカードに限った話ではなく、ほかのタイプのPCカードにもあてはまる。チャレンジャーな読者はいろいろ試してみてほしい。

接続	伝送モード	接続	伝送モード
a	11Mbps/5.5Mbps	b	11Mbps
c	11Mbps	d	11Mbps/5.5Mbps
e	11Mbps/5.5Mbps	f	1Mbps
g	5.5Mbps	h	1Mbps
i	5.5Mbps/2Mbps	j	x

表3 動作状況のテスト結果(アドホックモード)

```
リスト4 network.optsの設定例
case "$ADDRESS" in
*,*,*,*)
INFO="Sample private network setup"
BOOTP="n"
DHCP="n"
IPADDR="192.168.1.1"
NETMASK="255.255.255.0"
NETWORK="192.168.1.0"
BROADCAST="192.168.1.255"
;;
esac
```

```
リスト5 Configの設定(SL-1100、PCC-11)
# TEST for ICOM SL1100
card "ICOM SL-1100"
manfid 0x0156, 0x0002
bind "wvlan_cs"

# TEST for Corega PCC-11
card "Corega Wireless LAN PCC-11"
manfid 0xc00f, 0x0000
bind "wvlan_cs"
```

はやっぱり ベンチマークだよの巻

今回の任務のそもそもの始まりは「11Mbpsの無線LANカードも出揃ったし……」というところにあった。やはり、伝送速度の計測をやらないわけにはいかないだろう。「テストだけでもLinux magazineラボのほうでやってくれないかな」と思いつつも、テスト結果の信憑性にもかかわるので、ここからはピシッといくのである。

テスト方法については、6月号のネットワークカード特集でLinux magazineラボが用いた方法を参考にした。ftpを使って巨大なファイル(今回は約60Mバイトのtarボール)を転送し、その実行時間をtimeツールを使って計測するという方法である。ほかのネットワーク環境からの影響をなくすために、編集部LANから切り離れた

	メーカー	備考
Let's note AL-N1 (T515J5)	Panasonic	Pentium 150Mhz、RAM48MB搭載
LaVie NX (LT-2344A)	NEC	Pentium 233Mhz、RAM64MB搭載
自作デスクトップPC	-	Celeron 466Mhz、RAM128MB、ノンブランド100BASE-TX LANカード (PCI) 搭載
Fast SW -P8	コレガ	100BASE-TXスイッチングハブ
EtherLink (3C589D)	3Com	10BASE-T PCMCIA LANカード

表4 ベンチマークテストに使用した機器

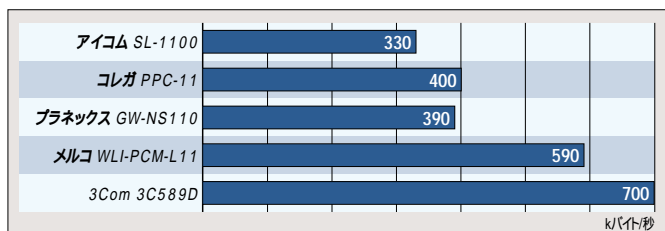
ローカルな環境で行っている。

有線LAN側には、100BASE-TXのスイッチングハブに接続した同じネットワークセグメント上にあるデスクトップ機を用意した。アクセスポイントは同一のもの(メルコ WLA-L11)を使用。本来はセット製品なので同一メーカーの組み合わせでの計測も必要なのだろうが、有線LAN部分の機能に差(10BASE-Tと100BASE-TX)があること、プラネックスのアクセスポイントが都合により入手できなかったため、すべての組み合わせをテストできないことを考慮した結果である。評価基準として、10BASE-TのPC LANカード

でも同一操作での計測を行った。

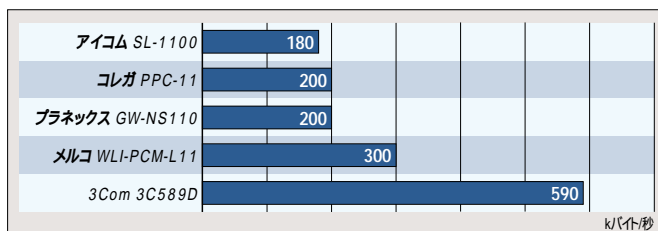
さて、結果はというと、これがあんまり良くないのである。詳しくは下段の解説にゆずるが、10BASE-Tのカードの前に全滅とは……。しかしまあ、実用上は100Kバイト/秒くらいの違いなんてわかんないだろうし、そのぶんモジャモジャと邪魔になるケーブルがなくなるワケで、さらには、あちこち移動もできる。また、体験した感想から言わせてもらおうと「これは使える」というのが実感。ワイヤレスでアクティブにいこうではないか!

グラフ1 ノートPCからデスクトップへ



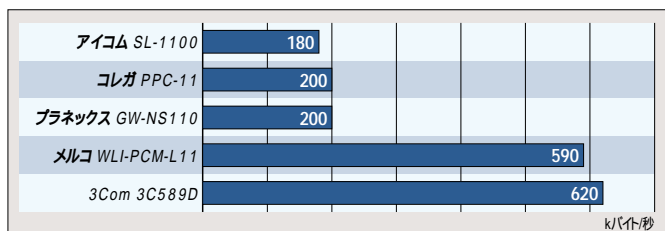
インフラストラクチャモードで接続し、アクセスポイント経由でデスクトップPCにデータ転送を行った。WLI-PCM-L11が他の3機種を引き離す高い値を示した。WLI-PCM-L11のみがPCMCIAカードサービスでサポートされていることから、カード自体の性能というよりドライバとの相性の問題なのだろうか。3C589Dはノーマルにハブ経由での接続。

グラフ2 ノートPCからAP経由でノートPC



同じアクセスポイントに接続したノートPC間でデータ転送を行った。スロットの問題で同一機種で試せないもの(SL-1100とPPC-11)は、受信側にGW-NS110を使用。受信側のパフォーマンスの違いがでたのか、有線LANへの転送に比べると4機種とも数値が極端に落ちた。3C589DはAL-N1にプラネックスのENW-3503(10BASE-T)を装着して計測。

グラフ3 アドホックモード



アドホックモードで2つのノートPCを接続してデータ転送を行った。同一機種で試せないケースではGW-NS110を使用。ここでもメルコが高い数値を示した。他の3機種は同じコントロールチップ(Prism)を搭載しており、やはりドライバの問題で十分な性能を発揮できていないと考えられる。3C589DはENW-3530とクロスケーブルで直結。

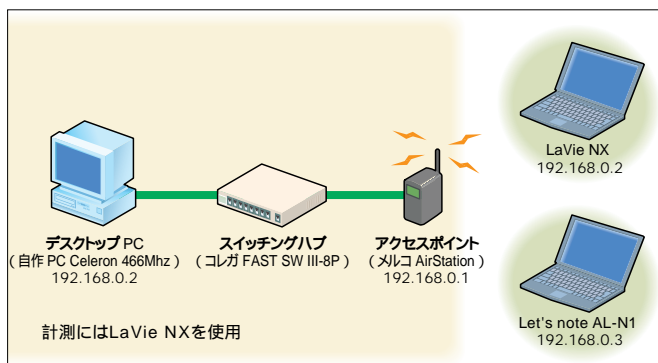


図4 テストに使用したネットワークの構成

アイコム WAVEMASTER SL-1100 / AP-3

価格 SL-1100 2万6800円
AP-3 5万4800円

<http://www.icom.co.jp/>

アクセスポイントAP-3は、テストした3機種の中で唯一、金属性の筐体である。これは放熱性や耐久性を考えるとやはり優位だろう。有線LAN側のインターフェイスは100BASE-TXの全二重接続に対応。無線側はPCカードスロットにSL-1100（本体に付属）を装着する方式になっている。本文中では触れられなかったが、本機の特徴でもあるDHCPサーバ機能は動作を確認できなかった。SL-1100のコントロールチップはLinux側での認識から、IntersilのPrism と思われる。



プラネックスコミュニケーションズ GeoWave GW-NS110/GW-AP110

価格 GW-NS110 オープンブライズ（参考価格 2万1900円）
GW-AP110 オープンブライズ（参考価格 4万9900円）

<http://www.planex.co.jp/>

GW-NS110は、下の写真からわかるとおりLEDがアンテナ部ではなくカード本体部分に付いている。これはアンテナ部分が取り外し可能になっているためなのだが、通常の状態では視認性が若干良くない。SL-1100同様、カードサービスの認識からIntersil Prism を搭載していると思われる。現時点では未対応だが、GW-NS110はLinuxに対応予定となっている。アクセスポイントGW-AP110は、スタイリッシュな外観が特徴的。有線側は10BASE-Tにのみ対応する。



コレガ Wireless LAN Pack-11

価格 6万5400円

<http://www.corega.co.jp/>

本文中で紹介したPA-11とPCC-11×2のセット製品。アクセスポイントPA-11は、設定が簡単で、シリアルポートを持っているのが大きな特徴である。ケーブル接続してターミナルからログインできるので、LAN環境がまったくない状態でも設定が可能である。有線側のインターフェイスは10BASE-Tにのみ対応している。PCC-11は、LEDが1つしかなく状態を把握しづらいのがやや難点。コントロールチップは、IntersilのPrism だ。



メルコ AIRCONNECT WLS-L11W

価格 6万5400円

<http://www.melcoinc.co.jp/>

WLA-L11とWLI-PCM-L11×2のセット製品。アクセスポイントWLA-L11は縦型の筐体だが、足の底の部分がゴムになっていて見た目より安定している。前面と左側面の2箇所にLEDがあり、接続状況も確認しやすい。有線側のインターフェイスは100BASE-TXに対応している。ベースモデルである同機のほか、モデム、ISDN、CATVに対応したモデルもある。WLI-PCM-L11は、今回唯一のルーセントテクノロジー製のコントロールチップ搭載製品。現時点では間違いなくベストチョイスだ。



すべてのロープレはここから始まった!

Rogue のすべて

今から18年前、「RogueのないUNIXなんてUNIXじゃない」と言い切ったBSDの達人がいた。そんなRogueとは一体……

文: 竹山 正寿
Text: Seiju Takeyama

3D CG: 児玉 敏彦
3D CG: Toshihiko Kodama



Rogueって何?

.....R o g u e

ゲームソフトの世界に、ロールプレイングゲーム (RPG) というジャンルが確立されて久しい。コンピュータを使ったRPGの元祖といえばRogueである。もともとDECのミニコン (大型コンピュータ。その中では小型なので、「ミニコン」である) PDP-8用のUNIX 4.1BSDで動くゲームプログラムとして、UCLAの Electrical Engineering and Computer Science Faculty, Computer

System Research GroupのMichael C. Toy、Kenneth C. R. C. Arnoldが中心となって作られたものだ。

言うまでもなく、彼らはBSD版UNIXの開発と深いかわりを持っていた人物であり、純粹にゲームとして作ったわけではなく、まだ誰も作ったことのないロジックやシステムの創造を目指していたはずだ。また、完成までには、周囲のUNIXユーザーの惜しめない献身 (デバッグやテスターには事欠かなかったはずだ) 協力もあつたであろう。

もちろん、RPG自体の成り立ちは、Rogueよりも古い。ゲームマスター (ゲームの世界の中では神のような存在) が、あらかじめ用意した世界や設定の中で、プレイヤーが演劇のように役割を演じて、ゲームマスターの設定した課題をクリアしていくテーブルトーク形式が、そもそものRPGの始まりで、特に有名なものがD&D (Dungeons and Dragons) というTSR社のゲームだ。

D&Dのパッケージの中には世界や登場するモンスターをどのように作り、どのような数式で戦闘判定を行うかというマニュアルと、多面体のサイコロ

が入っているだけである。ゲームマスターはこのマニュアルにしたがって世界をつくり、ワナを置き、目的を設定する。あとはプレイヤーに状況説明を行い、どのような行動を起すかによって、関数やパラメータに従い判定を行ってゲームを進行していく。必要なものは想像力だけというすさまじいものだ。

このD&Dでもっとも面倒な役割であるゲームマスターをパソコンにやらせて、自分達は心から楽しむだけにできないか? さらに、何度も飽きずにプレイできないかという命題に、乱数を使って毎回異なった迷路を生成するロジックや、自由度の高いゲームシステムとストーリー性をパラメータと関数で解決したのがRogueであった。この流れはトルネコの冒険、ディアブロといった専用機のゲームへも引き継がれている。



Rogueを解くためのプログラムまであった

.....R o g u e

Rogueのリリース後に、関連する多くのプログラムが発表されたことからわかるように、この画期的なプログ



写真1
アスキーから発売されていたPC-9801版Rogueのパッケージ

Rogue のすべて



ラムは、ハッカー諸兄の好奇心やチャレンジ精神を揺さぶったようだ。

そのなかには「Rogueを解くためのエキスパートシステム」なんてのもまであったのだ。“ROG-O-MATIC”と名付けられたこのプログラムは、ハッカーの溜り場として有名なカーネギーメロン大学が国防省の援助によって開発したものだ。国防省のお役人は、Rogueを自動で解くプログラムで、どこで戦争するつもりだったのだろうか? 当時でもこのニュースは驚きと呆れで迎えられたという。



Rogueの歴史

Rogueが研究室から出て、一般に流れ始めたのは1980年ごろと言われている。4.1BSDのリリースとともにRogue 5.2が、4.2BSDとともに5.3がリリースされた。しかし、このころからソースが流通しなくなってしまった。また商用として、5.3をベースにAI DesignがIBM PC用の“IBM Rogue”を発売、アスキーからはPC-9801シリーズや汎用MS-DOS版などが発売された(写真1)。そのほかMacintosh向けにGUIを持った“mac rogue”なるものも登場したが、あまり評判は良くなかったようだ。

Rogueの子孫たち

RogueがPCへ移植され始めたころ、UNIX上では、Rogueをベースにより複雑なシステムを導入したHack、時間という要素を組み込んだIarnなどが生まれていた。中でもHackはNetHackへと進化を遂げ、最近ではついにグラフィック化された(画面1)。

興味のある人はRogue系のゲームを一同に集めたサイトを参照するといいたいだろう(画面2)。

なお、オリジナルRogueのソースプログラムは消失したという噂があったのだが、米国の某サイトでHP9000用にインプリメントされたものが公開されたことがある。Linuxで動作させるには、セーブ、ロード、inventoryなど一部を書き換える必要があるが、一応ソースそのものは存在している。編集部でも入手したが、再配布の可否が締切までに確認できなかったため、CD-ROM収録などは見合わせた。

米国のニュースグループやサイトをのぞいていると時々、ソースについての情報が流れることがある(6月23日現在ではIBM-PC用のRogueのソースが公開されていたが...?)。

日本語対応のRogueクローンはLinux JG ProjectのWebサイトから入

手可能だ(<http://jg.linux.or.jp/>)。

なお、BNNより発売されていたRogueの攻略本「Rogueハンドブック」はPDF化され、Web上で章ごとに順次公開予定である(<http://www.bnn.co.jp/rogue/rogue.html>)。

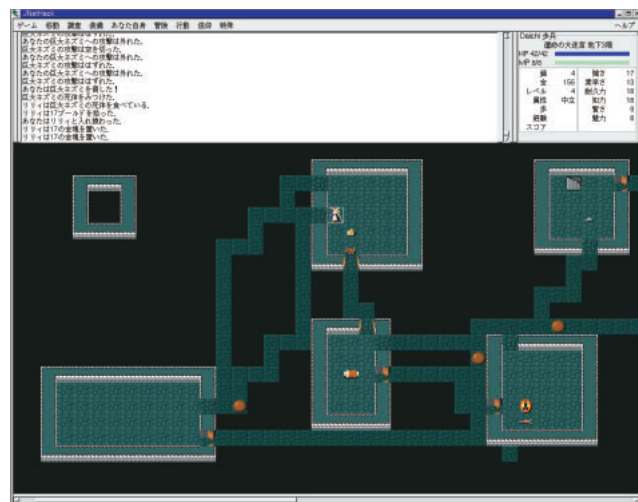


CD-ROM収録ソースについて

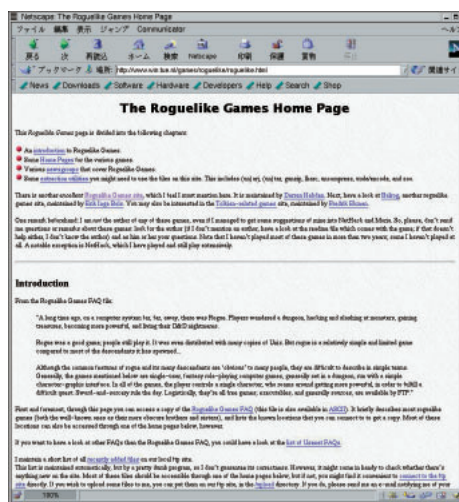
.....R o g u e
オリジナルに近い雰囲気を楽しむRogueクローンをCD-ROMに収録した。“Roguelike Games”で公開されているものである。ひとつは“Rogue5.3 clone”(Irogue0.1.tar.Z)、もうひとつは“The Rogue98.5”(rogue985.tar.Z)である。ゲーム内容には差がないが、使用する環境によってはうまく動かない場合があるので、試用してみても安定して動くほうを選択して欲しい。インストールの方法は、

```
$ tar xzf lorange0.1.tar.Z
$ tar xzf rogue985.tar.Z
```

として解凍した後、「make」を行う。その後PATHが通っているディレクトリ(/usr/local/binなど)にプログラム本体をコピーすれば、インストール完了だ。



画面1 Japanese NetHackのメインウィンドウ。



画面2 Roguelike Games Homepage <http://www.win.tue.nl/games/roguelike/roguelike.html>

起動はコマンドラインから、

```
$ TERM=vt100 rogue
```

とする。シェルとしてbashを使用していれば、上記のように環境変数を設定して、特定のコマンドを実行できる。CD-ROMに収録したRogueクローンがうまく動かなかったり、英語のメッセージが苦手な場合は、Linux JG Projectが配布している日本語版を利用することをお勧めする。こちらは、RPM形式のバイナリ/ソースパッケージで提供されている。



基本操作

.....R o g u e

ゲームを始めると画面3のようなシンプル!?な画面が表示される。画面の一番上がメッセージエリア、真ん中がマップ、一番下がステータス表示エリアである。

@がプレイヤーでモンスターはアルファベット、ダンジョンの構造やアイテムは特殊文字で表示される。ゲームの最終目標は地下20階より深くにあるイエンダーのお守りを取って戻ってくる。志半ばで倒れた場合には持つ

ている宝が多いほど高いスコアとして記録される。

まずは“@”の動かし方をマスターしよう。図1のようなキーに動く方向が対応している。このキーはクラシクなテキストエディタviのカーソル移動と同じだ(ただしviには斜め移動はないので、気をつけよう)。慣れると快適だが最初は地獄かも?。“SHIFT”キーを併用すると、壁やモンスターなどに当たるまで一気に移動する。必要に応じて、“SHIFT”を使いながら動かすとよいだろう。

ときどき、出入口がない部屋や、廊下の先に部屋がありそうなのに出入口がないことがある。そういう場合には“s”コマンドで出入口を探すことになる。1回くらい“s”キーを打ってもなかなか隠し扉は現れない場合、何度か繰り返すことになる。面倒ならば繰り返す回数を先に入れるとよい。たとえば、9回“s”を押す代わりに“9s”と入力すればよい。この数字キー+コマンドキーの組み合わせは他のコマンドでも利用可能だ。

部屋にはワナがしかけられていることがあり、毒を受けたり、別の部屋や階に飛ばされることがある。ワナを探

すには“s”で探しながら歩くのだが、面倒くさい。

入った部屋が暗い場合には、自分の周囲しかの壁と扉しか表示されないの、1歩1歩注意深く進もう。

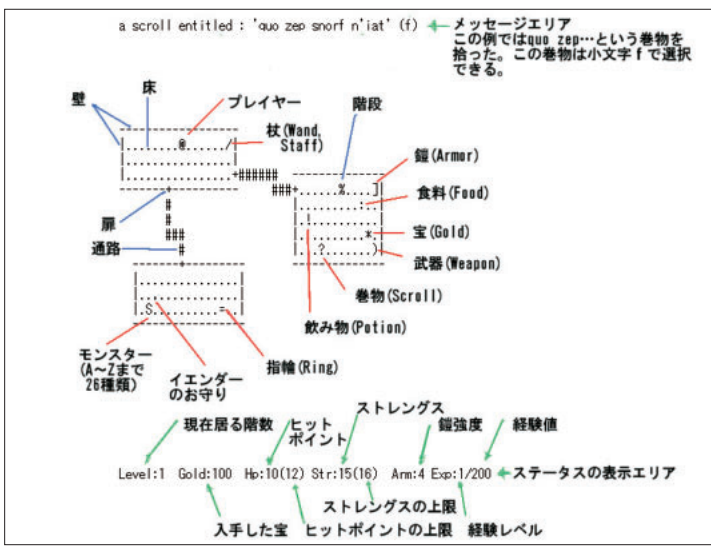
次の階に移動するには、階段の上に乗って“>”で降りることになるが、イエンダーのお守りを持っていない限り、階段は一方通行で上ることはできない。

腹が減ったら?

動き回ると腹が減り、ステータス表示エリアの右端にhungryという表示が出る。さらにそのままにしておくとweak、faintとなって満足に動けなくなり、最後は餓死してしまう。腹が減ったら、“e”+「食料」で食事をする必要がある。

動き回っても喉は渴かないのだが、飲み物(potion)も存在している。こちらは喉の渴きをいやすためではなくて、飲むと体力が回復したり、毒で攻撃力が下がったりする魔法の薬だ。飲むには“q”+「飲み物」だ。

ステータス表示エリアには腹具合とともに、耐久力を表すヒットポイント(Hp)、攻撃力(Str)、防御力(Arm)などのパラメータも表示されている。戦闘などで被害を受けるとヒットポイ



画面3 Rogueの画面

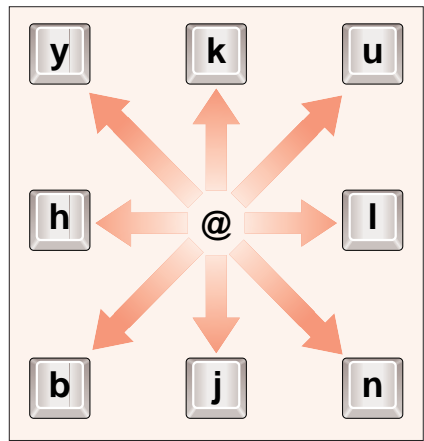
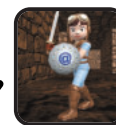


図1 “@”の移動方向



ントが下がるのだが、動いたり、何か行動をするごとに上限まで自動的に回復する。何もすることがない場合には“.”で動かないという手もあるし、healingやextra healing potionを飲んで回復する方法もある。

攻撃力は、毒を飲んだり、毒を持つRに噛まれたり、ワナにはまると落ちてゆく。値が小さくなるほど攻撃力も小さくなり、ゼロになると死んでしまう。回復するにはrestore strength potionを飲めば上限まで回復、add strength potionを飲めば1だけ回復する。また、攻撃力が上限の時にadd strength potionを飲むと、上限の値が1上がる。

Armは鎧の能力なので、着替えることで変わる。

リスト1 コマンド一覧			
コマンド		キー	
移動	左	1コマ	h
	右	1コマ	l
	上	1コマ	k
	下	1コマ	j
	左上	1コマ	y
	左下	1コマ	b
	右上	1コマ	u
	右下	1コマ	n
	左	進めるだけ	H
	右	進めるだけ	L
	上	進めるだけ	K
	下	進めるだけ	J
	左上	進めるだけ	Y
	左下	進めるだけ	B
	右上	進めるだけ	U
	右下	進めるだけ	N
探索	周囲の壁や床を探索		s
	ワナの種類を表示する		^
持ち物	確認		i
	モノを投げる。矢を射る		t
	武器を握る		w
	鎧を着る		W
	鎧を脱ぐ		T
	食料を食べる		e
	飲み物を飲む		q
	巻物を読む		r
	杖を振る		z
	指輪をはめる		P
	指輪を外す		R
	モノを捨てる		d
	名称を変更する		c
	ゲームを終了する		Q
	その他	セーブする	
コマンドリストを表示する			?

アイテムを拾って使う

アイテムの上に乗ればそのアイテムを拾ったことになる。ただし、巻物(scroll)、杖(wand、staff)、飲み物(potion)、指輪(ring)については拾っても、その材質しかわからず、効果については使ってみて類推するか、identify scrollという巻物で識別するしかない。材質と効果の組み合わせは、新たに冒険を始めるたびに化する。

常に持って歩けるのは26個までで、それぞれがアルファベット1文字で表される。今持っているアイテムは“i”で確認できる。

それぞれのアイテムの使い方はいろいろだ。たとえば、鎧(armor)の場合はまず“T”で脱いでから“W”+「鎧を表すアルファベット」で着替える。アルファベットで指定するのは、ほかのアイテムでも同様だ。武器(weapon)を持ち替えるのは“w”+「武器」だ。弓矢のような投擲武器は、“w”キーで弓に持ち替えてから“t”+「方向」キー+「矢」で発射する。ちなみに、矢以外のものも投げられる。鎧や食料を投げてもいいのだ。

巻物(scroll)は“r”で読むというのが基本。読むと、その効果に対応したメッセージが表示される。しかし、読んででも不可思議なメッセージが表示されるだけでその効果がわからないこともある。これは魔法の効果が効く状況にないのである。このあたりは何度となくプレイすることで、効果のほどがわかってくる。

どんなアイテムか識別するidentifyや鎧の能力を上げるenchant armor、武器の能力を上げるenchant weaponを読んだ場合には、何に対して魔法をかけるのか聞いてくるので、アルファベットで指定する。これらの巻物は、一度読むと消滅してしまう。また、scare

monsterという巻物だけは読まずに“d”で足元に落とすと、その上にいる間だけどんなモンスターにも攻撃を受けないし、一列に並んでいるモンスターに対して投げつけると全部やっつけるという効果がある。

杖(wand、staff)は“z”+「方向」で魔力を放射する。使える回数が決まっていて、それを知るにはidentifyという巻物を読まなければならない。魔力を使い切ると、“z”を使っても何も起こらないが、投げつけて当たると効果がある。

指輪(ring)は左右の腕に1つしかつけられない。“P”+「指輪」+「RかL」で装着、“R”+「RかL」で外すことができる。slow digestion以外は身に着けている間は、通常の倍の速度で腹が減るので要注意だ。

ぶつかれば戦闘

モンスターに接触して、さらにその方向へ動こうとした場合には戦闘とみなされる。経験レベル、攻撃力、使っている武器の性能によって相手に与えるダメージが変わってくる。また、大型の武器は、攻撃力が小さいと命中しにくい。

離れている敵に対しては、弓矢などの飛び道具や魔法の杖などを使って攻撃することができる。一般に飛び道具は近接している敵に対して使うと当たりにくい。

モンスターはその特性によって、戦ったほうがいいもの、なるべく戦わないで済ませたほうがいいもの、近づかず、離れたまま戦ったほうがいいものなどいろいろいるので、それぞれの敵に合った戦い方を身に付けなくてはならない。

基本的なことは説明したので、あとは実践あるのみ。いざダンジョンへ!



Rogue Live!

.....R o g u e

ようこそダンジョンへ

2000年6月某日深夜、私は一人で旅に出ることを決意した。旅立つ前に城へダンジョン（地下迷宮）へ行く申請を出したところ、魔法によって防御力を1だけ上げたring mail（鎖かたびら）攻撃力を1だけ上げたmace（つちほこ）とshort bow（小型の弓）とarrows（数十本の矢）、1食分の食料を渡された。+1 ring mailと+1 maceを身に付けてダンジョンへ降りたった（画面4）。

勇んでダンジョンへ飛び込んだのだが、敵もめぼしい宝もなにもない。とりあえず、何か忘れ物はないか確認しておこう。i（inventory）コマンドで持ち物を確認してみる（画面5）。忘れ物はなさそうなので、部屋の出入口へ進み、通路を道なりに進み、次の部屋の扉を開けた。

部屋には宝と巻物が落ちていないか。ただ、その手前にはBat（B）がいるのでこいつを倒さねばならない（画面6）。

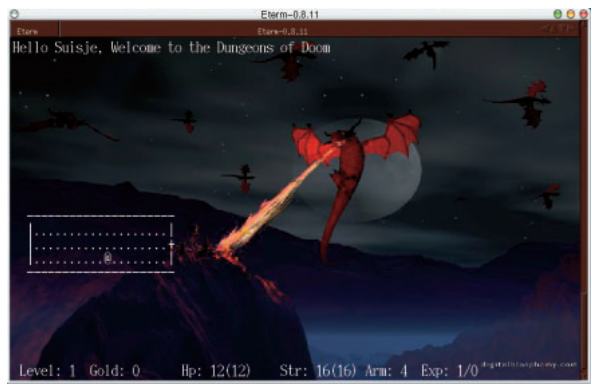
まっすぐ一歩、部屋へ踏み出したが、コウモリ野郎は攻撃してこない。もう一歩踏み出し、斜め下にいるコウモリを一撃。反撃してくるかと思ったら、上へ1コマ上がっただけ。どうやらこいつはランダムに飛び回る性質があるらしい。さらに一撃。まだ、死なないうえに、今度は反撃してきて私のHpが10に減った。ここでひるまず、さらにもう一撃してようやく倒した。経験値は1しか増えなかった。

お宝と巻物を拾って、右の扉へ抜けると、その次の部屋にはHobgoblin（H）が1匹。こいつは手ごわい。1回目の戦闘でHpが一気に8まで下がる。2回めで6に。まだ倒せない。次の一撃で4まで減った。最初の一撃でHpが4も減ったことを考えると、次の攻撃で

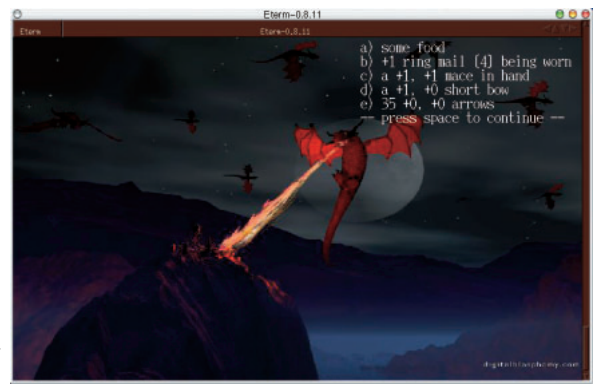


やられる可能性があるので、ヒットポイントが5に回復するまで、退却する。そして起死回生の一撃を加え、ようやく倒すことに成功した。

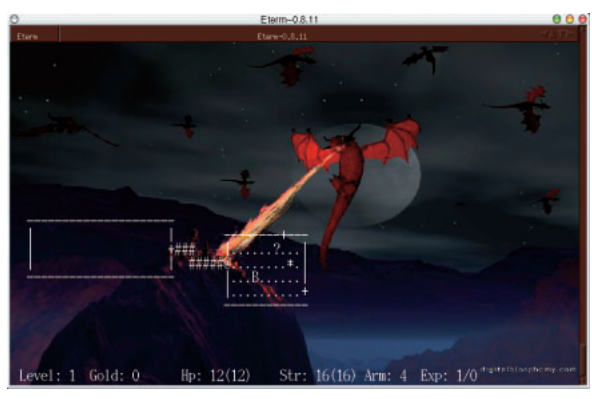
その後、Snake（S）とKestrel（K）が2匹、Hが2匹ずつ出現した。途中で経験値が10を超えたところでレベルが2になり、Hpの最大値が12から15へ増えた。幸い階段が近いのでそのまま下ってしまおう（画面7）。



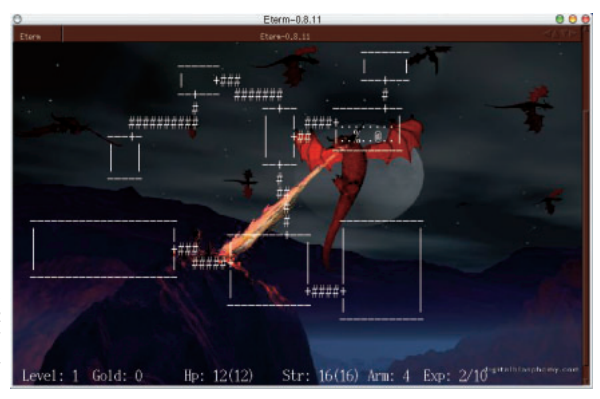
画面4
いよいよ冒険の始まりだ。雰囲気を出すために、Etermの背景にイカス画像を貼ってみた。



画面5
これが全財産だ。



画面6
最初の敵はBat（B）。油断禁物だが、これくらいなら勝てるだろう。



画面7
たくさん戦ったので、レベルが2に。そろそろ下の階に行ってみるとするか。



地下2階から地下4階まで

地下2階へ降りると、SとKが2匹ずつ出迎えてくれた。Kは素早いので、まごまごしていると囲まれてしまう(画面8)。急いで、出入口へ飛び込んで戦う。出入口は狭いので、囲まれる心配がなくなるのだ。まあ、この状態で後ろからはさまれるということもあるが、四方からタコ殴りにされるよりはましだ。ヒットポイントが8にまで落ちたが、なんとかクリア。“20.”で20回休息して回復させた。

このくらいの階ではレベルが4以上になるまではHとI(アイスモンスター)にさえ注意すれば問題はない。特に、眠っているIを不用意に叩くと、冷気で凍らされてひどい目に合う。

幸い、私が最初にIに出会った部屋は広がったので、遠くまで離れ、武器をshort bowに持ち替えてやっつけた。

地下4階まではこれといった問題はなかった。ただ、拾ったアイテムは鎧が3個、武器が4本、巻物6本、飲み物3本と正直言って重い。最大26個まで荷物を入れられる袋を持っているのだが、仮に鎧を26個拾ったら歩いて移動できるのだろうか?(できます)

武器や鎧は、今使っているものより良さそうなものを見つけたら、すぐ取り替えるのが基本だ。私も武器をtwo handed sword、鎧もplate mailへ着替えた。

potionやscroll、wandといったアイテムについては、一度使ってみれば、後で同じ物を拾った時にどんなものかわかって利用しやすいが、毒などで不利な状況になることもある。また、ある程度数が揃ってから使うと、組み合わせによってひどい状態を回避しやすくなる。とりあえずアイテムをためながら先へ進むことにした。

そろそろ腹が減ってきた

地下3階くらいになるとRattle Snake(R)が現れるのだが、こいつは毒を持っていて、不用意に戦うと攻撃力が下がってしまう。gain strength/sustain strengthでできるだけ攻撃力の上限を上げておきたいところだ。そこで、数本たまったpotionを飲むことにした。

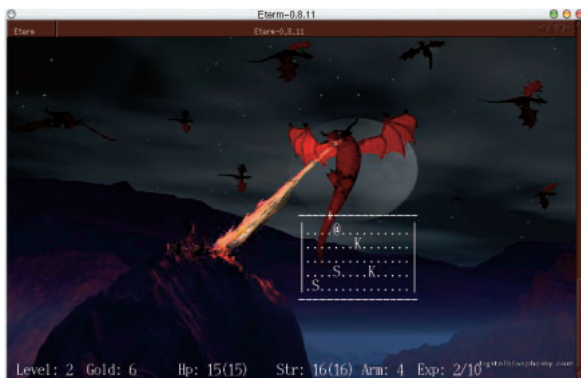
しかし、1本目のpotionを飲むと“a

cloak of darkness falls around you”というメッセージが…。要するに目が見えなくなったのである。部屋の真ん中にいると、どこから攻撃されるかわからないので、通路へ移動。残ったpotionをさらに飲んでみる。うまくhealingやextra healing、see invisible potionに当たれば、視力は回復するはずだ。しかし、残っていたpotionはparalysisとpoisonであった! 体が動かない! ストレngthは下がる! と大変なことになった。“30.”でとりあえずマヒ状態から回復。“99.”と休息しながら、殴られたら左右へ移動し、ぶつかった方向のモンスターと戦ってどうにか切り抜けた(画面9)。

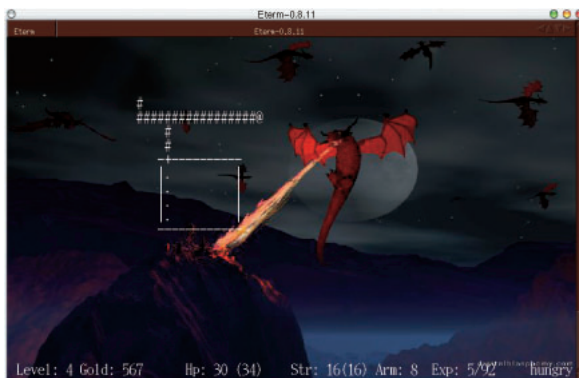
無駄な動きが多かったこともあり、腹が減ってきたので食事をとって、階段のところまで移動する。

散々な目にあったが、さらにここで巻物を読むことにした。階段の上で巻物を読めば、たとえaggravate monstersを読んでしまい、モンスターに取り囲まれてもすぐに階下へ逃げられるのだ。巻物はsleep、food detection、identifyの3種類であった。identifyで持っていたruby ringの識別をしてみると、なんとおなかの減り方が半分になるslow digestionだ! もちろんすぐに右手に装着だ。

危なかったなと思いながら、地下5階へ降りた。降りた部屋には何もなく、



画面8
いきなりの試練。不思議なことに、今下りてきたばかりの階段は上れないので、上の階に逃げることはできない。



画面9
敵が見えないというのは、とても不安だ。通路でじっとして、敵が来ないようにお祈り中。



2つ目の部屋にも食料が1つあるだけ。さらに、その先へ行くと!?(画面10)。

これは俗にいうモンスター部屋だ。ふだんは寝ているようなLeprechaun (L) やNymph (N) まで起きている。もちろん、不用意に殴れば、せっかくのお宝や集めたアイテムが減ってしまう。こういったモンスター部屋にはアイテムがたくさん落ちているものなので、差し引きゼロとも考えられるが...。一番手前にいるNはどうにも避けようがないので、目をつぶって殴りつける。

やはりshort bowを盗まれてしまった。まあ、これくらい仕方ないな。

次はCentaur。このあたりでは最強のモンスターだ。Hpが6まで下がったが、まだNがくる。もうどうでもいいやーということで次々に殴っていると、レベルが上がり、またヒットポイントが増えた。ようやく全員倒した時には、残りHpは4まで減っていた。しかも2番目のNに盗まれて、食料がなくなりました。よく見ると部屋の中に1つあるので、喜んで部屋へ入ったら、いきなりワナにかかり、地下6階へ落ちた。モンスター部屋にはワナも多いことを忘れていた。

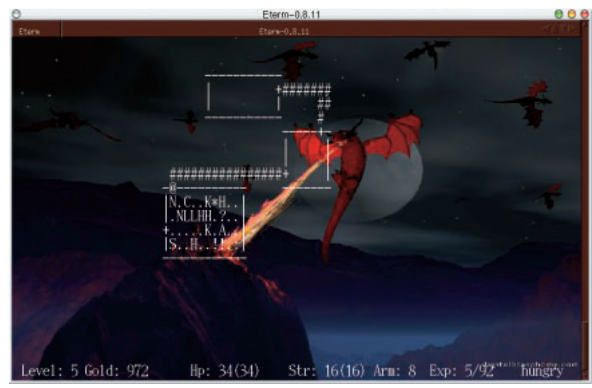
地下6階にはcross bowが落ちていたので、盗まれたshort bowの代わりにちょうどよいと思い、早々に持ってみたらなんと呪われていた! 呪われたアイテムは、マイナスの魔法がかかっていて能力が低いのだ。それに、cross bowじゃあ、接近戦では役立たない。困った! 困ったと思いつつ、次の部屋へゆくと、巻物が2つもある。もしかすると呪いを解く巻物があるかもしれないので早々に

読んだところ、どんぴしゃり “ You feel as though someone is watching over you. ” というメッセージだ。

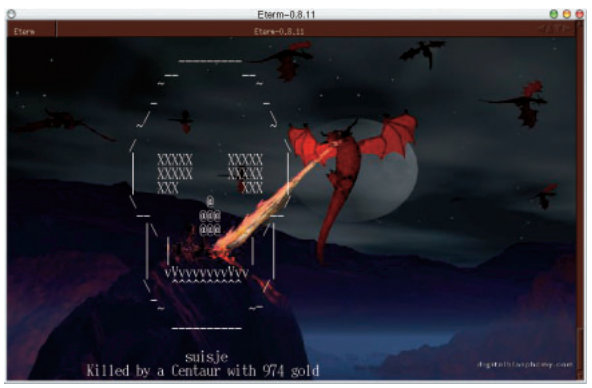
一息ついて、もう1歩踏み出すと、ジャンプ。またワナを踏んだのであった。なんとジャンプした先は、またまたモンスター部屋のだ真ん中。万事休すな状況だが、最後に持ち物をチェック、奇跡を信じて、正体不明の巻物を1つ足元へ置いてみた。もし、scare monsterならば殴られずに済むのだが...世の中はそんなに甘くはなかった(画面11)。

こうして私の冒険は幕を閉じた。クローンでない本物のRogueでは死亡すると持っていたお金の10%を取られるが、立派なお墓を立ててもらえた*。だが、Rogueクローンではお墓の代わりに、りっぱな髑髏が出迎えてくれるだけだ。あんまりな気がするが、まあ仕方あるまい。さあ、また次の旅の準備をしようか... (最初に戻る)

*余談だが、あと一殴りでやられる瞬間に“Q”でゲームを終了すれば、所持金を10%とられることはない。しかしハイスコアランキングには、途中で逃げ出したことが記録されてしまうので、そうやってハイスコアを出すような輩は、もちろん「チキン野郎」呼ばわりされてしまうのだ。



画面10
大ピンチ! 己と剣を信じて戦いぬくしかない!



画面11
Amulet of Yendor を手に入れるまでに、何回この「どくろ」とご対面するんだろう?

リスト2 ワナ一覧		
種類	デメリット	メリット
arrow	矢が飛び出し、当たると体力が下がる	矢が落ちていることも...
bear	数ステップ動けなくなる	
dart	投げ矢(ダート)が飛び出し、当たると体力が下がる	
poison dart	毒の塗られた投げ矢が飛び出し、当たると攻撃力が下がる	
trap	sleeping gas 1ステップだけ眠って動けなくなる	
teleport	同じ階のどこかに飛ばされ、しばらく方向がわからなくなる	モンスターから逃げ出すのに利用できる
trap door	落とし穴。下の階へ落ちる。しばらく方向がわからなくなる	モンスターから逃げ出すのに利用できる
weird	何の効果もない	
rust	鎧が錆びる	



リスト3 モンスター一覧

名称	出現深度	特徴
Aquator	浅~中	水のモンスターで、攻撃すると金属製の鎧は錆びる
Bat	浅	コウモリだけに、プレイヤーの周りをランダムに飛び回るため攻撃が当たりにくい
Centaur	中	ギリシャ神話に登場したケンタウロスの英語名。高い攻撃力を持っている
Dragon	深	言わずと知れた竜。火炎を吐き、離れていても被害甚大だ。倒せれば高い経験値が得られる
Emu	浅	エミュウがなんで洞窟に住んでいるのかは不明。防御力はそこそこあるが攻撃力はない
venus Flytrap	中~深	食虫植物なので歩いたりしないが、花卉にはさまれ拘束されることがある
Griffin	深	グリフィン(頭が鷲、体が獅子)。高い攻撃力を持つ。高い経験値が得られる
Hobgoblin	浅	金貨を隠し持つといわれている小型の鬼。序盤ではもっとも手ごわい
Ice monster	浅	正体不明の冷凍モンスター。離れていても冷凍ガスによる攻撃を受けて動けなくなることがある
Jabberwock	深	正体不明だがトカゲのような姿と言われている。高い攻撃力を持つが高い経験値が得られる
Kestrel	浅	隼の一種であるチョウゲンボウ。動作が機敏で、逃げても追いつかれる
Leprechaun	中	アイルランド民話に登場する妖精。宝を隠し持っている。手癖が悪い
Medusa	深	ギリシャ神話に出てくる女性モンスター。うっかり攻撃すると魔法をかけられ方向がわからなくなる
Nymph	中	ギリシャ神話に出てくる妖精。アイテムを隠し持っているが、うっかりすると盗まれることも
Orc	浅~中	オーク鬼。金に対する執着心が強く、落ちている金があるとその上に乗って動かなくなる
Phantom	深	幽霊だけに目に見えない。そのうえ、ふらふらと動き回る
Quagga	中	1本の角をもったキリンの親戚。蹴られると結構痛い
Rattlesnake	浅~中	ガラガラヘビだけに噛まれると毒で攻撃力が下がる
Snake	浅	ヘビ。序盤の典型的なザコモンスター
Troll	中~深	北欧に伝わる伝説の巨人。体力があるので倒すのには苦労する
black Unicorn	深	巨大な尻尾を持つモンスター。高い攻撃力を持っている
Vampire	中~深	吸血鬼だけに、かみつかれるとヒットポイントの上限が下がる
Wraith	中	怨念が形になったもの。眠っていることが多い。攻撃すると経験値を奪われる
Xeroc	深	モノや他のモンスターに化ける。階段だと思って近づくとこいつだったりする
Yeti	中~深	雪男。まあまあ高い攻撃力を持つ
Zombie	浅~中	ゾンビといっても、大した能力はない

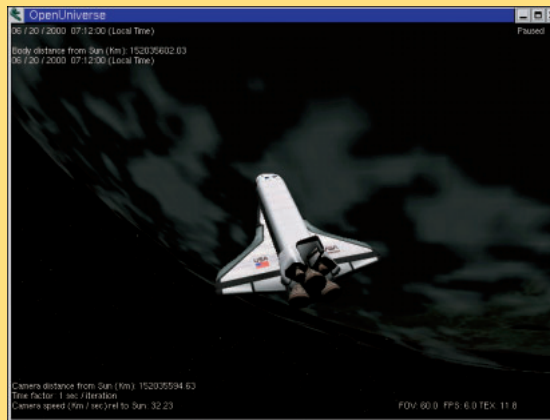
大文字が画面に表示される文字

リスト4 アイテム一覧

名称	効果	名称	効果
aggravate monsters	眠っているモンスターを起こす	blindness	かなり長い間、目が見えなくなる
balnk paper	何も書いてない	confusion	しばらく混乱してしまう
create monster	そばにモンスターを出現させる	extra healing	体力を大幅に回復する
enchant armor	鎧の能力を上げる	gain strength	攻撃力の上限を1上げる
enchant weapon	武器の能力を上げる	haste self	移動や攻撃の速度を2倍に高める
food detection	食料の位置を知ることができる	healing	体力を少し回復する
hold monster	接触しているモンスターを動けなくする	magic detection	魔法がかかったアイテムの位置がわかる
identify	アイテムの名前が分かる	monster detection	モンスターの位置がわかる
magic mapping	その階のマップを表示する	paralysis	一瞬動けなくなる
巻 monster confusion	読んだ後に最初に攻撃したモンスターを混乱させる	poison	攻撃力が下がる
物 remove curse	呪いを解く	raise level	レベルが1上がる
scare monster	上に乗るとモンスターから攻撃を受けなくなる。もったいないが投げつけることも可能	restore strength	攻撃力を上限まで回復する
sleep	眠ってしまう	see invisible	目に見えないモンスター(P)を見たり、目が見えない状態から回復する
teleportation	同じ階のどこかにワープ。ワープ後しばらくは混乱する。	thirst quenching	喉の渇きを癒す
vortalize weapon	特定のモンスターに対して高い攻撃力と、魔法を使うことができるようになる	hallucination	幻覚症状を引き起こす
genocide	特定のモンスターを出現できなくする	cancellation	特殊攻撃や魔法を無効にする
protect armor	鎧を錆びなくする	cold	冷気を浴びせる
add strength	攻撃力を上げる	drain life	自分の体力(Hp)が半分になる
adornment	ただの飾り	fire	火炎を吹きかける
aggravate monster	眠っているモンスターを起こす	haste monster	モンスターの速度が2倍になる
dexterity	素早さを上げる	light	部屋を明るくする
increase damage	攻撃力を上げる	lightning	稲妻を浴びせる
maintain armor	鎧を錆びなくする	杖 magic missile	魔法による打撃を与える
protection	防御力を高くする	nothing	効果なし
指 regeneration	回復力を高める	polymorph	異なったモンスターに変える
輪 searching	隠された扉やわなを見つけやすくする	slow monster	モンスターの動きを遅くする
see invisible	目に見えないモンスターを見る	teleport away	近くのモンスターを遠くへ飛ばす
slow digestion	腹が減りにくくなる	teleport to	遠くのモンスターを近くへ呼び寄せる
stealth	眠っているモンスターを起さない	striking	杖で殴る
sustain strength	ストレンクスが下がらなくなる	invisibility	モンスターの姿を見えなくする
teleportation	ランダムにワープする		

Free Free Application Showcase

文：出井 一
Text : Hajime Dei



Open Universe P.156



Heroes P.158



GnomeKiss P.159

散布図などの科学技術系2次元グラフを描画する

SciGraphica



147

FTPクライアントソフトの定番ソフトがバージョン1.0に

IglooFTP PRO



150

アプリ起動中に砂時計カーソルを表示する

Xalf



153

ハードウェア情報の表示と設定を行う

HardDrake



154

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



156

華麗な画面とエフェクトの新世代ワームゲーム

Heroes



158

世界に広がる着せ替えシステムの輪

GnomeKiss



159

XMLを利用した音楽メディア用データベース

M3X



160

各種パスワードの暗号化と取り出し

Figaro's Password Manager



161

紹介したソフトは、すべて付録CD-ROMに収録されています。

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\(GtkExtra\)](http://magnet.fsu.edu/~feiguin/gtk(GtkExtra))

