

NEWS EXPRESSES

Distribution

Software

Hardware

Headline

Event

e-ビジネス領域を広げる 次世代プラットフォーム Oracle9i

日本オラクルは、e-ビジネス領域でスケーラビリティと可用性、パフォーマンスを実現したプラットフォーム製品「Oracle9i」を10月1日より発売する。

Oracle9iは、「Oracle9i Database」と「Oracle9i Application Server」で構成される。

Oracle9i Databaseは、新たなデータベース・クラスタリング機能である「Oracle9i Real Application Clusters」の採用によりアプリケーションにまったく変更を加えることなく、全てのアプリケーションに対してクラスタ化されたデータベースの可用性、スケーラビリティ、パフォーマンスを提供。「Oracle9i Data Guard」による事故、災害など計画外停止による障害からの保護機能を強化した。また新たに「Oracle Internet File System」を導入。これによってイメージや音声、ドキュメント、XML文書などのあらゆる情報をデータベースで管理、検索でき、多様なデータの管理・活用が可能となった。

Oracle9i Application Serverは、Java対応機能を強化し、高速化した統合型アプリケーションサーバ。J2EEパフォーマンスを向上させ、大規模なユーザのアプリケーション/データ連携を実現している。Webサービスからのリクエストを受け付けるためSOAPに対応、あらゆるWebコンテンツを高速配信するためのキャッシング機能が強化された。

Oracle9iは、データベースとアプリケーションサーバを組み合わせることで、e-ビジネス、ブロードバンドなどのマーケットに対応したプラットフォーム製品となっている。



発売日 2001年10月1日
発売 日本オラクル株式会社
TEL 03-3511-5331
価格 16万円(5ユーザー)~
URL <http://www.oracle.co.jp/9i/>



ターボリナックス、主要ディストリビューション開発を日本に集約

2001年8月30日

ターボリナックス ジャパンは、CEO Special Advisorとして2001年1月から同社に駐在していた米Turbolinux本社のJesse Casman氏が、9月1日よりActing Presidentとしてターボリナックス ジャパンのオペレーションを統括すると発表した。現社長の小島國照氏は2001年8月31日付けで辞任し、同社特別顧問に就任する。

ターボリナックスでは、従来より日本を中心として、日米中3国でLinuxディストリビューションを開発してきたが、今後も、ビジネスニーズに応える信頼性と安定性の高いLinuxディストリビューションを、継続的に安定して供給するために、主要Linuxディストリビューションの開発を日本に集約する。

IA64、S390をはじめとするハイエンド向けディストリビューションは、引き続きTurbolinux本社で開発し、ハイエンド・エンタープライズシステムの開発を手がけるTurboLabsで、商用のLinuxソリューション製品の開発を行う。

TurboLabsの最初の製品は、大量のLinuxサーバを効率的に管理・運用するためのソリューションで、今秋には日本市場にも投入する予定。

「米国でのLinux製品の販売から事実上撤退」と一部の報道があったが、北米市場においてはパートナー企業へのOEMに特化してLinuxビジネスを強化するとしている。

ターボリナックス ジャパン

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

「GCC 3.0.1」リリース

2001年8月22日

「GCC 3.0.1」が公開された。「GCC

3.0」からの主な変更点は以下のとおり。

- ・C++での誤ったコード生成を修正
- ・標準C++ライブラリのクロスコンパイラサポートを改良
- ・特定の組み込みターゲットで「GCC 2.95.3」が動作し「GCC 3.0」が動作しなかった問題を修正
- ・「exception-handling bugs」を修正
- ・「S/390」アーキテクチャへの移植版を作成

gcc-3.0.1.tar.gzのサイズは約17Mバイト。

[GCC Home Page \(http://gcc.gnu.org/\)](http://gcc.gnu.org/)

OSDNジャパン運用開始

2001年8月21日

VA Linux Systemsジャパンは、米VA Linux Systemsが運営する「OSDN」の日本語版である「OSDNジャパン」の運用を21日から開始した。

米OSDNは、SlashdotやLinux.com、Freshmeat、SourceForgeといった、オープンソース関連サイトを集めたポータルサイト。OSDNジャパンでは、5月末に運用を開始した「スラッシュドット・ジャパン」をはじめとした、海外のオープンソース関連サイトの日本語版開設や、日本国内のオープンソースソフトウェアの情報を、米OSDNに発信するといった活動を行っている。正式な発表はなされていないが、VA Linux Systemsジャパンによると、次に日本語版を開設する予定のサイトはすでに3つに絞り込まれており、秋には新しいサイトの運用を開始する予定。具体的なサイト名は明らかにされなかった。

今後の予定として、国内独自サイトの設立や、必要であれば国内のオープンソースコミュニティとの提携も行う。VA Linux Systemsジャパンのハードウェアビジネスについても、これまでどおり継続するということだ。

また、今回のOSDNジャパン開設にとともに、米OSDNがこれまで行ってきた開発者向けイベントを日本国内でも開催する。第1弾として、9月14日に「Linux

Kernel Conference」を開催し、国内外のカーネル開発者を招いて講義を行う。10月には、「LinuxWorld C & D/TOKYO 2001」(10月24、25日)会場内で「Debian Conference」を開催するそうだ。

[OSDNジャパン \(http://osdn.jp/\)](http://osdn.jp/)

[VA Linux Systemsジャパン](http://www.valinux.co.jp/)

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

ボーランド、DelphiとKylixのダウンロードを8月24日より開始

2001年8月21日

ボーランドは、8月24日より「Delphi 6 Personal日本語版」および「Kylix Open Edition日本語版」の無償ダウンロードを開始すると発表した。それぞれのダウンロードサイトは以下のとおり。

「Delphi 6 Personal日本語版」(<http://www.borland.co.jp/delphi/personal/>)

「Kylix Open Edition日本語版」(<http://www.borland.co.jp/kylix/openedition/>)

ボーランドでは同時に、「Delphi 6 & Kylix Startup Kit」送付サービスを開始する。先着2万人に、「Delphi 6 Personal」、「Kylix Open Edition」、「Delphi 6 Enterprise 60日トライアル版」、「Kylix Server Developer 60日トライアル版」の収録されたキットを送付する。申し込みは8月24日からで、同社Webサイトで受け付ける。

[ボーランド \(http://www.borland.co.jp/\)](http://www.borland.co.jp/)

テンアートと東芝エンジニアリング、大学、専門学校向けLinux技術者養成プログラムを発表

2001年8月21日

テンアートと東芝エンジニアリングは、レッドハットの協力を得て、学校向けLinux技術者養成プログラム「Linuxアカデミープログラムbased on Red Hat」を開始することを発表した。

このプログラムはテキスト、カリキュラム、講師向けトレーニングおよび技術者資格試験を含む総合的なプログラムで、大学、専門学校向けに提供される。テン

アートと東芝エンジニアリングは共同で、このプログラムの企画、開発、運営、カリキュラム/テキスト開発を行う。テキストおよびカリキュラムはレッドハットの認定により、レッドハットの教育プログラムを受講したのと同等に扱われるため、RHCE認定試験の受験資格を得ることが可能だ。このプログラムにより、学校教育側は実践的なLinux教育を学生に提供することが可能になり、産業界側にとっては即戦力となるLinux技術者を育成することができるという。

テンアートニ

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

東芝エンジニアリング

(<http://www.toshiba-eng.co.jp/>)

シャープ、ARMコアのシステムLSIに最適化した組み込みLinux「axLinux」を開発

2001年8月21日

シャープは21日、携帯情報端末などのモバイル機器向けのARMコアシステムLSIに最適化したLinuxベースの組み込み用OS「axLinux」を開発したと発表した。これを基に同社は、モバイル機器向けに開発環境を含むトータルソリューションを提供する。

axLinuxは、Linuxをベースに、動画再生向けにスケジューリングをチューニングしたほか、インターネット接続機能を搭載し、組み込み用に1Mバイト以下のサイズにしたOS。アプリケーションの開発環境として、CPUソフト/エミュレータ、視覚的デバッグツールなども提供する。これにより機器メーカーがソフトウェアの開発期間の短縮やコストの低減を図ることができるとしている。

同社は6月26日に32ビットARMコアシステムLSI「LH79531 / LH79532 / LH79533」を発表しており、このLSIとソフトウェアのサポートをビジネスパートナーのアクセスと共同で行っていく。また、アクセスが開発中のオープンソースでフリーの組み込みGUI環境「式神(しきがみ)」を今後、ミドルウェアとしてアクセスから提供するとしている。

シャープ (<http://www.sharp.co.jp/>)

アクセス (<http://www.axe-inc.co.jp/>)

米IBM、大学生のLinux技術コンテストを開催

2001年8月15日

米IBMは、世界中の大学生、大学院生を対象としたLinuxの技術コンテスト「Linux Scholar Challenge」を開催すると発表した。18歳以上で大学または大学院に在籍していれば、誰でも参加することができる。

このコンテストのルールは、学生は21の与えられたテーマから1つを選ぶか、自分で設定したテーマに従い、英文で1200語以内のレポートを作成、提出するというもの。

参加には事前登録が必要。登録期間は9月15日までで、米IBMのホームページで登録を受け付けている。レポートは11月30日に締め切れ、12月14日に受賞者が発表される。採点はIBM基礎研究所の技術者などがあたり、「創造性」、「思慮深さ」、「効果」の3点を審査する。

上位25名には「IBM ThinkPad T Series」が贈られ、さらに上位3名は2002年夏にLTC (IBM Linux Technology Center) へのインターンシップに招待される。また、10名以上の学生が登録し、平均点がもっとも高い大学には、「IBM eServer xSeries」のLinuxクラスが「IBM eServer zSeries」のLinuxサーバが贈られる。

米IBM (<http://www.ibm.com/>)

「Linux Scholar Challenge」のページ

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

「レーザーファイブドットネット」誕生

2001年8月7日

Linux教育事業を行ってきたドットネットは7日、レーザーファイブとの提携関係を強化し、社名をレーザーファイブドットネットとすると同時に、レーザーファイブ代表取締役CEOの窪田敏之氏が同社会長に就任したことを発表した。

ドットネットでは、これまで主にネットワーク技術者やJavaプログラマーを養成する講座を開講してきたほか、「LASER5 Linux」を教材とした「LPI」

資格取得講座などのLinux関連の講座も行ってきた。

今回の提携強化により、レーザーファイブとレーザーファイブドットネットは、「LASER5 Linux」の開発および販売から、教材の開発、講師派遣、技術者育成といった教育事業までを連携してトータルにカバーすることが可能になる。

レーザーファイブドットネット

(<http://www.laser5.net/index.htm>)

ぶらっとホームと日本アイ・ピー・エムがLinuxソリューションで提携

2001年8月7日

ぶらっとホームと日本アイ・ピー・エムは7日、「IBM eServer iSeries 400」向けのLinuxソリューションに関する提携を発表した。今回の提携の枠組みは以下のとおり。

- ・Linuxサーバを統合できる「OS/400 V5R1」向けのLinuxシステムの導入、運用、保守をぶらっとホームが提供
- ・「IBM eServer iSeries 400」向けのLinuxアプリケーションをぶらっとホームが開発
- ・日本アイ・ピー・エムのビジネスパートナーと協業して、顧客ニーズにあったシステムを開発

これにより、顧客は既存の「IBM eServer iSeries 400」の基幹システムと連携したLinuxソリューションを容易に構築することが可能になるという。また、両社ではLinuxアプリケーションの移植および新規開発をサポートする「マイグレーションサポートセンター」の設立や、サポートメニューの充実などをし、ハードウェアベンダーやSI企業への支援も行ってゆく。ぶらっとホームは、独立系のソフト/ハードウェアメーカーやディストリビュータとの協業および顧客対応のための窓口を新設するとしている。

ぶらっとホーム

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

日本アイ・ピー・エム

(<http://www-6.ibm.com/jp/>)

Hardware

省電力Linuxワンボードサーバ
L-Board rev.1

発売日

2001年7月26日

URL <http://www.laser5.co.jp/>

レーザーファイブは、超省電力Linuxワンボードサーバ「L-Board rev.1」を7月26日から発売した。

L-Boardは、CPUにNEC VR4122を採用、64ビットレジスタ、180MHzクロック周波数、32ビット外部バスを持ち、2.2Wの消費電力で稼動する。他の主な仕様は、メモリ32Mバイト、FLASH ROM 16Mバイト、100Base-TX×2、シリアルインターフェイス×2、UltraATA/66×2（ATAPIデ

バイス4台まで拡張可能）。PCサーバ内に組み込んでサーバinサーバとして活用できる。バックアップ電源として外部にスイッチングタイプのACアダプタも併用できる。

主な用途は、100Mbps光インターネット用アドレス変換ルータ、ワンボードファイアウォール、DNS等の軽量サーバ、100Base-TXの高速ルータなど、各種組み込み機器として利用できる。



Hardware

HDDを最大4台搭載可能な1Uラックマウントサーバ
Vintage 1U Rackmount Server Type J

発売日

2001年9月上旬

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

日本コンピューティングシステムは、JCS Rackmount Solutionsの新製品として、4基のホットスワップ可能なハードディスクベイを持つ1U Rackmount Serverシリーズ Type Jを9月上旬より発売する。

製品は、SMP対応のマザーボード(Dual CPU対応)を搭載し、同社が独自に設計、製造したケースを採用。サイズは、483(W)×605(D)×45(H)mm。オプションのRAIDカードを搭載することで、RAIDレベル5(3台のハードディスクを使用)で

ホットスタンバイ用のハードディスク(1台)を備えた可用性に優れた1Uサーバシステムを構築できるようにしてある。

主な仕様は、CPU Pentium 1GHz 2基搭載、メモリ128Mバイト(最大4Gバイト)、Ultra160 SCSI HDD 18.4Gバイト(最大293.4Gバイト=73.4GバイトHDD×4)、24倍速CD-ROM、100Base-TX×2、シリアルポート×1、パラレルポート×1、USB×2。



Hardware

e-コマースサイト構築システム
ECOSS Solution OnePack

発売日

2001年8月8日

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

エコスは、NTTソフトウェアと共同開発してきたe-コマース構築システム「ECOSS Solution OnePack」を8月8日より発売した。

ECOSS Solution OnePackは、Web、アプリケーション、データベースの3台のサーバと関連ネットワーク機器で構成されるハードウェアと「ECOSS Solution Enterprise」Linux版からなるe-コマースサイト構築のためのオールインワンシステム。

OSにはLinuxを採用し、コストを低減している。商品購入、決済、物流などのe-コマース業務に関わるすべてのステータスをシステムが自動的に管理し、一元的に運用する。運用管理はWeb上で実行可能。物流モジュールにはNTTソフトウェアが開発した「Logiway」を実装し、複数の物流会社との連携を実現した。中小規模の事業者を対象に1システム1550万円で販売されている。



Hardware

モバイル対応のグループウェアサーバ
Miracle Linux INSUITEサーバモデル

発売日

2001年8月末

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

コンパックコンピュータは、サーバにLinuxをブリーインストールした「Miracle Linux INSUITEサーバモデル」を8月末より発売した。

Miracle Linux INSUITEサーバモデルは、コンパックの「Compaq ProLiant ML330e」をベースに、OSに「Miracle Linux Standard Edition V1.1」とドリーム・アーツのモバイル対応グループウェア「INSUITE」をブリーインストールしたソリューションパッケージ。

INSUITEは、コンパックのPDA「iPAQ Pocket

PC」やiモードに対応しており、モバイル環境からもオフィス同様に利用できるナレッジマネジメントをサポートしたWebベースのグループウェア。

今回発売するモデルは、CPU Pentium 800MHz、メモリ320Mバイト、UltraATA/100 HDD 20Gバイト。

サーバにLinuxをブリーインストールして販売することで運用管理面でのユーザーの負担を軽減し、Linuxの専門知識がなくても活用できる。



Hardware

2Uラックマウントサーバ
Express5800 / 120Rc-2 他

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

NECは、最新Pentiumプロセッサを搭載した2Uラックマウントサーバ「Express5800/120Rc-2」を8月7日から発売した。

主な仕様は、CPU Pentium®-S 1.26GHz（最大2個搭載）メモリ128Mバイト（最大4Gバイト）、さらにUltra160 SCSI対応73.2GバイトHDD（ホットプラグ対応）を最大5基まで搭載可能にして、内蔵ディスク容量を現行機種との2倍に拡大した。

同時に「Miracle Linux Standard Edition 1.1」

発売日

2001年8月7日

発売	日本電気株式会社
TEL	03-3455-5800
価格	55万2000円～

や、クラスミドルウェア「CLUSTERPRO for Linux」、2Uラックマウントサイズのデータストレージ「ST12000」も販売する。

価格は120Rc-2が55万2000円から。ST12000が120万円。Miracle Linux Standard Edition 1.1が16万5600円。CLUSTERPRO for Linuxが80万円（2ノード用）、Miracle Linuxの価格にはインストール代行サービス、問い合わせサービス、運用管理ツールなどが含まれる。



Hardware

OS付きインターネット専用パソコン
Web COM

URL <http://www.logicaleffect.com/>

ロジカルイフェクトは、Linux OSをプリインストールした、インターネット専用PC「Web COM」のフルモデルチェンジを行い、8月3日より発売した。

主な変更点は、OSをVine Linux 2.1から、Vine Linux 2.1.5にアップグレード。全パーツを見直し、再構成、パフォーマンスを向上。内蔵モデム搭載モデル（Net / 56K-H）の追加。ブラウザ、メールを標準搭載、モデム搭載モデルで

発売日

2001年8月3日

発売	ロジカルイフェクト株式会社
TEL	
価格	3万9700円～

はインターネット設定済み（同社「びゅんびゅんネット」対応）にて出荷。テレビ出力機能搭載モデル（Net TV QCP）の追加。ビデオ端子を用いることでパソコンモニタがなくても家庭用テレビを用いてインターネット利用が可能になる。

主な仕様は、CPUがCeleron 766MHz、メモリ64Mバイト（最大512Mバイト）、UltraATA/100対応10GバイトHDD。



Hardware

Itaniumプロセッサを搭載した次世代サーバ
hp server rx4610

URL <http://www.jpn.hp.com/go/computing/>

日本ヒューレット・パカードは、Itaniumプロセッサ（733MHzまたは800MHz）を最大4個搭載できるサーバ「hp server rx4610」を8月1日より発売した。

rx4610は、Intelの64ビットItaniumプロセッサ対応エントリーサーバ。主な仕様は、CPUに64ビットプロセッサItanium（733MHz / 800MHz）2個（最大4個）、メモリ1Gバイト（最大64Gバ

発売日

2001年8月1日

発売	日本ヒューレット・パカード株式会社
TEL	0120-081565
価格	679万円～

ト）、HDD 72Gバイト（36Gバイト×2）、拡張PCIスロット10基（64ビット / 66MHz・ホットプラグ対応×8、32ビット / 33MHz×2）

今回発売されるモデルに搭載されるOSは、同社の「hp-ux11iバージョン1.5」だが、今後Red Hat Linuxにも対応する予定。

価格は、Itanium 733MHz搭載マシンが、679万円、Itanium 800MHz搭載マシンが867万円。



Hardware

オールインワン・ブロードバンドルータ
Accton broadDirect Series -SMC7004ABR

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

アクトンテクノロジーは、ADSL、ケーブルTVインターネットに対応するブロードバンドルータ Accton broadDirectシリーズの新製品を8月20日より発売した。

新製品の主な仕様は、WANポート×1、LANポート×4、プリンタポート×1。LANポートは10 / 100Mbps N-Way対応スイッチングハブと同じ機能を持ち、MDI / MDI-Xの自動切り換え機能があるため、ハブなどの追加にもクロスケーブルを必

発売日

2001年8月20日

発売	アクトンテクノロジー株式会社
TEL	
価格	1万9800円

要としない。LANポートを利用して、LAN側の複数のパソコンから同時にインターネットを利用することができ、プリンタポートにプリンタを接続すればLAN上のパソコンに対してプリンタサーバとしても利用可能。

IPマスカレード機能、DoSプロテクション機能、ハッキングロギング機能などがあり、常時接続による外部からの不正アクセスを防ぎ、不正アクセスログの参照ができる。ルータの設定は、ブラウザから行う。



Software

日本語全文検索ソフトウェア
MitakeSearch V3.0URL <http://www.compaq.co.jp/>

コンパックコンピュータは、日本語全文検索ソフト「MitakeSearch」の新バージョン3.0を8月6日より発売した。

MitakeSearchは、イントラネット/インターネットなどに対応しMicrosoft Office製品、PDF、XMLなどの文書の検索が可能。多数の同時アクセス、大量の文書にも対処でき、Linux、Windows NT / 2000、Tru64 UNIXで稼動する。

発売日

2001年8月6日

発売	コンパックコンピュータ株式会社
TEL	0120-018589
価格	9万円～

V3.0では、扱えるファイルの上限を10万 / 10万に設定したモデルに加えて、500万ファイルまで扱えるエンタープライズモデルを新たに提供。64ビットマシンの性能をフルに活用するVLMを生かし、大量の文書の高速度検索を可能にした。さらに、携帯電話用HDML / CHTML対応の検索CGIとインデクサー・オプションを用意しており、iモード、J-SKY、EZwebでの利用も可能になった。



Software

Webアクセスマネジメントソフトウェア
InterScan WebManager Linux版URL <http://www.trendmicro.co.jp/>

トレンドマイクロは、安全で効率的な企業のインターネット環境を提供するWebアクセスマネジメントソフトウェア「InterScan WebManager」を、8月20日より発売した。

InterScan WebManagerは、有害サイト、業務効率を低下させると考えられるWebサイトへのアクセスを禁止し、安全で効率的なインターネット環境を提供するソフトウェア。

発売日

2001年8月20日

発売	トレンドマイクロ株式会社
TEL	03-5334-3650
価格	23万8000円(30アカウント)～

規制対象となるWebサイトをデータベース(規制リスト)に登録してサイトへのアクセスをブロックする。データベースは2001年8月現在で17万件のデータ、28のカテゴリーがある。企業はポリシーに沿ってカテゴリーを選択する。所属グループや時間帯ごとに異なる設定もできるため柔軟な規制ができる。データベースの更新は土日祝日を除く毎日行われている。



Software

販売 / 会計 / 給与業務を統合したWebアプリケーション
clear worksURL <http://www.clearworks.co.jp/>

ミロクドットコムは、Linuxを利用した販売、会計、給与業務を統合したWebアプリケーション「clear works」を9月1日より発売した。

clear worksは、会計を核として販売、給与を統合したオープンソース技術を駆使したWebアプリケーション。

主な特徴は、販売、会計、給与業務用アプリケーションを統合型のWebアプリケーションとして構築してデータを相互に連携し、業務を効率化

発売日

2001年9月1日

発売	株式会社ミロクドットコム
TEL	
価格	12万円(3クライアント)～

する。OSにLinux、データベースとしてPostgreSQL、アプリケーションサーバにPHPというオープンソース技術を活用した製品。clear worksをインストールしたサーバを置くだけでアプリケーションの設定は完了。ソフトのインストールやネットワーク設定などの煩雑な手間が省ける。

税理士や公認会計士によるclear worksの操作方法指導を含むコンサルティングサポート「clear works Accountingパートナー制度」がある。



Software

Sun Cobalt専用ファイアウォールソフトウェア
FireWall+URL <http://www.netman.co.jp/>

ネットマンは、Sun Cobalt Qube / RaQサーバプライアンス対応のファイアウォールソフトウェア「FireWall+」を9月1日より発売した。

FireWall+は、Qube / RaQの専用パッケージのため、インストールが簡単に行える。セキュリティの設定には、ボタン1つでできる自動設定モードと、設定の変更が可能な手動設定モードの2種類がサポートされている。

Qube / RaQの状態は毎日診断され、診断結果は

2001年9月1日

発売	株式会社ネットマン
TEL	03-3523-5100
価格	オープンプライス

メールで通知される。送信日時は自由に設定可能で、システムの資源監視(CPU、メモリ、HDDなど)、サービスの状態監視(Webサービス、SMTPサービスなど)、セキュリティ監視(不正アクセス、システム改ざんなど)、外部からのサーバ監視(サーバやポートの監視)を診断する。

サービスダウンなどの不具合については、電源インジケータが点滅して知らせるほか、背面のLCDによってエラーメッセージが表示される。



Software

発売日 2001年9月上旬

CD-ROM上で動作する統合インターネットサーバソフト
EASY NET SERVER

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

プリンストンテクノロジーは、CD-ROMで動作するサーバソフト「EASY NET SERVER」を9月上旬から発売する。

EASY NET SERVERは、CD-ROMから直接起動して、画面の支持に従って設定項目を入力するだけでサーバを構築できるLinuxをベースにした統合型のインターネットサーバソフトウェア。これ1本でDNSサーバ、Webサーバ、メールサーバ、FTPサーバ、DHCPサーバなどの構築が可能。

発売	プリンストンテクノロジー株式会社
TEL	03-3863-7131
価格	3万8000円

CD-ROMから直接起動、動作するためインストールが不要。そのためコンピュータウイルスや事故によってプログラムがダメージを受けることがないという特徴がある。

必要なシステム構成は、CPU Pentium以上、チップセットIntel 440BX以前を推奨、LANアダプタ、メモリ64Mバイト以上、HDD 1Gバイト以上、LANに接続可能でInternet Explorer 5.0以上が動作可能なPC。



Software

発売日 2001年9月19日

Windows上でLinuxを使用
Virtual PC for Windows 日本語版

URL <http://www.ascii.co.jp/netmedia/>

アスキーは、米Connectix Corporation社の新製品「Virtual PC for Windows 日本語版」を9月19日より発売する。

Virtual PC for Windows 日本語版は、Windows 2000 / NT 4.x、Windows Meが起動しているパソコン上で、Windows 95 / 98 / Me、Windows 2000 / NT、Linux、MS-DOSなどのOSを1つのアプリケーションとして起動するソフトウェア。1台のパソコンで異なるOS上で動作するアプリケーション

発売	株式会社アスキー
TEL	
価格	4万2800円

を同時に使いたい時に役立つ。

インストールと設定は、自動的にホストOSのシステムの設定を検出し、その設定をゲストOSに反映するので、ゲストOS用にネットワークやプリンタの設定をする必要はない。さらに、Drive Container機能により、ハードディスクのパーティションの再編成も不要。アプリケーションを切り替えるように、OSが切り替えられるため、システムを再起動せずに複数のOSを扱える。



Software

発売日 2001年10月上旬

Javaアプリケーション解析のための総合ツールキット
JProbe ServerSide Suite 3.0J

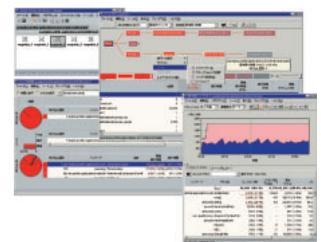
URL <http://www.boc.co.jp/>

文化オリエントは、J2EEベースのサーバサイドJavaアプリケーションのためのパフォーマンス解析ツール「JProbe ServerSide Suite 3.0J」を10月上旬に発売する。

JProbe ServerSide Suite 3.0Jは、アプリケーションの実行時間とメモリの使用量を9つの測定基準で分析、非効率なアルゴリズム、メソッドの過剰呼び出しなどを素早く特定し、メモリリークや

発売	文化オリエント株式会社
TEL	022-373-0360
価格	オープンプライス

パフォーマンスボトルネックの検出を行うJProbe Profiler、Javaプログラムのスレッドを分析しスレッドの機能停止や競合を検出するJProbe Threadalyzer、テスト済み、および未テストのコードを検出報告し、Java開発における効率的なテストを可能にするテスト支援ツールJProbe Coverageの3つのツールで構成される。Javaプログラムを的確に診断し開発の効率化を助ける。



Software

発売日 2001年10月上旬

Javaソフト開発用グラフ&グリッドコンポーネント
JClass Chart 5.0J & JClass LiveTable 5.0J

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

文化オリエントは、JavaコンポーネントJClassシリーズから、2つの製品「JClass Chart 5.0J」と「JClass LiveTable 5.0J」をバージョンアップ、10月上旬から発売する。

JClass Chart 5.0Jは、優れた操作性と高速な描画速度で、リアルタイム処理や、データ解析用グラフを作成する。10種類の基本グラフをもとに、同時に複数のグラフを表示できる。マウスやキー

発売	文化オリエント株式会社
TEL	022-373-0360
価格	12万8000円～

による画像操作やグラフカスタマイズ機能がある。

JClass LiveTable 5.0Jは、セルに多彩な拡張性を持たせた表を作成する。セルの描画と編集に特化したオブジェクトで、文字列のワードラップ、ブール値のチェックボックス、イメージ表示などが多彩になった。さらに印刷・プレビュー機能、20種類におよぶ表計算関数、XMLをサポートしている。



日刊アスキー Linux Business Report



clear worksに見るオープンソースの成功例

<http://www.linux24.com/>

8月6日、ミロク情報システムの子会社であるミロクドットコムは、販売、会計、給与の処理機能を統合したWebアプリケーションソフトウェア「clear works」を発表した。このソフトウェアは、Linux上で動作し、データベースにPostgreSQLを、アプリケーションサーバにはPHPを使用するなど、まさにオープンソースの土台の上に構築された製品となっている。PostgreSQLとPHPで構築されたWebアプリケーションが、基幹系のパッケージとして実現したということは注目度が高い。今回は、ミロクドットコム（ブランド名は「m works」）の開発部長・大谷明氏と、チーフディレクター、営業担当・吉岡賢司氏にお話を聞いた。

clear works誕生の経緯

ミロクドットコムは、税理士・公認会計士事務所とその顧問先企業向けの業務用アプリケーション開発や経営情報サービスなどを手がけ、全国の会計事務所の4分の1と取り引きがあるミロク情報サービスの子会社だ。

ミロク情報サービスは業務ソフトを作成しているが、時代とともに構築するシステムがハイエンド化して高価になってきていた。こうした時期に、中小企業を対象とした、Windows上で動作する会計パッケージソフトが登場し、シェアを伸ばしていく。こうした会計パッケージソフトが広まるうちに、会計事務所側が使用しているミロク情報サービス製ソフトとのデータ互換など

の問題が浮上してきた。

そこで、エンドユーザー層に使われるソフトウェア作成というミッションをもって、ミロクドットコムの設立とclear worksの開発が進められることになる。

次にくるのはWebアプリ
その次はコスト競争だ！

こうした、「中小企業に受け入れられる業務ソフトウェア作成」というミッションに対し、大谷氏の出した答えが、Webアプリケーションであり、オープンソースであった。「いまさらWindowsでハコ売りもないだろうと悩んだ結果、MS-DOSがあってWindowsがあって、その次はなんだろうと考えた。時代が変わるところがマーケットシェアが変わるところだ。次に来るのはWebアプリケーションだろう。誰に聞いてもそういう答えが返ってきた」（大谷氏）

ただ、中小向けのWebアプリケーションで大成功しているのはサイボウズだけといってもよい状況だ。グループウェアならばまだしも、業務用はまだまだWindowsだという意見もあったそう。しかしそこはあえてWebアプリケーションで打って出て、いわば「待ち駒」をしようというプランにしたという。

また、Webアプリケーション化には、中小企業と会計事務所の変化も影響している。「中小企業ではどうしても会計などの面倒なことを後回しにしてしまう傾向があるが、ITを導入すれば経営

のスピードを速くできるという利点がある。また、会計事務所側も、規制緩和で法人化され、競争にさらされる。効率化は必須だし、2003年からは申告をデータで送る電子申告も開始される。こうした流れから、会計情報、売り上げ情報、業務情報を集中管理し、会計事務所と企業の情報スピードも上げねばならない。Webアプリケーションであれば、企業が会計事務所に自社システム（この場合自社のclear works）へのIDを渡してしまえばそれで済む。会計事務所側も企業を訪問する必要がなくなる」（吉岡氏）

そして、製品のコストを抑えるには、オープンソースの利用が欠かせなかった。大谷氏が考えるWebアプリケーションの形態は主に3つある。

- (1) Microsoft路線……Microsoft製品で固めた、ASPなどで展開する路線
- (2) Microsoft以外だが、ソフトウェアにコストがかかってもかまわないというもの。たとえば、Linux + Java + Oracleなど
- (3) オールオープンソースで低コストを狙うもの



ビジュアルにも凝ったというclear worksのスタート画面

1と2は、支払う先は違えどもソフトウェアのライセンス料金は発生する。だが、3はハードウェアや人件費を除けば、ある意味タダである。

「Webアプリケーションは、最初の段階では移植しただけで『がんばったね』ともてはやされるかもしれないが、そのあとはコスト競争になるだろう。そのときに、オープンソースならばハードウェアと僕らのコストだけで、OSやデータベースのコストがそっくりなくなる。最後にコスト競争になったときに、3つの路線で有利なのはオープンソース」(大谷氏)。

オープンソースにおける人材確保

オープンソースの強みはわかるが、現在オープンソースやLinuxに強い技術者は少ないのが現状だ。ミロクドットコムの場合、clear worksの開発は外部のフリーランスに協力を要請したという。

「3年前から現在のフリーランススタッフの協力による開発チームのビジョンはあった。ピザパーティや飲み会に参加し、コミュニケーションをとった。誰が何に詳しいというリストが頭の中でできあがっていた。今回もそうした方々にお願いした」(大谷氏)。

親会社のミロク情報サービスの社員数は700人以上。3年の時間があるのならば、社員を教育してもよかったのではないだろうか。「社員を多く抱えてコスト競争を行うのは不利。できたら10人以下の社員で、あとはその時の仕事にベストな外部スタッフの協力を仰ぐほうがいい」(大谷氏)。3年前からの信頼関係の蓄積こそが、オープンソースビジネスに必要なことといえそう。

大谷氏は、ベンダーのソフトウェアを使う路線を選択するのであれば、お

金を払えばうまくいくだろうが、オープンソースということになると、今はコミュニケーションをベースとした活動しかないと思う、という。

「ベンダーソフトウェア使用の路線は(どこの部品メーカーがいいとか、誰がどのようなリソースを持っていて優位であるといったような)すでに仕組みができあがっている。ところがオープンソースは、誰に聞いたらいいかわからない。オープンソース路線でのベスト1、ベスト2はまだ決まっていない。だからまだベスト1になれる可能性が残っている。我々はできたら業務アプリケーションのベスト1になりたい。」(大谷氏)。

celar worksの今後の戦略

(1) リソースに則ったビジネス

ところで、Webアプリケーションと聞いて思い浮かぶのが、ASPとしての展開だ。だが、clear worksは今のところ、ASP形態での展開は行っていない。これは主に、業務システムASP展開の難しさと、親会社であるミロク情報サービスの持つリソースによるところが大きい。

業務アプリケーションのASP展開は、現状では難しいという事情がある。売り上げその他の基幹データを自社外のサーバで管理するのに抵抗を覚えるのは当然の心理だろう。そこで登場するのが、親会社のミロク情報サービスが持つ、会計事務所のネットワークだ。

「我々が考えているASPIは、ひとつはパッケージとしてサーバごと売る方式。もうひとつは、会計事務所を意識した展開だ。会計事務所がASP事業を行えば、すでに顧問先を獲得しており顧客の数が見えるため、どの程度のコストをかけたらいいいのか、費用対効果

が見える。単に一般に開放するだけのASPでは、顧客の数が見えない。また、企業にとっても、会計事務所の先生が、『やっても大丈夫』と言ったシステムであれば安心だ。大型会計事務所のWebサーバを我々が構築したり、中型会計事務所向けにはミロクが構築したデータセンターをASPとして利用してもらおう。こうすることで会計事務所も業務スピードを上げ、自由競争で戦う道具にさせていただくことができる」(吉岡氏)。

「自分たちでASPをやらなくても、8000の会計事務所がある。そういうところがASPをやりたいと言ったら、どうぞやってくださいと。親会社の大切なお客様がやりたいと言っているのだから、我々はソフトと情報を供給すればいい。すでにそういう話 coming」(大谷氏)。

コミュニケーションを基盤としたフリーランス技術者による開発と、業界の展開や自社リソースを考慮した戦略、そして小規模人員による機動的なソフトウェア開発。企業活動ありきでもなければ、「開発者が作りたいモノ」ありきでもないclear worksの実現は、今後のオープンソースビジネスに貴重な成功事例をもたらしたものだといえるだろう。



ミロクドットコム開発部長・大谷 明氏(左)と、チーフディレクター、営業担当・吉岡賢司氏(右)

Caldera Forum 2001

August 19-22, 2001 Santa Cruz

8月20日から22日まで、米国サンタクルーズ市でCaldera社主催のForum 2001が開催された。このイベントは、本来SCO (Santa Cruz Operation) 社が同社のOSユーザーやリセラー向けに開催していたイベントだが、SCOからUNIX関連ビジネスがCaldera社へ移ったため、今回はCaldera社の主催となった。



文：塩田紳二
Text : Shinji Shioda

CalderaとSCO

商用のUNIXであるOpenUNIXで知られたSCO社は、インテル系CPU用UNIXのライセンスを保持していた。これは、UNIXの元祖AT&TからNovell社などを経て、SCO社に移ったものだ。

SCOは、Microsoftが移植したIntel CPU系のUNIXであるXenixに起源を發するOpenServerと呼ばれる製品と、AT&TオリジナルのUNIXである、OpenUnixが主たる製品ラインだった。

そのSCO社のUNIX OSビジネスは、Linuxディストリビューターのひとつ、Caldera社に販売され、OpenServer、OpenUnixとUNIXに関連する知的財産は、Calderaが持つことになった。そのため、今回のイベントはCaldera主催となったのである。

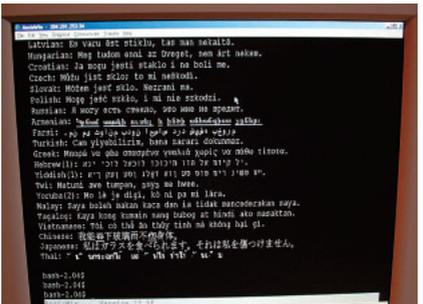
会場は大学

Forumの会場はカルフォルニア大学サンタクルーズ校である。夏休みで学生がいない教室や講堂などを使ってコンファレンスが行われる。なお、希望すれば、学内にある宿泊施設(ただし相部屋)を利用できるのだとか。米国人は大学時代を思い出していいのかも。また、Forumの参加証(パス)には、今年で何回目の出席かという数が書かれており、全体としては、大学の学園祭という感じである。最近のLinuxWorldなどに比べると、規模も小さいが、その分、かなり親しみやすい雰囲気である。

さて、こういうイベントにつきものの、基調講演 (keynote speech) の会場も変わっている。屋外なのである。この大学にある、崖のようなところにある会場で朝から基調講演が行われる。



キーノートスピーチは、屋外の会場で行われる。朝8時30分からなのだが、Santa Cruzは海沿いなので、結構寒い。



Unicodeで各国語に対応したWindows用のtelnetプログラムAnzio Win. Rasmussen Softwareの製品。SSHにも対応しているという。

Santa Cruzは海辺の都市で、太平洋の寒流のため朝夕はかなり寒い。真夏だということに、セーターを着て、さらにその上に一枚羽織らないといけないうくらいだ(昼間は暑いのだが)。

この基調講演も、ただ会社の役員や招待話者が話すのではなく、前後に、曲芸や手品などの出し物(いちおうプロが来ているようである)が入る。このあたりもなんやら、学園祭の演芸会という雰囲気だ。

Calderaの製品とロードマップ

現在のCaldera社の製品は、LinuxディストリビューションのOpenLinux (Server版とDesktop版がある)とインテル系CPU用のUNIXであるOpenUnix、および、OpenServerである。

このOpenUnixには、LKP (Linux Kernel Personality) と呼ばれる機能があり、カーネルより上のレベルで



IBMもサーバなどを展示。ここだけ、普通のイベントのようにブースが作られていて、ちょっと浮いた感じもする。

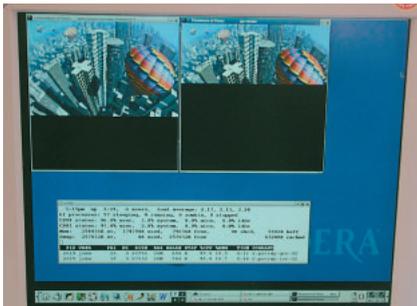
Linuxとの互換性がある。つまり、Linux用アプリケーションであれば、OpenLinuxでもOpenUnixでも動作させることができるわけだ。

Linuxもマルチプロセッサなどのサポートが充実してきたが、すでに10年以上前にマルチプロセッサのサポートを行っているUNIXカーネルに比べるとかなりの差がある。また、高負荷時の安定性などにもUNIXカーネルは優れた機能を持つ。

今年の第4四半期には、OpenLinux 3.1.1とIA-64対応のOpenLinux64 (First Release) を出す予定。Calderaは、IA-64版のLinux開発で、UNIXカーネルに使われている技術の一部を提供している。

OpenUnixは、LKPを持つOpen Unix8がリリース中。今後、OpenUnixについては、カーネルを開発者などに公開するとのこと。ただし、最初は、Read Only、つまり見ることはできるが、改造して使うことはできない。その後、体制を整えて、ユーザーや開発者が行った改良をフィードバックするようにするという。カーネルはGPLにはならないが、なんらかの方法でオープンソース化することを考えているとのこと。

また、UNIXのツールは、最終的にはGPLで公開するという。その手始めとして、awkとgrepがGPL化する。



インテルも別室でIntel Compilerの展示を行う。これは、作成されたプログラムの実行速度をgccと比較しているもの。

このほか、サーバ製品としてVISIR Messaging Serverが今年10月にリリースされる。これは、SMTP、POP3、IMAP4、LDAP、SSLやWebベースメールクライアントなどを統合したパッケージ。

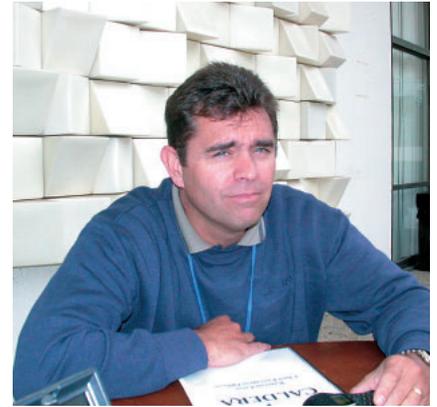
Calderaの戦略

今回の取材で、Caldera社CEOのRansom Love氏に同社のビジネスについて話を聞いた。

まず、SCOのUnixビジネスの買収だが、Calderaがビジネスを成功させるためには、体制を強化し、ユーザーを多数確保する必要があった。しかし、それをこれから始めたのでは時間がかかりすぎる。そこで、SCOとの交渉を開始、両社のビジネスを統合することで、より大きく発展できる可能性があることを提示したという。

現在の状況では、従来のUNIXは技術的には安定していても、世間の話題はLinuxであり、旧SCOのサードパーティとしてもLinux対応に迫られていた。

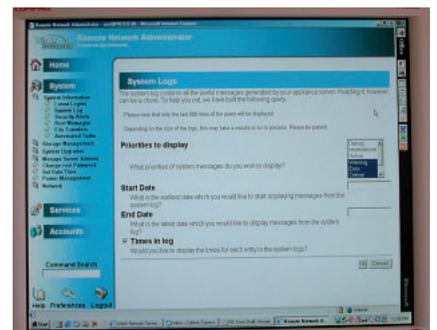
結局、Calderaは、従来のUNIXとLinuxを両方サポートすることで、単なるLinuxディストリビューターから、UNIX、Linuxをサポートするソフトウェアメーカーになったわけである。Linux、UNIXをとともにサポートしているのはIBMやHPといったハードメーカーの



Caldera社CEOのRansom Love氏。SCOの買収によりCalderaはプラットフォームを提供する企業になったという。現状、固定ユーザーのいる旧SCO系のビジネスが大きい。今後は、顧客の要望に合わせ、Linuxも提供していくという。

み。しかもUNIXは、各社の専用ハードウェアである。これに対して、Calderaの強みは、インテル系PCのハードウェアをサポートしていることで、ハードウェアアーキテクチャを変更することなくより大きなシステムへ発展可能になるわけだ。

ただ、複数OSを扱うことは、開発コストなどの上昇を招く。そこで、ひとつの方法としてCalderaでは、UDI (Uniform Driver Interface) の推進を考えているという。また、XenixをベースにしているOpenServerについては、LKPの技術を応用して、Open Unix、OpenLinux上で互換性を実現するPortable OpenServerを開発中とのことである。



Compaqの展示会場でデモされていたWireX。セキュリティを高めたDNSやメールサーバをWebブラウザから設定できる。Small Business用とISP向けのバージョンが用意される。