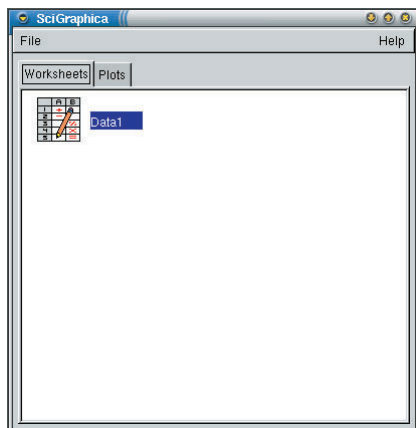
ビルドとインストール

SciGraphicaとGtkExtraは、ソース一式をtar + gzipしたtarボールのみ配布されている。GtkExtraは、マトリックス(グリッド)やプロット、ディレクトリツリーなどのウィンドウ部品を提供するライブラリ。SciGraphica 0.3の実行には0.99.6が必須だ。ビルドとインストールは、GtkExtra SciGraphicaという順番で行う。いずれも、「./configure」「make」「make install」という一般的な手順でOKだ。

RPMパッケージの作成も可能

両者のtarボールには、RPMパッケージ作成用のSPECファイルが含まれている。これらを利用してRPMパッケージを作成する場合、SPECファイルの内容に一部不備があるため、以下のように修正する必要がある。

- gtkextra.specの行末が「CR + LF」なので、すべて「LF」に修正。



画面1
ワークシートとプロットにページ分けされたメインウィンドウ。

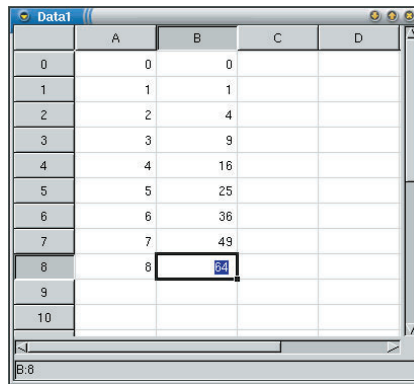
- sg.specの「Requires:」に記述されている「gtkextra」を「gtk + extra」に修正。
- sg.specの「%files」セクションの内容を以下のように変更。

```
%doc AUTHORS COPYING ChangeLog
NEWS README TODO
%{prefix}/bin/sga
%{prefix}/share/*/*/*
```

/usr/src/redhat以下のSPECSディレクトリに修正したSPECファイル、SOURCESディレクトリにtarボールをそれぞれコピーし、「rpm -ba SPECファイル名」とするとソースとバイナリのパッケージが作成される。あとは、バイナリパッケージを通常と同様にインストールすればいい。

ワークシートに数値を設定

「sga&」として起動すると、Sci



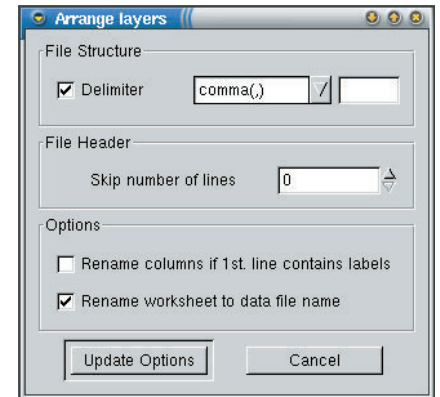
画面2
データの入力、表計算ソフト風のワークシートで行う。

SciGraphicaは、科学技術系2次元グラフ描画ソフトだ。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。

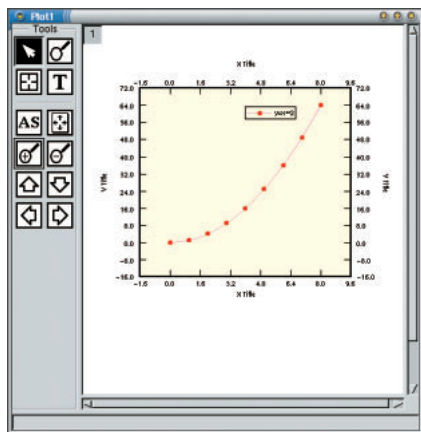
Graphicaのメインウィンドウが開く(画面1)。データ入力を行う「ワークシート」の一覧が表示される[Worksheets]ページと、グラフを描画する「プロット」の一覧が表示される[Plots]ページをタブで切り替えることが可能だ。

起動直後は、「Data1」(名前は変更可)というワークシートが用意されている。なお、SciGraphicaでは、プロジェクト内で複数のワークシートを扱える。追加は[File] - [New] - [Worksheet]で行う。

メインウィンドウのアイコンをダブルクリックすると、その内容が別ウィンドウに表示される(画面2)。ワークシートは表計算ソフト風のセルが縦横に並んだグリッド形式で、グラフの表示に利用する数値データを入力できる。サイズは5列(カラム)×20行(ロウ)だが、右クリックメニューで行・列の追加や挿入、削除が可能だ。



画面3
テキストファイルから読み込む場合のデリミタなどを設定する。



画面4

プロットウィンドウには、グラフと操作アイコンが表示される。

データのインポートも可能

このほか、外部のテキストファイルからワークシートにデータを読み込むこともできる。まず、ワークシート上で右クリックし、[File] - [ASCII Options]で設定ダイアログを開く（画面3）。ここでは、データを区切る文字（デリミタ）や、先頭を読み飛ばす行数などを設定可能だ。

インポートは、右クリックメニューの[File] - [Import]で行う。ディレクトリツリーとファイルリストを含むダイアログが開くので、[File Format]を[ASCII]に変更した後、読み込むファイルを指定しよう。

このほか、[File] - [Export]で保存した他のワークシートのデータをインポ

ートすることも可能だ。この場合は、データは専用のXML形式でやりとりされるので、デリミタなどを設定する必要はない。

プロットの作成

データの入力終了したら、プロットにグラフを描画する。起動直後は1つも用意されていないので、まずはプロットを追加しよう。メニューの[File] - [New] - [Plot]を選択すると、「Plot1」（名前は変更可）というプロットのアイコンが[Plots]ページに表示される。ワークシートと同様に、プロジェクト内で複数のプロットを扱うことも可能だ。

プロットのアイコンをダブルクリックすると、その内容が別ウィンドウとして開く（画面4）。ウィンドウの左には操作のアイコン、右側にはグラフが表示される。なお、プロットウィンドウの大きさを変更しても、グラフの大きさは追従しない。フィット用のアイコン（右列上から3番目）をクリックする必要がある。

データセットの作成

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SciGraphicaでは「データセット」と呼ぶ。1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1つのプロットで使用することも可能だ。

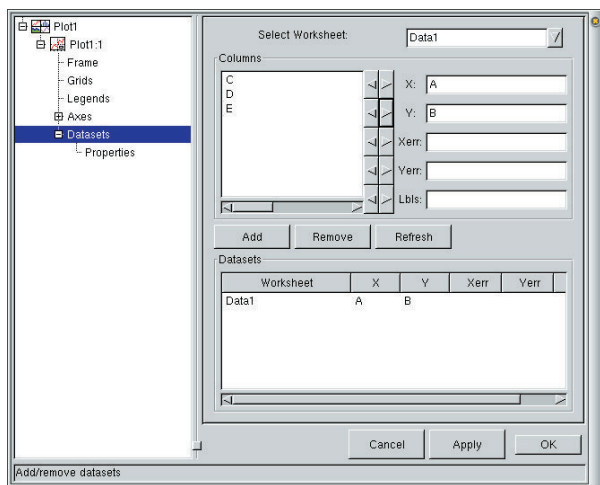
まずは、グラフ左上の[1]ボタンをダブルクリックして、レイヤコントロールダイアログを開く（画面5）。[Select Worksheet]で目的のワークシートを選択したら、その下のカラム一覧から対象となるカラムを選択し、[X:]や[Y:]の横にある右向き三角ボタンでX、Y軸との関連付けを行う。

[Add]ボタンを押すと、現在の設定でデータセットが追加され、ダイアログ下に表示される。[Apply]ボタンを押して、データセットの内容に基づいてプロットウィンドウにグラフが描画されるのを確認しよう。

また、ダイアログ右側のツリーで[Properties]をクリックすると、ダイアログの内容がデータセットごとのグラフの属性設定に切り替わる。データ間の線の結び方やデータ点の記号、ラベルのフォント、各部の色などさまざまな設定が可能だ（画面6）。

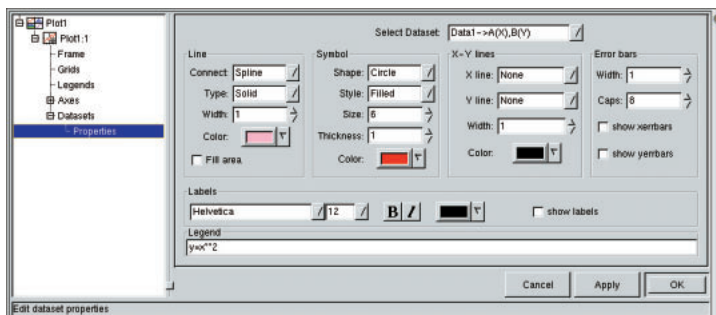
さまざまな設定を変更できる

レイヤコントロールダイアログでは、左側のツリーの項目をクリックして内



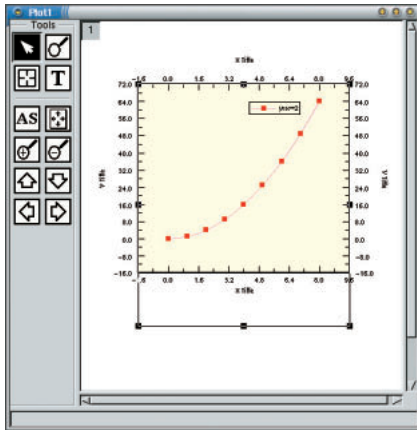
画面5

レイヤコントロールダイアログでデータセットを登録する。

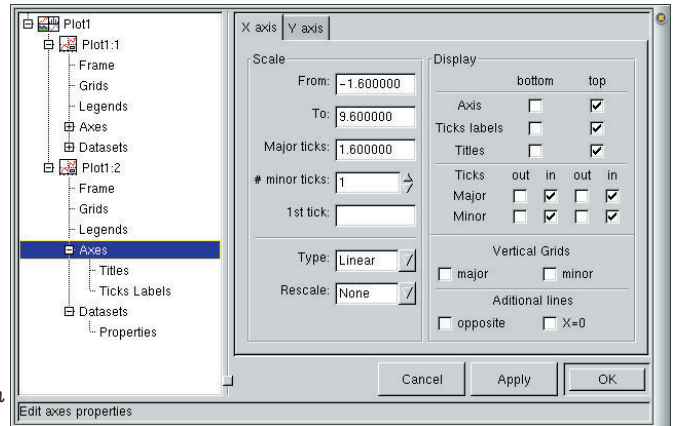


画面6

データセットごとに線の種類やデータ点の記号などを設定できる。



画面7
グラフの大きさは、枠をドラッグすることで直接変更できる。



画面8
複数のレイヤでそれぞれ枠や軸の設定が可能だ。

容を切り替えることで、さまざまな設定を行える。

たとえば、ツリー最上部の[Plot1]では、グラフを描画する紙のサイズや向き、その下の[Plot1:1]では、グラフの背景色などを変更できる。なお、この「Plot1」はプロット名の初期値で、ユーザーにより変更可能だ。以下、[Frame]ではグラフの周囲の枠に関する設定、[Grids]ではグリッドに関する設定、[Axes]ではX、Y軸に関する設定、[Legends]では凡例に関する設定を行える。

なお、こうした設定の中には、プロットウィンドウで直接変更できるものもある。たとえば、グラフの大きさを変えるには、グラフをクリックし、四隅と四辺に表示される をドラッグすればいい(画面7)。また、各軸の表示範囲やスケールについては、ウィンドウ左に並ぶアイコンをクリックするだけで変更できる。

複数のレイヤを使用する

これまでの方法でも、1つの枠内に同一スケールで複数のグラフを表示できる。しかし、いくつかのグラフを並べたり、スケールの異なるグラフを重ねたりするには、複数のレイヤを利用する必要がある。

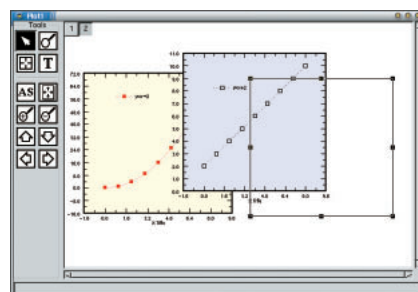
プロットに新しいレイヤを追加する

には、グラフ上で右クリックし、[Layer] - [New Layer]を選択する。グラフ左上に追加された[2]ボタンをダブルクリックしてレイヤコントロールダイアログを開き、データセットなどの設定を行おう。

レイヤコントロールウィンドウのツリー中では、各レイヤは[プロット名:レイヤ番号]という形式で表示される(画面8)。枠や軸、データセットの設定はレイヤごとに独立しているため、異なるスケールで複数のグラフを表示できるわけだ。

異なるスケールのグラフを同じ位置に重ねる場合は、軸のラベルや数値の表示が重ならないように注意しよう。表示の制御は、各レイヤのツリーの[Axes]で行える。レイヤ番号が大きいものが手前に来るので、背景色を設定する場合には注意されたい。

一方、複数のグラフを別の位置に表



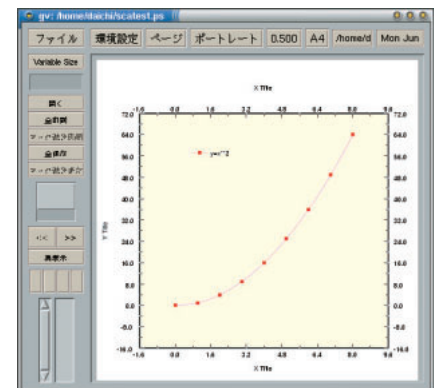
画面9
複数のグラフを並べて配置するには、ドラッグを利用するとよい。

示するには、レイヤコントロールダイアログで設定を変えるより、グラフを直接ドラッグしたほうが簡単だ。動かしたいグラフをクリックで選択したあと、目的の位置までドラッグで移動すればいい(画面9)。

グラフの保存と印刷

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、メニューの[Files] - [Save] ([Save As])でXML形式でファイルに保存できる。

一方、作成したグラフを他のアプリで利用したり、プリンタで印刷する場合には、グラフ上で右クリックし、[File] - [Exports]を選択してPostScript形式で保存する。出力されたPSファイルは、gvなどを使ってプレビューや印刷が可能だ(画面10)。



画面10
PostScript形式で保存したグラフを、gvでプレビュー。

FTPクライアントソフトの定番ソフトがバージョン1.0に

IglooFTP PRO

バージョン: 1.0.0

ライセンス: シェアウェア (29.95USドル)

<http://www.littleigloo.org/iglooftp.php3>

IglooFTP PROは、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だ。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。なお、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。動作にはGTK+が必要だ(1.2.6以降を推奨)。

インストール

IglooFTP PROはシェアウェアのため、ソースファイルは公開されておらず、RPM形式およびtarボールのバイナリパッケージが配布されている。このほか、シェアウェアになる前の古いIglooFTPのソースがtarボールで配布されているが、PROと比べると大幅に機能が劣るため、あまりお勧めできない。

Red Hat系ディストリビューションを使っている人はRPMパッケージを使おう。「rpm -Uvh IglooFTP-PRO-1.0.0-1.i386.rpm」としてインストールすると、/usr/local/IglooFTP-PRO以下に実行ファイルや各種データがまとめてインストールされる。/usr/local/binに実行ファイルのシンボリックリンクが作成されるので、各ユーザーのPATHの設定に/usr/local/binが含まれるようにしておこう。

一方、tarボールのほうは、glibc2系ディストリビューション用とlibc5系ディストリビューション用の2種類用意さ

れている。自分の環境に合ったほうを適当なディレクトリに展開し、スーパーユーザー (root) になった状態で「./Install」とすると、/usr/local/IglooFTP-PRO以下にインストールが行われる。

起動と初期設定

ktermなどのコマンドラインで「IglooFTP-PRO&」として起動すると、多数のペイン(領域)に分割されたウィンドウが開く(画面1)。操作の中心となるのは、中央部の左右のペインで、左にはローカル側のファイル一覧、右にはFTPサイトのブックマークやリモート側のファイル一覧などが表示される。各ペインのサイズはマウスをドラッグすることで自由に変更可能だ。

まずは初期設定を行う。ツールバー左から4番目の[Set Preferences]ボタンを押すと、ユーザー設定ダイアログが開く(画面2)。このダイアログはカテゴリ別にページ分けされており、IglooFTP PROのあらゆる設定をこの

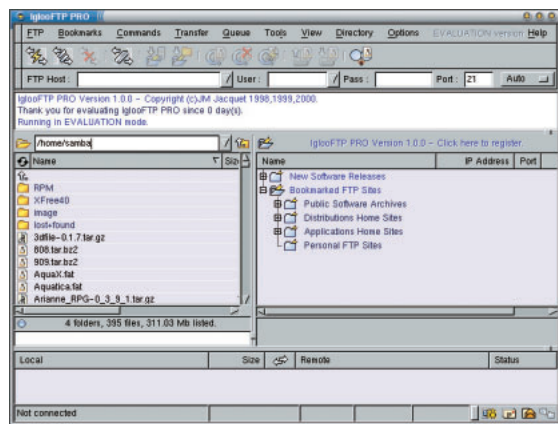
ダイアログで変更できる。

とりあえず、先頭の[General]ページで、ユーザー名とパスワード、ローカル/リモートのディレクトリの初期値を設定しておこう。通常は、ユーザー名として「anonymous」、パスワードにはあなたのメールアドレスを設定しておけばいい。

ブックマークの登録

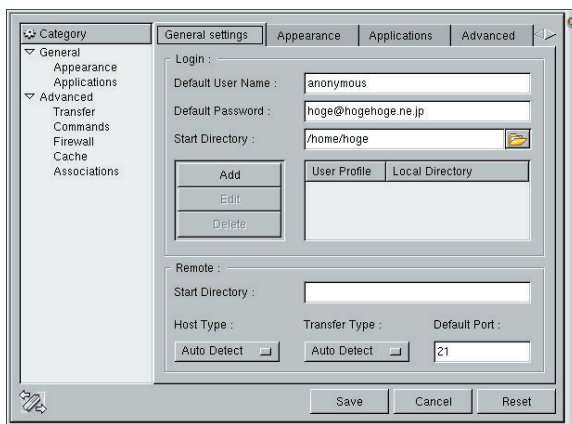
FTPサイトに接続していない状態では、右のペインにはFTPサイトのブックマークと、IcwWalker.comが提供する新着ソフト情報がツリー表示されている。ブックマークは、グループ別のフォルダで分類され、Red HatやDebianなどのディストリビューション、GTK+やGNOMEプロジェクトのFTPサイトなどが登録されている(画面3)。

もちろん、ユーザーが新たにFTPサイトのブックマークを登録することもできる。ツリー中のフォルダをクリックし、右クリックメニューの[Add



画面1
ウィンドウ内は複数のペインに分割され、ファイル一覧をはじめとしてさまざまな情報が表示される。

画面2
全設定をダイアログで変更できるため、設定ファイルをエディタで編集する必要はない。



Name	IP Address	Port	User
New Software Releases			
IceWalkers.com			
Bookmarked FTP Sites			
Public Software Archives			
SunSite	sunsite.unc.edu	21	anonymous
TSX-11	tsx-11.mit.edu	21	anonymous
GNU Archive	prep.ai.mit.edu	21	anonymous
Distributions Home Sites			
RedHat (Official)	ftp.redhat.com	21	anonymous
RedHat (Sunsite mirror)	sunsite.unc.edu	21	anonymous
Debian	ftp.debian.org	21	anonymous
Slackware	ftp.cdrom.com	21	anonymous
Applications Home Sites			
Personal FTP Sites			

画面3
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。

Site]を選択しよう。サイトマネージャダイアログが開くので、FTPサイト名やアドレス、ユーザー名、パスワードなどを設定する。

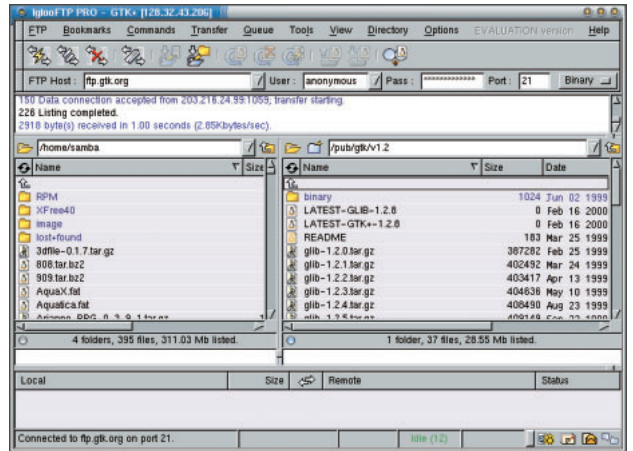
パスワードについては、「Save Password」(設定して保存)、「Use Default Email Address」(初期設定のメールアドレスを使用)、「Ask for Password on Connect」(接続時に入力)の中から選択できる。誰でも接続できるアノニマスFTPサーバの場合は、「Use Default...」にしておけばいい。

なお、一番下にある[Set Options Applying...]を押すと、ダイアログに[Advanced]ページが追加され、接続時の初期ディレクトリなどをサイトごとに設定できる。

接続と基本的な操作

ブックマークに登録したFTPサイトは、ダブルクリックするだけで接続処理が行われる。また、一時的なアクセスなら、ホスト名やアカウントなどを

画面4
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する。



ツールバー下のFastConnectバーに入力し、ツールバー左端の[Connect/Reconnect]ボタンを押せばいい。いずれにせよ、接続情報が上部のコンソールペインに表示され、FTPサイトに接続するとウェルカムメッセージが内蔵ビューアのウィンドウに表示される。

FTPサイトに接続中は、右のペインがリモート(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(画面4)。ディレクトリの切り替えは、サブディレクトリのダブルクリックやディレクトリ名の直接入力で可能。このほか、ディレクトリツリー表示に切り替えたり、ひとつ上のディレクトリに移動するボタンも用意されている。

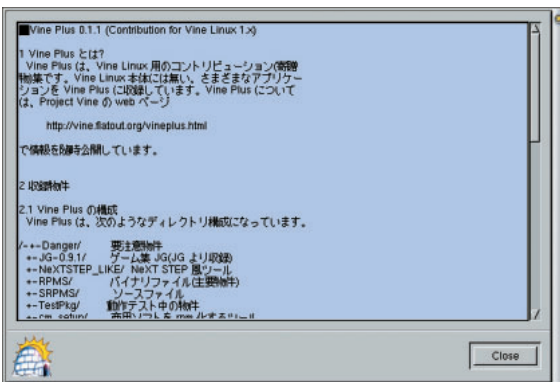
ファイルを選択するには、単にマウスでクリックすればいい。Shift - クリ

ックやドラッグによる範囲選択、Ctrl - クリックによる複数ファイルの選択にも対応している。

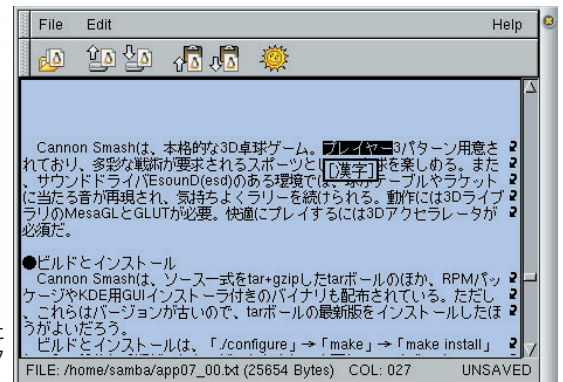
ファイルの閲覧や編集も可能

ローカル/リモートどちらのファイルに対しても、右クリックメニューの項目を選択するだけで、選択したファイルの閲覧や編集、削除、属性変更、ディレクトリの作成といったファイル操作を行える。

初期設定では、ファイルの閲覧やウェルカムメッセージの表示には内蔵ビューア(画面5)、ファイルの編集には内蔵エディタ「IglouEdit」(画面6)が使われる。ソースが提供されないため、日本語への対応が気になるところだが、日本語環境ではどちらも日本語EUCで書かれたファイルを問題なく扱えるので安心されたい。



画面5
ファイル閲覧やウェルカムメッセージの表示に使われるビューア。



画面6
日本語の表示や入力に対応しているエディタ「IglouEdit」。



画面7

同名ファイルがすでに存在する場合は、ここで処理を選択する。

なお、IglOOEditの機能に満足できないなら、XEmacsやviなどの使い慣れた外部エディタを利用しよう。設定ダイアログの[Application]ページの[Text Editor]に、エディタの実行ファイルを設定すればいい。このほか、ウィンドウ右下のボタンで起動するWebブラウザやメールクライアントもこのページで設定できる。

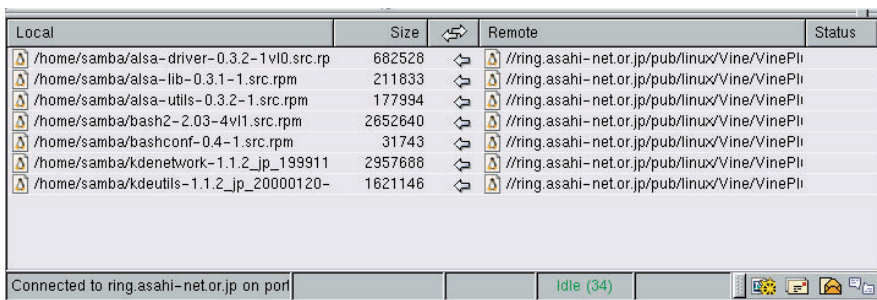
2種類のファイル転送方法

FTPクライアントの要といえるファイル転送に関しては、2種類の方法が用意されている。ひとつは、選択したファイルを即座に転送する方法、もうひとつは、ファイルの情報をいったん「キュー」に溜めておき、あとでまとめて一括転送する方法だ。

その場でファイルを送受信したい場合は最初の方法を利用する。ファイルを選択した状態で、ツールバーの[Upload Selection] / [Download Selection]ボタンを押すか、右クリックメニューから[Upload] / [Download]を選択すればいい。

転送中は、ウィンドウ最下行のステータスバーに、ファイル名や転送状況を示すグラフ、転送速度、受信時間、残り時間などが表示される。

なお、同名のファイルがすでに転送先に存在する場合は、ダイアログが表



画面8

一括転送用のキューにファイル名などが表示される。

示されるので、上書きやリネーム、スキップなどの処理を選択する(画面7)。もちろん、ファイルの転送を中断した箇所から再開するレジュームにも対応している。

キューを利用した一括転送

ファイルの選択と受信に間が空く場合はキューを利用した一括転送を利用しよう。対象となるファイルを選択し、ツールバーの[Add to Queue]ボタンでキューに登録する。

現在のキューの内容はウィンドウ下部のキューウィンドウに表示され(画面8)、ツールバーの[Transfer Queue]ボタンを押すと、一番上のファイルから順に転送が開始される。ファイルの順番を変更したり、キューから削除することも可能だ。

キューの内容は、FTPサイトとの接続を終了しても削除されず、[Transfer Queue]ボタンを押した時点でFTPサイトへの接続が自動的に行われる。このため、複数のFTPサイトを巡回して興味のあるファイルを次々にキューに登録し、後ほど一括して受信するという使い方も可能だ。

なお、キューの内容はIglOOFTP PROを終了すると失われるため、キューの内容をファイルに保存したり、ファイルから読み込む機能が用意されている。転送を行う前にIglOOFTPを終了しなくてはならない場合は、これら

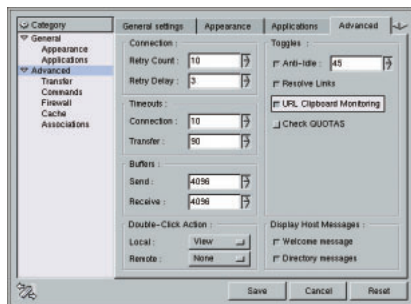
の機能を利用するとよい。

クリップボードの監視も可能

IglOOFTP PROには、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が用意されている。

たとえば、IglOOFTP PROを実行中に、NetscapeでFTPへのリンクを右クリックし、[リンクの場所をコピー]を選択すると、リンクのURLで指定されたFTPサイトのディレクトリが自動的に表示される(ファイルの受信は手動で指示する必要がある)。

この機能は、設定ダイアログの[Advanced]ページにある[URL Clipboard Monitoring]で設定する(画面9)。初期値はオンなので、必要ない場合はオフに変更しておこう。このほか、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」機能なども用意されている。



画面9

クリップボード監視機能なども用意されている。

アプリ起動中に砂時計カーソルを表示する

Xalf

バージョン : 0.3

ライセンス : GPL

<http://www.lysator.liu.se/~altic/projects/xalf/>

ビルドとインストール

Xalfは、ソースのtarボールとRPMのソース/バイナリパッケージが配布されている。Red Hat系ディストリビューションでは、RPMバイナリパッケージを「rpm -Uvh xalf-0.3-1.i386.rpm」としてインストールする。tarボールのほうも、「./configure」「make」「make install」という一般的な手順でOKだ。

コマンドラインの頭に付けるだけ

使い方は簡単で、実行したいコマンドラインの先頭に、「xalf」とフィードバック方法を示すオプション(-c、-s、-i)を付けるだけでいい。たとえば、-cオプションを使って、

```
$ xalf -c freeamp&
```

とすると、Freeampの起動中はマウスカーソルの形状が砂時計に変わり、起

動後は元に戻る。

また、-sオプションを指定するとスプラッシュスクリーンのウィンドウが開き(画面1)、-iオプションではGNOME/KDEのタスクリストに「(freeamp)」のようにカッコ付きでアプリ名が表示される(画面2)。これらのオプションを省略した場合は、-iオプションを指定したものと見なされる。

X Window Systemでは、アプリ起動中もマウスカーソルの形状が変化しないため、ウィンドウが開くまでに時間がかかる「重い」アプリを起動する際に、起動に失敗したと勘違いして再度起動してしまうことがある。Xalfを使えば、砂時計カーソルなどのフィードバックにより、こうした失敗を避けることができるわけだ。

問題が生ずる場合はxalfoff

一部のラッパースクリプトをXalfで起動すると、ウィンドウが開いている

Xalfは、X用アプリ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だ。ウィンドウが開くまで時間のかかる大きなアプリを誤って何回も起動してしまう失敗を防ぐことができる。フィードバックの方法としては、砂時計カーソルのほかに、スプラッシュスクリーンの表示、GNOMEやKDEのタスクリストにカッコ付きでアプリ名を表示、と計3種類用意されている。

のにタイムアウトになるといった問題が生ずる。たとえば、Red Hat系ディストリビューションで提供されるNetscape起動用スクリプト(/usr/bin/netscape)がこれに該当する。

こうした場合、スクリプトで実行ファイルを起動している行の直前に「xalfoff」という1行を追加すればいい(netscapeの場合、サンプルがXalfのパッケージに付属する)。

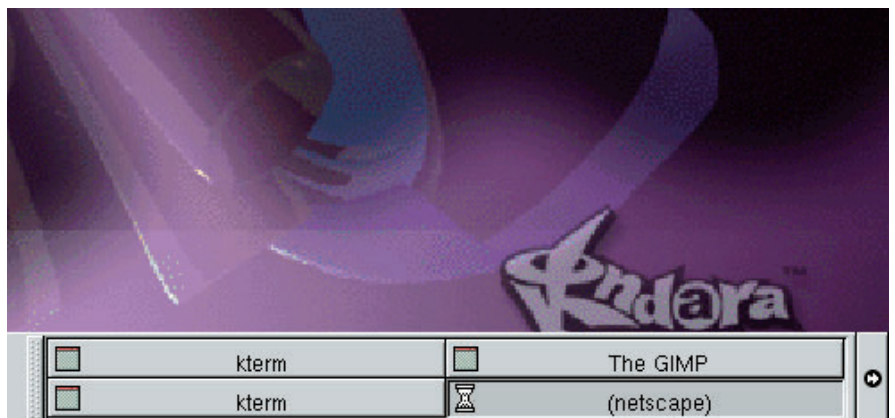
なお、GNOMEの最新版「GNOME 1.2」では、Xalfがビルトインサポートされ、いちいち先頭に「xalf」をつけなくてもよくなった。Xalfの機能を有効にするには、あらかじめ「export GNOME_USE_ALF=1」などとして、環境変数GNOME_USE_ALFを設定する必要がある(値はなんでもよい)。

また、KDEの最新版「KDE2」でもXalfと同様の仕組みが導入されるようだ。



画面1

-sオプションではスプラッシュスクリーンを表示する。



画面2

-iオプションではタスクトレイにカッコ付きでアプリ名を表示。

各種ハードウェア情報の表示と設定を行う

HardDrake

バージョン: 0.7

ライセンス: GPL

<http://www.linux-mandrake.com/harddrake/>

ビルドとインストール

HardDrakeは、本体の「Hard Drake」以外に、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」で構成されている。このうち、sounddrakeとetherdrakeについては、必要がなければインストールしなくてもよい（ALSAを使っていない場合など）。

いずれも、tarボールとRPMのソース/バイナリパッケージが配布されている。ただし、RPMソースパッケージをリビルドするのは大変なので（SPECファイルの記述に不備があるようだ）tarボールかRPMバイナリパッケージの導入をお勧めする。

(1) RPMバイナリパッケージの場合

```
detect HardDrake sound
```

drake / etherdrakeの順に「rpm -Uvh パッケージ名」としてインストールする。なお、detectにはisapnptools 1.21以降、etherdrakeにはnewt 0.50.8以降、sounddrakeにはALSA 0.5.7以降、sox 12.16以降、newt 0.50.8以降、modutils 2.3.9以降が必要だ。

たとえば、Kondara 1.1では、isapnptoolsとnewtをFTPサイトにある最新版のRPMパッケージで差し替える。そして、sounddrakeのインストール時に「--nodeps」オプションを付ける必要がある（ALSAドライバがカーネルパッケージに含まれるため）。

(2) tarボールからビルドする場合

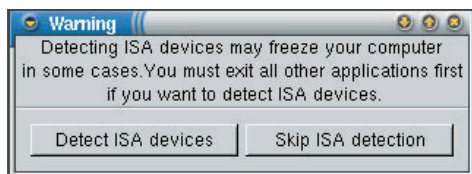
RPMバイナリパッケージと同じ順番でビルドとインストールを行う。いずれもconfigureスクリプトを利用した一般的な手順だ。

HardDrakeは、ハードウェア情報の表示や設定をX上で行うソフトだ。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系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、ALSAのサウンド設定と、イーサネットの設定を行うモジュールが作られている。動作にはGTK+が必要だ。

ただし、HardDrakeのビルドを行う前に、モジュールのパスを修正するパッチを当てておこう。さらに、etherdrakeは「./configure --sbindir=/usr/sbin」、sounddrakeは、「sh autogen.sh --sbindir=/usr/sbin」として、インストール先を/usr/sbinに変更する必要がある。

起動とハード情報の表示

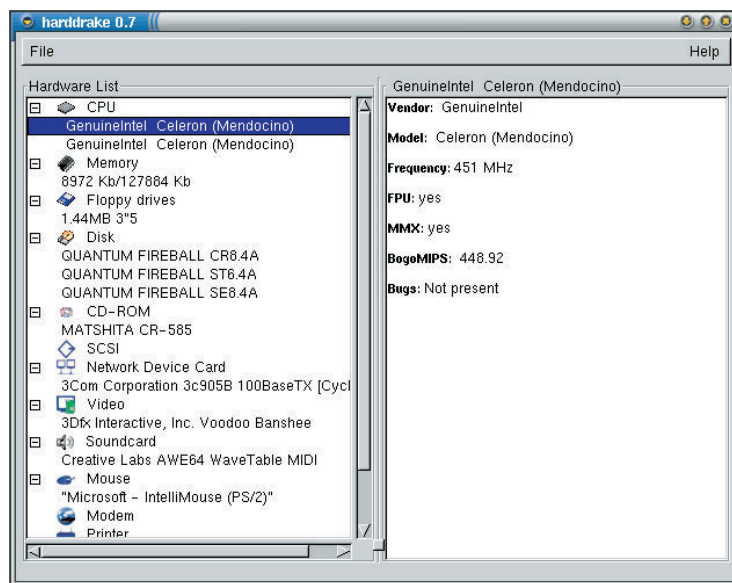
スーパーユーザー（root）になった状態で「harddrake&」として起動すると、最初にISAデバイスを検出するかどうか尋ねられる（画面1）。ISAデバイスの検出時にフリーズする可能性があるためだ。ISAデバイスを検出する場合は[Detect ISA devices]ボタン、検出を回避する場合は[Skip ISA detection]ボタンを押そう。



画面1
最初に、ISAデバイスを検出するかどうか尋ねられる。



画面2
スプラッシュ画面を表示しながらハード情報を検出する。



画面3
左にはハード情報の一覧、右にはハードの詳細な情報を表示。

検出中はスプラッシュウィンドウが表示され（画面2）、しばらくするとHardDrakeのウィンドウが開く。表示は左右のペインに分かれている。左ペインにはCPU/メモリ/ネットワークカード/サウンドカードなど、ハードウェアの分類と検出されたハードがツリー表示されている。右ペインにはツリーをクリックすることによって選択された、ハードウェアに関する詳しい情報が表示される（画面3）。

ネットワークの設定を行う

ネットワークカードのハードを選択した場合は、右側のペインに詳しい情

報が表示されるだけでなく、その下に[Run Configuration tool]ボタンが表示される（画面4）。

このボタンを押すと、イーサネット設定モジュールetherdrakeが起動され、設定を変更するかどうか尋ねられる（画面5）。設定を変更する場合は[OK]ボタンを押そう。etherdrakeのウィンドウが開いてネットワークカードの一覧が表示される（画面6）。

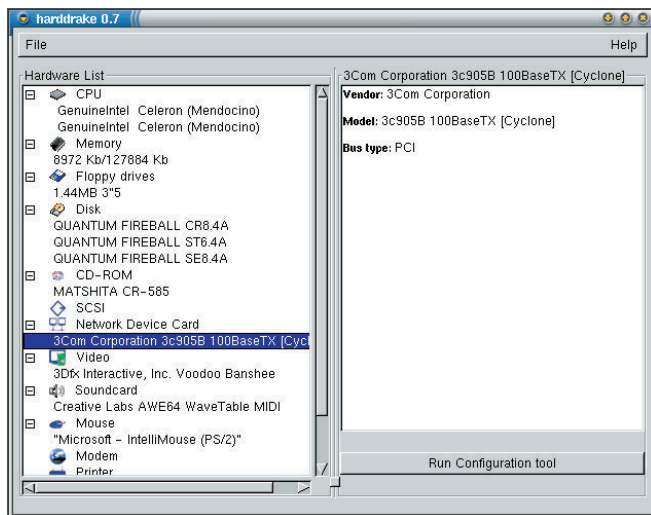
PCIデバイスの場合は、一覧中から自分のカードを選択するだけでいい。ISAデバイスの場合は、I/OアドレスやIRQの設定も同時に行う。[OK]ボタンを押すと、/etc/conf.modules内のネットワ

ークカードの設定部分に変更される。あらかじめ、/etc/conf.modulesのパックアップをとっておくといいだろう。

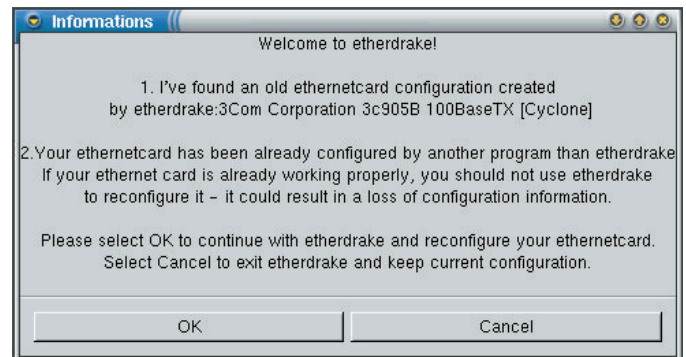
サウンドの設定を行う

サウンドカードに関する設定もネットワークカードの場合とほぼ同じで、[Run Configuration tool]ボタンを押すと、サウンドカード設定モジュールsounddrakeが起動される。

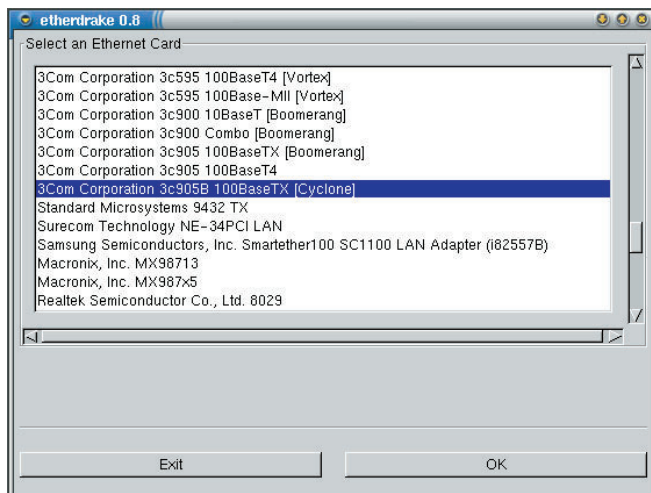
ウィンドウにはサウンドカードの一覧が表示されるので、自分のカードを選択すればいい（画面7）。ネットワークカードの場合と異なり、ほとんどのカードでI/OアドレスやIRQ、DMA（8ビットおよび16ビット）などの設定も同時に行う必要がある。



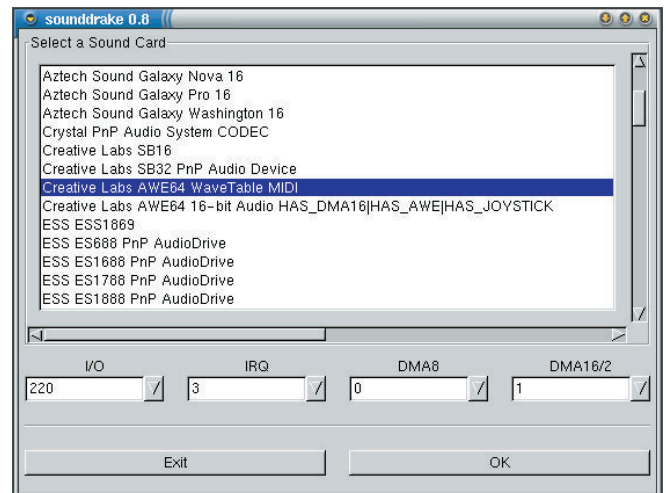
画面4
ネットワークカードのハード情報の下にボタンが表示される。



画面5
etherdrakeで設定を変更する場合は[OK]ボタンを押そう。



画面6
ネットワークカードの設定を行うetherdrake。



画面7
サウンドカードの設定を行うsounddrake。

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/>
<http://www.mesa3d.org/> (Mesa)

ビルドとインストール

Open Universeは、ソースや画像データなどをtar + gzipしたtarボールのみ配布されている。ビルドとインストールは、「./configure」「make」「make install」という一般的な手順でOKだ。なお、3DライブラリのMesa 3D (3.2を推奨)とGLUTは、Open Universeをビルドする前にあらかじめインストールしておく必要がある。さらに、GLUIライブラリ (2.0以降) を利用するとGUIによる操作が可能だ。

このほか、地球・火星・木星の衛星イオのハイレゾ画像が別アーカイブ (ouhires.zip) に収録されている。これらを利用するには、標準画像をハイレゾ画像で直接上書きすればいい。なお、ハイレゾ画像を使用した場合は、Open Universeのメモリ消費量が大幅に増えるので注意されたい。

まずはデモモードで楽しむ

「openuniverse&」として起動すると、数種類あるスプラッシュ画面 (画

面1) のひとつを表示しながらデータが読み込まれ、地球と使用キーの説明が表示される。

操作には、通常はキーボードを利用する。各キーに機能が割り当てられており (リスト1)、慣れてくると素早い操作が可能だ。

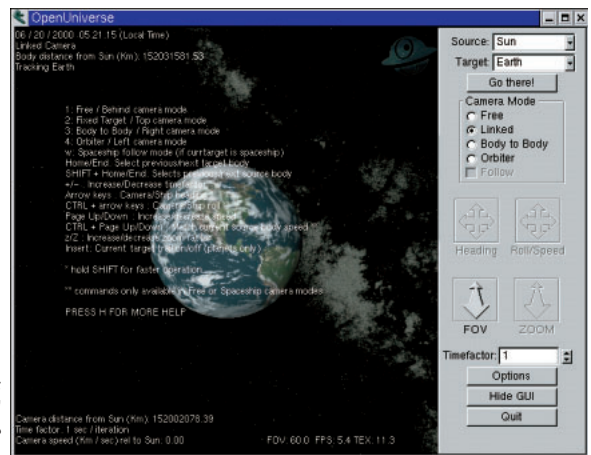
また、GLUIライブラリと組み合わせた場合は、ウィンドウ右側に操作用のパネルが表示され、マウスでも操作できる (画面2)。

まずは、ユーザーが特に操作をする



画面1

起動時の画面は数種類用意されており、毎回ランダムにそのうちのひとつが選択される。



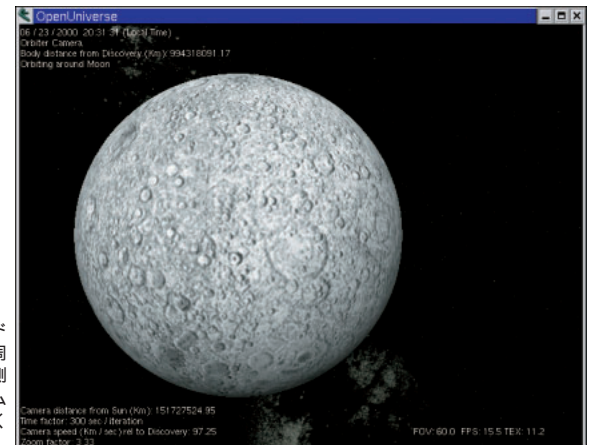
画面2

GLUIライブラリを利用すると、マウスを使った操作も可能になる。



画面3

デモモードでは一定時間ごとに自動的に表示対象が切り替わる。



画面4

「Orbiter」モードで月の軌道上を周回しつつ月の裏側を眺める。タイムスケールを大きくするといい。

必要のない「デモモード」で概要を掴むとよいだろう。デモモードへの切り替えにはdキーを使用する（終了も同じ）。一定時間ごとに表示対象（天体あるいは宇宙船）が切り替わり、自動的にカメラが適切な位置に移動する。美しいテクスチャの張られた惑星の眺めなどを楽しもう（画面3）。

キーボードでカメラを切り替える

デモモードを抜けて、自分の思い通りの画面を表示してみよう。まずは表示対象（Target）となる天体・宇宙船を、HomeキーとEndキーで選択する。画面左上の4行目に表示される対象名を参考にしよう。

カメラモードは1~4キーで切り替える。カーソルキーで角度を変更できる「Free」（1キー）、常に対象に向く「Linked」（2キー）、他の天体・宇宙船から眺める「Body To Body」（3キー）、軌道上から眺める「Orbiter」（4キー）の4種類だ（画面4）。

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できる。元の状態に戻すにはeキーを押す。このほか、惑星や恒星の名前を表示したり（画面5）

近くの天体への向きと距離をマップ表示することも可能だ。

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、Sキーでスナップショットを撮ろう。JPEG形式の画像が/usr/local/share/openuniverse/scrshotsディレクトリに保存される。なお、このディレクトリはあらかじめスーパーユーザーとなって作成し、一般ユーザーの書き込みを許可しておく必要がある。

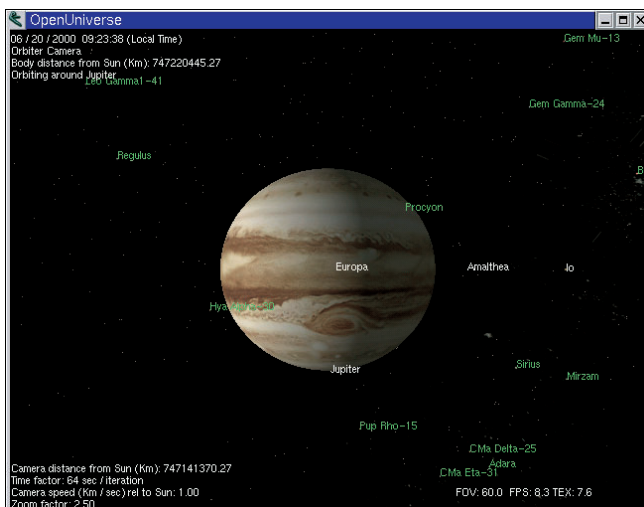
宇宙船を操縦する

Open Universeには、スペースシャトルのエンデバー、アポロ13号、ハッブル望遠鏡（HST）、ステーションVという人工天体のデータも含まれている（画面6）。これらは、単に角度を変えて表示するだけでなく、ユーザーが操縦することも可能だ。

具体的には、Home/Endキーで表示対象をこれらの人工天体にした後、wキーを押して「カメラ追尾モード」に切り替える。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。再度wキーを押すと元の状態に戻る。

基本的な操作	
1~4	カメラモード設定
w	カメラ追尾モード（宇宙船）
d	デモモード
h/H	オンラインヘルプの表示
i	情報表示
Esc	Open Universeの終了
カメラ / 宇宙船の操作	
Home / End	対象の切替
Shift - Home / End	観察位置の切替（モード2）
n	対象に接近（モード1、2）
	向きの変更
Ctrl -	ロール
PageUp / Down	スピード増減
+ / -	時間スケール増減
e	実時間に設定
p	ポーズ
S	スクリーンショット
j	ジョイスティック操作
表示に関する操作	
Ins	対象の軌道表示
b	ラベル表示
B	星のラベル表示
G	ロゴ表示
t	テクスチャ表示
l	ライト
f	スムージング
a	大気表示
r	マップ表示
s	恒星の表示
m / M	恒星の明るさを増減
o / O	表示領域の増減
u / U	マップレンジの増減
y	重力（宇宙船のみ）
z / Z	ズーム増減（モード2、3）

リスト1 操作キーの一覧



画面5

惑星や恒星の名前を重ねて表示することも可能だ。



画面6

カメラ追尾モードでスペースシャトル「エンデバー」を操縦する。

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: 0.2

ライセンス: GPL

<http://www.realtech.scene.org/heroes/>
[http://www.ggi-project.org/\(GGI/GII\)](http://www.ggi-project.org/(GGI/GII))
[http://mikmod.darkorb.net/\(LibMikMod\)](http://mikmod.darkorb.net/(LibMikMod))