Products

- 28 6基のハードディスクを内蔵可能な2Uサイズラックマウントサーバ
NL Server 2150
- 30 RAIDハードディスクを内蔵した超小型アプライアンスサーバ
SERVER THE BOX

6基のハードディスクを内蔵可能な2Uサイズラックマウントサーバ



NL Server 2150

ブロードバンド化が進むインターネットでは、サーバ側の強化も必要である。NL Server 2100の後継機種であるNL Server 2150は、増大するコンテンツを格納するためにストレージ容量を拡大し、冗長性を強化して信頼性を高めたLinuxサーバである。

製品名 NL Server 2150
価格 32万円～
問い合わせ先 ノーザンライツコンピュータ株式会社
TEL 044-850-2238
<http://www.nlcomputer.com/>

ノーザンライツコンピュータから、2Uサイズラックマウントサーバ「NL Server 2150」(以下NL2150)が発売された。NL Server 2100の後継機種にあたり、ハードディスクドライブベイを4基から6基に増やし、チップセットを440GXからServerSet LEに変更したことで、FSB 133MHzへの対応や64ビットPCIバスの採用などの高性能化が図られている。

CPUはPentium の866MHz～1GHzを最大2基搭載可能。メモリは4スロットのDIMMスロットにECC対応SDRAMを装着することで128Mバ

イト～4Gバイトまで拡張できる。

ハードディスクは、9G～73GバイトのUltra160対応SCSIドライブを最大6台まで装着することができる。73Gバイトのドライブなら430Gバイト(RAID 5の場合350Gバイト)もの大容量を実現した。SCAタイプなのでRAIDを構成することでホットスワップが可能になる。標準ではマザーボードに搭載されているSCSIコントローラSymbios 53C1010を利用して接続するが、オプションでハードウェアRAIDコントローラ搭載のPCIカードを装着することもできる。

前面パネルには、スリムタイプのCD-ROMドライブとフロッピードライブや、データセンターなどでのメンテナンス性を考慮して、シリアルポートとUSBコネクタ、LANやシステムファンなどのステータス表示LEDが用意されている。

拡張スロットは、フルレングスの64ビット66MHz対応PCIスロットを2スロットと、32ビットのロープロファイルPCIスロットを1スロット備えている。電源は、ホットスワップ可能なリダントタイプで、小型ながら300Wと十分な容量を持ったユニ



写真1 ドライブベイ
2Uラックマウントサイズの前面に、6基のハードディスクベイとスリムタイプのフロッピー/CD-ROMが装着される。

ットが2基装着されており、信頼性を向上させている。

出荷時には、Red Hat Linux 6.2JのFTP版をベースにノーザンライツがチューニングしたOSがプリインストールされる。最新のNL-1.5では、カーネルを2.2.19にバージョンアップし、カーネルコンフィグレーションを物理メモリ2Gバイトへの対応やIDEハードディスクのDMA転送を行うように変更している。また、Red Hatが公開しているバグフィックス/セキュリティ対策パッチの適用と、一般ユーザーがroot権限を持つことを防止するために、mount / umount / uum / fld / kon / newvcといったプログラムのsetuidビットを外すというセキュリティ対策を行っている。

NASサーバのベースマシンに最適

NL Serverシリーズは、同社のSOOXi 2001アプライアンスサーバシリーズのベースマシンとしても利用されている。今回は、NL2150をNAS (Network Attached Storage) サーバとして試用してみた。

付属のCD-ROMを使って、SOOXi 2001 Appliance NASサーバをNL2150にインストールすると、ハードディスクが適切なパーティションで分けられ、Linuxと設定管理ツール「SOOXi Esperanto」が一緒にインス

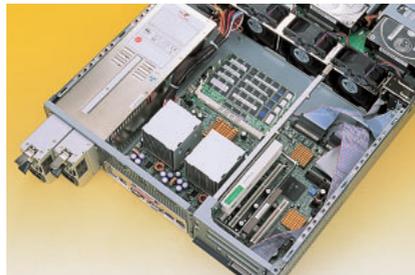


写真2 NL Server 2150の内部
デュアルPentium 1GHzを搭載。300Wのホットスワップ可能な電源を二重化して装備している。

トールされる。Linuxは、カーネル2.4をベースにNASサーバとして必要な機能をコンパクトに最適化したものである。試用したNL2150は、ハードディスクに18Gバイトのドライブを4台装着していた。NL-1.5では、それぞれ別のドライブとしてフォーマットされていたが、NASサーバのインストーラでは、システムが自動認識されドライブ4台を利用するソフトウェアRAID 5にパーティションを設定し直してくれた。

NASサーバでは、AppleTalkによるMacとのファイル共有、Windowsファイルサーバ、HTTP (Web) サーバ、NFSサーバの機能が提供される (画面1)。サーバの設定は、別のマシンのWebブラウザからすべてEsperantoで行うことができる。SSLを使用して通信内容が暗号化されるので、インターネットを通じてリモ



画面1 SOOXi NASサーバとして利用することも可能Webベースの設定ツールEsperantoのメニューから、各種ファイルサーバの設定が行える。



画面2 EsperantoのHDDの容量チェックメニュー
ハードディスクを複数台搭載した場合は、NASサーバのインストーラが自動的にRAID構成でフォーマットする。

ートで設定を行っても安心だ。

デュアルCPUによる高性能とハードディスク6台による大容量の実現、電源の冗長化による高信頼性を備えながら、2Uラックマウントサイズに収めたNL2150は、データセンターでのWebホスト、データサーバ、ファイルサーバなどの用途に向いているだろう。

CPU	Pentium 866MHz ~ 1GHz (デュアル/シングル)
チップセット	ServerWorks ServerSet LE
RAM	128MバイトECC SDRAM (PC133、最大4Gバイト)
ハードディスク	9 ~ 73Gバイト (Ultra160 SCSI、最大6台、ホットスワップ可能)
CD-ROM/DVD-ROM	24倍速CD-ROM (スリムタイプ)
フロッピードライブ	3.5インチ1.44Mバイト
グラフィックス	オンボードATI RAGE XL (8Mバイト、AGP)
ネットワーク	オンボード10/100BASE-T x 2ポート (Intel 82559)
SCSIコントローラ	オンボードSymbios 53C1010 (Ultra160)
インターフェイス	RS-232C x 2、パラレル x 1、USB x 4、PS/2 x 2ポート
PCIスロット	64ビット (66MHz) x 2、32ビット (33MHz、ロープロファイル) x 1
キーボード/マウス	オプション
サイズ (mm) 重量	482.6 (W) x 664 (D) x 88 (H)、約17.8kg
電源	300W (ホットスワップ可能)

表1 NL Server 2150の主な仕様

SERVER THE BOX



24時間稼働のサーバマシンには高い信頼性が要求される。B5サイズほどのコンパクトなケースにハードディスクを2台内蔵し、ミラーリングでデータの安全性を高めた本機は、小規模なオフィスやSOHOで、多目的サーバとして運用するのに最適だ。

製品名	SERVER THE BOX
価格	22万円
問い合わせ先	株式会社ゲメックス TEL 03-3440-9191 http://www.gemx.co.jp/

ゲメックスから、ハードディスクを内蔵したLinuxサーバとしては最小サイズの「SERVER THE BOX」が発売されている。ゲメックスは医療向けの画像取り込み装置やコンピュータシステムなどを企画・設計・販売している企業で、そのノウハウを生かして設計している。



2台のハードディスクを
ミラーリングで利用する

SERVER THE BOXは、24時間運転を行うサーバとしての利用を想定しているため、そこそこのスペックを備え安定したCPUでありながら低消費電力であるCeleron 566MHzを採用している。性能が要求される場合には、オプションでPentium 700MHzに変更することも可能だ。

IDEタイプのRAIDカード、

PROMISE FastTrack100に搭載されているコントローラチップPDC20267をマザーボード上に搭載している。ハードディスクは、3.5インチの40GBバイトIDEドライブを2基内蔵しており、RAID 1(ミラーリング)で使用する。

10/100BASE-TXネットワークインターフェイスを2ポート備えているので、片方をインターネット、もう片方をLANに接続することで、ファイアウォール機能を持たせたインターネットサーバとして使うのもよいだろう。

なお、本機にはフロッピーやCD-ROMのドライブは内蔵されていないため、OSの再インストールやソフトウェアのバージョンアップについて心配されるかと思うが、BIOSレベルでUSBデバイスをサポートしている

ので、ノートPC用の外部接続ドライブとして市販されているUSBタイプのドライブを利用可能だ。

本体のカバーは、ジュラルミン(5000系アルミ合金)前後パネルはステンレスでできていて、美しい仕上げになっている。内部のケースの右側面にはファンがあり、カバーとの間の隙間から放熱している。カバーの四隅を45度にカットした八角形の「オクタゴン・カット・デザイン」は、空気の流れを確保するという意味も持っている。

外形は、190(W)×264(D)×74(H)mmで、B5サイズより少し大きめ、小型サーバの代表的マシンCobalt Qubeと比べても容積は半分以下と非常にコンパクトで、家庭で使う場合でも邪魔にならない大きさだ。消費電力も70VA以下に抑えられている。

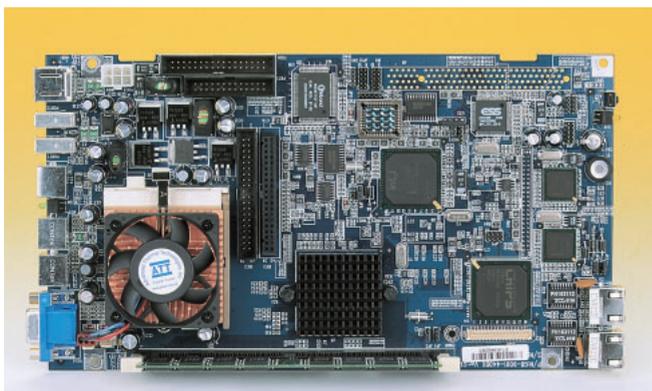


写真1 マザーボードのCPU搭載面
CPUはCeleron 566MHz。チップセットは440BXを採用。メモリスロットはDIMM×1で128MバイトのSRAMを装着。

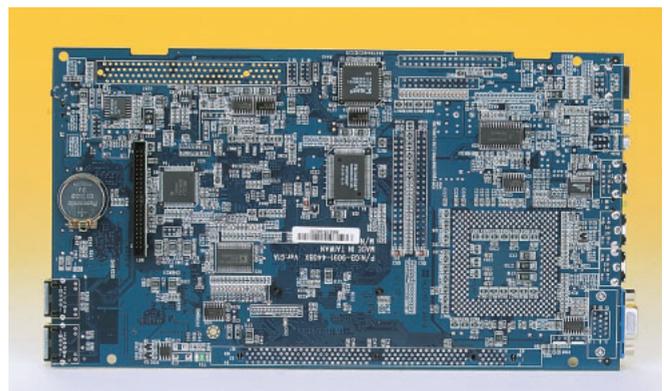


写真2 マザーボードの裏面
専用設計のマザーボードには、高い信頼性を確保できる部品を採用したという。PROMISE FastTrack100をオンボードに搭載している。



画面1 Webminのメインメニュー

Webブラウザから各種システム設定を行うことができるWebmin。英語版だが一部のメニューは日本語で表示できる。



画面2 Webminのサーバ設定メニュー

Web、DNS、DHCP、FTPなどのサーバの設定が可能。インストールされていないソフトのメニューも表示されている。



画面3 Webminのコマンドシェル

Webminではブラウザから直接コマンドを入力することもできる。hdparmの結果はかなり高速だ。



写真3 本体内部

2台のハードディスクとマザーボードは、ケーブルではなくコネクタで直接接続される。



写真4 本体前面パネル

左から、VGA出力、PS/2キーボード、PS/2マウス、シリアル、IEEE 1394 x 2、USB x 2といったインターフェイスコネクタが並んでいる。



写真5 本体背面パネル

2ポートの10/100BASE-TX対応のLANインターフェイスと電源スイッチがある。右側内部の側面のスイッチで100/115Vと220/240Vを切り替え可能。

また、本機を縦に6台並べて、ラックマウントするアダプタがオプションで用意されていて、業務用サーバシステムとしても利用可能だ。ラックマウントで利用する場合は、カバーは不要なので取り外してアダプタに装着する。ANSI 19インチラックならば高さ4U (177mm) サイズ、JISならば高さ200mmに収納できる。

OSはRed Hat Linuxを採用

本機のハードディスクには、Linuxがインストールされて出荷される。現在は、Red Hat Linux 6.2のFTP版をベースにしており、カーネルを2.2.17にバージョンアップし、USBやIEEE1394、FastTrack100ドライバのサポートを組み込んでいる。

また、OpenSSLやUSBマネージャ、Webminなどのツールを足してあるほか、Windowsファイルサーバを

samba-2.0.7-jaに変更するなど、いくつかのパッケージがアップデートされている。

本機には、Webブラウザからシステムの設定を変更可能な、Webminがインストールされている。インストールされているWebminは残念ながら英語版だが、メニューから言語を日本語に設定したところ、一部は日本語で表示された(画面1)。日本語版への差し替えも検討中とのことだ。

Webminは非常に多機能で、ネット

ワークやシステムといったLinuxの基本設定はもちろん、メール(Sendmail)、Web(Apache)、FTP、DNS(BIND)、ファイル(Samba)、プリンタ、プロキシ(Squid)などのサーバ機能を設定できる(画面2)。

SERVER THE BOXには、通常必要なサーバの機能をほとんど備えているが、グループウェアなどの業務用アプリケーションをインストールするベースマシンとして使用するのがよいのではないだろうか。

CPU	Celeron 566MHz (オプションでPentium 700MHz)
チップセット	Intel 440BX
RAM	128MバイトSDRAM (PC100、DIMMスロット×1)
ハードディスク	40GバイトIDE×2基 (RAID 1ミラーリング)
RAIDコントローラ	PROMISE PDC20267 (FastTrack100)
CD-ROM、フロッピードライブ	なし
グラフィックス	オンボードC&T 69000 (2Mバイト)
ネットワーク	オンボード10/100BASE-T×2ポート (Intel 82559)
インターフェイス	RS-232C×1、USB×2、PS/2×2ポート、IEEE 1394×2
PCIスロット	なし
キーボード/マウス	なし
サイズ (mm) 重量	190 (W) × 264 (D) × 74 (H)、約3.5kg
電源	AC100/115Vまたは200/240V切換

表1 SERVER THE BOXの主な仕様

これからのスタンダードはどれだ！？

ディストリビューション 最新カタログ

カーネル2.4、KDE 2.1、GNOME 1.4など新機能を搭載したディストリビューション続々登場！
Red Hat、Turbo、LASER5、Kondara、Vine、HOLON、Debian、Plamo、MLD5、
Omoikane、SuSE、Mandrake、OpenLinux、Slackware、LinuxPPC、etc...





Official
製品版

7.2

Meister

Linux
Mandrake

7.2

Linux Media Lab. Distribution

Linux

redhat

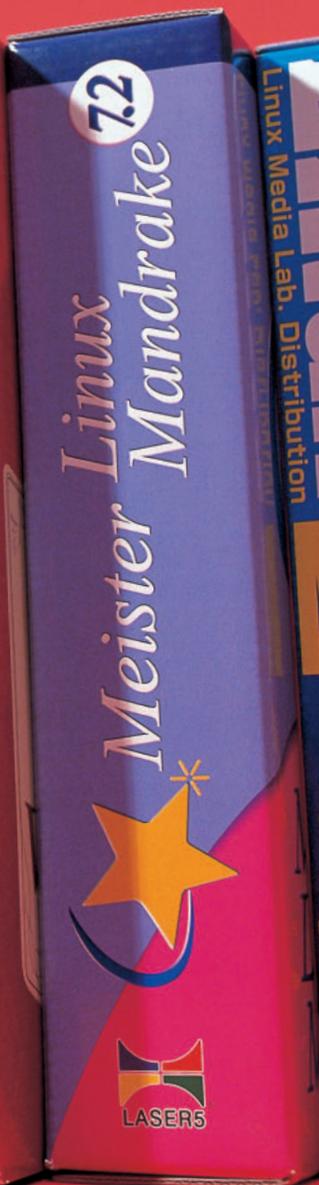
Linux PPC 2000 Q4 日本語版

DELUXE
WORKSTATION

Official
製品版

Vine Linux 2.1 CR

デジタルリナックス
エキスパートまで
すべてのユーザに贈る最新 & 最強Linuxパッケージ



Official
製品版

7.2

Meister

Linux
Mandrake

7.2

Linux Media Lab. Distribution

Linux

redhat

Linux Linux PPC 2000 Q4 日本語版

DELUXE
WORKSTATION

Official
製品版

Vine Linux 2.1 CR

デジタルリナックス
エキスパートまで
すべてのユーザに贈る最新 & 最強Linuxパッケージ



Official
製品版

7.2

Meister

Linux
Mandrake

7.2

Linux Media Lab. Distribution

Linux

redhat

Linux Linux PPC 2000 Q4 日本語版

DELUXE
WORKSTATION

Official
製品版

Vine Linux 2.1 CR

デジタルリナックス
エキスパートまで
すべてのユーザに贈る最新 & 最強Linuxパッケージ



Official
製品版

7.2

Meister

Linux
Mandrake

7.2

Linux Media Lab. Distribution

Linux

redhat

Linux Linux PPC 2000 Q4 日本語版

DELUXE
WORKSTATION

Official
製品版

Vine Linux 2.1 CR

デジタルリナックス
エキスパートまで
すべてのユーザに贈る最新 & 最強Linuxパッケージ



Official
製品版

7.2

Meister

Linux
Mandrake

7.2

Linux Media Lab. Distribution

Linux

redhat

Linux Linux PPC 2000 Q4 日本語版

DELUXE
WORKSTATION

Official
製品版

Vine Linux 2.1 CR

デジタルリナックス
エキスパートまで
すべてのユーザに贈る最新 & 最強Linuxパッケージ

最新ディストリビューション事情

今回の特集は、本誌では昨年7月号以来ほぼ1年ぶりとなるディストリビューション企画となるわけだが、その間にどのような変化や傾向が見られたのだろうか。特集を始めるにあたって、まずはそのあたりの事情を探っていくことにしよう。

カーネル2.4の登場

Linuxシステムの中核（というより本来はLinuxそのもの）であるカーネルのバージョンが2.4にアップしたことは、当然ディストリビューション業界にも大きなインパクトを与えた。各ベンダーとも、積極的にこの新カーネルの採用を発表し、新バージョンの開発をスタートさせた。

そして今年4月には、業界最大手であるRed Hatからカーネル2.4を正式採用したRed Hat Linux 7.1リリースされ、以降2.4搭載ディストリビューションのリリースラッシュを迎えることになった。今回紹介するディストリビューションでも、リリース時期によって

リビジョンの違いが多少あるものの、その多くが2.4系のカーネルを採用している。

引き続き2.2系カーネルの採用を選択したVine LinuxとPlamoも、2.4系からのバックポートパッチを適用し、USBやReiserFSのサポートといった新機能を取り入れている。

カーネル2.4の採用に伴い、従来の標準ファイルシステムであるext2に代わる新しいファイルシステム（ReiserFS、ext3など）をサポートするディストリビューションも増えてきた。今のところ、パッケージとして収録されているだけであったり、すでにインストーラから選択してルートファイルシステムとして使えるようになっていたり、とサポート状況はまちまちだが、今後、これらの新しいファイルシステムが標準となっていくことは間違いない。

ファイルシステム以外にも、カーネル2.4の採用によってユーザーが享受できるメリットは多い。搭載できるメモ

リ/IDEコントローラ/ネットワークカードの上限や、同時実行可能なプロセスの数、ファイル/ファイルシステムの最大サイズといったシステムリソースの制限が大幅に引き上げられ、マルチプロセッサ環境での処理性能の効率化、ネットワークサブシステムの改良といった機能強化とあわせて、スケラビリティが大幅に向上している。また、USB、IEEE1394、ACPIといったPCハードウェア技術のサポートも大きな強化ポイントだ。

このように、サーバ用途、デスクトップ用途の双方で即効性のある大きな変更がなされているだけに、カーネル2.4を採用したディストリビューションではその機能を十分に生かすことのできるシステム設定が必要であるし、マニュアルなどのドキュメント類の整備とサポートの充実といったユーザーに対するケアもより重要になってくるはずだ。

シングルバイナリ化と多言語対応

従来は、英語、フランス語、日本語といった各言語ごとに違うバージョンのバイナリパッケージのバージョンが用意されていたが、Li18nux（Linux Internationalization Initiative）の活動や、glibc、GTK+、Qt、XFree86などコンポーネント自体でのマルチバイト文字処理の機能強化によって、1つのバイナリバージョンで多言語に対応できるようになった。こうした状況を受けて、ディストリビューションのシングルバイナリ化が進んでいるのも



画面1 カーネル2.4のコンフィグ画面
USBやIEEE1394など周辺機器のサポートが大幅に強化されている。



最近の傾向のひとつだ。主なところでは、SuSE、Mandrake、Red Hatなどがシングルバイナリ化を進めているが、日本語処理については完成度にばらつきがあるようだ。

Debian GNU/Linuxは独自の多言語サポートを早くから進めており、最新の安定バージョンであるPotato (2000年8月リリース)からは、日本語のインストールパックも標準で用意され、インストーラから選択できるようになっている。

また、国内での展開がメインである日本語ディストリビューションでも、韓国語、中国語などのアジア圏の言語を処理できるようロケールデータやフォント、IMEなどの環境を用意する傾向が見られる。この点に力を入れているのがKondara MNU/Linuxだ。Kondara Projectでは、Unicode (UTF8)への対応も早くから進めており、最新のバージョン2.0にはその成果が多く取り入れられている。

汎用性か？ 用途の特化か？

組み込みシステムから、デスクトップ、さらにはサーバまで、Linuxはもともと幅広い用途が特徴とされるオペレーティングシステムである。前述したようにカーネル2.4では、さらにスケラビリティが向上しており、より多くの局面で活躍できる可能性を持つに至っている。

すでにほとんどのディストリビューションには、サーバ用途に特化したパッケージが用意されているが、カーネルのスケラビリティの向上を受けてこの傾向が強まっていくと予想される。サーバ用途だけでなく、統合プログラム開発環境やオフィスアプリケーションなどをバンドルしたミッドクラスの

製品を提供するディストリビューターも増えてくるだろう。

デスクトップからサーバまでひと通りの製品ラインアップを揃えるディストリビューターがある一方で、特定の用途やユーザー層に的を絞ってくるところも出てくるのではないだろうか。現在でも、ある程度ディストリビューターの「棲み分け」がなされているが、Linuxの商品単体としての市場がほぼ飽和状態にあることを考えると、より鮮明に独自性を打ち出していかなざるを得ないはずである。

先頃リリースされたメディアラボのLinux MLD miniのようなユニークなディストリビューションの登場は、その先駆けともとれるだろう。

現行の勢力図

ディストリビューターの棲み分けと独自性の追及という話題が出たところで、最後に、どのディストリビューションが最もよく使われているのかという現状について触れておこう。

とはいえ、Linuxは再配布が自由なオープンソースソフトウェアという性格上、正確なインストールベースの統

Vine Linux	42.9%
TurboLinux	35.2%
Red Hat Linux	35.2%
Kondara MNU/Linux	14.1%
LASER5 Linux	9.6%
Slackware	6.4%
Debian GNU/Linux	5.7%
Plamo Linux	5.7%
Linux MLD	3.8%
HOLON Linux	2.5%

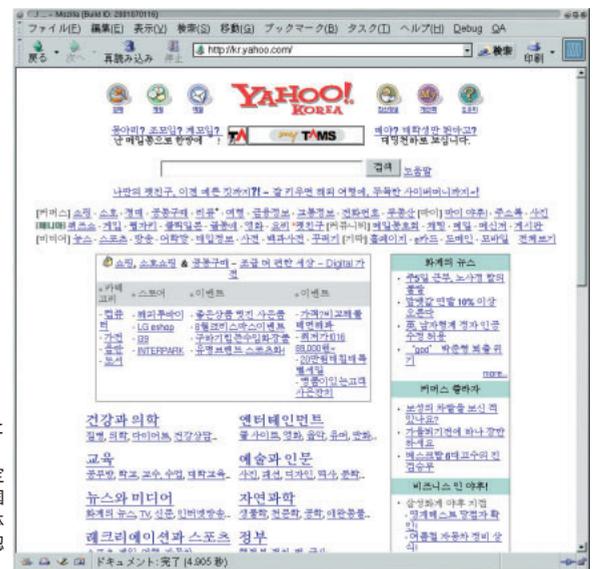
表1 Linux magazine読者が使っているディストリビューション・ベスト10 (複数回答可)

計はない。ここでは、本誌がWebサイトで実施している読者アンケートの結果を紹介する。

他のアンケート項目から浮かび上がる本誌の読者像は、「パソコン歴は長い、Linuxに関しては初・中級者」という、おそらくは最も一般的と思われるLinuxユーザー層とほぼ重なっている。それらのユーザーが出した解答は表1のとおり(ただし、本誌では取り上げることの少ないDebianの数字は低めになっていると思われる)。

この月以外の結果でも、Vine、Red Hat、Turbolinuxの3つは安定した数字を残しており、多くの固定的な支持層を獲得しているようだ。新しいバージョンが出揃ったところで、この数字がどう変化していくのが注目である。

画面2 Mozillaで韓国語を表示したところ
Kondara MNU/Linuxでは、特別な設定をすることなく、この画面のように韓国語が表示できた。中国語の簡体/繁体文字、タイ語も表示できることを確認した。



ディストリビューション選びのツボを探る

このパートでは、ディストリビューションを選ぶ際の参考となるようなポイントを押さえながら、先ほどとは違った角度から最近の傾向を分析して解説する。なお、このパートおよび40ページ以降のレビューでは、以下の15種のディストリビューションを対象とする(掲載順)。

- Red Hat Linux 7.1 日本語版
- Turbolinux 7 Workstation
- LASER5 Linux 7.1
- Kondara MNU/Linux 2.0
- Vine Linux 2.1
- HOLON Linux 3.0
- Debian GNU/Linux 2.2r3
- Plamo Linux 2.2.1
- Linux MLD 5
- Omoikane GNU/Linux 1.2
- SuSE Linux 7.2
- Meister Linux Mandrake 7.2

- OpenLinux Workstation 3.1
- Slackware Linux 8.0
- LinuxPPC 2000 Q4 日本語版

用途に合ったものを選ぶ

前のパートで解説したように、Linuxはデスクトップからサーバまで幅広い用途に対応するオペレーティングシステムである。基本的には、どのディストリビューションを使っても、必要なソフトウェアを追加したり、適切に設定を変更することで用途に合わせたシステムを構築することが可能だ。とはいえ、そのためには一定水準の専門知識と経験が必要になってくるし、インストール以外に追加の作業が発生するのも面倒に感じられるだろう。

そこで、なるべくインストール後の作業が発生しないように、カーネル、

glibc、XFree86といった主要コンポーネントのバージョン、必要なアプリケーションや使い慣れたツールの有無を確認しておきたいところだ。ディストリビューターによっては、Webサイトに収録パッケージの完全なリストを用意している場合もあるのでチェックしてみよう(画面1)。

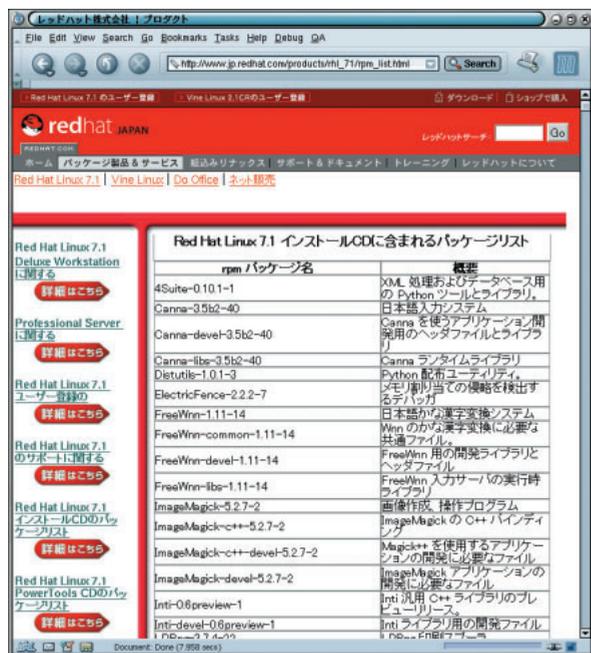
リストが見つからない場合でも、少なくとも製品版にバンドルされる商用ソフトウェアに関する説明はあるはずだ。あとは、本誌を始めとする雑誌・書籍などに添付されているCD-ROMを直接確認すれば、詳細なパッケージ構成を調べることができる。

たとえばRPMパッケージを採用しているディストリビューションでは多くの場合、インストールCD-ROMの以下のパスにパッケージファイルが収録されている。

`/<ディストリビューション名>/RPMs/`

なお、40ページからのカタログページでも、製品版にバンドルされる商用ソフトウェアと主な収録パッケージについて解説しているので参考にしてほしい。

用途とあわせて事前に確認しておきたいのが周辺機器などのサポート状況だ。これもやはり、ディストリビューターのWebサイトに情報が掲載されているかどうかチェックしておこう。正式にサポートされているハードウェアであれば、インストール後に動作しない場合でも設定方法などを問い合わせることができる。



画面1 Red HatのWebサイトに掲載されているパッケージリスト



インストーラと アップデートツール

インストーラは、ユーザーにとって最初にそのソフトウェアに触れるインターフェイスとなる。第一印象はやはり大切で、インストーラの使い勝手が悪かったり、不具合があったりすると全体として悪いイメージを持ってしまいうだろう。

Linuxの場合、以前は「WindowsやMacintoshに比べてインストールと初期設定が難しい」という印象を与えることが多かったようだが、Red HatのAnaconda（画面2）、SuSEのYaST2（画面3）といったグラフィカルなインターフェイスを持ったインストーラが登場し、設定項目なども整理されて操作性が大幅に向上した。

すでに前回のディストリビューション特集の時点（2000年7月号）で、ほとんどのディストリビューションがグラフィカルなインストーラを採用していたが、今回取り上げているディストリビューションではさらに、独自のインストーラを備えているもの（TurboLinux、MLD 5、Mandrakeなど）や、ベースとなるインストーラに改良を加



画面2 Red Hatのグラフィカルインストーラ「Anaconda」

えたもの（Kondara、Vine、HOLON、Plamoなど）が多く見られた。

インストールという面で他のディストリビューションと大きく違う特徴を持っているのがDebianである。インターネットを介した配布が主流となっており、ベースのシステムをインストールしておき、あとは必要なパッケージをネットワークインストールするという方法が一般的だ。これを実現しているのが「apt」というパッケージ管理ツールである（詳しくは52ページからのDebianの紹介記事を参照）。

aptはパッケージのインストールと管理を行うユーティリティとして高い評価を受けており、RPMパッケージ向けに移植されたものがVineやHOLONといったディストリビューションでも採



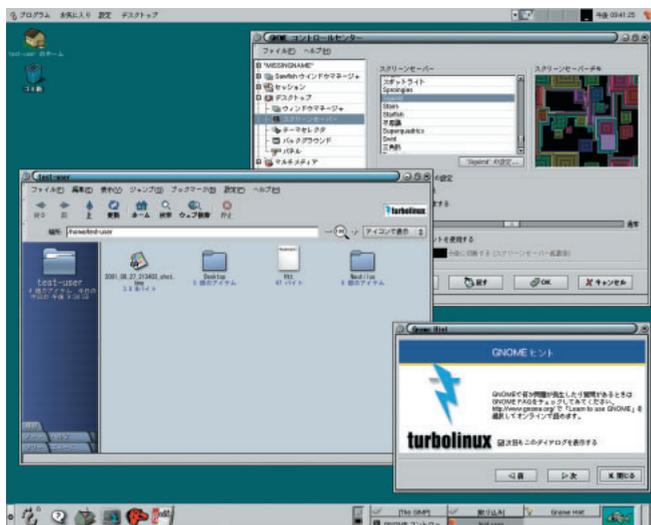
画面3 SuSEのインストール&システム設定ツール「YaST2」

用されている。

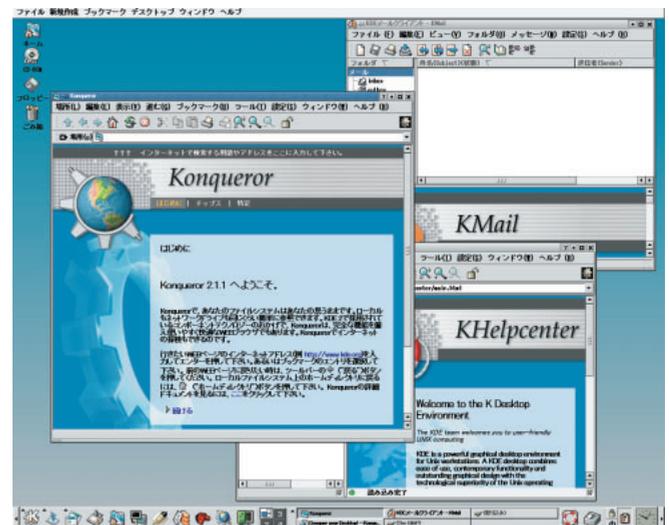
apt以外にも、Red Hatのアップデートエージェント、Kondaraのmph、TurboLinuxのZabom、OpenLinuxのCaldera System Updaterなどネットワーク経由のインストールに対応したアップデートツールが増えてきている。これらのツールの利点を100パーセント生かすには、高速で安定したインターネット環境を利用できることが求められる。このことは、選択のポイントとして押さえておいてほしい。

統合デスクトップ環境と 日本語処理

サーバ用途はともかく、パーソナルユースで常用する場合には、自分の気に入ったユーザーインターフェイスを



画面4 GNOME 1.4のデスクトップ



画面5 KDE 2.1のデスクトップ

備えていることが非常に重要なポイントとなるだろう。特にWindowsやMacintoshからの移行を考えている初級者にとっては、操作性の良いGUI環境が整っているほうが何かと便利はずだ。

Linuxでは、統一されたGUIを備えたデスクトップ環境としてGNOME(画面4)とKDE(画面5)を利用できる。また、統合デスクトップ環境に近い機能性を持ったウィンドウマネージャに、Window MakerとEnlightenmentがある。どれが一番優れているのかということについては、好みの問題でもあり難しいところだ。

さらに、ほとんどのディストリビューションでは、これらのうちの複数が標準で収録されており、グラフィカルログイン時に選択するか、設定ツールを使って切り替えることによってGUI環境を選べるようになってきている。このため、ディストリビューション間での差異は、実はそれほど大きくない。選択の基準をひとつ挙げるとすれば、標準デスクトップ環境としてどれが採用されているかということになる。

もちろん、すべてがきちんと整備されているのが望ましいのだが、やはり各ディストリビューションとも、デフォルトで選択されるデスクトップ環境

に最も気を配っているはず。使用するデスクトップ環境が決まっている場合は、それがデフォルトとなっているディストリビューションを選ぶようにするとよいだろう。

デスクトップ環境に関連する事項として、日本語処理についても触れておこう。

エディタ、ワープロ、Webブラウザ、メーラなど、パーソナルユースで日常的に使われるアプリケーションで日本語を表示・入力できるというのが最低ラインだと思うが、日本語サポートを謳うディストリビューションであれば、この点についてはクリアしているようである。

以前は、Vineの日本語サポートが群を抜いている印象もあったが、ほかのディストリビューションの努力もあって、かなりその差は縮まってきている。ただし、アプリケーションごとの細かな設定などの面では、Vineが依然として一歩リードしていることは否めない。

日本語処理に関して、特に製品版でチェックしておきたいのが、バンドルされるかな漢字変換ソフトと商用TrueTypeフォントだ。本格的にデスクトップ用途で使っていくには、どちらも欠かせない要素である。

かな漢字変換ソフトは、66ページの

機能一覧を見てもらえばわかるとおりWnn6 Ver.3とATOK 12 SEが主流になっている。その中で、TurbolinuxとLASER5がATOK Xをバンドルしている。ATOK Xは、Windows版のATOKとほぼ同等の操作性とカスタマイズ機能を備えており、現在Linuxで利用できるかな漢字変換ソフトとしては最高の水準にある。日本語処理を重視する人にとっては見逃せないポイントだ。

商用フォントでは、DynaFontを採用したディストリビューションが圧倒的に多いが、注目は唯一リコーのフォント(5書体)を採用したTurbolinuxだろう。リコーはWindowsの標準フォントの供給元でもあり、小さなサイズの文字の見やすさに定評がある。Webページをブラウズする際などにWindowsに近い表示になる(画面6、画面7)。Turbolinuxでは、FTP版にもリコーのTrueTypeフォント(ゴシック1書体)を収録している。

フリーの日本語TrueTypeフォントでは、東風明朝と東風ゴシック(画面8)が新たに採用されるようになった。この2書体は、Turbolinux、Kondara、HOLONに標準収録されている。

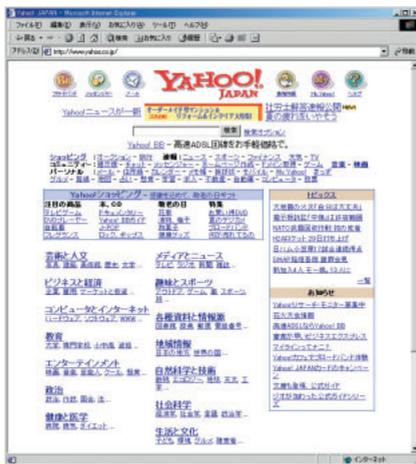
ユーザーサポート

製品版で保証されるユーザーサポートの範囲については、66ページの一覧を参照してもらおうとして、ここでは別の角度からユーザーサポートについて考えてみたい。

まずは、マニュアルなどのドキュメントの充実度が挙げられる。マニュアルは実際に見てみるまで、自分に合った内容なのか知ることができないが、ほとんどの場合はインストールガイドとLinuxのチュートリアル的な構成に



画面6 Turbolinuxのリコーフォントで表示したYahoo!



画面7 Windows 2000の標準フォントで表示したYahoo!



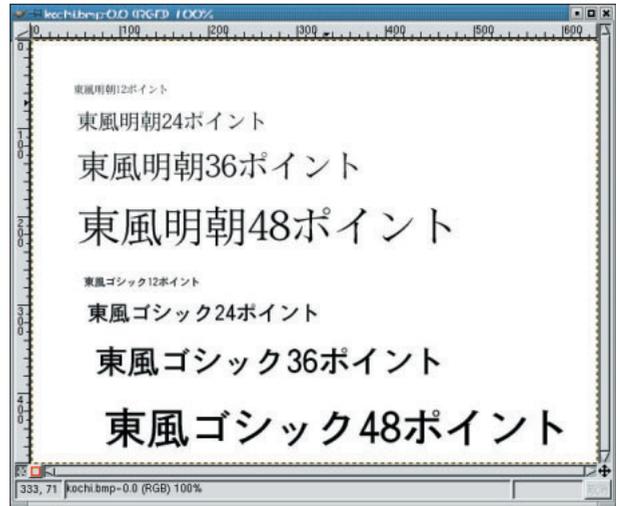
なっているので、初級者には役立つはずだ。また、Red HatやTurbolinuxのように、各種ドキュメントをまとめて収録したCD-ROMを製品版と同梱しているディストリビューションもある。

Webサイトを通じた技術情報の提供も、ユーザーサポートのひとつの形態である。サポートページの充実度は事前にチェックできる。中には、製品版の登録ユーザーにのみ公開されるページを用意しているところもあるので、そのあたりも確認しておこう。

製品版のないDebianとPlamoのような非商用ディストリビューションは、Webでの情報提供がユーザーに対するケアの第一の手段となる。Debianでは、インストールマニュアルをWebサイトを通じて配布しているほか、パッケージの紹介やメーリングリストの案内などユーザーに有用な情報を提供している。Plamoは開発者のこじまみつひろ氏とユーザーの有志のサポートによって進められているプロジェクトで、人的なリソースに限りがあるなか、リリース状況や設定のポイントなどの情報をWebページに掲載してくれている。

もう1つユーザーサポートのための場

画面8 東風明朝&東風ゴシック



がある。先ほどチラッと触れたユーザーメーリングリストがそれで、ユーザーの情報交換だけでなく、新しくユーザーとなった人のケアといった相互扶助的な役割も果たしているのだ。製品版でのユーザーサポートの保証という手段を取れないDebianとPlamoにとっては、特に重要なサポートの場となっている。商用ディストリビューションのうちでも、開発プロジェクトがもとめと非営利の組織であるVineなどは、ユーザーメーリングリストで活発なやり取りが行われているようだ。

こうしてひと通り概観しただけでは、ディストリビューション間の差異があまり感じられないことがわかる。現在のところ、各ディストリビューションとも独自性を模索中といったところなのだろう。

選択するユーザー側から見れば、より細かな部分にも目を向け、コストパフォーマンスなどを含めたトータルな評価が必要になってくる。次ページからは、各ディストリビューションについて、その特徴や本誌なりの評価をお伝えしていく。ディストリビューション選びの参考にしてほしい。

製品紹介ページにあるアイコンの意味

次ページからのカタログコーナーでは、各ディストリビューションにおける新機能のサポート状況をアイコンを使って表している。ここで、それぞれのアイコンの意味を説明しておこう。

カーネル2.4のアイコン

見てのとおり、カーネル2.4を採用しているかどうかを表している。ただし、ボーナスCDなどに収録されていて標準的な方法でインストールされない場合は除いてある。

XFree86 4.xのアイコン

カーネルと同じく標準でXFree86 4.xがインストールされるかどうか。3.x系と4.x系の両方をサポートし、インストーラでビデオカードを自動判別してそのカードに合

ったバージョンを使うように設定するディストリビューションもある。この場合、サポートあり(チェックマークあり)となっても、環境によっては4.xが使われないことに注意。

GNOME 1.4とKDE 2.1のアイコン

最新のデスクトップ環境のサポート状況を示している。やはり、標準でインストールされる場合のみ、サポートありとしている。なお、KDEについては、2.2がリリースされたばかりなので2.1を対象とした。

日本語のアイコン

Linuxの国際化もかなり進展し、海外のディストリビューションでも日本語のフォントやロケールデータを含んでいることが多くなっているが、

ここではそのディストリビューションが公式に日本語対応版としてリリースされているかどうかを基準としている。

ReiserFSとext3のアイコン

ファイルシステムのサポート状況を示している。パッケージとしてインストールされるだけでなく、インストーラからそのファイルシステムを設定できるかどうかを基準としている。

8G超のアイコン

8Gバイトを超えるディスク領域にインストールしてブートできるかどうかを示す。インストーラからLILOのlba32オプションを指定でき、8Gバイト境界の問題をクリアしたブートローダを使う場合、このアイコンがチェックされている。



Red Hat Linux 7.1

Deluxe Workstation 日本語版

レッドハット株式会社

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

1万4800円

✓ カーネル2.4

✓ XFree86 4.x

■ GNOME 1.4

✓ KDE 2.1

✓ 日本語

■ ReiserFS

■ ext3

■ 8G超

前バージョンであるRed Hat Linux 7Jは、カーネル2.4インストールレディでありながら、カーネル2.2.16を採用するという中途半端なリリースであったが、このRed Hat Linux 7.1（以下、Red Hat 7.1）ではカーネル2.4を正式に採用した。前バージョンの7Jからは、世界に先駆けて国際化ライブラリを採用したRed Hat Linuxだが、さらにこのバージョンからは世界共通バイナリとなり、「J」といった日本語を意味するバージョン番号が削除された。ただし、世界共通バイナリとはいえ、日本語版には日本独自のバンドルソフトウェアなどが付属しており、パッケージ構成は英語版とは異なっている。

日本で発売されているRed Hat 7.1には、Deluxe Workstationと、Professional Serverの2種類があり、それぞれデスクトップ向け、サーバ向けとなっている。なお、Professional Serverの詳細については、サーバ系ディストリビューションの項目を参照してほしい。

ソフトウェア構成

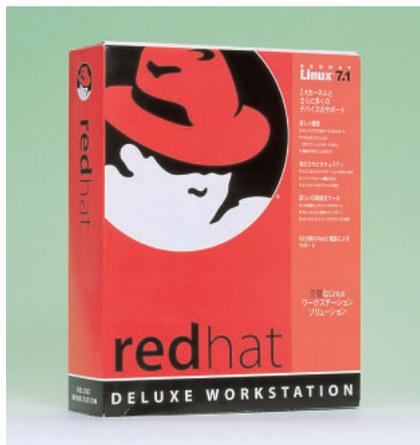
Red Hat 7.1は、比較的早い時期（5月11日）にカーネル2.4を採用して発売されたこともあり、ほかのカーネル2.4採用ディストリビューションに比べると各種ソフトウェアのバージョンが古めとなってしまう。たとえば、カーネル2.4.2、GNOME 1.2、XFree86 4.0.3を採用しているなど、ほかのカーネル2.4採用ディストリビューションと比べると若干劣りする感がある。

また、インストーラであるAnacondaが最新機能に対応していないため、インストール時に最新機能を利用できない面もある。たとえば、カーネルはReiserFSに対応しているものの、インストール時のファイルシステムにResierFSを選択できない点や、LILOは8.4G以降にあるカーネルイメージをロードできるものの、インストール時に設定できない点などだ。こうした点は、仕方がないとはいえやや残念だ。

主なバンドルソフトウェア

Deluxe WorkstationにはHancorn Word、DynaFont、CanonBJプリンタドライバといった商用ソフトがバンドルされている。なお、Red Hat Linuxでは今まで、商用日本語入力システムとしてジャストシステムのATOKと、オムロンソフトウェアのWnn 6 Ver.3をバンドルしていたが、今回のRed Hat 7.1ではオムロンソフトウェアのWnn6 Ver.3のみとなった。

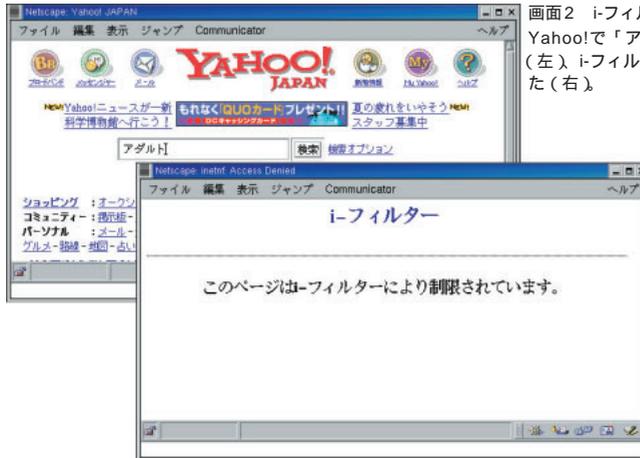
変わったバンドルソフトとしては、デジタルアーツのi-フィルターがある。i-フィルターはPICS（Platform for Internet Content Selection。ページの内容を格付けした基準）、ホワイトリスト、ブラックリスト、キーワードの4つのフィルタを使い、アダルトサイトや有害サイトへのアクセスを制限するためのフィルタリングソフトだ（画面2）。Red Hat 7.1にバンドルされているi-フィルターはバージョン1.0の



画面1 Red Hat Linux 7.1のデスクトップ画面
Mozillaも使えるが、標準WebブラウザはNetscapeになっている。

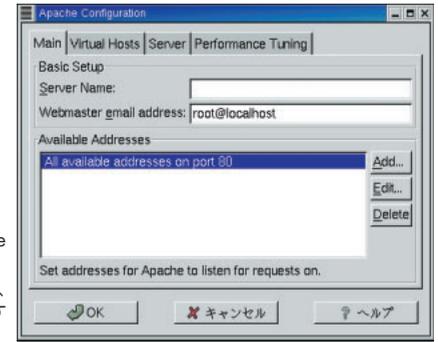
Red Hat Linux 7.1 Deluxe Workstation日本語版





画面2 i-フィルターによるフィルタリング
Yahoo!で「アダルト」を検索してみたところ
(左)、i-フィルターによってフィルタリングされ
た(右)。

画面3 Apache Configuration
GNOMEメニューから、
「Apache設定」を選択す
ること起動する。



Business Editionの評価版だが、「アダルト」といったキーワードなどを含むページのブラウズを抑制することができる。

また、SimCity 3000など、実際にプレイできる2本のゲームタイトルのデモ版も収録されている。

なお、バンドルソフトウェアのインストールは、英語版については専用のインストーラが付属しており簡単にインストールすることができるが、日本語版のバンドルソフトについてはREADMEファイルなどを参照しながらひとつひとつ手でインストールする必要がある。さらに、READMEファイルなども用意されていないソフトもあるなど、初心者がインストールするには若干不安が残る。せっかく英語版には専用のインストーラが付属しているのだから、日本語版にもこのような配慮が欲しいところだ。

新しい設定ツール

Red Hat 7.1では、従来のLinuxconfに加え、GUIでWebサーバを設定できるApacheの設定 (apacheconf。画面3)、DNSサーバの設定を行うことができるBIND設定 (bindconf) ツールが用意された。また、プリンタ設定やプリ

ントキューの管理ができるプリンタ設定 (printconf-gui。画面4) ツールも用意されている。これらのGUIツールによって、面倒だったDNSやプリンタの設定が簡単に行えるばかりでなく、ユーザーインターフェイスがWindowsライクになっており、Windowsユーザーでも違和感なく設定することが可能だ。

気になるセキュリティの設定については、簡単にファイアウォールを構築できるLokkitファイアウォールも用意されており、初心者でも安心してインターネットに接続できるようになっている。

ネットワーク アップデートツール

Red Hat 7.1では、Red Hat Networkというインターネット経由によるアップデートサービスが用意された。これは、Errataやアップデートパッケージを追加する仕組みで、Kondaraに用意されているmphなどと同等のものだ。これにより、常にユーザーは最新のパッケージに更新することが可能になる。

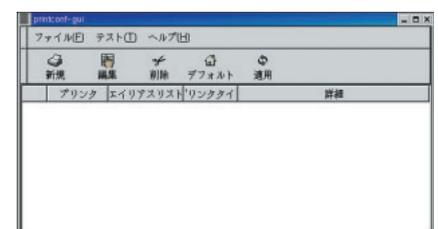
Red Hat Linux 7.2

前述したように、早い時期に発売されたRed Hat 7.1は若干機能が古くな

ってしまっている。しかし、数カ月後にはRed Hat Linux 7.2の発売も控えており、各種ソフトウェアがバージョンアップされるだろう。実際、編集部で入手したRed Hat Linux 7.2のベータ版を評価したところ、当然ながら最新の機能が盛り込まれていた。ベータ版なので、詳細は変更される可能性があるが、インストーラでのext3ファイルシステムへのインストールや、8.4Gバイト以降にあるカーネルイメージを読み込めるオプションなども指定できるようになっている。

また、ブートマネージャもグラフィカルなLILOとGRUBを選択できるようになり、マルチブート機能が大幅に強化される予定だ。

なお、Red Hat Linux 7.2ではGNOME1.4、ファイルマネージャとしてNautilusが採用される予定となっている。Nautilusでは、使い易いインターフェイスに加え、フォントやアイコンが美しく表示されるため、デスクトップの視認性や使い勝手が大幅に向上するだろう。



画面4 printconf-gui
GNOMEメニューから、「プリンタ設定」を選択すること起動する。使い勝手は、Windowsに似ている。

Turbolinux 7 Workstation

ターボリナックスジャパン株式会社
http://www.turbolinux.co.jp/

1万5800円

✓ カーネル2.4

✓ XFree86 4.x

✓ GNOME 1.4

✓ KDE 2.1

✓ 日本語

✓ ReiserFS

✓ ext3

✓ 8G超

Turbolinux 7 Workstation (以下、Turbolinux 7) は、9月7日発売の最新ディストリビューションだ。前バージョンのTurbolinux Workstation 6.0から、約1年半ぶりのバージョンアップとなる。なお、今回紹介するディストリビューションの中では、もっとも後発にあたるため、最新のソフトウェア構成になっている。

大幅に改善された インストーラ

Turbolinuxといえば、CUIによるテキストインストーラという印象があるが、Turbolinux 7ではTurbolinux Server 6.5から採用されたグラフィカルなインストーラ、Mongooseが採用されている。

Mongooseは、ユーザーインターフェイスを研究したというだけあって非常にわかりやすいインストーラで、初心者でもとまどうことなくインストール作業を行うことができるようになっている。さらに、インストール時に

ReiserFS、ext3、JFSなどを選択することができるようになり、最新のジャーナリングファイルシステムをすぐに利用できるのも嬉しい機能だ。

もちろん、インストーラでの8Gバイトを超える領域にあるカーネルイメージの読み込みもサポートされている。

さらに、このバージョンではグラフィカルなLILOが採用され、Windowsとのマルチブート環境を容易に構築することが可能になった。

最新のソフトウェア構成

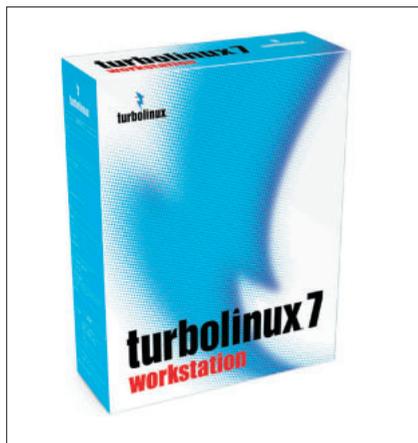
Turbolinux 7では、標準のデスクトップがGNOMEからKDEに変更された。これは、Windowsユーザーが違和感なく使えるインターフェイスを採用するというコンセプトによる変更のようだ。またKDEには、KWord、KSpreadを始めとするオフィススイートKOfficeが用意されているため、Windows + Microsoft Officeといったビジネスユースにも重点を置いた構成

ともいえるだろう。

標準WebブラウザとしてはMozillaが採用されたほか、タブ機能を持つWebブラウザとして注目されているGaleonも利用できるようになっている。ただし、Galeonのメニューは日本語化されているものの、Mozillaのメニューは日本語化されていない。レンダリングエンジンにGeckoを使っているためWebページの日本語表示は問題ないものの、そのほかの日本語環境がしっかり作られているだけにやや残念な点である。

標準のファイルマネージャとしては、KDEではKonquerorを、GNOMEではNautilusを採用している。特に、KonquerorではPDFファイルをブラウズすることができるため、付属のPDFマニュアルを参照する際に威力を発揮する。

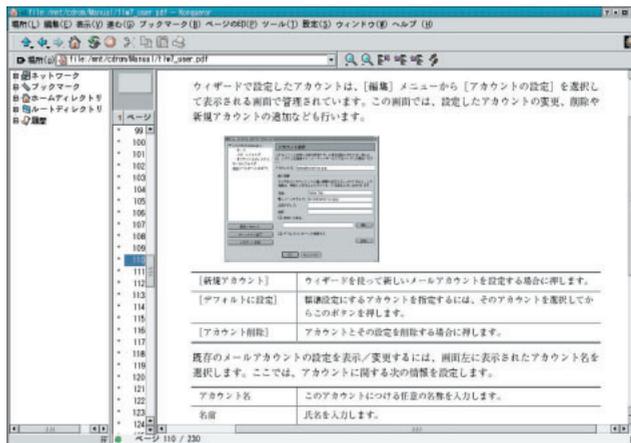
各種ソフトウェアのバージョンは、KDE 2.1.1、GNOME 1.4、XFree86 4.1.0、glibc 2.2.3と最新構成となっている。



Turbolinux 7 Workstation

画面1 KonquerorによるPDFファイルのブラウズ

Turbolinuxに付属するPDFマニュアルを表示した。アンチエイリアスされた美しいフォントで表示され、オンラインマニュアルを読むツールとして十分機能する。





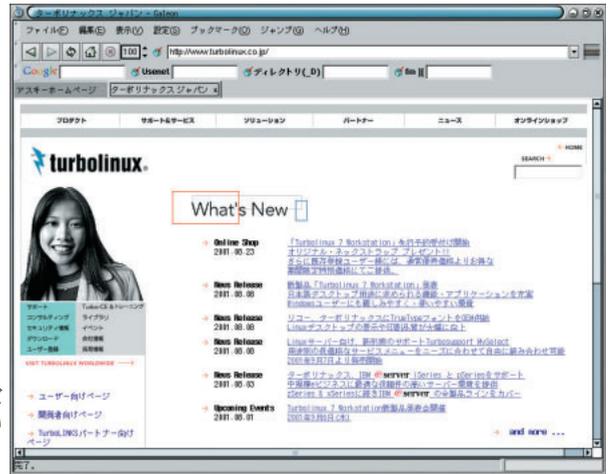
気になる設定ツールだが、turboxcfg、turboserviceなど、以前からあるCUI設定ツールに加え、X Window System上やLAN経由で、Webブラウザから設定できるWebminが利用できるようになってきている。

主なバンドルソフト

商用バンドルソフトは、ATOK X、リコーTrueTypeフォント5書体と若干少なめだ。しかし、ATOKをバンドルするディストリビューションが少なくなる中、ATOKファンには嬉しいバンドルソフトといえるだろう。また、リコーのTrueTypeフォントのうち、TLゴシック、TL明朝は、WindowsのMSゴシックとMS明朝に準拠しているため、Windowsユーザーには違和感なく使えるだろう。特に、WebサイトなどWindowsでブラウザされることを意識したページなどは、レイアウトが崩れることなく表示することができる。もちろん、表示だけでなく、印刷に関してもこれらのフォントが利用可能だ。こうした点からも、Turbolinux 7ではWindowsを意識したディストリビューションとなっている。

なお、FTP版のTurbolinux 7にもTLゴシックの常用漢字版（2719文字）がバンドルされる。1書体、常用漢字

画面2 GaleonによるWebブラウザ機能を持ったGaleonは一度使い始めると手放せないWebブラウザだ。完全に日本語化されている。



版だけとはいえ、FTP版に商用フォントがバンドルされるのは初めてのことで、今後のディストリビューションのターニングポイントとなる可能性を秘めている。

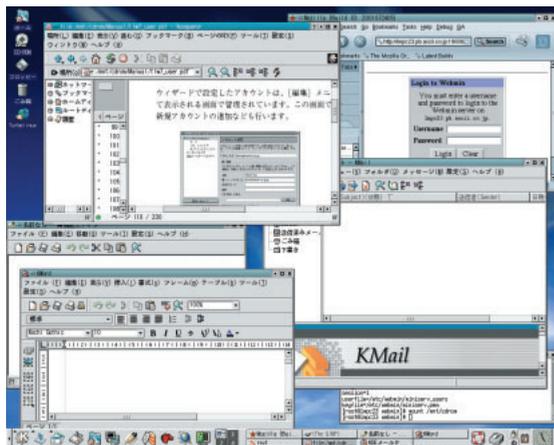
ノートPCとUSBへの対応

ノートPCでは、特殊な解像度のためにX Window Systemが利用できなかったり、ジョグダイヤル、カメラといった専用機器が利用できなかったことが多いが、Turbolinux 7ではこうしたノートPCへの対応が強化されている。また、各種USB機器への対応も強化され、USB CD-ROMや、デジタルカメラといった、今までLinuxで使いにくかった機器も特別な設定をすることなく利用できるようになった。

先進的な取り組み

Turbolinux 7は見た目やインターフェイスに関する最新機能だけでなく、前述した新しいファイルシステムへの対応や、USAGI Projectによって開発されているIPv6プロトコルスタックの取り込み、UNIXのディレクトリ構成を統一するFHS（Filesystem Hierarchy Standard）にほぼ準拠するなどシステムとしての先進性も高いものとなっている。

また、開発環境が充実しているのも特徴だ。たとえば、JDK 1.3.0、Forte for Java、Borland Kylix 1.0 Open Editionがバンドルされるなど、ソフトウェア開発のプラットフォームとしても魅力的になっている。



画面3 KDEによるデスクトップ
TurbolinuxのデフォルトデスクトップはKDEに変更され、Windowsインターフェイスとの統一性をアピールしている。

画面4 GNOMEによるデスクトップ
GNOMEでも、Nautilusを採用した美しいデスクトップが利用できる。フォントのバランスも美しい。



LASER5 Linux 7.1

レーザーファイブ株式会社
http://www.laser5.co.jp/

9800円~

カーネル2.4
 XFree86 4.x
 GNOME 1.4
 KDE 2.1
 日本語
 ReiserFS
 ext3
 8G超

レーザーファイブは、Linuxの初期の頃からLinuxパッケージをCD-ROMで提供してきた老舗のディストリビューターである五橋研究所から独立した会社である。LASER5 Linuxを「ベターRed Hat」として日本市場に合わせたディストリビューションとして提供している。

LASER5 Linux 7.1は、Red Hat Linux 7.1 (US版) とコンパチビリティを持たせ、システムの最適化と日本語環境を充実させたディストリビューションだ。Red Hat Linux 7.1をベースに、ユーザーからの聞き取り調査を元に全体の15%のパッケージ(200点以上)が改訂されている。

製品は、標準パッケージ「LASER5 Linux 7.1デラックス」と、それにATOK Xと開発者向けのアプリケーションをバンドルした「LASER5 Linux 7.1 Devel」の2種類が用意されている(表1)。

基本コンポーネントは、カーネル2.4.3、glibc 2.2.2、XFree86 4.0.3を採

用した。デスクトップ環境はGNOME 1.2.4、KDE 2.1.1を選ぶことができる。

前バージョンのLASER5 Linux 6.4は、カーネル2.2.16、glibc 2.1.3、XFree86 3.3.6、GNOME 1.2.4、KDE 1.1.2だった。標準のデスクトップであるGNOMEは最新版の1.4ではなく1.2.4のままなので見た目はあまり変わっていないが、内部的には大幅な変更が行われている。

初心者にも優しいマニュアル

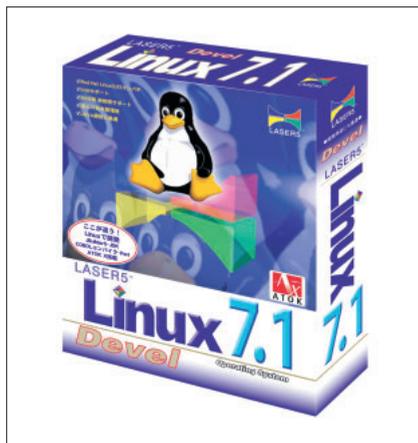
製品パッケージには、CD-ROM 4枚、インストールフロッピー、インストール&活用ガイドが含まれている。従来は分冊だったマニュアルが1冊にまとめられ、500ページ弱の分厚いものになった。インストールのオプションやハードディスクのパーティションの切り方やシステム設定の方法はもちろん、GNOMEの使い方にも多数のページを割いており、Linux初心者がつまずきやすいところも詳しく解説されている。

基本サポートは、インストールからX Window Systemが起動するまでをユーザー登録後90日間受けることができ、従来は3インシデントまでだったところが回数無制限になった。電話、FAX、Web質問フォームによって平日の10時~18時のあいだ受け付ける。

オプションサポートとして、Xの基本操作、基礎コマンドの使い方、ブラウザ/メーラの設定・操作、エディタの基本的な使い方を、1インシデント5000円で受けることができる。

Red Hat Linuxとの違い

まず、レーザーファイブのオリジナルプログラムXIM Switcher(画面1)が入っていることだ。これによって、ATOK X、Wnn6、Cannaといった日本語入力メソッドを切り替えることができる。また、POP3/IMAP4対応の日本語対応メールクライアントSylpheedや、全文検索システムNamazu+kakasiなども含まれている。



	Devel	デラックス
かな漢字変換ソフト「ATOK X」		x
かな漢字変換ソフト「Wnn6」		
商用TrueTypeフォント「DynaFont 5書体」		
Java用ビジュアル開発ツール「JBuilder5 Personal」		x
Java開発用ツール「JDK 1.3」		
COBOLコンパイラ「Tiny COBOL」		x
英日辞書引きソフト「eWnn」		
アンチウイルスソフト「SOPHOS ANTI-VIRUS」		
はがき作成ソフト「K筆」		
リモートコンソールツール「VNC」		
価格	1万6800円	9800円

表1 LASER5 Linux 7.1 Develとデラックスに含まれる主なアプリケーション

LASER5 Linux 7.1 Devel



また、Windowsファイルサーバ機能を持つSambaは、セキュリティホールへの対策が行われたSamba-2.0.10に変更されている。

なお、DNSサーバソフトbindはbind-8.2.3が使われている。Red Hat Linux 7.1ではbind-9.1.0にバージョンアップされているが、従来のバージョンと設定ファイルの互換性を保つためと思われる。

また、インストーラの画面にはLILOブートローダのオプションとして指定するところはないが、LBA32に対応しているため、ハードディスクの1024シリンダ(8Gバイト)を超える領域にインストールが可能になった。

**DevelにはJava
開発ツールを同梱**

常時接続環境に対応する機能が強化されている。インストーラにファイア

ウォールのレベルを設定する機能が増えたほか、ADSLなどのブロードバンド接続に使われるプロトコルPPPoEに対応した設定ツール「RP-PPPoE」も装備されている。ファイアウォールの設定は簡易設定ツール「Lokkit」(画面2)を利用してGUIで設定できる。アンチウイルスソフト「Sophos Anti-Virus for Linux」(画面3)は、企業などで利用する場合はライセンスを購入する必要があるが、個人ユーザーでの利用に限って無償で利用可能だ。

商用ソフトとして、かな漢字変換にはWnn6とATOK X(Develのみ)、DynaFont 5書体、英日辞書引きソフトeWnnをバンドルするほか、DevelにはTiny COBOLコンパイラ、JDK 1.3、JBuilder 5 Personal(画面4)といったソフトウェア開発ツールが付属する。

そのほか、クロスプラットフォーム

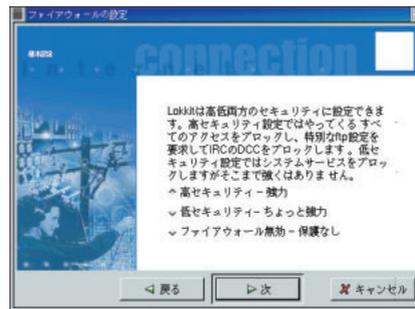
で操作可能なリモートコントロール「VNC」(Virtual Network Computing)や、はがき作成ソフト「K筆」(画面5)などのアプリケーションが利用可能だ。ただ、体験版ソフトが多数収められていたContrib CDと、マルチブートマネージャSystem Commander Liteが付属しなくなったのが、少し残念なところだ。

Red Hat Linuxの日本語版のインストールCDには商用ソフトが含まれていないが、LASER5 Linux 7.1では、インストールCDにATOK XやWnn6、DynaFontが含まれているため、インストールが終了した時点で日本語環境が整えられている。

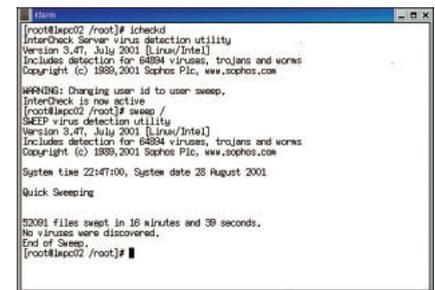
現在の主流であるRed Hat Linuxの最新版の機能を備え、日本語環境を使いやすく改良されたLASER5 Linux 7.1は、初心者ユーザーにお勧めのディストリビューションだ。



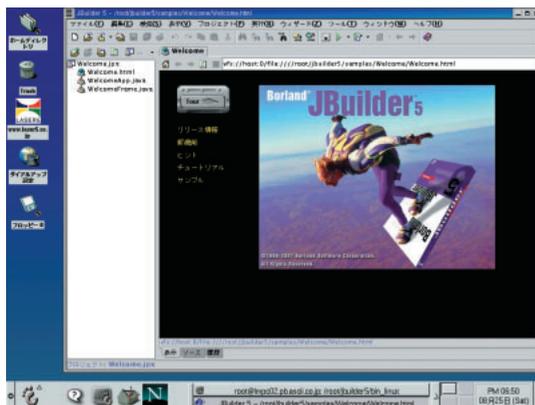
画面1 XIM Switcher
インストールされている日本語入力システムから利用するものを切り替えることができる。



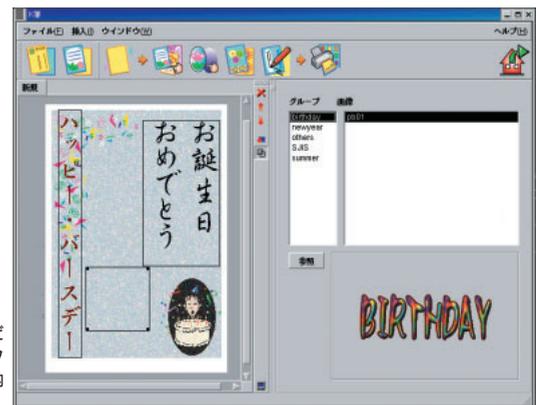
画面2 Lokkit
ファイアウォールをWebサーバ、メールサーバ、ssh、telnetなどの項目の質問に答える形式で簡易設定できる。



画面3 Sophos Anti-Virus for Linux
SWEEPでディスク上のウイルス検索をする。クライアントからサーバにアクセスされたときにウイルス検索を行うことも可能。



画面4 JBuilder 5 Personal
Javaビジュアル開発ツールJBuilderの最新エントリー版。JDK 1.3もバンドルされている。



画面5 K筆
年賀状や暑中見舞いなどのはがきを作成するソフト。専用のフォントも内蔵している。

Kondara MNU/Linux 2.0

デジタルファクトリ株式会社
http://www.digitalfactory.co.jp/

1万2800円

✓ カーネル2.4

✓ XFree86 4.x

✓ GNOME 1.4

✓ KDE 2.1

✓ 日本語

✓ ReiserFS

ext3

✓ 8G超

Kondara Projectによって開発されているKondara MNU/Linuxは、つねに最新の技術を盛り込んだパッケージングで定評がある。

これは、自らが使いたい新機能をいち早く取り入れるというKondara Projectのメンバーの姿勢の表れともいえるだろう。このため、Kondara MNU/Linux 2.0 (以下、Kondara 2.0) では、カーネル2.4.4、GNOME 1.4、KDE 2.2Alpha 1採用など、最新のソフトウェアで構成されているほか、SylpheedやPostfix、ProFTPDなど、使いやすいソフトウェアがパッケージングされている。

また、アップデートやセキュリティパッチの公開が早いのも特徴だ。

さらに、発売元のデジタルファクトリによる可愛いペンギンデスクトップなど、楽しい要素もふんだんに盛り込まれており、初心者から開発者まで人気の高いディストリビューションとなっている。

最新のソフトウェア構成

前述したように、最新のソフトウェアパッケージをいち早く取り入れるKondaraらしく、NautilusやKonquerorといった、話題のファイルマネージャが利用できるようになっている。これら2つのファイルマネージャは、インターネットとのシームレス化、アンチエイリアスされた美しいフォント、洗練されたユーザーインターフェイスなど、今後のX Window Systemの標準となる最新ソフトウェアだ。

標準のWebブラウザもNetscapeからMozillaに変更された。このMozillaもメニューが日本語化されている。

また、デスクトップ以外のソフトウェアも先進的な機能が積極的に取り入れられている。

たとえば、ネットワーク層でIPパケ

ットの暗号化と認証を行うIPSecの取り込みや、新しいファイルシステムであるReiserFSへの対応などだ。

IPSecは非常に強力で、しかもアプリケーションから見た場合、透過的に利用できるという優れた機能を持っている。ただし、期待されていたUSAGI ProjectのIPv6プロトコルスタックは標準では組み込まれていない。この点はやや残念だ。

Kondara MNU/Linuxでは早くからUTF-8への移行を表明しており、日本語を含む多国語対応のディストリビューションとしての特徴も持っている。UTF-8は、Unicodeの符号化方式のひとつだが、ASCIIコードと互換性が高いためアプリケーション側での実装が容易であることなどから、UNIXでの一般的なロケールとなりつつあるものだ。なお、UTF-8を採用することで、Kondara 2.0は高い国際化が実現され、

画面 1 Kondara MNU/Linux 2.0で採用されたNautilusファイルマネージャとして機能するが、Webブラウザとしても機能する。



Kondara MNU/Linux 2.0





X Window Systemのログイン画面で各ロケールを選択できるようになっている。

さらに、UNIX系OSのディレクトリ構造を規定したFHSに準拠するなど、Kondaraの先進性が感じられる構成となっている。ただし、FHSに準拠したため以前とはディレクトリ構成が一部変更されたので、以前からのユーザーは注意が必要だ。

バンドルソフトウェアの充実

以前はバンドルソフトウェアの少なかったKondara MNU/Linuxだが、現在は多くのソフトウェアが付属するようになった。主なものは、日本語入力システムWnn6 Ver.3、日英・英日翻訳ソフトである翻訳魂、マルチブートユーティリティSystem Commander Lite、3DCGソフトであるShade for Linux体験版、EPSON KOWAのプリンタドライバなど、多くの商用ソフトウェアがバンドルされている。もちろん、フォントに関してもDynaFont5書体がバンドルされている。

Kondaraは、数少ないAlpha System用のディストリビューションでもある。今回も、同一パッケージにx86用とAlpha System用のバイナリを同梱している。Alpha System用のバンドルソフトには、COMPAQのC、

C++コンパイラや、Fortranコンパイラのほか、数値演算ライブラリなどが含まれている。

初心者には厳しくなった構成

多くのディストリビューションのインストーラでは、GUIを使ったAnacondaが使用されているが、Kondara 2.0ではGUIのインストーラは付属せず、テキストによるCUIのインストーラしか付属していない。これは、インストーラ作成の労力の割に、ユーザーに恩恵がないためだということだが、初心者ユーザーとすれば若干不安となる要素かもしれない。もっとも、CUIのインストーラもしっかり作られており、商用バンドルソフトのインストールについてもきちんと考慮されているなど、インストーラの完成度は高い(画面4)。さらに、インストール時にファイルシステムにReiserFSを指定できることや、8.4Gバイトを超えた領域にあるカーネルの読み込みが可能となるLBA32の指定も自動的に行われる。

なお、Kondara 2.0ではLinuxconfを始めとする多くの設定ツールは削除されてしまった。これによって、各種の設定を行うために直接設定ファイルを変更しなければならなくなってしまったものが多い。最新機能を盛り込んだために設定ツールの改良が間に合わ

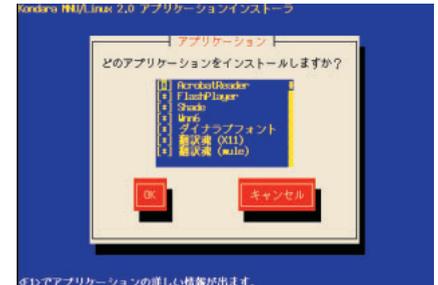


図1でアプリケーションの詳細情報が出ます。

画面4 CUIによるインストーラ 残念ながら、GUIによるインストーラは付属していないが、商用ソフトウェアのインストーラなども一緒に行ってくれるCUIインストーラが付属している。

なくなったためだと思われるが、こういったツールが利用できなくなると、初心者ユーザーには少し辛いかもしれない。Kondara Projectでは今後、設定ツールとしてWebminを採用する予定としているとのことなので、こちらに期待といったところだろう。

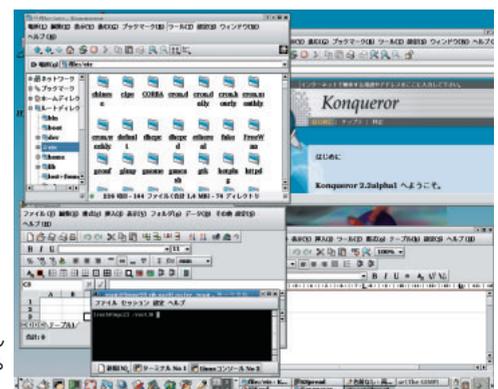
しかし、こうした点を除けばKondara MNU/Linuxは初心者でも使いやすいディストリビューションである。付属している2冊のマニュアルはわかりやすく、インストールから各種設定まで詳しく解説しているし、mphによる自動アップデート機能によって簡単にパッケージのアップデートができるようになってきている。

さらに、充実したサポート体制も魅力だ。ほかの多くのディストリビュータのサポートがインストールサポート程度なのに比べ、サポート内容と件数に関して制限なしのうえ、電話による受付というサポート体制は特筆すべき点だ。



画面2 GNOMEによるデスクトップ 標準Webブラウザには、Mozillaが採用された。画面では、Nautilusでディレクトリをブラウズしている。アンチエイリアスされた美しいフォントが目惹く。

画面3 KDEによるデスクトップ KDEでは、ファイルマネージャとしてKonquerorが採用された。こちらもWebブラウザとして機能する。



Vine Linux 2.1CR / 2.1.5

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

1万2800円

■ カーネル2.4

■ XFree86 4.x

■ GNOME 1.4

■ KDE 2.1

✓ 日本語

■ ReiserFS

■ ext3

✓ 8G超

充実した日本語環境を提供することを主眼に、Project Vineが開発しているのがVine Linuxだ。日本語の表示、入力はもちろん、manページやコマンドの出力メッセージの日本語化にも力を注ぎ、Linuxの初心者でも気軽に入門できるディストリビューションに仕上がっている。

レッドハットが販売する商用パッケージ版のVine Linux 2.1CRは、Vine Linux 2.1にWnn6や翻訳魂などの商用ソフトや、日本語TrueTypeフォントをバンドルした製品だ(表1)。これには、Intel x86アーキテクチャ版のほか、SPARC版、Power Macintosh版、Alpha版(開発バージョン)も含まれている。270ページからなるユーザーガイドには、インストール方法、システム設定や基本操作などがまとめられている。さらに、ユーザー登録後、30日間のサポートを受ける権利も付属する。

Vine Linux 2.1.xは、Red Hat Linux 6.2をベースとしており、高い互

換性を確保しているので、Red Hat Linux 6.2用のソフトウェアもほとんどがそのまま利用可能だ。

デスクトップ用途はVineにおまかせ

Vine Linuxは、ただ単に日本語対応ソフトウェアを集めたディストリビューションではなく、インストール直後の状態で問題なく日本語の入出力が行えるように設定されている。さまざまなサイズの日本語ビットマップフォントを用意し、ほとんどのアプリケーションが初期設定のままで日本語をきれいに表示する(画面1)。そんなことは当たり前のように思う方も多いだろうが、Linuxのディストリビューションではここまで丁寧に初期設定をしているものは少ないのが現状だ。

安定稼働を目指したシステム構成

主要なシステムコンポーネントを見てもみよう。Vine Linux 2.1CRは、カーネル 2.2.17、glibc 2.1.3、XFree86

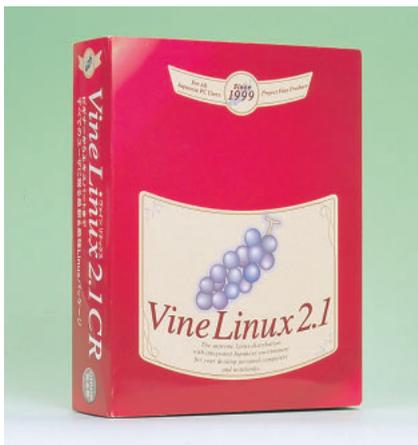
3.3.6を採用している。マイナーバージョンアップ版のVine Linux 2.1.5ではカーネルが2.2.18となっている。現在ではいずれも新鮮味のない構成だが、Vine Linuxをインストールしたシステムが安定して動作する理由ともなっている。

X環境では、標準のウィンドウマネージャにWindow Makerを採用し、軽快で実用的なデスクトップ環境を提供する。統合デスクトップ環境のGNOMEを選ぶことも可能なので、マシンパワーに余裕のある方はこちらを利用するのもよいだろう(画面2)。統合デスクトップ環境としてGNOMEと双璧をなすKDEは、Vine Linux本体のCD-ROMに収まらないソフトウェアを集めた「VinePlus」に収録されているので、別途インストールする必要がある。

初心者の強い味方

Vine Linuxには、Linuxに関する日本語の文書が数多く付属しており、

画面1 Vine Linuxのデスクトップ標準のウィンドウマネージャにはWindow Makerを採用。インストール直後からNetscape Navigatorの日本語表示がきれいなのはさすが。画面右側は、簡単に使えるメールクライアントソフトウェアのSylpheed。



Vine Linux 2.1CRのパッケージ



Namazuという全文検索エンジンを使って簡単に情報を検索することができる(画面3)。多くの人が悩む点というのはだいたい同じところと相場が決まっているので、先行者たちは問題の解決方法を文書に遺しているのだ。

また、WindowsやMacintoshを中心に利用してきたユーザーにとって、viやEmacsのようなUNIXのエディタは操作方法が独特で使いづらいかもしれない。そんなユーザー向けに、veditという簡易エディタが用意されている。簡易エディタとはいえ、設定ファイルを書き換える程度なら、十分な機能を持っている。

大きなマイナーバージョンアップ

FTP版の最新バージョンであるVine Linux 2.1.5は、Vine Linux 2.1をマイナーバージョンアップしたもので、基

本的な部分は大きく変更されていないものの、より完成度が高くなっている。もっとも大きな変更点は、ソフトウェアのアップデートツール「apt」を採用したことだろう。aptは、Debian GNU/Linuxの一部として開発されたツールで、インターネットを介してソフトウェアのアップデートを簡単に行えるのが特徴だ。

このほかに、インストーラが改良され、Linuxをハードディスクの8Gバイト以降の領域にインストールできるようになったのも大きな進歩だ。ブートローダのLILOには、ハードディスクの1024シリンダ以降(およそ8Gバイトになる)に置かれたLinuxカーネルを起動できないという制限があったが、最近のバージョンでは、“lba32”というオプションを作って問題を解決している。Vine Linux 2.1.5では、インスト

ーラでこのオプションを選べるようになっていた(Vine Linux 2.1CRでもLILO自体は新しいものなので、フロッピーディスクで起動してから/etc/lilo.confに1行“lba32”と書き加え、/sbin/liloを実行すればハードディスクから起動できるようになる)。

次期バージョンは年内登場か？

2001年8月21日、次期バージョンのVine Linux 2.5のベータ版の開発を始めるとのアナウンスがあった。開発パイロット版の「VineSeed」をベースに、年内のリリースを目指すという。Vine Linuxは、いたずらに最新機能を取り込んでシステムの安定を崩すことはしないというスタンスで開発を進めているが、動きの速いLinuxの世界において、Vine Linux 2.1.xはシステムコンポーネントやデスクトップ環境などがやや古く感じるようになってきたのも事実。次期バージョンの登場は、数多くのVineファンが待ち望んでいた。

ベータ版のスナップショットは、各ミラーサイトで公開されるということなので、不具合を発見したらProject Vineにバグレポートを送りVine Linuxの完成度向上に協力してほしい。

収録物	概要
インストールCD	Intel版、PPC版、SPARC版、Alpha版の4枚を収録
ソースCD	収録されるソースパッケージは各プラットフォームで共通
VinePlusCD	JGやKDEなどの追加パッケージを収録
アプリケーションCD	翻訳魂や各種試用版の商用ソフトを収録
ブート用FD	インストーラ起動用のフロッピー(ノートPC用もあり)
ユーザーガイド	インストールから活用方法までを解説
Wnn6 Ver.3	UNIX環境で定番の日本語入力プログラム
DynaFont 5書体	日本語表示のための商用フォント
HancomWord	Microsoft Word互換を目指すオフィスアプリケーション
翻訳魂	和訳、英訳ソフト

表1 Vine Linux 2.1CRの収録物



画面2 統合デスクトップ環境のGNOMEも利用可能
ウィンドウマネージャを切り替えるsetwmコマンドを使い、簡単にGNOME環境にスイッチできる。商用版のVine Linux 2.1CRには、DynaLabの日本語TrueTypeフォント5書体がバンドルされている。



画面3 資料は全文検索エンジンで検索
全文検索エンジンのNamazuを使ってJF (Japanese FAQ)などの文書をキーワード検索できる。検索フロントエンドのTkNamazuを使うと、目的の文書を簡単に探せる。

HOLON Linux 3.0

株式会社ホロン
http://www.holonlinux.com/

1万1800円~

✓ カーネル2.4
✓ XFree86 4.x
✓ GNOME 1.4
✓ KDE 2.1
✓ 日本語
■ ReiserFS
■ ext3
✓ 8G超

HOLON Linuxは、Windows版のタイピング練習ソフトなどでもおなじみのホロンから発売されているディストリビューションだ。

Linuxでのグラフィックス処理を全面に打ち出した「Linux 2000G」から、バージョン2.0のリリースに伴い現在のHOLON Linuxという名称に変更された。製品パッケージやWebサイトでは、グラフィックス処理だけでなく、使いやすさや豊富なアプリケーションなど、初級者向けのより汎用性の高いディストリビューションとしてアピールしている。

4種類の製品ラインアップ

HOLON Linuxは、Intel x86アーキテクチャとApple Power Macintoshをサポートしており、各アーキテクチャごとに別の製品パッケージ（「for PC/AT」と「for PPC」）として提供される。また、バンドルされる商用ソ



HOLON Linux 3.0 for PC/AT 豪華

フトウェアやユーザーサポートの範囲が異なる「豪華」と「並」という2タイプがあり、合計4種類の製品パッケージが用意されている。

最新バージョンであるHOLON Linux 3.0での各製品パッケージの概要を表1にまとめてある。購入時には、これら4つの製品から、ハードウェア環境、使用目的、予算などに合わせた選択が可能だ。

なお、以下のレビューでは特に記述していない限りHOLON Linux 3.0 for PC/AT 豪華を対象としている。

インストーラにひと工夫

ユーザーが最初に体験する特徴が、インストーラに見られる「ひと工夫」だ。Red Hat由来のインストーラAnacondaに独自の改良を加え、インストール先のディスク指定画面でパーティションサイズを変更できるほか、コマンドラインのパーティション設定ツール「parted」をインストーラから起動できるようになっている。「既存のパーティションのデータを損なうこと

なくディスクに空き領域を確保する」という操作をインストーラから行えるのは大きな利点である。

また、1024シリンダを超えるディスク領域へのインストールを可能にするために、LILOの設定画面に「LBA32」オプションの指定が追加されているのも改良点のひとつとして挙げられる。

なお、インストールに関して言えば、Disc 1からインストーラを起動した場合にはカーネル2.2.19が、Disc 2から起動した場合はカーネル2.4.5がインストールされる点もユニークである。ただし、この仕様によってCD-ROMの入れ替えが頻繁に発生しているのは、いただけない。今後の改善を望みたいところである。

豊富なパッケージ群

収録パッケージの「豊富さ」というのが、ディストリビューションとしてのHOLON Linuxの特徴のひとつ。

バンドルされる商用ソフトウェアは、日/英翻訳ソフト「翻訳魂」、かな漢字変換ソフト「ATOK12 SE」と

	for PC/AT 豪華	for PC/AT 並	for PPC 豪華	for PPC 並
ATOK12 SE (かな漢字変換)				
Wnn6 Ver.3.0 (かな漢字変換)				
DynaFont (日本語TrueType)	15書体	5書体	15書体	5書体
翻訳魂				
OSS商用サウンドドライバ				
サポート	90日間無制限の電話、FAX、メールサポート	90日間無制限のメールサポート	90日間無制限の電話、FAX、メールサポート	90日間無制限のメールサポート
価格	1万1800円	3980円	9800円	4800円

表1 HOLON Linux 3.0の製品構成と商用バンドルソフト



「Wnn6 for Linux Ver.3.0」、サウンドドライバ「OSS」、TrueTypeフォント「DynaFont15書体」となっている。これだけなら、フォントの書体が多い程度で、特にほかのディストリビューションと変わらないが、GPLに従ったオープンソースソフトウェアや独自のライセンスで無償配布されているソフトウェアの中から、評価の高いものをチョイスして積極的に採用しているのがポイントだ。

たとえばグラフィックス・マルチメディア関連では、定番のGIMPを始め、3Dレンダリングソフト「Blender」、Video4Linux対応のビデオ編集ソフト「Broadcast2000」、デジカメからのデータ転送をGUI操作で行える「gPhoto」、ストリーミング再生の「RealPlayer7」、CDリッパー/MP3エンコーダ「Xripper」、CD-Rライティングソフト「X-CD-Roast」など、充実したパッケージ構成となっている。

このほかにも、AcrobatReader (PDFファイルリーダー)、Sylpheed (メール)、Bluefish (HTMLエディタ)、JPilot (Palm Pilot用ユーティリティ)、iwashi (アプリケーションランチャ)といった多彩なアプリケーション、ユーティリティがバンドルされている。

ゲームに力を入れているのも特徴で、「さめがめ」や「上海」など多くのディストリビューションに含まれている定番ものだけでなく、「Cannon Smash」、「TuxRacer」、「Tux Kert」などの3Dゲームも標準で収録されている。