ビルドとインストール

Heroesは、ソース一式をtar + gzipしたtarボールのみ配布されている。ビルドとインストールは、「./configure」「make」「make install」という一般的な手順だ。

実行に必要なGGI / GIIライブラリは、Heroesをビルドする前にインストールしておこう。サウンド機能を使う場合はLibMikModも必要だ。

なお、Heroesの画面表示サイズは320x200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いいだろう。

ワームを操作してアイテムを取れ

「heroes&」として起動すると、凝った画面エフェクトのタイトルが表示される(画面1)。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2)。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

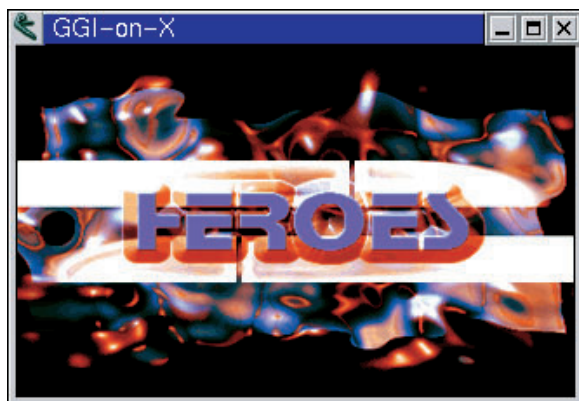
加速できる。

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。

対戦プレイやデスマッチも可能

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人間同士の対戦が可能になる(画面3)。左側のプレイヤーはe/s/d/xキーでワームを操作しよう。

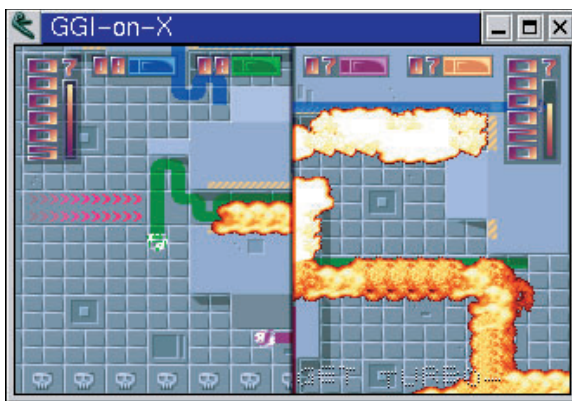
また、最後まで生き残ったものが勝者となる「DEATH MATCH」や、制限時間内に自分と同色の宝石を数多く集めたものが勝者となる「COLOR」など、全部で5種類のゲームモードが用意されている。



画面1
派手なBGMとともにタイトル画面が表示される。



画面2
カーソルキーでワームを操作して、各種のアイテムを取ってゆく。



画面3
2人同時プレイの場合は、画面が左右に分割表示される。

世界に広がる着せ替えシステムの輪

GnomeKiss

バージョン: 0.3

ライセンス: GPL

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

ビルドとインストール

GnomeKissは、ソース一式をtar + gzipしたtarボールでのみ配布されている。ビルドとインストールは「./configure」「make」「make install」という一般的な手順だ。

なお、KISS用データとして、今回の本誌付録CD-ROMには、

- ・「きせかえサキちゃん」(s_kssaki.lzh、しょうちゃん氏作、原画山いもとるとる氏)
 - ・「ぢなるネコっばい(*>_<*)むしゅめ」(slmmknn3.lzh、SiLent's MerMaid氏作)
 - ・「分割パズル壊れたチェス盤」(fk_cpzl2.lzh、kazMaster氏作)
- の3データを収録している。

これらは、適当なディレクトリ(/KISSなど)を作成して展開する。Windows用のKISSローダは、lhaでアーカイブされたファイルの内容を直接参照できるが、GnomeKissにはそのよ

うな機能はない。ファイル名の重複を考慮し、データごとにサブディレクトリに分けるとよいだろう。

着せ替えからパズルまで

「gnomekiss&」として起動し、[ファイル]-[開く]で設定ファイル(拡張子は.cnf)を選択すると、対応するKISS画像が表示される。起動時のコマンドラインで設定ファイル名を指定することも可能だ。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(画面1)。

あとは、マウスのドラッグを利用して、自由に服や靴、アクセサリなどを着せ替えればいい。「ぢなるネコっばい(*>_<*)むしゅめ」(画面2)のように、あらかじめ設定された複数の「ビュー」を0~9キーで切り替えられたり、複数の設定ファイルを用意しているデータもある。

さらに、French KISSの拡張機能を

GnomeKissは、日本発の着せ替えシステム「KISS」の拡張仕様である「French KISS 1.0」に対応したGNOMEアプリだ。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、人物(たいがいは女の子)の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。動作にはGNOMEとGTK+が必要だ。

利用すると、単なる着せ替えにとどまらないインタラクティブなコンテンツを作成可能だ。たとえば、「分割パズル壊れたチェス盤」(画面3)では、プルダウンメニューやポップアップウィンドウ、ボタンなどが用意され、チェス盤にピースをはめ込むパズルを実現している。

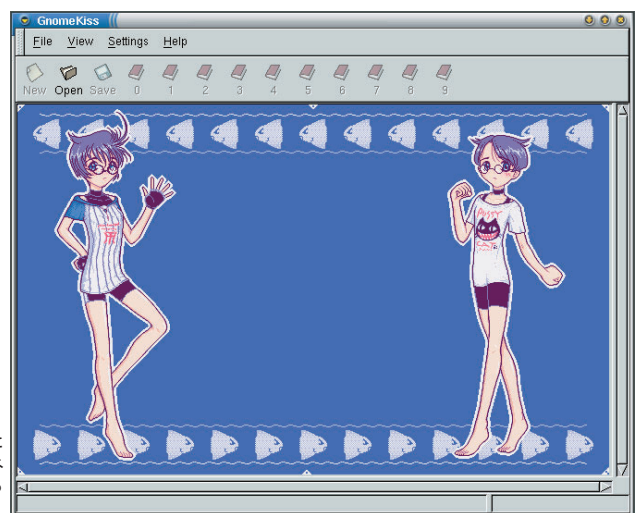


画面3 ポップアップウィンドウなどを駆使した「壊れたチェス盤」



画面1 今回のデータでは最もシンプルな「きせかえサキちゃん」

画面2 雰囲気はなんとも言えない「ネコっばいむしゅめ」



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: 0.1.7

ライセンス: GPL

<http://madhouse.lonyay.edu.hu/m3x/>

ビルドから実行まで

M3Xは、ソース一式をtar + gzipしたtarボールでのみ配布されている。ビルドとインストールは、「./configure」「make」「make install」という一般的な手順だ。ただし、/bin/shがbashバージョン2以降にリンクされていないと、configureスクリプトが正常に実行されないのに注意されたい。

「m3x&」として起動すると、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)。曲目については、[Add]ボタンで開くダイアログを使ってひとつずつ追加する(画面3)。

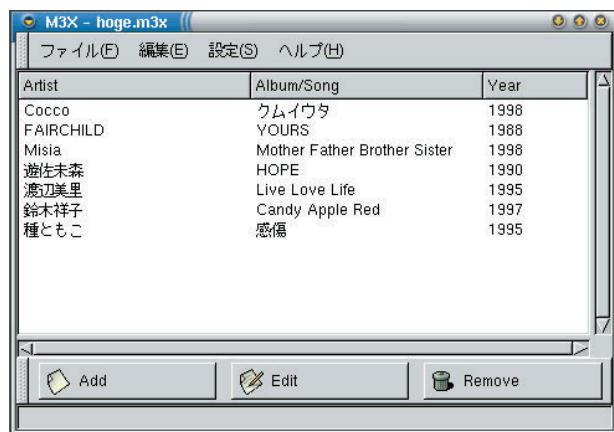
作成したデータは、[ファイル] - [Save database]を選択して、XML形式で保存しておこう。次回からは[ファイル] - [Open database]で読み込むことができる。

コンソールベースでの操作

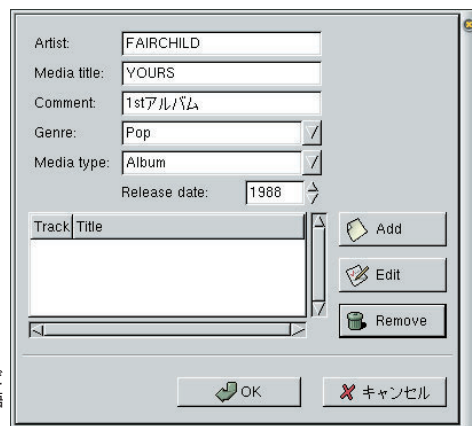
M3Xには、コンソールベースでデータを操作する「m3x-text」も用意され

ており、[ファイル] - [Launch m3x-text]で起動できる(画面4)。m3x-textでは、loadや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。

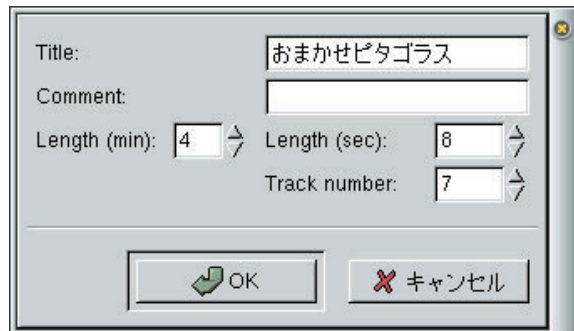
ただし、日本語を含むデータを登録している場合、内蔵ターミナルでは文字化けしてしまうので、外部ターミナルを利用する必要がある。[設定] - [設定]でダイアログを開き、[M3X-Text Preference]ページでktermを利用するように変更すればいい。



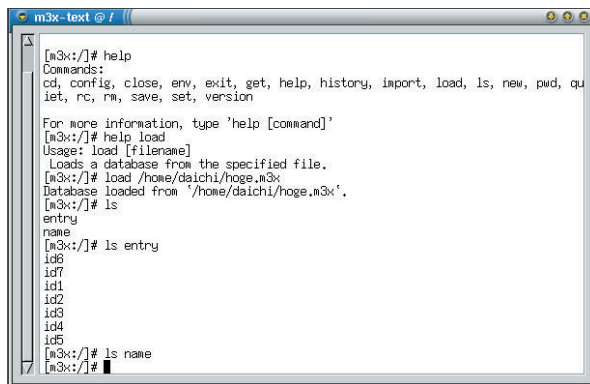
画面1
ウィンドウにはアーティスト名などが一覧表示される。



画面2
音楽CDのデータなどを登録しよう。日本語の入力も可能だ。



画面3
曲目を設定するダイアログ。ひとつずつ曲目や長さを設定しよう。



画面4
コンソールベースで操作するm3x-textも用意されている。

各種パスワードの暗号化と取り出し

Figaro's Password Manager

バージョン: 0.5.0alpha ライセンス: GPL

<http://www.figaro.org/fpm/>

ビルドとインストール

FPMは、tarボールとRPMのソース/バイナリパッケージが配布されている。Red Hat系ディストリビューションではRPMパッケージを使うとよいだろう。tarボールからのビルドとインストールも、「./configure」「make」

「make install」という一般的な手順だ。

パスワードの保存

「fpm&」として起動すると、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で管理するユーザー名やパスワードを暗号化する際のキーとしても使われる。

パスワードを入力すると、FPMのウィンドウが開く(画面1)。ウィンドウには、設定済みのエントリのタイトルやURL/コマンド、ユーザー名が一覧表示されている。

エントリを新規作成するには、ツ

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は、[Launcher]を[Web]に設定し、[URL/Arg]にURLを設定しておこう。

入力したパスワードは、通常「*」でマスクされて見えないが、[Show Password]ボタンを押すと内容を確認できる。また、[Generate]ボタンで開くダイアログで、パスワードを自動生成することも可能だ。

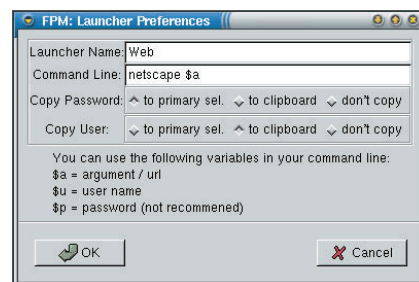
パスワードの取り出し

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レクションに内容がコピーされ、Ctrl-Vキーやマウスの中ボタンでペーストされる。

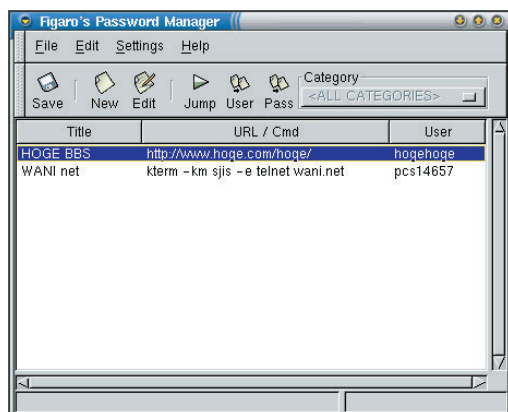
また、[Jump]ボタンを押すと、指定したURLのページをNetscapeで表示する。この場合、ユーザー名はクリッ

ボード、パスワードはプライマリセレクションにコピーされ、それぞれAlt-Vキーとマウスの中ボタンでペーストできる。Netscapeのほか、sshや任意のコマンドを起動することも可能で、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3)。

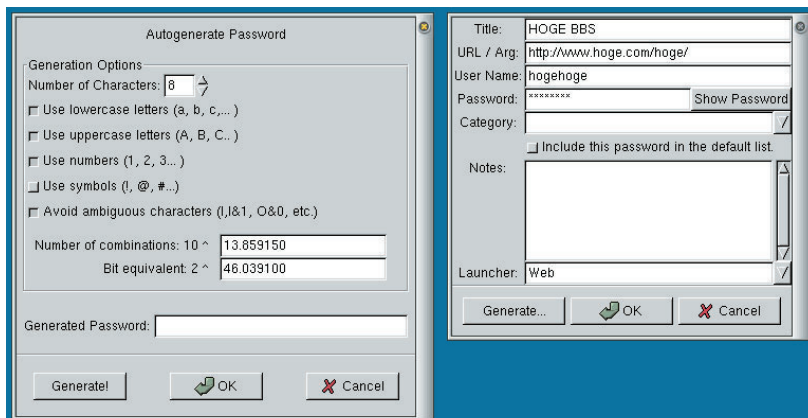
Webブラウザを用いた、ショッピングやオークションなど、ユーザー名とパスワードを要求される機会は増える一方だ。FPMを用いれば、モニタの横にパスワードを書きとめたりせずに、スマートな管理が可能になる。



画面3 起動するコマンドやユーザー名、パスワードの取り出し方を設定。



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。自動生成も可能だ。

Linux上でのビジュアル開発を実現する

JBuilder 3.5
Foundation 日本語版

最終回 Javaプログラムの設計・デバッグ

文：加藤大受

Text: Daiju Kato

daiju@ms.tokyo.jcom.ne.jp

今回はアプレットを作成し、Webサーバに配布してブラウザ上で実行しました。前回作成したような簡単なJavaプログラミングの場合は、それほど不具合に悩まされることはありません。しかし、ある程度複雑なプログラムを作成していくときにはプログラムの設計が必要ですし、不具合を修正していくためにデバッグを行います。今回は、少し論理的なお話となりますが、プログラムの設計とJBuilder 3.5 Foundationが提供しているデバッグ機能について説明します。

プログラムの設計

Javaだけでなく、実際にプログラムを書く場合は事前準備が必要です。事前準備に何を行うかは、作成するプログラムの規模によって大きく異なりますが、一人で作成するのか、人と分業するのかで大きく分けることができます。分業で行う場合、どのような製品を作成していくかのコンセプトを明確にして、仕様書を作成します。本当は一人でやる場合でもきちんとした仕様書を作成するのが理想です。

昔のプログラム設計のスタイルは、まず仕様書を完璧に作成し、それに従いプログラム開発をしていくというものでした。これをウォーターフォール手法といいます。最近では、JBuilderのようなRADツールと呼ばれる、ビジュアルで画面設計をしたり、ウィザードを

利用してコードを生成する製品が一般的となってきています。このため、従来のスタイルとは異なり、仕様書とプロトタイプを同時に作成しながら、詳細をつめていく方法が一般的になっています。これをプロトタイピング手法といいます。画面イメージ、印刷イメージがプロトタイプによって明確になりやすいので、使いやすいプログラムになる傾向が強いのが特徴です。

どちらのスタイルであっても仕様書を作ることに変わりません。ちょっとしたプログラムであっても開発時のメモなどをきちんと作成する習慣を持つ必要があります。現在は初心者であっても、慣れていくに従いだんだん大規模なプログラムを開発できるようになりますので、それに合わせるようにプログラムの設計に関する知識も身につけていただきたいと思います。プログラム開発手法、プログラム設計手法

に関する書籍は数多く出版されていますので、興味がある方はこれらの書籍をまずは読まれることをお勧めします。

デバッグとは

プログラムを作成するときはコードを書くことも大切ですが、書いたコードの不具合を修正することも非常に重要です。不具合のことを「バグ」といい、このバグを取り除く作業を「デバッグ」といいます。また、デバッグ時に使用するツールを「デバッガ」といいます。JBuilderのようなRADツールには統合開発環境(Integrated Development Environment: IDE)が提供されており、開発ツール内からデバッガを呼び出すことができます。画面1はJBuilder内のデバッガです。一般的にデバッガが提供している機能には次のようなものがあります。

- ・変数の値を表示
- ・ソースのブレークポイントの設定
(行指定、条件指定)
- ・エラー内容の表示
(論理エラーおよび実行時エラー)
- ・ソースコードレベルでのトレース実行、ステップ実行

これらの機能を利用してプログラムのデバッグを行い、不具合のあるソースコードを修正していきながら、プログラムの完成度を高めていきます。場合によっては、プログラムを作成するよりも時間をかけることもあります。

バグの種類

一般的にバグといってもさまざまな種類のものがあります。ソフトウェアにはバグのないものがないといわれるほど、バグは多かれ少なかれ存在します。市販のソフトウェアの場合は、影響の大きい問題はほとんど取り除かれています。時々新たなバグがユーザーの手によって発見されることも多いのが実状です。

筆者は以前、QAエンジニア(ソフトのテスト・品質管理を行うエンジニア)をしていましたので、このときに使用していたバグのカテゴリーを使いながら、バグの種類を考えてみることにしましょう。

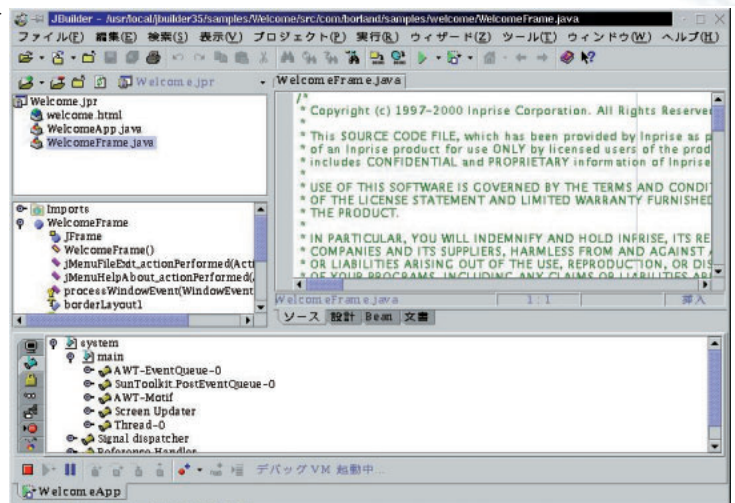
・クラッシュバグ

マシンがフリーズしてしまう、ソフトウェアが止まってしまふ、落ちてしまふなどの非常に致命的な現象を引き起こすなど

・メジャーバグ

コマンド、メニューから呼び出される機能などが動かない、間違った処理を行うなど

画面1 JBuilder
のデバッグ



・マイナーバグ

コマンドのある特定の引数、オプションなどが動かない、間違った処理を行うなど

・スペックバグ

動きは問題ないが、機能のスペック(仕様)自体が間違っている

・マニュアルバグ

マニュアル、ヘルプなどの記述が間違っている

このカテゴリとバグの頻出率をベースとした優先順位を設定し、修正にすべて順位をつけていきます。この優先順位(プライオリティといいます)を設定することが市販のソフトウェアの場合には大切になります。また、基幹システムではクラッシュバグについては、どんなに頻出する可能性が低くても与える影響が膨大になる傾向が多いので高いプライオリティに位置づけられます。このようなバグの分類を行うことによってバグの修正を行っています。また、バグの修正内容や仕様変更など、正確な記録を残すようにしています。

では、なぜバグが発生するのでしょうか。バグの発生原因を少し考えてみましょう。筆者が考えるに、次の3つがパターンがあると思います。

・開発者側の問題

単純なケアレスミス、論理的な手続きの間違い、変数の初期化・設定のし忘れなどが考えられる

・ソフトウェア側の問題

開発ツールのバグやOSのバグによって引き起こされるもので、開発者にとっては一番嫌なもの

・環境の問題

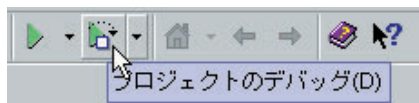
他のアプリケーション、ハードウェアなど使用する環境によってもたらされる問題

ソフトベンダーからしてみると、すべての組み合わせはテストできませんので、最後の環境の問題は製品出荷後にユーザーから報告されて初めてわかるものが多いのが実状です。できるだけバグの少ないコードを書くこと、短期間で効率よくバグを修正していくことがプログラムを作成していくうえでもっとも難しいものかもしれません。しかし、知識と経験で自分なりの考えが備わってくることで完成度の高いプログラムを作成することができるようになります。プログラマーになりたいと思っている読者の方は、早い時期からいろいろな知識を身につけておくことで優れたプログラマーになれる可能性があります。

JBuilder 3.5 Foundation のデバッグ機能

少し汎用的な話をしてきましたが、話をJBuilder 3.5 Foundationに戻してこの製品のデバッグ機能をみてみましょう。JBuilder 3.5 Foundationのデバッガは、メニューの[実行(R)] - [プロジェクトのデバッグ(D)]か、Shift+F9、またはツールバーのデバッグ実行アイコンボタン(画面2)から起動することができます。

デバッガを起動すると画面1のように、左側にアイコンのついたタブ、下にデバッガツールバーにアイコンボタンが表示されたウィンドウが表示されます。このウィンドウを「デバッグビュー」といい、左側のタブによって7つのビューを表示することができます。



画面2 ツールバーのデバッグ実行アイコンボタン

各デバッガビューは表1のような機能を提供しています。

デバッガツールバーには、プログラムの停止、再起動/再開、一時停止を表すアイコンやスマートステップ、ステップオーバー、ステップイン、ブレークポイントの追加、監視式の追加、現在のスタックフレームを表示するための各アイコンが用意されています(表2)。

これらの機能を利用して作成しているJavaプログラムのデバッグ処理を行います。それでは簡単なプログラムを作成して実際にデバッグ処理を行ってみましょう。

デバッグ処理の実行

まず、簡単なJavaプログラムを作成してデバッグを行ってみましょう。これから作成するプログラムは生年月日を入力し、年齢を表示するプログラムです。プログラムとしては非常に簡単

ですが、年齢の計算ルーチンなどはビジネスアプリケーションではよく使われるものですので、ロジックも参考にしたいと思います。このアプリケーションを利用して、ブレークポイントの設定方法、変数の値の参照方法を説明します。

JBuilder 3.5 Foundationを起動し、[ファイル(F)] - [新規作成プロジェクト(P)]を選択し、プロジェクトウィザードを起動します。プロジェクトファイル名を「/home/jbprojects/myDemoApp/myDemoApp.jpr」として[終了]ボタンをクリックします。続いて、[ファイル(F)] - [新規作成(N)]を選択して[オブジェクトギャラリー]ダイアログボックスを表示し、「アプリケーション」を選択すると、Javaアプリケーションの雛形を作成するアプリケーションウィザードが起動します。パッケージ名を「mydemoapp」、クラス名を「myApp」とし[次へ]ボタンをクリックします。次のFrame

タブ	ビュー	説明
	コンソール出力、入力、エラービュー	プログラムの出力とプログラムのエラーを表示する。プログラムに必要な入力も指定できる。アイコン上のイメージは、プログラムの出力があるかエラーメッセージが表示されると変化する
	スレッド、コールスタック、データビュー	プログラム内のスレッドグループを表示する。各スレッドグループは、展開するとそのスレッドが表示され、現在の呼び出しシーケンスを表すスタックフレームトレースが入っている。各スタックフレームを展開すると、利用可能なデータが表示される(静的データはこのビューではなく、読み込み済みクラスと静的データビューに表示される)
	同期モニタビュー	スレッドが使う同期モニタとその現在の状態を表示する
	データ監視式ビュー	追跡したいデータメンバや変数の現在の値を表示する
	読み込み済みクラスと静的データビュー	プログラムによって現在読み込まれているクラスを表示する。クラスを展開すると、そのクラスの静的データがあればそれが表示される。ツリー内にパッケージが表示されている場合、読み込まれているクラスの数が表示される
	データ/コードブレークポイントビュー	ファイル内に設定されたすべてのブレークポイントとその現在の状態を表示する。この情報は、[表示(V)]-[ブレークポイント(B)]コマンドからデバッグを開始する前に利用できる
	トレースが不可のクラスビュー	JBuilder 3.5 Foundationでは使用できない

表1 デバッガビュー

アイコン	動作	説明
	プログラムのリセット	現在のアプリケーションの実行を終了して、メモリから解放する。これは[実行(R)]-[プログラムのリセット(E)]と同じ
	プログラムの再起動/再開	現在のデバッグセッションの続行、完了またはリセットされたデバッグセッションを再起動する。これは[実行(R)]-[プログラムの再開(S)]と同じ
	プログラム一時停止	現在のデバッグセッションを一時停止する。これは[実行(R)]-[プログラムの一時停止(U)]と同じ
	スマートステップ	JBuilder 3.5 Foundationでは使用できない機能
	ステップオーバー	現在のコード行をステップオーバーする。これは[実行(R)]-[ステップオーバー]と同じ
	ステップイン	現在のコード行をステップインする。これは[実行(R)]-[ステップイン]と同じ
	ステップアウト	現在のメソッドからステップアウトし、呼び出し元に戻る。これは[実行(R)]-[ステップアウト]と同じ
	ブレークポイントの追加	ブレークポイントを現在のデバッグセッションに追加する。ブレークポイントの種類を選択するには、アイコンの右側の下矢印をクリックする。これは[実行(R)]-[ブレークポイントの追加(B)]と同じ
	監視式の追加	現在のデバッグセッションに監視式を追加する。これは[実行(R)]-[監視式の追加]と同じ
	現在のスタックフレームを表示	現在のスレッドのコールスタックを表示する

表2 デバッガツールバー

クラスの設定画面で、クラスを「myAppFrame」、タイトルを「年齢を調べるプログラム」とし、[終了]ボタンをクリックします。これで作成するJavaアプリケーションの雛形ができました。

雛形ができたら、[設計]のタブを選択し、デザイナ画面をアクティブにします。次に、左側の構造ペインから「contentPane」を選択し、Layoutプロパティを「null」に変更します。続いて、コンポーネントパレットのSwingのページから、JLabelコンポーネントを2つ、JTextFieldコンポーネントを1つ、JButtonコンポーネントを1つフォームに配置します。プログラムを読みやすくするため、jLabel2コンポーネントのnameプロパティの値を「resultLabel」に、jTextField1コンポーネントのnameプロパティの値を「ageTextField」に変更します。また、jLabel1コンポーネントとjButton1コンポーネントのtextプロパティの値を画面3のように変更します。

ユーザーインターフェイスが設計できたら、jButton1のactionPerformedのイベントを設定します。jButton1コンポーネントを選択し、インスペクタでイベントページを選択し、actionPerformedイベントをダブルクリックします。ここで、ソースコードエディタに戻り、年齢を計算しresultLabelオブジェクトに設定するコードを記述します。コードはリスト1のようになります。

コメントを細かく書いておきましたので、だいたいの流れはつかめると思います。まず、現在の日付を取得し、年、月、日に分解します。同様に、入力された生年月日も分解します。ただし、生年月日が正しく入力されていない場合を想定する必要がありますので、try~catch構文を使用して例外処理を

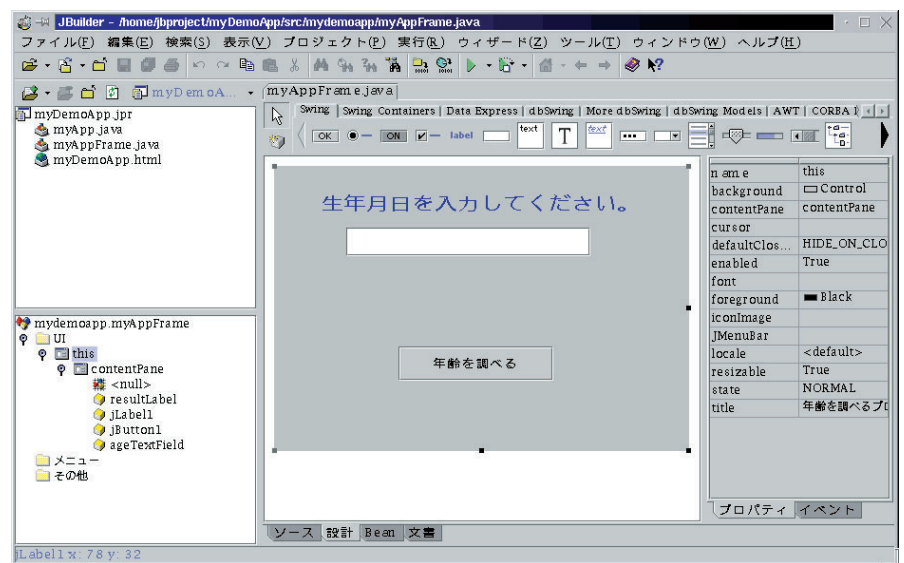
行っています。

次に、今年から生まれ年を引き、満年齢を計算します。続いて誕生日前かどうかの処理を行い、誕生日前なら満年齢から1を引きます。もし、今日が誕生日ならば結果を表示する時に「Happy Birthday!」を追加するようにしています。

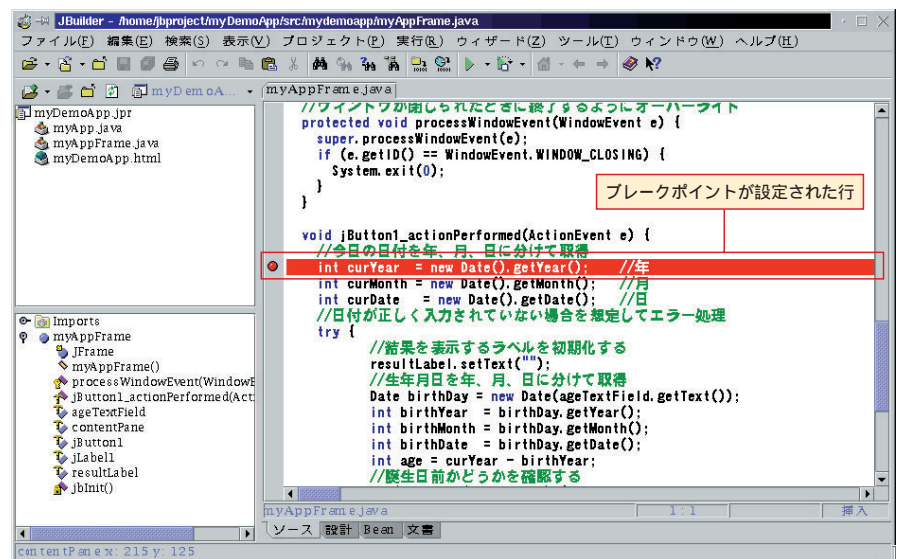
ボタンのイベントを実装したら、ブレークポイントを設定してみましょう。イベントの「int curYear = new Date().getYear();」行にカーソルを置

き、マウスの右クリックで表示されるメニューやエディタ横の青丸をクリックするとブレークポイントが設定されます。ブレークポイントが設定されると、画面4のように設定された行が赤く表示されます。

それではデバッガを起動してデバッグを開始してみましょう。[実行(R)] - [プロジェクトのデバッグ(D)]を選択してデバッガを起動します。デバッガが起動すると、作成されたフォームが起動します。もし、フォームの



画面3 作成したユーザーインターフェイス



画面4 ブレークポイントを設定したところ

起動処理までのコードにブレークポイントが設定されている場合、そのコードの位置で実行が停止します。今回の場合は、ボタンをクリックしたときに発生するactionPerformedのルーチンにブレークポイントが設定されていま

すので、クリックすると指定されたブレークポイントの位置で実行が停止します（画面5）。ここで、[実行(R)] - [ステップイン(I)]または[F7]キーを押して1行ずつ実行していくことで、年齢の計算のルーチンを1行

ずつ実行していくことができます。もし、年齢を格納する変数である「age」の値を参照したいときは、監視式のビュー（眼鏡のアイコン）をアクティブにし、右クリックで表示するメニューで監視式を追加することでageの値を

リスト1 myAppFrame.java

```
package mydemoapp;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;

public class myAppFrame extends JFrame {
    JPanel contentPane;
    JButton jButton1 = new JButton();
    JTextField ageTextField = new JTextField();
    JLabel jLabel1 = new JLabel();
    JLabel resultLabel = new JLabel();

    //フレームの構築
    public myAppFrame() {
        enableEvents(AWTEvent.WINDOW_EVENT_MASK);
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }

    //コンポーネントの初期化
    private void jbInit() throws Exception {
        jButton1.setText("年齢を調べる");
        jButton1.setBounds(new
            Rectangle(111, 193, 163, 36));
        jButton1.addActionListener(new
            java.awt.event.ActionListener() {

                public void actionPerformed(ActionEvent e) {
                    jButton1_actionPerformed(e);
                }
            });
        contentPane = (JPanel) this.getContentPane();
        contentPane.setLayout(null);
        this.setSize(new Dimension(400, 300));
        this.setTitle("年齢を調べるプログラム");
        ageTextField.setBounds(new
            Rectangle(58, 64, 257, 30));
        jLabel1.setFont(new
            java.awt.Font("Dialog", 1, 20));
        jLabel1.setForeground(Color.blue);
        jLabel1.setText("生年月日を入力してください。");
        jLabel1.setBounds(new
            Rectangle(54, 23, 296, 30));
        resultLabel.setFont(new
            java.awt.Font("Dialog", 1, 20));
        resultLabel.setForeground(Color.red);
        resultLabel.setBounds(new
```

```
Rectangle(24, 118, 352, 39));
        contentPane.add(jLabel1, null);
        contentPane.add(ageTextField, null);
        contentPane.add(jButton1, null);
        contentPane.add(resultLabel, null);
    }

    //ウィンドウが閉じられたときに終了するようにオーバーライド
    protected void processWindowEvent(WindowEvent e) {
        super.processWindowEvent(e);
        if (e.getID() == WindowEvent.WINDOW_CLOSING) {
            System.exit(0);
        }
    }

    void jButton1_actionPerformed(ActionEvent e) {
        //今日の日付を年、月、日に分けて取得
        int curYear = new Date().getYear(); //年
        int curMonth = new Date().getMonth(); //月
        int curDate = new Date().getDate(); //日
        //日付が正しく入力されていない場合を想定してエラー処理
        try {
            //結果を表示するラベルを初期化する
            resultLabel.setText("");
            //生年月日を年、月、日に分けて取得
            Date birthDay = new
                Date(ageTextField.getText());
            int birthYear = birthDay.getYear();
            int birthMonth = birthDay.getMonth();
            int birthDate = birthDay.getDate();
            int age = curYear - birthYear;
            //誕生日前かどうかを確認する
            if (curMonth < birthMonth) {
                age--;
            } else if (curMonth == birthMonth) {
                if (curDate < birthDate) {
                    age--;
                } else if (curDate == birthDate) {
                    resultLabel.setText(
                        "Happy Birthday!");
                }
            }
            resultLabel.setText(resultLabel.getText()
                + " 年齢は "+age+" 歳ですね。");
        } catch (Exception ex) {
            //生年月日の入力がおかしいとき
            resultLabel.setText(
                "生年月日が正しく入力されているか確認してください");
        }
    }
}
```

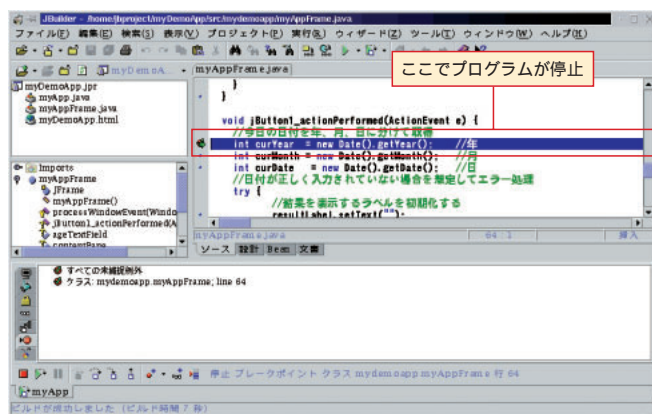
参照することができます(画面6~8)。

ステップ実行を行い、デバッグしたい領域を通り越したら[F9]キーを押して通常実行します。デバッグのときは、ブレークポイントの設定 デバッガでの実行 ブレークポイントで停止 変数値参照 ステップオーバー([F7]キー)またはステップイン([F8]キー) 通常実行([F9]キー)の順序でプログラムの不具合を修正するようにしてください。デバッガを使いこなすこと、デバッグ処理を重要視すること、例外が発生する可能性を考慮することが安定したプログラムを作成するうえでのコツになるでしょう。

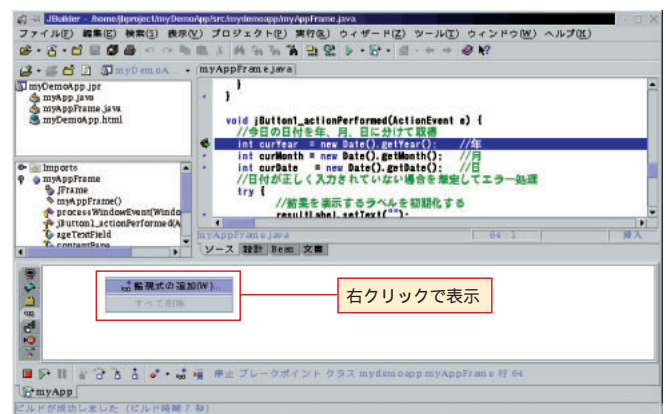
最後に

3回にわたり、インストール、アプレット作成、デバッグまで、非常に簡単ではありますが、Javaプログラムを書いていくうえで必要な情報を説明してきました。Linuxの世界にビジュアルなプログラミングの世界を提供したことは、Linuxをさらに発展させていくこと、ユーザー数を拡大させていくことに大きく貢献することでしょう。この連載にて少しでも読者のみなさんのお役に立てれば幸いです。ぜひ一度、Javaでプログラミングしてください。

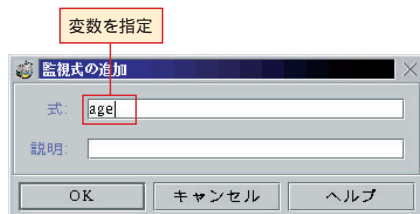
本連載で使用したJBuilder 3.5 Foundation日本語版と、JDK (Java 2 SDK Standard Edition version 1.2.2)は、6月号のCD-ROMに収録されています。詳しいインストール方法も記載されていますのであわせてご覧ください。なお、JBuilder 3.5 Foundation日本語版はインプライズのWebサイト (<http://www.inprise.co.jp/jbuilder/foundation/download/>)から、JDKはSun MicrosystemsのWebサイト (<http://java.sun.com/products/jdk/1.2/ja/download-linux.html>)からそれぞれダウンロードすることもできます。ただし、サイズが非常に大きいので注意が必要です。



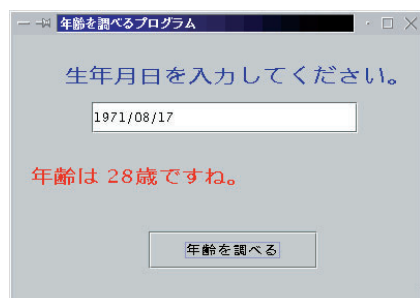
画面5 ブレークポイントで止まったところ



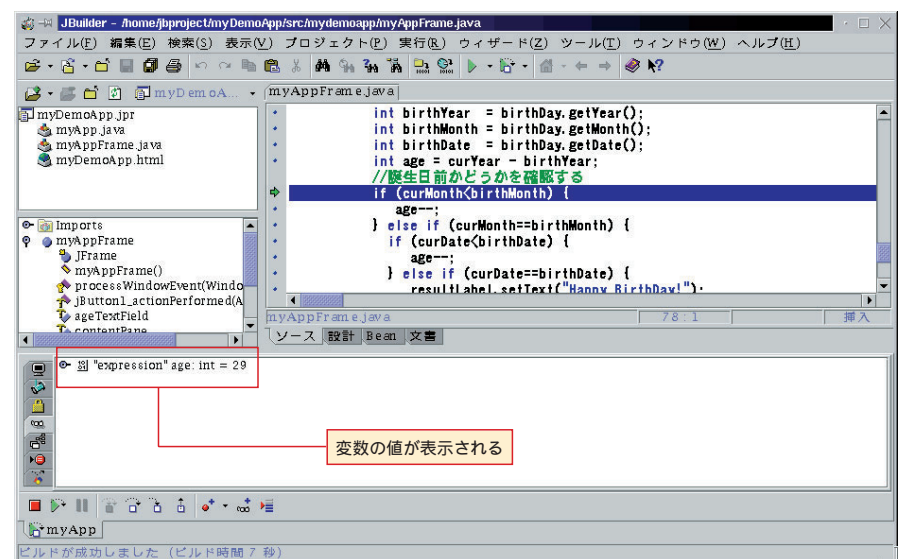
画面6 監視式の追加



画面7 監視式の追加ウィンドウ



画面9 実行したところ



画面8 ageの値の参照

カフカの手紙とインターネット

文：豊福 剛
Text : Tsuyoshi Toyofuku

カフカは、距離を克服するための技術（交通手段）と亡霊的な技術（通信メディア）という2つの技術系列を区別する、とドゥルーズ=ガタリは指摘している。後者の技術系列には、電信、電話、そしてあきらかにインターネットも含まれるのだが、それが必ずしも情報技術に限定されないのは、郵便もこの系列に属するからだ。すると、手紙というメディアをカフカがどのように使用したか、について考えることで、通信メディアにおける亡霊的なものに対する重要な観点が得られるだろう。

言表行為の主体と言表の主体

ドゥルーズ=ガタリは、カフカをエクリチュールの機械/文学機械として分析した。カフカの手紙は、公表を前提として書かれたものではない。そもそもカフカは「自分の書いたものすべてを、それらがあたかも手紙であるかのように破棄すること」を考えていた。それでも、手紙が重要なのは、それが「ひとつの根茎・網・クモの巣」であり、「文学機械の不可欠の歯車、動力の部分だから」であるからだ。さらに、手紙を書くことの欲求は、手紙そのものだけでなく、その送付や配達のマカニズム全体の運動性とつながっているのである。

このことを理解するためには、カフカ独特の手紙の使用を考察しなければならない。

カフカの手紙には「言表行為の主体と言表の主体という2つの主体の二重性が保持されている」とドゥルーズ=ガタリは指摘する。

言表行為の主体とは、手紙を書く本人のことであり、一方、言表の主体とは、手紙の内容が語っている主体のことである。この2つの主体は、手紙を書く主体と手紙に書かれた主体、と考えるとわかりやすいかもしれない。

2つの主体は、通常は、同一の主体であることが前提されている。手紙を読むときには、手紙に書かれた内容から、手紙を書いた本人の意思を読み取るのであり、普通は、この2つをあえて区別したりはしない。

それでは、なぜこの2つの主体が区別されなければならないのだろうか。それは、カフカが手紙を「倒錯的または悪魔的」に使用したからである。どういうことか。

ドゥルーズ=ガタリは「カフカ的な愛の例」について、次のように書いている。

「或る男が一度しか会ったことのない女に夢中になって、非常に沢山の手紙を書く。彼は《来る》ことができな

い。彼は郵便袋のなかの手紙から離れることはない。そして決裂、つまり最後の手紙の翌日、田舎にある自分の家へ夜になって帰ってきた彼は郵便配達人を殺す」

男は女に会わないし、女のところへ来ることもしない。女のところに来ることを告げるために手紙を用いるのではない。そうではなく、「手紙の送付、手紙の行程、郵便配達人のコースと身ぶりが、来ることの代わりになる」のだ。

そして手紙を書く欲求は、「相手から手紙を奪いたいという錯乱した欲求」によって駆り立てられている。《上を見ても下を見てもその階段はすべてあの読まれた手紙によってうず高く覆われていました。……それは本当の欲求の夢でした。》とカフカは書いている。カフカにおけるこのような手紙の使用は、どこかフェティシズムに通じるものが感じられる。手紙に書かれた主体は、ある虚構を生きているからだ。

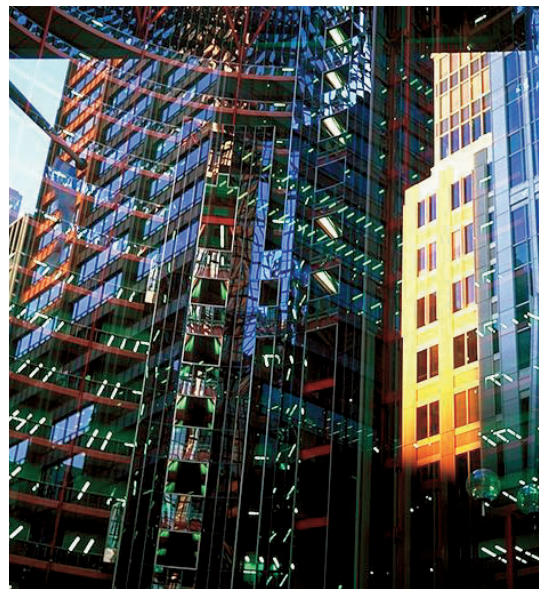
カフカにとって手紙は分身であり、手紙というメディアにおいて現出する主体は、亡霊的である。映像メディアや音声メディアにおいて現出する主体が、分身であり、亡霊的であるのは、ある意味でわかりやすい。重要なのは、映像や音声ではなく、手紙という書かれた言葉によって分身化が成り立つことである。それでは、カフカ的な手紙の使用を、メディアを使用する主体とメディアにおいて表象される主体の二重性として、一般化することはできるだろうか。

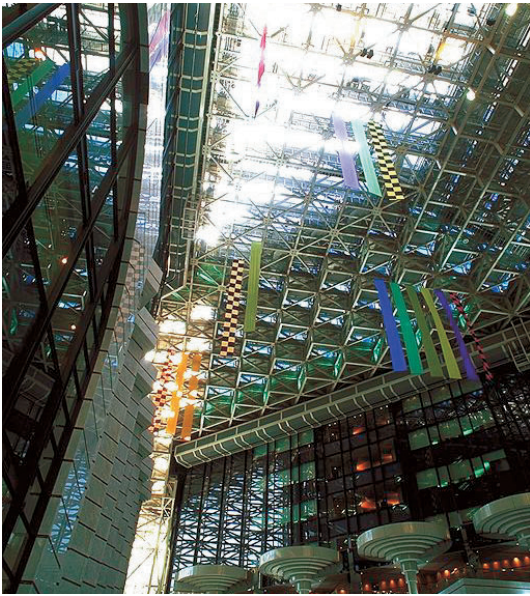
掲示板 / BBS とネットニュース

言表行為の主体と言表の主体の二重性は、今日では、それほど特別なものではない。たとえば、掲示板やBBSにおけるハンドルネームに、この二重性を確認することができる。

ハンドルネームは、ネットにおけるニックネームではない。ニックネームは、その本人のことを知っている他人によって名付られる名前であるのに対して、ハンドルネームは、その本人のことを知らない他人に向けて、本人によって名付けられる名前である。したがって、ハンドルネームには、多かれ少なかれ、本人の自己演出の意識が反映しているし、ある種の虚構を生きているといえるだろう。

ここ数年で、インターネットにおける掲示板やBBSの数は急増したように思う。興味深いのは、掲示板やBBSは、もともとパソコン通信の文化を代表するものであり、ハンドルネームも、そのような文化から発生した現象だったのに対して、かつてのインターネット文化は、メーリングリ





ストにしるネットニュースにしる、メールアドレスや署名によって、誰が発言しているのか明確にしようという了解があったことだ。

『インターネットヒストリー』（Neil Randall 著、村井純監訳、オライリージャパン）で、USENETの誕生に大きく関わったスティーブ・ペロバンは、USENETを開発するうえで、検討はしたものの、けっきょく採用を取りやめたアイデアとして、電子署名の使用、メッセージのコントロールとキャンセルの機能、匿名によるメッセージ送付、この3つをあげている。

ここで、電子署名以外の2つのアイデアについては、それを採用しなかった理由として、これらの機能が悪用されることへの懸念があったからだ、と語っている。とくに、匿名メッセージについては、匿名が必要とされる状況があることを十分認識したうえで、それでも「送付するメッセージには責任を持つべきだ」という考えがあったことを述べている。

このようなUSENET～ネットニュースの系譜に対して、最近のインターネットにおける掲示板やBBSの急増は、パソコン通信文化への揺り戻しであるように感じられる。パソコン通信の掲示板やBBSは、そのプログラムがデプロイされているホストがあり、そこにアクセスするユーザーは、ホストに対する端末になっていた。インターネットの掲示板やBBSは、そのほとんどがCGIで実装されているため、中央のホストとそれにアクセスする端末という図式が再現されてしまった。