こうしたパッケージ構成からは、Linux 2000Gのころからの売りである「グラフィックス処理にLinuxを!」というコンセプトを引き継ぎつつ、ほかのカテゴリーのアプリケーションも充実させることで、幅広い用途への対応を目指すという方向性が感じられる。

aptによる アップデートが可能

HOLON Linuxでは今回のバージョン3.0から、Debian GNU/Linuxで使われているパッケージ管理ツールとして知られる「apt」を採用している。さらに、X上で動作するaptのGUIフロントエンド「synaptic」もバンドルされており、aptの持つパッケージ管理機能をわかりやすいインターフェイスで活用できる (画面2)。

ただし、インストール直後の状態ではsynapticがうまく動作しない。これは、aptのバージョンによる既知のバグが原因であるため、修正にはapt自体

のアップデートが必要だ。ホロンのユーザーサポートページに詳しい解決方法が掲載されているので、そちらを参照するとよいだろう (http://www.holonlinux.com/support/manual/apt.html)。

インストーラの改良やaptの採用に代表されるように、ユーザーにとって便利であると思われるものを積極的に採用する姿勢は大きなプラスポイントだろう。また、マルチメディア系アプリとゲームソフトの充実、入門者にとってLinuxに対する間口を広げる役割を果たすはずだ。

一方で残念なのが、日本語入力やフォントなどの設定に不十分な部分があるといった作り込みの甘さが目立つところだ。ただ、この点についても、Webサイトとユーザーメーリングリストによるサポートの強化を図っているようなので、ユーザーからのフィードバックを活かし、さらなる改善が行われるよう期待したい。



画面3 iwashi
名前のとおり鰭の頭と尾びれを持ったユニークなランチャソフトだ。fvwmなどランチャの設定が面倒なウィンドウマネージャで使うとよいかもしれない。ただし、動作にはGTK+ライブラリが必要だ。



画面1 改良されたインストーラ
パーティション設定のオプションとして「リサイズ」ボタンが追加された。インストーラからパーティションのサイズを変更できるので、場合によってはとても重宝するはずだ。



画面2 synaptic
aptのGUIフロントエンドとして動作する。インストールはもちろん、RPMパッケージのインストール状況やアップデートバージョンの有無を確認できる。ネットワークだけでなく、CD-ROMからのアップデートも可能。

Debian GNU/Linux 2.2r3

Debianプロジェクト
<http://www.debian.org/>

■ カーネル2.4

■ XFree86 4.x

■ GNOME 1.4

■ KDE 2.1

✓ 日本語

■ ReiserFS

■ ext3

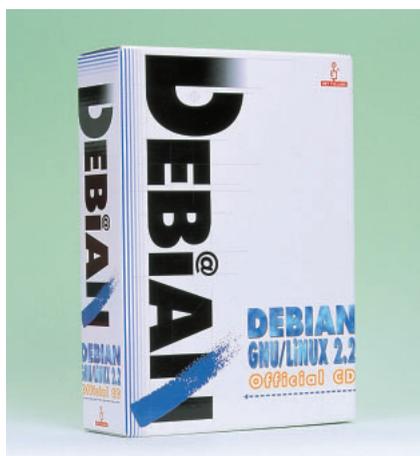
✓ 8G超

Debian GNU/Linux (以下、Debian) は、世界各国の400人以上のボランティア開発者が開発するディストリビューションだ。

Red Hat系のディストリビューションがパッケージ管理方式にRPMを採用しているのに対して、Debianはdpkgというパッケージ管理方式を採用しているのが特徴だ。また、日本語対応のパッケージが数多く収録されているので、日本語デスクトップ環境としても使用可能だ。

現在の安定版はDebian 2.2r3で、カーネル2.2.19、glibc2.1.3、XFree86 3.3.6を基本コンポーネントとして採用している。また、GNOMEのバージョンは1.0.55である。

Debian 2.2がリリースされた頃は、KDEの動作に必要なQtライブラリの配布問題が解消されていなかったのに、Debian 2.2にはKDEの公式パッケージは用意されていない。



Debian GNU/Linux 2.2
 ネットビルド版のパッケージ

Debianには、Red Hat Linuxのような公式の発売元が存在せず、CD-ROMを使ってインストールする場合は、CDのイメージファイル作成用のスクリプトを使って作成するか、表1のベンダーからCD-ROMを購入する必要がある。

高機能パッケージ管理ツールapt

Debianでは、aptというツール群を使って、パッケージのインストールやアンインストールを行う。

aptは、dpkgのフロントエンドで、ユーザーは、CD-ROM、FTPサーバ、Webサーバのどこにパッケージがあるのかを意識することなくパッケージにアクセスできる。

たとえば、リレーショナルデータベースPostgreSQLをインストールする場合、まずapt-cacheコマンドを使ってパッケージを検索する(画面1)。

PostgreSQLのパッケージが見つかったら、apt-getコマンドを使ってパッケージをインストールする。

PostgreSQLを動作させるのに必要なライブラリなどがインストールされていない場合は、PostgreSQLパッケージとあわせて必要なパッケージがインストールされるので、ユーザーがパ

ッケージの依存関係に悩むことはない。

この要領で、標準の2.2系カーネルを2.4系のカーネルにアップグレードできたり、システム全体を新しいバージョンにアップグレードできたりするところは、Red Hat系ディストリビューションには見られないDebianシステムの特長と言える。

さて、Debianをはじめとする多くのLinuxディストリビューターは、セキュリティホールが見つかったプログラムに対策をほどこして、アップデート用のパッケージを日々FTPサーバなどにアップロードしている。

しかし、どのアップデートパッケージをダウンロードしてインストールしてよいかわからず、セキュリティホールを放っておくユーザーは少なからず存在する。

apt-getコマンドは、システムにどのパッケージがインストールされているのか、どのパッケージがアップデートされたのかなどの情報をデータベースに照会して、パッケージの依存関係の解消まで面倒をみてくれるので、Debianユーザーは、何も考えずにapt-getコマンド一発で、それらのアップデートパッケージをインストールできる(画面2)。

Debian関連サイト	URL	概要
Debian JP Project	http://www.debian.or.jp/	日本のDebianプロジェクト
Debian Planet	http://www.debianplanet.org/	Debian情報が豊富なポータルサイト
有限会社ハイパーコア	http://www.hypercore.co.jp/	DebianのCD-ROMを販売
ネットビレッジ株式会社	http://www.netvillage.co.jp/	DebianのCD-ROMを販売
よしりん合名会社	http://www.shishamo.com/	DebianのCD-ROMを販売

表1 Debianの関連サイト



パッケージを検索するコマンド

```

kterm
debian:~$ apt-cache search --names-only postgresql
postgresql-slink - Package to ease upgrade of postgresql from Debian 2.1 to 2.2
postgresql-contrib - Additional facilities for PostgreSQL
postgresql-test - Regression test suite for PostgreSQL
postgresql-pl - A procedural language for PostgreSQL
postgresql-dev - Header files for libpq (postgresql library)
postgresql - Object-relational SQL database, descended from POSTGRES.
postgresql-doc - Documentation for the PostgreSQL database.
postgresql-client - Front-end programs for PostgreSQL
odbc-postgresql - ODBC support for PostgreSQL
debian:~$
debian:~$
debian:~$ sudo apt-get install postgresql
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  libpq5 postgresql-client
The following NEW packages will be installed:
  libpq5 postgresql postgresql-client
0 packages upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 888kB of archives. After unpacking 2470kB will be used.
Do you want to continue? [Y/n]
Get:1 http://ring.asahi-net.or.jp dists/proposed-updates/ libpq5 6.5.3-27 [633kB]
Get:2 http://ring.asahi-net.or.jp dists/proposed-updates/ postgresql 6.5.3-27 [637kB]
Get:3 http://ring.asahi-net.or.jp dists/proposed-updates/ postgresql-client 6.5.3-27 [637kB]
69% [2 postgresql 520720/637kB 75%]
  
```

パッケージをインストールするコマンド

postgresqlに必要なパッケージ

画面1 aptを使ったパッケージの検索とインストール
apt-cache コマンドを使ってパッケージを検索し、apt-get コマンドを使って目的のパッケージをネットワーク経由でインストールしている。apt-get コマンドを使えば、インストールするパッケージに必要なパッケージも一緒にインストールされる。

システムをアップデートするコマンド

```

kterm
debian:~$ sudo apt-get -f upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
22 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 26,996/32,796 of archives. After unpacking 3492kB will be used.
Do you want to continue? [Y/n]
Get:1 http://ring.asahi-net.or.jp dists/proposed-updates/ xenocs21-basesupport 1999.12.15-1.2 [17,308]
Get:2 http://security.debian.org stable/updates/main apache 1.3.9-14 [359kB]
Get:3 http://security.debian.org stable/updates/main apache-common 1.3.9-14 [719kB]
Get:4 http://security.debian.org stable/updates/main libwvaster1 0.61.1-4.1 [76,6kB]
Get:5 http://security.debian.org stable/updates/main samba 2.0.7-3.4 [1039kB]
Get:6 http://security.debian.org stable/updates/main samba-common 2.0.7-3.4 [570kB]
Get:7 http://security.debian.org stable/updates/main samba-doc 2.0.7-3.4 [1730kB]
Get:8 http://security.debian.org stable/updates/main smbclient 2.0.7-3.4 [560kB]
Get:9 http://security.debian.org stable/updates/main smbfs 2.0.7-3.4 [351kB]
Get:10 http://security.debian.org stable/updates/main swat 2.0.7-3.4 [233kB]
Get:11 http://security.debian.org stable/updates/main w3m 0.1.10-0.1 [1196kB]
Get:12 http://security.debian.org stable/updates/main wv2k 0.61.1-4.1 [1523kB]
Get:13 http://security.debian.org stable/updates/main exim 3.12-10.1 [851kB]
Get:14 http://security.debian.org stable/updates/main rrvt-ml 1:2.6.2-2.1 [263kB]
Fetched 20,886 in 6m24s (64,1kB/s)
(データベースを読み込んでいます... 現在 26218 個のファイルとディレクトリがインストールされています。
 locales 2.1.3-18 を(../locales_2.1.3-18_i386.deb で) 置換するための準備をしています...
 locales を展開し、置換しています...
 libc6-dev 2.1.3-18 を(../libc6-dev_2.1.3-18_i386.deb で) 置換するための準備をしています...
 libc6-dev を展開し、置換しています...
 libc6 2.1.3-18 を(../libc6_2.1.3-18_i386.deb で) 置換するための準備をしています...
 libc6 を展開し、置換しています...
  
```

アップデートされるパッケージ

画面2 aptを使ったシステムのアップデート
セキュリティホールが見つかるなどしてアップデートされたパッケージを、apt-get コマンドを使ってまとめてインストールしてシステム全体をアップデートしている。

apt-get を使えば、システムを最新状態に保てるというわけだ。

Linux 初心者には不向き

apt ツール群を使っただけの保守性を誇る Debian だが、Red Hat 系ディストリビューションと比較すると、初心者から見た敷居は高い。

インストーラはテキストベースで英語表示のみだし、ネットワークカードやビデオカードの自動認識もない。

また、現在は Red Hat 系ディストリビューションが事実上の標準になっているので、Red Hat 系ディストリビューションと比べて活用方法などの情報を入手しにくい。

システムの設定に関して、Red Hat 系ディストリビューションに付属するような GUI ツールはほとんどなく、多くの場合、エディタを使って直接設

定ファイルを編集することになる。

ハードウェアの自動認識や、GUI 設定ツールが当然の Windows 環境から移行してきたユーザーが Debian を使うのは、ちょっと無理があるだろう。

apt ツール群を使って高い保守性を体感してみたいが、インストールや設定が難しいのはちょっと、という読者は、Debian をベースにして独自のインストーラや設定ツールが用意されている Omoikane GNU/Linux や Projeny Debian を使ってみるのがよいだろう(表2)。

次期バージョンのリリースはいつだ？

Debian には、stable、testing、unstable という3つの開発ブランチが存在する。

stable は現行の安定版で、現在は Debian 2.2 (コードネーム potato) が

それに当たる。unstable は次期バージョンのための開発版(コードネーム sid)で、testing は、unstable に収録されるパッケージで、動作の安定性が確認されたもののみを集めた「開発版の安定版(コードネーム woody)」といった位置付けた。

現在、Debian は次期バージョンのリリースに向けて、testing (woody) がフリーズ段階に入っている。

woody は、カーネル 2.2.19、glibc 2.2.4、XFree86 4.0.3 を基本コンポーネントとして、また、GNOME 1.4.0 と KDE 2.1.1 をデスクトップ環境として開発が進められている。

Debian プロジェクトは、「クリスマス前をめどに、woody を Debian 3.0 としてリリース予定だ」とアナウンスしているので、開発が順調に進めば年内に、新しい Debian 3.0 がお目見えするだろう。

ディストリビューション	URL	概要
Omoikane GNU/Linux	http://www.omoikane.co.jp/	日本のベンダーが開発し自家 Debian とほぼ完全な互換性を有する
Projeny Debian	http://www.projeny.com/	GUI のインストーラや設定ツールなどを採用して Debian 2.2 を独自拡張
プロサーバ Linux	http://www.fujifilm.co.jp/fmd/linux/lintop.html	サーバ用途に特化され、独自ツールを使った集中管理が可能

表2 Debian ベースのディストリビューション
上記のディストリビューションは、すべて Debian 2.2 をベースにして開発されており、独自のインストーラや設定ツールが付属する。

Plamo Linux 2.2.1

Plamo LinuxのWebページ
<http://www.linet.gr.jp/~kojima/Plamo/>

■ カーネル2.4

■ XFree86 4.x

■ GNOME 1.4

✓ KDE 2.1

✓ 日本語

■ ReiserFS

■ ext3

✓ 8G超

Plamo Linuxは、Slackware 3.xをベースにして、日本語化を行ったディストリビューションだ。メンテナーはこじまみつひろさんと、氏を中心に、ボランティアベースで開発が進められている。大きな特長のひとつは、NECのPC-98x1シリーズへの対応だ。1枚のCD-ROMに、IBM PC/AT互換機用とPC-98x1用のパッケージが両方含まれている。

Plamo Linuxの起源となったSlackware 3.xは、すでに古いディストリビューションであるが、Plamo Linuxは、Slackwareとは独立した進化を遂げている。最新版のPlamo Linux 2.2.1がリリースされたのは2001年の6月で、主なシステムコンポーネントのバージョンは、カーネル2.2.19、glibc 2.2.2、XFree86 3.3.6となっている。また、ユーザーの有志により、GNOME 1.4や、XFree86 4.0、各種パッチを適用したXFree86 3.3.6のバイナリパッケージなどが作成され、イン

ターネットを通じて公開されている。

実用重視のディストリビューション

インストーラは、テキストベースの日本語表示のもので、動作は快適だ。デスクトップマシン用とノートPC用のお勧めパッケージを選ぶことが可能なので、これらを利用すれば個別パッケージの選択に迷わずに済む。

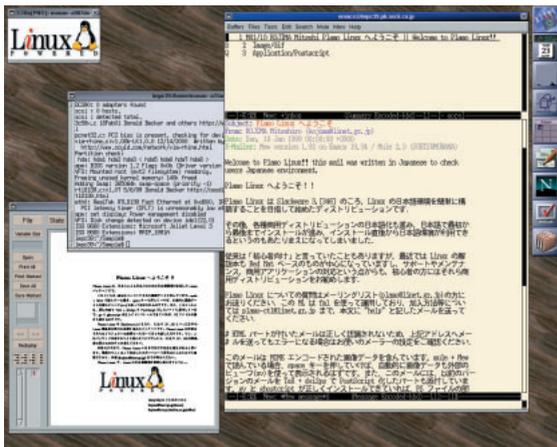
標準のウィンドウマネージャはAfterStepだ(画面1)。見た目は派手だが、動作が鈍い統合デスクトップ環境ではなく、実用的なウィンドウマネージャを採用しているところはPlamo Linuxの性格をよく表しているといえよう。統合デスクトップ環境を利用したければ、CD-ROMのcontribディレクトリにKDE 2.1.1も用意されているので、これをインストールすればよい(画面2)。

ソフトウェアパッケージは、Slackwareからの伝統を受け継ぎ、“バイナ

リ+設定用シェルスクリプト”のtar玉となっている。RPMやdpkgのように、ソフトウェア間の依存関係などは管理しないので、初心者には取っつきにくいかもしれないが、ソフトウェアをソースからビルドしてインストールすることに慣れているユーザーにとってみれば、パッケージ管理システムのことを気にしないで済むメリットは大きいだろう。

ツールとしてのLinux

Plamo Linuxでは、「なにかの設定をする エディタで設定ファイルを書き換える」ということなので、まったくのLinux初心者にはお勧めしない。しかし、わからないことは自分で調べて解決しようという気概のある方なら、文書・グラフの作成やインターネットアプリケーションの利用など、日常作業を行うための便利な環境を構築することが可能だ。使い込むほどに手になじむ、Plamo Linuxはそんなディストリビューションだ。



画面1 Plamo Linuxのデスクトップ標準のウィンドウマネージャとして、AfterStepを採用している。インストール直後から日本語環境が整っているのがうれしい。

画面2 KDE 2.1.1をインストールしたCD-ROMのcontribディレクトリには、KDEやNetscape Communicatorのバイナリパッケージが収録されている。





Linux MLD 5

メディアラボ株式会社
http://www.mlb.co.jp/

9800円

✓ カーネル2.4 ✓ XFree86 4.x ■ GNOME 1.4 ■ KDE 2.1 ✓ 日本語 ■ ReiserFS ■ ext3 ✓ 8G超

Linux MLD 5は、昨年12月にいち早くカーネル2.4を採用したディストリビューションである。Red Hat Linuxをベースに、カーネル2.4-test8、glibc 2.2、XFree86 4.0.1を採用している。

デスクトップ環境はGNOME 1.2.1、Webブラウザmozilla-M18、メーラ Sylpheed、グラフィックツールGimp、かな漢字変換にはWnn6が、日本語 TrueType フォントにはDynaFont 5 書体が標準でインストールされる。付属するCD-ROMからKDE 2.0、Netscapeなどをインストール可能だ。

Linux MLDの特徴は、WindowsユーザーがLinuxを容易にインストールできるように工夫されている点だ。Windows環境にMLD 5のインストールCDを挿入することで、自動的にインストーラsetup.exeが起動される(画面1)。

インストールするディレクトリ(この場合はC:¥LOOP)と、Linuxに割り当てるディスク容量を指定して、OKボタンを押せばインストールが行われる。

インストールが終了すると、次にLinuxブート用フロッピーの作成画面になる。あとは、Windowsを終了し、先ほど作成したブートフロッピーから起動すれば、MLD 5が立ち上がる。

通常、WindowsマシンにLinuxをインストールしてデュアルブートで使用するには、まずハードディスクのWindows用のパーティションを小さくしてLinux用の空き領域を作る必要がある。そして、Linuxのインストーラでは、そのマシンに応じたネットワークやX Window Systemの設定を行わなければならない。

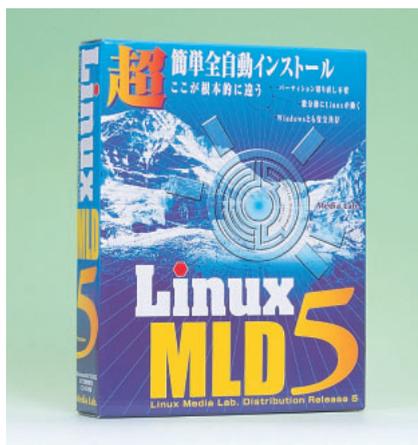
しかし、MLD 5では、Windowsでの設定値を自動的に取得して設定するため、ユーザーがハードウェア構成を入力する必要がない。また、WindowsのFATファイルシステム内に、Linuxのファイルシステムをイメージファイルとして作成するため、ハードディスクのパーティションを切り直してLinux専用領域を作らなくて済むことから、Linux初心者でも気軽にインストールできるディストリビューションだといえる。

し、フレームバッファコンソールを利用するようになった。さらにFATだけではなくWindows NT / 2000のNTFSにもインストールできるようになり、Windows 2000のntloaderから起動可能に改良されている。

標準でインストールされるソフトは、X Window System (ウィンドウマネージャはBlackBox)、Webブラウザはコンソールベースのw3m、メーラはmuttといった基本的なものに限られているが、少ないメモリ(32Mバイト)/CPUパワー(Pentium 166MHz)のマシンでも軽快に動作する。

Linux MLD miniは、オンライン販売で提供され、価格は3000円。今後、GNOMEやKDE、開発環境、サーバソフトなどを機能別のCDで順次発売していく予定だ。

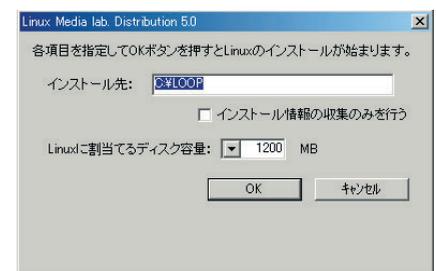
Linux MLDでは、Windows上にインストールした場合のディスクアクセスを高速化するために、loopデバイスにブロックのremapを実装している。FATやNTFSドライバを経由せずにアクセスするため、余計なシークを減らし、ext2と変わらないパフォーマンスを得ている。



Linux MLD 5

Linux MLD mini

Linux MLD miniは、MLD 5と同様にWindows上にインストールが可能で、インストール領域として100Mバイト程度で済むようにコンパクトに設計されているディストリビューションだ。カーネル2.4.3、XFree86 4.0.3を採用



画面1 Windows上で動作する専用インストーラ

Omoikane GNU/Linux 1.2

オモイカネ株式会社

<http://www.omoikane.co.jp/>

■ カーネル2.4

■ XFree86 4.x

■ GNOME 1.4

■ KDE 2.1

■ 日本語

■ ReiserFS

■ ext3

■ 8G超

Omoikane GNU/Linux 1.2 (以下、OGL 1.2) は、オモイカネがDebian GNU/Linux(以下、Debian)をベースに開発するディストリビューションだ。

OGL 1.2は、基本コンポーネントにカーネル2.2.19、glibc2.1.3、XFree86 3.3.6を採用している。また、デスクトップ用の「WorkStation」をはじめ、表1のように4種類のバージョンが存在する。

OGL 1.2は店頭販売されておらず、FTPサイト(たとえば、<ftp://ftp.kddlabs.co.jp/>)からCD作成用のイメージファイルをダウンロードするか、オモイカネのWebサイトから通信販売を利用して入手する。

ただし、「WorkStation++」はFTP配布されておらず、オンライン販売のみだ。また、オモイカネのWebサイトでは、サポート付きのCD-ROMも販売されている。

Debian環境をらくらく構築

OGL 1.2が開発のベースとしているDebianは、依存関係を考えながらパッケージをインストールするaptというツールを備えており、システムの保守性

パッケージ名	概要
WorkStation	Webブラウズなど一般的なデスクトップ用
WorkStation++	WorkStationに多くのアプリケーションを追加(オンライン販売のみ)
Server	qmailやOpenLDAPなどを採用するサーバ用のパッケージ
X-term	CD-ROMから起動して使うX端末用パッケージ

表1 Omoikane GNU/Linux 1.2のラインナップ
通信販売で購入する場合、WorkStation、Server、X-termの3つはCD-ROMが1枚で1000円、CD-ROMが2枚のWorkStation++は1600円だ。CD-ROMにサポートが付く製品も5000円から販売されている。

が高い。

しかし、Debianのインストーラは英語表示であるうえに、ハードウェアの自動検出機能がないので、Linux初心者にはいささか敷居が高い。

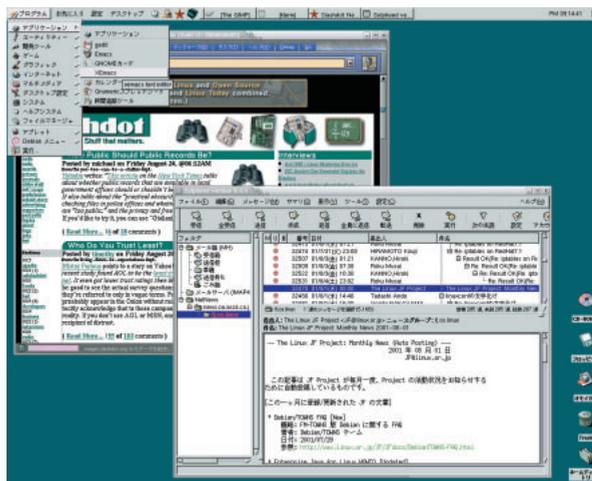
また、Debianに収録されているパッケージのバージョンは古く、新しいコンポーネントを試したいユーザーにとっては物足りない構成だ。

そこで、OGL 1.2はDebianの保守性の高さをそのままに、インストーラなどDebianの短所を埋める方向で開発された。

OGL 1.2のインストーラは、日本語で表示されるテキストベースのもので、ネットワークカードなどのハードウェアの自動認識に対応しているし、インストールされるパッケージがあらかじめ決められているので、パッケージの選択で迷うことがない。

また、オモイカネにより独自にパッケージングされた日本語対応のGNOME 1.2(画面1)や、レンダリン

画面1 OGL 1.2のデスクトップ
Debianに収録されているGNOMEのバージョンは1.0系と古いので、オモイカネは、GNOME 1.2を独自にパッケージングしてデスクトップ環境を強化している。



グ能力をチューニングしたWebブラウザのMozillaが収録されており、インストール直後から日本語デスクトップ環境として使える。

このほかにも、独自のX設定ツールや、セキュアで高速なメール配送エージェントqmailが収録されており、Debian入門者から上級者までが使える構成だ。

次期バージョンは?

現在、オモイカネは次期バージョンとなるOGL 2.0を開発中だ。

OGL 2.0は、GUIインストーラの採用、ジャーナリングファイルシステムのサポート、カーネルとライブラリのバージョンアップがなされ、現在複数あるOGLのバージョンが一本化されて、10月にリリースされる予定だ。

また、OGL 2.0からはFTP版と製品版が用意され、店頭販売も始められるとのことだ。



SuSE Linux 7.2

SuSE Inc.
<http://www.suse.com/>

29.95 USドル~

- カーネル2.4
- XFree86 4.x
- GNOME 1.4
- KDE 2.1
- 日本語
- ReiserFS
- ext3
- 8G超

SuSE Linux 7.2 (以下、SuSE 7.2) は、ドイツのSuSE Inc.が開発するディストリビューションだ。

SuSE Linuxは、新しいパッケージを意欲的に採用するのが特徴で、SuSE 7.2の基本コンポーネントには、カーネル2.4.4、glibc2.2.2、XFree86 4.0.3が、デスクトップ環境には、KDE 2.1.2とGNOME 1.4が採用されている。また、Sambaもいち早くバージョン2.2が収録されている。

高機能なGUIツール

SuSE 7.2には、高機能で使いやすいインストーラとGUIツールが付属する。

グラフィカルなインストーラ自体は珍しくないが、SuSE 7.2のインストーラは、すでにインストールされているWindows 9x/Meの領域サイズを変更してLinux用の領域を確保できる(画面1)。また、複数のパーティションを仮想的に1つのドライブとして扱うLVM領域を作成できるなど、初心者と

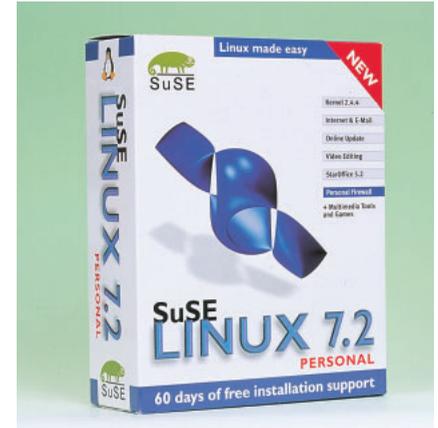
上級者の両方に向けた機能拡張がなされている。

さて、標準の設定ツールYaST2(画面2)は、Windowsのコントロールパネルのように各設定項目が統合されており、Linuxの設定ファイルやコマンドの扱いに不慣れなユーザーでもWindows環境と同じような感覚でシステムを設定できる。

このほかにも、ウィザード形式のX設定ツールSaX2が付属する。SaX2は、Windowsの画面の「プロパティ」に近い感覚で操作でき、3Dハードウェアアクセラレータを有効にするモジュールの設定もできる。

日本での入手に難あり

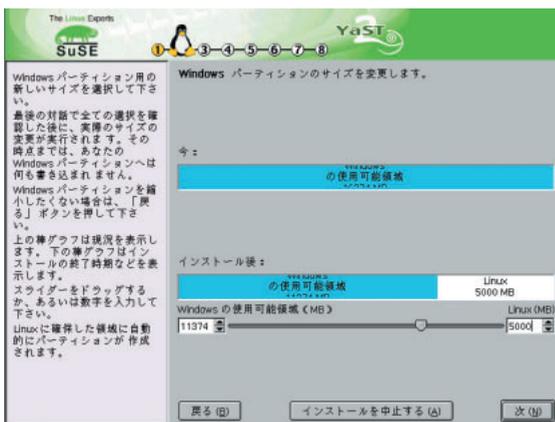
SuSE 7.2は、デスクトップ向けのPersonalや、上級者向けのProfessionalをはじめとする製品ラインナップで構成される。また、製品にはドイツ市場向けのGermanと、各国語に対応する2種類のInternational版の、あわせて3



SuSE Linux 7.2 Personal

種類のバージョンがあり、このうちInternational-2が日本語環境をサポートしている。

しかし、SuSE 7.2が標準で収録しているオフィスアプリケーションStarOfficeの日本での商標問題により、SuSE Inc.は、製品を日本へ輸出できない。ジャーナリングファイルシステムなどの新しい技術と、安定した環境を同時に提供する魅力的なディストリビューションだけに、今後この問題の解決が望まれるところだ。



画面1 SuSE Linux 7.2のインストーラ
SuSE Linuxのインストーラは、マウスを使ってFAT32のWindows領域を縮められる。



画面2 設定ツールYaST2
このツールを使えば、Windowsユーザーでも違和感なくシステムを設定できる。

Meister Linux Mandrake 7.2

レーザーファイブ株式会社 <http://www.laser5.co.jp/> 6800円

- カーネル2.4
- ✓ XFree86 4.x
- GNOME 1.4
- KDE 2.1
- ✓ 日本語
- ✓ ReiserFS
- ext3
- ✓ 8G超

Linux Mandrakeは、フランスのMandrake SAが開発しているディストリビューションで、Red Hat Linuxをベースに、独自のインストーラや設定ツール、各種のフリーソフトウェアをバンドルしている。

ここで紹介する Meister Linux Mandrake 7.2は、Linux Mandrake 7.2に、Wnn6をはじめとする商用アプリケーションソフトウェアや、日本語TrueTypeフォント、レーザーファイブ独自の日本語対応強化キットをバンドルしたパッケージだ。低価格である代わりに、サポートは提供されない。

RPM系パワーユーザー向け

標準デスクトップ環境にはKDE 2.0.1を採用(画面1)。GNOME 1.2の環境を選ぶことも可能だ。標準でインストールされるカーネルのバージョンは2.2.17、標準Cライブラリのglibcはバージョン2.1.3となっており、最新の環境とはいええないものの、十分に安定し



ていて実用上の問題はない。

Meister Linux Mandrake 7.2には、カーネル2.4にアップグレードするキットが同梱されているうえ、rpmfind.net (<http://www.rpmfind.net/>) を通じて、各種ソフトウェアの最新RPMパッケージを入手できるので、高速なインターネット接続環境をお持ちのパワーユーザーなら、最先端のLinux環境を手にすることも可能だろう。Mandrake用のRPMファイルは、Cookerと呼ばれる開発者たちによって日夜作られ、発表されている。

また、インストールされるプログラムは、Pentium以上のCPU向けの最適化オプションをつけてコンパイルされているので、該当するCPUを搭載したマシンでは若干パフォーマンスが高くなる。

ちょっと物足りない日本語化キット
Linux Mandrake 7.2は、オリジナ

画面1 Meister Linux Mandrake 7.2のKDEデスクトップ
Red Hat Linuxとの高い互換性を保ちながら、より使い勝手を向上させた「ベターRed Hat」として高い人気を誇る。

Meister Linux Mandrake 7.2のパッケージ

ル版で国際化されているので、インストール時から日本語を選択できる。日本語対応強化キットは、Linuxが起動してからインストールすることになるのだが、このキットには「KDEのメニューなどで文字が化ける」、「日本語TrueTypeフォントがインストールされない」など、いくつかの不具合がある。そのため、レーザーファイブのWebページを参照して設定を修正する必要がある(<http://www.laser5.co.jp/package/mandrake/fault/index.html>)。

Linux Mandrake 8.0も登場
オリジナルバージョンの最新版は、Linux Mandrake 8.0になっており、こちらはカーネル 2.4.3、glibc 2.2.2、KDE 2.1.1、GNOME 1.4を採用している。しかし、8.0のインストーラには不具合があり日本語が表示できなくなっているのが残念だ。完全日本語対応版の登場が期待される。





OpenLinux Workstation 3.1

Caldera International, Inc.
<http://www.caldera.com/>

99.00 USドル (ノンサポート版は59.00 USドル)

✓ カーネル2.4 ✓ XFree86 4.x ■ GNOME 1.4 ✓ KDE 2.1 ■ 日本語 ■ ReiserFS ■ ext3 ✓ 8G超

OpenLinux Workstation 3.1は、米国のCalderaが開発したデスクトップ向けディストリビューションの最新版だ。今回のレビューでは英語版を使用しているが、現在、日本語化の作業が進められており、順調にいけば今秋にも日本語対応バージョンがリリースされる予定となっている。

なお、英語版は、ぷらっとホーム株式会社で取り扱っている。オンライン販売も行っているため、入手したい方は同社のWebサイト (<http://www.plathome.co.jp/>) を参照してほしい。ちなみに、オンラインでの販売価格は7980円だ。

主要コンポーネントを一新

OpenLinuxといえば、日本語版もリリースされたバージョン2.3の頃から、独自のGUIインストーラとさいもんフォントを採用した整備されたデスクトップ環境を備えた特徴のあるディス

トリビューションとして定評があった。日本でも固有の支持層を得ていたが、その次のバージョンであるOpenLinux eDesktop 2.4は残念ながら日本語化されなかった。

OpenLinux Workstation 3.1は、そのeDesktopの後継にあたり、以下のよう主要コンポーネントがアップグレードされている。

- カーネル2.4.2
- glibc 2.2.1
- XFree86 4.0.2
- KDE 2.1

標準デスクトップ環境はKDEで、通常のインストールでは、ほかのデスクトップ/ウィンドウマネージャはインストールされないが、パッケージに同梱されているオプションCD-ROM (Supplemental Open Software CD) から、GNOME (1.2) やEnlightenmentなども追加インストールすること

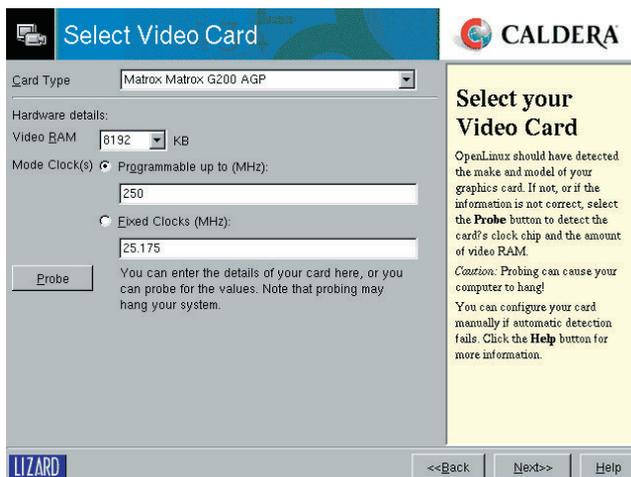
ができる。オプションCDには、このほかにも、PostgreSQL、ProFTPD、Mozilla、VNCといった、さまざまなジャンルの500を超える追加パッケージが収録されている。

幅広い用途に対応

プログラム開発プラットフォームとしての機能を強化しているのもOpenLinux Workstation 3.1の特徴のひとつだ。JBuilder、Forteといった商用の統合開発環境をバンドルし、Linuxの標準的なプログラム開発ツールやKDE / Qtのネイティブな開発ツール (Kdevelop、Qt Designer)、Java SDKなどとあわせて、充実したプログラム開発環境が用意されている。

もちろん、標準デスクトップがKDEなので、KOfficeを始め、KMail、Konquerorといったアプリケーション/ユーティリティもひと通りそろっており、通常のデスクトップ用途でも十分に使えるパッケージングとなっている。

また、ネットワーク上のLinuxシステムを集中管理するサーバソフトウェア「Volution」のクライアントエージェントを含んでおり、OpenLinux Server 3.1 + Volutionと組み合わせることで、中・大規模なサイトにおけるトータルなシステム管理ソリューションが構築できるなど、非常に幅広い用途に対応したディストリビューションである。



画面1 LIZARD
OpenLinux独自のGUIインストーラ「LIZARD」のビデオカード設定画面。基本的にハードウェアは自動認識されるので、ほとんどマウス操作のみでインストールが完了する。待ち時間にゲームで遊べるのは、前のバージョンでもおなじみのユニークな機能だ。今回は「ソリティア」が楽しめる。

Slackware Linux 8.0

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

39.95 USドル

✓ カーネル 2.4

✓ XFree86 4.x

✓ GNOME 1.4

✓ KDE 2.1

■ 日本語

■ ReiserFS

■ ext3

✓ 8G超

Slackware Linuxは、現存するディストリビューションの中でもっとも長い歴史を誇るもののひとつだ。Linuxの黎明期において、日本語対応パッケージの「JE (Japanese Extensions)」がSlackware Linux向けに開発されていたこともあり、日本では事実上の標準ディストリビューションだったこともある。

最新環境をしっかりとサポート

最新版のSlackware Linux 8.0は、カーネル2.2.19 (2.4.5も同梱)、glibc 2.2.3、XFree86 3.3.6 / 4.1.0といった最新のシステムコンポーネントからなっている。

デスクトップ環境も最新装備で、GNOME 1.4、KDE 2.1.1といった統合デスクトップ環境にWebブラウザのGaleon、新世代ファイルマネージャのNautilusを組み合わせている (画面1)。ウィンドウマネージャは、AfterStep、BlackBox、Enlightenment、Window Makerなどを利用可能だ。



Slackware Linux 8.0

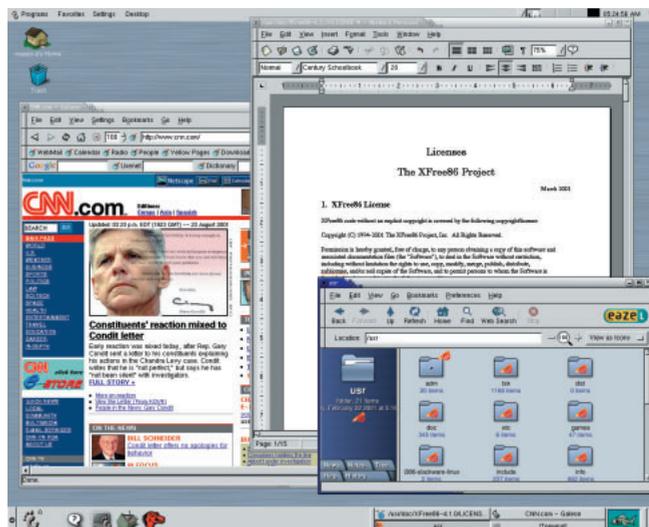
初心者には難しい？

Slackware Linuxでは、RPMなどのように複雑なパッケージ管理システムではなく、単純なパッケージインストールの仕組みを用いている。ソフトウェアのパッケージは、実行ファイルと、インストール後の設定を行うシェルスクリプトをtarとgzipでアーカイブした形式で配布されており、これをpkgtool、installpkgというコマンドでインストール/アンインストールする。

実行バイナリのパッケージが用意されていないソフトウェアは、ソースファイルからコンパイルしてインストールすることが前提となるので、その作業に慣れていないユーザーにとっては使いこなすのが難しいかもしれない。また、システム設定も、ほとんどの場合は設定ファイルを直接書き換えることになる。

ソフトウェアがバージョンアップしたときに、バイナリパッケージを待つ

画面1 Slackware Linux 8.0のGNOMEデスクトップ。GNOME 1.4による最新のデスクトップ環境を提供する。WebブラウザにはGaleonを採用し、ファイルマネージャのNautilusも利用可能だ。GNOMEのほか、KDE 2.1.1やAfterStep、Enlightenmentなどを使うこともできる。



ことなくソースファイルを入手してコンパイル、インストールするのが好きだという上級ユーザーには魅力的なディストリビューションだろう。

日本語化への道

Slackware系の日本語対応ディストリビューションには、本誌54ページで紹介したPlamo Linuxがある。手間をかけることなく日本語環境を手にしたのならこちらを利用することをお勧めするが、Slackware Linuxで日本語対応ソフトウェアを利用したいという方は、ぶらっとホームオープンラボラトリーのWebページ (<http://openlab.plathome.co.jp/linux/>) が参考になるだろう。ここには、各種ソフトウェアの日本語対応版が順次公開されている。もちろんぶらっとホームからのサポートは提供されないが、同社のオープンソースコミュニティに対する姿勢は高く評価したい。



LinuxPPC 2000 Q4 日本語版

アマレット株式会社
<http://www.amulet.co.jp/>

4800円

■ カーネル2.4

■ XFree86 4.x

✓ GNOME 1.4

✓ KDE 2.1

✓ 日本語

■ ReiserFS

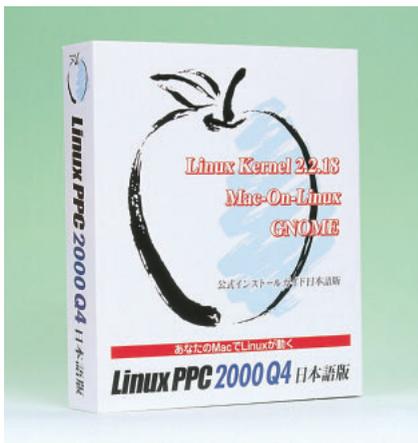
■ ext3

LinuxPPC 2000 Q4 日本語版は、LinuxPPC, Inc.が開発するMac用のディストリビューションLinuxPPC 2000を、アマレットが独自に日本語拡張した製品だ。LinuxPPC 2000 Q4 日本語版は、表1のようにPowerPCとPCIバスを搭載するMacで動作する。

LinuxPPC 2000 Q4 日本語版は、基本コンポーネントにカーネル2.2.18、glibc2.1.3、XFree86 3.3.6が、デスクトップ環境にGNOME 1.4.0とKDE 2.1.1が採用されている。

製品には、インストールCDとソースCDがそれぞれ2枚と、追加パッケージを収録するCDの合計5枚のCD-ROM、そしてインストールを解説する小冊子が収録される。

また、ユーザーはLinuxPPC日本語化プロジェクト (<http://www.linuxppc.ne.jp/>) のWebページで、インストールからXの起動までについて、3件までのサポートを受けられる。



LinuxPPC 2000 Q4 日本語版

使いやすいインストーラ

LinuxPPC 2000 Q4 日本語版のCDをセットして、「C」を押しながらマシンの電源を入れると日本語対応のインストーラが起動する。インストール作業は、PerDisk (画面1) などを使ってのパーティション設定と、インストールするパッケージグループの選択くらいで、手順はとても少ない。ハードウェアの設定も自動で行われ、編集部のiMacにも簡単にインストールできた。

以前のバージョンでは、英語版のLinuxPPCをインストールしたあとに、日本語化パッケージを追加して日本語環境を構築していたが、LinuxPPC 2000 Q4 日本語版からは、あらかじめインストールCDに含まれる日本語パッケージがインストールされるので、日本語環境構築の手間が大幅に減っている。

ただ、初めてLinuxをインストールするMacユーザーのことを考えると、付

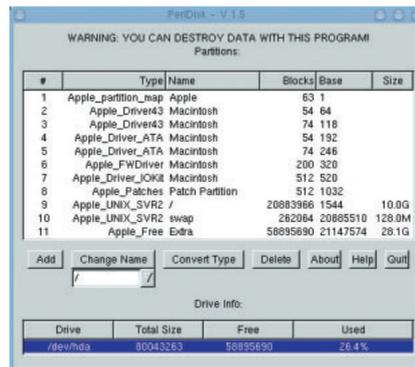
属のインストールガイドは決してわかりやすいものではないので、初心者がいちからLinuxを学べるレベルのマニュアルを製品にバンドルしてほしいところだ。

LinuxPPC 2000ならではの

LinuxPPC 2000 Q4 日本語版には、Linux上でMac OSをエミュレートするMac-On-Linuxが付属する。このエミュレータを有効に使えるかどうかはユーザー次第だが、Linux上でMac OSが動作すること自体は楽しめる。

また、無線LANの対応が強化されていて、特に、AirMacとWaveLANは通信内容を暗号化するWEPに対応している。今後、無線LANユーザーが増えていくのは確実なので、新しい技術へのいち早い対応として評価できる。

なお、LinuxPPC 2000 Q4 日本語版のインストールCDには、開発ツールなどの欠落という不具合がある。発売元のアマレットは、インストールCDを無償交換しているため、LinuxPPC 2000 Q4 日本語版ユーザーは、11月30日までに交換手続きをしておこう。



画面1 パーティション作成ツールPerDisk
インストール前に、このツールを使ってパーティションを作成する。

動作機種	
PCIバスを搭載するPowerMac	
iMac (Rev A ~ Rev D iMac DV)	
PowerMac G3 / G4 (PCI G4、AGP G4、G4 Cube)	
PowerBook 2400以降のPowerBook (G3各種)	
iBook (Firewireを含む)	

表1 LinuxPPC 2000 Q4 日本語版が動作する機種
PowerMac G4の2001年モデルには未対応で、PowerBook G4には限定的に対応している。

サーバ系ディストリビューション徹底比較

Linuxブームに火がつく以前は、Linuxディストリビューションに、ワークステーション用やサーバ用などという製品区別はなかったが、Linuxがサーバとしての地位を固めるにつれて、インストール直後からサーバとして使えるようなディストリビューションが、リリースされ始めた。

比較のポイントは？

サーバ系ディストリビューションを購入する動機は、やはり手間をかけずに目的のサーバを構築することだ。

それにはまず、付属のマニュアルやサポートを利用して、簡単にLinuxをインストールできるかどうかのポイントになる。

次に、サーバアプリケーションの設定という手間のかかる作業を支援するために、使いやすい設定ツールと、サーバ設定に関するユーザーサポートの有無が重要な比較項目になる。

このほかに、サーバとして使うために、大容量メモリやジャーナリングファイルシステムのサポートといった、

基本部分のチューニングもポイントになるだろう。

これらの項目の中で、Linuxの導入については、どのディストリビューションのインストーラも高いレベルに到達しているし、製品にはインストールガイドとサポートが付属するので、今回の比較項目からは、外すことにした。それでは、

- ・サーバ設定
- ・基本性能のチューニング

という2つの項目を中心に、国内で入手しやすいサーバ系ディストリビューションを比較していこう。

設定ツールとユーザーサポート

現在、Linuxディストリビューションに付属する設定ツールは、

- ・テキストベースのもの
- ・X上で動作するもの
- ・Webブラウザで操作するもの

と大きく3つに分類される。

歴史的には、まずテキストベースのものが開発され、X上で動作するもの、Webブラウザで操作するものという順番で、ディストリビューションに収録されてきた。

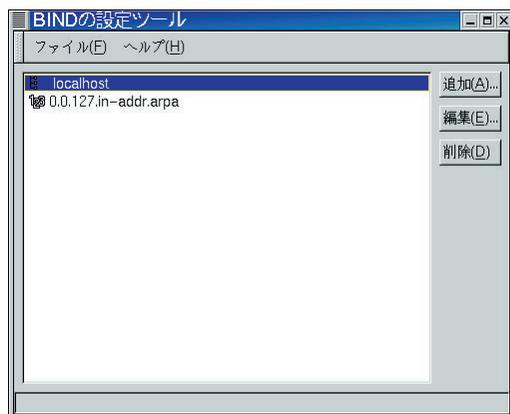
Webブラウザで操作するタイプは、WindowsでもLinuxでも環境を選ばずに設定できるので、このタイプを採用するディストリビューションが増えてきている。

また、サーバ設定を支援するもうひとつの要素であるユーザーサポートは、ディストリビューションによって、付属するかどうか異なる。

Red Hat Linux 7.1 Professional Server

Red Hat Linux 7.1 Professional Serverには、apacheconfやbindconf(画面1)といったX上で動作するGUI設定ツールと、netconfigやntsysvといったテキストベースの設定ツールが収録されている。

これらツールはシンプルに作られているので設定しやすいが、Apacheと



画面1 bindconf
DNSサーバとしてポピュラーなBINDを設定するGUIツール。Red Hat Linux 7.1 Professional Serverに付属する。



画面2 turboportcfg
turboportcfgは、テキストベースの設定ツール群TurboToolsのひとつとして収録され、inetdから起動されるデーモンのアクセス制御を設定するツールだ。



BINDはGUI設定ツールで、SambaはSWATを通してWebブラウザで、Webキャッシュサーバはテキストの設定ファイルを直接編集して、と設定方法に統一感がない。

Red Hat Linux 7.1 Professional Serverユーザーに提供されるサーバ設定のサポートは、ApacheやBINDに関する基本的な項目のみで、ほかのサーバ設定についての詳細なサポートを受ける場合は、別途サポートパックを購入する必要がある。

Turbolinux Server 6.5

Turbolinuxのラインナップには、もともとTurboTools (画面2) というテキストベースの設定ツールが付属する。

また、Turbolinux Server 6.5には、TurboToolsに加えて、商用のHDE Linux Controller 2.0 Express Edition (画面3) が収録されている。

HDE Linux Controllerは、Webブラウザから操作する設定ツールで、Windowsからでも、Linuxからでも同じようにシステム管理やサーバ設定を行えるのが特徴だ。各設定項目は初期画面に集約され、設定画面も見やすく操作性に優れている。

製品には、TurboToolsを使ったサーバの設定方法を解説するユーザーズガイドと、ネットワークサーバの設定

に関するサポートが付属する。

LASER5 Linux 6.5 Secure Server Edition

LASER5 Linux 6.5 Secure Server Editionには、セキュリティ対策のためにXが収録されていない。そのため、設定ツールも、Red Hat Linuxのnetconfigなどテキストベースのもののみが収録されている。

製品に付属するサーバリファレンスでは、テキストの設定ファイルの編集方法が詳細に解説されているので、レーザーファイブは設定ツールの使用をあまり想定していないのかもしれない。

とはいうものの、サーバリファレンスはとても丁寧に解説されているし、サーバ設定に関するサポートを180日間受けられるので、設定についてそれほど困ることはないだろう。

Kondara MNU/Linux Server 1.1

Kondara MNU/Linux Server 1.1には、X上で動作するGUI設定ツールmphが付属する。mphは、Kondara Projectが開発する高機能なパッケージ管理ツールだ。Kondara MNU/Linux Server 1.1では、デジタルファクトリが独自に拡張したGUI版のmphをシステム管理とサーバ設定の中心に据えている。

Kondara MNU/Linux Server 1.1に付属するGUI版のmphは、Webサーバ、ネームサーバ、メールサーバの設定をサポートしており、mphの使い方を簡潔に説明したマニュアルと、90日間提供されるサポートが付属する。

Miracle Linux Standard Edition V1.1

Miracle Linuxに付属するテキストベースの設定ツールMiracleToolsは、Turbolinuxに付属するTurboToolsをベースにしているので、設定範囲や操作感はほとんど同じだ。

さて、サーバ設定を解説したマニュアルは付属するものの、サポートはインストールに関するもののみで、サーバ設定に関するサポートを受けるには、別途サポート権を購入する必要がある。

HOLON Linux 2.0 Server

HOLON Linux 2.0 Serverでは、Webブラウザで操作するオープンソースの設定ツールWebmin (画面4) が前面に押し出されている。

収録されるWebminは、デフォルトでSSLに対応しているし、設定項目が多く、日本語表示できるよう拡張されている。また、Webminの使い方が詳しく書かれたマニュアルと、90日間の件数無制限サポートが付属する。



画面3 HDE Linux Controller 2.0 Express Edition

このツールは、Turbolinux Server 6.5とMiracle Linux Standard Edition V1.1にバンドルされる。Webブラウザで操作するので、LinuxでもWindowsでも同じ操作感で設定できる。

画面4 Webmin

HOLON Linux 2.0 Serverは、Webminでの設定を前提に開発されている。Webブラウザから設定するので、ユーザーはコマンドを意識せずに設定可能だ。



プロサーバ Linux ver3.0

プロサーバ Linux ver3.0には、X上で動作する独自の統合設定ツール GIDEON WHITEBOXが付属する。

簡潔にサーバの設定方法を解説したマニュアルが付属するので、コマンド操作に慣れないユーザーでもサーバを設定できるだろう。なお、サーバ設定に関するサポートは提供されない。

サーバ用としての基本性能

大規模サイトでLinuxをサーバとして使う場合は、

- ・ Rawデバイス
- ・ 大容量メモリ
- ・ LVM

・ジャーナリングファイルシステム

などの機能が必要となる。

Red Hat Linux 7.1 Professional Server

Red Hat Linux 7.1 Professional Serverは、今回取上げたディストリビューションの中で、唯一カーネル2.4を採用している。

先にあげたサーバに必要な4つの機能は、カーネル2.4では標準的にサポートされているものなので、Red Hat Linux 7.1 Professional Serverは、特別なカーネルのチューニングをしなくても、そのまま大規模サイトで利用できる。ジャーナリングファイルシステムとしては、ReiserFSがサポートされる。

Turbolinux Server 6.5

Turbolinux Server 6.5が採用するカーネル2.2には、先に挙げた4つの機能をすべて実現するように、カーネル2.4からバックポートされたパッチが適用されている。

ジャーナリングファイルシステムとしては、ReiserFSとext3がサポートされており、インストール中のパーティション作成の場で選択可能だ。

LASER5 Linux 6.5 Secure Server Edition

LASER5 Linux 6.5 Secure Server Editionは、ほかのディストリビューションと比べてリリースされる時期が早かったせいか、ほとんどカーネルはチューニングされていない。

サーバ系ディストリビューション機能一覧表

製品名	Red Hat Linux 7.1 Professional Server	Turbolinux Server 6.5	LASER5 Linux 6.5 Secure Server Edition
販売元	レッドハット株式会社	ターボリナックス ジャパン株式会社	レーザーファイブ株式会社
URL	http://www.redhat.com/jp/	http://www.turbolinux.co.jp/	http://www.laser5.co.jp/
価格	3万4800円	3万9800円	4万9800円
カーネル	2.4.2	2.2.18	2.2.14
大容量メモリ			-
Raw I/O			
ジャーナリングファイルシステム	ReiserFS インストール時は選択不可	ext3, ReiserFS	-
RAID			
LVM			-
サポート内容	インストール、基本的なシステム設定	インストール、インターネットサーバの設定	インストール、サーバの設定と運用
サポート方法	Web、電話	メール、Web、電話、FAX	メール、Web、電話
サポート期間	90日間	90日間	180日間
サポート件数	無制限	5件	3件
同シリーズのほかの製品	-	Turbolinux Advanced Server 6 Turbolinux DataServer 6 for Oracle8i Turbolinux Cluster Server 6 Turbolinux CLUSTERPRO Server 6	-
おもな管理/運用ツール	bindconf、ntsysv、netconfigなど	TurboTools、HDE Linux Controller	ntsysv、netconfigなど
付属商用ソフト	PartitionMagic SE Wnn6 Ver 3.02 DynaFont 5書体	HDE Linux Controller 2.0 Express Edition Netvault 6 Turbolinux OEM版 RYOBI TrueType フォント 5書体	-
おもな特徴	カーネル2.4の採用により、最大64Gバイトメモリの使用など大規模なサーバ運用も可能	商用ソフトに、WebブラウザからLinuxを管理できるHDE Controllerと、バックアップツールNetvaultをバンドルして、サーバとしての機能を強化	セキュリティツールNessusや、セキュリティが要領よく解説されたマニュアルなどがバンドルされ、セキュリティ対策はピカイチ



Kondara MNU/Linux Server 1.1
 Kondara MNU/Linux Server 1.1は、Turbolinuxと同様に、カーネル2.4の新機能を実現させるためのパッチを、デフォルトのカーネル2.2に当てている。先にあげた機能の中では、すべての機能がサポートされている。また、ReiserFSもインストール時に選択可能だ。

Miracle Linux Standard Edition V1.1

Miracle Linux Standard Edition V1.1はTurbolinuxをベースにしているので、Turbolinux Server 6.5と同じく、2.2カーネルで、先にあげた4つの機能に対応している。

サポートするジャーナリングファイルシステムも、Turbolinux Server 6.5

と同じく、ReiserFSとext3だ。このほかにも、Oracleデータベースのパフォーマンスを上げるために、いくつかのカーネルパラメータが変更されている。

HOLON Linux 2.0 Server
 プロサーバ Linux ver3.0

この2つのディストリビューションは、大規模サイトでの利用が想定されていないためか、カーネルに特別な機能拡張はなされていない。

お勧めディストリビューションは？

冒頭で述べたように、サーバを構築するまでの支援、基本性能のチューニングという2つの項目を中心に、サーバ系ディストリビューションを比較

してきた。その結果、TurbolinuxServer 6.5が、もっとも高いレベルで、2つの評価項目を満足しているようだ。また、マニュアルとユーザーサポートの充実度で、HOLON Linux 2.0 Serverは、サーバ管理の初心者に適したディストリビューションだといえそうだ。

なお、今回取上げたディストリビューションは、リリース時期に大きな開きがあるので、LASER5 Linux 6.5 Secure Server Editionなどは、ほかのディストリビューションと比べてどうしても見劣りしてしまう。

レーザーファイブは、9月に新しいサーバ用ディストリビューションをリリース予定なので、製品を購入する際は、新製品の動向も気にしながら、目的にあったものを選択していただきたい。

Kondara MNU/Linux Server 1.1	Miracle Linux Standard Edition V1.1	HOLON Linux 2.0 Server	プロサーバ Linux ver3.0
デジタルファクトリ株式会社	ミラクル・リナックス株式会社	株式会社ホロン	富士マグネディスク株式会社
http://www.digitalfactory.co.jp/	http://www.miraclelinux.com/	http://www.holonsoft.co.jp/	http://www.fujifilm.co.jp/fmd/linux/
2万4800円	5万円	3万4800円	1万9800円
2.2.15	2.2.18	2.2.18	2.2.14
		-	-
		-	-
ReiserFS	ext3、ReiserFS インストール時は選択不可	-	-
		-	-
インストール、サーバ設定	インストール	インストール、サーバアプリケーションの設定、基本的なネットワーク設定	インストール、管理ツールの起動
メール、電話、FAX	メール、FAX	電話	メール
90日間	30日間	90日間	無制限
5件	3件	無制限	2件
Kondara MNU/Linux 2000 Cluster for Web Kondara MNU/Linux HPC 2000	Miracle Linux for Samba Miracle Linux for PostgreSQL Miracle Linux with Oracle8i Miracle Linux with Oracle8i/Kylix Miracle Linux with Oracle8i/NetVaultほか	-	-
mph	MiracleTools、HDE Linux Controller	Webmin	GIDEON WHITEBOX プロサーバ
DynaFont 5書体	ATOK 12 SE R.2 for Linux HDE Linux Controller 2.0 Express Edition RYOBI TrueType フォント5書体	-	-
GUIのmphを使って、サーバの設定からカーネルの再構築まで、ユーザーの環境に合わせてチューニング可能	Oracleのプラットフォームやファイルサーバなど、製品が用途別に開発されており、インストール直後から目的のサーバとして運用可能	Webブラウザから操作する管理ツール Webminを使って、サーバ設定やバックアップを簡単にできる	インストールからサーバ設定までLinuxに不慣れなユーザーも安心して使える

製品名	Red Hat Linux 7.1 Deluxe Workstation 日本語版	Turbolinux 7 Workstation	LASER5 Linux 7.1 Devel
販売元	レッドハット株式会社	ターボリナックスジャパン株式会社	レーザーファイブ株式会社
URL	http://www.redhat.com/jp/	http://www.turbolinux.co.jp/	http://www.laser5.co.jp/
提供メディア	CD8枚	CD6枚	CD4枚、FD1枚
マニュアル	1冊	4冊、CD (PDFにて収録)	1冊
価格	1万4800円	1万5800円	1万6800円
動作条件			
CPU	x86互換 (Pentium以上推奨)	x86互換 (Pentium以上推奨)	Pentium 300MHz以上推奨
メモリ	32Mバイト以上 (64Mバイト以上推奨)	64Mバイト以上 (128Mバイト以上推奨)	64Mバイト以上 (128Mバイト推奨)
ハードディスク	650Mバイト以上 (1.2Gバイト以上推奨)	2Gバイト以上推奨	1Gバイト以上推奨
インストール			
インストーラ	日本語 / グラフィカル	日本語 / グラフィカル	日本語 / グラフィカル
ブートメディア	CD-ROM、FD、DOS	CD-ROM、FD、DOS	CD-ROM、FD、DOS
インストール元メディア	CD-ROM、HDD、PCMCIA、FTP、HTTP、NFS	CD-ROM、HDD、PCMCIA、FTP、NFS	CD-ROM、HDD、PCMCIA、FTP、HTTP、NFS
インストーラ8G超対応	×		
新しいファイルシステム	-	ext3、ReiserFS、JFS	-
システム構成			
パッケージ管理システム	RPM	RPM	RPM
デスクトップ環境	GNOME、KDE、fvwm2、Enlightenment、twm、WindowMaker	GNOME、KDE、Sawfish、AfterStep、twm	GNOME、KDE、fvwm2、Sawfish、Enlightenment、twm、WindowMarker
日本語入力システム	Canna、FreeWnn、ssk、Wnn6	Canna、Wnn6、skk、ATOK X	Canna、skk、Wnn6、ATOK X
バージョン			
カーネル	2.4.2	2.4.5	2.4.3
標準Cライブラリ	2.2.2	2.2.3	2.2.2
XFree86	4.0.3	4.1.0	4.0.3
GNOME	1.2.4	1.4.0	1.2.4
KDE	2.1.1	2.1.1	2.1.1
設定ツール			
ユーザー管理	linuxconf、kuser	kuser、turbousercfg	linuxconf、userconf、kuser
ネットワーク	linuxconf、netcfg、netconf、netconfig	netcfg、turbonetcfg	linuxconf、netcfg、netconf、netconfig
Xサーバ設定	Xconfigurator、xf86conf	turboboxcfg、xf86config	Xconfigurator、xf86config、xf86conf
ウィンドウマネージャ変更	wmconfig	turbowmcfg	switchdesk
サウンド	sndconfig	alsaconf	sndconfig
起動デーモン	linuxconf、ksysv、ntsysv、tksysv	ntsysv、ksysv、turboservice	linuxconf、netsysv、tksysv、ksysv
パッケージ管理	gnorpm、kpackage	kpackage、turbopkg	gnorpm、kpackage
アップデートツール	Red Hat Update Agent	zabom	-
主なバンドルソフト			
日本語入力システム	Wnn6	Wnn6、ATOK X	Wnn6、ATOK X
日本語TrueTypeフォント	DynaFont 5書体	リコーTrueTypeフォント 5書体	DynaFont 5書体
ブートマネージャ	BootMagic 6.0	-	Smart Boot Manger
その他	Hancom Word、ifilter、SimCity3000デモ版、SidMeier's AlphaCentauriデモ版	Sun JDK 1.3.0、Sun Forte for Java、Community Edition 日本語版、Borland Kylix 1.0 Open Edition	K筆、VNC、eWnn、Tiny COBOL、JDK 1.3、JBuilder5 Personal
サポート			
サポート内容	基本的なLILO、パーティション、Windows 9Xとのデュアルブートの設定	インストール、サウンドカード設定、USB機器設定	インストールからX Window System起動まで
サポート方法	Web、電話	メール、Web	電話、FAX、Web質問フォーム
サポート期間	60日間	60日間	90日間
サポート件数	無制限	無制限	無制限
備考			



Kondara MNU/Linux 2.0	Vine Linux 2.1CR [2.1.5]	HOLON Linux 3.0 PC/AT互換機「豪華」	Debian GNU/Linux 2.2r3
デジタルファクトリ株式会社	レッドハット株式会社	株式会社ホロン	—
http://www.digitalfactory.co.jp/	http://www.vinelinux.org/	http://www.holonlinux.com/	http://www.debian.org/
CD8枚	CD8枚、FD3枚	CD5枚	-
2冊	1冊	2冊	—
1万2800円	1万2800円	1万1800円	—
Pentium以上または、AMD K5以上推奨	Pentium100MHz以上	PentiumPro相当以上	i386以上
64Mバイト以上推奨	32Mバイト以上 (推奨64Mバイト以上)	128Mバイト以上	12Mバイト以上
3.2Gバイト以上 (最小210MB以上) 推奨	600Mバイト以上	1Gバイト以上	64Mバイト以上
日本語 / テキスト	日本語 / グラフィカル	日本語 / グラフィカル	英語 / テキスト
CD-ROM、FD、DOS	CD-ROM、FD、DOS	CD-ROM、FD、DOS	CD-ROM、FD、DOS
CD-ROM、HDD、PCMCIA、FTP、HTTP、NFS	CD-ROM、HDD、PCMCIA、FTP、HTTP、NFS	CD-ROM、HDD、PCMCIA、FTP、HTTP、NFS	CD-ROM、HDD、PCMCIA、FTP、HTTP、NFS
x	x []	—	—
ReiserFS	—	—	—
RPM	RPM	RPM	dpkg
GNOME、KDE、fvwm、fvwm2、Sawfish、Enlightenment、twm、Qvwm、AfterStep	GNOME、KDE (VinePlusに収録)、fvwm2、Sawfish、twm、WindowMaker	GNOME、KDE、fvwm、fvwm2、Sawfish、Enlightenment、twm、WindowMarker、AfterStep	GNOME、fvwm2、fvwm95、WindowMarker、AfterStep、Sawfish、Qvwm、twm、Icewm、Enlightenment、Blackboxなど
Canna、skk、FreeWnn、Wnn6	Canna、FreeWnn、Wnn6	Canna、FreeWnn、Wnn6、ATOK12 SE	Canna、FreeWnn、SKK
2.4.4	2.2.17 [2.2.18]	2.4.5	2.2.19pre17
2.2.2	2.1.3	2.2.3	2.1.3
4.0.3	3.3.6	4.0.3	3.3.6
1.4.0	1.2.1	1.4.0	1.0.55
2.2 alpha1	1.1.2	2.1.1	—
kuser	linuxconf、userconf	kuser	linuxconf
netconfig	linuxconf、netcfg、netconf、netconfig	netcfg、netconfig	linuxconf
Xconfigurator、xf86config	Xconfigurator、XF86Setup、xf86config	configurator、xf86cfg、xf86config	anXious、xf86config
sdr	setwm	switchdesk	—
sndconfig	sndconfig	sndconfig、soundconf	alsaconf
ksysv、ntsysv、tksysv	linuxconf、ntsysv、tksysv	tksysv、ksysv	update-rc.d
gnorpm、kpackage	gnorpm	gnorpm、kpackage、apt	apt、aptitude、gnome-apt
mph	— [apt]	apt	apt
Wnn6	Wnn6	Wnn6、ATOK12 SE	—
Dynafont 5書体	Dynafont 5書体	Dynafont 15書体	—
System Commander Lite	—	—	—
翻訳魂	翻訳魂、Hancorn Word	翻訳魂、OSS	—
shade Preview Kit	—	—	—
制限無し	インストールまで	インストールからX Window Systemの起動まで	—
メール、電話	メール、FAX、書面	メール、電話 (通話料無料)、FAX (通話料無料)	—
サポート登録日より次期バージョン発売日まで	30日間	90日間	—
無制限	無制限	無制限	—
	フルインストール時1.2Gバイト以上	「軽量」オプション選択時はPentium以上、メモリ48Mバイト以上、ハードディスク600Mバイト以上	

Linuxディストリビューションヒストリー

Linux OSのコアの部分であるカーネルは、ご存じの通りLinus Torvalds氏を筆頭に多くの人々によって開発されている。また、OSをサポートする基本的なコマンドやアプリケーション、そして商用ソフトなどLinuxの周辺ソフトウェアはたくさんある。

それらをユーザーが使いやすいように組み合わせ、インストール環境を整えた配布パッケージが、Linuxディストリビューションである。これまでのページで紹介してきたように、数多くのLinuxディストリビューションが存在している。

Linuxカーネルの歴史

基本コンポーネントであるLinuxカーネルがバージョンアップされれば、当然のことながらディストリビューションもそれに対応することになる。まずはLinuxカーネルの歴史についておさらいしておこう。

Linuxは、フィンランドのヘルシンキ大学の学生だったLinus氏が1人で開発を始めたOSである。最初のバージョン0.01が完成し、ニュースグループ

comp.os.minixに「AT互換機用のフリーなOSを作成中」と投稿した日が1991年8月25日であり、それから10年経ったことになる。それを記念してLinux誕生10周年のWebサイト(画面1)ができ、パーティが8月25日にカリフォルニア州サニーベールの公園で行われた。ちなみに、現在のPCの原型ともいえる初代IBM PCの発売は1981年で、こちらは20周年だそうだ。

さて、実際には、0.01はソースを一部の人に公開しただけで、一般にバイナリを公開したのは1991年の10月である。それからいろいろな改良が行われ、1994年3月にVer. 1.0がリリースされた。1996年5月にはカーネル2.0がリリースされ、現在のLinuxの基本システムとしてほぼ完成された。

カーネル2.2

カーネル2.2のリリースは1999年1月で、アーキテクチャの大幅な変更は行われず、パフォーマンスやメモリ管理の向上などが図られた。また、x86互換プロセッサ以外(特に64ビットCPU)の数多くのCPUへの対応や、SMPマルチプロセッサシステムでの性能を改善した。デバイスドライバのカーネルモジュール化の推進やFAT32、NTFSなどのファイルシステムのサポート、ソフトウェアRAID機能を備えた。

カーネル2.4

現在の安定版の最新カーネル2.4は2001年1月にリリースされた。インターネットの普及に合わせてサーバとしての能力を高める必要があり、マルチ

プロセッサシステムでの安定性を向上させた。ネットワーク層の全面的な書き換えが行われ、ファイアウォールやインターネットプロトコルの処理がカーネル内に組み込まれた。

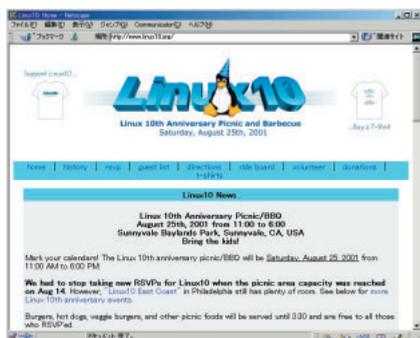
また、USBやPCカードへの対応、ファイルシステムを改善したことでファイルサイズの最大2Gバイトという制限がなくなり、LVM(論理ボリュームマネージャ)機能の追加などが行われている。

安定版と開発版の2種類

オープンソースで開発されているLinuxカーネルは、いろいろな機能を盛り込むための開発版カーネルと、開発がある程度一段落した時点で新機能の追加を中止して、動作確認とバグフィックスを行った安定版カーネルの2種類を提供している。開発版カーネルと安定版カーネルは、バージョン番号の2番目の数字で区別していて、奇数が開発版、偶数が安定版となっている。

ディストリビューションが採用しているのは、もちろん安定版のほうであるが、ディストリビューターによっては開発版カーネルの機能をパッチの形でバックポートすることで、次期カーネルの新機能を取り込んでいることもある。

特にカーネル2.2以降は、ディストリビューション間の開発競争が激しくなったことと、カーネル2.4のリリースが遅れたこともあって、カーネル2.2.xを採用していてもカーネル2.4相当の一部の機能を持たせたディストリビューションも多い。



画面1 Linux 10
http://www.linux10.org/



ディストリビューションの歴史

日本国内で使われている主なLinuxディストリビューションのリリース時期を、図1にまとめた。この図では、メインとなるワークステーション版のバージョンのみを記載した。

1998年から1999年にかけて、国内でLinuxブームが起り、多くのディストリビューションがリリースされ始めた。この時期の変化としては、ハードウェアの自動検出機能を組み込んだり、日本語インストーラを備えたりして、インストールが容易になったことが挙げられる。

ユーザーの声を取り入れながら進歩してきたのと同時に、既存のディストリビューションでは満足できないと、新たに立ち上がったディストリビューターもいる。それぞれのディストリビ

ューションの開発の歴史を振り返ってみよう。

Slackware

1993年6月より配布が始まったSlackwareは、その後日本語環境をまとめたJE (Japanese Extensions) を含んだCD-ROMの形で入手が容易だったため、Linuxがブームとなる前の1997年ごろまでは、国内でも多くのユーザーに利用されていた。

Slackwareは、商用ではなくボランティアベースのディストリビューションとして、最新機能を取り込みつつ、バージョンアップが行われているが、そのままでは日本語に対応していないことなどから日本でのユーザーはかなり少なくなっていると思われる。

1997年に、Slackwareをベースに日本語化したPlamo Linuxが登場し、PC-98x1シリーズへの対応など、独自

の進化を遂げている。

Debian GNU/Linux

Free Software FoundationのGNU Projectの協力で、1993年にDebian Projectが開始した。Debian GNU/Linuxと名付けられ、GNUの基本理念に沿ったLinuxを提供するため、商用ソフトウェアを含まない、ライセンス的にクリーンなディストリビューションとして開発されている。

なお、日本ではDebian JP ProjectからDebian-JPという日本語化のためのパッケージがリリースされていたが、Debian 2.2 (potato) 以降は本家のDebian Projectに統合された。

1999年に、Debian GNU/LinuxをベースにしたCorel Linux、Storm Linuxなどのディストリビューションがリリースされたが、現在では開発が中止されている。Debianベースのディ

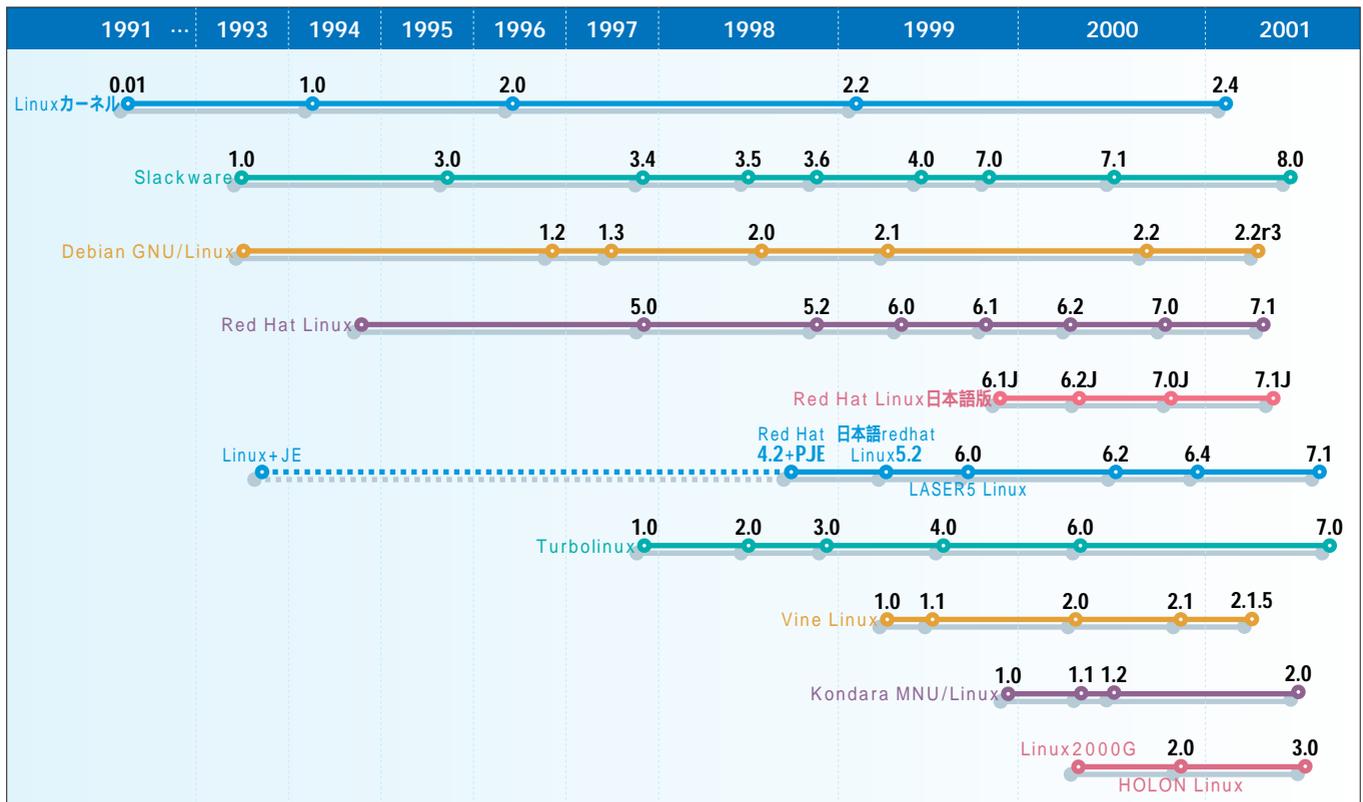


図1 主なLinuxディストリビューションのリリース時期

ストリビューションとしては、2000年よりオモイカネからOmoikane GNU/Linuxが提供されている。

Red Hat Linux

Linux商用ディストリビューターとして1994年に開始したRed Hatは、Linuxの世界でもっとも有名かつ成功している会社だ。

Red Hatは、先進的な技術をいち早く取り込んでリリースしている。RPMパッケージ管理システムや、CUI、GUIの使いやすいインストーラを開発し、Linuxユーザーのインストール/アンインストール作業を楽にしたことで、今日のLinux普及の原動力と言えるだろう。そして、それをベースにした多くのディストリビューションの基本システムとなっている。

1999年の9月にRed Hatの日本法人レッドハットが設立され、日本向けのパッケージを販売している。世界中で利用されているRed Hat Linuxは、国際化対応を進めており、最新のRed Hat Linux 7.1からは、インストールCDを共通化したためバージョンに「日本語版」や「J」といった文字を使わなくなった。

商用フォントや商用日本語入力システムなど日本独自のアプリケーションCD-ROMが、別に添付される形になるため、基本システムをインストールしたあとの作業が発生する。多くのRed Hatベースのディストリビューションは、そういった使い勝手の不満を改善するために生まれたといってもよいだろう。

LASER5 Linux

五橋研究所はLASER5ブランドで、1993年10月よりSlackwareにJE日本語パッケージを追加したCD-ROMを発売

して以来、国内での日本語対応のLinuxを提供してきた。

その後、ベースディストリビューションをRed Hat Linuxに変更し、1999年3月には「日本語redhat Linux 5.2」を発売した。1999年8月にレーザーファイブを設立し、「ベターRed Hat」のディストリビューションとして、LASER5 Linuxをリリースしている。

Turbolinux

ターボリナックス ジャパンの前身であるパシフィック・ハイテック (PHT) は、WindowsやMac用のフリーソフトウェアやシェアウェア、SlackwareのCD-ROMなどを取り扱っていた会社である。

そのPHTが、1997年12月に初めて日本語インストーラを備えた「TurboLinux日本語版1.0」を発売した。Red Hat Linux 4.2をベースに独自ツールを加え、商用TrueTypeフォント、Wnn6、各種日本語パッケージなどをバンドルした。当時は、英語版のRed Hatをインストールしたあとで日本語パッケージを追加するのが普通だったので、インストール直後から日本語環境が整うことの意義は大きかった。

その後、1.4、2.0、3.0と約1年間の間に4回のバージョンアップが行われ、X Window Systemの日本語環境などが改善され、より使いやすく改良されていた。

また、Linuxで動作する初めてのオフィススイート製品「Applixware日本語版 for TurboLinux」や、それをバンドルした「TurboLinux Pro日本語版3.0」を発売し、当時のLinuxブームの火付け役と言ってもよいだろう。

サーバ用として1999年3月には「TurboLinux Server 1.0」を発売し、クラスタリング機能を搭載したディス

トリビューションの提供を開始するなど、新しい試みを先駆けて行ってきた。

その後本社を米国Turbolinuxに移し、海外拠点を複数設立して世界に進出した。さらに、1999年7月に社名変更して現ターボリナックス ジャパンとなっている。

Vine Linux

Red Hat LinuxやSlackwareを日本語化するアドオンパッケージPJE (Project Japanese Extensions) を開発していたメンバーが、1998年に発足させたのがProject Vineだ。

Vine Linux 1.0はRed Hat Linux 5.xをベースに開発され、1999年3月にリリースされた。現在のVine Linux 2.1.xは、Red Hat Linux 6.2をベースにしており、PPC版、SPARC版、Alpha版も用意されている。

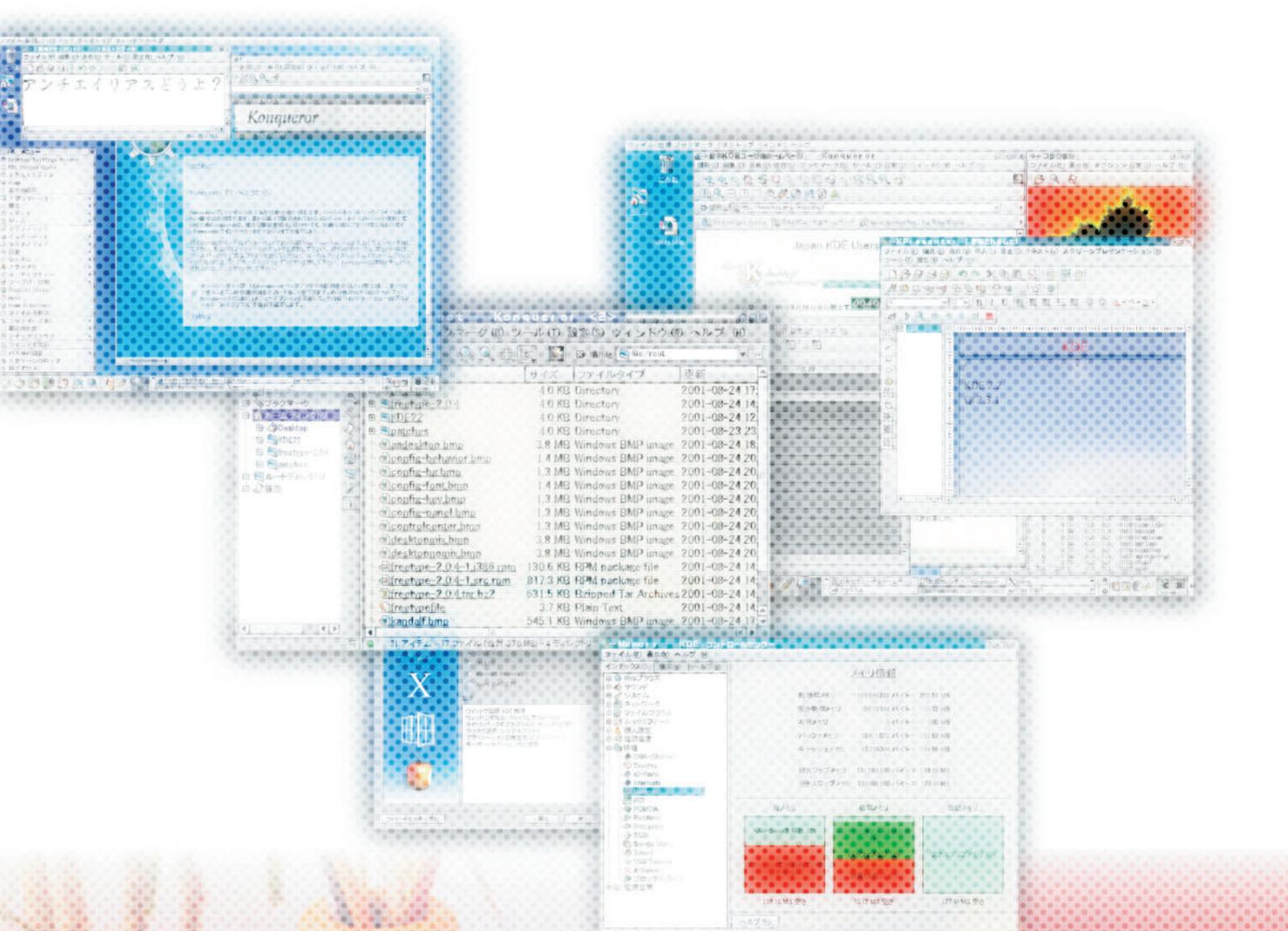
製品版は当初、技術評論社より発売されていたが、Vine Linux 2.1CRからはレッドハットが販売している。

Kondara MNU/Linux

Kondara Projectが開発しているディストリビューションで、デジタルファクトリから、1999年12月にKondara MNU/Linux 1.0が発売された。製品パッケージにはAlpha版も同梱されているという特徴がある。

HOLON Linux

ホロンが開発するHOLON Linuxの最初のバージョンは、2000年4月に、x86版とPPC版を同梱した「Linux2000G」という名前で始まった。次のバージョンからは、HOLON Linuxと名称が変わり、x86版とPPC版が分けられ、なおかつバンドルする商用ソフトの違いによって「豪華」と「並」の別パッケージとして発売している。



Linuxデスクトップ環境は、ここまできた!