もちろん、ネットニュースと掲示板/BBSを比較して、両者の優劣を単純に結論づけることはできない。匿名でないネットニュースでも問題発言はある。問題発言があったとき、掲示板/BBSでは、その発言を本人や管理者が削除することができるが、ネットニュースでは、それは困難である。このように、ネットニュースと掲示板/BBSは一見すると似たような機能ではあるが、その背景にある考え方の違いが、これらの機能の実現方式の違いに反映している点は、確認しておきたい。

チャットにおける亡霊的なものの貧困

話をハンドルネームに戻すと、掲示板/BBSのほかに、チャットやMUDといったものもある。MUDについては日本語で利用できるものを見たことがないのだが、欧米ではかなり人気があるようで、MUD中毒が社会問題になっ

たりもするようだ。シェリー・タークルの『接続された心』でも、このMUDに夢中になりすぎたユーザーの取材や心理分析に、かなりのページがさかれている。たとえば、会話している相手が実はイライザのような人工知能プログラムであったにも関わらず、その相手がほんとうの人間だと信じていた、といった話だ。

このような話は、原理的にはありえると思うのだが、日本語でそのような「リアルさ」を演技できるプログラムを知らない。このような事例を根拠にして、インターフェイス的主体、つまりスクリーンの背後にいる主体が人間なのかプログラムなのか、そのことをもはや問わない、という態度変更を強調するのは、少なくとも日本語文化圏においては無理があるように思う。

一方、チャットについては、CGIで実装したものの多くは、プログラムのには掲示板/BBSとそんなに大きな違いがあるわけではない。あえて違いを探すとしたら、特定のユーザーとだけ会話できる「個室モード」機能が実装されていることぐらいだろう。最近Web上に見られるチャットの多くには、いわゆるデートクラブ的なものが目立つのだが、たまにそういうサイトを観察してみた限りだと、どこが面白いのか、さっぱりわからない。「さんが退室しました」といったたぐいのメッセージや、挨拶に毛がはえたみたいなお話ばかりで、なんとも退屈で失語症的な会話に思える。

これは、ハンドルネームがあるとはいえ、チャットにおける発言が、日常とさほど変わらない意識でなされているからではないだろうか。それで、チャットという場には、いつの間にか常連のユーザーができて、日本的な村っぽい雰囲気が醸し出されてくる。

某有名ポータルサイトのチャットなんかでも、たとえば女っぽいハンドルネームでアクセスしていると、たちまち男たちから誘いのメッセージがやってくる。逆に、チャットで女らしさを演じている人の多くは、実は男で、しばらくすると暴力的な発言が出てくるあたりは、おかまを演じきれない未熟さを感じる。だいたい、そのような男が演じる女らしさというのは、男からみた女らしさであるのだが、それにまんまと騙されるウブな男の子もいたりして、苦笑いしてしまうことも多い。

言表の主体に徹することができず、往々にして言表行為の主体がさらけ出されることが多いチャットの現実、亡霊的なものからは遠く、あまりに自然で粗野な光景であるようだ。

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。

英語が嫌いだからオープンソースなのだ

文：安田幸弘
Text: Yukihiro Yasuda

日本のオープンソースソフトのユーザーにとって、最大の問題は、いうまでもなく英語である。

日本生まれの少数のソフトを除けば、インストールのドキュメントもマニュアルも英語、画面に表示されるメッセージ類も英語、日本語は入力さえできないということも少なくない。日本でいち早く、*BSDやLinuxの普及に取り組んだグループにとって、最優先の問題はドキュメントの日本語化だったし、彼らの努力のおかげで、いまでは日本語のドキュメントも整備され、オープンソースソフトが使われるようになったのだと言っても過言ではないだろう。

日本人が英語がヘタな理由

日本人は世界で一番英語がヘタクソだという。そういうぼくも、英語のダメさ加減には、ちよいと自信がある。以前、あるワークショップでフィリピンの人から「彼の会話はわかりにくいけど、メールの英語は立派な英語だ」と言われたりもした。これはメールの英文を誉められたのか、それとも英会話の実力に引導を渡されたのか、よくわからない。まあ、言われるまでもなくヘタクソは自覚している。ほっといてくれ。

そうかと思えば別のワークショップで話したときは、われながらひどい英語だったので「ヘタクソな英語に付き合ってくれてありがとう」と言ったら、マレーシアの人に「英語がヘタクソでも気にする必要はないよ」となぐさめられてしまった。よっぽどヘタクソだったに違いない。

どこかで読んだか聞いたかしたことなのだが、日本人の英語がヘタクソな理由は、日本では必要なものはプロが何でも翻訳しちゃうので、どうしても英語が必要だという切実な要求がないからだという。英語での講義も英語のレポートもない大学が成立できるのは、日本ぐらいのものだそう。

要するに英語を使わなくても生きていけるのだから、英語ができるようになるわけもないという

ことだ。もっともな理由である。

インターネットのおかげで、英語が必要な場面が増えてきた。さすがのぼくも、数年前は「マジメに英語を勉強しないとイケないかなあ」と思っていたのだが、ところが日本人はそんなに軟弱じゃなかった。

最近ではインターネットを探せば、英文サイトの翻訳ページも次々とヒットする。インターネットで爆発的に英語の情報が増えた分、日本語の翻訳もやはり爆発的に増えているのである。ソフトにしても、必要なツールは誰かが日本語で使えるようなパッチを作って配布してくれているばかりか、メニューからヘルプまで、日本語化されているものさえある。おかげで最近では、オープンソースソフトと付き合うときも、ほとんど不便は感じない。どうやら、ぼくの英語を改善するチャンスは永遠に失われてしまったようだ。

英語力とデジタルデバインド

先日、インドでIndiaLinkという非営利ネットワークを運営している知人から、Spruceというメールソフトについて意見を聞かせてくれというメールが届いた。彼は現在、インドの各地に点在するIndiaLinkのユーザーのために、Linuxをベースとするクライアント環境を作っているのだが、適当なメールソフトがないのが彼の悩みだった。

Spruceは、まだアルファ版レベルながら結構よさそう。国際化されているGTKアプリなので、将来は日本語化にも期待できそう。ただ、今の段階ではMIMEの実装が「とりあえず」のレベルなのは残念、というような返事を書いた。そしてそのついでに、前々から気になっていた素朴な質問　インドのユーザーは、ヒンズー語のメールは使わないのか　をしてみた。

彼からの答えをまとめると、ざっとこんな具合だ。「多言語国家のインドで唯一の共通語は英語なので、NGOのスタッフやビジネスマンは、ふだ

ん英語を使う。だからいまのところ英語さえ使えれば問題はない。しかし、農村部や貧困地域には英語を話さない人は非常に多い。私自身は使わなくても、ローカルな言語のサポートがこれからの問題だと思う。そうでなければ、インターネットやコンピュータは英語を話せる人だけのものになり、英語を話す富裕層と英語を話さない貧困層の経済的、文化的、社会的な格差はますます広がるだろう……」

英語一辺倒で突き進むグローバリゼーション、そしてIT革命の、ちょっと困った一面がここにある。

幸運な日本人

あたりまえだけど、決して日本人だけが英語を話せないわけじゃない。

人口だけで言えば、世界の大多数の人は、日々の暮らしの中で英語を使う必要も感じず、英語を話さないのだ。だけど巨額の利益をあげているIT産業は、市場価値がないという理由で、彼らの言葉に見向きもしない。われわれ日本人は、お金を持っていておかげでIT革命の恩恵をこうむっているけれど、世界中をひとつに結ぶはずのインターネットの世界で、英語を話さない人々は、存在することさえ許されないという現実があるわけだ。英語がヘタクソなぼくには、とても他人事とは思えない。

マイクロソフトやIBM、アップル、Sunといったソフトウェアメーカーは、UNICODEでこの問題を解決できると思っているのかもしれない。しかし、UNICODEにできることは、開発コストを下げるくらいのものでしかない。

日本のLinuxや*BSDの先駆者たちが、せっせとドキュメントを翻訳し、ソフトウェアの日本語化を進めることができたのは、それがオープンソースであり、フリーソフトウェアだったからだろう。ネットを探せば、さまざまなソフトウェアの日本語訳マニュアルが見つかるのは、それらがフリー

だからだ。決して、Linuxがカーネルの内部コードにUNICODEを使っているからなんかじゃない。

グローバリゼーションと英語支配

いくらインターネットでもオープンソースでも、それだけでは英語を話さない貧困層の人々の腹を満たしたり、少数民族の文化を守ることができるわけじゃない。だけど、英語を話さない、地球上の多くの人々、特に少数民族にITへのアクセスの窓口を提供できるのは、オープンソースしかないのではないだろうか。

日本語や中国語、韓国語以外のほとんどの言葉は1バイトで表記できる。GNOMEのように国際化されたオープンソースのプラットフォームなら、ソフトのローカライズには日本語化ほどの手間はいらないだろう。

先進国の政府は、情報化による格差の増大を防ぐための支援を検討しているという。その具体的な内容は知らないけれど、彼らに必要なものは、決してウィンテルパソコンの無償供与なんかじゃないだろう。「世界化」というキーワードが「米国化」の意味じゃないとすれば、むしろ、安くて使いやすいオープンソースのシステムの配布であり、ローカライズの技術、コストの支援ではないだろうか。

グローバリゼーションの時代だから英語だなんて、そんなのアメリカの陰謀に決まってる。グローバリゼーションの時代だからこそ、世界中の人々が自国の言葉で生活を楽しめるようにならなきゃウソだ!!……と主張したい。英語がダメダメなぼくとしては。

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

- 6・7 Microsoftに2分割命令
- 6・5 AMD Thunderbird発表
- 6・5 インテルTimnaの延期
- 5・16 Corel、インプライズとの提携を解消

話題のWindows 2000なんですけど、せっかくセットアップして、アプリケーションもインストールしたんだけど、途中で落ちたり、起動時にブルースクリーンが出るのが頻発したので、結局お蔵入り。やっぱりService Packが出るまで待たなきゃだめかな。

Microsoftに分割命令

さて、長らく続いていたMicrosoft対司法省の裁判だが、ついに分割命令が出た。この裁判は昨年ひとまず、「Microsoft有罪」が決定しており、いままでは、その是正方法についての議論をしていた。司法省は2分割案を提案し、Microsoftは、自ら課する制限を提案していたが、結局、司法省の案が取り上げられ、Microsoftに対し、4カ月以内に分割プランを提出しろということになった。だがこれは、地方裁

判所の判決なので、Microsoftはこれに対して控訴する。そのため、実際には、分割はまだ行われていない。

ずいぶん前に、IBMも独禁法で訴えられていたが、こちらは、有罪かどうかが決まらずに13年も経過して、結局提訴取り下げということになった。理由は、状況が変わってしまったためである。それに比べれば、まだ、Microsoftの裁判は早いほう。だが、控訴するので、また最初からやり直しである。

司法省の分割案は、OSの会社とアプリケーションの会社の2つに分割するというもの。だが、業界では結局どちらの分野でも強力な会社を作ってしまうだけだと効果に疑問を持つ声も。なにせ、WindowsもOfficeも圧倒的に高いシェアを持っているものねえ。

だが、分割されるとしたら、今度はアプリケーションの会社のほうが、重要なプレイヤーとなるのではないかと

筆者は見ている。というのは、Windowsが選択される理由は、Officeを使いたいからであり、iMacでもいいやと思うのはOfficeがあるからである。また、Linuxを使いたいんだけど、ちょっと……と思う理由のひとつは、Officeがないからである。Corelは、WordPerfect Office for Linuxを3月に発売しているが、これは実は昨年ぐら以来まで低価格をウリにWindows用に販売されてきたOfficeスイートを、Linux上のWindowsエミュレータであるWineを使って動かしているもの。一時はCorelも善戦したのだが、Officeの牙城を崩すには至っていない。

Corelの不運

そのCorelはインプライズの買収を断念したようである。この買収は株式交換によって行われる予定だったのだが、米国でもハイテク株を中心に値下がりがあり、Corelの株も下がってしまった。それで、インプライズの買収がダメになってしまったのである。さらに、経費削減のため人員整理を行うのだとか。ちょうどLinux系をビジネスの中心にしようとしていた矢先のことで、どうも株価が下がるタイミングが悪すぎたようである。

もっともこれは、インターネットやLinuxバブルがはじけたのだという見方もあり、インプライズの買収だって、株が高かったからこそその計画であったわけだ。ちょっとバブルがはじけるのが早かったんでしょうかね。

人員整理といえば、米国TurboLinuxも行ったそうである。LinuxCareとかも苦しい状況らしいし、VA LinuxやAndOverNetなんかの株価もドーンと下がってしまった。もっとも、Microsoftの株もドーンと下がっているの、裁

判の影響といえなくもない。アメリカでは株価はかなり重視されるので、株価が下がってしまうと、会社としては厳しい状況に陥ってしまう。株式公開を考えていたところは、この厳しい現実を前に躊躇するところもあるらしい。

そういえば、グラフィックカードメーカーの老舗のひとつNumber Nineが廃業しちゃったのだとか。S3はグラフィックチップ部門をVIAに売却しちゃうし、子会社のダイヤモンドマルチメディアはMP3プレーヤに熱心みたい。ノートPCに多く使われていたNeoMagicも、グラフィックチップからは撤退したしねえ。グラフィックカードメーカーといえ、かつては勢いがあったんだけどね。

その背景として言われているのが、グラフィックチップもチップセットやCPUに統合されていくという話。一部のゲーマー以外、3Dの強力なグラフィックカードとかあんまり縁がないものねえ。インテルは、すでにグラフィックデバイス統合した810、815というチップセットを持っていて、低価格のFLEX ATXタイプのカードでは、こちらがかなり伸びているとか。

インテルは、この次にTimnaと呼ばれるチップセットやグラフィックデバイスを統合したCPUを計画しているんだけど、当初の予定では今年末だったものが、来年に延期されてしまった。理由は、820で問題になったMTH (Memory Translator Hub)。これは、本来RAMBUS用に設計されたメモリバスにSD-RAMを接続するデバイス。820のものと同等のMTHを使う計画だったが、MTHにバグが発見され、これを変更するのに時間がかかるということらしい。

だが本当の理由は、RAMBUSの値段が下がらないからではないだろう

か？ たぶん、インテルの最初の計画では、Timnaの登場時には、RAMBUSがかなり安くなっている予定だったのだ。このTimnaは、いまの810 + Celeronよりも安い価格帯のパソコンを想定しているため、全体のコストが問題。ここに高価なRAMBUSメモリは使えないのである。それで、もう少し待ってみようということになったのではないか。

Linuxベースのインターネット専用機

AOLは、Gatewayと組んでインターネットアクセス専用機を作る計画らしい。しかも、OSはLinuxベース、CPUは、TransmetaのCrusoeだという。つまり、Microsoftもインテルも関係ないマシンなのである。これは、Webブラウザやメールなどが可能なもので、ソフトには「Gamera」(ガメラ)って名前がついている。子会社NetscapeのNavigatorは、昔からMozillaという名前が使われていたけど、これは、MosaicとGodzillaからきたもの。Netscapeの親会社だから、たぶんブラウザは、Mozillaベースなんだろうけど、

だからってねえ。Gameraはないだろう(次は日活のGappaか)。Cursoeは、6月末のPC EXPOでIBMなどが製品を展示する予定なのだとか。今年後半には、いろいろと製品が出てくるのかも。

MicrosoftもWindows CEベースで、MSN Companionという製品を計画していて、今年1月のCESには大々的に展示していたが、そのあとどうなったんでしょかね。同じくインテルもCPUはCeleronだが、OSは、Linuxベースというインターネットアクセス専用機を発表していたんだけど、こっちはようやく準備が整ったようだ。

あちこちが、こうした「インターネットアプライアンス」を計画しているのだが、どうでしょうね。たぶん、Linuxを使うというのは、コストを下げたいからでしょう。なにせ、いまパソコンで最も高価な部品は、Windowsのライセンス料金だと言われてますから(ウワサによるとハードディスクの仕入れ値よりも高いのだとか)。とりあえず、Linuxベースのマシンが出るのはいいのだけど、みんなが使い始めると、最後はパソコンというか、ワープロとかやりたくなるんじゃないかなあ。



分割命令に対するビル・ゲイツ会長の反論コメント。「マイクロソフトは上訴において非常に有力な立場にあると思っています」と述べている
<http://www.microsoft.com/japan/presspass/releases/060800doj2.htm>



インテルが6月23日に発表した「Dot.Station Web Appliance」。OSにLinuxを採用し、Web、メール、電話、ホームアプリケーションを内蔵する。直販はせず、ネットサービス企業が特定サービス用端末としてアプリケーションを組み込み、会員向けに販売する。
<http://www.intel.com/pressroom/archive/releases/in062200.htm>

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

～日本発 GIMPのプラグイン「SIC」～

Linuxのキラーアプリケーションのひとつである「GIMP」。そのプラグインソフトが、この日本で開発されている。SIC (Synergistic Image Creator) と名づけられたそのソフトウェアは、人間がモノを見て、認識するに至る過程をシミュレートし、そのデータをもとに画像を加工する。SICは、従来のフィルタ処理とはまった違った発想を持つ画像処理ソフトウェアなのである。

SIC (Synergistic Image Creator) は、東京工芸大学の芸術学部デザイン学科コミュニケーションデザイン研究室の笠尾敦司氏が作成した画像処理ソフトウェアであり、GIMPのプラグインとして動作する。

笠尾氏から連絡を受けた当初、日刊アスキー編集部では、「日本発のGIMPのプラグインソフト」という認識で取材させていただくことになったのだが、お話を聞いてみると、単なるプラグインソフトの範疇にとどまらない、新しい発想のソフトウェアだということがわかってきた。

GIMPのプラグインソフトとして動作するとはいえ、単なるフィルタなどとは異なり、画像データの解析から加

工までを行ってしまうれっきとした画像処理ソフトなのである。

まずは、サンプルの画像を見ていただきたい。これは、絵画ではなく写真データを元に、SICを使って加工したコンピュータグラフィックである。

単に画像ソフトのフィルタを使っただけでは、最初に大まかにフィルタをかけて、そのあとに細かい部分に手を加えてゆくといった作業になり、ここまで詳細で絵画的な画像にするのは、たいへん難しいであろう。しかしSICでは、細かい修正をしなくても、このサンプル画像のような表現が可能なのだ。その理由は、従来の画像ソフトとは一線を隔した画像解析方法を用いているからなのである。



人間がモノを認識する過程をシミュレート

SICの画像解析方法をひとことで表わすならば、「人間がモノを認識する過程そのものをシミュレートして解析している」ということになる。

人間がモノを見る（認識する）場合、目から情報が入り、情報が処理されて最終的に認識に至るわけだが、この情報の処理には、人間が「この対象をこのように見たい」という意識が絡んでくる。たとえば、芸術的な絵画を見ているとき、地図から情報を取得したいときでは、モノの見え方は違うのである。

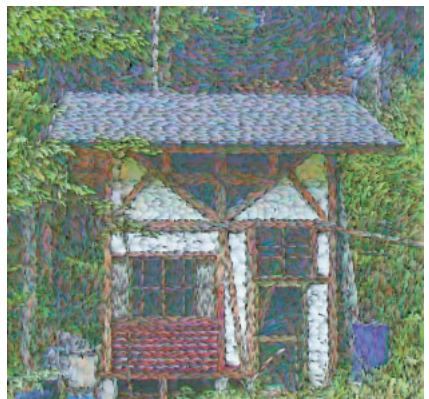
また、人間の目の特性として、明る



画面1 SICで加工を加える前の写真



画面2 画像を人間の目で見える最小領域に分割



画面3 SICによる変換例

さの認識も実際とはかなり違うのだそう
うだ。人間の目は、少しだけ明るいも
のがあったとしても、周囲がまっ暗で
あれば、ものすごく明るく見えてしま
う、といったように、ひどく相対的な
ものなのである。

こうした、人間がモノを見るとき
の特性を踏まえたうえで、SICは画像
を細かい領域に分割する。この領域が、
人間がモノを見て「ひと塊り」と認識
する単位となるのだそうだ。通常のソ
フトウェアが画像を領域分割する場合
は、「いかに効率よく領域を切り出し
ていくか」というところに重点が置か
れていたが、SICの場合は「人間には
どう見えているか」という点に注目し
て領域分割が行われている。

また、カラーマッピングにしても、
単純に同色明度の領域を同じ色にする
のではなく、人間の目の相対的な認識
に合わせた形でグループ化がなされる。
笠尾氏は、「領域のひとつひとつは、
画家が絵を書いていくときのひと筆ひ
と筆に近い」という発想を得たのだそ
うだ。とすれば、この領域 = 筆の1タ
ッチとして処理を施せばよい。画面2
を見て、人間の目というのは、かなり
細かい領域分割をしているのだな、と
思われた方もいらっしゃるかもしれな
い。しかしこれは一番細かく分割した
領域であって、ここから「こうした領
域を、人間がどう意識して筆のタッチ
とするか」という段階に入る。つまり、
その人が領域の境界線（エッジ）にこ
だわってタッチを決定すれば、この領
域ひとつひとつは活かされるであらう
し、モノの境界線をぼんやりと捉える
ようなタッチであれば、領域がグルー
プごとに連結されるだろう。色にして
も同じだ。

こうしてSICは、人間がものを見る
時の最小単位の領域まで画像を分割、

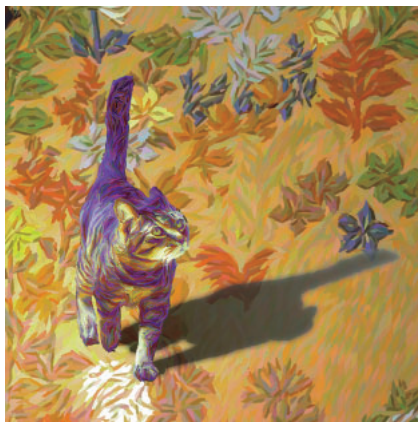
ユーザーの好み通りの処理を施してい
くことが可能となる。機械的、効率的
に領域を分割してしまううえ、単なる
数種類のパラメータしか入力できない
ソフトウェアとは、結果が異なること
がわかるだろう。

SICは、2000年秋に公開予定となっ
ており、開発メンバーを募集中とのこ
とだ。興味のある方は、<http://www.dsn.t-kougei.ac.jp/cp/sic/>にアクセス
していただきたい。

（日刊アスキー編集部・吉川）



画面4 SICによる変換例その2



背景とネコを別々に処理して合成



画面5 Photoshopで水彩画フィルタをかけた場合。
草むらの表現が、画面3、4とは全然違う



細かい筆触で作られた木のある風景。大沢 唯氏作「都会の森」第2回 東京グラフィックスCGアート展入選作

初級Linuxer養成講座

第11回 ファイルの移動と削除

求人などの際に「コンピュータを使えるか否か」を判断するもっとも簡単な基準は、「ファイルのコピーや削除などのファイル操作ができるかどうか」である。しかし、一言でファイル操作といっても、その裏側で何が起きているかを知っていないと、とんでもない落とし穴にはまってしまうことがある。いつもはなにげなく行っているLinuxでの「ファイル操作」について、少し詳しく調べてみよう。

文：竹田善太郎
Text：Zentaro Takeda

きわめて個人的な話なのだが、最近航空管制ゲームにはまっている。飛行機の操縦を楽しむフライトシミュレータ系のゲームは、もう10年以上も前からずっと遊んでいるのだが、単なる交通整理をするだけの航空管制ゲームについては、その存在は知っているあまり興味がなかったのだ。航空趣味が昂じて、自宅で「航空無線」の聴取を始めたのは数年前だが、それでも「管制するのは忙しい仕事だな」という程度の感想しか持たなかった。しかし、ちょっとした暇つぶしのつもりで、ダウンロードしてきた航空管制ゲーム「ATCC」（この時代になんと「DOS」専用のプログラムである）を動かしてみたところ、その奥の深さにはまってしまったのだ（画面1）。

シミュレーションゲームといっても、派手なビジュアルは皆無で、グリーンのもнокロCRTを模した画面上には、単色でレーダーからのデータが、文字や線だけで表示される。そこには、自分の担当する「管制圏」の中にある航空機について、簡単なシンボルと便名、高度、速度、進行方向などが表示されるだけであり、この限られた情報から航空機同士の位置関係や将来の動きを予

測しなければならない。そして、各航空機に対して無線で指示を与えて、航空機同士が衝突することなく、目的地へ効率よく向かえるように誘導するのだ。

このように文章で書いてみると、やっぱり面白そうには思えないのだが、実際に遊んでみると、これはリアルタイムのパズルゲーム以外の何物でもなく、管制対象の飛行機が増えてくると難易度の高いシューティングゲーム並みの早業で、マウスやキーボードを操作しなければならなくなる。ぼんやりしていると、あっという間に画面上の航空機は混乱状態になり、パイロットたちの不平不満で無線通信はパンクし、ニアミスや衝突事故が発生してゲームオーバーになってしまうのだ。

航空路上を次々に飛んでくる航空機の行列の中に、空港からの出発便をうまくはめこむことができたり、四方八方から押し寄せてくる到着便を1本の列に整然と誘導できた場合の達成感、グラフィックやサウンドは凝りまくっているのに内容は薄っぺらなものが多いコンソール用ゲームでは得られないものだ。

そもそも、昔のコンピュータゲーム

は、\$とか%とか@などの単なる記号に対して、多大な想像力を働かせて感情移入し、その世界に没入するというものが多くて、これがまた多くのユーザーの支持を得ていたものだ。いや、コンピュータゲームに限らず、およそ「ゲーム」というものは、将棋やチェスの「駒」、石ころやビー球などといった、実物とはかけ離れたシンボルに感情移入して楽しむのが本来の姿だと思う。

むろん、リアルタイム3Dグラフィックスの世界を否定するつもりはなく、自分もフライトシミュレータゲームなどでそのありがたみを享受しているのだが、無機質な物理シミュレーションの世界ではなく、人間の心象を表現する世界で「リアルな3D表現」というものが成立するかどうかについては疑わしく思っている。マネキン人形みたいな「3D美少女」がディスプレイ画面上でニタリと微笑むのを見て吐き気を覚えるのは、筆者の感性がひねくれているからなのだろうか？

Linuxのファイルシステム

さて、前回Linuxで扱われているファイルの種類の見分け方について説明

したのだが、今回はファイルの中身ではなく、ファイルそのものの仕組みについて考えてみたい。ファイルの中身や扱い方が問題なのであって、**ファイルの仕組みなんて一般ユーザーには関係ない**と考える人もいるだろう。もちろん、アプリケーションだけを使っているぶんにはあまりファイルの仕組みを意識することはないだろうし、それを知らないからといって、すぐに困ったことになるわけでもない。しかし、見かけ上は同じように見えるのだが、Linuxのファイルの仕組みは、WindowsやMacintoshなどとは根本的に違う部分があるのだ。ファイルを移動したりコピーしたり削除したりする場合に、ひょんなことからその違いが顔を出して、びっくりさせられることがある。

ここ数年で、UNIXやLinuxのファイルシステム自体やファイル操作コマンドには、大小さまざまな改良が加えられてきていて、以前は不便だったところが現在では改善されていることも多い。このため、表面上はLinux (UNIX) のファイルシステムに固有の特徴も見えにくくなっているのだが、基本的な構成や特性は変わっていない。この特性を少しでも理解していれば、それをうまく利用して他人をびっくりさせるようなファイル操作技を習得することもできる。

そもそも「ファイル」とは？

前回説明したように、「ファイル」とはコンピュータの記憶装置（主にディスク装置）上に記録されているデータの塊のことで、ユーザーやプログラムがつけた名前を指定することで、その内容を読み出したり、書き換えたりできるものである。この「ファイル」

の記録方法やアクセス方法などを定めたものが**ファイルシステム**であり、Linuxに限らず、およそ「OS（オペレーティングシステム）」と呼ばれるものは、必ずといってよいほど、なんらかのファイルシステムを備えている。

たとえば、Windows（あるいはMS-DOS）は、「FATファイルシステム」あるいは「DOSファイルシステム」と呼ばれるファイルシステムを持っていて、ハードディスクやその他の記憶媒体上のデータを管理できるようになっている。また、Macintoshでも「HFS」（Hierarchical File System）や「HFS+」といった独自のファイルシステムが使われている。

ひるがえってLinuxではどうかというと、現在はext2というファイルシステムが**主に**使われている。「主に」という断り書きをしたのは、Linuxはこのext2ファイルシステムだけでなく、DOSのFAT、MacintoshのHFSなど、他のOS用のものを含めたさまざまなファイルシステムを「直接」利用できるからである。この件については、いずれまた触れることにする。「ext2ファイルシステム」の正式名称は「Linux second extended file system」で、

あえて日本語に訳せば「Linux拡張ファイルシステム第2版」となるだろう。

ext2ファイルシステムは、いわゆるUNIXファイルシステムと同等の機能を持っていて、ファイル管理の方法はUNIXファイルシステムと変わらない。いや、ひとくちに「UNIXファイルシステム」といっても、UNIXを実装するメーカーによってさまざまな「変種」があるので、ext2ファイルシステムもその**変種のひとつ**といったほうがすっきりするかもしれない。

ext2ファイルシステムを含むUNIXファイルシステムの特徴のひとつは、**階層化ファイルシステム**により、複数のディレクトリを使ったファイルの整理ができる点にある。ルートディレクトリ（/）を根っこに枝分かれして成長する「木」のような構造で、多数のファイルを分類整理できるようになっているのだ。階層化ファイルシステムについては、WindowsやMacintoshなどでもおなじみのものなので、改めて説明することもないだろう。実際、Linuxの階層化ディレクトリもWindowsのそれも、ユーザーの側から見ればほとんど違いはなくて、ファイルの操作もまったく同じようにで

画面1 航空管制シミュレーション「ATCC」
DOS用アプリケーションなので、Linux上のDOSエミュレータでも動くはずだが、まだ成功していない。本物の管制用レーダースクリーンとほぼ同じ内容が表示されるが、この文字記号だけが並ぶ画面が、「3次元」に見えるようになることが管制官への第一歩だという...
....



きる（もちろん、使用するGUIツールやコマンドラインコマンドは違うが）。

「ディレクトリ」の正体

階層化されていないファイルシステム、たとえば、大昔のMS-DOSやCP/Mなどでは、ディスク装置（あるいはフロッピーディスク）1台（1枚）ごとに1つのディレクトリしかなくて、すべてのファイルは平板なディレクトリ中の一覧で参照するようになっていた。実は、この名残は現在のWindows（Windows NTは除く）にも残っていて、「FAT16」形式でフォーマットされたドライブ（メーカー製のWindows 98プレインストールPCでは、CドライブがFAT16形式になっている場合が多い）では、各ドライブのルートディレクトリには最大256個（いわゆる「8.3形式」の短いファイル名ばかりの場合は512個）のファイルしか作ることができない（画面2）。これは、FAT16形式の場合、大昔の平板なファイルシステムのディレクトリ

構造が、ルートディレクトリに限ってそのまま使われているからだ。ところが、ルートディレクトリ以外のサブディレクトリについては、このような制限はない。ディスク容量の許す限り、いくらでも多数のファイルを作ることができるのだ。これは、サブディレクトリ中のファイルについての情報を、自由に容量を増やすことのできる特殊なファイル中に保存するようにしてあるからだ（より新しい「FAT32」形式では、ルートディレクトリもそのように改良された）。実は、MS-DOS（Windows）の階層化ファイルシステムは、UNIXのそれを真似してつくられたのだ。もちろん、大元のファイルシステムはまったく別物なので、外見上は似ていても中身はまったく異なる。

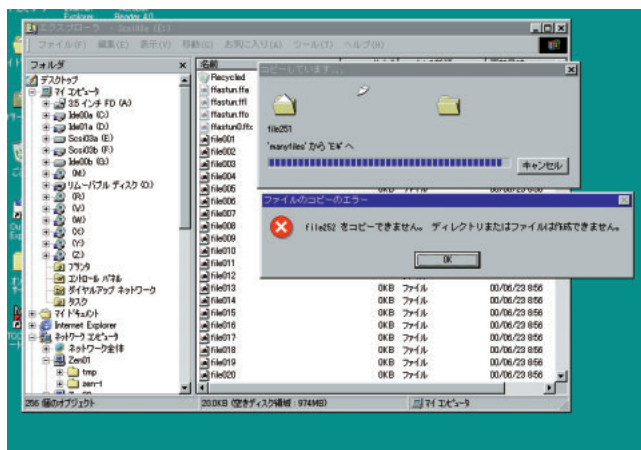
Linuxのext2ファイルシステムの場合も、UNIXファイルシステムに倣って、ディスク中に記録されているファイルの情報を、「ディレクトリファイル」という特殊なファイルに保存している。新しくディレクトリを作成する場合は、このディレクトリファイルを

新たに追加するだけでよく、ディレクトリ中のファイルの数が増えたとしても、通常のファイルと同様に、それに応じてファイルのサイズを増やせばよいのだ。もちろん、このような処理はLinux本体が自動的に行ってくれるので、ユーザーやアプリケーションのプログラマーがそれを意識する必要はない。

実は、ディレクトリファイルの実体はただのデータファイルでしかないのだが、ユーザーが勝手に読み出したり書き換えたりできないようになっている（画面3）。ユーザーやアプリケーションプログラムが勝手にディレクトリファイルの中身を書き換えてしまうと、ファイルにアクセスできなくなるようなトラブルが発生する危険性が高いからだ。

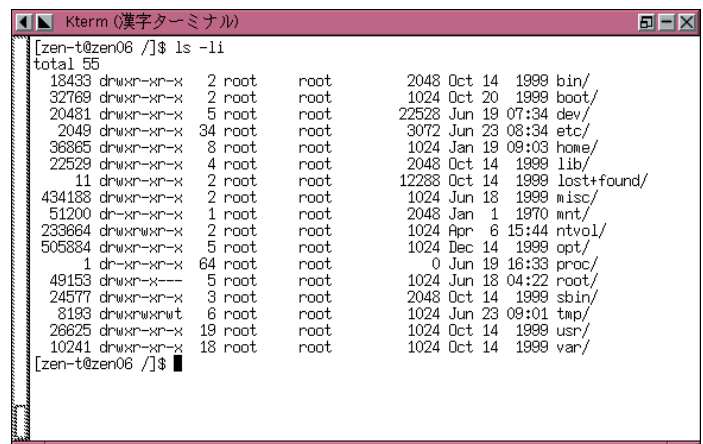
ファイル名の実体

Linuxのファイルについての情報は、そのファイルが存在するディレクトリの「ディレクトリファイル」の中に、一覧表のような形式で記録されている。



画面2 Windows 98の「ルートディレクトリ」

Windows 98では、FAT16形式でフォーマットされたドライブのルートディレクトリに空のファイルばかりを大量に作成（コピー）しようとする、257個目のファイルをコピーしようとした時点でこのようなエラーが出る（隠しファイルなども個数に含まれている）。ドライブをFAT32形式でフォーマットしなおしたり、Windows NTやWindows 2000の場合なら、NTFS形式でフォーマットしてあればこのような制限はない。もちろん、Linuxでもext2ファイルシステムでディスクをフォーマットすればこのような制限はない。



画面3 ディレクトリの正体

lsコマンドに-lオプションと-iオプション（後述する「iノード番号」を表示）を与えて実行すると、通常のファイルだけでなくディレクトリにも「ファイルサイズ」が表示されているのに気づくだろう。同じようなことをMS-DOSやWindowsのDIRコマンドで行ってみても、ディレクトリには「サイズ」に相当する値は表示されない。Linuxのディレクトリが内部的にはファイルの一種であることと表れである。固有のiノード番号が表示されることから、ディレクトリがファイルであることがわかる。

しかし、実際にディレクトリファイルに記録されているのは、ファイル名の文字列と1つの数値だけである。この数値は、ディスク上のどこにファイルのデータが記録されているかを直接示すものではない。それでは、データのありかはどこに記録されているかというと、ディレクトリファイルとは別に、各ディスク（パーティション）ごとに、ディスク中のファイルの住民基本台帳とでもいうべき一種のデータベースがあるのだ。このデータベースには、各ファイルごとにつけられた背番号であるiノード番号と、ディスク上で実際にデータが記録されている場所（ブロック番号）の一覧が記録されている。実は、先ほどのディレクトリファイルにファイル名と組み合わせで記録されていた「数値」は、そのファイルについての情報が何番目のiノードに記録されているかを示すためのものだったのだ（図1）。

少し複雑な話になってしまったので、混乱している読者もいるかもしれないが、簡単にいえば、Linuxのファイルは、実際には「ファイル名」ではなく「背番号」（あるいは、アメリカの「社会保障番号」のようなものと考えてもよい）で管理されているのである。文字列のファイル名が使われるのは、最初にファイルを開くときだけで、あとはすべてこの背番号（iノード番号）を使ってデータの読み書きをしていると考えればよい。

わざわざこのような管理方法を使っているのには、複数の理由があるのだが、もっとも端的な理由は、コンピュータは文字列の扱いが苦手だからだろう。人間にとって便利な「ファイル名」も、コンピュータにとっては意味のないデータ列でしかなく、単純な数値である「iノード番号」

でファイルを扱ったほうが処理が簡単に済むからだ。

個々のファイルのiノード番号がどうなっているかは、簡単に調べることができる。lsコマンドに-iオプションをつけて実行すればよいのだ。

```
% ls -i file1.txt
```

すると、

```
200733 file1.txt
```

といったように、ファイル名の前に6桁の数値が表示される。これがそのファイルのiノード番号なのだ。もちろん、iノード番号がわかったところで、この値はユーザーにはあまり関係はない。この番号をユーザーが直接使うようなことは、まずありえないからだ。ただし、後述するファイルの「リンク」の関係を調べるときには、役に立つこともある。

ファイルのコピーと移動

さて、もう少し現実的な話題に立ち返ることにしよう。パソコンの「ファイル操作」と聞いて、真っ先に思い出すのはファイルのコピーと移動だろう。Linuxのコマンドラインでこれらの操作を行うコマンドは、コピーなら

cpコマンド、移動ならmvコマンドを使うことは、ご存じだろう。

ファイルのコピーとは、あるファイルとまったく同じ内容のファイルを、同じディレクトリ上に別の名前で作成したり、あるいは別のディレクトリ上に同じ名前のファイルを複製するような作業である。たとえば、同一のディレクトリ上に別の名前で複製を作りたいければ、

```
% cp file1.txt file2.txt
```

のように入力するし、別のディレクトリに同じ名前のファイルの複製を作りたいければ、

```
% cp file1.txt /tmp
```

のように入力する。

このとき、Linuxの内部で行われている処理についても、特段変わったことはない。コピー先となるファイルを作ってから、そこに元のファイルの内容を流し込むだけである。したがって、cpコマンドを作る代わりに、次のようなコマンドを実行しても、まったく同じことができる。

念のために言うと、>という記号は、コマンドの出力を指定したファイルに「流し込む」ように指定するためのものである。

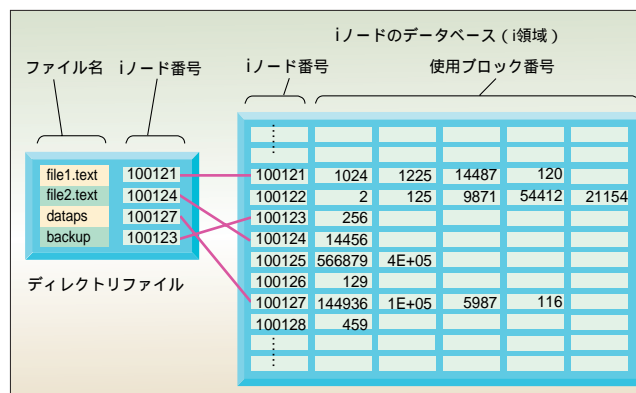


図1 ディレクトリファイルとiノードの関係
ディレクトリファイルに記録されているのは、基本的にファイル名とiノード番号の対だけである。ファイルの中身にアクセスするときは、iノード番号を元にiノードのデータベースを調べて、実際のデータの記録される場所（データブロックの番号）を知る。もちろん、これらの処理はすべてLinuxの内部で行われるので、ユーザーやアプリケーションのプログラマの手が煩わされることはない。


```
% cat file1.txt > file2.txt
```

ただし、別のディレクトリに同名のファイルの複製を作りたい場合は、cpコマンドとは異なり、複製先のファイル名もいちいち指定してやる必要がある。

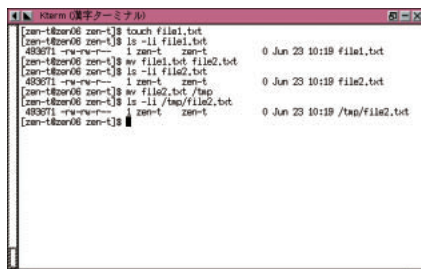
```
% cat file1.txt > /tmp/file1.txt
```

ファイルの移動の場合、ディレクトリは現在のままで、ファイルの名前を変えたい場合も、別のディレクトリに（名前を変えずに）ファイルを移動したい場合も、mvコマンドの使い方はcpコマンドとほとんど変わらない。

```
% mv file1.txt file2.txt
```

```
% mv file1.txt /tmp
```

しかし、内部で行われていることはcpコマンドに比べると少し複雑である。まず、Linuxの場合、mvコマンドを実行しても、ファイルそのものは一切動かされないのである。mvコマンドの前後で、ファイルのiノード番号を調べてみると分かる（画面4）。iノード番号が移動の前後で変わらないということは、ファイルそのものには一切手付けられていないことを意味する。mvコマンドの結果変わったのは、そのファイルの情報を記録してい



画面4 ファイルの移動とiノード番号
同じディレクトリ上でmv（ファイル名を変更）しても、ファイルを別のディレクトリに移動しても、iノード番号は変わらない。すなわち、名前や記録されているディレクトリファイルが変わるだけで、ファイルの本体は一切動いていないことを意味するのだ。

るディレクトリファイルの内容なのだ。iノードの一覧表にあるファイルの情報はまったく変わっておらず、ファイルのデータが保存されているディスク上の位置もまったく変わらないのだ（図2）

ディスク間の移動

ところで、上記のmvコマンドの実験を自分でやってみると、前後でiノード番号が変わってしまう場合があるかもしれない。iノード番号が変わらないなんてウソじゃないかという抗議が来そうだが、場合によってはファイル自体が動く場合もあるのだ。それは、移動元のディレクトリと移動先のディレクトリが、別のディスク（あるいはパーティション）にある場合である。「パーティション」とは何かについては、ここでは述べないが、Windowsでいうところの「Cドライブ」、「Dドライブ」などというときの「ドライブ」と同じもの思っていてほしい。

たとえば、/homeというディレクトリと/tmpというディレクトリが別々のパーティション上に存在する場合、

```
% mv /home/file1.txt /tmp
```

というコマンドを実行すると、その前後でfile1.txtのiノード番号は変わるはずだ。ここでは、実はiノード番号が変わっているだけでなく、ファイルの記録されている物理的な場所も、/homeのあるパーティションから/tmpのあるパーティションに移動しているのだ。先ほど「iノード」について説明したときに、「各ディスク（パーティション）ごとに存在するファイルのデータベース」という意味のことを述べたが、iノード番号も、それぞれのディスクごと

に別々に割り振られるようになっている。したがって、ファイルのあるパーティションから別のパーティションに移動した場合は、移動先のパーティションでまったく新しいiノード番号が割り振られることになるので、移動前後でiノード番号が変わるのである（ごくごくまれに、移動前後でiノード番号が一致することもあるだろうが、これはたんなる「まぐれ」である）。

逆にいえば、Aパーティション中のiノード200001番とBパーティションのiノード200001番は、番号は同じだが表している物理的な場所は、まったく別々ということである。これは、住所を表すときに、Aという町に「中央1-2-3」という住所があるのと同時に、Bという町にも同じ「中央1-2-3」という住所がありえるのと同じである。

実は、昔のUNIXでは、mvコマンドでは別々のパーティションにまたがったファイルの移動はできなかった。先ほどの例と同じように、

```
% mv /home/file1.txt /tmp
```

を実行すると、ファイルシステムが違うなどというエラーメッセージが表示されることがあったのだ。このような場合は、面倒でも、

```
% cp /home/file1.txt /tmp
```

```
% rm /home/file1.txt
```

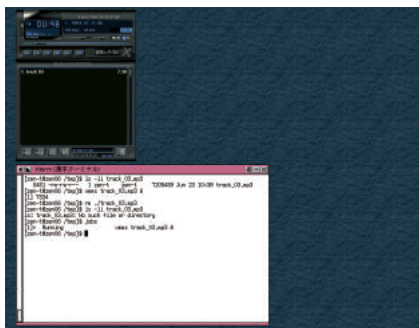
といったように、cpコマンドを使ってファイルを複製してから、元のファイルをrmコマンドで消去する必要があったのだ。ところが、現在のLinuxのmvコマンドはこの点が改善されていて、パーティションが異なる場合でも、mvコマンド一発でファイルの移動ができるようになっている。ただし、

この場合でも内部では上記のような「cpとrm」の組み合わせとほぼ同じ処理が行われているのだ。

この違いをはっきりと意識するのは、大量のファイルを移動するような場合だ。たとえば、画像や音声など数百Mバイトのファイルをmvコマンドで移動する場合、同じパーティション上の移動なら、まさに一瞬で終わるのだが、別のパーティションにあるディレクトリに移動する場合、ファイルの内容そのものをコピーすることになるので、長時間ディスクへのアクセスが止まらないことになる。ましてや、NFSやSambaなどネットワーク経由で共有しているドライブの間でファイルを移動しようとする、数分間も待たされる場合もある。あたりまえのことのようだが覚えておいてほしい。

ファイルの削除

ファイル操作の最後の削除だが、これはもっとも簡単そうに見えて、実はもっとも複雑な処理である。コマンドラインからファイルを削除するのは、先ほどの例でもちょっと出てきたよう



画面5 使用中のファイルをrmで削除してみる
ちょっと乱暴のように思うかもしれないが、xmmsで再生中のmp3ファイルをいきなりrmコマンドで消去してみる。ディレクトリからはファイルは消え去るのだが、xmmsは何事もなかったかのようにファイルの再生を続けている。ファイルの実体は消去されていないのだ。このあと、xmmsの実行を終了すれば、いずれはファイルのデータも完全に消去される（どの時点で消去されるかは不定である）。

に、rmコマンド一発でできる。ところが、rmコマンドは実はファイルのデータそのものを削除したり、あるいはiノードのデータベースを変更したりはしない。単に、ディレクトリファイル中のそのファイルに関する情報を削除するだけなのだ。この操作を、UNIXの専門用語でアンリンクと呼んでいる。アンリンクされたファイルは時間をおかずにシステムによって本当に消去されるのだが、アンリンクが行われた時点で、そのファイルが何者かによって参照されている場合は、ディレクトリファイル上の情報は消去され、lsコマンドなどからは見えなくなるものの、iノード番号によって参照されるファイルの実体そのものは生きつづけている。

ここで、読者にも簡単な実験をしてもらいたい。適当な音声ファイル（MP3形式の音楽データなど）を1つ用意して、それを適当なMP3再生ソフト（xmmsなど）で再生する。再生中におもむろにそのファイルをrmコマンドで削除すると、当然lsコマンドではそのファイルを見ることはできなくなり、一見、ファイルは完全になくなったかのように見える。しかし、MP3再生ソフトは、何事もなかったかのように、ファイルの再生を最後まで続けることができるのだ（画面5）。

同様のことをWindows 98やWindows

2000でやってみると、ファイルを削除しようとしたとたんに、ファイルにアクセスできませんあるいは共有違反ですというエラーメッセージが出る。Windowsの場合は、Linuxとはまったく異なるファイル管理方法を採用して、ファイルの削除を行うと、直ちにファイルの内容そのものを削除（あるいは「ごみ箱」へ移動）するようになっているからだ。あるアプリケーションが使用中のファイルを削除してしまうと困ったことになるので、ファイルの使用中にはファイルを一切削除できないようにロックしているのだ。

WindowsもLinuxもどちらもマルチタスクのOSなので、複数のプログラムから同じファイルを操作しようとした場合の対策はとられているのだが、ファイルの「削除」に関しては、このようにその方法には大きな違いがあるのだ。もちろん、この違いはWindowsの祖先がシングルタスクのMS-DOSであることとも関係する。

最後に「リンク」について

ところで、先ほどrmコマンドは「アンリンク」という処理を行っていると述べたが、アンリンクとは「リンクを解除する」という意味である。ではリンクとはなんだろうか。次回はこの「リンク」について話することにしよう。

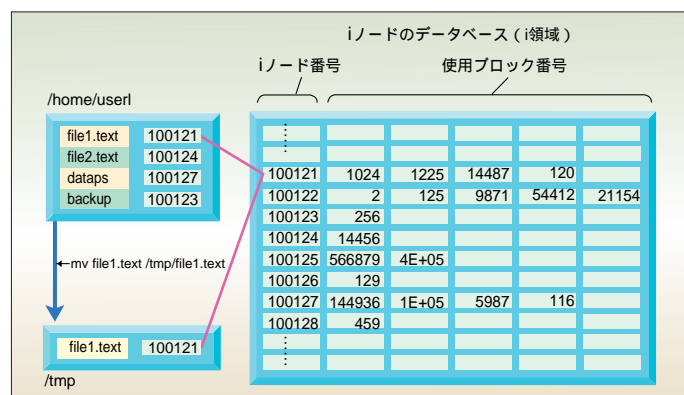


図2 ファイルの「移動」
同一パーティション上でのファイルの移動は、ディレクトリファイル中のファイル情報を書き換えたり、別のディレクトリファイルに移すだけで、ファイルのiノード番号や実際にファイル中のデータが記録されている場所は一切変わらない。

Javaプログラミング入門

Javaで3Dゲームを作ろう

今回は、Javaはすべてクラス単位で構成され、そのクラスに持たせたい機能をメソッドとして記述することを学びました。「クラス」が人間だとすれば、「メソッド」は歩くとか食べるという振る舞いだと考えればよいでしょう。今回は、「スレッド」という基本概念を学びます。

第2回 スレッドとクラスを作る

文：おもてじゅんいち / かざぐるま
Text: Junichi Omote / Kazaguruma

前回の宿題 「クラスとメソッド」

本題に入る前に、まず前回の宿題を片づけておきましょう。「Hello java world!」というテキストを表示するプログラム(リスト1)の「g.drawString()」は、どのクラスのメソッドになるのか」という内容でした。クラスの機能を見分けるための練習です。