快樂デスクトップ環境

KDE

多くの新機能を取り込みながら、進化を続けるKDEの現在を探る

KDE ~進化は止まらない!~

KDEは、UNIX系OS上の統合されたデスクトップ環境(画面1)を提供するソフトウェア群のことであり、「The K Desktop Environment」の短縮形である。

UNIX用のデスクトップ環境としては、MotifツールキットをベースにしたCDE(Common Desktop Environment)というものがあり、商用UNIXでは広く採用されている。名前からして、KDEがCDEを意識してデザインされていることは、まちがいない。

Turbolinux 7 Workstation、Holon Linux 3.0、Linux Mandrake 8などのディストリビューションが、KDEをデフォルトのデスクトップ環境として採用している。同様の環境を目指しているものとして、GNOME(GNU Network Object Model Environment)がある。最大手のLinuxディストリビューションであるRed Hatが、GNOMEをデフォルトにしているため、KDEはどちらかというマイナーなイメージがあるが、

コミュニティの活発さでは、KDEもGNOMEも同等であり、実際に使用している人数もほとんど変わらないように思える。

Qt / KDEの歴史

KDEは、ノルウェーのTrolltech社(画面2)が開発したツールキット、Qtを採用している。QtはX Window SystemとWindowsの両方で使える、クロスプラットフォームなツールキットとして知られていた。同じソースファイルをコンパイルし直すだけで、X用、Windows用のバイナリが作成できる。

Qtには商用版と無償版の2種類が提供されている。無償版は、X上でフリーソフトウェアを作成する場合に限って使用できる。

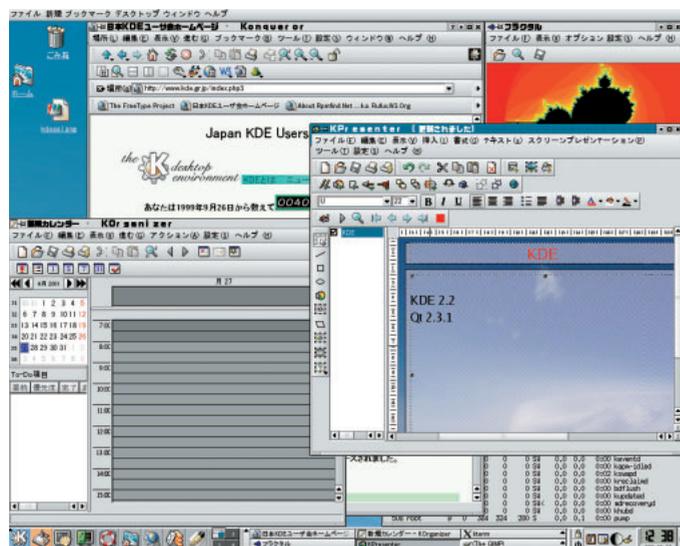
初期の無償版Qtのライセンスは、Qt自体の変更が認められなかったため、日本語に対応させることさえできなかった。当時の状況については、日本

KDEユーザ会の高木淳司氏による記事(Linux magazine No.2に掲載)が参考になるだろう。以下のページにPDFが置いてあるので、興味のある方は読んでみてほしい。

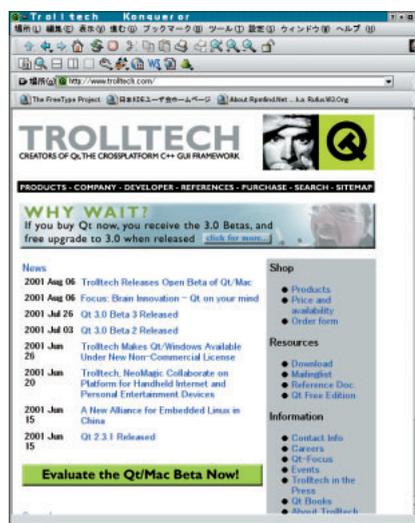
http://linux.ascii24.com/linux/allascii/linuxmag/no_02.html

Qt 2.0以降は、オープンソースライセンスの一種であるQPL(Q Public License)が採用された。しかしQPLは、GNU GPL(General Public License)とは互換性がなく、GPLの運用に厳格なDebian GNU/Linuxは、Qt/KDEをディストリビューションに含むことはなかった。

Trolltech社は、2000年9月にQtを2.2.0にバージョンアップし、同時に無償版のQtをQPL/GPLのデュアルライセンスで提供することを発表した。これ以降、KDEの開発に参画するプログラマーが増加し、KDEは規模を拡大し



画面1 KDEのデスクトップ



画面2 Trolltech社のWebサイト

つつ進化の速度を速めていく。

現在の最新バージョンは、Qtが2.3.1、KDEが2.2である。

最新版KDE 2.2の特徴

本稿執筆中に、KDEの最新版2.2がリリースされた。バージョンの数字的には、マイナーバージョンアップのように思えるが、実に多くの改良/新機能が加えられている。

オブジェクトファイルのプレリンク

現在のアプリケーションは、多くの共有ライブラリを読み込むため、起動に時間がかかる。起動時間を短縮するために、オブジェクトファイルのプレリンク機能を実験的に実装した。その結果、Linuxシステムでは起動時間が50%改善されている。

HTMLレンダリングとJavaScript

標準のKHTMLレンダリングエンジンの改良により、Webページをより正しく表示できるようになった。また、JavaScriptの扱いも改善され、安定性も向上した。

IMAPサポート

メールクライアントKMailが新たにIMAP (SSLを含む) をサポートした。

新しい印刷システム

プラグイン方式でさまざまなプリンタに対応できる、新しい印刷システムを導入した。一般的なLPD、LPR、CUPSなどに加え、ファックスや電子メールゲートウェイのような擬似プリンタをサポートしている。

Konquerorプラグインの追加

Webブラウザとしても使えるファイ

ルマネージャである、Konquerorに有用なプラグインが追加された。AltavistaのBabel Fish翻訳、イメージギャラリー生成、HTMLの品質評価などがある。

KPersonalizerの導入

KDEの初回起動時に、Kpersonalizerというセットアップ用ウィザードが起動し、ユーザーの好みに合わせた設定が、簡単にできるようになった。

インターフェイスの改善

アプリケーションを起動する際に、マウスカーソル近くに小さなアプリケーションアイコンが点滅することで、起動途中であることがわかりやすくなった。

またGNOMEと同様に、メインメニューがティアオフできるようになった (画面3)。

GNOMEとの比較

一般的に「二大陣営の対決」という構図は好まれやすいので、GNOMEとKDEも「GNOME vs KDE」といった見地で比較されることが多い。

しかし機能面では、両者に決定的な差はないし、ツールキット (KDEはQt、GNOMEはGTK+) がインストールされていれば、KDEでGNOMEアプリケーションを動かすことも、GNOMEでKDEアプリケーションを動かすことも可能だ。

結局最後は、デザイン的にどちらがより好みかで決めることになる。筆者の主観では、ウィジェットの配置方法など細かい点でKDEのほうが洗練されているように見える。また後述するように、KDEはフォントのアンチエイリアス表示が可能になったので、デスク

トップの美しさで一歩リード、ということだ。

KDEの行く末

現在、Qtの開発版である、バージョン3.0 Beta3がリリース中である。Qtのメジャーバージョンアップに合わせて、KDEもQt 3系列への対応が行われるであろう。

KDEの進化は止まらない。最新の情報は、日本KDEユーザ会のWebサイト (画面4) で確認しよう。



画面3 ティアオフメニュー
 メインメニュー下の点線をクリックすると (画面左) 独立したウィンドウとして表示できる



画面4 日本KDEユーザ会
<http://www.kde.gr.jp/>

KDE最新版をインストールする

8月15日にKDEの最新版である2.2がリリースされた。すでにいくつかのディストリビュータから、バイナリパッケージが提供されているが、今回は自分の手を動かしてtarボールからインストールしてみよう。

以下の例では、Red Hat Linux 7.1にKDE 2.2をインストールしている。ディストリビューションによっては、細かい差異があることをお断りしておく。

必要なもの

KDE 2.2をコンパイルするためには、以下のソフトウェアが必要だ。

- Automake 1.4
- Autoconf 2.13以降
- C++ コンパイラ
(GCC 2.95以降またはegcs 1.1.x)
- bzip2 / bunzip2

Red Hat Linux 7.1の場合、インストール時に、開発用ツールを選択しておけば、これらはすべて最初から含まれている。

そのほか、KonquerorでWebブラウザする際にSSL (Secure Socket Layer) を使用するためには、opensslの0.9.6、Netscape用のプラグインを利用するためには、lesstifが必要だ。

KDEは大規模な環境であり、ソースファイルは機能別に多数のtarボールで提供されている(表1)。パッケージ間に依存関係があるため、インストールする順番は重要だ。以下に示す順番でインストールを行おう。

- (1) Qt (2.2.4以降)
- (2) kdelibs
- (3) kdatabase
- (4) そのほかのパッケージ
- (5) kdeaddons

KDE 2.2のテスト用に、Athlon 1GHzと256Mバイトのメモリを搭載したPCを用意した。このマシンで上記のパッケージをすべてコンパイルするのに、半日以上かった。

Qt

KDE 2.2には、Qt 2.2.4以降のバージョンが必要だ。付録CD-ROMには、安定版の最新バージョンである2.3.1を収録してあるので、これを用いよう。

安定版とはいえ、プログラムにバグはつきものだ。また、Qtは国際化されているが、日本語のハンドリングに関する細かい部分に問題が残っている。日本KDEユーザ会(JKUG)のWebサイトで、これらの問題に対するパッチ(表2)が提供されている(<http://www.kde.gr.jp/patch/index-ja.html>)。ていねいな解説も記されているので、作業前に参照しておくといだろう。

tarボールの展開~パッチ当て

今回は、Qtを/usr/local以下にインストールすることにした。まず、/usr/localにQtのtarボールとパッチファイル(表2の“qt”で始まる5つ)を置き、ソースファイルを展開する。

```
# tar xzf qt-x11-2.3.1.tar.gz
```

ソースファイルが展開できたら、続いて以下のようにパッチを当てる。

```
# patch -p0 <qt-2.3.1-xim-20010617
```

パッケージ	内容
kde-i18n-2.2.tar.bz2	各国語のメッセージカタログやヘルプファイル
kde-i18n-ja-2.2.tar.bz2	日本語のメッセージカタログやヘルプファイル
kdeaddons-2.2.tar.bz2	konqueror, kateなどいくつかのアプリケーションの追加プラグイン
kdeadmin-2.2.tar.bz2	管理者向けのツール
kdeartwork-2.2.tar.bz2	テーマ、サウンド、壁紙などの追加セット
kdebase-2.2.tar.bz2	KDEの中心となるパッケージ。
kdebindings-2.2.tar.bz2	C、Java、Perlなどの言語とDCOPのインターフェイス
kdegames-2.2.tar.bz2	KDE上で動作するゲーム集
kdegraphics-2.2.tar.bz2	ビューア、ドローツールなど画像関連のアプリケーション
kdelibs-2.2.tar.bz2	すべてのKDEパッケージで必要になる共有ライブラリ
kdemultimedia-2.2.tar.bz2	サウンド、動画プレイヤーなどのプログラム
kdenetwork-2.2.tar.bz2	MUA、ニュースリーダなどネットワーク関連のアプリケーション
kdepim-2.2.tar.bz2	携帯電話、Palmなどと連携するツール
kdesdk-2.2.tar.bz2	開発者向けのユーティリティ
kdetoys-2.2.tar.bz2	キッカーアプレットなどジョーク系のプログラム
kdeutils-2.2.tar.bz2	電卓、プロセスモニタなどのユーティリティ
kdevelop-2.0.tar.bz2	統合開発環境
kdoc-2.2.tar.bz2	KDEのヘルプファイルを作成するためのツール
qt-x11-2.3.1.tar.gz	2001年8月時点での最新安定版Qt

表1 KDE 2.2インストールに必要なパッケージ

.diff

そのほかの4つのパッチも同様だ。エラーが出ないことを確認しよう。

コンパイル

実際にコンパイルを行う前に、環境変数QTDIRを設定する。

```
# cd qt-2.3.1
# export QTDIR=/usr/local/qt-2.3.1
```

QTDIRの設定は、`/.bashrc`にも書いておこう。次にMakefileの作成、コンパイルを行う。

```
# ./configure -sm -gif -system-jpeg
-xft
# make
```

configureスクリプトのオプションは、それぞれ以下の意味を持つ。`-sm`はQtにセッション管理を行わせること、`-gif`はGIF形式の画像のサポート、`-system-jpeg`は既存のライブラリでJPEG形式の画像のサポートを表す。また`-xft`はfreetypeパッケージを用いてアンチエイリアス表示を実現する。

エラーが出なければ、続けてコンパイルを行う。前述のマシンで、だいたい140分ほどかかった。

最後にQtライブラリの置き場所をシステムに知らせるために、`/etc/ld.so.conf`ファイルに、以下の行を追加する。

```
/usr/local/qt-2.3.1/lib
```

より古いqtのディレクトリが指定されている場合は、その行を消しておこう。追加し終わったら、`ldconfig`コマンドを実行する。

KDE本体

最初に、環境変数KDEDIRを指定しておく。

```
# export KDEDIR=/usr/local/kde2
```

KDEの各パッケージのインストール方法は、すべて共通である。以下の作業は、tarボールとパッチを同じディレクトリに置いている。最初にインストールする必要があるのは、kdelibsである。

```
# tar xjf kdelibs-2.2.tar.bz2
# patch -p0 <kdelibs-2.2-keditcl-20010816.diff
# patch -p0 <kdelibs-2.2-khtml-autoencode-ja-20010816.diff
# cd kdelibs-2.2
# ./configure
# make
# make install
```

tarボールを展開し、パッチを当て、コンパイルをするという流れは、Qtと同じ

だ。kdelibsの次に、kdebaseを同様の手順でインストールすれば、KDEが利用できる。コンパイルに要した時間は、前述のマシンで、kdelibsが60分、kdebaseが80分くらいだ。

そのほかのパッケージも、同じ手順、つまりtarボールの展開、(必要なら)パッチ当て、コンパイルという手順でインストールする。

KDEを起動するには、コマンドラインからxinitコマンドを実行し、表示されたターミナル内でstartkdeコマンドを実行すればいい。しかし、毎回同じことをするのは面倒なので、XのディスプレイマネージャからKDEが選べるようにしておこう。`/etc/X11/gdm/Sessions`に、以下の内容を持ったKDEという名のファイルを置けばよい。

```
#!/bin/bash
exec /etc/X11/xdm/Xsession kde2
```

このファイルには実行属性が必要なので、`chmod`コマンドで設定しておく。これでグラフィカルログイン時にKDEというセッションが選択できる。

パッチ	内容
qt-2.3.1-xim-20010617.diff	KDE上でATOK Xを利用可能にする
qt-2.3.1-qclipboard-20010617.diff	アプリケーション間の日本語文字列のコピーペーストを実現する
qt-2.3.1-qstring-toDouble-i18n-20010617.diff	non-latin1文字取り扱いの修正
qt-2.3.1-qpsprinter-ja-20010620.diff	日本語の印刷を可能にする
qt-2.3.1-showFullScreen-fix-20010624.diff	フルスクリーンモードのバグを修正する
kdelibs-2.2-keditcl-20010816.diff	KEditのカット&ペーストの問題を修正する
kdelibs-2.2-khtml-autoencode-ja-20010816.diff	HTMLの文字コードを自動判別する
kdebase-2.2-konsole-ja-20010818.diff	Konsole上で日本語入力を可能にする
kdebase-2.2-kpersonalizer-charset-20010816.diff	kpersonalizerの不具合を修正する
kdenetwork-2.2-ksirc-ja-20010816.diff	ksircで日本語を扱えるようにする
kdenetwork-2.2-kmail-unknowncharset-cjk-20010816.diff	charset不明のメールに対し、自動的にcharsetを設定する
kdenetwork-2.2-knewsticker-utf8-20010816.diff	knewstickerで日本語のニュースが扱えるようにする
kdenetwork-2.2-kmail-honyaku-20010818.diff	KMailで翻訳魂を使用する
kdevelop-2.0-font-i18n-20010816.diff	決めうちされているフォントをデフォルトフォントに変更する
kdevelop-2.0-i18n-20010817.diff	サイドバーの表示、makeなどの出力を修正
kdeaddons-2.2-babelfish-ja-20010816.diff	konquerorで翻訳を行うプラグインを追加する

表2 Qt 2.3.1 / KDE 2.2用パッチファイル

KDEデスクトップ徹底活用術

KDEのバージョン2以降は、画面デザインが一変され、洗練された外観を持つようになった。また、多くの新機能が取り込まれ、商用のOSと比較しても遜色のないレベルに到達したといってい

ていいだろう。
ここでは、機能/デザイン両面でKDEをカスタマイズするためのポイントであるコントロールセンターに注目してみよう。

コントロールセンター

WindowsにもMac OSにもコントロールパネルと呼ばれるフォルダがあり、その中にはさまざまな設定用のアプレットが含まれている。何か設定を変更したい時や、カスタマイズをするためには、とりあえずコントロールパネルを開けばいいようになっており、不慣れなユーザーでも必要な機能を見つけやすくなっている。

一方、以前のUNIX系OSでは、設定ファイルがあちこちのディレクトリに

散在しており、何らかの設定を行う際には、設定ファイルの置き場所を正しく把握していなければならなかった。コマンドラインとGUIの差は別にしても、これがUNIXの使い勝手を損なっていたのは確かだ。UNIXユーザー=熟練者だったころは、それで問題はなかったのかもしれない。しかし一般的なPCで動作するLinuxやFreeBSDの普及によって、初心者がUNIX系OSに触れる機会も多くなった。その結果、「インストールはできたんだけど、次に何をすればいいのか?」という質問がよく聞かれるようになった。また、雑誌の「設定系」の記事は、いつでも人気があるものだ。

モダンなデスクトップ環境であるKDEは、商用のOSにならって、設定関係のアプレットを、「コントロールセンター」に集約している(画面1)。多くの機能がカテゴリごとに分類されており、項目の選択が簡単に行えるようになってい

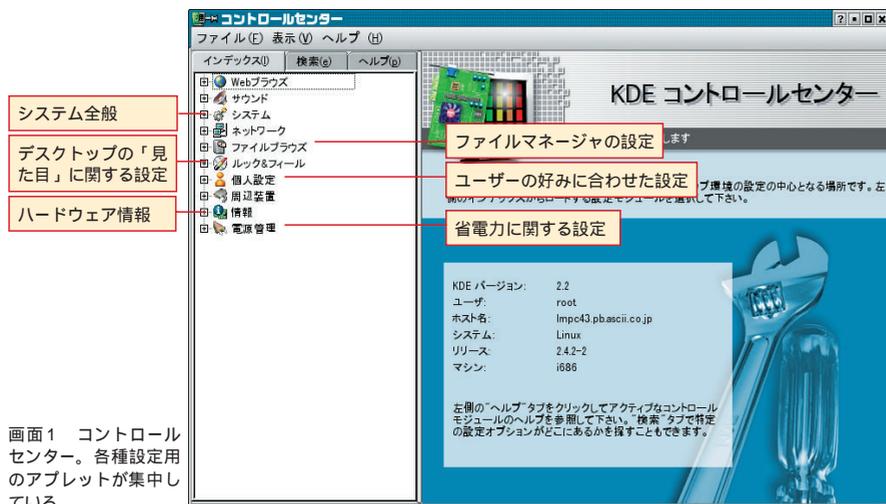
る。Windowsのコントロールパネルが集められていることがわかるだろう。コントロールセンターでは、機能ごとに分類されているため、よりわかりやすいと言ってもいいかもしれない。

システム

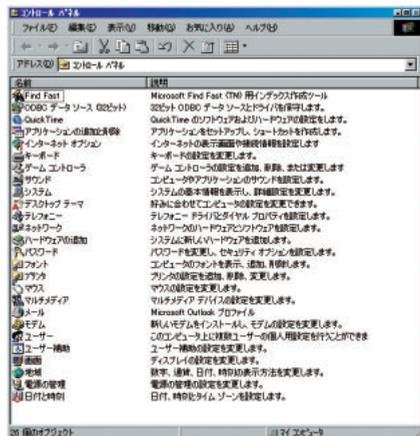
「システム」グループ内には、システム全般に関する項目が集められているが、実際にはほかのグループに入れられなかった項目をまとめたようにも見える。もっともバラエティに富んだグループだ。

どんなOSにも必要な日付と時刻の設定は、「Date & Time」で行う(画面3)。機能が同じなので、見た目もWindowsの「日付と時刻」アプレットと同じだ。

画面3を見ると、画面1ではアイコンがツリー状に配置されていたウィンドウの左半分にヘルプが表示されていることがわかるだろう。これは、上部のタブで「ヘルプ」を指定すると表示される。残念ながら、現状ではすべての



画面1 コントロールセンター。各種設定用のアプレットが集中している。



画面2 Windows 98 SEのコントロールパネル。各アプレットが一元的に配置されている。

項目にヘルプが用意されているわけではない。また一方では、別ウィンドウを開いて、より詳細な説明が表示される項目も存在する。

そのほかには、プリンタやディスプレイマネージャ（ログインマネージャ）セッションマネージャに関する設定などが行える。

さらにLILOの設定ファイルまでGUIで変更できるようになっている。これについては、やはりテキストエディタを開いたほうが便利な気もする。そういう意見が出るのを予想していたのか、「エキスパート」タブを開くと、なんとシンプルなエディタ画面が登場した（画面4）。ここで、GUIによって変更した結果を確認するといいい。

ファイルブラウズ

ファイルマネージャの機能は、ローカルやネットワーク上にあるファイルの移動/複製/削除を行うとともに、ファイルの種類に応じて適切なアプリケーションを起動して、そのアプリケーションにファイルを渡すことである。

「ファイルブラウズ」グループでは、フォルダを開く際に新しいウィンドウを作成するかどうかといったことに加え、ファイルの種類とアプリケーションの関連付けを設定する。

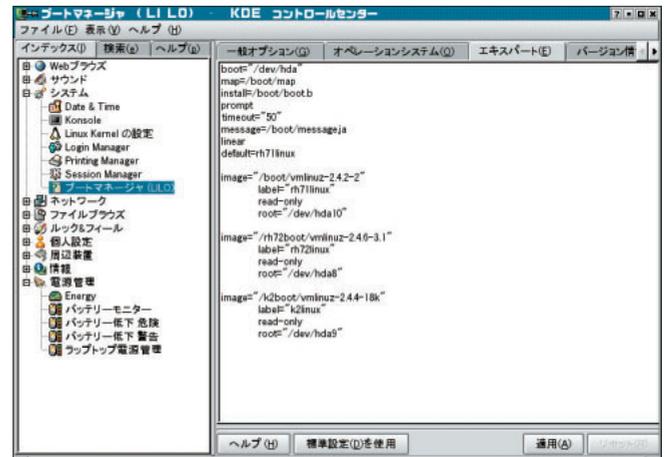
データファイルの判別方法は、Windowsと同じように、ファイル名の拡張子で判断している。つまり

report.txtはテキストで、moonlight.mp3はMP3エンコードされたサウンドデータというわけだ。

KDEは、ある種類のデータファイルと複数のアプリケーションを関連付けることができ、より柔軟な対応が可能になっている。もちろんアプリケーション間の優先度も自由に変えられる（画面5）。優先度が低いアプリケーションでファイルを開きたい時は、対象となるファイルを右クリックしてコンテキストメニューを表示し、アプリケーションを選択すればいい（画面6）。たとえば画像ファイルに対して、ビューア、グラフィックエディタの順で関連付けをしておけば、クリックで画像を開覧し、右クリックからメニューを



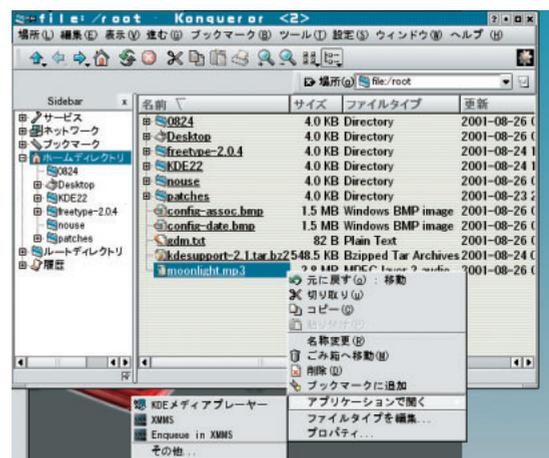
画面3 「Date & Time」Windowsの「日付と時刻」と同じ外観を持つ。



画面4 lilo.confを直接編集できる、シンプルなエディタ。



画面5 複数のアプリケーションを関連付けることが可能だ。



画面6 優先度が低いアプリケーションで開く時は、コンテキストメニューを利用する。

たどって編集を行うといった使い分けが可能だ。使い慣れてくると非常に便利な機能である。

電源管理

コンピュータの周辺機器で消費電力が大きいものといえば、ディスプレイだ。KDEでは「Energy」アプレットを用いて、ディスプレイのスタンバイ/サスペンド/電源を切るまでの時間を設定可能だ。

画面7の3つのスライダーは、上のものほど時間が短くなるはずなので、上にあるスライダーを大きく右に動かせば、下のスライダーが同調して動くようになっている。

ノートPC用のバッテリーに関連するアプレットについては、試すことができ

なかった。

情報

このグループに含まれるアプレットを用いて、ハードウェア/ソフトウェアに関する情報を参照することができる。ハードウェアの情報に関しては、/procディレクトリで参照できるデータがGUIで表示されていると考えればいい。

「Memory」アプレット(画面8)は、現在の物理メモリと仮想メモリの使用状況をグラフィカルに表示する。この画面を表示したまま、他のプログラムを起動すると、表示されている数字が目まぐるしく変化するのが見てとれる。

「X-Server」、「PCI」はそれぞれxdpyinfo、lspciコマンドの出力に相当

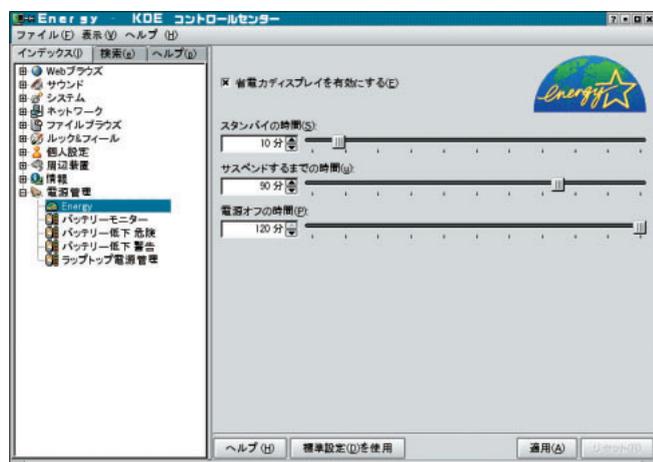
するデータを表示する。

「ブロックデバイス」だけは、情報を表示するだけの他のアプレットと異なり、画面上でパーティションやデバイスのマウント/アンマウントが行える。

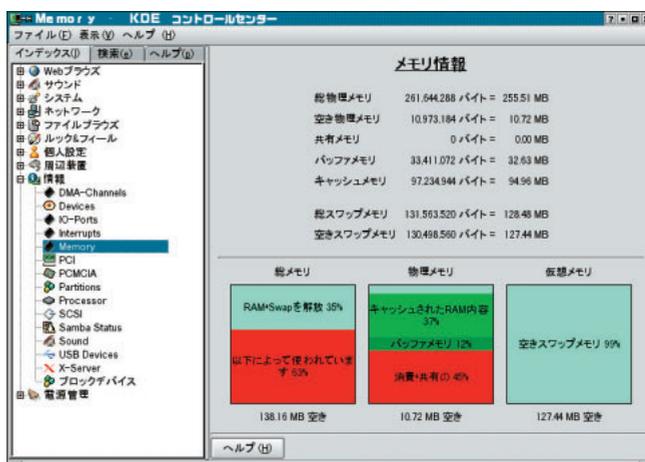
ロック&フィール

使い勝手に直接影響を及ぼすわけではないが、デスクトップ環境の見た目は、とても重要だ。クールなデスクトップなら、作業効率もきっと上がるはずだ。

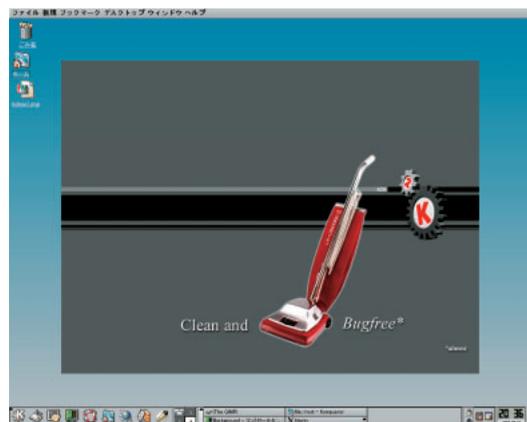
自由度が高すぎて、どこに手を加えていいかわからないので、見た目を変更しないユーザーも多いという。せっかくある機能を使わないなんて、もったいない話だ。試行錯誤を繰り返しつつ、「俺的デスクトップ」を作り上げよう。



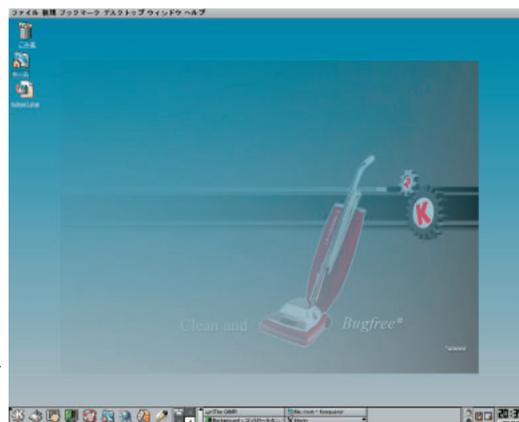
画面7 Energy Star対応ディスプレイの設定は、ここで行う



画面8 メモリの使用状況をリアルタイムにモニターできる。



画面9 背景の上に壁紙が重なっているデスクトップ。



画面10 背景と壁紙をブレンドした芸術的(?)なデスクトップ。

Background

いわゆる「壁紙」の設定は、「Background」アプレットで行う。KDE 2では、背景と壁紙が別々に存在し、背景の上に壁紙が重なっている(画面9)。また「詳細」タブの中の、「混ぜ合わせ」という項目を選択すると、背景と壁紙がブレンドされた良い感じのデスクトップ(画面10)を作り出すことができる。

デザインを変えるアプレット

ウィンドウの見た目を変更するのは、「Colors」、「Style」、「Window Decoration」そして「Theme Manager」の4項目だ。

Colorsは、Windowsでいうと画面のプロパティ内の「デザイン」タブに相当するものだ。タイトルバーやメニューなどの各項目を、好みの色に変更してオリジナルのセットを作ってもいいし、Windowsや今は亡き(!) BeOSライクなセットを利用して、思い出にひたることもできる。

それに対してStyleは、ウィジェット(Widget = Window Gadget)と呼ばれる、画面上のボタンやメニューバーのデザインを選択する。これもまた各種OSに似せたデザインのセットが提供されているので、好みの「あのOS」

ばい外観でKDEを使うというのもありだ(画面11)。

また、ツールチップやメニューの現れ方もいろいろと設定できる。Windows 98ライクなアニメーション表示や、Windows 2000と同様のフェードイン表示を選択可能だ。Windowsで最初に見た時は、「なんてCPUパワーを無駄使いしてるんだ」と憤ったりもしたが、今やKDEも同レベルに達してしまっただけで、ちょっと複雑な気持ちである。

「Window Decoration」では、ウィンドウの枠やタイトルバーのデザインを変更する。たとえばタイトルバーを左寄せでちょっと短めにすれば、BeOSっぽくなる。Colorsで配色もBeOSに合わせれば、もう完璧にBeOSの世界である(画面12)。

Window Behavior

GUIを持ったOSは、どれもボタン、タイトルバーなどよく似たウィジェットを持っている。しかし、それらのウィジェットをマウスでクリックしたときの挙動は、微妙に異なっている。そのため、あるOSではできた操作が、別のOSではできなかつたり、異なる結果が返ってくることもあり、ユーザーは混乱してしまう。完全に違っているほ

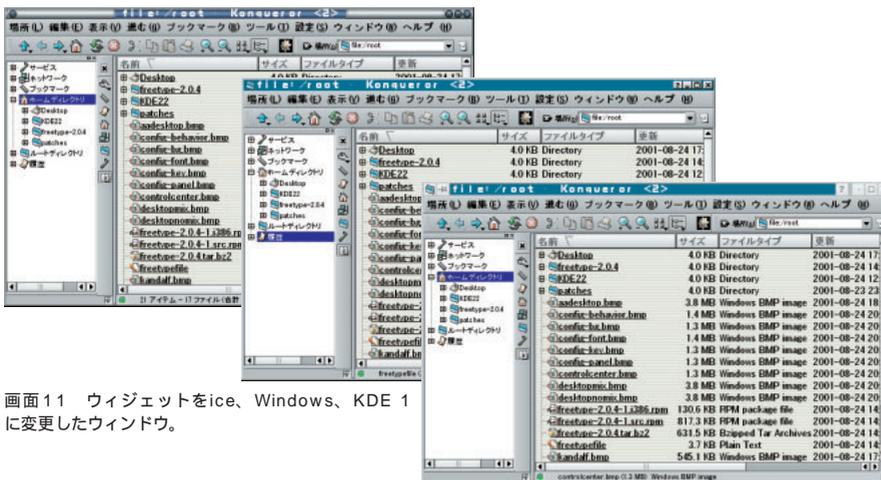
うが、まだました。

KDEでは、ウィジェットをクリックした際の挙動を非常に細かく設定可能で、今まで慣れていた環境の挙動に揃えることができる。他の環境から乗り換えてきたユーザーにとっては、便利な機能だ。

KPersonalizer

ルック&フィールに関する設定項目は多岐に渡るが、それらをまとめて「よしなに」行ってくれるのが、KPersonalizerだ。KDE 2.2を最初に起動した際に出てくるウィザードがKPersonalizerの正体だ。いったん設定したあとでも、「システム」メニュー内の「Desktop Settings Wizard」で呼び出すことができる。

まず最初にロケールの設定(画面13)を行い、続いてウィジェットをクリックした時の挙動や操作体系を決める。各項目を選ぶのではなく、WindowsやMac OSなど既存の環境に近くなるように設定する(画面14)。さらにさまざまな画面上のギミックをどの程度利用するかを選択する。スライドバーを用いて、シンプルにするか満艦飾にするかを選ぶこともできるし、個別の項目を選択していくこともできる(画面



画面11 ウィジェットをice、Windows、KDE 1に変更したウィンドウ。



画面12 BeOSライクなウィンドウデザインにして、思い出にひたっている。

15)。速いCPUとたくさんのメモリを持っているなら、とりえず全部試してみるといいだろう。最後にテーマをKDE / Mac OS / Windows / Solarisの中から選べば終了だ(画面16)。

アンチエイリアス

ここまでKDEの画面をたくさん見てきたが、何か気が付いたことはないだろうか? 一般的なXのデスクトップ画面と比べると、フォントが美しく表示されているように見える。今回の画面写真は、アンチエイリアス機能を有効にして撮られているのだ。

XFree 86 4.0.2以降では、Render拡張という機能が取り入れられ、Qt 2.3.0 / KDE 2.1以降のバージョンではそれを利用して、アンチエイリアスのかかったフォント表示が可能となった。新しいKDEを利用する最大のメリットといってもいいだろう。それほど美しいのだ。

フォントについて

コンピュータで用いられているフォントには、「ビットマップフォント」と「アウトラインフォント」の2種類がある。

ビットマップフォント

ビットマップフォントは、文字の形状を小さな点(ピクセル)を縦横に並べて表現している。

上の例は、4×5の点で「A」を表している。実際に用いられるフォントは、もっとピクセル数が多く、より複雑な字体も表現できるようになっている。

ビットマップフォントは、データ量が少なく、高速に表示が行える。また、表示サイズに合わせて最適にデザインすることができるため、高解像度のビットマップフォントは見栄えがいい。

一方で、元のサイズ以外で用いようとして拡大/縮小すると、字の形が崩れて見づらくなってしまいうという欠点がある。最近のOSのようにさまざまなサイズのフォントで画面を構成する場合には、見づらいのを我慢するか、表示サイズに合わせて複数のデータを用意するしかなかった。

アウトラインフォント

文字の形状を輪郭線として持っているのが、アウトラインフォントだ。アウトラインフォントで文字を表示する際には、輪郭線のデータからどのピクセルを塗りつぶせばいいのかを計算し、その結果を出力する。

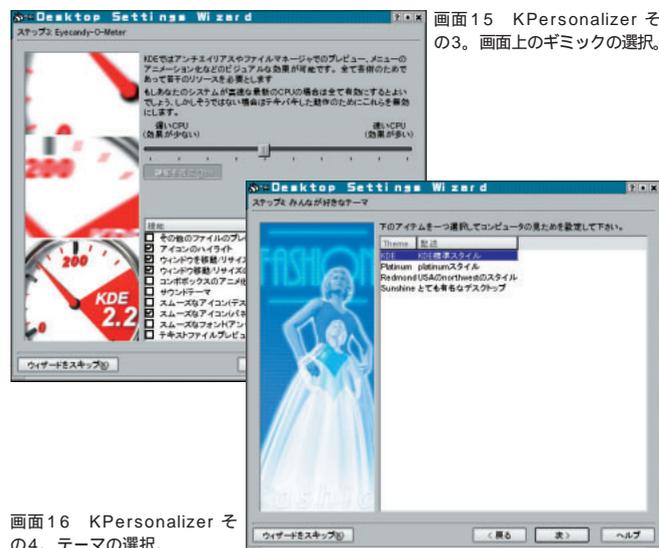
アウトラインフォントは、表示するサイズに合わせて計算が行われるため、文字を拡大/縮小したり、斜体のように変形しても字の形が崩れないのが特徴だ。

だが、ビットマップフォントと比較すると、データ量が多いうえに、表示のたびに計算を行うため、ある程度強力なマシンが必要になる。今どきのPCなら、ハードディスクの容量も大きく、CPUも高速なので、これらの特性はもはや欠点とは言えないだろう。

アウトラインフォントには、いくつの種類があるが、Mac OSとWindowsで使われているTrueTypeが最も普及している。X Window Systemでも、freetypeなど、TrueTypeを利用する仕組みがある。

アンチエイリアス処理

画面上の文字は、ピクセルの集まり



で表現されるため、ディスプレイの性能によっては、斜めの部分にでこぼこ(ジャギー)が見えてしまうことがある(図1上)。これは、アウトラインフォントでもビットマップフォントでも変わらない。

そこで、アウトラインフォントのデータから字体を作成する際に、各ピクセルを描画する/しないだけでなく、中間色のピクセルを置くという手法が考案された。たとえば白地に黒の文字を表現する場合に、白/黒だけでなく、グレーのピクセルを置くといった具合だ(図1下)。効果のほどは、本特集の画面を見てもらえば、一目瞭然だろう。

設定方法

KDEにTrueTypeを用いたアンチエイリアスの機能が組み込まれたのは、バージョン2.1からである。比較的新しい機能であるため、特に日本語のTrueTypeフォントはうまく利用できないこともあるようだ。日本KDEユーザ会のWebサイトには、KDEでアンチエイリアスを利用するためのHOWTOが公開されているので、こちらも参考にしよう(<http://www.kde.gr.jp/kom/aa.html>)。

今回利用したTrueTypeフォントは、

パッケージ版Turbolinux Worostation 6.0に付属していたRYOBIフォントだ。上記のHOWTOで動作実績があり、字体の美しさにも定評がある。

準備

まず、各パッケージがアンチエイリアスに対応しているかどうかを確認する。XFree86は、バージョン4.0.2以降のものが必要だ。またコンパイル時にレンダラ拡張の機能が有効にされていなければならない。xdpyinfoコマンドを実行し「number of extensions」という項目の中に「RENDER」があることを確認しよう。freetypeは2.0.1以降、Qtは2.3.0以降が必要だ。

各パッケージが正しくインストールされていれば、コントロールセンターの「ルック&フィール」グループ内の「Fonts」に「フォントとアイコンをアンチエイリアス表示」という項目があるはずだ。なお、KDEのバージョンによっては、この項目は「Style」にある。

フォントファイルを配置

TrueTypeフォントのファイルを、ディスク内の適切なディレクトリに置き、ディレクトリ名を/etc/X11/XftConfigファイルに記述する。以下のように書けばいい。

```
dir "/usr/X11R6/lib/X11/fonts/TrueType"
```

freetypeは、/etc/X11/XftConfigを元にフォントファイルを検索してくれるので、設定はこれだけで済む。

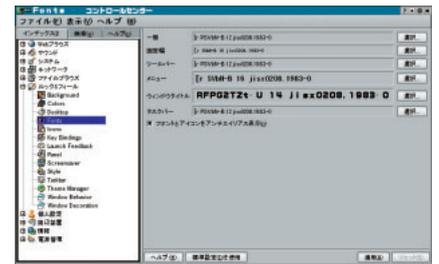
コントロールセンター

「フォントとアイコンをアンチエイリアス表示」にチェックを入れ、「適用」ボタンを押すと、KDEの再起動を促される。

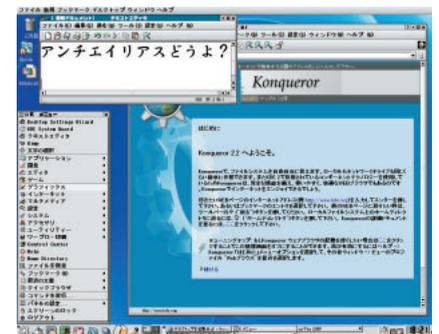
編集部で用いた実験環境では、再起動すると、メニューやデスクトップに日本語がいつさえい表示されなくなってしまった。しかし、再度コントロールセンターを開き、「Fonts」アプレットで各項目を選択してやると、正しく表示されるようになる(画面17)。

原因は不明だが、フォントファイル内部でフォント名が日本語で書かれているためと思われる。

多少の小細工は必要だが、日本語も含めたアンチエイリアス表示のクールなデスクトップが実現された(画面18)。



画面17 文字化けしている「Fonts」アプレットの表示。



画面18 アンチエイリアス処理されたクールなデスクトップ。

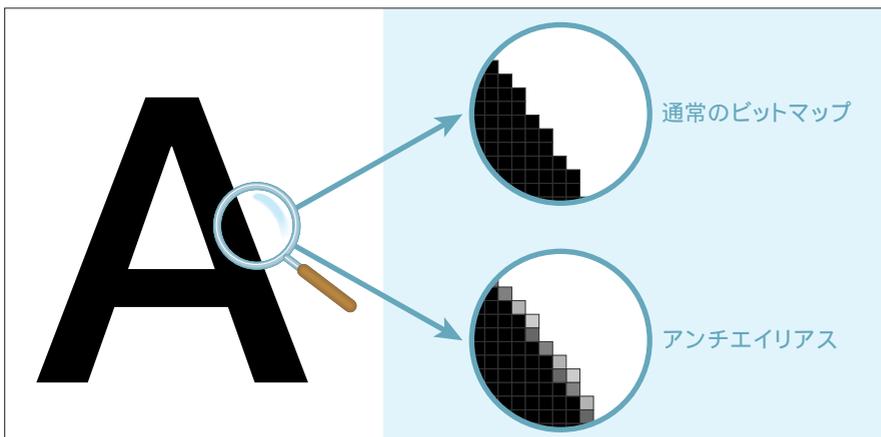


図1 アンチエイリアスは、中間色のピクセルを置くことで、表現力を向上させる。

Konqueror

Konquerorは、KDE 2の中心なアプリケーションである。ファイルマネージャとしての機能だけでなく、WebブラウザやFTPクライアントの機能も持っている。ローカルのディスク上にあるファイルと、ネットワークの先にあるファイルを区別することなく扱える。

KDEのバージョン1のファイルマネージャであったkfmも、Webブラウズ機能を持っていたが、WebページのレンダリングはNetscapeなどの専用Webブラウザと比べるとだいぶ見劣りがするものだった。

Konquerorは、新たに作成されたKHTMLレンダリングエンジンにより、より正しくレンダリングが行えるようになった。この場合「正しい」というのは、事実上の標準であるMicrosoft Internet Explorerの解釈に近くなったという意味である。Konqueror本体とレンダリングエンジンは独立しており、KHTML以外にもMozillaで利用さ

れている「Gecko」でレンダリングをすることも可能だ。

「Konqueror」という名前は、もちろんKonqueror（征服者）のKDE的表記である。未知の大陸をNavigator（航海者）が発見し、Explorer（探検家）が探索したら次はKonquerorの出番というわけだ。実にクールなネーミングセンスだ。おそらく次世代のファイルマネージャ / Webブラウザは、Emigrant（移住者）と名付けられることだろう（本気にしないように）。

エクスペローラ的な外観

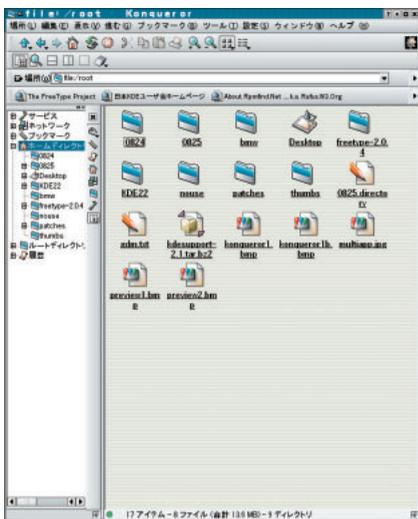
ファイルマネージャとしてのKonquerorは、ウィンドウ上部にいくつかのツールバー、左側に機能別に分けられたディレクトリツリー（サイドバー）、右側にファイル表示部というありがちな、はっきり言ってWindowsのエクスペローラによく似た外観を持っている（画面1）。別にこれは悪いこと

ではない。WindowsユーザーがKonquerorを使い始める際に、戸惑わずに済むのだから。ほかの環境を使っていたユーザーに対して、わざわざ敷居を高くする必要はないのだ。

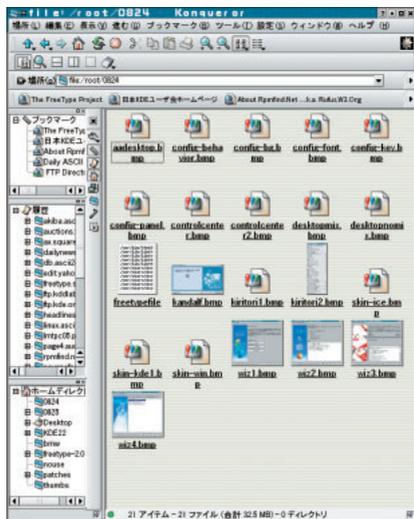
KDE 2.2からサイドバーは「拡張サイドバー」と呼ばれるようになり、右側に付いているボタンを押すことで、サイドバーがグループごとに分割できるようになった（画面2）。複数のカテゴリ間を行き来する際に、効率良く移動ができる。以前と同じように、すべてのカテゴリをまとめて表示したい場合には、「クラシックサイドバー」を選択すればいい。

拡張サイドバー内のカテゴリは、以下の6つがある。

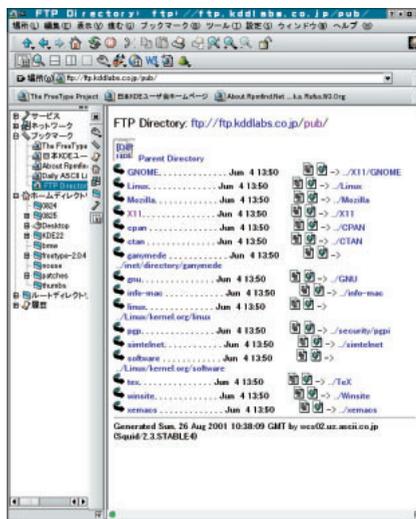
- (1) ブックマーク
- (2) 履歴
- (3) ホームディレクトリ
- (4) ルートディレクトリ
- (5) ネットワーク



画面1 ファイルマネージャのKonqueror



画面2 拡張サイドバーは、グループごとに分割して表示できる



画面3 FTPクライアントのKonqueror。ダウンロードは、アイコンをドラッグ&ドロップするだけだ

(6) サービス

ブックマークや履歴は、Webブラウザでおなじみの項目だ。ホームディレクトリは、各ユーザーのホームディレクトリを表している。またルートディレクトリから、ローカルのファイルシステムすべてをたどることができる。

ネットワークカテゴリーには、FTPアーカイブ、Webサイトというサブカテゴリーがあり、ネットワークの先にあるリソースを、ローカルのファイルと同等に扱うことができる。画面3は、KonquerorでFTPサイトを表示したようすを示している。FTPサイトにあるファイルをローカルのディレクトリやデスクトップにドラッグ&ドロップすれば、ダウンロードが始まる。

また、ローカルネットワークというサブカテゴリーを通じて、LAN上のSambaやNFSで共有可能なディレクトリにアクセスできる。

サービスカテゴリー内には、オーディオCD上のデータファイルへアクセスする仕組みが用意されている。将来的には、デジタルカメラの内蔵メモリへアクセスする仕組みが提供される予定である。

ファイルビューア

一般的なファイルマネージャの機能に



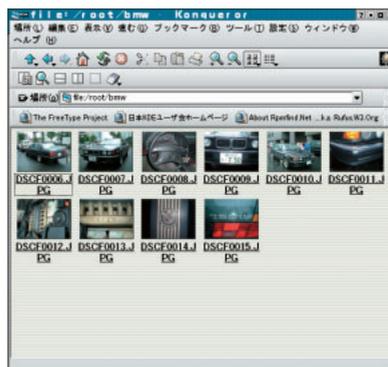
画面4 プレビュー可能なファイル。アイコンの代わりにサムネイル画像が表示される

加えて、Konquerorには各種データファイルのビューア機能も持っている。たとえば、bmp、jpgといった拡張子を持った画像ファイルをクリックすると、ファイル一覧が表示されていた右側に、その画像が表示される。同様に、テキストファイルならその内容が表示される。

また、画像ファイルやHTMLファイル、PDFファイルのプレビューをアイコンの代わりに表示する機能も持つ。Konquerorの「表示」メニューからプレビューを選ばると、プレビュー表示可能なデータ形式の一覧が表示される(画面4)。ここで画像ファイルを意味するImagesを選択すると、画面4のディレクトリの表示が画面5のように変化する。Konquerorがあれば、あらためてサムネイル表示可能な画像ビューアを用意する必要はない。

Webブラウザ

WebブラウザとしてのKonqueror(画面6)は、インターネットの各種標準規格に準拠した、本格的なものだ。すなわち、JavaScript、Java、XML、HTML 4.0、CSS (Cascading Style Sheets)、SSL (Secure Socket Layer) をサポートしている。さらにNetscape用のプラグイン (LessTifライブラリが必要) や、いくつかのActiveXコンポーネントにも対応している。



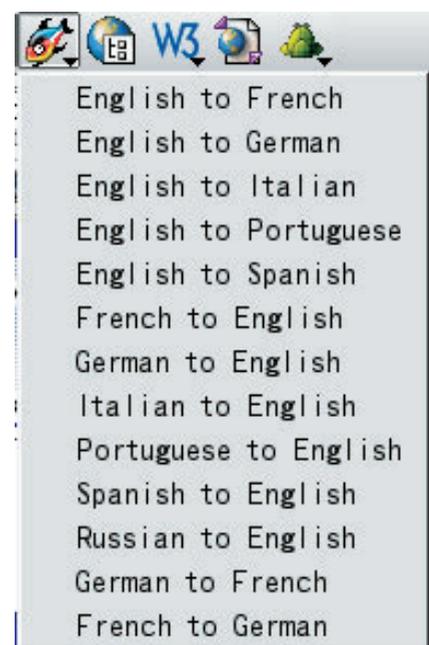
画面5 画像ファイルのサムネイルが表示されている

また、NetscapeやMozillaのブックマークをインポートする機能もあり、Webブラウザとしても十分に実用的なレベルと言えるだろう。

Webページを閲覧している場合には、ツールバーにアイコンが追加される。AltaVistaのBabel Fishを用いたWebページの翻訳やHTMLチェッカーなど、ネットサーファーなら便利さを実感できる機能満載のWebブラウザ、それがKonquerorである。



画面6 WebブラウザのKonqueror



画面7 Babel Fishを用いた翻訳機能

オフィススイートKOffice

UNIX系OSがクライアント用途でも使われるようになるためには、ワードプロセッサや表計算ソフトなど、一般的な業務で使われるソフトウェアが必要だ。これらのソフトは、ひとまとめにして「オフィススイート」と呼ばれる。商用/非商用を問わず、Linuxで利用可能なオフィススイートの種類は増えてきている。

KOfficeは、KDEで利用可能なオフィススイートである。KDE 2.2にはKOffice 1.1が含まれる予定だったが、

本稿執筆の時点では、正式リリースはされておらず、最新版はKOffice 1.1 RC1 (Release Candidate) (以下RC1と表記する) である。RC1は、分類上は「unstable」であるため、付録CD-ROMへの収録は見送った。以下の記事は、RC1を元に作成しているのので、正式版とは異なる可能性があることをお断りしておく。



RC1には、以下の8つのアプリケーションが含まれている。

- (1) KWord
- (2) KSpread
- (3) KPresenter
- (4) Kontour
- (5) Kivio
- (6) Kugar
- (7) KChart
- (8) KFormula

KOffice本家のWebサイト (画面1) のアナウンスでは、上記8つのうちKFormulaがなく、代わりにKrayonが

入っていた。

KWord

オフィススイートの要、ワードプロセッサである。他のアプリケーションが作成したデータを配置することのできる「フレーム」として機能を持ち、画像や表を取り込んだドキュメントを作成可能だ (画面2)。

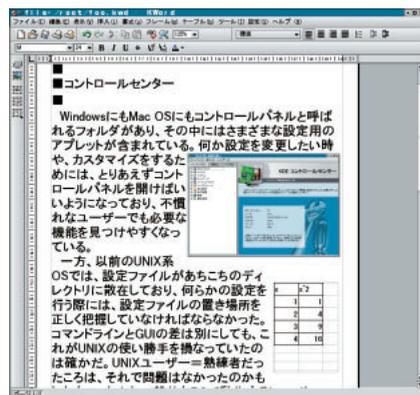
作成したドキュメントは、KWord独自形式 (拡張子がkwd) のほか、リッチテキストフォーマット、通常のテキスト、SGML、TeX、HTML、AbiWord (GNOMEのワードプロセッサ) の各形式でファイルに書き出すことができる。

ほかのワードプロセッサで作成したドキュメントについては、Microsoft Word、Applixware、Abiwordのものが読み込めることになっている。だが、Microsoft Word 2000で作成した文字だけのファイルをKWordで読み込むと、日本語部分が欠け落ちていた (画面3)。

KWord単体での日本語文章作成については、ほぼ満足できるレベルと言ってよい。だが、ほかのアプリケーションで作成したドキュメントのインポ

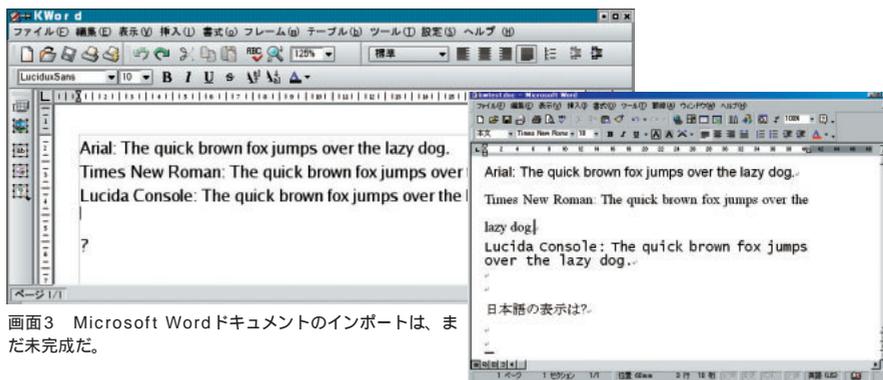


画面1 KOfficeのWebサイト



画面2 KWord

画像とスプレッドシートを貼り込んだドキュメント。



画面3 Microsoft Wordドキュメントのインポートは、まだ未 completion。

ートは、まだ未完成である。アルファベットは正しく表示されているので、おそらく英語以外の言語に特有の問題と思われる。

KSpread

KOfficeのスプレッドシートがKSpreadだ。「Excelみたいなもの」というほうが、わかりやすいだろうか。Excelに似通った操作体系を持ち、Excelに慣れていれば、直感的に操作ができるはずだ(画面4)。

作成したドキュメントは、KSpread独自形式(拡張子がksp)のほか、カ

ンマ区切りテキスト(CSV形式)通常のテキスト、Gnumeric(GNOMEのスプレッドシート)の各形式でファイルに書き出せる。

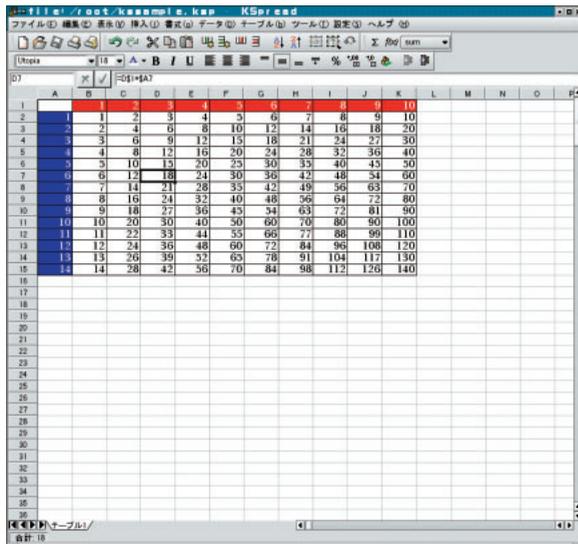
他のアプリケーションで作ったデータのインポートは、Microsoft Excel、Gnumeric、Applix Spreadsheets、Quattro Proに対応しているようだ。しかし、Microsoft Excel 2000形式のドキュメントは、データが違っており、正しく読めていなかった。念のため、Excel 95形式で保存をしても、結果は同じだった。CSV形式のテキストなら何とか読み込めた(画面5)が、CSV

では計算式が保存できないため、元のデータが再現できたわけではない。

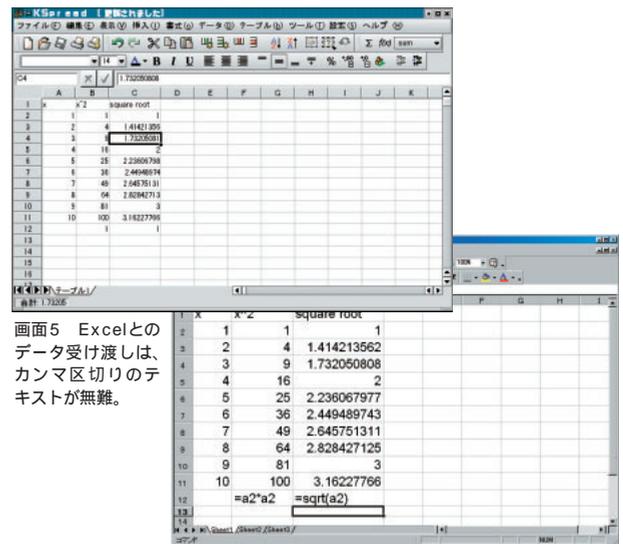
KPresenter

プレゼンテーション用の資料を作成するのが、KPresenterだ。「PowerPointのようなもの」である。ルック&フィールはPowerPointに似ていて、直感的に操作ができる(画面6)。

PowerPoint 2000のデータを読み込ませてみたが、文字データはアルファベットだけだったにもかかわらず、まったく再現できなかった。



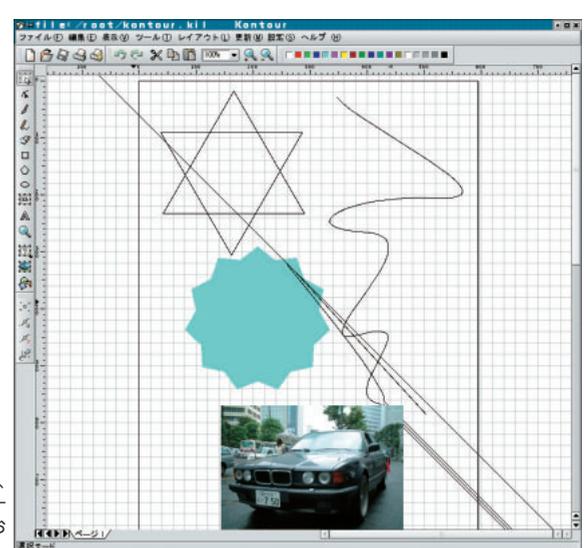
画面4 KSpread Excelに似た操作体系を持つ。



画面5 Excelとのデータ受け渡しは、カンマ区切りのテキストが無難。



画面6 KPresenter PowerPointだと思って操作すれば間違いはない。



画面7 Kontour 以前の名前は、Killustratorだ。一般的なドロー系お絵描きツールである。

Kontour

ベクトル形式のデータを扱うドロー系描画ソフトだ。ドロー系のツールではあるが、ビットマップ画像も扱うことができる(画面7)。

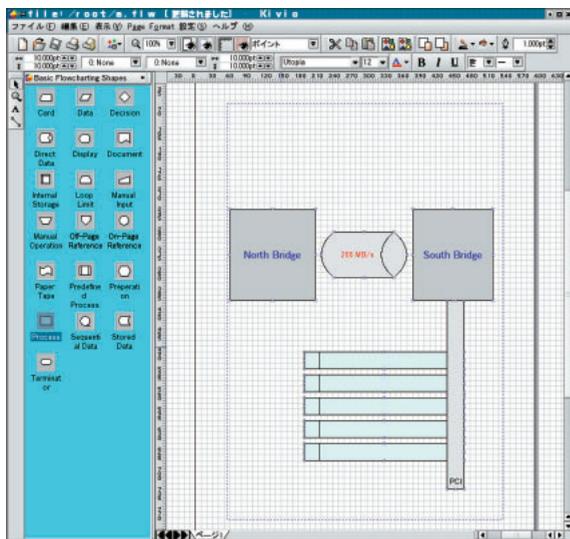
「Kontour」は例によってContour(等高線)のKDE的表現である。以前は、KIllustratorという名前だったが、アドビ社のIllustratorとバッティングするため、変更したようだ。

ドロー系ツールの一般的な操作方法、つまり多角形や直線、ベジェ曲線などの部品を配置していくという方法で、目的の画像を組み立てていけばいい。

作成したデータは、XPM、PPMイメージやEPS(Encapsulated Post Script)形式のファイルに出力できる。特にEPSに対応しているため、ほかの環境でデータの再利用ができないといった心配はないだろう。

Kivio

Visioライクなフローチャート作成ソフトである(画面8)。あらかじめ準備されている部品を組み合わせることで、フローチャートや各種の模式図作成に威力を発揮する。



画面8 Kivio
Visioと同様に、フローチャートや模式図作成に特化した画像作成ソフトである。

KChart

さまざまな形式のグラフを作成するためのソフトだ。Excelの「グラフウィザード」機能が独立したものと考えれば、いいだろう(画面9)。

現在のバージョンは0.9で、安定版とみなされているのだが、編集部のマシンでは、データを変更しようとするだけでクラッシュするといった不安定な挙動を見せた。

KFormula

数式を作成するためのソフトだ。単独で使う機会はあまりないだろう。画面10のように、KFormulaで作成した結果を、KWordのドキュメントなどに貼り込むのが一般的だ。

今後のさらなる発展に期待

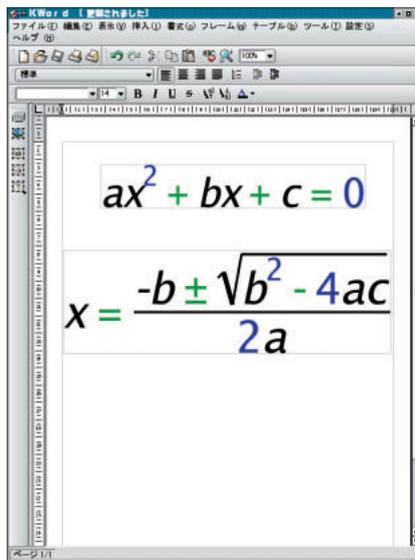
今のところ、上記の7種類以外に、ビジネスレポート作成ツールのKugar、ビットマップ系画像ソフトのKrayon、そしてプロジェクト管理ツールであるKPlatoを加えた10種類が、KOfficeを構成するアプリケーションである。

実験中にクラッシュしたのはKChartくらいで、それ以外は安定して動作し

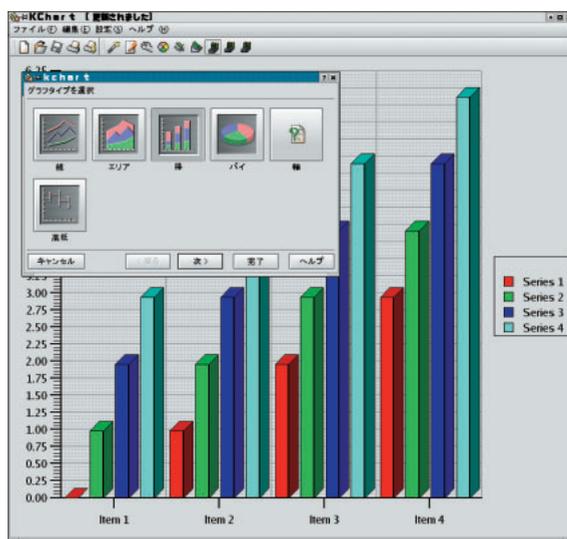
ていた。

問題なのは、Microsoft Officeの各アプリケーションのドキュメントを正しく扱えない点だ。デファクトスタンダードであるWord/Excelドキュメントのインポートは、オフィススイートの必須機能と言ってもいい。

Microsoft Office製品のファイルフォーマットは、正式には公開されていないため、困難だとは思うが、ドキュメントレベルでの互換性をぜひ実現してもらいたいものだ。



画面10 KFormula
複雑な数式でもそれほど手間をかけずに作成できる。



画面9 KChart
Excelのグラフウィザードに相当する機能を持つ。

新連載

携帯電話 de Javaプログラミング

NTTドコモのiモード503iシリーズ、KDDIのC451よりスタートしたezplus、J-SH07よりサービス開始されたJ-フォンのJava対応コンテンツサービスなど、国内の主要キャリアが携帯電話のJavaコンテンツサービスを開始しました。この連載では実際にこれらのキャリアのJava対応について説明していきながらLinux上で各携帯電話で動作するJavaアプリケーションを作成していきます。

第1回 iアプリを構築する環境の作成

文：加藤大受

Text: Dajju Kato dkato@jcom.home.ne.jp

iアプリを始めとするJavaアプリケーションの開発を始める前に、NTTドコモ、KDDI、J-フォンの各キャリアがどのような形でJava対応を行っているか整理してみましょう。

まず、2001年1月18日に発表されたiアプリのサービスは、1月26日より発売されたF503iとP503iによって開始されました。現在、NTTドコモではiアプリ対応した503iシリーズ6機種(表1)を販売しており、さまざまなオフィシャルコンテンツサービス(表2)が提供されています。

続いて、6月22日よりJ-SKYとしてJava対応コンテンツのサービスを開始したJ-フォンですが、現在はシャープ製のJ-SH07(6月22日発売)の1機種

だけが販売されており、占い・ゲーム、エンターテインメント、交通・トラベル、バンキング・マネー、検索・実用・moreの5種類のメニューからなる29種類のJavaコンテンツサービスを行っています。さらに8月中には4096色対応の三菱電機製J-D05と、65536色対応でフラッシュ付きカメラが装備された東芝製のJ-T06が発売されるので、J-フォンのJava対応とJavaコンテンツがさらに増えていくことになるでしょう。これらの機種は、3Dを利用できるなどエンターテインメント性を非常に重視しています。

最後にKDDIですが、7月4日よりezplusサービス開始しました。現在は日立製作所製のC451H(7月4日発売)

と、カシオ製のC452CA(8月11日発売)の2機種のみとなります。なお、ezplusサービス対応のJavaには、NTTドコモやJ-フォンとは異なり、モバイルエージェント機能が提供されています。モバイルエージェント機能とは、同じJavaアプリケーション同士で通信できる機能です。モバイルエージェント機能を利用したJavaアプリケーションの構築方法についてはこの連載の中で解説していきます。

J2MEとProfileについて

Javaの開発環境は現在、次の3つの分類があり、これらは開発するアプリケーションの規模に応じて分けられています。一般的なPC上のGUIをとまなうアプリケーションの開発を行う場合は、通常、J2SEを使用します。

メーカー	機種名	発売日
富士通製	F503i	2001年1月26日発売
松下通工製	P503i	2001年1月26日発売
NEC製	N503i	2001年3月3日発売
ソニー製	SO503i	2001年3月9日発売
三菱電機製	D503i	2001年3月19日発売
松下通工製	P503iS	2001年5月11日発売

表1 現在発売中のNTTドコモ503iシリーズ

取引系サイト	株価情報や為替情報など
情報系サイト	天気情報やニュースなど
データベース系サイト	乗換案内や地図、地域情報など
エンターテインメント系サイト	ゲームやカラオケなど

表2 NTTドコモが提供しているオフィシャルコンテンツサービス

- Java 2 Micro Edition(J2ME)
組み込み機器で動作するアプリケーションの開発向け
- Java 2 Standard Edition(J2SE)
PC向けなどの一般的なアプリケーション開発向け
- Java 2 Enterprise Edition(J2EE)
Webシステムなどの大規模なサーバアプリケーション開発向け

NTTドコモ、J-フォン、KDDI3社の携帯電話で動作するJavaアプリケーションは、J2MEと呼ばれる組み込み機器で動作するJavaの仕様に準拠しています。

J2MEはJ2SEのサブセットとして定義されており、組み込み分野においてJavaを利用できるように定義されています。J2MEで規定されているライブラリはすべてJ2SEにも提供されており、完全なサブセットとなっているため、J2ME準拠で書かれたアプリケーションはJ2SEの環境でも使用できるように設計されています。ただし、携帯電話で稼働するJavaアプリケーションは、各キャリアや各携帯機器メーカー独自のライブラリを使用していることが多く、J2SEの環境で手軽に利用することは難しくなっているため、上位互換性は非常に乏しい状態です。しかし、今まで組み込み機器で使用されるアプリケーションはまったく互換性がなく、機器を変更する場合は作り直しが多かったことを考えると、Javaの持つオープン性のメリットは非常に大きいと思われれます。さらに、Javaプログラミングができる人であれば手軽にアプリケーション開発ができるので、組み込み機器で稼働するアプリケーション開発の敷居が低くなり、より多くの開発者がこういった組み込み分野で活躍できるようになっています。

J2MEでは、組み込み機器を次の2種類のカテゴリーに分けています。

- CLDC (Connected Limited Device Configuration) 仕様
- CDC (Connected Device Configuration) 仕様

CLDC仕様は携帯電話やPDAなど、最も小さな組み込み機器を対象にしており、CDC仕様はセットトップボックスのような、CLDC仕様よりも大きくPCよりも小さい機器を対象にしています。CLDC仕様、CDC仕様とも「Profile」と呼ばれる、用途別の機能が用意されています。携帯電話用に用意されているProfileが、MIDP(Mobile Information Device Profile)と呼ばれており、MIDPはサン・マイクロシステムズ、モトローラ、ノキアを中心とするMIDP Expert Groupによって作成されています。日本からはNTTドコモ、KDDI、J-フォン、富士通、日本電気、松下電器、三菱電機、ソニー、日立製作所、シャープらが参加しています。このMIDPに対応しているのが、J-フォンとKDDIで、実行できるアプリケーションのサイズの違いなどはありますが、相互で利用できる可能性が非常に高くなっています。

このCLDC仕様で動作するJava VMをKVMと呼びます。KVMは少ないメモリと低速なCPUで動作するように開発されたJava VMです。このKVMが各携帯電話に搭載されているため、iアプリなどの携帯電話に対応したJavaアプリケーションが動作するわけです。

CLDCやKVMの仕様については表3のWebサイトからドキュメントをダウンロードすることができます。

CLDCの仕様について	http://java.sun.com/products/cldc/docs/CLDC1.0_ja_docs.zip
KVMの仕様について	http://www.sun.co.jp/software/consumer-embedded/kvm/

表3 CLDCおよびKVMの仕様の入手先

ンロードすることができます。

iモード対応Javaの環境について

NTTドコモのiモードサービスに対応したJavaであるiアプリは、先ほど説明したJ2MEのCLDCに対応したProfileであるMIDP対応ではなく、NTTドコモ独自仕様となっています(図)。

これは、iアプリの発表時にはMIDPの仕様はまだ固まっていなかったため、NTTドコモがCLDC仕様に準拠した独自のProfileを策定したためです。つまり、NTTドコモのProfileはCLDC + NTTドコモ独自となっています。

また、各携帯機器メーカーでの独自拡張を認めているため、ある機種で動作するiアプリがほかの機種では動作しない可能性も高くなっています。

NTTドコモではDoJaと呼ばれる、iモード専用のクラスライブラリを用意しています。DoJaではユーザーインターフェイス、ネットワーク、Javaアプリケーションから利用可能な記憶領域であるScratchPad、テキスト変換などのAPIを用意しています。J2MEのクラスライブラリとDoJaを組み合わせることによって、iアプリを開発していくこととなります。

iモードJavaでは、携帯電話内ですべての動作を行うスタンドアロン型のアプリケーションと、ネットワークからデータを受け取りながら動作するネットワーク型アプリケーションの両方に対応しています。ただし、iモードJavaには以下のような制限があり、この制限内でアプリケーションを開発する必要があります。

- ・携帯電話からは同時に1つのiアプリしか起動できない
- ・他のiアプリを起動したり制御することはできない
- ・ScratchPadの領域は現行機種では10Kバイトまでダウンロード時に決まり、あとからの変更はできない
- ・iアプリから携帯電話の基本機能にアクセスできない
- ・iアプリはそれをダウンロードしたサーバとしか通信ができない
- ・iアプリ自身のファイルサイズはJAR圧縮で10Kバイトまで

特に、iアプリ自身のファイルサイズが10Kバイトまでですので、Java言語の利点であるオブジェクト指向プログラミングなどを駆使すると、すぐにこの制限を超えてしまいます。また、プログラムのサイズを圧縮するために、JARファイルによる圧縮を利用しています。JARファイルとは、作成したiアプリのプログラムファイルとプログラムで使用するGIFファイルなどの画像データをまとめたファイルです。JARファイルはZIPファイルとコンパチブルとなっています。

こうした制限のため、iアプリを作成するときは綺麗なコーディングよりも、いかに小さいプログラムを書くかが重要になってきます。また、使用する画像ファイルのサイズについても考慮しなければいけません。

10年以上前に、アセンブラでDOSプログラムを書いていた経験がある人であれば、この10Kバイトという制限の中でプログラムを書く大変さが理解できると思いますし、その頃の経験が再び役立つことでしょう。

iモードJavaにはJAM (Java Application Manager)と呼ばれる、iアプリの起動と強制終了の管理を行うコンポーネントが用意されています。こちらのJAMはKVMから独立した機能となっていますので、iアプリから制御することはできません。このJAMがあることにより、iアプリの起動中に電話がかかってきた場合、iアプリの実行を中断させ、電話を受けることができるようになっています。

また、ScratchPadと呼ばれるiアプリがデータエリアとして利用できる領域がありますが、こちらの領域は各iアプリごととなっており、他のiアプリから

データを読み書きできないようになっています。

JAMのような重要な機能や、ScrachPadのようなデータ領域が守られているため、iモードJavaは非常に高いセキュリティの上で動作する仕様となっています。これらの制限により、初心者でも安全にiアプリを作成することができます。

なお、iアプリのJavaの詳細な仕様は、表4のWebサイトからドキュメントをダウンロードすることができます。興味がある方は一読されることをお勧めします。



iモードJavaであるiアプリを開発する方法について説明しましょう。iアプリを開発するには、JDKに含まれるJavaコンパイラ、CLDCやiモードJava拡張クラスライブラリ、JDKに含まれるJARアーカイバが必要です。これらのツールは表5のWebサイトから入手することができます。

エディタとこれらのツールを利用してiアプリを作成することもできますが、iアプリを簡単に作成できるツールが公開されています。これらのツールを利用することで非常に簡単にiアプリを作成することができます。

画面1はNTTドコモが提供しているJ2ME Wireless SDK for DoJaです。このツールはNTTドコモのWebサイトからダウンロードすることができます(表5)。J2ME Wireless SDK for DoJaは、開発したiアプリが仕様に従っているかどうかを確認することができるエミュレータです。J2ME



図 iモードJavaの仕様

iモード対応Javaコンテンツ開発ガイド(詳細編)1.1版	http://www.nttdocomo.co.jp/i/java/pdf/jguide010514.pdf
iモード対応Javaコンテンツ開発ガイド(APIリファレンス編)1.1版	http://www.nttdocomo.co.jp/i/java/pdf/jguide_api010514.pdf

表4 iモード対応Javaコンテンツ開発ガイドの入手先

携帯電話 de Javaプログラミング

Wireless SDK for DoJaはForte for Java Community Editionと呼ばれるサン・マイクロシステムズが無償で公開している開発ツールから呼び出すことができるようになっていまして、この組み合わせでiアプリを開発し、エミュレータで動作を確認することができます。ただし、こちらのツールはWindows環境のみのサポートとなっており、Linuxで使用することができません。

Linuxで使用できるエミュレータとしては、ゼンテック・テクノロジージャパンのi-JADEがあります。こちらも本来はWindows環境のみのサポートですが、製品が100% Pure Javaで書かれているので、一度Windows環境にインストールしたファイルをLinux環境へコピーすることで、Linux環境でも

使用することができます。なお、i-JADEの一番下のエディションであるi-JADE Lite (画面2)は無償で提供されており、この製品を利用して数多くのフリーのiアプリが開発されています。また、i-JADEはBorland JBuilderなどの開発ツールから呼び出すことができるように設計されているため、ふだん使用している開発ツールを使用しながら、i-JADE Liteで作成したiアプリのエミュレーションを行うことができますようになっていまして。この連載ではi-JADE LiteとBorland JBuilder 5 Personalを利用してiアプリの開発を行うことにします。

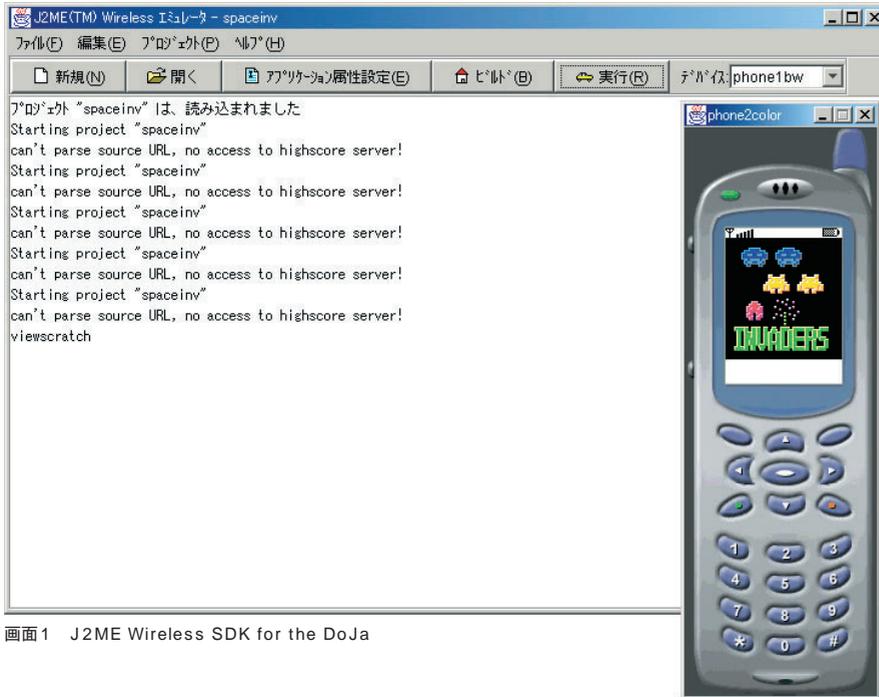


それではiアプリを作成する環境を作

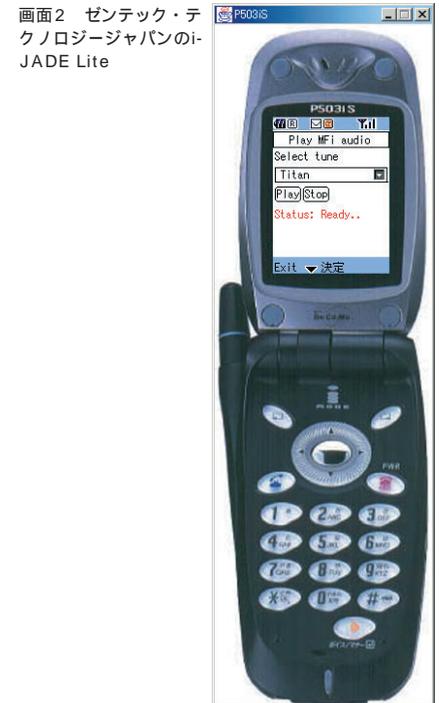
成していきましょう。まず、開発ツールであるBorland JBuilder 5 Personal (以下、JBuilder)をインストールします。JBuilderは表6のWebサイトから無償でダウンロードすることができます。また、製品を使用する前にユーザー登録を行い、インストール番号とインストールキーを入手しておく必要があります。

JBuilderは、本体、ドキュメント、サンプルプログラムそれぞれにインストーラが付属しています。JBuilderをダウンロードしてtarボールを展開し、それぞれのインストーラを実行してください(画面3)。ここでは/tmpディレクトリにダウンロードされたファイルが納められているものとして説明しています。

インストールプログラムを起動する



画面1 J2ME Wireless SDK for the DoJa



画面2 ゼンテック・テクノロジージャパンのi-JADE Lite

Javaコンパイラ、JARアーカイバ (J2SE Version 1.3.1 Linux Versionに含まれる)	http://java.sun.com/j2se/1.3/ja/download-linux.html
CLDC (Sun Community Siteへの登録が必要)	http://www.sun.com/software/communitysource/j2me/cldc/download.html
J2ME Wireless SDK for the DoJaのダウンロード	http://www.nttdocomo.co.jp/i/java/tool.html
Forte for Java Community Editionのダウンロード(Sun Community Siteへの登録が必要)	http://www.sun.co.jp/forte/ffj/index.html
i-JADE Liteのダウンロードサイト	http://www.zentek.com/i-JADE/ja/index.html

表5 iアプリ開発ツール関連のWebサイト

と画面4のようなGUIのインストールプログラムが起動されます。このインストールプログラムの手順に従っていくことで、JBuilderのインストールを行うことができます。

GUI上から起動できるように、GNOMEのパネルに登録する場合は次のように行います。画面5のようにパネルの上で右クリックしてスピードメニューを表示し、[パネル] - [パネルに追加] - [ランチャ]を選択し、[ランチャアプレットを作成する]のダイアログボックスを表示します。次に、このダイアログボックスに画面6のよう

Borland JBuilder 5 Personalのダウンロード	http://www.borland.co.jp/jbuilder/jb5/download/linux.html
Borland JBuilder 5 Personalのユーザー登録	http://www.borland.co.jp/jbuilder/webreg/jb50per.html
J2SEのダウンロード	http://java.sun.com/j2se/1.3/ja/download-linux.html

表6 Borland JBuilder 5 Personalおよび、J2SEの入手先

に入力し、JBuilderの起動スクリプトを設定します。アイコンは/opt/jbuilder5/のディレクトリに納められているイメージファイルを設定します。

インストールが終了したら、JBuilderを起動してみましょう。最初の起動時は画面7のような登録画面が表示されるので、インストール番号とインストールキーの入力を行います。あらかじめボーランドのWebサイトで

ユーザー登録を行い、入手したインストール番号とインストールキーをここに入力します。

画面8はJBuilderのIDE画面です。このあとで、i-JADE Liteの設定を行いますので、JBuilderは終了させずにこのままにしておいてください。

Red Hat Linux 7.1へのインストールJBuilderがサポートしているディストリビューションはRed Hat Linux 6.2のため、Red Hat Linux 7.1環境へインストールする場合は注意しなければなりません。Red Hat Linux 7.1環境では、JBuilderに付属しているJDKが動きません。しかし、事前の設定と別途JDKをインストールすれば動作させることが可能です。ここでは、JBuilderをRed Hat Linux 7.1環境で使用するためのインストール方法を紹介します。なお、LASER 5 Linux 7.1版でも確認したところ、同様の設定とインストールが必要になります。

Red Hat Linux 7.1 Disc1のCD-ROMをCD-ROMドライブに入れ、次の2つのRPMパッケージをインストールします。

```
# cd /tmp
# tar zxvf jb5_linux.tar.gz
# /tmp/per_install.bin
# tar zxvf jb5docs.tar.gz
# /tmp/doc_install.bin
# tar zxvf jb5samples.tar.gz
# /tmp/smp_install.bin
```

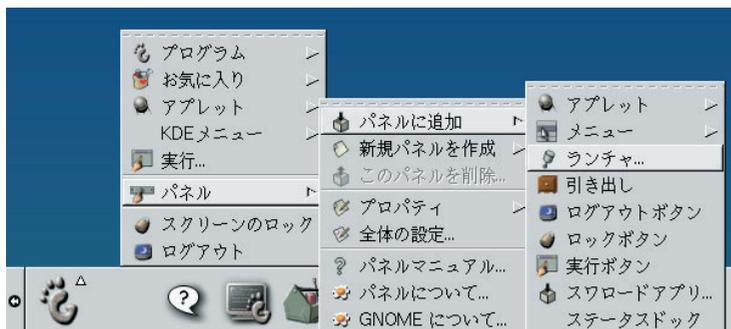
画面3 Borland JBuilder 5 Personalのインストール



画面4 JBuilderのインストールプログラム



画面6 ランチャの設定



画面5 GNOMEランチャの追加



画面7 JBuilderライセンス登録画面

携帯電話 de Javaプログラミング

```
compat-glibc-6.2-2.1.3.2.i386.rpm  
compat-libstdc++-6.2-2.9.0.14.i386.rpm
```

使用しているPCのCPUがPentiumの場合、環境変数を次のように設定します。

```
# export LD_ASSUME_KERNEL=2.2.5
```

常にJavaを使用する場合は、起動スクリプトに追加してしまうといいでしょう。

なお、Red Hat LinuxがインストールされているPCのCPUがPentiumではない場合は、この設定は必要ありません。

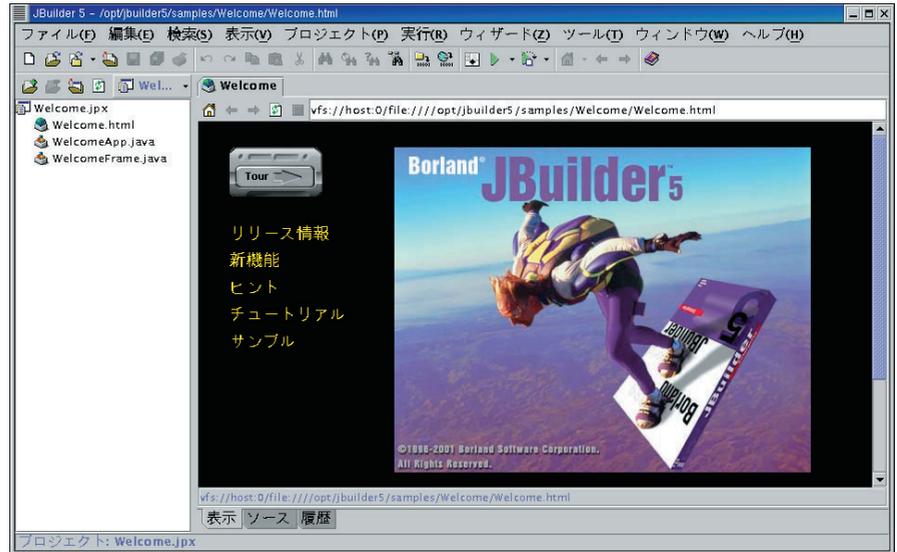
サン・マイクロシステムズのWebサイトからLinux版のJava 2 Standard Edition(J2SE) Version 1.31をダウンロードし、インストールを行います。

RPM形式とtar形式の2種類が用意されていますが、ここではRPM形式を例にとってインストール作業を進めてみましょう(画面9)。

インストールが終了したらJavaコマ

ンドが正しく動作するかどうか確認してみましょう(画面10)。

続いてJBuilderのインストール作業を行います。ここでも/tmpディレクトリにダウンロードされたファイルが納



画面8 JBuilder 5のIDE画面

```
# chmod 744 j2sdk-1_3_1-linux-i386-rpm.bin
# ./j2sdk-1_3_1-linux-i386-rpm.bin

Sun Microsystems, Inc.
Binary Code License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED SUPPLEMENTAL LICENSE
TERMS (COLLECTIVELY "AGREEMENT") CAREFULLY BEFORE OPENING THE SOFTWARE
MEDIA PACKAGE. BY OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE

.....中略.....

Do you agree to the above license terms? [yes or no]
yes
Unpacking...
Checksumming...
0
0
Extracting...
UnZipSFX 5.40 of 28 November 1998, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  inflating: jdk-1.3.1.i386.rpm
Done.

# rpm -ivh jdk-1.3.1.i386.rpm
Preparing...
1:jdk
```

画面9 J2SEの展開とインストール

められているものとして説明していきます。まずは本体のインストールプログラムを画面11のように指定して起動します。本来、JBuilderのインストールプログラムはInstall AnyWhereと呼ばれる100% Pure Javaで書かれたインストーラで作成されており、製品付属のJDKを利用して動作するようになっています。しかし、JBuilderに付属

するJDKがそのままではRed Hat 7.1の環境では使用することができないため、外部のJDKを利用してInstall AnyWhereを起動してインストールを行う必要があります。

続けて、ドキュメントとサンプルプログラムについても本体と同じ手順でインストールを行います（画面12・13）。

Red Hat Linux 7.1およびLASER 5 Linux 7.1の場合は、JBuilderに付属のJDKを使用せずに、先ほどインストールした最新版のJDKを使用します。このため、/opt/jbuilder5/bin/jdk.configの設定をリストのように変更する必要があります。

 i-JADE Liteのインストール

前述したように、i-JADEはiアプリの動作をPC上で確認できるエミュレータです。本来はWindows環境のみが動作環境ですが、製品が100% Pure Javaで書かれていますので、Windows環境にインストールしたあと、インストール後のファイルをLinux環境にコピーすることでLinux環境でも使用することができます。

筆者はWindows 98とLinuxを同一マシンにインストールしていますので、Windows 98で使用している領域をLinux環境でマウントしてファイルをLinuxに移動させることができます（画面14）。別マシンにインストールされた方はFTPなどを利用してファイルを転送するといいでしょう。

コピーが終了したら、i-JADE Liteを起動してみましょう（画面15）。起動方法はWindows環境で使用する場合と一緒です。

i-JADE Liteを起動すると画面16・17のような画面が表示されます。エミュレータが正しく動作しているかサン

リスト /opt/jbuilder5/bin/jdk.configの修正

```
# JDK PATH
#javapath /opt/jbuilder5/jdk1.3/bin/java
javapath /usr/java/jdk1.3.1/bin/java
# tools.jar を追加
addpath /opt/jbuilder5/jdk1.3/lib/tools.jar
vmparam -Xverify:none

.....省略.....
```

コメントアウト

インストールした
JDKのパスを指定

```
# /usr/java/jdk1.3.1/bin/java -version
Java version "1.3.1"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1-b24)
Java HotSpot(TM) Client VM (build 1.3.1-b24, mixed mode)
```

画面10 インストール確認

```
# cd /tmp
# tar zxvf jb5_linux.tar.gz
# /tmp/per_install.bin LAX_VM /usr/java/jdk1.3.1/bin/java
```