さて、このプログラムコードで宣言されているクラスは「app01」というAppletクラスだけです。であれば、「g.drawString()」はAppletクラス「app01」のメソッドなのでしょうか。それを判別する鍵は、「g。」の部分にあります。

あるクラスのメソッドを実際に呼び出して使用するときは、

クラス名.メソッド名()

と書きます。たとえば、Appletクラス「app01」の「paint」メソッドを呼び出すときは、

```
app01.paint()
```

と書きます。つまり、問題の「g.drawString()」の「g」はクラス名だったのです。それでは「g」というクラスはどこにあったのでしょうか。コードの最初から順に、「g」という名前が宣言されていそうな箇所を探してみましょう。

```
public void paint (Graphics g) {
```

というpaintメソッドの始まりを表す行

の、()内を見てください。ここに、

```
Graphics g
```

という記述があります。()内に書かれているのは、paintメソッドと受け渡しをするためのパラメータですが、同時に、ここで『「g」という変数はGraphicsクラスを扱うよ』と宣言しているのです。

paintメソッドに渡されるパラメータはGraphicsクラスであり、そのGraphicsクラスをここでは、『「g」という変数を使って取り扱う』ということはこの1行で表しています。

リスト1 テキストを表示する (app01.java)

```
import java.applet.*;
import java.awt.*;
/*
<applet code="app01" width=200 height=200></applet>
*/

public class app01 extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello java world !",20,100);
    }
}
```


Graphicsクラスは、文字やグラフィックを描画する対象のことで、ここではWebブラウザがアプレットに対して、「ここに描画しなさい」という場所（対象）を指示するために、paintクラスのパラメータとして渡されてきます。つまり、問題の「g.drawString()」は、Graphicsクラス「g」のメソッドであり、このクラスはpaintメソッドのパラメータとして、Webブラウザからアプレットクラス「app01」に渡されるといことになります。

これからプログラミングをするうえで、それぞれのクラスの機能を整理することが重要となってきます。

並列処理を可能にする スレッド

宿題の解説が終わったところで、先月で少し紹介した「動きのあるアプレット」を作ってみましょう。

リスト2は先月と同じプログラム、「app03.java」です。

この「動きのある」アプレットでは、paintメソッドのほかに、initメソッド、runメソッドといった記述をしていく必要があります。それにしても、単に「動き」をつけるだけでなぜ急にプログラムの行数が増えてしまうのでしょうか。

アプレットをWebブラウザ上で実行するとき、ブラウザ側から単純に「実行せよ」という命令を受けるだけではなく、じつはタイミングに合わせ、結構細かい指示で「動き」を作っているからなのです。

主な指示として、

- ・初期化せよ (init)
- ・スタートせよ (start)
- ・実行せよ (run)
- ・描画せよ (paint)
- ・描き直せ (update)

・ストップせよ (stop)

などがあります。そして、Webブラウザが指示を出すということは、アプレット側に用意された（ ）内のメソッドを呼び出すということなのです。

まず、Webブラウザからアプレットが最初に読み込まれた時に、アプレット内のinitメソッドが実行され、次にstartメソッドが実行され、runメソッドと続きます。画面を描画するときにはpaintメソッドが実行され、ウィンドウが移動したり、ウィンドウサイズが変わったりして画面に「欠け」ができたときに、updateメソッドで描き直しを行います。Webブラウザがクローズしたり終了する場合は、同時にアプレットを終了させるためにstopメソッドが実行されます。

このように、アプレット内のメソッドを必要に応じて呼び出し、実行することによって、Webブラウザとアプレットは連携をとっているのです。

「動きのない」アプレットであれば、Webブラウザが「描画しなさい」と指示したときだけ描画すればいいのですが、アプレット側で常に何かを動かしている場合は、単にWebブラウザの号令で実行するだけでなく、独立して自分の仕事をし続けていく機構が必要になるのです。

そこで、メインの（ここではWebブラウザ）プログラムとは別に、独立した形で実行できる軽量プログラム「ス

リスト2 スレッドを使って円を動かす (app03.java)

```
import java.applet.*;
import java.awt.*;
/*
    <applet code="app03" width=200 height=200></applet>
*/
public class app03 extends Applet implements Runnable {
    Thread    t;
    int       x = 0;

    public void init() {
        t = new Thread(this);           // スレッドを開始する
        t.start();
    }

    public void run() {
        try {
            while(true) {
                repaint();              // 再描画を要求する
                Thread.sleep(100);      // 100ms制御を解放
                x += 10;
                if ( x > 200 )          x = 0;
            }
        } catch(Exception e) {}
    }

    public void paint(Graphics g) {
        g.fillOval(x,100,30,30);
    }
}
```

レッド」を作って、「グラフィックを動かしながら描画する」という仕事をさせることとなります。スレッドとはまさに、もうひとつの頭脳のように別個の独立したプログラムとして動きます。このように「動きのある」アプレットには不可欠なもので非常に重宝しますが、他のプログラム言語では、そのためにプログラミングが難解になったりします。Javaではもともとスレッドの機構を備えていますので、比較的簡単にスレッドを作ることができるので安心して下さい。スレッドの概念が今一つわからないのであれば、アプレット自身がWebブラウザの動きとは独立して動くことができる（スレッド化する）ようになったらと思えばよいでしょう。

それでは実践です。円を動かすプログラム（リスト2）を例にして考えてみましょう。まず、スレッドに関する記述は、initメソッド内の2行、

```
t = new Thread(this);
t.start();
```

と、runメソッドの、

```
Thread.sleep(100);
```

になります。たったこれだけで、立派に独立して働くスレッドを作れます。

スレッドそのものは簡単に実装できるのですが、Webブラウザとの連携には気をつけなければなりません。ここで必要となってくるのが、先ほど解説したinitメソッド（初期化）とrunメソッド（実行）です。

まずinitメソッドでスレッドを作成します。スレッドの作成は、アプレットの実行に先立って行っておかなければならないので、アプレットが読み込ま

れてまず実行される、initメソッドに記述しておきます。

アプレットが独立して動き続ける（スレッド化する）ためには、Webブラウザから「実行せよ」といわれた後、runメソッドとrunメソッド内のwhileループを用意します。Webブラウザからrunメソッドを呼び出し、アプレットが「実行せよ」という指示を受け取ったら、

```
while(true) {
    ....
}
```

という記述により、{ から } までを永遠にループ（繰り返し）します。つまり、Webブラウザの動きとは独立してこのループ内のプログラムを何度も何度も実行し続けるのです。

そうはいつてもいつかは終わらなければならぬわけで、Webブラウザのウィンドウが閉じるか、Webブラウザが終了するときだけ例外として、

```
catch(Exception e) {}
```

の行に移り、ループは終了します。少し解説すると、このcatchとその前にあるtryというのは、Javaの重要な言語仕様で、これらは「例外処理」と呼ばれます。いちいちアプレットが「Webブラウザが終了したか」「ウィンドウが閉じられたか」などを監視する必要はなく、「例外が生じたら、ここに飛べ」という記述ができるのです。

スリープとrepaintで「動き」を調節

話は戻って、whileループ内にある、

```
Thread.sleep(100);
```

は、スレッドをスリープさせる処理で、パラメータの数字は1/1000秒単位です。スリープ（眠らせる）といっても、処理が停止するのはあくまでスレッドだけで、その間にOSや他のアプリケーションが動くこととなります。

そもそもスレッドは他のアプリケーションたちと同時に動いているわけではなくて、自分が少し動いたら、他のアプリケーションがまた少し動き、ときどきシステム（OS）も少し動き……ということを繰り返しているわけです。アプリケーションやシステムが処理を譲り合ってマルチタスクが実現されているというわけです。いくつものアプリケーションやOSが0.1秒とか0.01秒単位で切り替わって動いているために、人間の目にはすべて同時に動くように見えているだけなのです。

ですから、このThread.sleep()の数値を小さくすればCPUを占有する時間が長くなり、アプレットの動きはなめらかになります。逆に数値を大きくすれば、他のアプリケーションにかかる時間が長くなるため、アプレットの動きがギクシャクしてきます。マシンのスピードにも関係しますので、適当な数値を試してみてください。

リスト2のpaintメソッドでは円を描画していますが、円を描画するx座標が数値ではなく、xという変数になっていることに注目してください。つまり、この変数xの値を次々と変化させることにより描画する横軸を変えることができ、絵が動くようになるのです。

しかし、paintメソッドで描画するだけでは、描画のタイミングはWebブラウザにまかせることになってしまい、絵の動きに合わせてタイミング良く描画される保証はありません。もしかすると、paintメソッドはアプレットの開始時に一度だけしか実行されないかも

しれないのです。「動きのない」アプレットならそれでもよいのですが、「動きのある」アプレットでは困ります。

そこで、適当なタイミングで画面を更新させるために、runメソッド内に

```
repaint();
```

という記述をしています。これは、Appletクラスのメソッドで、画面を再描画するというものです。実際には、paintメソッドを呼び出すだけなので、このrepaintメソッドが実行されるたびにpaintメソッドも実行されることになり、毎回描画が行われるわけです。

話はそれますが、頭脳明晰な読者の方はrepaintはメソッドなのに、「クラス名・メソッド名」のクラス名の部分がないことに気づかれたかもしれません。もちろんこれは間違いではなく、repaintメソッドはAppletクラスのメソッドなので、同じクラスの内部から呼び出すときはいちいちクラス名をつけなくてもよいのです。厳密には、

```
app03.repaint();
```

となります。または、

```
this.repaint();
```

と記述することもできます。thisとは、自分自身のクラスを表し、ここではAppletクラスであるapp03のことを意味します。しかし、省略することができるということも覚えておきましょう。

話を戻しましょう。リスト2のループの最後のrunメソッドにある、

```
x += 10;
if ( x > 200 )    x = 0;
```

は、円のx座標に10を加えてゆき、200を超えるとまた0に戻すということを繰り返しています。これは、paintメソッド内のg.fillOval()のx座標を変更するための記述です。

以上が、スレッドを使った「動きのある」アプレットの構造です。円を左から右に動かすだけなのに、結構複雑なプログラムになるものだという印象を受けたかもしれませんが、この構造は最終目標の「3Dブロックくずし」でも同じで、ここで学んだ骨組みに肉付けしていくことになります。「動きのある」アプレットを作るために必要な基本構造は、リスト2をマスターしていればほぼ十分でしょう。

クラスを作って ボールを動かす

Javaでは、すべてのプログラムがクラスで構成されると解説しました。

これまでは、それぞれのプログラムはAppletクラスだけを記述していましたが、まさにその通りなのですが、何でもかんでもAppletクラスに含めてしまうのはあまりよくありません。

そうなるとpaintクラスなどにどんどんプログラムコードを追加していくことになり、作っている本人も何をやっているかわかりにくくなるのです。5行や10行くらいまでなら、なんとなく把握できますが、これが30行、50行、100行.....ともなると、不具合が生じたときに、どこを直せばよいのか見つけるのがひと苦労です。

そこで、自分でクラスを作って追加していくのです。

ひとつのクラスを大きくしていくよりも、いくつかのクラスに分けて部品化していくほうがプログラムが見やすくなり、修正も楽になります。また、一度作ったクラスは、別のプログラム

を作った場合にそのまま流用できる場合もあります。

本連載の「3Dブロックくずし」は、ボール、ラケット、ブロック、壁、そしてアプレット本体という感じで、登場する部品がはっきりしています。この5つの構成部品をそのままクラスとして考えれば、わかりやすいプログラムを作ることができるでしょう。

では、リスト2で動かした円をボールとみなして、Ballクラスを作成してみましょう。

クラスを作るときは、そのクラスに関する動きや描画といった機能を、もれなくメソッドにして記述しなければなりません。リスト2でのボールの機能は、「動き」として、

```
x += 10;
if ( x > 200 )    x = 0;
```

の2行であり、描画としては、

```
g.fillOval(x,100,30,30);
```

の1行になります。

この機能が把握できたら、クラス名、動きと描写それぞれのメソッド名を決めます。クラス名はBall、動きのメソッド名をprocess()、描画のメソッド名をdisp()とすると、もう新しいクラスができるでしょう。

```
class Ball {
    int x;
    Ball() {
        x = 0;
    }
    void process() {
        x += 10;
        if ( x > 200 )    x = 0;
```



```

}

void disp(Graphics g) {
    g.fillOval(x,100,30,30);
}
}

```

これで、processと、dispというメソッドを持ったBallクラスが完成しました。

よく見ると、もう1つBallというクラスと同じ名前のメソッドが登場していますが、これはコンストラクタと呼ばれるもので初期化などのために使います。コンストラクタは、このクラスが最初に作成されるときに自動的に実行されます。

ボールを反射させるには

それでは、再び実践です。このBallクラスを用いてアプレットを作成してみましょう。リスト2と同じような動きではおもしろくないので、ボールを壁に反射させてみます。

目標は3Dですが、その前にとりあえず2Dでの動き、すなわちタテとヨコでボールを反射させることにします。

そもそも、ボールが反射するということはどういうことなのでしょう。今、ボールが動き回る範囲をx、y座標とも0~400とすると、ボールの座標がx=400になったときに壁に当たるといふことになり、反射させなければなりません。そのとき、ボールが壁に対して斜めであっても、垂直であっても、右の壁(x=400)に当たると、横の進行方向だけが逆転します。縦方向は何も変わりません。現実世界では摩擦等の関係により、縦方向にも変化はあるそうなのですが、ここでは考えないことにします。

このように、x=0またはx=400のと

リスト3 ボールが壁に当たって反射する (ball2d.java)

```

import java.applet.*;
import java.awt.*;
/*
   <applet code="ball2d" width=400 height=400></applet>
*/
//-----
// Applet Class
//-----
public class ball2d extends Applet implements Runnable {
    Thread    t;
    Ball      ball;

    public void init() {
        ball = new Ball(400, 400);           // Ballクラスを作成
        t = new Thread(this);
        t.start();
    }

    public void run() {
        try {
            while(true) {
                ball.process();           // ボールを動かす
                repaint();
                Thread.sleep(10);
            }
        } catch (Exception e) {}
    }

    public void paint(Graphics g) {
        g.setColor(Color.white);           // 背景をクリア
        g.fillRect(0, 0, 400, 400);

        ball.disp(g);                       // ボール描画
    }
}

//-----
// Ball Class
//-----
class Ball{
    int    vx,vy;
    int    gx,gy,gr;

    Ball(int w, int h) {                   // コンストラクタ
        gx = 80;                           // ボールの初期位置
        gy = 130;

        vx = 5;                             // ボールの速度
        vy = 2;

        gr = 40;                             // ボールの直径
        process();
    }

    void process() {                       //---- ボールの動き
        gx += vx;                           //ボールを動かす
        gy += vy;

        // 反射
        if ( gx > 360 && vx > 0 )    vx = -vx;
        if ( gx < 40 && vx < 0 )    vx = -vx;
        if ( gy > 360 && vy > 0 )    vy = -vy;
        if ( gy < 40 && vy < 0 )    vy = -vy;
    }

    public void disp(Graphics g) {         //---- 表示
        g.setColor(Color.red);
        g.fillOval(gx-gr/2, gy-gr/2, gr, gr);
    }
}

```

きの反射は横方向の速度vxの符号を反対にし、y=0またはy=400のときの反射は縦方向の速度vyの符号を反対にする、「反射」を表現できるのです。

この処理は、Ballクラス内の動きのメソッドであるprocess()で行います。壁に当たったかどうかの判断は、ボールの座標に加えて厚み（直径分）を考慮することも忘れてはいけません。

それでは、本体のAppletクラス内の記述にかかります。initメソッドでBallクラスを作成し、ballという変数で取り扱えるように記述しました（リスト3）。このとき、自動的にコンストラクタと呼ばれるBallメソッドが呼ばれますので、画面の大きさを設定します。

次に、runメソッドでは先程の、

```
x += 10;
if ( x > 200 )    x = 0;
```

の代わりに、

```
ball.process();
```

として、先ほど作ったBallクラスのprocessメソッドを呼び出し、ボールの動きを全面的に任せてしまいます。Appletクラスからはprocessメソッドを呼び出す作業をするだけで、その先でどんなことが行われているかには関与していません。

続いてはpaintメソッドです。ここで

も先程の、

```
g.fillOval(x,100,30,30);
```

の代わりに、

```
ball.disp(g);
```

として、これもBallクラスにボールの描画をまかせています。描画するには、その対象となるGraphicsクラスが必要になりますので、paintメソッドのパラメータとして渡されたGraphicsクラスを、そのままBallクラスのdispメソッドにパラメータとして渡すことを忘れないようにしましょう。また、ball.disp(g);の前にある、

```
g.setColor(Color.white);
g.fillRect(0, 0, 400, 400);
```

は、背景と直前に描いたボールを消すために必要な記述です。

では、早速このリストを入力してコンパイルし、実行して動作チェックしてみましょう。

いよいよWebブラウザでの実行

これまで、アプレットビューアでアプレットを実行させてきましたが、そろそろWebブラウザ上で実行させるようにしてみましょう。

Webブラウザでアプレットを実行するには、そのためのhtmlファイルを用意する必要があります。リスト3のアプレットを実行させるためのHTMLは、リスト4のようになります。

アプレットのためのタグは、

```
<applet.....></applet>
```

になり、ここに、

```
code = "ball2d"
codebase = "."
width=400
height=400
```

という内容を記述します。

このcodeには、実行するアプレット名を記述します。これは、プログラムを書いたjavaのファイル名（拡張子java）、プログラム内のアプレット名と一致していなければなりません（コンパイルした結果、生成されるファイルも、ball2d.classのように同じ名前になっています）。

codebaseは、実行するアプレットの拡張子classのファイルが存在するディレクトリまたはURLを記述します。ここではhtmlファイルと同じディレクトリにあるということで"."とします。width、heightはアプレット画面の大きさを、リスト2のアプレットについては、縦横ともに400ですので、width=400、height=400になります。

<applet>タグ以外は、普通のhtmlです。このように、htmlファイルにアプレットを埋め込むことができるので、自分のホームページなど、htmlだけでは表現力の乏しいと思われる部分や、インタラクティブに反応をさせたい部分に自由自在にアプレットを埋め込んで魅力のあるものにしてください。

リスト4 アプレットを実行するためのhtml (ball2d.html)

```
<html>
<title></title>
<body bgcolor="#ffffff">
<center>
<applet code="ball2d" codebase="." width=400 height=400>
</applet><br>
</center>
</body>
</html>
```

Ruby で行こう

プログラミングに基礎は大事ですが、プログラムから学ぶことも重要です。今回も、前回に引き続き、プログラムを見ながら学ぶことにしましょう。今回のテーマは、「プログラム設計」にしてみました。しかし、設計はなかなか難しい分野です。

第8回 ウォーゲーム

文：赤松智也

Text: Tomoya Akamatsu

今回の例題は、タイトルどおりゲームです。ゲームといっても、自分でプレイするのではなく、コンピュータ自身に遊んでもらいます。今回は、簡単なコンピュータ対戦ゲームのゲーム盤と、ゲームプレイヤーの両方を開発します。ゲームのルールは以下のとおりです。

- ・プレイヤーは2人(2台?)
- ・マス目上のフィールドで争う(デフォルトのフィールドサイズは10×10)
- ・各ターンに移動、攻撃、索敵の各行動を選択できる
- ・移動、攻撃は縦、横、ナナメの8方向に可能
- ・移動は1マスずつ
- ・移動により壁にぶつくと1.0ポイントのマイナス
- ・移動中に相手プレイヤーにぶつくと双方とも1.0ポイントのマイナス
- ・攻撃はその方向すべてに有効だが、離れるほど効果が低い。距離が1マスであった場合1.0ポイントの効果がある。相手に当たったかどうかは知ることができる
- ・索敵は相手の現在位置がわかる(強力すぎ?)
- ・双方のライフの初期値は5.0ポイント

このルールはそれほど吟味していないので、ゲームバランスが悪いかもしれません。実は、私はあまりゲームをしないので.....(そのわりにRogueにはまったりする)。

と、ここまで書いてきて気づいたのですが、これでも

しかすると(もしかなくても)本誌6月号の114ページで紹介された「RealTimeBattle」のチープなまがいものだったりして.....。まあ、これは例題ですから、気を取り直して続けることにしましょう。

ゲームの構成

このゲームプログラムを実現する場合、あなたならどのように構成しますか? 私が考えたのは以下の3つの案です。

A案

ゲーム盤とプレイヤークラスを用意します。ユーザーが実現する各プレイヤーはプレイヤークラスを継承することで実現します。

・メリット

全体がオブジェクト指向で設計されてきれい。オブジェクト指向言語Rubyにふさわしい。

・デメリット

プレイヤープログラムがその気になれば、触ってはいけない部分にアクセスできる。たとえば、敵の位置が常にならなくなったり、自分のライフを増やせる。

B案

A案の欠点を、プレイヤープログラムをRubyのセキュ

リティ機能を使って「信頼できないモード」で動作させることにより克服します。

・メリット

A案のメリットをほとんど失わずに、A案の欠点を克服できる。

・デメリット

A案より複雑になる。それに現在の安定版であるRuby1.4系では、セキュリティ機能はもともと動作するかどうか検証されていないような気がする。1.5系では直ってるバグがそのままになってるようだし。

C案

プレイヤー部分は子プロセスとして起動し、それぞれはstdin/stdoutで通信します。

・メリット

プレイヤーがアクセスすべきでない情報を隠しておくことができる。子プロセスとの通信機能はもともとRubyが持っているし、プログラムの構造が単純になりそう。プレイヤープログラムはRubyに限らず、いろいろなプログラム言語で開発できる。

・デメリット

双方向プロセス間通信は、注意しないとデッドロックになる。通信プロトコルを設計する必要がある。プレイヤーがRubyでなくてもよいというのは、この連載の意味が揺らぐかも。

今回は、A案とB案には不安があるので、単純さを重視してC案を採用します。ああ、ますますRealTimeBattleのまがいものになるなあ（笑）。

プロトコル

C案を採用した以上、プロセス間通信が発生します。となると、通信手順あるいはプロトコルを決める必要がありますね。などというとおおげさに聞こえますが、結局は双方のプログラムがどんなデータをやり取りするのかを決めるだけのことです。

今回は、プレイヤーはすべての情報を親からもらうので、図1のような情報のやり取りが必要ではないかと思えます。

移動コマンドと攻撃コマンドは、8方向のうちの1つを指定します。

あと、現在のライフの状態がわからないとゲームが成立しないような気がするので、コマンドの結果の一部として、現在の自分のライフを送信することにしましょう。

プロセス間通信のプロトコル（と云うほどおおげさなものではありませんが）は、行単位でやり取りします。そうすると、Rubyの行入出力がそのまま使えて便利です。

ゲームが開始すると、親（ゲーム盤）プロセスは起動した子（プレイヤー）プロセスに対して、以下のような5つの数字を送ります。

```
9 9 3 4 5.0
```

これはそれぞれ、「x方向の最大値」、「y方向の最大値」、「現在のx座標」、「現在のy座標」、「ライフポイントの初期値」を意味します。

以後は、プレイヤーからコマンドが送られ、ゲーム盤から返答が来るというターンの繰り返しになります。コマンドは前述のとおり、3種類あります。

移動

移動コマンドは以下の形式になります。

```
move n
```

2番目は移動方向を表す文字列です。これは東西南北の組み合わせで、「n」、「ne」、「e」、「se」、「s」、「sw」、「w」、「nw」のいずれかで指定します。ゲーム盤からの返答は、次のような形式です。

```
4 5 5.0
```

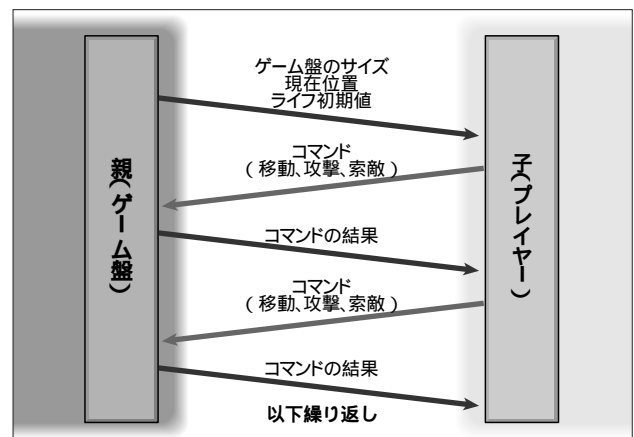


図1 親プロセスと子プロセス間の通信

1番目が移動後のx座標、2番目がy座標、3番目が現在のライフポイントです。

攻撃

攻撃コマンドも、移動コマンドとほぼ同様です。

```
attack nw
```

方向は移動コマンドと同様に、東西南北で指定します。ゲーム盤からの応答は以下のとおりです。

```
hit 5.0 (当たったとき)
```

```
not 5.0 (当たらなかったとき)
```

2番目の値は、自分のライフポイントです。

索敵

索敵は単に、

```
search
```

と送るだけです。すると、応答は以下のようになります。

```
6 4 4.5
```

最初の2つは敵の座標、最後の値は自分のライフポイントです。敵のライフポイントを知る方法はありません。

プログラムの構成

このプログラムのプロセス構成は、図2のようになります。親子間はパイプによってつながっています。パイプというのは、プロセス間をつなげるストリームのことです。ストリームについては、本連載の第4回(2000年4月号)で解説しました。入出力先をファイルにしたり、プログラムにすることが自由にできるのが、UNIXの特長です。

パイプの作り方あるいは繋ぎ方ですが、UNIX本来のC言語によるやり方は、実はなかなか複雑です。Cプログラミングの連載ではないので、簡単にしか説明しませんが、おおよそ以下の手順になります。

1. pipeシステムコールで入出力がペアになったファイルデ

ィスクリプタを生成する。双方向のパイプを作る場合には、これを2回繰り返す

2. forkシステムコールで子プロセスを作る

3. (子プロセス側) dup2システムコールを使って、ファイルディスクリプタを標準入力(または標準出力)に置き換える。dup2が済んだら、ファイルディスクリプタをクローズする

4. (子プロセス側) execvシステムコールなどを使って、実際に子として実行するプログラムを起動する。forkした子プロセスをそのまま使う場合には、このステップは必要ない

5. (親プロセス側) こちらのプロセスでは使わないファイルディスクリプタをクローズする

これで、親プロセスのファイルディスクリプタと子プロセスの標準入出力がつながることになります。なかなか面倒でしょう?

一方、「らくちん」をキーワードとするRubyではどうなるでしょう。実はこれは非常に簡単です。たとえば、random.rbというファイルの子プロセスとして起動するのであれば、

```
open("|random.rb", "r+")
```

または

```
IO.popen("random.rb", "r+")
```

とするだけです。もちろん実際には、上に示したような手順で行われるわけですが、プログラマーはそのようなことは考える必要はありません。ああ、らくちん。

ゲーム盤

さて、実際のプログラムを見てみましょう。ゲーム盤の

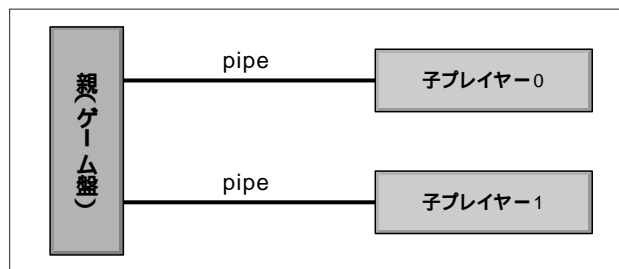


図2 プロセス構成

プログラムはwargame.rbです。まず、リスト1-1を見て下さい。説明のため、リストは一部前後しています。

class Playerの先頭で定数を定義しています。このうち、NUMとTURNは数を要素とする配列になっています。この2つは定数としてではなく、NUM[0]やTURN[0]をクラス全体からアクセスできる変数として使っています。このような変数をクラス変数といいますが、Rubyでは長らくクラス変数が提供されなかったため、このようなテクニックが使われてきました。1.5系ではクラス変数(@@variable)が導入されて、このようなテクニックは不要になったのですが、1.5系はまだ安定版ではありませんし、ほとんどの方はまだ1.4系をお使いでしょうから、ここでは古いテクニックを使うことにします(もしかしら、まつもとさんの予告どおり、今月号が書店に並ぶころには新しい安定版の1.6がリリースされているかもしれません)。

その次のinitializeメソッドは、オブジェクトが新たに生成されたときに呼び出されるメソッドです。C++のコンストラクタに近いといえば近いかもしれません。先ほど説明した子プロセスに対するパイプのオープンと、オブジェクトの処理化を行っています。

attr_readerメソッドは、引数に指定したインスタンス変数の名前から@を付けたメソッドを用意して、外側からインスタンス変数のアクセスを許す、という意味です。

プレイヤーからのコマンドはmove、attack、searchの各メソッドが処理します(moveメソッドはリスト1-2、attackメソッドとsearchメソッドはリスト1-3)。moveメソッドとattackメソッドは、東西南北の各方向に対する

リスト1-1 wargame.rb (その1)

```
#!/usr/bin/env ruby

class Player
  X_MAX = 9
  Y_MAX = 9
  LIFE = 5.0
  NUM = [0]          # クラス変数(プレイヤー番号)
  TURN = [0]         # クラス変数(ターン数)

  def initialize(cmd) # オブジェクト初期化
    @pipe = IO.popen(cmd, "w+")
    @name = sprintf("%s:%d", cmd, NUM[0])
    NUM[0] += 1
    @life = LIFE
    @x = @y = 0
  end

  attr_reader :x, :y, :name, :other, :life
end
```

処理を行うので、case文で分岐しています。そして、出力はインスタンス変数@pipeから参照されているプレイヤープロセスとのIOに対して行います。

残りは、下請けのメソッドです(リスト1-4)。turnメソッドは、各ターンのプレイヤーからのコマンドを受け付けます。startメソッドは、プレイヤープロセスをスタートさせます。overメソッドはゲームオーバーの処理を、

リスト1-2 wargame.rb (その2)

```
def move(dir)          # 移動コマンド処理
  x = @x
  y = @y

  case dir
  when "n" # north
    y -= 1
  when "ne" # north east
    x += 1
    y -= 1
  when "e" # east
    x += 1
  when "se" # south east
    x += 1
  when "s" # south
    y += 1
  when "sw" # south west
    x -= 1
    y += 1
  when "w" # west
    x -= 1
  when "nw" # north west
    x -= 1
    y -= 1
  end

  if x < 0 || X_MAX < x || y < 0 || Y_MAX < y
    # 壁にぶつかったら
    printf "player %s (%d,%d) crashed to wall %s\n",
      @name, @x, @y, dir
    self.damage(1.0)
  elsif x == other.x and y == other.y
    # 敵にぶつかったら
    printf "players crashed each other at (%d,%d)\n",
      other.x, other.y
    other.damage(1.0)
    self.damage(1.0)
  else
    printf "player %s (%d,%d) moved to (%d,%d)\n",
      @name, @x, @y, x, y
    @x = x
    @y = y
  end

  @pipe.printf " %d %d %f\n", @x, @y, @life
end
```


hitメソッドは命中処理を、damageメソッドはダメージ処理を行います。

これでクラスPlayerの定義は終わりです。あとは、srandで乱数を初期化して、game_loopメソッドでゲームをスタートさせます(リスト1-5)。

このプログラムは、各ターンごとに何をしたのかを表示するだけです。どちらかのライフがゼロになったらゲームオーバーです。そして、勝者を表示します。はっきりいって全然おもしろくありません。が、まずはこんなところから始めましょう。

一応、オブジェクト指向っぽくクラスを定義しているのですが、このオブジェクト構成は図3のようになっているだけです。全体を司るゲーム盤オブジェクトも何もありません。ただ、Player0とPlayer1とが交替で自分のターンを実行するだけです。手抜きなのがバレバレですね。

プレイヤーライブラリ

一方、プレイヤープログラムは戦略以外のゲーム盤との

やり取りには共通の処理が多いので、プレイヤー用ライブラリ(player.rb)としてまとめます。

プレイヤーを実装するとき気をつける必要があるのは、標準出力のバッファリングを停止しないとデッドロックになってしまうということです。wargame.rbは手を抜いて(いつも手を抜いてばかりですね)、タイムアウト処理などを行っていませんから、デッドロックが発生するとゲーム全体が停止してしまい、悲しいことになります。

player.rbでは先頭の、

```
STDOUT.sync = true
```

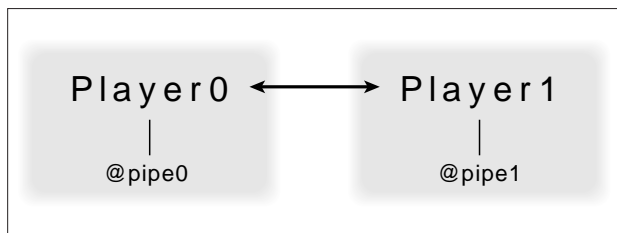


図3 オブジェクト構成

リスト1-3 wargame.rb(その3)

```

def attack(dir)      # 攻撃コマンド処理
  printf "player %s (%d,%d) attacked to %s\n", @name,
    @x, @y, dir

  ox = other.x
  oy = other.y
  hit = falsez

  case dir
  when "n" # north
    if ox == @x and oy < @y
      hit = true
      other.hit(@x, @y)
    end
  when "ne" # north east
    if @x-ox == @y-oy and ox > @x and oy < @y
      hit = true
      other.hit(@x, @y)
    end
  when "e" # east
    if oy == @y and ox > @x
      hit = true
      other.hit(@x, @y)
    end
  when "se" # south east
    if @x-ox == @y-oy and ox > @x and oy > @y
      hit = true
      other.hit(@x, @y)
    end
  when "s" # south
    if ox == @x and oy > @y
      hit = true
      other.hit(@x, @y)
    end
  when "sw" # south west
    if @x-ox == @y-oy and ox < @x and oy > @y
      hit = true
      other.hit(@x, @y)
    end
  when "w" # west
    if oy == @y and ox < @x
      hit = true
      other.hit(@x, @y)
    end
  when "nw" # north west
    if @x-ox == @y-oy and ox < @x and oy < @y
      hit = true
      other.hit(@x, @y)
    end
  end

  if hit
    @pipe.printf "hit %f\n", @life
  else
    @pipe.printf "not %f\n", @life
  end
end

def search           # 索敵コマンド処理
  @pipe.printf "%d %d %f\n", other.x, other.y, @life
end

```

で、標準出力のバッファリングを停止しています。
実際のプレイヤープログラムでは、以下の手順でプレイ

リスト 1-4 wargame.rb (その4)

```
def turn          # 各ターンごとの処理
  line = @pipe.gets
  printf "turn %d: ", TURN[0]
  TURN[0] += 1
  case line
  when /^move ([news]{1,2})/
    self.move($1)
  when /^attack ([news]{1,2})/
    self.attack($1)
  when /^search/
    self.search
  else
    @pipe.print "cmd error\n"
  end
end

def start(other)  # 子プロセスのスタート
  @other = other
  loop do
    @x = rand(X_MAX+1)
    @y = rand(Y_MAX+1)
    break if @x != other.x and @y != other.y
  end
  @pipe.printf "%d %d %d %d %f\n", X_MAX, Y_MAX, @x,
    @y, @life
end

def over          # ゲームオーバー
  printf "player %s died at (%d,%d)\n", @name, @x, @y
  printf "winner player %s at (%d,%d) life %f\n",
    other.name, other.x, other.y, other.life
  exit
end

def damage(level) # ダメージを受けた
  @life -= level
  printf "player %s was hit at (%d,%d) damage %f
    life %f\n", @name, @x, @y, level, @life
  if @life <= 0
    over
  end
end

def hit(x, y)     # やられた~
  dx = @x-x
  dy = @y-y
  lost = 1.0/(Integer(Math.sqrt(dx*dx+dy*dy))+1)
  self.damage(lost)
end

end
```

ヤーを実装します。

1. Player クラスのサブクラスを作る
2. play メソッドを再定義する
3. new メソッドでサブクラスのオブジェクトを生成し、そのオブジェクトのstartメソッドを呼び

メインの処理のほとんどをスーパークラスで実現し、ごく一部をサブクラスで用意するクラス構成を「テンプレートパターン」と呼びます。このような典型的なクラス構成あるいは設計方針は、「デザインパターン」と呼ばれます。

テンプレートパターン以外にもいろいろ便利なパターンがあります。デザインパターンについては、『デザインパターン 改訂版』（ソフトバンク ISBN4-7973-1112-6）に詳しく解説されています。

プレイヤー

プレイヤーの戦略を考えるときに以下の点が影響してきます。

・相手の位置はわからない

索敵すれば位置を知ることができるし、攻撃がhitすれば方向だけはわかるが、実際に攻撃を行う前に敵のターンがあるので移動してしまうかもしれない。

・攻撃が制約される

東西南北8方向しか攻撃できないので、たとえ位置がわかってもすぐには攻撃できない。攻撃できる位置まで移動している間に敵は移動するかもしれない。近いほど相手の

リスト 1-5 wargame.rb (その5)

```
# ゲームのメインループ
def game_loop(cmd1, cmd2)
  player1 = Player.new(cmd1)
  player2 = Player.new(cmd2)
  player1.start(player2)
  player2.start(player1)
  loop do
    player1.turn
    player2.turn
  end
end

srand          # 乱数の初期化
game_loop(ARGV[0], ARGV[1])
```

ダメージが大きい、近づくのはそれなりにたいへん。すぐ隣にいれば、相打ち覚悟で体当たりという攻撃もある。

うーむ、結構微妙なルールです。制約がきつすぎたかもしれませんね。いろいろ考えたのですが、今回は索敵は使わないで、乱数に頼ることにしました。

作ってみたプレイヤーは以下の2種類です。

・ランダム型

ランダムに移動し、ランダムに攻撃します。ただ、壁にだけはぶつからないように移動します。ファイル名は random.rb です。

・スキャン型

上下に往復運動をしながら、横方向に攻撃を加えます。ファイル名は scan.rb です。プログラム例として scan.rb を載せておきます (リスト3)。

実際のプレイ

では、実際に対戦させてみましょう。それぞれのプログラムに実行属性を付けてください。また、プログラムのあるディレクトリには、PATHを通してください。とりあえず、今回はファイルの位置を指定して実行します。

```
$ ./wargame.rb ./random.rb ./scan.rb
```

とすると、それぞれのコマンドをだらだらと表示した後、勝敗を表示します。どうやら少々 scan.rb のほうが強いようです。しかし、両方とも賢くはないので、やたらターン数がかかります (画面1)。

なお、スキャン型を改造した、まず左端に寄ってから攻

```
turn 0: player random.rb:0 (3,5) moved to (3,4)
turn 1: player scan.rb:1 (8,4) moved to (8,3)
turn 2: player random.rb:0 (3,4) attacked to w
turn 3: player scan.rb:1 (8,3) attacked to w
.....中略.....
turn 398: player random.rb:0 (9,4) attacked to se
turn 399: player scan.rb:1 (8,4) attacked to w
turn 400: player random.rb:0 (9,4) moved to (8,5)
turn 401: players crashed each other at (8,5)
player random.rb:0 was hit at (8,5) damage 1.000000
life -0.410714
player random.rb:0 died at (8,5)
winner player scan.rb:1 at (8,4) life 1.750000
```

画面1 wargame.rbによる対戦

リスト2 player.rb

```
# wargame player library

STDOUT.sync = true
class Player
  def initialize      # 親から貰った値で初期化
    data = STDIN.gets.split
    @xmax = data[0].to_i
    @ymax = data[1].to_i
    @x = data[2].to_i
    @y = data[3].to_i
    @life = data[4].to_f
  end

  def move(dir)      # 移動処理。[x, y]を返す。
    STDOUT.printf "move %s\n", dir
    data = STDIN.gets.split
    @x = data[0].to_i
    @y = data[1].to_i
    @life = data[2].to_f
    [@x, @y]
  end

  def attack(dir)   # 攻撃処理。当たればtrueを返す。
    STDOUT.printf "attack %s\n", dir
    data = STDIN.gets.split
    @life = data[1].to_f
    if data[0] == "hit"
      true
    else
      false
    end
  end

  def search        # 索敵処理。敵の座標を返す。
    STDOUT.printf "search\n", dir
    data = STDIN.gets.split
    @life = data[2].to_f
    [data[0].to_i, data[1].to_i]
  end

  def play          # 実際の戦略を実現する
    # サブクラスで再定義する必要がある。
    raise RuntimeError, "should be implemented in subclass"
  end

  def start        # ゲームの開始
    begin
      play          # 戦略を実行(テンプレートパターン)
    rescue
      exit
    end
  end
end
srand
```


撃する scan2.rb がほかに対して圧倒的に有利でした。

今回のゲームはターン制で処理時間の制約がないので、きちんと考えるプレイヤーだとまた違った結果になるかもしれません。ただ、得られる情報が少なすぎるので、考えても一緒かもしれません。

とはいえ、wargame.rb ではプレイヤーの現在位置もよくわかりません。せめて、現在位置だけは視覚的に表示するようにしましょう。その辺を改造した wargame2.rb を用意しました。プレイヤーの移動がはっきりわかって、戦略が立てやすくなるかもしれません（画面2）。

プログラムについて

今回、開発した以下の「ウォーゲーム」プログラムを本誌付録CD-ROMに収録してあります。

wargame.rb	ゲーム盤プログラム
wargame2.rb	ゲーム盤プログラム(CUI版)
player.rb	プレイヤーライブラリ
random.rb	ランダム型プレイヤー
scan.rb	スキャン型プレイヤー
scan2.rb	スキャン型プレイヤー改

さて、ウォーゲームは一応の完成を見ました。とはいえ、まだまだ改善の余地はありそうです。誌面にも、時間にも限りがありますので、とりあえず宿題としましょう。

- もっと良い表示
- ターン制でなく、スレッドによるリアルタイム制
- もっと賢いプレイヤー

良いプレイヤーができれば教えてくださいね。

リスト3 scan.rb

```
#!/usr/bin/env ruby

require 'player'

class Scan<Player

  def play
    cur, save = "n", "s"
    loop do
      loop do
        break if cur == "n" and @y == 0
        break if cur == "s" and @y == @ymax
        move(cur)
        attack(rand(@xmax)>@x ? "e" : "w")
      end
      cur, save = save, cur
    end
  end
end

Scan.new.start
```

```
.....
.....
.....
.....
.....
.....
.....
1.....0...
.....
random.rb:0 (6,8) 0.000000 / scan2.rb:1 (0,8) 4.371032
turn 803: random.rb:0 (6,9) attacked to n
turn 807: random.rb:0 (6,8) attacked to e
turn 808: random.rb:0 was hit at (6,8) damage 0.166667
turn 808: random.rb:0 died at (6,8)
turn 808: **** winner scan2.rb:1 **** life 4.371032
-----
-
turn 792: scan2.rb:1 (0,4) attacked to e
turn 796: scan2.rb:1 (0,5) attacked to e
turn 800: scan2.rb:1 (0,6) attacked to e
turn 804: scan2.rb:1 (0,7) attacked to e
turn 808: scan2.rb:1 (0,8) attacked to e
```

画面2 wargame2.rb の表示

Column

今月のRuby 1.5

まだ、紹介していない1.5の新機能と改善点はいくつかあります。

- initialize の仕様の明確化

- 文字列結合
- %w() の実装でエスケープできるようになった
- 文字列操作のパフォーマンス向上
- slice、slice!
- スレッド対応デバッグ
- セキュリティ機能のデバッグ

- そのほかたくさんのバグとり

これらについても今後きちんと紹介しようと思います。予定どおりならば、今ごろ1.6がリリースされているはずですから、来月からは1.6の情報をお伝えできると思います。

Webサーバ構築術(第12回)

全文検索サービスはちょっと大きめのWebサイトなら必須の機能だ。入力したキーワードに対して、自サイトのコンテンツの中から該当するページを一覧表示することで、ユーザーの便宜を図ることができる。

Namazuを使った全文検索サービスの提供

文：中島昌彦

Text：Masahiko Nakajima

全文検索は閲覧者へのサービスだ。閲覧者が、確かこのサイトに目的の情報があつたはずというあいまいな状態でアクセスしてきたときに、全文検索エンジンの機能を提供していれば、それを使って自サイトの中を探し、目的の情報を閲覧できるわけだ。

マイWebサイトならば、自分でHTMLファイルを作ってアップロードしているはずなので、たいていはどのページにどういう記載をしているかわかるだろう。しかし、家族全員が公開コンテンツを作っているようなマイファミリーWebサイトともなると、コンテンツを作っている側でも、どこになんの情報があるのかわからなくなる。マイファミリーWebサイトでもこの状態のだから、会社や学校のWebサイトとなると、何がどこにあるのかとんと思ひ出せないのも当たり前だ。

数ページ程度のWebサイトなら全文検索エンジンは不要だが、数十ページのコンテンツがあるならば、全文検索エンジンは閲覧者に対しての必須サービスだと割り切りたい。それはたとえ

マイWebサイトであつてもだ。

grepを使った お手軽検索システム

全文検索システムといっても、セキュリティ無視の使い勝手無視、さらにサーバの負荷かけ放題ならば、力技でできないわけではない。数ページのマイWebサイトにちょっとした検索システムを加えるなら、search.html(リスト1)のHTMLファイルとsearch.cgi(リスト2)のスキプトの組み合わせでなんとかなる。

必要なものとなると、jcode.plの最新版(ここではjcode.pl-2.11を利用)ぐらいだろうか。search.cgiとjcode.pl-2.11をcgi-binディレクトリに入れて、search.htmlをドキュメントディレクトリのしかるべきところに置く。あとはリスト2の6-9行めの各コマンドパスを正しく設定すれば、動作する。targetdirには、ドキュメントディレクトリを指定し、patternには検索対象となるファイルパターンを指定する。作業としてはこれだけだ。

Linuxに限らず、UNIX系システムな

リスト1 search.html

```
<HTML><HEAD>
<META CONTENT="text/html;CHARSET=EUC-JP" HTTP-EQUIV=Content-Type>
<TITLE>簡易全文サーチ</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<FORM ACTION="/cgi-bin/search.cgi" method=post>
<B>簡易全文サーチ</B>
<TABLE border=1 cellPadding=5>
  <TR><TD>検索文字 : <INPUT TYPE="TEXT" NAME="search" SIZE="30"></TD>
  <TD><INPUT NAME="SUBMIT" TYPE="SUBMIT" VALUE="検索"></TD></TR>
</TABLE>
</FORM>
</BODY></HTML>
```

らばデフォルトで組み込まれている find や grep といった基本的なコマンドの組み合わせで、再帰的なディレクトリサーチと、特定文字列とのマッチングをしている。

スクリプトを眺めればわかると思うが、検索文字列として指定したものは EUC に変換しているため、検索対象の HTML ファイルは、すべて EUC で記述されているという前提で作っている。

リスト2 search.cgi

```
#!/usr/bin/perl

require "jcode.pl-2.11";

$buffer='';
$find='/usr/bin/find';
$targetdir='/home/httpd/html';
$pattern='*.html';
$grep='/bin/grep';

read(STDIN,$buffer,$ENV{CONTENT_LENGTH});

foreach $pair (split(/&/, $buffer)) {
    ($name, $value) = split(/=/, $pair);
    $value =~ s/\+//g;
    $value =~ s/%{[a-fA-F0-9][a-fA-F0-9]}/pack("C", hex($1))/eg;
    &jcode'convert(*value, 'euc');
    $QUERY{$name}=$value;
}

print "Content-type: text/html\n\n";
print "<HTML>"
<HEAD>
<META CONTENT=\ "text/html;CHARSET=EUC-JP\ " HTTP-EQUIV=Content-Type>
<TITLE>簡易検索システム-$QUERY{search}-検索結果</TITLE>
</HEAD>
<BODY>
<B>検索単語:$QUERY{search}</B><BR>\n
<BR>\n\n
";

open (SEARCH,"$find $targetdir/ -name \"\$pattern\" -exec $grep -l
\"$QUERY{search}\" { } \\\; |");
@getfind=<SEARCH>;
$count=@getfind;

print "合計 $count の対象ファイルが見つかりました。<BR>\n<BR>\n";

$i=0;
foreach (@getfind) {
    $i++;
    s/^\$targetdir//;
    print "$i : <A HREF=\"\$_\">$_</A><BR>\n";
}

print "</BODY>"
</HTML>
";
```

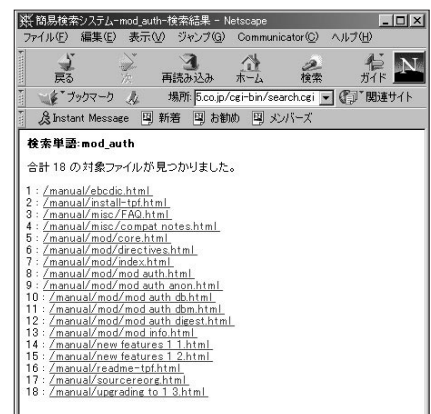
基本的なコマンドと、組み込みHTML タグを含めて50行にも満たないスクリプトで検索させようとする、このあたりが限度だろう。

この簡易検索システムの実際は、画面1、2のようなものだ。探し出すという面では実に目的に合っているが、goo や infoseek といった検索エンジンサイトの検索結果と比べると格段に見劣りする。たとえばマイWebサイトや、数十ページのマイコーポレーションサイトであっても、検索結果にスコア付けをしたり、タイトルやサマリを表示したりしたい。また、完全一致のgrepを使っている限りでは、フレーズ検索はできない。やはり、手間ひまかけずに検索システムを作ろうとしてもできるものではないようだ。

そこで、よく使われるフリー系の検索システムのNamazuを使うことになる。



画面1 search.cgiを使った検索
入力フィールドに文字を入力するだけ。



画面2 search.cgiの検索結果
お手軽な全文検索。質素だが一応動作。

頻繁な更新がなく、ある程度の量があればNamazu向き

Namazuは、本格的な日本語全文検索システムだ。あらかじめサイト全体をサーチして、検索インデックスを作っておく。検索というアクションが起きたときには、検索インデックスを探して検索結果を返すため、毎回全ファイルにgrepをかける先の簡易版とは比較にならない速度でレスポンスが返ってくる。

ただし、インデックスを作るという作業があるために、定期的にインデックスを作るという前処理が必要だ。さらに、インデックス作成部分はperlで記述されており、日本語解析をしながらインデックス付けをする。インデックスの作成には時間がかかるわけだ。1秒に1回とかいう高い頻度で更新される掲示板サイトに利用しようとするならば、Namazuは絶対に向かない。しかし、数時間間隔から1日に一度ぐらいしか書き替えないWebサイトならば、十分に利用できるシステムだ。メールマガジンの主催者がバックナンバー検索に使うといった用途にはまさにベストマッチする。

しかも、Namazuを使って得られる検索結果は、スコア表示やタイトル表示が含まれ、本格的な検索サイトに匹敵する。サイトを問わず、けっこうお勧めのソフトウェアだ。

Namazuを組み込むには関連モジュールが必要

Namazu本体はインデックス付けとインデックスからサーチする機能の両方を提供する。

インデックス作成スクリプトはPerlで記述されているので、インデックス作成部分を動かすためには、Perlと各

種モジュールが必要になる。Perlはたいていのシステムに組み込まれているだろうが、漢字コードフィルタのnkf、日本語わかち書き解析のKAKASI、もしくはChasen、ファイル種別を判断するPerlスクリプトFile::MMagicといったものが必要になる。さまざまなものを組み込んでいかなければいけないため、けっこう手間がかかることは事実だ。また、組み込んだからといって、設定なしで検索システムが動き出すわけでもない。Namazuを動かすまでの道のりはmakeして終わるわけではないので、今回は、パッケージを使ってインストールからセッティングまでを追っていく。

Namazuほか、必要なソフトウェアを入手

最初にやらなければいけないことは、Namazuをはじめとした必要なソフトウェアを入手することだ。

Namazuは、<http://www.namazu.org/>から手に入れられる。RPMならば、<http://www.namazu.org/redhat/i386/>にアクセスし、以下のファイルを手してくるといい。

```
namazu-2.0.4-1.i386.rpm
kakasi-2.3.2-1.i386.rpm
kakasi-dict-2.3.2-1.i386.rpm
perl-File-MMagic-1.06-1.i386.rpm
perl-NKF-1.71-1.i386.rpm
perl-Text-Kakasi-1.04-1.i386.rpm
```

KAKASIのかわりに、Chasenを使う選択もある。Namazuの作者によれば、KAKASIは簡単でレスポンスがよく、Chasenは品質がいいが少し遅いということだ。検索文字列の分かち書き分離品質を高めるならばKAKASIではなくChasenを選ぶとよさそうだ。

必要なファイルさえ入手できれば、rpm -i <パッケージ名> という手順でインストールしていただく。

この一連の作業で、Namazuのバイナリは、/usr/binに、NamazuやKAKASIのドキュメント類は、/usr/share/以下に組み込まれる。

Namazuを使ってインデックスを作る

Namazuを使うには、インデックス作成が必要だ。あらかじめ検索対象のディレクトリに対してインデックスを作り、キーワードごとの分類をすることにより、高速の全文検索が実現する。

たとえば、/home/httpd/html以下のドキュメントディレクトリを対象に、インデックスを作る場合、次のようなコマンドを動かす。

```
# /usr/bin/mknmz -O /tmp/indexes --allow=".*%.html" /home/httpd/html
```

-Oで、インデックスを書き出すディレクトリを指定する。このケースでは、/tmp/indexesにインデックスファイルをそっくり書き出す設定だ。そして、--allowを使い、インデックス対象のファイルパターンを".*%.html"に固定している。なお、--allowで指定するパターンは正規表現だ。ここではhtmlファイルだけが該当し、htm ファイルやshtmlという拡張子のファイルはすべて除外される。

mknmzの最後で指定している部分は、インデックス対象のディレクトリだ。mknmzは、このディレクトリ以下を再帰的にサーチして、--allowに合致するファイルパターンのものを、すべてインデックス化する。

しばらく待てば、インデックス付けが終了する。ちゃんとインデックスが

作られたかどうか、確認してみるといい。今回は、-Oで/tmp/indexesディレクトリにファイルを書き出しているので、

```
# ls /tmp/indexes/
```

と指定すれば、どんなファイルが作られているかわかるだろう。なお、-Oの指定をしなかったときには、カレントディレクトリにインデックスファイルを書き出す。-Oを付けずにいろいろなディレクトリでmknmzを起動すると、大変な目にあう。気をつけよう。

もうひとつ、日本語文書を扱う場合、Namazuに日本語環境であることを指示する必要がある。日本語版のLinuxを使っているなら不要だが、Webサーバ用だからと、英語版システムのままで動かしているような場合、Namazuが正しくindexを作れていないときがある。mknmzを動かしたとき、終了時に英語のメッセージになっている時は、日本語インデックスが作られていないので、日本語での動作をしない。bashをシェルに使っているならば、

```
# export LANG=ja_JP.ujis
```

として、環境変数LANGにja_JP.ujisを指定しておく。これさえ設定しておけば、Perlのwarningエラーが出るものの、Namazuのインデックス生成は無事終了する。

Apacheからの利用は namazu.cgi

インデックスができあがったので、このあとは検索のためのページを作成することになる。

シェルからNamazuを使って全文サーチをする場合、namazuコマンドを

使う。しかし、namazuコマンドはWebサーバから渡されてくる特定の 변수をチェックしているため、CGIから直接呼び出せない。そこで、namazuと同時に組み込まれたnamazu.cgiを利用することになる。

RPMパッケージでNamazuをインストールした場合、namazu.cgiは/home/http/cgi-bin/にインストールされている。もし、Apacheをソースからインストールしているならば、

```
# mv /home/http/cgi-bin/namazu.cgi
/usr/local/apache/cgi-bin/
```

として、正しいCGIディレクトリに移動しておく必要がある。

さらに、Namazuの環境設定ファイルをnamazu.cgiと同じディレクトリにnamazurcという名前で用意する。ベースのnamazurcは、/etc/namazu/namazurcにある。

```
# cp /etc/namazu/namazurc
/home/http/cgi-bin/.namazurc
```

最後に、.namazurcの設定を変更し、namazu.cgiが正しく動くように設定する。修正するのは次の部分で、namazurcの中の、Index、Template、Lang、Replaceの行だけを最低限修正しておく(リスト3)。

リスト3 .namazurcの修正リスト

```
#Index      /usr/local/var/namazu/index
Index       /tmp/indexes

#Template    /usr/local/var/namazu/index
Template    /tmp/indexes

#Lang       ja
Lang        ja

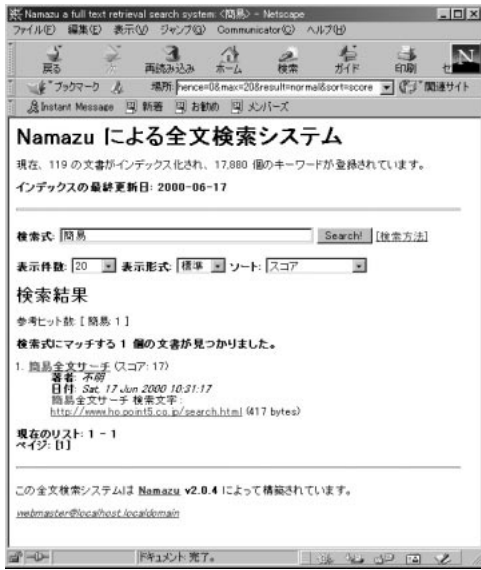
#Replace     /home/foo/public_html/ http://www.foo.bar.jp/~foo/
Replace     /home/httpd/html/ http://www.ho.point5.co.jp/
```

簡単に説明すると、Indexで、インデックス先のディレクトリを指定する。Templateは、HTMLを書き出すときのテンプレートファイルの場所だ。デフォルトでは、インデックスと同じ場所に出力するため、インデックスと同じディレクトリを指定しておくといい。また、Langにjaを指定しておく。

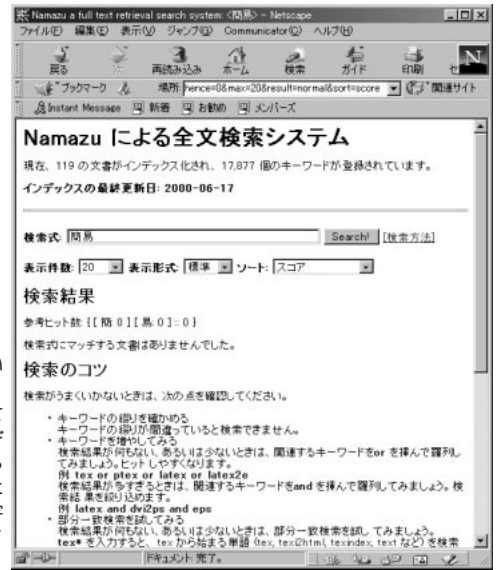
ちょっとややこしい部分がReplaceだ。Namazuのインデックスは、絶対パスとして、ファイルの場所を記録している。Apacheのドキュメントディレクトリの指定とは違うので、HTML出力をするときには、絶対パスに対して、URL表示するためのフィルタリングが必要になる。Replaceで指定しているURLは、各自のホスト名に変更しておく。

これで、いよいよNamazuが動き出す。Webブラウザからnamazu.cgiを呼び出して、実際に検索してみよう。試す手順は2つある。アルファベットの検索がうまくいくか、そして、日本語がうまく検索できるの2つだ。いずれも、自分のWebサイト上に確実にあるような文字列を指定して、検索する(画面3)。

アルファベットはたいいてい正しく検索できるはずだ。しかし、日本語は正しく動かないというようなときがある。たとえば、「簡易」というキーワードで検索した場合、検索結果に、



画面3 Namazuの検索結果
正しく日本語インデックスが
作られていると、ちゃんと日
本語検索ができる。



画面4 インデックスが正しくない
場合
検索結果が漢字ごとに分割され
てしまうようなときは、日本語イン
デックスができていないか、該当する
インデックスが存在しない。絶対に
存在する日本語でサーチしても漢字
単位に分割されるときは、英語イン
デックスしかできていない。

参考ヒット数: { [簡: 0][易:] :: 0 }

と出るようなときだ(画面4)。絶対に存在しているはずの単語でもこのような表示になるのは、英語インデックスしか作っていないときの症状で、日本語が検索対象に含まれていない。こういふときには、インデックスファイルをすべて消し、環境変数LANGを設定しなおしてから、もう一度インデックスを再作成する。

Namazuの出力テンプレートを修正する

Namazuのデフォルトの画面はシンプルで、文字だけで構成されている。人によっては、いろいろと手を加えたい部分があるだろう。そんなときは、自分ならではのNamazu画面を作っていけばいい。

namazu.cgiと同じ場所に置いたnamazurc内にテンプレートの場所を指定している。今回は、/tmp/indexesを指定しているため、/tmp/indexesのテンプレートを修正する。

/tmp/indexesを見るとわかると思うが、実に多くのファイルが作られてい

る。この中で、ターゲットファイルはNMZ.body.jaと、NMZ.footer.ja、NMZ.head.ja、NMZ.result.normal.ja、NMZ.result.short.ja、NMZ.tips.jaとなる。いずれもjaという拡張子で終わっているものと、jaという拡張子のないものがペアで1組ずつ存在している。jaがないファイルは日本語以外の環境下で表示されるテンプレートなので、今回は編集の対象から外しておく。

NMZ.head.jaは、検索結果を出すHTMLのヘッダ部分で、NMZ.footer.jaは、HTMLのフッタ部分となる。検索の結果、マッチする文字列がなかったときには、NMZ.tips.jaがボディとして表示され、検索式に何も入れずに検索したときには、NMZ.body.jaがボディとして表示されるという仕組みになる。

サイトによっては、検索結果も、自分のWebサイトに合ったデザインに変更したいという要望があるはずだ。そんなときには、NMZ.head.jaとNMZ.footer.jaの2つを修正すれば、それらしいものができあがる。

さらに、検索結果の表示形式も変更したいというときがある。namazu.cgiの出力した画面をよく見ると、表示形

式で標準と簡潔が選べるようになってい

る。標準を選んだときには、NMZ.result.normal.jaにそった結果表示となり、簡潔を選んだときには、NMZ.result.short.jaにそった結果表示となる。標準表示でも表示する項目を減らしたいというときには、NMZ.head.jaのファイルを修正することになる。デザインばりばりのパーフェクトなWebサイトを作るなら、このあたりのテンプレートファイル一式に手を入れることが必要だ。

cronを使ったインデックスの再作成

これで全文検索のシステムが動き出す

が、最後にまだ解決しなければいけない作業がある。まず、定期的にインデックスを再作成していかないと、増やしたり修正したページは全文検索の対象外となる点だ。

ニュース系サイトのように、なにかしらの入力ゲートウェイを使って入力する場合、修正や追加をするたびにCGIがmknmzを呼び出せばいい話だ。掲示板でも同じ仕組みが応用できるが、書き込むたびにmknmzを動かしている

と、完全なインデックスは作成できるものの、入力に時間がかかってしかたがない。逆に、入力ゲートウェイがなく、ローカルで作成したファイルをftpでアップロードしているというようなサイトでは、何かしらの形で定期的にインデックス更新をしないと、全文検索エンジンにひっかからないページが多数出てくる。

こんなときに使う仕組みがcronだ。たいいていのサイトなら、1日1回、アクセス数の少ない時間を見はからってインデックスを再作成するといい。通常であれば、朝4~5時あたりの時間帯だろう。ニュース系サイトや掲示板のように、アクティブなサイトでは、1~数時間ごとに動かしていればいい。インデックス再作成をしても、書き込み中でなければ、Namazuの全文検索機能はちゃんと動作しているようだ(リスト4)。

実際に利用すると直面する問題

かなりのサイトでNamazuは使われていると思うが、実際にNamazuを利用した全文検索機能を加えると、直面する問題がある。

Namazuのテンプレート機能は実に

よくできているが、テンプレートでは処理できないケースのページだ。SSIを使っていたり、アクティブにHTMLの一部を書き替えたいというときだ。当然、このような用途にはNamazuは利用できない。namazu.cgiを使わずに、独自のCGIを書き、そこからnamazuコマンドを使うという方法もあるが、namazuコマンドは環境変数をチェックして、CGIとして呼び出されているときにはnamazu.cgiを使うようにというメッセージを出す。

この問題を解決するには、namazuのソースに手を入れてチェック部分を外すか、CGIからnamazuコマンドを呼び出すときに特定の環境変数をクリアすればいい。そこまでしてNamazuを使おうというならば、ソースの中も読めると思うので、チャレンジしてみよう。それができないならば、検索結果の表示ページだけをほかのページと違う作りにするしかない。

もうひとつのよく直面する問題は、検索してはいけないページもNamazuが検索してしまうというケースだ。htaccessの下の認証ページであろうと、Namazuはまちがいにインデックスの対象としてしまうので、公開ディレクトリ以下に内緒のページは置かないことが肝心だ。特に、CGIで集めた個人データを公開ディレクトリ上においておくなどもってのほかだ。公開したくないページは、別のディレクトリを割り振って、ApacheのAlias機能でサイトに結合させるぐらいのことはしないと危険だ。

万一検索してもらいたくないページを公開ディレクトリ上に作らなければいけないようなときには、拡張子をhtmlではなくhtmにする。今回のケースでは、--allow=". * ¥.html"としてmknmzを起動しているため、拡張子がhtmlのものしかインデックス対象とならないので大丈夫だ。

リスト4 crontabとcronで呼び出されるシェルスクリプトの例。6時間単位に動作させている。

```
--crontab--
0 */6 * * * /root/makeindex.sh

--makeindex.sh--
#!/bin/bash
export LANG=ja
/usr/bin/mknmz -O /tmp/indexes --allow=". * ¥.html" /home/httpd/html
```

Column

glibc-2.0以降を用意しよう

かなり余談だが、これまでこの連載のテスト用サーバは、テスト用ネットワークを切り分けるルータマシンでもあった。ルータマシンだけに、ネットワークを落とすわけもいわず、かなり古いOSが動いていたが、今回マシンのOSを総入れ換えした。

Namazuの組み込みには、glibc-2.0以降が必要、すなわち、周辺をそっくり入れ換える

必要があった。

そこまで大げさなことをしたくなかったが、小手先の手段でごまかそうと思っても、あとで苦労するだけだ。そんなことならば、20分もかければ新規にLinuxを組み込める。結局、OSそっくり入れ換えということになった。

古めのOSにNamazuを入れてサービスを継続するというようなときには、手抜きをしないでglibcだけを入れ換えようとしてもうまくいかないことが多い。ちゃんと別のサーバマシンを構築して、マシンをそっくりリプレース

したほうが、かえってかかる手間も時間も少なくてすむ。

私の場合は、たまたまルータマシン兼用だったため、この記事のソース以外は入ってなかった。不幸中の幸いである。

ルータ兼Webサーバという構成はけっこう多いと思うが、この手のマシンでOSの入れ換えとなると、その間まったくネットワークが途切れることになる。ルータとほかのサービスは、やはり別のマシンでサービスすることが肝心だ。

賢く使うUNIX

これであなたもスマートなUNIX使い！

パス表現とカレントディレクトリの変更

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、ファイルやディレクトリのパス表現とカレントディレクトリの変更について取り上げ、bashが備えるカレントディレクトリ関連の機能の紹介や、さらに便利な移動を行うための関数作成を行う。

今月のお題

ディレクトリの一部を指定するだけでカレントディレクトリを変更できる関数を作成する

UNIX系OSでは、WindowsやDOSの「ドライブ名」の概念はなく、ルート(/)を先頭とするディレクトリツリーにすべてのファイルが配置されている。このため、どんなファイルも、ルートからのディレクトリをスラッシュ区切りで並べたリストと、それに続くファイル名で構成される「フルパス」(絶対パス)で一意に指定できる。たとえば、ルートにあるディレクトリhomeのサブディレクトリdaichiにあるhoge.txtというファイルは、フルパスで指定すると「/home/daichi/hoge.txt」となる。

しかし、このように毎回フルパスを入力するのは面倒なので、作業を行うディレクトリ「カレントディレクトリ」からの相対位置でファイル名を指定する「相対パス」も用意されている。先頭がルート(/)で始まらないパスはすべて相対パスだ。たとえば、「daichi/hoge.txt」や「../daichi/hoge.txt」、「hoge.txt」など。最後の例のようにファイル名のみ指定も相対パスに含まれるので、あなたがふだんコマンドラインで指定するファイル名の大半は相対パスのはずだ。

ファイル名を指定する場面で相対パスを活用して楽をするには、カレントディレクトリ(ワーキングディレクトリとも言う)の変更を頻繁に行う必要がある。このため、bashにはカレントディレクトリの変更に関連した機能が



Illustration : Manami Kato

文 : 大池 浩一
Text : Kouichi Ooike

くつも用意されている。

前半では、カレントディレクトリの変更を行うcd、表示を行うpwdといった基本的な組み込みコマンドをはじめ、ホームディレクトリの指定に便利なチルダ展開、ディレクトリ入力の手間を軽減するシェル変数CDPATHやcdable_varsオプション、以前のカレントディレクトリに素早く戻るための「cd -」や組み込みコマンドのpushd、popd、dirsの使い方を説明する。

後半の「今回のお題」では、指定した文字列を含むディレクトリをリストから検索し、カレントディレクトリを変更する関数を作成する。移動先となるディレクトリのリストをあらかじめ作成しておくため、検索自体にはそれほど時間はかからない。また、複数のディレクトリが検索された場合は、それらを一覧表示してユーザーが選択できるような仕組みを作成する。

カレントディレクトリの変更と表示

カレントディレクトリの変更にはcd、表示にはpwdを使用する。どちらもbashの組み込みコマンドだ。pwdの使い方は簡単で、

```
$ pwd
```

とすると、カレントディレクトリのフルパスが標準出力（通常は端末画面）に出力される。なお、同じ内容が環境変数 PWD にも設定されている。

カレントディレクトリを変更するには、cd の引数に移動先のディレクトリを指定する。たとえば、カレントディレクトリが /usr のとき、サブディレクトリ src に移動（カレントディレクトリを変更）するには、

```
$ cd src
```

とする。もちろん、移動先のディレクトリをフルパスで指定して、

```
$ cd /usr/src
```

としても OK だ。

相対パス指定では、カレントディレクトリを表わす「.」、1つ上のディレクトリを表わす「..」という特別なディレクトリが用意されている。このうち、cd コマンドでは「..」をよく使う。たとえば、カレントディレクトリが「/usr/src/hoge」の状態、

```
$ cd ..
```

とすると、カレントディレクトリが1つ上の「/usr/src」に変更される。「../」に続けてディレクトリを記述することもでき、たとえば、

```
$ cd ../hoge
```

とすると、カレントディレクトリが「/usr/src/hoge」から「/usr/src/hoge/hoge」に変更される。

ホームディレクトリに変換されるチルダ展開

ユーザーがログインした直後のカレントディレクトリを特別に「ホームディレクトリ」と呼ぶ。「/home」といったディレクトリに、ユーザー名と同じサブディレクトリを作成してホームディレクトリとすることが多いようだ（「/home/daichi」など）。実際のホームディレクトリは環境変数 HOME に設定されている。

ホームディレクトリにあるファイルやサブディレクトリを簡単に指定するために、bash には「チルダ展開」という仕組みが用意されている。これは、「~」（チルダ）をパスの先頭で指定すると、自分のホームディレクトリのフルパスに変換されるというものだ。

たとえば、自分のホームディレクトリを表示するには、echo を利用して、

```
$ echo ~
```

とすればいい。「/home/daichi」など、ホームディレクトリのフルパスが表示される。

ホームディレクトリのサブディレクトリに移動する場合にもチルダ展開を利用できる。たとえば、

```
$ cd ~/data
```

とすると、現在どのディレクトリにいるかに関係なく、ホームディレクトリのサブディレクトリ data が新たなカレントディレクトリになる。

なお、一般ユーザーはホームディレクトリで作業することが多いため、ホームディレクトリ自体への移動は特に簡単に行えるようになっている。具体的には、cd を引数なしで実行して、

```
$ cd
```

とすればいい。もちろん、「~」を単独で指定して、

```
$ cd ~
```

としてもいい。

このほか、自分以外のユーザーのホームディレクトリは、「~ユーザー名」で変換できる。たとえば、shino のホームディレクトリを参照したいなら、「~shino」とすればいい。また、現在と直前のカレントディレクトリもそれぞれ「~+」と「~-」で変換できる（環境変数 PWD、OLDPWD にも自動的に設定される）。

ディレクトリ入力の手間を軽減する

bash では、ホームディレクトリ以外のディレクトリについても、長いディレクトリの入力の手間を軽減する方法が

いくつか用意されている。

ひとつめは、CDPATH (ディレクトリ検索パス) を利用する方法だ。シェル変数 CDPATH に「:」(コロン) 区切りで複数のディレクトリを設定すると、cd の引数で指定したディレクトリを、それらのディレクトリのサブディレクトリとして順番に検索し、見つかった時点でカレントディレクトリを変更して検索を中止する。たとえば、

```
$ CDPATH=:/usr/doc
```

として、CDPATH に空文字列 (カレントディレクトリを意味する) と /usr/doc を設定する。この状態で、

```
$ cd mgp-1.07a
```

とすると、最初にカレントディレクトリで「mgp-1.07a」というディレクトリを探し、見つければそこをカレントディレクトリとする (通常の cd の動作)。見つからなかった場合、今度は「/usr/doc」で探し、見つかったら「/usr/doc/mgp-1.07a」をカレントディレクトリとする。

ディレクトリの指定は、相対パスであれば「/」を含んでいても構わない。たとえば、

```
$ cd mgp-1.07a/sample
```

とすると、「mgp-1.07a/sample」を検索する。なお、フルパスで指定したり、「./」や「../」で始まる場合は、CDPATH による検索は行われない。

サブディレクトリ名だけの指定でカレントディレクトリを変更できるので、頻繁に訪れるディレクトリにサブディレクトリがいくつも並んでいるような場合は、CDPATH を利用するといいたいだろう。

もうひとつの方法は、cd の引数としてシェル変数を指定する方法だ。たとえば、

```
$ mgp=/usr/doc/mgp-1.07a
```

とあらかじめ設定しておく、

```
$ cd $mgp
```

とするだけで、カレントディレクトリを「/usr/doc/mgp-

1.07a」に変更できる。

bash では、さらに cdable_vars オプションを設定することで、ディレクトリではない cd の引数をシェル変数の名前と見なすようになる。オプションの設定は、組み込みコマンド shopt の -s オプションを使って、

```
$ shopt -s cdable_vars
```

とする (解除は -u オプション)。

これ以後、シェル変数の先頭の「\$」を省略して、

```
$ cd mgp
```

としても、カレントディレクトリが「/usr/doc/mgp-1.07a」に変更される (ただし、カレントディレクトリにサブディレクトリ mgp がある場合はそこに移動する)。シェル変数をディレクトリの別名として使えるわけだ。

ディレクトリを保存して戻る pushd、popd、dirs

CDPATH や cdable_vars を使った方法では、あらかじめシ

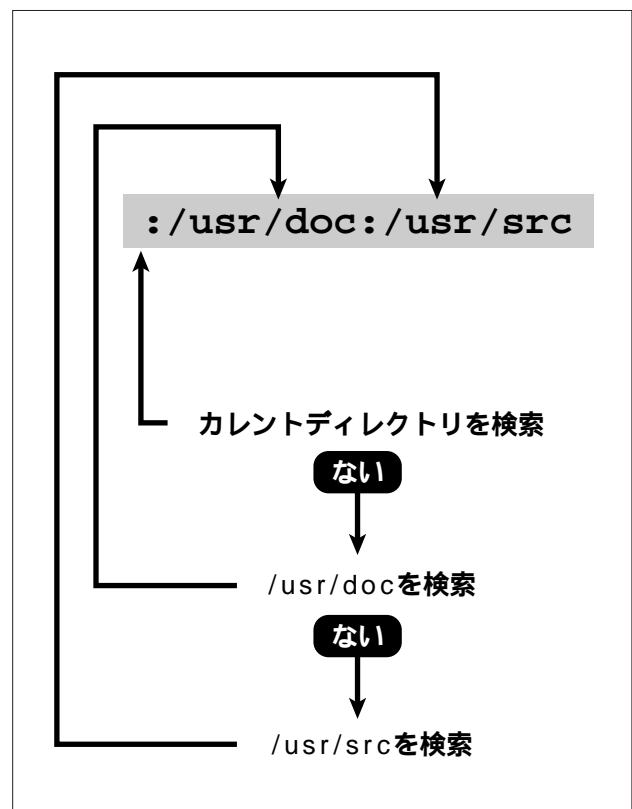


図1 シェル変数 CDPATH の仕組み

エル変数にディレクトリを設定する必要がある。2つのカレントディレクトリを交互に切り替えるだけなら、もっと簡単な方法が用意されている。

bashは直前のカレントディレクトリのフルパスを環境変数 OLDPWD に自動的に設定する。カレントディレクトリを OLDPWD の内容に変更するには、cd の引数として「-」だけを指定すればいい。つまり、一度ディレクトリを移動した後は、「cd -」を繰り返すだけで、2つのディレクトリを交互に切り替えられる。

ただし、「cd -」では、直前のカレントディレクトリだけしか戻れない。そこで、bashには別のディレクトリに一時的に移動する pushd と、後で元のディレクトリに戻る popd という、汎用性のある組み込みコマンドが用意されている。pushd と popd を使えば、3つ以上のディレクトリを記憶させて、いつでも好きなときに元のディレクトリに戻ることができる。

pushd は cd の代わりに使って、引数に一時的な移動先ディレクトリを指定する。元のディレクトリは専用の記憶領域（ディレクトリスタック）に保存されるので、移動先ディレクトリから cd を使ってさらに別のディレクトリに移動しても構わない。保存したディレクトリに戻るには popd を引数なしで使う（図2）。

たとえば、カレントディレクトリが「/home/samba」の状態、

```
$ pushd /usr/doc/mgp-1.07a
```

とすると、ディレクトリスタックに「/home/samba」と「/usr/doc/mgp-1.07a」がこの順番で記録され、「/usr/doc/mgp-1.07a」に移動する。適当に作業した後で、

```
$ popd
```

とすると、ディレクトリスタックの先頭の「/usr/doc/mgp-1.07a」が取り除かれ、先頭の「/home/samba」が新たなカレントディレクトリになる。

なお、ディレクトリスタックの「スタック」とは、「最後に追加した内容が最初に取り出される」データ形式のことだ。このため、pushd した状態でさらに pushd を重ねて実行すると、1回目の popd では最後に pushd したときのカレントディレクトリ、2回目の popd ではその前に pushd したときのカレントディレクトリ...という順で、ディレクトリスタックから取り出される。

ディレクトリスタックの内容は、pushd を実行した際に自動的に表示されるほか、組み込みコマンドの dirs を使えばいつでも確認できる。ディレクトリスタックは横並びで表示されるのが普通だが、-v オプション付きで dirs を実行すれば、ディレクトリを1行ずつ縦に並べて表示できる。なお、横並びの場合は左端、縦並びの場合は上端がそれぞれスタックの先頭になる。

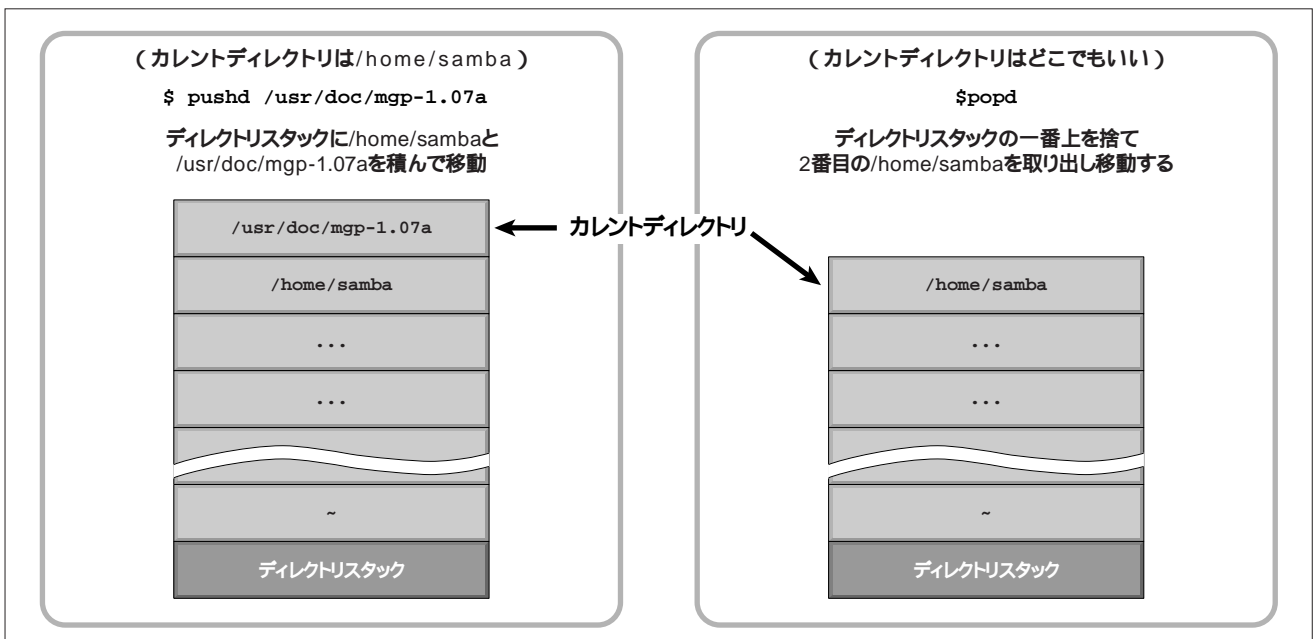


図2 pushd と popd

今月のお題



ディレクトリの一部を指定するだけでカレントディレクトリを変更できる関数を作成する

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、

ディレクトリの一部を指定するだけでカレントディレクトリを変更できる関数を作成する

というもの。あらかじめ移動先の候補となるディレクトリのリストをファイルに作成しておき、そのリストを検索して見つかったディレクトリに移動する。

ところで、今回のお題を「シェルスクリプト」ではなく「関数」として作成するには理由がある。シェルスクリプトをコマンドラインで直接実行すると、サブシェルと呼ばれるシェルのコピーが起動され、スクリプト中の処理はサブシェルで行われる。サブシェルと元のシェルのカレントディレクトリは独立しているのだから、スクリプト中でいくらカレントディレクトリを変更しても、元のシェルのカレントディレクトリには影響しないのだ。一方、関数の場合は元のシェルでそのまま実行されるので、カレントディレクトリの変更が有効になる。

移動先の候補となるディレクトリのリストは、ホームディレクトリの「.ccd」というファイルに保存することにする（「.」で始まる名前はlsでは表示されない）。以下では、ディレクトリリストを作成する関数 `addir` と、リストから検索したディレクトリに移動する関数 `ccd` を、それぞれ作成することにしよう。

ディレクトリリストを作成するコマンドライン

まずは、コマンドラインでリストを作成してみる。あるディレクトリ以下のすべてのディレクトリを出力するには、本連載でおなじみの検索コマンド `find` を判別式「`-type d`」付きで実行すればいい。たとえば、カレントディレクトリ以下のすべてのディレクトリのリストを、ホームディレクトリの `.ccd` に追加するには、

```
$ find . -type d >> ~/.ccd
```

とする（「>」では `.ccd` が上書きされてしまうので、「>>」を使うことに注意されたい）。

ただし、この方法では「`.ccd` に同じディレクトリが複数登録されてしまう」という問題がある。このような重複を解消するには、テキストデータを行単位で並び替える（ソートする）コマンド `sort` を、重複する行を1つにまとめる `-u` オプション付きで使えばいい。

たとえば、`~/.ccd` に保存されたディレクトリの重複を解消するには、

```
$ sort -u -o ~/.ccd ~/.ccd
```

とする。ここで、`-o` は出力ファイルを指定するオプションで、出力ファイルと入力ファイルが同じ（あるいは入力ファイルの一部に含まれる）場合には指定が必須だ。自動的に一時ファイルを利用して、入力ファイルの内容が失われるのを防いでくれる（コラム参照）。

上記の2つのコマンドラインは、パイプ（`|`）を使って接続できる。`find` の出力をパイプで `sort` に入力し、`~/.ccd` の内容と合わせて並び替えを行えばいい。具体的には、次のよ

Column

リダイレクト処理のタイミングに注意

UNIX系OSの初心者が犯しがちなミスのひとつに、「入力ファイルと同じファイルに出力をリダイレクトしてしまう」というものがある。

たとえば、上の `sort` のような場合、ついつい

```
$ sort -u /ccd > /ccd
```

としてしまう。

ところが、入出力リダイレクトの処理はコマンドの実行よりも先に行われる。特に、出力先に指定したファイルはこの時点で初期化されてしまうのだ。このため、`sort` が実行されてファイルの内容を読み込んだ時には、`/ccd` の内容はすでに空になっている。

うになる。

```
$ find . -type d | sort -u -o ~/.ccd - ~/.ccd
```

注意してほしいのは、sortの入力ファイル名として「~/ccd」のほかに「-」が指定されていることだ。「-」は標準入力を意味するファイル名で、これを指定しなかった場合は、パイプ経由で入力されたfindの出力がいったい無視されてしまう。

相対パスをフルパスに変換する

findの出力を利用するにあたって、ひとつ問題がある。findの検索開始ディレクトリに相対パスを指定すると、出力されるディレクトリも相対パスになってしまうのだ。カレントディレクトリの移動に使うことを考えると、相対パスよりもフルパス（絶対パス）のほうが都合がいい。カレントディレクトリを変更する際に、現在どこにいるかを気にする必要がないからだ。

そこで、ディレクトリの相対パスをフルパスに変換する方法を考える。利用する素材は、コマンド置換\${...}とcd、pwdの3つだ。

たとえば、カレントディレクトリが「/home/daichi」の状態で、

```
$ echo $(cd Desktop; pwd)
```

とすると、echoの出力は「/home/daichi/Desktop」とフルパスになる。

動作を詳しく見ていこう。まず、\${...}内部の「;」で区切られた「cd Desktop」と「pwd」が順番に実行される。前者によりカレントディレクトリが変更され、後者により変更後のカレントディレクトリがフルパスで出力される。続いて、コマンド置換により\${...}全体がその結果で置換されてechoの引数になる。なお、コマンド置換の内容はサブシェルで実行されるので、元のシェルのカレントディレクトリには一切影響しない。

実際の関数での処理は以下ようになる。

```
dir=$(if cd "${1:-.}" 2>/dev/null; then pwd; fi)
```

まず、関数の最初の引数（\$1）に設定されたディレクトリにcdで移動する（「\${1:-.}」は、\$1の内容が空の場合に「.」で置きかえるための仕掛け）。このcdはif制御構造の条

件部に置かれているので、指定したディレクトリに移動できなかった（正常終了の）場合はpwdが実行され、移動できなかった場合はなにも実行されない（「2>/dev/null」はcdのエラーメッセージの表示を抑制する）。コマンド置換により、最初の引数で指定したディレクトリに移動できれば、pwdの出力するフルパスがシェル変数dirに設定され、移動できない場合は空文字列が設定される。

ディレクトリリストを作成する関数addir

これでほぼ材料がそろったので、ディレクトリリストにディレクトリを追加する関数addirを作成できる（リスト1）。使い方は簡単で、移動先の候補に追加したいディレクトリを引数で指定すればいい。

たとえば、

```
$ addir /usr/doc/mgp-1.07a
```

とすると、/usr/doc/mgp-1.07aがディレクトリリスト（~/ccd）に追加される。相対パスやチルダ展開を利用してディレクトリを記述することも可能だ。引数を省略するとカレントディレクトリが追加される。

なお、通常は指定したディレクトリだけが追加されるが、-rオプションを指定した場合は、そのディレクトリ以下のすべてのディレクトリが追加される。

たとえば、

```
$ addir -r /usr/src
```

リスト1 関数addirの内容

```
1: function addir {
2:   local depth dir
3:   depth="--maxdepth 0"
4:   if [ "$1" == "-r" ]; then
5:     depth=""
6:     shift
7:   fi
8:   dir=$(if cd "${1:-.}" 2>/dev/null; then pwd; fi)
9:   if [ -z "$dir" ]; then
10:    echo "addir : $1 not found" > /dev/stderr
11:    return 1;
12:   fi
13:   find "$dir" -type d $depth | sort -u -o ~/.ccd -
14: }
```

とすると、`/usr/src`以下のディレクトリがすべてディレクトリリストに追加される。

このほかにも、引数に複数のディレクトリを指定したり、指定したディレクトリをディレクトリリストから削除するなどの処理も行いたいところだが、今回は省略する。腕に自信のある人は挑戦してみしてほしい。

それでは、関数の内容について解説しよう。3~7行目では、`-r`オプションの有無によりシェル変数 `depth` の内容を変更している。`-r` 指定時には「`-maxdepth 0`」、無指定時には空文字列が設定される。

8行目はすでに説明したように、最初の引数に設定されたディレクトリの相対パスをフルパスに変換してシェル変数 `dir` に設定する。続く9~12行では、ディレクトリが存在しない(`dir`が空文字列の)場合のエラー処理を行う。エラーメッセージを標準エラー出力(`/dev/stderr`)に出力し、終了ステータス1で終了する。

最後に、13行目の `find` によりディレクトリが検索され、その出力とディレクトリリスト(`~/ccd`)の内容が `sort` でソートされ、重複行がまとめられる。`dir`の内容はフルパスに変換されているため、`find`の出力も自動的にフルパスになることに注意されたい。なお、`find`のコマンドラインに含まれる `depth` の内容は `-r` オプションの有無により異なる。通常は `depth` の内容が「`-maxdepth 0`」なので、指定したディレクトリだけが出力される。一方、`-r` を指定すると `depth` が空文字列となり、指定したディレクトリ以下のサブディレクトリがすべて出力される。

ディレクトリリストを利用して移動する関数 `ccd`

前節で作成した関数 `addir` を適当なファイル(たとえば `ccdfunc`)に記述し、組み込みコマンド「`.`」を使って、

```
$ . ccdfunc
```

とすると、`addir` が現在のシェルに登録される(後述の関数も同様に登録されたい)。

`addir` によって作成されたディレクトリリスト(`~/ccd`)を検索し、カレントディレクトリを変更する関数 `ccd` のプロトタイプはリスト2ようになる。使い方は簡単で、引数として移動先のディレクトリ名の一部だけを指定すればいい。たとえば、

```
$ ccd mgp
```

とすると、ディレクトリリストに登録されたディレクトリのうち、「`mgp`」を含むディレクトリを検索し、見つかったディレクトリを表示してカレントディレクトリを変更する。ディレクトリ名に正規表現を使用することも可能だ(「`'`」で囲む必要がある)。

なお、引数を指定しなかった場合や、指定したディレクトリが「`.`」「`..`」やそれらで始まる相対パスの場合、さらにはフルパスの場合は、`ccd` はディレクトリリストの検索を行わず、`cd` と同じ動作をする。

それでは関数 `ccd` の内容を説明しよう。まず、3行目では、シェル変数 `dir` に最初の引数(`$1`)の内容を指定する。`$1`の内容が空だった(引数を指定しなかった)場合は、ホームディレクトリ(環境変数 `HOME` の内容)が設定される。

続く4~12行目では、`case` 制御構文を使って、通常の `cd` と同じ動作をさせたい場合と、ディレクトリリストを検索する場合を分ける。5行目は通常の `cd` と同じ動きをさせたいディレクトリのパターンのリストで、`dir` の内容がいずれかのパターンにマッチする場合、何もせずに `case` を抜けて13行目の処理に飛ぶ。

5行目のパターンにマッチしなかった場合、7行目のパターン「`*`」はあらゆる文字列にマッチするので、必ず8~11行目が実行される。まず、8行目の `$(...)` 内部の「`egrep "$dir" ~/ccd`」により、`dir` の内容を含むディレクトリを `~/ccd` から検索する。見つかったディレクトリはコマンド置換により配列 `dirs` に格納される。1つでもディレクトリが見つかった場合は、9行目で `dir` を最初に見つかったディレクトリに設定しなおす。

最後に、13行目で `dir` の内容が「`.`」とともに表示され、

リスト2 関数 `ccd` (プロトタイプ)

```
1: function ccd {
2:   local dir dirs
3:   dir="${1:-$HOME}"
4:   case "$dir" in
5:     |..|../*|../*/*|/*)
6:       ;;
7:     *)
8:       if dirs=$(egrep "$dir" ~/ccd); then
9:         dir="${dirs[0]}"
10:        fi
11:        ;;
12:   esac
13:   echo "--> $dir"
14:   cd "$dir"
15: }
```

14行目でカレントディレクトリが変更される。

複数のディレクトリが検索された場合の処理

関数 `ccd` のプロトタイプでは、複数のディレクトリが見つかった場合、常に最初に見つかったディレクトリに移動してしまう。複数のディレクトリが見つかった場合には、それらを番号付きでメニュー表示し、指定した番号のディレクトリに移動するように関数 `ccd` の内容を変更したほうがよいだろう。

番号付きのメニューを表示したり、入力をうながすプロンプトを表示し、入力された番号が適正範囲にあるか調べ、範囲外の場合は再度メニュー表示に戻る...といった処理は、前回紹介した数値変数や `((...))` 制御構造、組み込みコマンドの `read` などを駆使すれば実現できる。しかし、関数の行数は非常に長いものになるだろう。

その代わりに、今回は `bash` が用意している `select` 制御構造を利用する。この制御構造を使うと、上記のような処理をわずかに数行の記述で実現できるのだ。 `select` 制御構造の構文は以下の通りだ。

```
select 変数名 in リスト; do
```

変数名を利用したステートメント

```
done
```

たとえば、「配列 `dirs` の内容をメニュー表示し、選択した内容をシェル変数 `dir` に格納する」という処理は、次のように書ける。

```
select selection in ${dirs[@]}; do
    if [ -n "$selection" ]; then
        dir="$selection"
        break;
    else
        echo "ccd: invalid number" > /dev/stderr
    fi
done
```

`dirs` の全要素を列挙した「`${dirs[@]}`」がリストとして渡され、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、対応するディレクトリがシェル変数 `selection` に格納され、`do` と `done` に挟まれた部分が実行される。通常は、`selection` の内容を `dir` に設定し、`break` で `select` から抜ける。

一方、番号が範囲外の場合は `selection` の内容が空なので、エラーメッセージを出力して再度メニュー表示に戻る (`Ctrl-C` キーで中断可)。

`select` によるメニュー表示を取り入れた関数 `ccd` の最終版をリスト3に示す。プロトタイプ (リスト2) の9行目が、最終版では9~21行目に相当する。

まず、9行目で配列の要素数を調べ、見つかったディレクトリが1個の場合は10行目でそのまま `dir` に設定する。複数見つかった場合は、13~20行目の `select` 制御構造でユーザーが選択したものを `dir` に設定する。なお、12行目で設定している環境変数 `PS3` は、`select` 制御構造で使われるプロンプトだ。

リスト3 関数 `ccd` (最終版)

```
1: function ccd {
2:   local dir dirs selection
3:   dir="${1:-$HOME}"
4:   case "$dir" in
5:     |..|.|*|..|*|/*)
6:       ;;
7:     *)
8:       if dirs=( $(egrep $dir ~/.ccd) ); then
9:         if (( ${#dirs[@]} == 1 )); then
10:          dir="${dirs[0]}"
11:        else
12:          PS3='which directory? '
13:          select selection in ${dirs[@]}; do
14:            if [ -n "$selection" ]; then
15:              dir="$selection"
16:              break;
17:            else
18:              echo "ccd: invalid number" >
19:                /dev/stderr
20:            fi
21:          done
22:        fi
23:      ;;
24:    esac
25:    echo "--> $dir"
26:    cd "$dir"
27: }
```


IMAP4 環境を使ってみる

Emacs はじめました

第6回 続々・メールを使う

Mew でメールの読み書きができるようになったところで、欲張ってもうひとつ、新しいタイプのメールツール Wanderlust を紹介します。POP3 だけでなく IMAP4 サーバにも対応し、メールとニュースを統合して読めるのが特徴のこのメーラ、人より半歩前を歩きたいあなたにお勧めです。

文：佐々木太良
Text：Taroh Sasaki

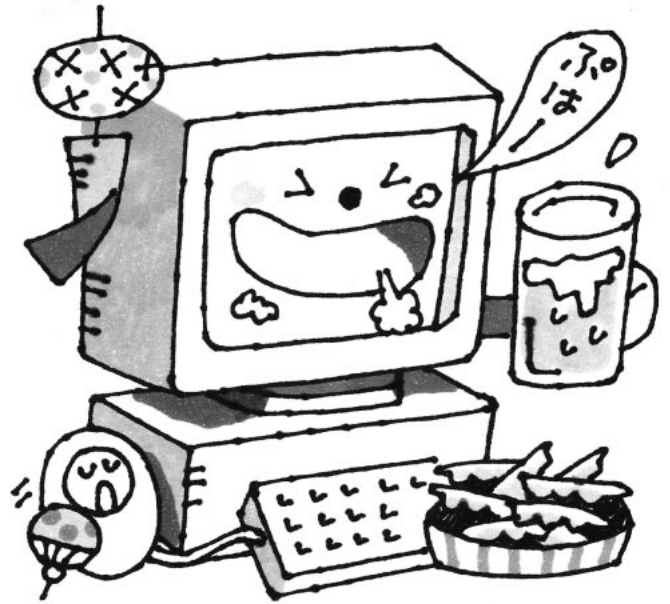


Illustration : Manami Kato

はじめに

皆さま、暑さにめげずに Emacs してますか～？ さてさて、今回取り上げるのは、Mew と同様にメールを読み書きする Wanderlust というアプリケーションです。局面によっては Mew よりも便利で、Mew と共存共栄させることもできます。複雑な家庭環境、いやメール環境なら、どっちも知っていたほうがお得かも。

Wanderlust は、POP3 を使ってサーバにメールを取りに行くことができますが、今回は IMAP4 (rev1) という方式にチャレンジしてみましょう。Wanderlust の特筆すべき特徴はそこにあるからです。POP3 環境の設定については最後の補足を参照してください。

自宅と会社でメールを読む

SMTP サーバ (メールサーバ) とクライアント (メーラ) のあいだでどのようにメールが保存されるかは、これまでにも触れてきました。今回は、1つのメールアカウントにあてたメールをいくつかのマシンで読んでみます。

POP3 だったら.....

POP3 では、まずサーバ (「郵便局」) のシステムメールボックス (「私書箱」) から到着しているメールを全部手元

のマシンのメールボックス (「郵便受け」) に取り込みます。自宅と会社など、いくつかのマシンでメールを取り込む必要がある場合、「サーバにメールを残す」設定にしておけば、メールを私書箱からコピーして「じぶんち」(自宅マシン) の郵便受けに取り込んだあとも、私書箱にはまだメールが残っています。そこで、「別荘」(会社マシン) で取り込んだときも、同じメールを別荘で受け取れるのです。最近の POP3 はなかなか賢くて、新着メールだけを取り込んでくれるので、「じぶんち」と「別荘」でおもいおもいに受信を繰り返したとしても、同じメールを重複して取り込むことがないようにしています。

しかし、こうしてサーバにメッセージをため込んでいくと、やがてプロバイダなどの「メールボックスの容量制限」とやらに引っ掛かって、新しいメールが受信できなくなってしまいます。おまけに、一方のマシンでせっかく既読マークをつけても、もう一方のマシンでは未読のままなので、メールを二重管理しなくてはなりません。

IMAP4 だと.....

これに対して IMAP4 は、メールをサーバ上で整理できることが特徴で、メールの管理を郵便局に任せることができます。郵便局に自分のメール整理箱を持ち込んで、「これはビル・G からの」「これは野犬観察メーリングリストから」という具合に整理ができるのです。頭の部分 (のコピー) だけ配達してもらってざっと読んだら、郵便局に

「捨てといてね」と頼むこともできます。

IMAP4という方式がいまいち普及していないのは、メールサーバの運用者から見て、「読み終わったメールをサーバに残す奴続出」「ディスク容量が不足」「悪」という図式があるからかもしれません。

むろん、「会社で自分のメールサーバを立てているので自由が効く」「(妻が許せば)自宅に専用線を引いてメールサーバを立てたい」などという方もいらっしゃるでしょう。プロバイダの容量制限だって、なくなる日も遠くないと思います(今どき10Mバイトのハードディスクがいくらだと思いませんか?)。環境さえ許せば、IMAP4は絶対お勧め。これだけは言えます。

さてIMAP4では、ちょっとメールを読み返そうというだけでもネットワークに接続しなければいけません。Wanderlustにはキャッシュ機能があるので、ある程度最近に読み出したメールは自動でコピーを取ってオフラインで読むことも可能です。

また、ネットニュースのメッセージと電子メールのメッセージ(またはそのほかのメディアによって配られたメッセージ)を統合して、同じように取り扱えるのも特徴です。郵便局の私書箱に届いた手紙を局内に保存したり、別荘に持ってきたり、新聞に載った(ネットニュースで流れた)記事を郵便局の私書箱で保存したりできます。

Wanderlustの用意

Wanderlustは、Emacs19.34 + Mule2.3、XEmacs21.1、XEmacs20の3つのバージョンで使えます。これらの違いを吸収するAPELというEmacs Lispライブラリを利用しているおかげです。したがって、インストールにあたってAPELが必要なのですが、そのほかMIMEに対応するSEMIやtmというライブラリ、climeやFLIMというメッセージ操作ライブラリなども必要です。

これらのライブラリのバージョンの組み合わせが悪いとうまく動作しないようですが、RPM環境であれば、<http://oscar.math.ryukoku.ac.jp/shaolin/linux/wl/>からパッケージをダウンロード可能で、インストールの方法もここから知ることができます。ほかに、Debianパッケージも<http://www.jp.debian.org/Packages/unstable/mail/wl.html>からダウンロードすることができます。

これらが利用できない場合は、自分でダウンロードしてきたEmacs Lispファイルの数々を、しかるべきディレクトリに置くことになります。動作可能な組み合わせはhttp://www.gohome.org/wl/doc/wl-euc_3.html#SEC3(infoと同じ)にあります。Wanderlust 1.1.1ならAPEL 10.2 + FLIM 1.13.2 + SEMI 1.13.7がよいようです。

Column

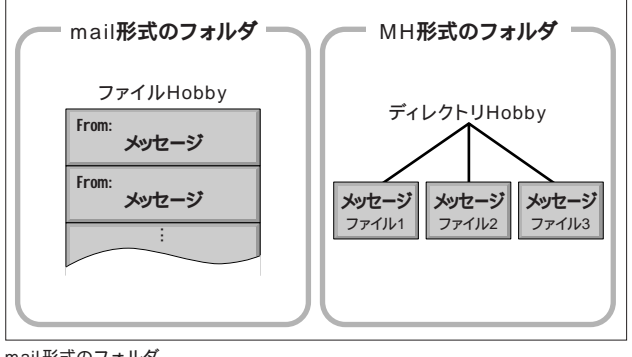
mail形式とMH形式

MUA(メールアプリケーション)がメールメッセージをファイルに保存するときの代表的な形式として、mail形式とMH形式の2つがあります。これ以外の形式もないわけではありませんが、UNIX系OSで発達してきたこの2つは、あとでメッセージを加工するツールも豊富ですし、Windowsのメールアプリケーションにもよく採用されています。

mail形式は、UNIXの伝統的なMTAであるbin/mail、sendmailのシステムメールボックス(%inbox:hoge@mail.server)に採用されてきた形式です。この形式では、フォルダの中のメッセージはつながられて1個のファイルになっています。メッセージの区切りがわからないと困りますから、1行の空行の次に行頭から「From」というパターンがあれば、その行をメッセージの始まりとみなすことになっています。このため、英文でメールを書いている段落先頭で「From the previous discussion,.....」などとやると、勝手にMTAが「>From the.....」と書き換えるので腹が立ちます(笑)。

もうひとつのMH形式は、MHというツールがメールの保存に採用して普及しました。前回に紹介したMewもこの形式を使います。フ

ォルダは同名のディレクトリとして作られ、メッセージはそのなかに1、2、.....と連続番号のファイル名をつけられて別々に保存されます。Perlなどの簡易言語でメッセージを操作するには向いていますが、細かいファイルがたくさんできてOSに負担をかける欠点もあります。Linuxでは、findやfsckの時間が長くなるかもしれません。+inboxに数万通のメールをため込んだ結果、メッセージはうまく扱えるのに、そのディレクトリでlsができなくなることもあります。もっとも、mail形式だったら1ファイル数百Mバイトというファイルができあがるはずで.....どっちもどっちですね。



mail形式のフォルダ

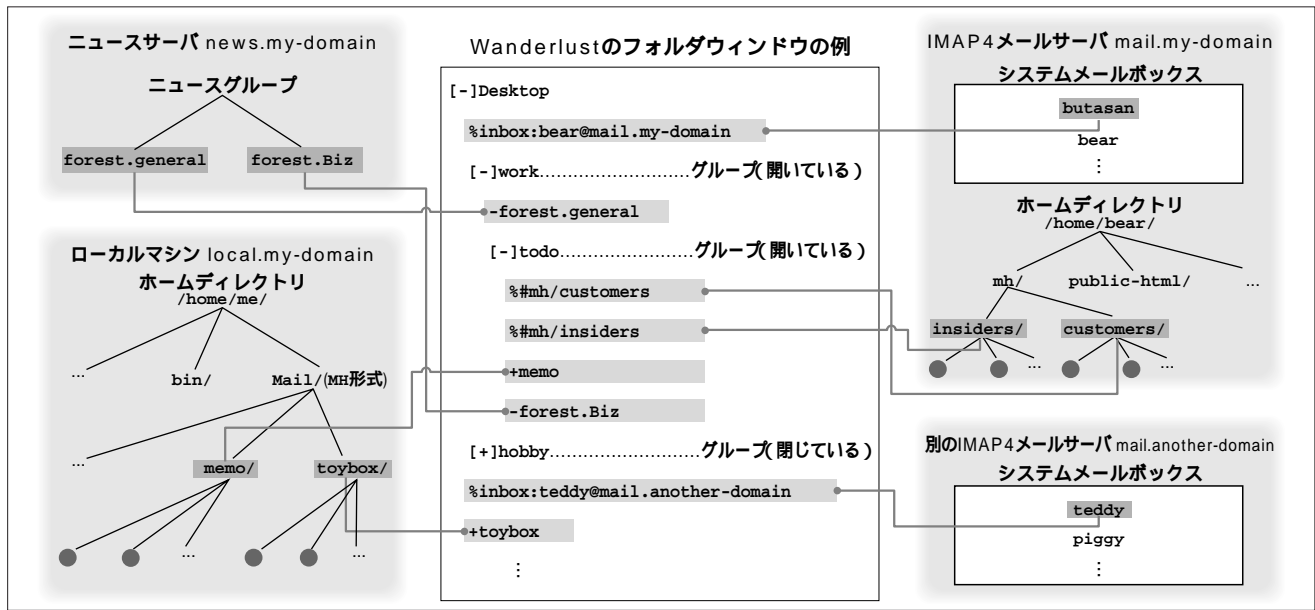


図1 フォルダウィンドウとフォルダのありか

Wanderlustの準備..... の前に心の準備

Wanderlustは.....というかIMAPサーバのシステムメールボックスはmail形式ですが、フォルダはmail形式でもMH形式でもよいことになっています(コラム「mail形式とMH形式」参照)。このため、Wanderlustのフォルダの実体は、mail形式であればメッセージがつながったファイルだし、MH形式なら各メールメッセージのファイルが保存されたディレクトリになります。この点を心に留めて、サーバに到着したメールメッセージがどのように保管・管理されるか、まとめてみましょう(図1参照)。

郵便局と私書箱とじぶんちの状差し

「郵便局」ことメールサーバに届いたメールは、まず「私書箱」ことシステムメールボックス%inbox:hoge@mail.my-domain (mail.my-domainはIMAPサーバ名、hogeはそのマシンでのユーザー名)に入れられます。

次に「郵便局」で整理されたメールは、自分のホームディレクトリにあるフォルダに置かれます。普通はフォルダに“%なんとか”(mail形式のとき)または“%#mh/なんとか”(MH形式のとき)という名前をつけます。これはすなわち、ホームディレクトリにWebページ以外のファイルのアップロードを許さないプロバイダでは、IMAP4が使えないことを意味します。ともあれ、間違ってもWebページを置く場所(IMAP4的には“%public_html/なんとか”)にはメールを整理しないようにしてください。い

や、人に見せたいなら別ですが。

Wanderlustを動かすマシンに取り込んだメールは、“+なんとか”というフォルダに整理することができます。POP3と違って、かならずしも「郵便受け」のような場所に取り込まなくてもよいことに注意してください。郵便配達人がいきなり状差しを整理してくれるようなものです。

さて、以上のフォルダ名はもちろんUNIXのファイル命名規則に従います。ということは、日本語は使わないほうが無難だし、フォルダ名に“/”を使うことはできません。しかし、家訓で%inboxは「受信箱」と呼ばねばならぬという場合には(うそ)、フォルダにWanderlust上で通用するニックネームをつけることができます。

フォルダのありか

システムメールボックスは当然mail形式の%inboxですが、そのほか「郵便局」内で整理されるメッセージは、mail形式ならば%hogeに保存され、それはホームディレクトリの/hogeというファイルに対応します。したがってメールサーバマシンを通常の作業用としても使うなら(パーソナルLinuxマシンとして使う)、ホームディレクトリを整理して、たとえば%mail/hogeというフォルダに保存するようにすれば、/mail/ディレクトリの下にhogeなどのフォルダごとのファイルができるという寸法です。

MH形式の場合、%#mh/hogeというフォルダ名をつけられ、MHディレクトリの下にhoge/ディレクトリが掘られ、そこに1、2、.....とメッセージごとのファイルが出来

あがるのです。MHディレクトリは、フォルダを配置する場所で、メールサーバの /mh_profileに “ Path: mh ” などと書いておけば、 /mh/に設定されます。こうすると、例にあげたメッセージごとのファイルは、 /mh/hoge/1、 /mh/hoge/2、のようになります。Wanderlustでは、ローカルマシン（じぶんちの手紙箱）にはMH形式で保存するのが一般的なようですが、これはあとに述べる /.wlという初期設定ファイルでありかを指定します。

グループとフォルダとメッセージ

Wanderlustには、グループという、メッセージ整理を楽にするための概念があります。これはリモート（郵便局内）とローカル（じぶんち） mail形式とMH形式、メールとニュースといった実体の違いを超えて、いくつかのフォルダを目的別にまとめて取り扱える「フォルダのフォルダ」です。Netscape Navigatorのブックマークの整理とよく似ている、といえわかりやすいでしょう。先のたとえば、「手紙箱」を入れる「ロッカー」ということになります。このロッカーは、引き出しの中にさらに引き出しを作ったり、引き出しを開けるとそこは「郵便局内の手紙箱」だったりという、ドラエモンも真っ青のハイパー4次元構造にできます。

リンク構造の常として、慣れるまではたくさんのグループを作らず、Desktopと呼ばれるトップグループの平面にずらっとフォルダを並べるとよいでしょう。また、あまり

深いグループ内グループを作ってしまうと、目指すグループがどこにあるのかわからなくなります。しかしそこはWanderlust、どんなにグループ階層が深くても、新着メールの見逃しを防ぐ親切機能もあります。

やっと設定だ！

Wanderlustの設定に使うファイルはおもに.emacsと.wlの2つです。Wanderlustを組み込むには、まず /.emacs（またはシステムのsite-start.el）に、

```
(load "mime-setup")
(autoload 'wl "wl" "Wanderlust" t)
(autoload 'wl-draft "wl" "Write draft with Wanderlust." t)
```

の3行を書きます。これ以外の設定はすべて /.wlに書きます。 .wlは.emacsと同様の Emacs Lisp ファイルですが、.emacsになんでも書いてしまうと見通しが悪くなること、個人別の設定を多々書かないといけないことから分かれているようです。パーソナルLinuxマシンではどちらでもよいのですが、学校や会社の共用マシンでは、管理者が共通の設定をsite-start.elファイルに書いておきます。

パッケージからインストールした方は、サンプルファイルがdot.wlなどにあるかもしれませんが。しかし何よりも、C-H iでinfo機能からwl-jaというメニューを探してみてください。起動までに最低限必要な.wlの内容を知ること

リスト1 必須&お勧め.wlの設定

<p>必須項目</p> <p>IMAP4サーバの設定</p> <pre>(setq elmo-default-imap4-server "mail.taroh.org")</pre> <p>SMTPサーバの設定</p> <pre>(setq wl-smtp-posting-server "mail.taroh.org")</pre> <p>-----</p> <p>設定したら便利な項目</p> <p>From の設定</p> <pre>(setq wl-from "SASAKI Taroh <taroh@taroh.org>")</pre> <p>ローカルPC上にメールを取り込んでくるディレクトリ (MH形式)</p> <pre>(setq elmo-localdir-folder-path "~/Mail")</pre> <p>デフォルトのフォルダ名の設定</p> <pre>(setq wl-default-folder "%inbox:taroh@mail.taroh.org") (setq wl-default-spec "%#mh/")</pre>	<p>以下のようにしてローカルマシンに設定することも可能 (標準)</p> <pre>(setq wl-default-folder "+inbox") (setq wl-default-spec "+")</pre> <p>オフライン操作をできるようにする</p> <pre>(setq elmo-imap4-use-cache t) (setq elmo-enable-disconnected-operation t) (setq wl-draft-enable-queuing t) (setq wl-auto-flush-queue t)</pre> <p>未読メッセージを次々と移動して読みたい人向け</p> <pre>(setq wl-auto-select-first t) (setq wl-auto-select-next t) (setq wl-summary-next-no-unread t) (setq wl-summary-move-order 'unread)</pre> <p>派手な画面が好き (サマリとフォルダを同時表示)</p> <pre>(setq wl-stay-folder-window t)</pre>
---	--

ができます。まったく同じドキュメントがhttp://www.gohome.org/wl/doc/wl-euc_toc.htmlにもあるので、うまくinfoをインストールできていないときに試してください。

.emacsの必須設定項目と、設定しておいたら便利と思われるものをリスト1にまとめます。

サーバの設定

今回、IMAP4サーバの設定は必須です。また、送信するメールをリレーしてくれるメールサーバも設定しないとイケません(これまでのPOP3の場合と同じです)。

フォルダの設定

実はドキュメントに書いてあるとおり、この設定はいい加減でも、あとでWanderlustから[新しいフォルダ・グループを作る][フォルダ・グループの並びを変える]という操作ができます。「Emacsはじめました」のお勧めは、まずWanderlustで操作してみて、.foldersファイルでテキストエディタで編集することです。

使ってみよう その1 Mewと同じ(ような)操作

開発者の意図もあって、Wanderlustの操作方法はMewと非常によく似ています。そこで、基本的な操作の説明はごく簡単にすませることにします。

フォルダモード

まずはオンラインで試してみましょう。PPPなどでネッ

トワークに接続した状態で、XEmacsまたはMuleを立ち上げます。そこからM-x wlをタイプすると、オープニング画面が表示されたあと、フォルダの一覧(Folders)に入ります(画面1)。はじめて使うときなら、フォルダは%inbox:hoge@mail.my-domainだけのはずです。

フォルダの名前にカーソルを合わせてSPCキーを押すと、サマリの一覧に入ります。

サマリモード

フォルダ内のメール(メッセージ)のサマリが一覧表示されます(画面2)。読みたいメッセージを選んでSPCキーを押すと、画面が2つに割れて本文が表示されます(画面3)。なんかMewと全然違うわい……。そのままSPCキーで読み進むもよし、C-nとC-pで別のメッセージを選んで読むもよし。

リスト1のようなwlを書いておけば、さらに楽ができます。最初にSPCキーを押してからSPCキーをどんどん押ししていくと、メッセージの終わりでSPCキーを押したときに次の未読メッセージが表示され、フォルダ内の最後のメッセージの終わりでSPCキーを押すと、未読メッセージがある次のフォルダを勝手に開いて最初の未読メッセージを……という具合に、細かいことが苦手な人には超ラッキーな仕組みです。しかしSPCキーだけでは返事は書けません。義理を欠かないように注意してください(筆者のことだうっ)。

新しいメール(返事)を書く

サマリモードでメッセージを選んでおいてAをタイ



画面1 フォルダの一覧
(XEmacsとWanderlust
の組み合わせ)



画面2 サマリの一覧

プすると、相手のメッセージを引用した返信メールが用意されます(画面4)。サマリまたはフォルダモードで w をタイプすると、空のメッセージが用意され、新規に書ける状態になります。メッセージ本文を書き終わったら、C-c C-c で送信できます。

削除

サマリモードでメッセージを選んで d をタイプすると、削除マークがつきます。ある範囲のメッセージを一括削除したいときは、範囲の最初(または最後)で C-SPC を押し、最後(または最初)で r d としてもよいでしょう。

手動リフィル(分類)

サマリモードでメッセージを選んで o をタイプすると、リフィルマークがつきます。勝手に(当社平成11年度実績に基づいて)分類先のフォルダ名を推定してくれるところまで Mew そっくりです。

気をつけないといけないのは、デフォルトで出てくるのは“+”マークですから、オンライン(サーバ側)に分類したい人はこれを消さないといけません。いちいち消すのが面倒なら、リスト1を参考に設定してみましょう。うっかり“+%#mh/hoge”と指定してしまうと、これはローカルコンピュータ上のディレクトリ /Mail/%#mh/ に保存、という意味になってしまうから要注意。

削除/リフィルの実行

サマリモードで x をタイプすると、マークしておい

たメッセージの削除/リフィルが実行されます。

終了

サマリモードで q をタイプすると、フォルダモードに戻ります。フォルダモードから q をタイプすると、Wanderlustが終了します。なお、フォルダから別のフォルダへジャンプしたいときは、サマリモードで g をタイプしたあと、フォルダ名を指定します(これはMewと同じですね)。

通勤電車でメールを読む準備

IMAP4の特徴を活かすには、オンラインでメールの読み書きや整理をするのがもっとも幸せなのですが、たまにはラップトップLinuxマシンにメールを取り込んで電車の中で読みたいこともあるでしょう。このための準備は充電.....じゃなくて(笑)設定です。リスト1を参照してキャッシュを有効にしておきましょう。これは、一度読んだメールはローカルに取り込まれていなくても、しばらくのあいだは選択すれば表示してくれるものです(ローカルのメールボックスに取り込まれているわけではありません)。ほかに、オフライン操作を有効にしたり、オフライン状態で書いたメールを溜めておけるように設定しています。

さて出勤前には、ネットワークに接続してWanderlustを起動し、フォルダ%inbox:.....を開いて l をタイプすれば、新着メールがひととおりキャッシュに取り込まれます(prefetch)。実は、フォルダモードでも同様にできるのですが、フォルダモードではサーバにある全フォルダの

画面3 メッセージの表示



画面4 返信を書く(ドラフトモード)

内容を prefetch しようとするので、サーバに100もフォルダがある人は要注意です。前回 Wanderlust を終了したあと、たとえ %inbox:... 以外の中身が新しくなっていないとも全部検査しちゃうのは、サーバ上のフォルダはほかのメーラから変えられていることがあるからでしょう。

もうひとつ、次に Wanderlust を起動したときにいきなりサーバにメールを取りに行かれては困るので（もしそうなら C-g で中止できます）、M-t でオフラインモードにしておきます。モード行、

```
[--]E_:--%%- [ON] Wanderlust.....
```

の “[ON]” が “[-]” になったら、出勤準備完了。オフラインでもメールを整理したり返事を書きためておくことができます。オフライン中の操作は、次回オンラインモードに変更したときにまとめて送り出される仕掛けになっています。

フォルダとグループの作成 / 整理 / 削除

いくら分類したくても、分類先のフォルダがなければできない.....え、できちゃった？ 実はそうなんです。分類先のフォルダを指定するとき、フォルダ（に対応するディレクトリ）がなければ、確認のあと自動的に作られるのです。

ところで、いざフォルダモードに戻って、こうやって作ったフォルダに飛び込もうとすると、フォルダが見当たりません。メールを整理しようとしても、TAB キーでフ

ォルダ名を補完することもできません。

これは Wanderlust の仕様なのだと思いますが、フォルダに対応するディレクトリがあっても、フォルダモードの一覧には見えないことがあります。こういうときは、あらためてディレクトリに対応するフォルダを作ってあげる必要があります。フォルダモード上で m a とすると、フォルダができます。

同様の機能で、グループは m g（または +）で作成できます。グループやフォルダの名前の上で C-k を押すと、確認もなしに削除（カット）されますが、実際にディレクトリやファイルが消されるわけではないので、間違えても深刻ではなく、C-y でカットされたものがペーストできるので安心です。

カットしたグループやフォルダは、フォルダを移動して C-y を押せば出現するので、メールの整理には便利です。ちなみに何度も削除すると、最後に削除したものが先に出現します。ラストイン・ファーストアウトといいますが、高速なカーブの曲がり方とは関係ありません。

フォルダモードでグループがごちゃごちゃしてきたらさらに深いグループを作ってしまうのもですが、グループ名の [+] の上で SPC キーを押すと、グループのフォルダが開いたり閉じたりして、見た目をすっきりさせることができます。

自動分類してみよう

これは前回の Mew での自動リフィルと同様な機能です。詳しい Emacs Lisp の文法が分からない方でも、リスト 2

Column

オフライン派？ オンライン派？

IMAP4 の現実的な使い方は、

- ・接続時に一気にメールを取り込み、オフラインで読む。整理や削除もオフラインのフォルダのみ（POP3 的な使い方）
- ・オンラインですべて読む。整理や削除もオンラインのフォルダのみ

という2派が主流だと思います。この中間的な使い方もできますが、あまり効率がよいとはいえません。

オフライン派は、%inbox:..... にメッセージを残しておく、システムメールボックスがすぐにいっぱいになってしまいます。プロバ

イダによっては制限をかけているので要注意です。そこで、接続されたらいきなり %inbox:..... に飛び込んで、ローカルマシンのメールボックス行きの整理マークをつまかくって（m a m o +inbox RET）、x で実行、のようにして POP3 的な取り込みができるでしょう。

オンライン派は、ローカルのフォルダで整理分類をすることはあまりないように思えます。「このフォルダはローカル、このフォルダはネット上.....」と決めるのがむずかしいからです。IMAP4 で、ローカルのフォルダからオンラインのフォルダに移動できると思いますか？ これができるんですね。フォルダという概念上、できて当たり前かもしれませんが、POP3 と逆に、メッセージをサーバに「アップロード」していることになります。ということは、メッセージの分類先を間違えてしまうと再分類にムダな時間がかかるので、必然的にサーバ上だけで分類するのが得策です。

を参考にいろいろなパターンを試してみてください。

リスト2では、“From”を使ってkobuta@forest.animalsさんからのメールを%mh/friendsフォルダに、.....のように分類されます。“Subject”の例でわかるように、正規表現も使用可能です（例では先頭から “[forest-admin”、つまりメーリングリストの自動付加サブジェクトを想定しています）。

メーリングリストでは、Subject:で分類する方法もありますが、場合によってはきちんと動かないことがあります。From:もまちまちだし.....というときには、To:で分類するのが正解でしょう。Cc:で分類することもできます。

ほかのヘッダでも分類可能ですが、この4つはIMAPサーバ側で該当するメールにマークをつけて分類することができるので高速です。%inboxのようにサーバ側のフォルダをリフィルするときは、このパターンしか使えません。

こうしておいていざ分類するときには、たとえば%inbox:.....の中（サマリモード）で、C-oを押してみます。するとあら不思議、手動でリフィルしたときと同様にマークがついていきますね.....え？ うまくいかない？ そんなときは、もういちど.wlを開いて、うまくいかないメールのヘッダ（From: とか To: の行）と見比べながら修正してみてください。いったんWanderlustを終了して、再度Wanderlustを起動してトライしてみましょう。自動分類の設定をはじめて書くとき、新しいメーリングリストに加入したあとなどは、ちょっとこの設定と格闘するかも。

リスト2 自動リファイルのルール設定（/.emacsの最後に書いておく）

```
(setq wl-refile-rule-alist
  '(
    ("From"
     ("kobuta@forest.animals" . "%mh/friends")
     ("ponpoko <tanuki@forest.animals>" .
      "%mh/friends")
     ...
    )
    ("To"
     ("info@forest.animals" . "%mh/forest")
     ...
    )
    ("Subject"
     ("^[forest-admin: " . "%mh/forest-admin")
     ...
    )
  )
)
```

Wanderlustを POP3クライアントとして使う

最後にWanderlustを、POP3環境で使用する方法も紹介しましょう。この場合は.wlに、

```
(setq elmo-pop3-use-cache t)
```

の1行を書いておきましょう。あとは、フォルダモードにいるときに!&bear/user@mail.my-domain!+inboxというフォルダを作成すればOKです。bearはメールサーバmail.my-domain上のユーザー名です。ネットワークに接続したら、自動的にPOP3でローカルの+inboxにメールを取り込んでくれます。

さて、次回は.....

いかがでしたか。3回にわたってEmacsにおけるメール環境を構築してきましたが、こうしてみるとEmacsを使っているあいだ、非常に高い割合でアプリケーションを実行させていることがわかります。

次回は、Emacsのカスタマイズに凝ってみることにしましょう。なお、この連載で取り上げてほしいEmacsの使い方がありましたら、ぜひtaroh@taroh.orgまでメールをお寄せください。

それではHappy Hacking!!

Column

Mewの自動整理ふたたび

7月号（前回）のMewの自動整理の記述について、読者の方からご指摘を受けました。「半自動振り分け その1」のように自動推測を繰り返してメッセージを振り分けるのなら、M-oを実行すればできるんですね。詳しくはM-x infoのMew、「楽々整理整頓」の「自動で整理整頓」を見てください。自動で推測してしまう（したがって間違ふこともある）ことを除けば、今回紹介したWanderlustのC-oとまったく同じ機能です。自動推測で間違ったフォルダに整理されるのを防止する方法（mew-refile-guess-controlという変数の中に一括整頓時の推測を抑止するマークをつける）もあります。

さらに「半自動振り分け その2」の、条件に当てはまるメッセージを全部抽出してから整頓する方法では、条件を検索したあとにm o daemon RETとしたほうが速いし簡単なのでした。ぺこん。

Linux 日記

第11回 名前解決(4)

今回からは、DNSサーバBINDを運用する場合に必要な設定ファイルを解説していきます。

文：榊 正憲

Text : Masanori Sakaki

部屋の模様替えと大掃除も一段落し、仕事場の環境がかなり快適なものになった。まだ、仕事場の一角がテナポリスペースのまま、棚に箱を積み上げたままになっているが、この辺もそのうち片付けようと思っている。2月から続いたせわしない暮らしに、今度こそ終止符が打てるはずである。

しかし、懸案事項がまだ残っている。去年の秋口から始めたSETI@Homeの問題だ。現在、我が家では7台(うち1台はノートPC)のコンピュータが24時間運転している。そのうち、本当に24時間稼働でなければならないのは、インターネットサービスに関連する2台だけである。残りのマシンは、すべてSETIのカウントを稼ぐために24時間運転しているのだ。現在、総合で6000位台、上位0.3%といった成績である。さて、その7台の常時運用マシンだが、ファイルサーバとして動いているWindows NT Serverは、24時間運転のほうが便利だが、起きてから起動し、寝る前に止めても問題はない。仕事に

使うクライアントマシンや、別の仕事の絡みで動いているWindows 2000 Serverなどは、まさに使う時だけ動かせばいいマシンだ。要するに、大半はSETIのためだけに24時間動いているのである。

エアコンをどうしよう

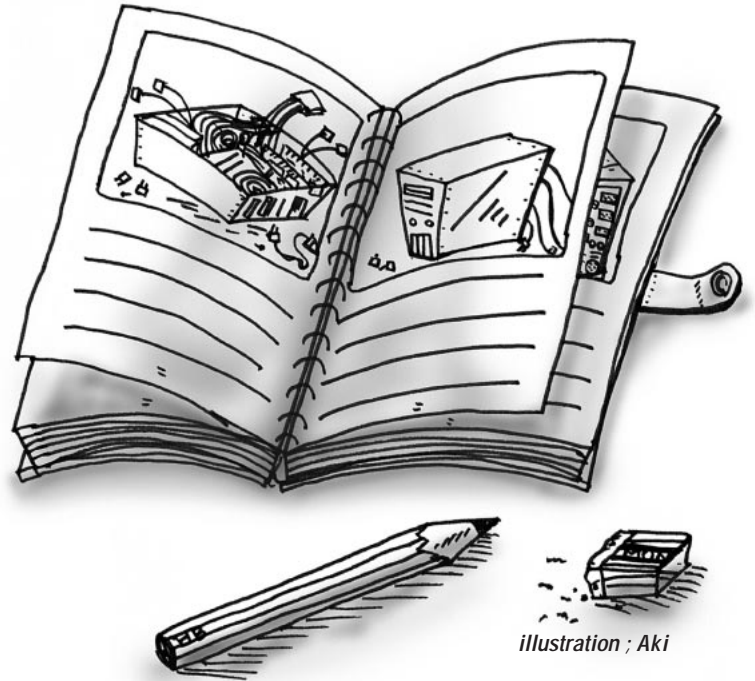
さて、問題は季節である。今までは外気温が低かったので、これだけのマシンを動かしていても問題はなかった。電気代は多少かさんだが、これからの季節(この原稿を書いているのは5月末である)、温度上昇を考えなければならぬ。この台数のマシンを運転していると、5月上旬の段階でも、室温は30度を越えてしまう。窓を開けて外気を通して、28度くらいである。エアコンを24時間運転しなければならぬのは時間の問題だ(窓が開けられるようになったのは、部屋の片付けのおかげだ。以前は窓までたどりつけなかった)。

筆者の仕事部屋は毎年これをやって

いる。例年だと、だいたい3台前後のマシンが24時間運転だったので、6月から10月くらいまでは、エアコンを24時間運転している。もっとも、筆者は冷房が好きではないので、エアコンの運転はあくまで機械のためだ。室温設定は29度程度である。今の機械はこの温度で問題なく動く。もっとも、ファンが故障したらおしまいが。

エアコンを数カ月に渡って24時間運転するというと、電気代が気になるだろう。だが、恐ろしくかかるといってもいい。会社勤めなどをしている、昼間不在の間にムオムオンに暖まった部屋を冷房するとなると、エアコンはフルパワーで運転する。しかし、24時間運転だと、フルパワー運転はまるでない。適当な間隔でローパワーの運転をするだけなので、たぶん読者が思うほどには電気代はかかっていない。

さて、問題は今の稼働マシン台数である。例年の倍以上のマシンが常時動いている。しかも、大型のディスプレイも増えた。この状態で24時間エアコ



ンを回すとなると、さすがに不安になってくる。どうしたものだろう。SETIの手を緩めるか、電気代を気にせずぶんぶん回すか？

省エネタイプの計算器

これからの電気代が気になったからというわけではないが、新たに省エネタイプの計算器を手に入れた。とはいっても、ストアプログラム方式ではない。それどころか、電気すら使っていない（もちろん、量子コンピュータとかでもない）。基本アーキテクチャは10進計算、エネルギー源は横に付いているハンドルである。そう、昔懐かしの（これを本当に懐かしいと思う人は50代以上のはずだ）手回し計算器（計算機ではないと思う）である。

今でこそ機械式計算器は骨董品になってしまったが、中をのぞくと（少なくともLSIチップよりは）面白い。機械式計算器は、基本的に歯車を使って計算を行う。10進1桁が10歯の歯車で構成されており、それを何歯進める、戻すという形で加減算を行う。ミソは、9から0に変わる時に上位桁の歯車を1歯だけ進める、あるいは0から9に戻る時に上位桁を1歯戻すという桁上げが（もちろんメカニカルに）行われることである。このような多桁の歯車による加減算機構が、いわば算術ユニットである。手回し計算器では、この部分をプロダクトレジスタというらしい。これとは別に、人間が数値を入力できる歯車群がある（チェックレジスタ）。ハンドルを1回回すと、プロダクトレジスタの歯車の各桁をチェックレジスタで指定した歯数だけ送る。つまり、プロダクトレジスタの内容にチェックレジスタの内容が加算されるわけだ。もちろん、ハンドルを逆に回せば減算を行える。ハンドルを何回も回せば、その

回数だけ加算が繰り返される。符号処理は、この構造を考えればわかると思うが、真数形式ではなく補数形式となる。-1は9999999.....になる。0をまたいで符号が変わるときには、チーンと音になる。いわば音によるオーバーフロー割り込みである。

この計算器は、シフト処理も行える。チェックレジスタとプロダクトレジスタの歯の噛み合わせを物理的にずらすことができるのだ。これにより、乗算が可能になる。やり方は基本的に筆算と同じだ。ただし、部分積の乗算は九九で行うのではなく、ハンドルを最大9回回し、加算を繰り返して行う。実際には、99を掛けるときには、100を掛けてから1回引くというような最適化を（人間が）行う。たとえば、 123×12 という計算の手順は次のようになる。

- ・プロダクトレジスタをクリアし、チェックレジスタに123をセットする。
- ・ハンドルを2回回す。プロダクトレジスタに123が2回加算され、246になる。
- ・プロダクトレジスタを1桁シフトする。
- ・ハンドルを1回回す。これによりプロダクトレジスタの2桁目から4桁目に123が加算される。つまり、1230が加算されることになり、答えの1476が得られる。

これら2つのレジスタとは別に、マルチプライヤレジスタというカウンタがある。これは基本的に、ハンドルを1回回すごとにインクリメント/デクリメントされる10進カウンタである。マルチプライヤレジスタはプロダクトレジスタと共にシフトし、その位置に応じた桁がインクリメントされる。たとえばさっきの 123×12 の計算であれば、

最初の $123 + 123$ の計算でハンドルを2回回すので、マルチプライヤレジスタの1の位は2になる。次の1230の加算（1桁シフトした123の加算）により、マルチプライヤレジスタの10の位がインクリメントされて1になる。つまり、乗算の終了後、マルチプライヤレジスタは12になっているのである。

乗算の時は、このレジスタを見なくても、ハンドルを何回回したかを数えれば済むのであるが、除算ではこのレジスタが重要になる。

除算は、回復法というアルゴリズムで行う。基本的には筆算と同じだ。被除数をプロダクトレジスタにセットし、除数をチェックレジスタにセットする。そして上位桁から始めて、何回引けるかハンドルを回していくのである（この時は減算を行うので、逆回しになる）。引きすぎてマイナスになったら（チーンと音がするのですぐわかる）、1回逆回しして加算する（これが回復処理である）。そして1桁ずらしてこの処理を繰り返す。マルチプライヤレジスタには、それぞれの桁で何回引いたかが記録されるので、最終的な商がここに現れる。ハンドルの回転方向とインクリメント/デクリメントの関係は、乗算と除算で逆になるのが、マルチプライヤレジスタのそばにそのモードを決めるレバーがある（図1）。

応用計算もいろいろあるようだが、筆者自身はすでに電卓世代なので、詳しくは知らない。今でこそ、よくまあこんなことを機械でやってたもんだなどと思ってしまうのだが、上には上がある。手回し式はまだ直感的である。アルゴリズムに相当する部分は人間が担当しているからだ。機械式計算器の最後の頃には、ちゃんとテンキー入力できる製品もあったのだから驚きだ。メカニカルなテンキーで各レジスタに



置数し、乗算の繰り返し処理を自動的に反復して行っていたのである。このような計算器は、手回しでなく、モーターが動力だった。キーを押すたびにガシャッという音がし、最後に=を押すと、ガッシャングッシャンという音が何秒かして(ナンバーランチャという形容がふさわしいだろう) 答えが出るのである。筆者は学生の頃、ゴミ捨て場でこのような計算器を拾ったのであるが、引っ越しの時に処分してしまった。今思えば惜しいことをしたと思う。

さて、歯車で実現するのはちょっと困難と思われるDNSの話の続きを始めよう。今回は、namedの動作モードについて解説した。今回はいよいよ設定ファイルについてだ。

ネームサーバの設定ファイル

namedの設定は、named.conf (8.Xの場合) というファイルに記述する。namedは起動時にこのファイルを読み込み、自身の動作モードを決定し、そして必要に応じてゾーンデータベースファイルをロードする。まず、named.confについて簡単に説明しよう。

named.confには、ネームサーバの動作モードやオプション、各種ファイル

のパス、セキュリティ設定などを指定する。named.conf自体のパスは、コンパイル時の設定で決まるが、namedに引数として渡すこともできる。

単純なnamed.confの例を見てみよう(リスト1)。このマシン上のnamedは、linux-mag.ascii.co.jp.ゾーンのマスターサーバであり、同時にascii.co.jp.ゾーンのスレーブサーバである。“options”は、namedの動作モードや各種ファイルの場所を示すためのオプションパラメータを列挙するブロックだ。この例では、各種データベースファイルを格納するディレクトリを指定している。ほかに、データベースファイルなどで指定できないnamedの動作モード、セキュリティ設定などを指定することができる。

“zone”は、このnamedが管轄するゾーンを定義する。“zone”に続けてゾーンのドメイン名を指定し、それに続くブロック中で、そのゾーンに関するnamedの動作モード、データベースファイル、ゾーン単位の各種動作オプションを指定する。最初のzoneブロックは、ルートネームサーバの静的な登録のためのものだ。ゾーン名の“.”により、ルートドメイン、つまりルートネームサーバをアクセスするためのゾ

ーンであることがわかる。そして、“file "named.root";”という文で、データベースファイルを指定している。このファイルには、ルートネームサーバの名前とIPアドレスが記述されている。このファイルはnamedのパッケージに含まれている。また、NIC (Network Information Center) などからFTPで最新版を取得できる(URL:ftp://ftp.rs.internic.net/domain/named.root)。“type”は、このゾーンについて、namedがどのような動作モードで機能するかを指定する。キーワード“hint”は、このデータベースファイルの内容を使用するものの、このゾーンについてはマスターでもスレーブでもないということを指定する。まさにヒントであり、実際に行っている処理は、ネームサーバのキャッシュにデータをプリロードすることである。

このnamedは、「linux-mag.ascii.co.jp」ゾーンのマスターサーバである。したがって、“type”は“master”であり、そのデータベースファイルは“linux-mag.db”である。namedはこのファイルの内容を読み込み、外部からのリクエストに対して「権威のある」応答を返す。また、このゾーンにスレーブサーバがあれば、このデータベースの内

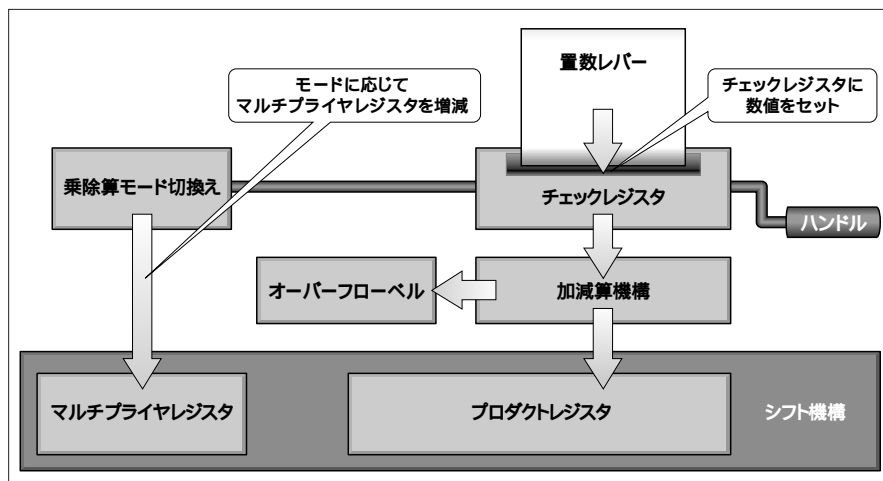


図1 手回し計算器のブロックダイアグラム

リスト1 named.confの例 その1

```
options {
    directory "/etc/namedb";
};
zone "." {
    type hint;
    file "named.root";
};
zone "linux-mag.ascii.co.jp" {
    type master;
    file "linux-mag.db";
};
zone "ascii.co.jp" {
    type slave;
    master 192.168.10.1;
};
```

容をゾーン転送する。そしてこのネームサーバは、「ascii.co.jp」ゾーンのスレーブサーバでもある。スレーブサーバの場合は、“master”というキーワードによって、そのゾーンのマスターサーバを指定する必要がある。これにより、マスターサーバと接続してゾーン転送を行うことができる。スレーブサーバは、外部からの応答に対して、マスターサーバと同じように「権威のある」応答を返すことができる。

このネームサーバは、「ascii.co.jp」ゾーンと「linux-mag.ascii.co.jp」ゾーンについては権威のある応答を返す。それ以外のドメイン名の解決を求めると、ルートネームサーバから始めて順にいくつかのネームサーバを検索し、最終的な結果を返す。つまりキャッシュサーバとして動作する。キャッシュサーバとして運用するための設定というのは特に必要ない（キャッシュサーバとして「運用しない」という設定は、“options”ブロックで指定できる）。

次に、フォワーダ指定を設定したnamedの設定ファイルを見てみよう（リスト2）。フォワーダを指定したネームサーバは、自身では多数のネームサーバに再帰的に問い合わせを送るということはしない。普通の名前解決クライアントと同じように、別のネームサーバ（フォワーダ）に名前解決リクエストを送り、結果を待つだけである。したがって、問い合わせを送るフォワーダのネームサーバIPアドレスを指定するだけでよい。

リスト2 named.confの例 その2

```
options {
    directory "/etc/namedb";
    forwarders {
        192.168.10.1;
        192.168.1.4;
    }
};
```

データベースファイル

ゾーンを管轄するマスターサーバには、そのゾーンのためのデータベースを用意しなければならない。これは必要な数だけレコードを列挙したテキストファイルである。まずは実際のデータベースファイルを見てみよう。先程のnamed.confで設定したnamedは、linux-mag.ascii.co.jpゾーンのマスターサーバである。このゾーンには、表1のようなホストがあるとしよう。これを定義するデータベースファイルはリスト3のようになる。

ゾーンデータベース中のレコードは、基本的に次の形式になる。

<名前> <クラス> <レコード種別>
<パラメータ>...

<名前>は、検索対象となるデータ（ドメイン名）の名前である。<クラス>は、そのデータのクラスで、“IN”はインターネットの意味である。<レコード種別>は、そのレコードのタイプを表す。<パラメータ>は、<レコード種別>に応じたパラメータである。namedは、<名前>を解決するように求められると、<パラメータ>を返すという形で動作する。

<名前>を省略した場合、その直前のレコードと同じ名前が指定されたものと解釈される。リスト3では、SOAレコードの後のNSレコードには名前が指定されていないが、この場合、NSにはSOAと同じ名前が指定されたことに

なる。FQDNで指定した名前はそのまま処理されるが、FQDNでない場合（名前がピリオドで終わっていない場合）は、そのゾーン名（linux-mag.ascii.co.jp）が付加される。単に“www”と記述した場合であれば、“www.linux-mag.ascii.co.jp.”という名前になるわけだ。

さて、どのようなレコードがあるかを見ていこう（DNSでは多数のレコードが定義されているが、インターネット接続で実際に使うレコードはそれほど多くない）。基本的ないくつかのレコードを以下に示す。

• SOA (Start Of Authority)

ゾーン定義の開始を宣言し、そのゾーンデータに関連するパラメータを定義するレコードである。<名前>は、そのゾーンを示すFQDNのドメイン名である。今回の例では、“linux-mag.ascii.co.jp.”である。リスト3では代わりに“@”という表記を使っているが、これはnamed.confで指定されたゾーン名を表現する省略表記である。named.confでは、このデータベースファイルを指定する際に、“linux-mag.ascii.co.jp”としているので、@はそのドメイン名をFQDNとしたもので置き換えられる。

最初のパラメータは、このゾーンのマスターサーバのドメイン名である。次に続くパラメータは、そのゾーンの管理者のメールアドレスである。“postmaster.ascii.co.jp.”となっているが、これは“postmaster@ascii.co.jp”と読み替える。次いで、カッコで囲まれていくつかの数値があるが、このカッコは以降の行が継続行であることを示すものである。

シリアル番号は、このデータベースファイルの履歴を示すもので、修正の

ドメイン名	IPアドレス	説明
ns	192.1.1.1	ネームサーバ
mail	192.1.1.3	メールサーバ
db-svr	192.1.1.4	データベースサーバ
www	192.1.1.5	Webサーバ
ftp	192.1.1.4	FTPサーバ
		(データベースサーバが兼用)

表1 linux-mag.ascii.co.jpゾーンのホスト



たびに値を大きくしなければならない。マスタ/スレーブ間のゾーン転送は、この値に応じて行われる。スレーブサーバの持っているデータよりシリアル番号が大きければ、データが更新されたものと見なして、ゾーン転送が行われるのである。この数値を大きくしないと、データの変更がスレーブサーバに伝搬されなくなる。この数値は、1から順に始めてもいいが、リスト3のように日付形式にしておいたほうがわかりやすいだろう。

リフレッシュ間隔は、データの更新があったかどうかをスレーブサーバがチェックする間隔（秒単位）である。実際にゾーン転送を行うかどうかは、シリアル番号に基づいて判断される。再試行時間は、リフレッシュのチェックを行えなかった場合（ネットワークトラブルやサーバのダウンなど）に、何秒後に再試行するかを指定する。最大有効時間は、チェックができないまま時間が経過した時に、スレーブサーバ上のデータを無効とするまでの時間を秒数で指定する。生存時間（TTL：Time To Live）は、このデータがほ

かのネームサーバでキャッシュされた時に、どれだけ時間データが有効かを示す。この時間が経過するまでは、キャッシュされたデータは有効なものとして扱われる。無効になると、再びネームサーバに問い合わせなければならない。この時間を長くすると、ネームサーバへのアクセス頻度が低下するが、変更の伝搬に時間がかかるようになる。短くすると、変更はすぐに反映されるが、ネームサーバへのアクセス頻度が高くなる。

• NS (Name Server)

<名前>で指定されるゾーンを管轄する（権威のある）ネームサーバの名前を指定する。このデータベースで定義しているゾーンを管轄するネームサーバを示すために使用する場合と、別のゾーンのネームサーバを指定する場合がある。このリストでは名前を指定していないので、SOAレコードと同じ名前となり、このゾーンのネームサーバを指定していることになる。

あるゾーンを管轄するネームサーバは、必ずしもそのゾーン内にある必要

はない。この例では、3つのネームサーバはばらばらの場所にある。

• A (Address)

ホストを示す<名前>に対して、IPアドレスを定義する。<名前>をFQDNで指定した場合は、そのドメイン名に対してIPアドレスが定義される。FQDNで指定しなかった場合（名前がピリオドで終わっていない場合は、その名前の後に、named.confで指定されたゾーンを示すドメイン名（@で表記される名前）が追加される。リスト3の例ではFQDNを使っていないので、nsなどの名前の後に“linux-mag.ascii.co.jp”というドメイン名が付加されることになる。

• CNAME (Canonical Name)

<名前>に対して別名を定義する。別名は正規名（ほかのAレコードなどで定義されている名前）でなければならない。

同じアドレスに複数の名前を指定する場合、同じアドレスを示す複数のAレコードを用意するという方法と、Aレコードを参照するCNAMEレコードを使うという方法があるが、通常はどちらの方法でも問題はない。一部のサーバソフトウェアでは、名前解決の際にAレコードとCNAMEレコードの違いが関係してくることがある。

サブドメインの管理の委任

サイトの規模が大きくなると、組織のドメインの下にさらにサブドメインを置くことになるだろう。たとえば、編集部と技術部があれば、linux-mag.ascii.co.jp、tech.ascii.co.jpといったサブドメインを定義したいかもしれない。そして、各部のホストにはその下位のドメイン名を割り当てる。たと

リスト3 linux-mag.ascii.co.jpゾーンのゾーンデータベースファイル

```

;      SOAレコード
@      IN SOA  ns.linux-mag.ascii.co.jp. postmaster.ascii.co.jp. (
                2000050401      ; シリアル番号 (2000/5/4の01版)
                10800           ; リフレッシュ間隔 (3時間)
                3600            ; 再試行時間 (1時間)
                604800          ; 最大有効期間 (1週間)
                86400           ; 生存時間 (1日)
;
;      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
ns     IN A     192.1.1.1      ; ネームサーバ
mail   IN A     192.1.1.3      ; メールサーバ
db-svr IN A     192.1.1.4      ; データベースサーバ
www    IN A     192.1.1.5      ; Webサーバ
;
;      別名
ftp    IN CNAME db-svr.linux-mag.ascii.co.jp.

```

リスト4 サブドメインを利用する場合のAレコード

www	IN A	192.1.2.3	; www.ascii.co.jp
db-svr.linux-mag	IN A	192.1.3.16	; db-svr.linux-mag.ascii.co.jp
file-svr.tech	IN A	192.1.4.8	; file-svr.tech.ascii.co.jp

例えばdb-svr.linux-mag.ascii.co.jpとか、file-svr.tech.ascii.co.jpといった具合になる。ネームサーバは複数レベルのドメイン名を同一のゾーンで管理できるので、このようにサブドメインを定義しても、リスト4のようにひとつのゾーンファイルに記述できる（このゾーンは、ascii.co.jpを想定している）。

しかし場合によっては、サブドメインを別のゾーンとし、独立したマスタサーバを用意し、サブドメインの管理をそちらに委任したい場合もあるだろう。前述のlinux-mag.ascii.co.jpゾーンは、ascii.co.jpゾーンを管理するネームサーバ（ns.ascii.co.jp）の下位に、linux-mag.ascii.co.jpゾーンを管理するネームサーバns.linux-mag.ascii.co.jpを用意したという形だった。この場合、linux-mag.ascii.co.jp以下のドメイン名に対する名前解決リクエストを処理するのはns.linux-mag.ascii.co.jpになる。ns.ascii.co.jpは、linux-magサブドメイン以下の名前解決リクエストに対して、そのゾーン中のネームサーバを紹介するという形で動作する（図2）。

この場合、linux-magゾーンの設定は、前に解説した通りである。そして、ascii.co.jpのゾーンデータベースに、linux-magというサブドメイン名と、そのゾーンのネームサーバを登録すればよい（リスト5）（NSレコードの行の数字はTTLである）。ただしこの例のように、紹介するネームサーバがそのサブドメイン以下にある場合はどうなるか？ ns.linux-mag.ascii.co.jpのIPアドレスを求められるのはns.linux-mag.ascii.co.jp自身である。このネームサーバの名前を紹介されただけでは、

そのIPアドレスはわからず、次の問い合わせを行えない。そのため、紹介するネームサーバのアドレスを示すAレコードも必要になる。linux-magの下にあるnsのAレコードは、本来linux-magゾーン中で定義されている情報であるが、そのゾーンにアクセスするために、ns.ascii.co.jpにも登録しておくのである。もしサブドメインのゾーン

用のネームサーバがlinux-magの下位ではなく、ascii.co.jpのレベルにあれば、このような重複登録の必要はない。

管理の委任に伴うネームサーバの紹介は、ルート、jpやcomなどのトップレベルドメイン、jpドメインを管理するサーバによる組織ごとのネームサーバの紹介でも行われていることである。

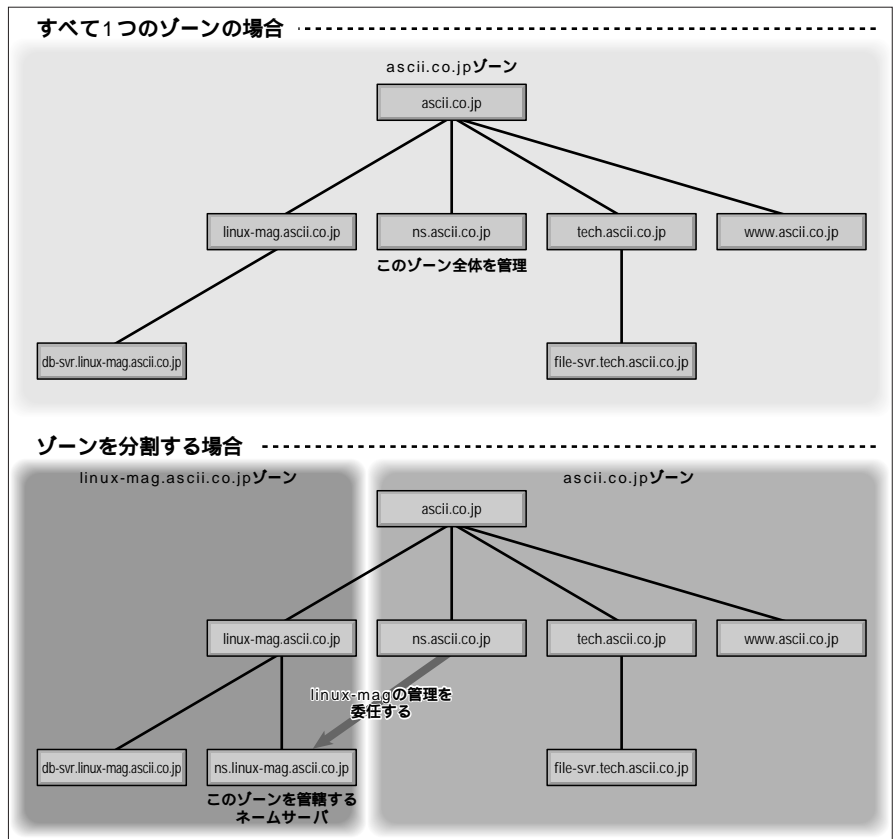


図2 ゾーン分割

リスト5 linux-magサブドメインを委任する

```

; ascii.co.jp.ゾーンのデータベース
; 以下はns.ascii.co.jpが解決
www                IN A      192.1.2.3
file-svr.tech      IN A      192.1.4.8
; linux-magサブドメインの委任
linux-mag          86400   IN NS    ns.linux-mag.ascii.co.jp.
    
```


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4回

- 【プロトコル】(ぶるところ、ぶるとこーる)
- 【RFC】(あーる・えふ・しー)
- 【FTP】(えふ・ていー・ぴー)
- 【HTTP】(えっち・ていー・ていー・ぴー)
- 【SMTP】(えず・えむ・ていー・ぴー)
- 【POP】(ぴー・おー・ぴー、ぽっぷ)
- 【PPP】(ぴー・ぴー・ぴー)
- 【NNTP】(えぬ・えぬ・ていー・ぴー)

プロトコル

【ぶるところ、ぶるとこーる】

(1) 英語で、国家間の協定のこと。転じて、守ったり守られなかったりする約束のこと。このコーナーの原稿締切もプロトコルの一種。

(2) ネットワークでのデータ通信手順のこと。インターネットでは「HTTP」「SMTP」などがなじみぶかい。これらの詳細は「RFC」で参照できるようになっている。

RFC

【あーる・えふ・しー】

(1) 「Request for Comments」の略で、インターネット上で使用されるデータ通信手順などの情報をまとめた一連の文書を指す。2000年5月現在で、3千近い数の文書が公開されている。

(2) インターネットで他人を非難・攻撃するための最高の武器。メーリングリストやネットニュースでの口論時、はじめに「RFCによると」と書いておけば戦闘力がアップする。実際には読んでいなくても、とりあえず冒頭にそう書いておけばよい。「RFC 4654によると」など、もっともらしい数字を書いておくとさらに効果的。

FTP

【えふ・ていー・ぴー】

「File Transfer Protocol」の略。アダルト画像やアダルト音声、アダルトテキストなどのファイルをやりとりするために用いられるインターネットプロトコル。ネット上の好



間違いのFTP

事家により長く使用されてきたが、ライトでカンタンなHを楽しめる「HTTP」の登場で存在意義が希薄になりつつある。現在もアダルトコンテンツのやりとりにFTPを使用しているのは、ごく一部のフェティシストのみ。「えっ? まだ生きてたの?」という意味をこめ、「RIP」(Rest In peace Protocol)とも呼ばれている。

HTTP

【えっち・ていー・ていー・ぴー】

「H Text Transfer Protocol」の略。いま、インターネットでもっとも広く利用されているプロトコルのひとつであり、読んで字のごとく、わいせつ文書を配布することを目的として策定された。配布される文書は専用の記述言語「HTML」(H Text Markup Language)で書かれ、HTTPはこのデータをHなWebサーバとクライアントソフト間でやりとりするために利用される。

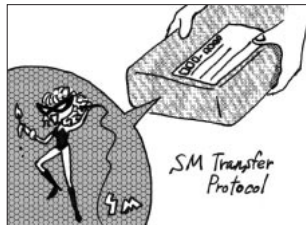
HTTPは、そもそも単純なテキスト転送を目的として設計されたため低機能で、これが配布されるHコンテンツの表現の「足かせ」となっていた。従来は、受信者側で妄想をめぐらせることで表現力のなさをおぎなってきたが、想像力のない初心者や未成年者のインターネット利用が増えてきたことで、その機能の向上が求められている。現在考えられてい

る対策としては、HTMLに代わるより高機能な記述言語の開発がある。これが「XHTML」で、活用すれば18歳未満閲覧禁止のよりハイレベルなコンテンツも実現可能だと目されている。今後は、さらにレーティングを上げた「XXHTML」「XXXHTML」なども登場しそうだ。しかし、識者のあいだでは“秘すればこそ花”という、昔ながらの奥ゆかしきHの伝統が失われるとの批判も根強い。

SMTP

【えず・えむ・ていー・ぴー】

HTTPでは満足できない加虐/被虐嗜好の好事家たちによって策定されたプロトコル。「SM Transfer Protocol」(SM配達プロトコル)の略で、名前のおと、近所の人に見られると



SM配達

恥ずかしい成人向け書籍を完全密送で配達する目的で用いられる。自宅に届くと親、子どもに見つかってしまう人のことを考え「郵便局留め置き」機能を持つため、手紙の配達に関係するプロトコルだとする解説書が散見されるが、それは誤解である。

「ESMTP」「LSMTP」などの拡張規格が存在し、特に前者は、かなりイらしいとの噂がある。マニア垂涎の無修正モノをやりとりできるのかもしれない。LSMTPは「Large SMTP」を意味し、大判の写真集の配送が可能だ。

SMTPは比較的早期に設計されたプロトコルであるため、現在の世情にそぐわない問題点を多く抱えている。ユーザー認証に対応していないことはそのひとつであり、せっかく自宅まで届けてもらっても本人が不在だと誰かれかまわず品物をわたしてしまう。通信経路が暗号化されていないため、配送途中でドライバーが「おー、こいつこんなスケベな本買ってるよ。あんたも好きねえへへへ」などと盗み読みする危険性も高い。よりセキュアなプロトコルの開発に期待したい。

POP

【ぴー・おー・ぴー、ぽっぷ】

若い連中のあいだでだけ通じるプロトコルのひとつ。POPによる通信の物理層としては主にTV、ラジオ、CDなどが利用される。これらの媒体を通じ、日がな大量のデータがマル

チキャストされているが、そのほとんどは、良識あるLinux magazine読者の中核をなすおじさん層には無価値である。POPで知人とデータ交換する際には、物理層にMDを使ったり、プレゼンテーション層にMP3を使用することは違法性が高いので注意が必要だ。

POPには、さまざまなバリエーションが存在するが、なかでも日本では「J-POP」というプロトコルが隆盛を誇っている。特に、カラオケボックスのネットワークシステムでは、J-POPによるトラフィックが帯域のかかなりの割合を占めるという。そしてこのプロトコルを牛耳っているのが秋元康、森雪之丞、小室哲哉、つんくら。免疫のない人間がカラオケボックスでJ-POPを使用すると、精神汚染され、セーラー服を脱がされてしまったり、コロ助とコロツケを作ってしまったのでリスクが大きい。あらかじめレパトリーを増やしていく、マイマイクを持参するなど、通信には万全の体勢でのぞみたい。最近の事例では、J-POP通信中にセクシービームを発射してしまった、あるいは6人だったのが7人になり5人になり10人になったりした、という憂うべきケースも報告されている。

before SMTP：よくわからないけれど、サドマゾのカップルがプレイする前にカラオケボックスでモー娘を歌って雰囲気盛り上げることかもしれない。

PPP

【ぴー・ぴー・ぴー】

西瓜やアイスクリームを食べ過ぎ(編集部注：限界をこえて下品だったので削除しました)。

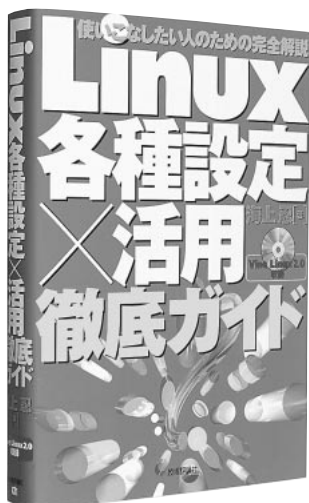
NNTP

【えぬ・えぬ・ていー・ぴー】

「NetNews de Tatakau Protocol」(ネットニュースで闘うプロトコル)の略。fj.*やjapan.*などのニュースグループで、不毛な口論を繰り返すための手続きを定めたプロトコルのこと。

最近では、HTTPにNNTPをカプセル化して載せる技術(「Web掲示板」と呼ばれる)が開発されたため、ピュアなNNTPにはスポットライトが当たらなくなってきたが、どうどうめぐりの議論、出口の見えない中傷合戦には最適だけに、復権を心待ちにする人が多い。

Books



Linux 各種設定 × 活用 徹底ガイド

海上 忍 著

技術評論社

A5判 / 432ページ / CD-ROM2枚付き

本体価格 2680円

「各種設定」と聞くとどこかあいまいな感じがするが、本書は「徹底」にウエイトをおいた本である。なにがそんなに徹底なのかというと、紹介している設定の多さだ。/etc/inittabをはじめとするシステム起動時に読み込まれるファイル、fstabやconf.moduleなどのハードウェアの設定に関連するもの、ユーザーのホームディレクトリにひっそりと置かれているドット(.)付きのスクリプト類、さらには、X Window Systemでの日本語環境、Emacs、GIMPと広範囲にわたっている。各ファイルの構文・書式、用例、オプション一覧、設定サンプルなども豊富で、かつ、きちんとまとまっている。

ひとつだけ残念なのは索引が充実していないこと。そのため、せっかくの内容もリファレンス性に欠けるところがある。とはいえ、Linuxのカスタマイズの際には重宝する1冊だ。2枚組みの付録CD-ROMには、Vine Linux 2.0 FTP版のバイナリとソースを収録している。

GTK+とGladeで作る Linux プログラミング超入門

鈴木哲也 著

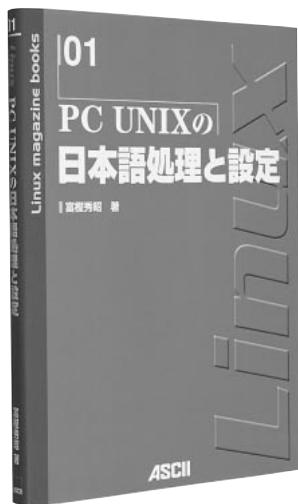
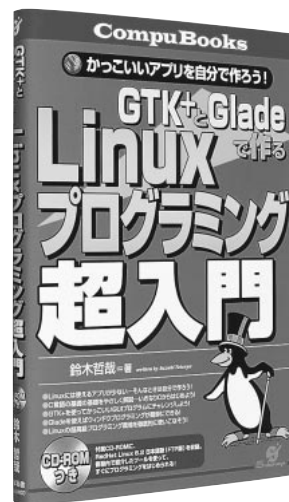
すばる舎

A5判 / 224ページ / CD-ROM付き

本体価格 2400円

ご存知の方も多いだろうが、GTK+は「Photoshopキラー」と呼ばれるグラフィック処理ツールGIMPを作成するために生まれたツールキットである。Linuxの統合デスクトップ環境として人気の高いGNOMEでも、GTK+が利用されている。Gladeはマウス操作でウィンドウなどの部品（ウィジェット）を組み立てるだけでGUIを構築できるインターフェイスビルダーである。gladeは作成したインターフェイスのCソースコードも生成する。

プログラミング入門書の古式に則って、C言語の「Hello World」プログラムから始まっている本書だが、GTK+とgladeのコンビネーションによって、非常に難しいというイメージの強いX環境のプログラミングを、初心者でも楽しみながら体験できるよう配慮されている。とかくネットワークでの利用ばかりが目されるLinuxだが、プログラム開発プラットフォームというもうひとつの魅力に触れてみたい人にお勧めしたい。



PC UNIXの日本語処理と設定

高樫秀昭 著

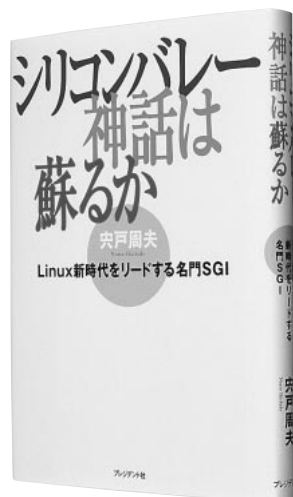
アスキー

A5判 / 184ページ

本体価格 1700円

日本語処理の問題で、まず気になるのがX環境での日本語の表示だ。個人的には、小さな文字のフォントがとても気になる。などと思っていたが、読み進めていくうちに、あらためてコンピュータで日本語を処理することの難しさを感じた。

創刊第2号から2000年3月号まで本誌において連載され、好評をいただいた「Linuxの日本語環境」に加筆・訂正を加えてまとめた1冊。著者も冒頭で述べているように、PC UNIXにおける日本語に関するすべてを網羅するものではなく、トピックごとに、ほどよいバランスで解説がなされている。ただし、UNIXそのものや基本的なコマンドに対する知識を前提に書かれているため、初心者にはやや敷居が高いかもしれない。それでも、そこかしこにヒントとなる「とっかかり」が用意されているので、コマンドリファレンス、システム設定ガイドなどを手元に用意して独自の日本語環境構築に挑戦してみよう。



シリコンバレー神話は蘇るか

宍戸周夫 著

プレジデント社

A5判 / 232ページ

本体価格1600円

サブタイトルは「Linux新時代をリードする名門SGI」。SGIといえば高性能グラフィックス処理という印象がたいへん強いが、科学技術分野での高度な計算処理にもSGIのマシンは利用されてきた。どちらも高度な演算能力が求められる分野であり、その中でSGIは超並列処理をはじめとする高い技術力を養ってきた。また、OpenGLを公開する一方で、独自のアーキテクチャに固執するという両面性をもった不思議な存在でもあった。

本書は、独自のアーキテクチャに独自のOSというクローズドな世界から、オープンソースビジネスモデルへの転換を図ったSGIの変遷を、Linuxとのかかわりを中心にとどっていく。いわゆる「ビジネス書」であり、少しホメすぎの感もなくはないが、日刊工業新聞出身という筆者の経歴を生かした技術的な側面からのアプローチもあって興味深い内容となっている。

改訂新版 標準Red Hat Linux リファレンス



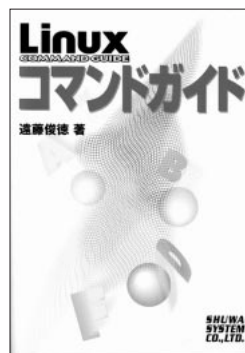
David Pitts、
Bill Ball ほか著
石川直太 訳
レッドハット株式会社 監修

インプレス

B5判変形 / 864ページ

本体価格 3980円

Linuxコマンドガイド



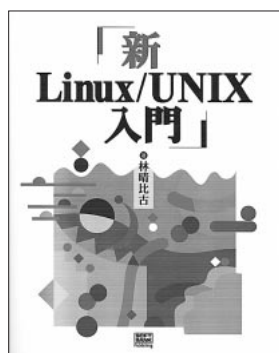
遠藤俊徳 著

秀和システム

A5判 / 272ページ

本体価格 1800円

新 Linux/UNIX入門



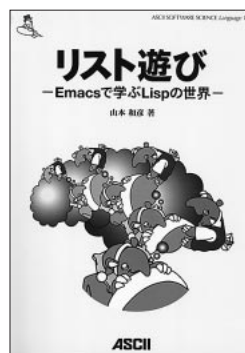
林 晴比古 著

ソフトバンク
パブリッシング

B5判変形 / 552ページ

本体価格 3800円

リスト遊び - Emacsで学ぶLispの世界 -



山本和彦 著

アスキー

A5判 / 128ページ

本体価格 1200円

読者の声

俺にも
いわせろ!

毎月たくさんのおハガキをありがとうございます。私事で恐縮ですが、今月は大学時代の友人からも読者ハガキをもらいました。嬉しいのはもちろんなのですが、名指しで「結婚しないのかぁ～」と書くのはやめてほしいトコロです。

7月号特集1と付録CD-ROMへのお便り☆

今月号はCD-ROMが4枚とお買い得でした。さらに、特集1「ディストリビューション戦国時代」もサーバ系について書かれていたのでよかった。デスクトップ系とサーバ系、どちらか一方だけということではなく、両方を比べて読める配慮がいい!

(岩手県 菅原慎一さん)

特集1を読み、また新しいディストリビューションが出るのかと楽しみが増え、幸せな気分です。付録のCD-ROMにも4つのFTP版ディストリビューションが収録され、迷うこともありそう……。欲をいえば、今度はDebian GNU/Linux 2.2も収録をお願いします。

(秋田県 渡辺 修さん)

Kondaraを導入してみたいと思っていたところ、貴誌に出会いました。うまく導入できたら報告します。

(神奈川県 立沢昌弘さん)

④Linuxには数多くのディストリビューションがあり、統一性がないという意見もあります。確かに、ディストリビューションによって設定の仕方が微妙に違うなど、混乱することも多いのですが、いろいろと試しながら使い道や好みに合ったものを探ることができるというのはLinuxの魅力でもあります。また、ディストリビューション間での切磋琢磨や、パズールモデルに則った開発成果の共有も重要なポイントですね。

7月号、特集2へのおハガキ☆

ちょうど、グラフィックスカードをMillennium G200から交換しようかと物色していたので、「Project X」は大変参考になりました。G400、Voodoo3、TNT2などを考えていたのですが、GeForceはLinuxでも速いんですねえ。もっと働いて予算を上積みしないと……。プレゼントが当たれば話は別ですが。

(神奈川県 富山智彦さん)

嬉しがって、XFree86 4.0をインストールしてみました!! めっちゃ良いです。ディスプレイの設定をしなくてもきれいに映るし、XawTVがフルスクリーンで表示できた!! でも、ひとつ問題が……。よく使うxdviが変だ～。ページ送りとかの欄がない! これじゃ使えないよ～。

(兵庫県 岡本 齊さん)

Windows 95とRed Hat Linux 6.2Jをデュアルブートで使用しています。Windows 95に画面を合わせるとRed Hat Linuxを立ち上げたとき、画面が横にずれていたのですが、7月号のXFree86 3.3.6の設定方法で解消されて嬉しかった(ちょっとしたことなんです)。

(島根県 阿式芳夫さん)

④XFree86は4.0になって大きく変わりました。まだちょっと不安定な部分もあるようですが、3D描画支援機能など、最近のグラフィックアクセラレータチップの性能を引き出せるようになってきました。特に、GeForce256の3D性能は目を見張るほどです。Linux magazineラボの担当者は、さっそく自宅マシン用にカードを購入したそうです(イーサネットカードを特集したときは、カードとハブを買っていました)。

XFree86の設定も今では専用ツールでできるようになっています。これらのツールは便利だけど、かゆいところに手が届かないことが多いのもまた事実です。設定ファイルXF86Configの内容を知っていれば、より細かいチューニングができます。こんなところもLinuxやPC UNIXの面白さなのかもしれません。

Javaプログラミング道を極めよう☆

「Javaプログラミング入門」とても

参考になりました。今、学校でJavaを勉強しているのですが、教授に質問しても全然教えてくれないので……。とても面白かったし、読みやすかったです。これからも楽しみにしています。頑張ってください。

(神奈川県 佐藤憲一郎さん)

Linux + Javaでプログラム開発に挑みたいと……。Javaもフリーだけど、まだまだ敷居が高いです。そのうえLinux……。Windowsをしていた人は大変です。あー、DOSの時代が懐かしい。

(富山県 山崎次郎さん)

☉ Javaは仮想マシン (JVM) 上で動作するため、CPUやOSの種類に依存しないところが魅力です。数年前はPCが非力だったこともあり、JVMのオーバーヘッドが無視できませんでした。でも、近頃のマシンではサクサクと動いてくれますね。同じプログラムがそのままLinux、Windows、Macintosh、OS/2で動くなると、いい時代になったものです。

6月号の付録CD-ROMに収録したJava開発環境JBuilder 3.5 Foundation日本語版によるプログラミングもぜひお試しください。

強運世界一？

ここに紹介されることを夢見て書き続けるので、いつか紹介してくださいね。

って、その間に懸賞に2回も当たってしまった。感謝、感謝！

(東京都 細川 誠さん)

☉ 2回も当選とは！！ しかも、このコーナーにも！ 細川さんは本当にラッキーな方なんです。担当もあやかりたいものです。

プレゼントコーナーは、人気があるもの

とないもので倍率が大きく違います。過去には、希望した方全員が当たりというものもありました。みなさん、ねらいを定めてバッチリ当ててください。

ネイティブ開発環境を強化！

Kylixはまだか？ まだなのか？！

(兵庫県 木下政幸さん)

☉ Inprise / Borlandは、Kylixプロジェクトとして、Delphi for Linuxを開発しており、今年の後半にリリースするというのです。また、Linux用のC / C++コンパイラやコンポーネントライブラリも開発中で、これらが完成すれば、DelphiやC++Builderで書かれたWindowsプログラムを容易にLinuxに移植できるとしています。

Linuxの開発環境は、gccを中心としたものが多いのはみなさんもごぞんじのとおりですが、このプロジェクトによって、開発環境の新時代が訪れるのかもしれない。

寮内LANでLinux

学生寮に住んでいるという利点を活かして、実験的に寮内LANを構築してみました。利用者は私を含めても3人しかいませんが、ひとりがLinux使い、もうひとりがWin使い。私がMac使いというわけで、設定するサービスの数が多くてクラクラきてます。まあ、Linuxの本領発揮ですね。

余談ですが、プライオリティを最大にしてSETI@Homeの計算をしていたら、サーバが遅いと文句をいわれました。反省。

(宮城県 本多洋平さん)

学生寮でLANというと、OCNエコノミーをみんなで使う話を思い出します。大都市ではNTTのフレッツ・アイで24時間イ

ンターネットを使い放題になりつつありますが、日本の通信環境はまだです。それはさておき、WindowsやMacintoshなどが混在するLANでもLinuxは大活躍です。これがすべてフリーソフトウェアで実現されているなんて、世界中にはすごいプログラマーがいるのだなあと感じますね。編集部でもSETIが原因でサーバが重くなったことがありました。このため、「サーバでSETIは禁止 by 副編集長」令が發布されました。

遠藤さんの疑問

最近、すごい発見をしました。なんと今まで気がつかなかったのでしょうか。それは、“HOGE”という文字です。Linuxに関連する書籍のほとんどが、例文、例題などにこの文字を使用しています。「ホゲ」と読むのか「ハゲ？」と読むのか定かではありませんが、なんか昔のテレビ番組「ゲバゲバ90分」に似ていて、つい「ホゲホゲ」といってしまいます。この言葉の語源、歴史などについて知っている方はいらっしゃいませんか？ ぜひ、探求して極めたいものです。それから、私もペンネームを「ホゲホゲ遠藤」といたします。あれ？ このコーナー、ペンネームありでしたっけ？ (^;)

(山形県 ホゲホゲ遠藤さん)

☉ 今月もお楽しみお八ガキをありがとうございます。“hoge”問題？ については、編集部でも盛り上がりまして、調べた結果を今の特集1でご紹介させていただきました。締切直前であることも忘れて調べまくった特1担当がその後、地獄を見たのはいうまでもありません。

ペンネームでの投稿は、もちろんアリアリです。来月もお便りをお待ちしております。

付録CD-ROMに収録した

LASER5 Linux 6.2のインストール

本誌付録CD-ROM収録のLASER5 Linux 6.2はFTP版です。非商用ソフトだけが含まれています。また、製品版を販売しているレーザーファイブからサポートを受けることはできません。



ブート用フロッピーディスクの作成

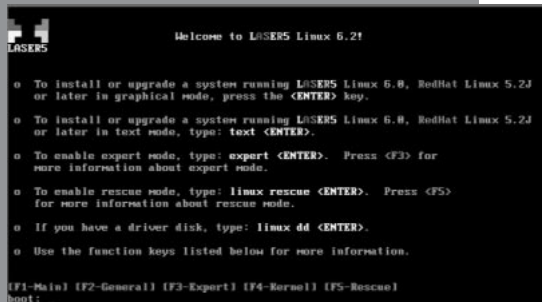
インストールするマシンがCD-ROMから起動できる場合は、CD-ROMからブートしてインストーラを起動します。CD-ROMから起動できない場合は、以下の手順で、インストーラ起動用のフロッピーディスクを作成します（ここでは、フロッピーディスクドライブがA:であるとして解説します）。

(1) Windowsのエクスプローラで、CD-ROMの「dosutils」というフォルダを開き、その中にある「rawrite」をダブルクリックします。

(2) DOS窓が開き、ファイル名の入力を促してくるので、

¥images¥boot.img

と正確にタイプして[Enter]を押します。さらに、フロッピードライブ名を求めてくるので、Aをタイプして[Enter]を押します。最後にフロッピーがセットされているかを確認して、[Enter]を押すと、フロッピーの作成が始まります。



インストーラの起動

作成したフロッピーディスクや、CD-ROMをドライブにセットして、マシンを再起動します。インストーラが起動し、「boot:」というプロンプトが表示されたら[Enter]を押します。

しばらくすると、Xを使ったグラフィカルな画面が表示されます。インストール画面がうまく表示されない場合は、「boot:」の箇所です「text」とタイプして[Enter]を押します。こうすると、テキスト画面のインストーラが起動します。

使用言語とキーボードの設定

Linuxで日本語を使う場合は、デフォルトのまま「Japanese」を選択して「次」を押します。

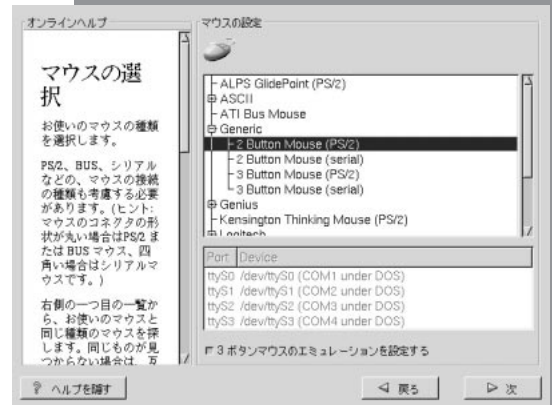
次のキーボード設定では、「モデル」から「Japanese 106-key」を、「レイアウト」から「Japanese」を選択することで、日本語106キーボードが使えるようになります。「デッドキー」の欄は「デッドキーを無効にする」を選択します。「デッドキー」はヨーロッパ言語環境でアクセント付き文字を入力するのに使われるキーのことです。



マウスの設定

デフォルトでは「3 Button Mouse(PS/2)」が選択されています。PS/2タイプの2ボタンマウスを使う場合は「2 Button Mouse(PS/2)」をチェックします。「3ボタンマウスのエミュレーションを設定する」は、2ボタンマウスの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を代用する機能です。2ボタンマウスを使用する場合は、チェックしたまま「次」を押します。

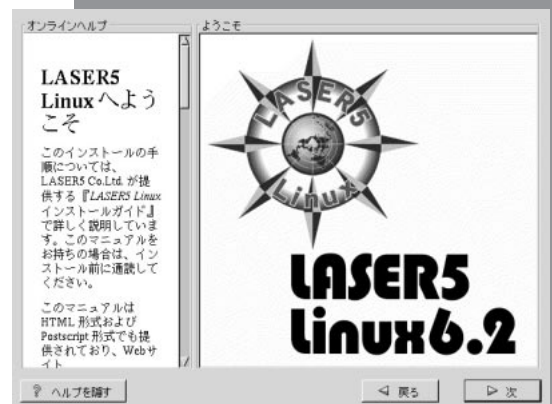
4



インストールの開始

さて、ここから本格的なインストール作業の開始です。「次」を押します。

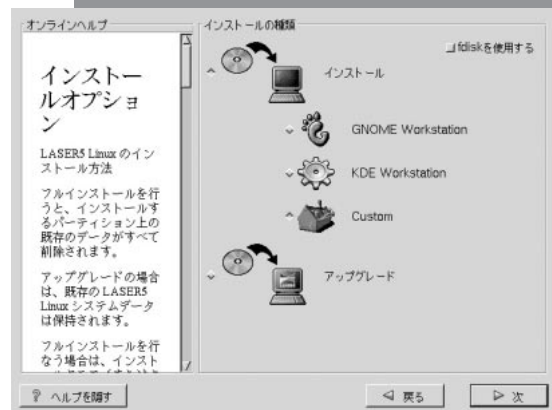
5



インストールタイプの選択

「GNOME Workstation」と「KDE Workstation」を選択すると、ブートローダLILOがMBRにインストールされます。Partition Magicなど、他のブートローダでOSを起動するときは、これらのタイプを選択してはいけません。ここでは、より柔軟なオプション選択ができる「Custom」を選択して「次」を押します。「アップグレード」は、すでにLASER5 Linuxの以前のバージョンをインストールしているユーザー用です。

6

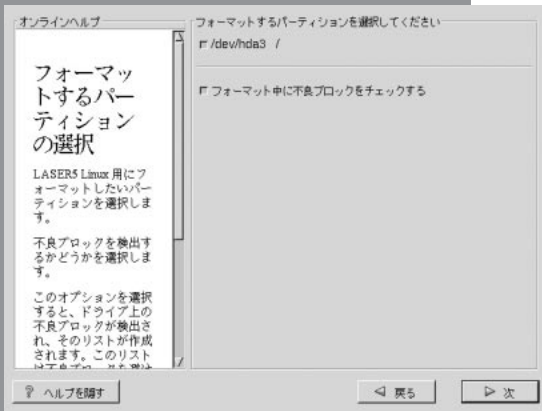


パーティションの作成

Linuxをインストールするには、最低限Linuxシステム用とSwap用の、2つのパーティションが必要です。「追加」を押すと、画面のようなダイアログが表示されます。Linuxシステム用のパーティションは、「マウントポイント」に「/」を、「パーティションタイプ」は「Linux native」を選択します。Swap用パーティションは「Linux swap」を選択します。各パーティションサイズは、以下を目安にして設定してください。

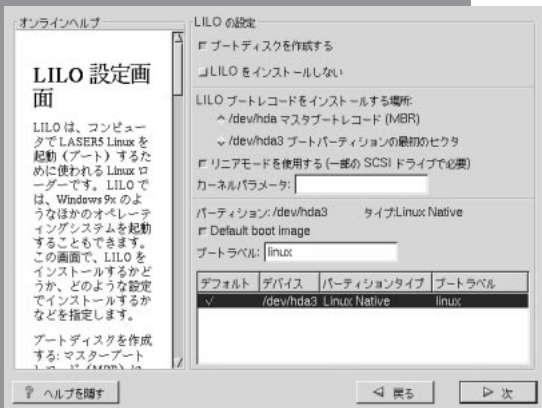
Linuxシステム用 : フルインストールの場合は2Gバイト程度以上
Swap用 : 搭載メモリの1 ~ 2倍程度





パーティションのフォーマット

前の場面で作成したパーティションをフォーマットします。チェックされている「/dev/hdax」パーティションがフォーマットされます。「フォーマット中に不良ブロックをチェックする」を選択しておくと、パーティションのフォーマット中に、ディスクの不良箇所を調べてくれます。急いでいなければ、チェックを入れておくと良いでしょう。最後にもう一度フォーマットするパーティションを確認して、「次」を押します。



LILOの設定

「LILOブートレコードをインストールする場所 (LILOを入れるレコード)」は環境に応じて選択します。

/dev/hdaマスターブートレコード (MBR)

- ・LILOを使用してWindowsとLinuxを起動時に選択する

- ・ディスクにはLinuxのみをインストールする

/dev/hda3ブートパーティションの最初の(第1)セクタ

- ・System Commanderなど、LILO以外のブートマネージャを使用する

緊急時のために「ブートディスクを作成する (ブートディスクの作成)」をチェックしておくとい良いでしょう。



ネットワークとタイムゾーンの設定

イーサネットを使い、ネットワークを利用する場合は、この場面で設定します。「起動時にアクティブにする」はチェックしたままで、DHCPでネットワーク情報を取得する場合は「DHCPを使用する」を選択し、DHCPを使わない場合は「DHCPを使用する」のチェックを外して、マシンのIPアドレス、DNSなど、ネットワーク利用に必要な情報を入力します。ダイヤルアップでインターネットに接続する場合は、インストール後に「kppp」や「netcfg」を起動して、ネットワークを設定します。

次のタイムゾーンの設定は、デフォルトで「Asia/Tokyo」が選択されています。マシンを日本で使うときは、このままの状態です。「次」を押します。



ユーザーアカウントと認証の設定

Linuxを使用するユーザーを設定します。まず、Linuxシステムに必須の管理者Rootのパスワードとして、「Rootパスワード」と「確認」に同じものを入力します。

次に一般ユーザーのアカウントを設定します。「アカウント名」、パスワードとして「パスワード」と「パスワード(確認)」に同じものを、「フルネーム」にユーザーのフルネームを入力して、「追加」を押します。この要領で複数の一般ユーザーを作成できます。ユーザー設定を終えたら「次」を押します。

認証の設定場面は、そのままの状態です。「次」を押します。

パッケージグループの選択

おおざっぱなパッケージグループを選択します。パッケージ群の意味がわかるユーザーは、好みのパッケージ群を選択し、どのパッケージ群を選んでよいかわからないユーザーはすべてのパッケージをインストールするために、最下段にある「Everything」を選択して「次」を押します。



Xの設定

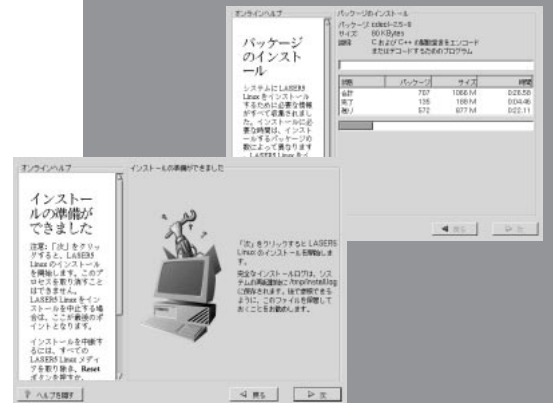
モニタの一覧が表示されます。リストに使用するモニタがない場合は、「Generic」の「Generic Monitor」を選択します。

次にXサーバの設定です。ほとんどのビデオカードは自動で認識されます。色数や解像度を手動で設定する場合は「X設定のカスタマイズ」をチェックし、グラフィカルなログイン画面を使う場合は「グラフィカルログイン」をチェックして「次」を押します。



パッケージインストールの開始

画面(手前)で「次」を押すと、パッケージのインストールが始まるので、インストールが終わるまでしばらく待ちます。



インストールの終了

LILOの設定場面で「ブートディスクを作成する」をチェックした場合は、手前の画面が表示されます。「次」を押すと、緊急時用のブートディスクの作成が始まります。フロッピーの作成が終わると、奥の画面が表示されます。フロッピーディスクを抜いて「終了」を押します。

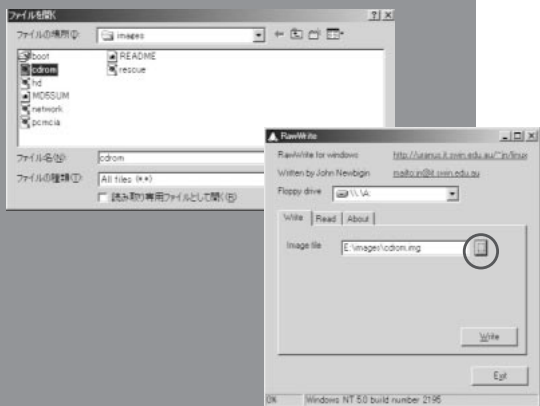
以上でインストール作業は終わりです。お疲れさまでした。



付録CD-ROMに収録した

Linux Mandrake 7.1の インストール

本誌付録CD-ROM収録のLinux Mandrake 7.1はFTP版です。非商用ソフトだけが含まれています。また、製品版を販売しているMandrakeSoft Inc.およびMandrakeSoft S.A.からサポートを受けることはできません。



ブート用フロッピーディスクの作成とインストーラの起動

マシンがCD-ROMから起動できない場合は、以下の手順でインストーラを起動するためのブートディスクを作成します。

- (1) CD-ROMと空のフロッピーをドライブへセット
 - (2) エクスプローラでCD-ROMの中の「dosutils」フォルダ内にある「rawwritewin」をダブルクリック
 - (3) 画面（手前）の丸印をクリックしてCD-ROM内の「images」フォルダから、「cdrom.img」を選択
 - (4) 「Write」ボタンを押してブートディスクを作成し、「Exit」で終了
- 作成したブートディスクかCD-ROMをドライブにセットして、マシンを起動すると、インストーラが立ち上がります。

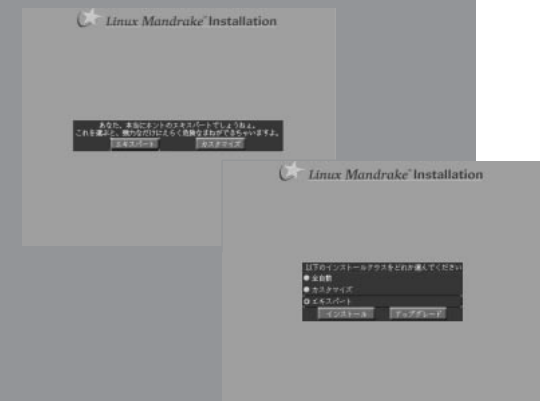


インストール中に使用する言語の選択

Mandrakeのインストーラが起動すると、各メニューが英語で表示されます。このあとの作業中に、メニューを日本語で表示するために「Japanese」を選択して「OK」を押します。

Mandrakeのインストーラには「もどる」というメニューがありませんが、左側にあるメニューをマウスでクリックすれば、その設定を再度実行できます。

2



インストールクラスの選択

インストールクラスは「全自動」、「カスタム」、「エキスパート」のいずれかを選択します。「全自動」はパーティションの作成を自動で行うため、MandrakeをほかのOSと共存させる場合に選択してはいけません。「カスタム」と「エキスパート」は、「全自動」よりも柔軟な設定ができるインストールクラスです。ここでは、あとの設定場面で、ブートローダとXサーバが選択できる「エキスパート」クラスで解説しますが、「カスタム」を選択した場合も、ほぼ同じ手順でインストールできます。

次の場面でも「エキスパート」を押します。

PCIカードとSCSIカードの検索

マシンに搭載されているPCIカードとSCSIカードをインストーラがスキャンします。それぞれの環境に合わせて設定してください。

4

インストールタイプの選択

「ノーマル」と「開発用」の主な違いは、C言語のコンパイラなどを含むかどうかです。「ノーマル」を選択すると、コンパイラはインストールされません。インストール後に、アプリケーションをソースからコンパイルして使う場合は「開発用」を選択してください。Linuxをサーバ専用にする場合は「サーバ」を選択します。

5

マウスとキーボードの選択

使用するマウスタイプを選択します。PS/2タイプの2ボタンマウスを使う場合は、「汎用マウス (PS/2)」を選択して「OK」を押します。

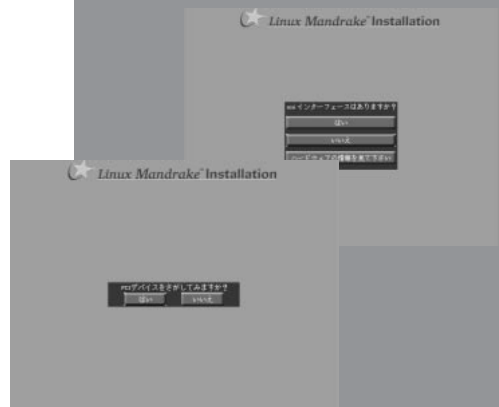
次にキーボードの設定です。日本語キーボードが選択できないので、ここでは「アメリカ式キーボード」を選択します。日本語キーボードへの設定変更は、本文最後の説明を参考にしてください。

6

Linux使用時の言語選択

インストール後にLinuxで使用する言語の選択です。Linuxで日本語を使用するときは「Japanese」を選択して「OK」を押します。

次の「その他の質問」は、そのまま「OK」を押して次へ進みます。





パーティションの作成

Linuxを使用するには、Linuxシステム用とSwap用の2つのパーティションが必要です。棒状のエリアをマウスでクリックして「作成」を押すと、画面のようなダイアログが表示されます。Linuxシステム用のパーティションは、マウントポイントから「/」を選択します。MandrakeではLinuxシステム用に、現在Linuxで標準のext2ファイルシステムと、ジャーナリング機能を備えたReiserFSが選択可能です。せっかくなので、パッケージのフルインストールにそなえ、ReiserFSのパーティションを2Gバイト程度作成してみましょう。

同じ要領でSwap用パーティションを作成します。サイズはマシンが搭載する実メモリの1~2倍程度を目安にします。



パーティションのフォーマット

黄色の星マークでチェックされたパーティションのデータは全て消去されます。本当にフォーマットしてよいかどうかを念入りに確認して「OK」を押します。

次の場面ではパーティションの不良箇所をチェックします。時間に余裕があれば、パーティションをチェックして「OK」を押します。



CD-ROMの選択

本誌に収録されているのはインストール用のCD-ROMのみなので、「キャンセル」を押して次へ進みます。



パッケージグループの選択

デフォルトでは、すべてのグループがチェックされています。必要のないパッケージグループはチェックを外しますが、どれを外してよいのかわからない場合は、「個別パッケージの選択」のみチェックを外して「OK」を押します。

パッケージのインストール

ここは100%のままで「OK」を押します。パッケージのインストールが終わるまで、しばらく待ちます。

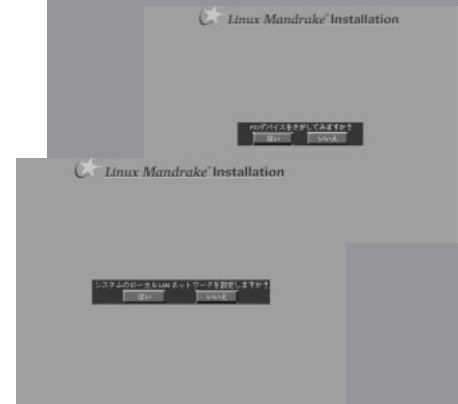
12



ネットワークの設定 (その1)

Mandrakeでは、イーサネットとモデム(TA)を使ったネットワークの設定が可能です。ここではイーサネットを使ってのネットワーク設定を説明します。次の場面で「はい」を押すと、インストーラがネットワークカードをスキャンします。

13



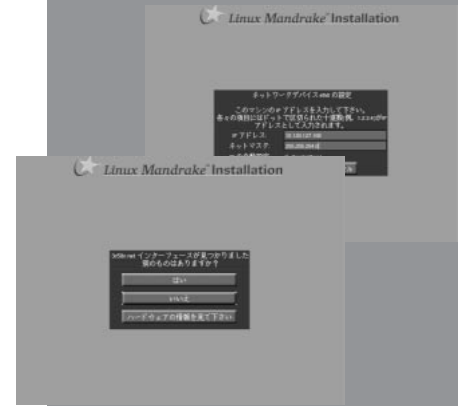
ネットワークの設定 (その2)

ネットワークカードのスキャン結果が表示されます。ネットワークカードを2枚使うときは「はい」を選択し、自動認識されたネットワークカードのみを使う場合は「いいえ」を選択して次へ進みます。

IPアドレスなどのネットワーク情報を、DHCPで自動取得する場合は「IPの自動設定」をチェックし、そうでないときは各欄にIPアドレスやネットマスクなどの情報を入力します。

次の「暗号ソフトのダウンロード」は「いいえ」を押して次へ進みます。

14



タイムゾーンの設定

マシンを日本で使う場合は「Tokyo」を選択して「OK」を押します。マシンの時間がGMT(グリニッジ標準時)に設定してある場合は「はい」を、そうでない場合は「いいえ」を選択して次へ進みます。

15





デーモンの設定

Linux起動時に立ち上げるデーモン（バックグラウンドで動作するプログラム）を選択します。各メニューをマウスでポイントすると、そのデーモンの簡単な説明が表示されます。どれを選択してよいかわからない場合はそのままの状態ですべて「OK」を押します。

16



ユーザーアカウントの設定

Linuxの管理に必要なrootのパスワードとして、「パスワード」と「パスワード（もう一度）」に同じものを入力して「OK」を押します。

次は一般ユーザーの作成です。上から順番に「実際の氏名」、「ユーザー名」、「パスワード」、「パスワード（もう一度）」を入力して「受け付けるユーザー」を押します。この作業をくり返すことで、複数の一般ユーザーを作成できます。ユーザーを作り終えたら「終了」を押します。

17



緊急時用ブートディスクの作成

1台のハードディスクに複数のOSをインストールする場合など、ブートローダが上書きされて、Linuxが起動できなくなることがあります。このような緊急時のために、できるだけブートディスクを作成しておきましょう。

18



ブートローダの設定

Mandrakeでは、多くのディストリビューションで採用されているLILOのほかに、起動するOSをテキストのメニュー画面で選択する、Grubというブートローダを選択できます。MandrakeではGrubの使用が推奨されているようです。ここではGrubを選択しますが、好みで選択してください。

「ブートデバイス」は、System Commanderなどを使う場合は「/dev/hdax」を選択し、LILOやGrubのみでOSを起動する場合は「/dev/hda」を選択して「OK」を押します。

19

Xの設定(その1)

Mandrakeを「エキスパート」クラスでインストールすると、ここでXFree86 4.0が選択できます。より安定した環境でLinuxを使いたい場合は「いいえ」を、新しいXFree86を試してみたい場合は、「はい」を押して次へ進みます。

使用するモニタがリスト中に入っていない場合は、「汎用」からモニタに適したものを選択してください。モニタのマニュアルを参照すると確実な作業が行えます。



Xの設定(その2)

まず解像度と色数を選択します。選択後は何度でも表示のテストができるので、うまく表示できるまで設定変更して、表示をテストします。なお、Xの表示テストがうまくいった場合、「はい」と「いいえ」が文字化けした状態で表示されます。左側のボタンが「はい」なので、うまくXが表示できたら、左側のボタンを押してください。

Linuxのログイン画面をグラフィカルなものにするときは「はい」を、テキスト表示にするときは「いいえ」を押して次へ進みます。「いいえ」を選択した場合は、Linuxが起動したあとに、コンソール上で「startx」とタイプすれば、X Window Systemを使用できます。



自動インストール用フロッピーの作成

ここで「はい」を選択すると、複数のマシンに同じ環境のMandrakeをインストールするためのフロッピーが作成できます。「はい」を選択する場合は、先に作成した緊急時用のブートディスクをドライブから抜き、別のフロッピーをセットしてください。1台のマシンでMandrakeを使う場合は「いいえ」を選択します。

以上でMandrakeのインストール作業は終わりです。お疲れさまでした。



Linux Mandrake 7.1は、インストール時に日本語キーボードが選択できません。日本語キーボードを使う場合は、インストール後に以下の手順で設定してください。

```
XFree86 4.0を使用する場合
# vi /etc/X11/XF86Config-4
とXの設定ファイルを開き、
Option "XkbModel" "pc105"
Option "XkbLayout" "us"
```

の「pc105」を「jp106」へ、「us」を「jp」へと書き換えて、[CTRL]+[ALT]+[Backspace]としてXを再起動します。

```
XFree86 3.3.6を使用する場合
# vi /etc/X11/XF86Config
とXの設定ファイルを開き、
XkbModel "pc105"
XkbLayout "us"
```

の「pc105」を「jp106」へ、「us」を「jp」へと書き換えて、[CTRL]+[ALT]+[Backspace]としてXを再起動します。