画面11 Red Hat Linux 7.1へのBorland JBuilder 5 Personalのインストール（本体）

```
# tar zxvf jb5docs.tar.gz
# /tmp/doc_install.bin LAX_VM /usr/java/jdk1.3.1/bin/java
```

画面12 Red Hat Linux 7.1へのBorland JBuilder 5 Personalのインストール（ドキュメント）

```
# tar zxvf jb5samples.tar.gz
# /tmp/smp_install.bin LAX_VM /usr/java/jdk1.3.1/bin/java
```

画面13 Red Hat Linux 7.1へのBorland JBuilder 5 Personalのインストール（サンプルプログラム）

```
# mkdir /tmp/win98drive
# mount /dev/hdc1 /tmp/win98drive -t vfat
# cp -r /tmp/win98drive/i-jade /usr/local
```

Windowsの領域をマウントさせるディレクトリを作成

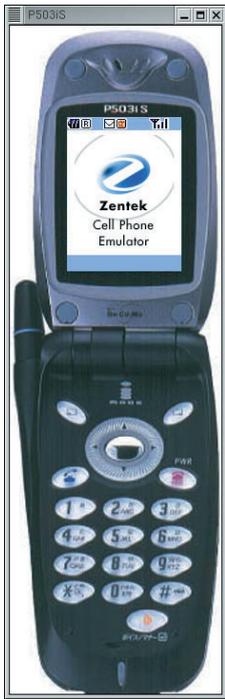
Windowsの領域をマウントさせるディレクトリを作成

i-JADEをLinux環境にコピー

画面14 Windows環境のi-JADE Liteのコピー

```
# /usr/java/jdk1.3.1/bin/java -jar /usr/local/i-jade/i-jade-ps.jar
```

画面15 i-JADE Liteの起動



画面16 i-JADEをLinuxで起動させたところ

ブルプログラムを読み込んで動作を確認してみましょう。[ファイル] - [iアプリのオープン]でファイル選択ダイアログボックスを表示させ、/usr/local/i-jade/samples/uitestディレクトリから「UITest.jam」を選択し、[実行]ボタンをクリックしてください。iアプリがエミュレータにロードされ、実行されたことを確認できると思います。i-JADE Liteを終了させるときは、[停止]ボタンを押してiアプリの実行を終了させたあとで、[終了]ボタンを



画面17 i-JADEのコントロールパネル

押してエミュレータを終了させてください。

エミュレータが動作しましたので、これで開発したiアプリをLinux上で確認することができます。

i-JADE LiteのJBuilderへの登録

i-JADE Liteは単独で動作するだけでなく、JBuilderのIDE画面から呼び出すことができます。それではこちらの設定を行ってみましょう。

[ツール] - [ライブラリ設定]から画面18のようなライブラリ設定ダイアログボックスを呼び出します。このダイアログボックスで、[新規]ボタンをクリックして、新規ライブラリウィ

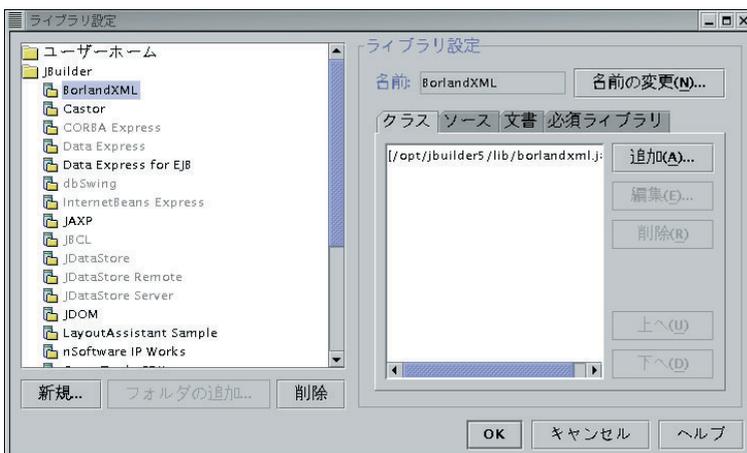
ザードを起動します。名前に「i-JADE」、場所は「ユーザーホーム」とし、追加ボタンをクリックしてi-JADEのライブラリを追加します。ここでは/usr/local/i-jade/i-jade-ps.jarを選択します(画面19)。これでJBuilder内からi-JADEエミュレータを呼び出すことができるようになりました。

実際にJBuilderからi-JADE Liteを呼び出すためにはiアプリを作成してやる必要があります。実行時にi-JADE Liteのメインクラスを指定することによって、JBuilderで作成したiアプリをi-JADE Lite上で実行することができません。実際の使用方法については、次回に簡単なiアプリを作成していく中で説明します。

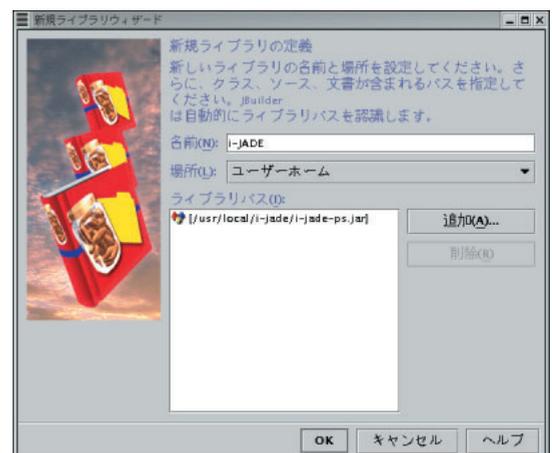
今回、初めてJavaプログラミングを始める方は、関連書籍を一読されることをお勧めします。

関連書籍

- ・『図解そこが知りたい! よくわかるJava』
夏目大著
翔泳社 ISBN4-88135-947-9
- ・『iモードJavaプログラミング スタンドアロン・アプリケーション編 改訂新版』
アスキー書籍編集部編
アスキー ISBN4-7561-3790-3
- ・『iモードJavaプログラミング 拡張APIリファレンス編』
アスキー書籍編集部編
アスキー ISBN4-7561-3803-9



画面18 ライブラリ設定ダイアログボックス



画面19 新規ライブラリウィザード

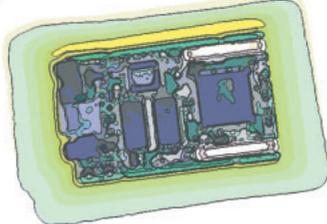
レーザーファイブ L-Card+

カード型

Linuxマシンで遊ぶ

レーザーファイブ L-Card+

<http://www.laser5.co.jp/embedded/lcard/>



文：吉田 功
Text: Isao Yoshida

第2回 開発環境編

L-Card+は、MIPS系のVR4181というマイクロプロセッサを搭載しています。このため、インテルのx86系用のバイナリを動かすことはできません。L-Card+で動くプログラムを作るには、ソースリストをL-Card+のうえでコンパイルするのが一番ですが、コンパイル速度はお世辞にも速いとはいえませんし、大容量のコンパクトフラッシュ(CF)やマイクロドライブを用意しないと扱いにくいものになってしまいます。

そこで、x86系のLinuxで動作するMIPSアーキテクチャ用のクロスコンパ

イラを使用して、L-Card+のプログラムを開発してみましょう。

まずは、プログラム開発環境の整備から

L-Card+に付属しているCD-ROM(写真2)には、クロスコンパイラなどが含まれています。オリジナルは、SGI社のFTPサイトから取ってきたものなのですが、CD-ROMには必要なものがまとまって入っていますので、こちらを使うことにしました。

開発用マシンを準備する

プログラム開発には、それほどマシンパワーを必要としないので、少し古めのパソコンでも構いませんが、ソースリストや開発ツールなどをインストールしなくてはならないので、ハードディスクの容量は大きいほうがよいでしょう。また、CFが扱える環境を一気に書き込めるので便利です。

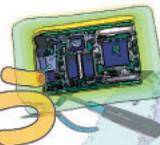
コンパイル、テスト用にx86アーキテクチャ用のCコンパイラをインストールしておくとう便利ですし、gccは使わなくてもmakeは使用するのので、開発関連のコマンドは必要です。筆者は開発専用機に、Turbolinux Server 6.5をインストールして使用していますが、開発関連のソフトもインストールしたかったので、「インストールタイプの選択」は「すべて」を選択しておき、HTTPサーバなどの無用なサーバ系ソフトを外してインストールしました。このとき、ファイル転送に使うFTPサーバは残しておくとうよいでしょう。



写真1 名刺サイズのLinuxマシンL-Card+
今回は、クロスコンパイル環境をインストールして、LEDを制御するプログラミングをする。



写真2 L-Card+に付属するCD-ROM



クロスコンパイル環境の構築

次は、クロスコンパイル環境のインストールです。L-Card+ 付属のCD-ROMには、カーネルをコンパイルできるようにクロスコンパイラなどが入っています。このクロスコンパイラは、環境も含めたコンパイル済みのバイナリと、ソースからコンパイルしてインストールする2種類が入っています。今回は、お手軽なコンパイル済みの環境をインストールしてみます。

CD-ROMから、Cross-VR.tar.bz2を /usr/localの中にコピーします。これを展開すると、/usr/local/Cross-VRというディレクトリができ、中にクロスコンパイラなどの実行ファイルやヘッダファイル、ライブラリなどがコピーされます。

実行ファイルは、/usr/local/Cross-VR/binの中に置かれますので、ここにパスを通しておきます。毎回パスの設定をするのが面倒ならば、ホームディレクトリの中にある、.bash_profileでパスを設定しておくとう便利です。

ヘッダファイルやライブラリは、/usr/local/Cross-VR/mipsel-linuxの中にあります。mipselの最後のelはリ

トルエンディアン用を意味します。VR4181はリトルエンディアン専用になっているので、ビッグエンディアン用は使えません。

また、/usr/local/Cross-VR/mipsel-linux/includeの中にある、asm、linuxの2つのシンボリックリンクは、../src/linux/include/asm (linux)へ相対的にリンクされています。つまり、/usr/local/Cross-VR/src/linux/include/asm (linux)を指していることとなります。ここには、カーネルのソースが入ることが前提になっているようです。

さらに、カーネルをコンパイルするときには、/usr/local/Cross-VR/bin/mipssl-linux/include/asm (linux)にもカーネルのソースへのシンボリックリンクを張る必要があるようです。binの中に、mipssl-linux/includeというディレクトリを作

って、asmとlinuxのリンクを張ります。ソースを、/usr/local/Cross-VR/src/linuxの中にインストールするならば画面1のように設定します。

ちなみに、カーネルのソース内でのasmもシンボリックリンクになっていて、これは、asm-mipsというディレクトリを指します。

カーネルをコンパイルする

クロスコンパイラのファイルからカーネルのソースリストをリンクしていますので、カーネルもインストールしておきます。ついでに、コンパイラのテストを兼ねてコンパイルをしてみましよう。

L-Card+のシステムは、フラッシュメモリ (FROM) から起動して、カーネルを実行し、CFをマウントできるようになってから、CF上の各種ファイルを参照・実行して動きます。ここでコ

```
# mkdir /usr/local/Cross-VR/bin/mipssl-linux/include
# cd /usr/local/Cross-VR/bin/mipssl-linux/include
# ln -s /usr/local/Cross-VR/src/linux/include/asm asm
# ln -s /usr/local/Cross-VR/src/linux/include/linux linux
```

画面1 ディレクトリを作成し、リンクを張る

Column

リトルエンディアンとビッグエンディアン

CPUが、2バイト以上の数値をメモリに書く場合、のように、数値の上位の桁から書き込むか (ビッグエンディアン)、下位の桁から書き込むか (リトルエンディアン) の2通りの方法があります (データを読み出す場合も同様です)。

MIPS系のCPUは、両方を切り替えて使用できるものも多いのですが、VR4181はリトルエンディアンに固定されています。また、x86系のCPUもリトルエンディアンを使用しています。



図 リトルエンディアンとビッグエンディアン

ンパイルして生成できるファイルは、FROMに書き込むためのファイル（カーネル）になります。

カーネルのソースのインストールには、付属CD-ROMの中から最新のタイムスタンプが付いている、linux.20001215a.tar.gzを使います。カーネルのリビジョンは、2.4.0-test9となっています。これを/usr/local/srcで展開すると、linux.20001215aというディレクトリができ、この中にソースファイルが作られます。

以下のようなシンボリックリンクを張って、クロスコンパイラから参照できるようにしておきます。

```
# ln -s linux.20001215a linux
```

カーネルのコンパイルは以下のよう
な手順で行います。

```
# make config
# make dep
```

```
# make
```

make configで、カーネルの設定を行い、最後のmakeでコンパイルが始まります。コンパイルが終了すると、vmlinuxというバイナリファイルができます。このファイルにはデバッグ用のシンボルテーブルが書かれています。mipsel-linux-stripコマンドを使ってシンボルテーブルをカットすることで、ファイルサイズが小さくなります。

```
# mipsel-linux-strip vmlinux
```

さらに、FROMに書き込むには、L-Card + のモニタプログラムを使います。L-Card + をリセットしたり、電源投入時に、RS-232Cの端末から「ス

ペーキー」を押すとモニタモードに入りますので、これを使ってカーネルを書き込みます。このモードでは、ファイルはS-encodeフォーマットになっている必要があるため、画面2のようにしてmipsel-linux-objcopyを使ってバイナリをS-encodeに変換します。

カーネル（FROM）を書き換えるには危険が伴うことを覚えておきましょう。万が一、FROMの書き換えに失敗すると、L-Card + は2度と立ち上がらなくなってしまう可能性があります。こうなってしまうと、フラッシュメモリ（FROM）を基板から外して書き換えるか、オプションの「ROM Board for LC」を使って、FROMを書き直すか、メーカーに修理に出さないと元には戻せません。

```
# mipsel-linux-objcopy -O srec vmlinux vmlinux.srec
--adjust-vma=0x20000000
```

画面2 カーネルのバイナリをS-recordに変換する

Column

クロスコンパイル環境をソースから作成する方法

付属のCD-ROMには、クロスコンパイル環境をソースから作成するためのファイルも含まれています。すでにMIPS用にパッチが当たっているファイルが含まれているので、これを使います。ファイルは、コンパイラなどが入ったegcsと、ツール類が入ったbinutilsのソースと、コンパイル済みのライブラリglibcがあります。

```
egcs-1.1.2.tar.bz2
binutils-19991011.tar.bz2
glibc-mipsel-linux-softfloat-2.0.7-crossdev-bin.tar.gz
```

ソースは、好きなディレクトリに展開しておきます。ライブラリは、開発環境をイン

ストールしたいディレクトリを作成し、そこに展開します。たとえば、ソースは、/usr/local/srcに、ライブラリは、/usr/local/mipsに置いて、それぞれ展開します。

まず、最初は、binutilsのコンパイルです。展開後、そのディレクトリの中に、target-mipsel-linuxというディレクトリを作り、そこに移動します。その中から、以下のようにタイプします。/usr/local/mipsは、開発環境を置きたいディレクトリなので、別の名前にしても構いません。

```
% ../configure
--prefix=/usr/local/mips
--target=mipsel-linux
% make CC=gcc CFLAGS= LDFLAGS=-s
```

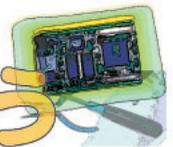
このあと、suコマンドでルート権限を取得し、make installでファイルをインストールし

ます。インストールされるディレクトリは、先ほど、--prefixで指定したディレクトリになります。

次は、egcsのコンパイルです。binutilsと同じようにディレクトリの中に、target-mipsel-linuxを作り、その中で以下のようにタイプします。開発環境ディレクトリの設定も同じです。

```
% env CFLAGS=-O2 ../configure
--prefix=/usr/local/mips --enable-threads --target=mipsel-linux
% make LANGUAGES="c c++ objc f77"
```

インストールも同様に、suコマンドでルート権限を持って、make installで行います。コンパイラなど実行ファイルが入ったディレクトリは、/usr/local/mips/binになりますので、パスを通しておきましょう。



L-Card+のLEDを制御してみよう

これで、クロスコンパイル環境も整いました。いよいよ今月のメインテーマである、制御用のプログラムを作成しましょう。

L-Card+には、シリアルインターフェイスやイーサネットは付いていますが、それ以外の入出力の手段は、5つのLED(写真3)を除けば、拡張バスコネクタに拡張基板を取り付けて、内部の信号を外部に引き出さないと使えません。そこで、手始めに、基板単体で手軽に実験できる、LEDを制御するプログラムを作ってみることにします。

ここで紹介するプログラムは、レーザーファイブのL-Card+製品紹介のWebページに参考資料として掲載されている「L-Card+のLED3~6、LED7の制御プログラム(長岡あつしさん提供)」を解説しやすいようにリメイクしたものです。こちらのプログラムも合わせてご覧いただければ、より理解が深まると思います(<http://www.laser5.co.jp/embedded/lcard/support.html>)。



写真3 プログラムで制御できる5つのLED

LED番号	GPIO端子番号
3	29
4	31
5	8
6	2

表1 LEDとGPIO端子の関係

VR4181の汎用I/Oについて
L-Card+でのプログラミングの前に、CPUのVR4181とLEDの関係を見てください。LEDは、VR4181の機能のひとつである汎用I/Oユニット(GIU)を使ってON/OFFできるよう

になっています。GIUを使ってLEDを制御するには、LEDが接続された汎用I/O端子(GPIO端子)が出力に設定されている必要があります(図1)。

GPIO端子は全部で32個ありますが、汎用I/O機能だけでなく、RS-232Cや

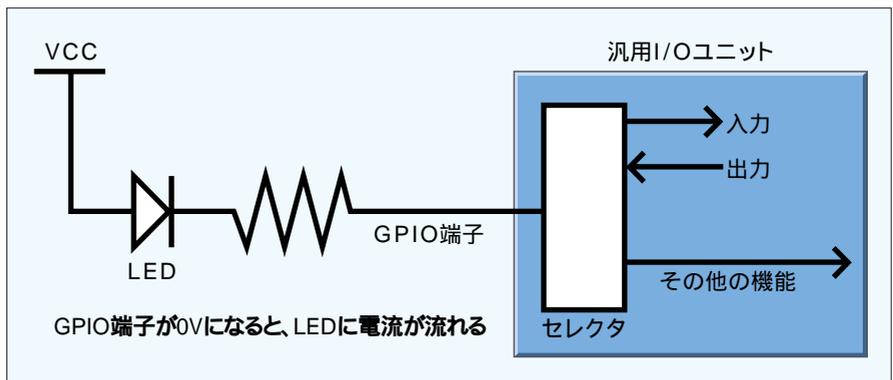


図1 LEDと汎用I/Oユニットの関係

GPIO端子番号	内部使用 / 拡張バス	機能
0	拡張	GPIO0
1	拡張	GPIO1
2	拡張	GPIO2 / LED6
3	拡張	PCS0
4	拡張	GPIO4
5	拡張	GPIO5
6	拡張	GPIO6
7	拡張	GPIO7 (Reserved)
8	拡張	GPIO8 / LED5
9	拡張	#CD2
10	拡張	#CD1
11	拡張	CLK48 / PCS1 / GPIO11
12	拡張	LCD-FPD4 / GPIO12
13	拡張	LCD-FPD5 / GPIO13
14	拡張	LCD-FPD6 / GPIO14
15	拡張	LCD-FPD7 / GPIO15
16	拡張	#IORD
17	拡張	#IOWR
18	拡張	IORDY
19	拡張	#IOCS16 / GPIO19
20	拡張	#UBE
21	拡張	#RESET
22	内部	ROMCS0#
23	内部	ROMCS1#
24	内部	ROMCS2#
25	内部	RS232C-RXD1
26	内部	RS232C-TXD1
27	内部	RS232C-RTS1
28	内部	RS232C-CTS1
29	内部	LED3
30	内部	(Reserved)
31	内部	LED4

表2 L-Card+でのGPIO端子の使用状況

LCDコントロールにも使えるようになっていて、入力か出力を設定するだけでなく、ほかの機能に割り当てるかどうかを設定しなくてはなりません。

GPIO端子で制御できるLEDは4個あります。割り当てられているGPIO番号は表1のようになっています。参考として、汎用入出力として使う場合の設定と、32個のGPIO端子のうち、L-Card + 内部で使っているもの、拡張バスに信号が出ているものを表2にまとめておきました。

VR4181のメモリアクセスについて
x86系CPUは、メモリ空間とI/O空

間が分かれています。VR4181は同じアドレス空間に割り当てられています。したがって、メモリをアクセスするようにI/O空間にもアクセスできます(図2)。

また、データバスは16ビット幅になっています。DRAMやROMも16ビット幅ですが、I/Oは、8ビットまたは16ビット幅が使われます。GPIOへのアクセスは16ビットになっています。32ビット分あるGPIOは2度に分けてアクセスすることになります(表3)。

1個のLEDをON / OFFさせるには、16ビットの中の1ビットだけ进行操作します。特定の1ビットだけ、0にしたり、

1にする必要があるのですが、GPIOへのアクセスは16ビット単位でしか行えないので、GPIOを書き換える前に、一度読み出して、特定のビットだけ書き換えたあと、16ビット分のデータを一度に書き換えるという手順が必要です(図3)。

I/Oプログラミングの方法

UNIXなど、マルチタスクOSでは、I/O関連のアクセスは慎重に行わなければならない。たとえば、複数のタスクから1台のプリンタを使って印刷させようとしたとき、あるタスクでプリンタが使用中なら、OSは印刷が終わる



図2 VR4181のメモリ空間

デバイス	データ幅
ROM、フラッシュメモリ	16ビットのみ
DRAM	16ビットのみ
コンパクトフラッシュ	8ビットまたは16ビット
外部I/O	8ビットまたは16ビット
外部メモリ	8ビットまたは16ビット

表3 システムバスで対応可能なデバイスのデータバスの幅

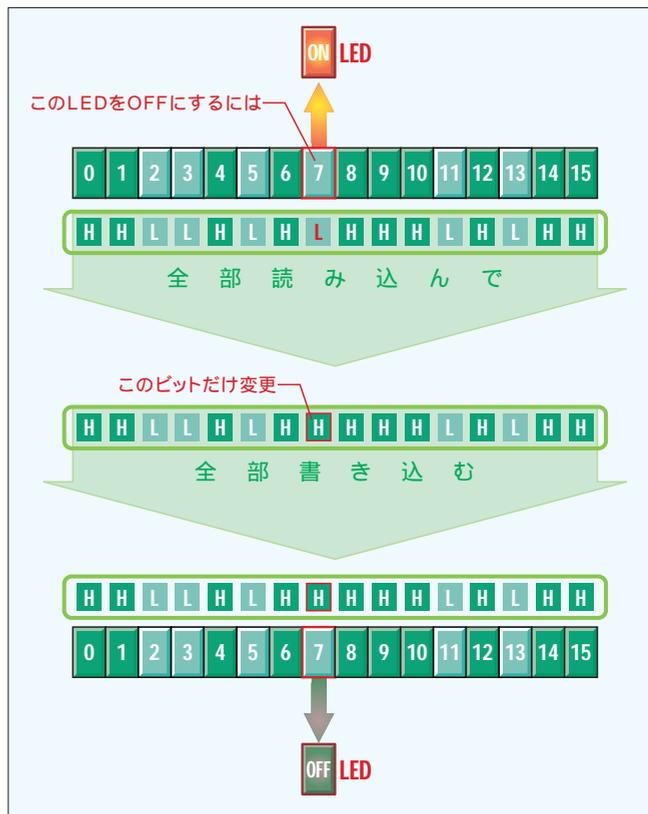
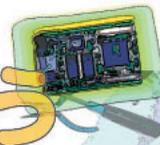


図3 1ビットだけデータを書き替える時も、1ワード全部書き換える



までほかのタスクからのリクエストを抑制しなくてはなりません。もちろん、Linuxで、LEDをON / OFFするだけの簡単な制御をする場合も同じことが発生します。簡単なI/O操作でも、タスク間の調停を行えるように、デバイスドライバとしてプログラミングするのが礼儀正しいプログラムの方法ということになります。

ただし、LEDを制御するプログラムを1つだけしか走らせないことを前提にするなら、ユーザープログラムの中で勝手にI/O操作してしまっても、操作の衝突は起こりません。制御用のコンピュータでは、複数の人間(タスク)から使用されることがないなら、問題はありません。そして、root権限のあるプログラムなら、メモリへのアクセスも可能です。そこで、LED制御のプログラムも、簡単にプログラミングできる、ユーザープログラムとして作成することにしました。

プログラムのしくみ

リスト1は、LED制御用のプログラムです。このプログラムは、L-Card+の4つあるLEDのうち、LED3とLED4をONに、LED5とLED6をOFFにして終了します(写真4)。

まず、C部でメモリデバイスをオープンします。オープンできなかつたら、エラーを表示して終了します。

次に、D部でVR4181の内部I/O空間をマッピングします。アドレスは、



写真4 作成したプログラムでLEDをON / OFFしてみる

リスト1 LED制御プログラム(その1)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/mman.h>
#include <sys/stat.h>
#include <asm/page.h>
#include <asm/types.h>

/* GPIOレジスタアドレス定義(A) */
#define GPMD0REG 0x300
#define GPMD1REG 0x302
#define GPMD2REG 0x304
#define GPMD3REG 0x306
#define GPDATHREG 0x308
#define GPDATLREG 0x30A

unsigned char *mapadrs ; /* メモリアクセス用ポインタ */

/* LED操作サブルーチン(B) */
void led6off(void)
{
    *(short *) (mapadrs + GPDATLREG) |= 0x0004 ;
}
void led6on(void)
{
    *(short *) (mapadrs + GPDATLREG) &= 0xFFFF ;
}
void led5off(void)
{
    *(short *) (mapadrs + GPDATLREG) |= 0x0100 ;
}
void led5on(void)
{
    *(short *) (mapadrs + GPDATLREG) &= 0xFEFF ;
}
void led4off(void)
{
    *(short *) (mapadrs + GPDATHREG) |= 0x8000 ;
}
void led4on(void)
{
    *(short *) (mapadrs + GPDATHREG) &= 0x7FFF ;
}
void led3off(void)
{
    *(short *) (mapadrs + GPDATHREG) |= 0x2000 ;
}
void led3on(void)
{
    *(short *) (mapadrs + GPDATHREG) &= 0xDFFF ;
}

int main()
```

A

B

0x0B000000から、0x0BFFFFFFまでに
なります。ポインタ (mapadr) にマッ
ピングしたアドレスを入れ、それに、
GPIOのオフセットアドレスを加えて、
GPIOをアクセスできるようにします。

E部では、LEDで使用するGPIO端
子の初期化を行っています。汎用出力

設定は「10 (2進)」に設定する必要が
あるので、AND演算子で「00」にし
てから、OR演算子で「10」に設定し
ています。

B部が具体的にLEDをON / OFFし
ている部分です。似たようなサブルー
チンが8個ありますが、4つのLEDのそ

れぞれをON / OFFする関数です。結
果を書き込む前に、そのポートのデー
タを読み出し、ビットをクリア (0) す
るときはAND演算子を使い、セット
(1) するときにはOR演算子を使って読
みとった値を変更します。一度変更す
れば、上書きするまでLEDは同じ状態
を保ちます (図4)。

G部、最後は、マッピングしたメモ
リマップとメモリデバイスを解放して
終わります。

コンパイルからコマンド実行までの手順
先ほどインストールしたクロスコン
パイラでリスト1をコンパイルしてみま
しょう。

led.cを入れるディレクトリはどこで
も構わないのですが、今回は、
/usr/srcの中に作りました。クロスコン
パイラのあるディレクトリ
(/usr/local/Cross-VR/bin) にパスが
通っていれば、以下のようにタイプし
てコンパイルします。

```
# mipsel-linux-gcc led.c -o led
```

実行ファイル (led) は、ソースと同
じディレクトリ内にできます。もちろ
ん、この実行ファイルは、MIPS用な
ので、コンパイルしたPCでは実行でき
ません。

開発マシンからL-Card +へ転送する
最も手軽な方法は、FTPを使うことで
しょう。L-Card +には、はじめから
FTPクライアントがインストールされ
ていますが、サーバは入っていません
ので、PC側でFTPサーバを動作させ
ておき、L-Card +側から実行ファイル
を取りに行きます。

L-Card +には、シリアルでログイン
するか、イーサネットを介してtelnet
で接続することになるので、操作は全

リスト1 LED制御プログラム (その1)

```
{
int fd;
unsigned int st, len;
char *dev="/dev/mem";

/* メモリデバイスをオープン(C) */
fd=open(dev, O_RDWR);
if(fd<0)
{
    fprintf(stderr, "cannot open %s\n", dev);
    exit;
}

/* メモリのマッピング(D) */
st=0x0B000000;
len=0x1000;
mapadr=mmap(0, len, PROT_WRITE, MAP_SHARED, fd, st);
if(mapadr == MAP_FAILED)
{
    fprintf(stderr, "cannot mmap\n");
    exit;
}

/* GPIOモード設定(E) */
*(short *) (mapadr + GPMD0REG) &= 0xCFFF; /*LED6汎用I/O設定0*/
*(short *) (mapadr + GPMD0REG) |= 0x4000; /*LED6汎用I/O設定1*/
*(short *) (mapadr + GPMD1REG) &= 0xFFFC; /*LED5汎用I/O設定0*/
*(short *) (mapadr + GPMD1REG) |= 0x0002; /*LED5汎用I/O設定1*/
*(short *) (mapadr + GPMD3REG) &= 0x33FF; /*LED3 LED4汎用I/O設定0*/
*(short *) (mapadr + GPMD3REG) |= 0x8800; /*LED3 LED4汎用I/O設定1*/
*(short *) (mapadr + GPDATLREG) |= 0x0140; /*LED5 LED6出力設定*/
*(short *) (mapadr + GPDATHREG) |= 0xA000; /*LED3 LED4出力設定*/

/* LED ON/OFF指定(F) */
led6off();
led5off();
led4on();
led3on();

/* 終了処理(G) */
munmap(mapadr, len);
close(fd);
exit;
}
```

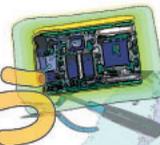
C

D

E

F

G



部PC側から行うことも可能です。転送が済んだら、ファイルのモードを実行可能に変更しておきましょう。

```
% chmod u+x led
```

大量のファイルを一気に転送する場合などは、CFをPCにマウントして、ファイルをコピーしたほうがよいかもしれませんが、デバッグ時に、1~2つのファイルを転送するなら、FTPによる方法が手軽です。

LEDのON / OFFだけでも役に立つ

LEDをON / OFFさせるだけでは、ただの実験にしかならないように思えますが、L-Card+ は単体では、ディスプレイなどの出力デバイスを持ちません。サーバ用途などでも、メールが来ているとか、データが溜まっているとか、LEDで外部に知らせることができます。LED点灯プログラムを作っておき、スクリプトから呼び出して、何かを知らせるといった簡単なことでも役立ちそうです。

筆者は、プロバイダのメールサーバにある未読のメールの中から、重要なものだけを厳選し、自宅に不在で、かつ起きている時間帯だけ、携帯電話に転送するシステムをL-Card+ で動かそうと思っています。不在の時にパソコンを動かさなければなしにするのも気が引けますし、ルータやハブより消費電力が少ないL-Card+ は、こんな用途にはピッタリです。それに、CFにアクセスしていないときは、電源をいきなり切っても大丈夫なもの便利です。

さて、本題に戻りましょう。次回は、いよいよ、L-Card+ にセンサーなどを取り付けて、制御や情報収集など、少し実用的なシステムを作りたいと思います。

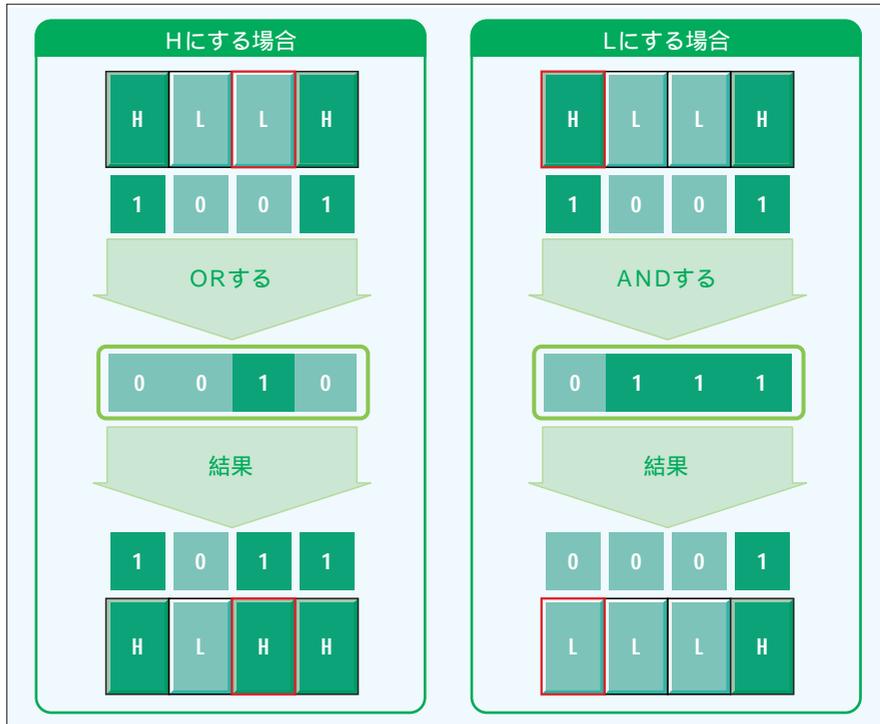


図4 1ビットだけデータを書き替える手順

Column

もうひとつのLED制御方法

CPUのVR4181には、LEDコントロールユニットというものが付いています。L-Card+ では、LED7がそのユニットに割り当てられています。

LEDコントロールユニットは、LEDをON / OFFするというより、ハード的に点滅させるユニットです。点灯時間、消灯時間を別々に設定でき、これを繰り返します。

さらに、オートストップ点滅回数を設定することで、自動的に点滅を停止させることができます。このとき、割り込みを発生してソフトウェアに知らせるとい、タイム的な機能まで持っています。

点灯時間は「LEDHTSREG」、消灯時間は「LEDLTSREG」、点滅や点滅オートストップ設定は「LEDCNTREG」、オートストップ点滅回数は「LEDASTCREG」を使います。リスト1のA部とF部を、リストに入れ替え、B部とE部を消して実験してください。

リスト LED制御プログラム(その2)

A部と入れ替え

```
/* LEDコントロール レジスタアドレス定義 */#define LEDHTSREG 0x240
#define LEDLTSREG 0x242
#define LEDCNTREG 0x248
#define LEDASTCREG 0x24A
#define LEDINTREG 0x24C
```

F部と入れ替え

```
/* LED コントロールユニット設定 */
*(short *) (mapadr + LEDCNTREG) = 0x0000; /*一時停止*/
*(short *) (mapadr + LEDHTSREG) = 0x0008; /*0.5秒点灯*/
*(short *) (mapadr + LEDLTSREG) = 0x0008; /*0.5秒消灯*/
*(short *) (mapadr + LEDCNTREG) = 0x0001; /*点滅開始*/
```

箱の中の ペンギン たち

文：みわよしこ

Text : Yoshiko Miwa miwachan@pp.ij4u.or.jp



第9回 増えてきた組み込み Linux ツールベンダー

組み込みシステムの分野でのLinuxの普及に伴い、組み込みLinuxツールのディストリビューターやベンダーも増加しつつあります。この連載でも、第2回で「Zeta Linux」とその開発者である生越昌巳氏、第3回で「Si-Linux」の開発者である海老原祐太郎氏を紹介しましたが、もちろんほかにも多数の組み込みLinuxディストリビューターやベンダーが存在します。

これらのディストリビューターやベンダーが作成した数多くのディストリビューションやツールは現在、それぞれの特徴を発達させながら「適材適所」への棲み分けを行おうとしている段階にあるように筆者には見受けられます。

今回は、組み込みLinuxディストリビューション「TASTE」を供給しているタクシア（TUXIA。http://www.tuxia.com/）の例を通して、特徴あるディストリビューションが開発されてくる過程を見てみましょう。



ドイツのITの発展とともに育ってきた
「TUXIA」

タクシアの本拠地はドイツにあります。その前身は1989年、ネットワークコンピューティングの会社として設立された「IGEL」です。

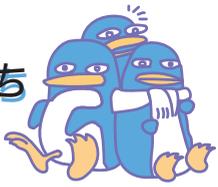
ドイツではインターネット以前から電子商取引などの仕組みが発達しており、オフィスでのコンピュータネットワーク

利用が一般的になっていました。さらにWebの発展を起爆剤として、Webブラウザが使える端末が強く求められるようになりました。このあたり、1995年に「Windows 95」とともにパソコンが一般家庭に普及し、その後、パソコンでのインターネット利用が企業・家庭ともに一般的になっていった日本とはかなりの相違があります。

このような状況の中で、IGELは早くからインターネットに特化した機器を開発・販売するビジネスに注力し、1993年には、まだカーネルバージョンが0.9.6だったLinuxを組み込んだX端末を開発・販売しました。このX端末は、おそらく組み込みLinux製品としては最古であろうと見られています。ただしこの時点では、IGELはLinuxを採用していることを前面に出さず、あくまで「UNIXのX端末」として製品を販売しています。

IGELはこの後、ドイツでは主にネットワークコンピュータのベンダーとして、米国では主にシン（Thin）・クライアントのベンダーとして発展してゆきました。

この間、三度にわたる社名変更・買収等を経て、2000年からは「TUXIA」となり、Linuxに特化したビジネスを進めるようになりました。さらに2000年末からはオープンソース環境でのツール開発プロジェクト（http://www.tuxia.org/）のスポンサーにもなっています。ちなみにタクシア株式会社は同社の日本法人です。



TUXIAのLinuxの特徴

TUXIAはソフト・ハードともに独自のLinux製品を供給してきましたが（現在はソフトウェアのみを事業としています）ソフトウェアに関してはいわゆる「Linuxディストリビューション」と一線を画したツールの供給を行っています。

TUXIAが供給しているツールは、「TUXIA Appliance Synthesis Technology（TASTE）」と呼ばれる独自のソフトウェアのセットです。通常のLinuxディストリビューションと異なる点はアーキテクチャです。

TASTEでは、グラフィック画面を表示するためにオープンソースのプロジェクトで開発された「NanoX」という小さなXサーバが利用可能です。「NanoX」上では「Doom」などのグラフィックスゲームも楽しむことができます。

さらにアプリケーションのレベルでは、「Mozilla」を改造した「Nanozilla」という、サイズの小さなWebブラウザが利用できます。もちろん通常のXサーバやWebブラウザも利用できるのですが、これらは組み込み用途に特化し、プログラムのサイズや必要とするリソースを抑えて作られています。機能的には通常のXサーバや「Mozilla」を組み込み用途に適するように凝縮したものといえます。たとえば、「Nanozilla」では「Mozilla」のプラグインがそのまま利用でき、MP3形式の音楽ファイルを再生することができます。

さらにTASTEは、ファイルシステムのレベルにも特徴を有しています。「Compression Block Layer」と呼ばれる機能を利用して、ブロックデバイスを本来のサイズの1/3まで圧縮することのできるファイルシステムを採用しています。アプリケーションからはブロックデバイスが圧縮されていることをま

ったく意識せず、透過的にファイルシステムを利用することができます。「不要なコマンドやライブラリを削る」といった通常のアプローチと、この「Compression Block Layer」を組み合わせることにより、組み込み用途で多く求められる「サイズの縮小」を果たしているのがTASTEの特徴です。

これらの特徴により、TASTEは「X Window Systemとブラウザのある組み込みLinuxシステム」を開発するのに非常に適したLinuxシステムとなっています。競合相手は他の組み込みLinuxディストリビューションというより、むしろ「Windows CE」といったところです。

ブラウザにこだわる理由

アプリケーションのひとつに過ぎないWebブラウザにTUXIAがこだわるのは、ネットワークコンピュータやシン・クライアントのベンダーとして同社が発展してきた歴史と大いに関係があります。

たとえば、電子商取引ではサーバのリソースは取引のデータベース処理に大きく費やされます。しかしネットワークコンピュータやシン・クライアントは、もともと大きなストレージは持たず、必要以上の機能を搭載せず、なるべくサーバのリソースを利用するという方針で作られます。サーバとクライアントのどちらにどのような機能を実装するかのバランスは難しい問題ですが、電子商取引の場合、グラフィック画面の処理という負荷の高い処理はサーバではなくクライアント側で行い、ネットワークを流れるのは商取引のデータだけという状態にしておくのが最も望ましい姿でしょう。

そうすると、ネットワークコンピュータやシン・クライアントといえども、グラフィック画面の処理・表示は「自前」で行い、サーバに必要な以上の負荷をかけないことが望まれます。このようなことから、ネットワークコンピュータを中心に発展してきたTUXIAは、Webブラウザに強くこだわることを要請されてきたといえます。

ユーザービリティを追求したツール群

TASTEには、小さな処理能力とストレージしか持たないクライアントや、PDAなどでのユーザービリティを追求したツールが数多く含まれています。その中には、たとえばWindowsサーバ上の「Excel」「Word」を利用するための「RWin」、PDFファイルを読むための「xpdf」、Javaアプレットを実行するためのインターフェイス「OJI」といったも



写真1 TSEの開発環境と、開発したシステムをストレージにコピーする機能を一体化した「TSE Toaster」

のがあります。「クライアントなんだから」「PDAなんだから」機能が少なくても仕方がない、という割り切りもひとつの方向性ではありますが、このようなツールにはハードウェアの制約の中で、可能な範囲でユーザービリティを向上させるという方向性が感じられます。

リアルタイム性を追求しない方向性

TASTEは、小さなストレージしか持たない組み込みシステムで、「X Window Systemのある組み込みLinuxシステム」を実現するには非常に適した環境ですが、リアルタイム性はなく、ごく通常のLinuxと同等です。産業用途で利用されることの多い組み込みOSの世界では、若干異彩を放っているといえます。

これは、TUXIAが「ネットワークコンピュータの会社」として成長してきたところに由来しています。産業用途での機器の制御を行う場合とは異なり、ネットワークコンピュータにおいてはそれほど高いリアルタイム性は要求されません。

組み込みシステムは、あくまで個別対応が要求される世界です。目的とする用途によっては、組み込みシステムといえどもリアルタイム性を追求する必要がないのです。したがって、リアルタイム性を要求されない用途に特化した開発環境を追求すると、このような「解」もありうるということになります。

このような方向性には、人員による制約といった面もあります。TUXIAの全社員は世界全体で120人程度、そのうち開発者は50人程度です。この人数で、ありとあらゆる用途を想定した開発環境の開発を行うことは不可能です。ネットワークコンピュータに特化した開発環境という方向性は、人



写真2 TUXIAのインターネット端末「WebPAD」。X Window Systemが起動している。

員の制約から必然的に導かれるものでもあります。逆に言えば、開発人員が少なければ、「得意分野」に特化するのが戦略として有利ということでもあります。

組み込みシステムの専門家でも開発ができる環境づくり

TUXIAでは、「TUXIA Synthesis Environment (TSE)」という開発環境も提供しています。TSEが通常のプログラム環境・デバッグ環境と異なる点は、「組み込みシステムに詳しくない技術者でも開発が行える」ということです。

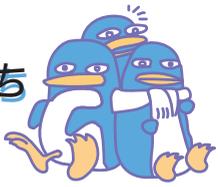
通常の組み込みシステムではストレージが小さいという制約から、搭載するコマンドやプログラム、ライブラリ類を最小限に抑えますが、この際に厄介なのはいわゆる「依存関係」、つまり「あるプログラムの動作に、どのプログラム、ライブラリが必要か」ということのチェックです。これはかなり面倒な作業ですし、慣れていなければ間違いも犯します。

TSEはこの問題に対して、ユニークなアプローチを採っています。GUI画面上で組み込みたいコマンドやプログラムを選択すると、それらの動作に必要なものがすべて自動的に選択され、合計サイズが計算されます。「これでOK」という状態になれば、製品のストレージに焼き込むためのイメージを作成することもできます。あとはそのイメージをストレージにコピーすれば組み込み作業は終了です。

このようなツールがあれば、組み込みLinuxシステムに詳しくない技術者でも、少ないストレージに必要な機能を盛り込むことが容易にできます。日本法人ではTSEを冗談めかして「TUXIA三間組み込みLinux」と呼んでいるそうです。ただし動作環境が「bash2」であり、「bash2」が問題なく利用できる「SuSE」以外ではまだ動作確認ができていないことから普及にはしばらく時間がかかりそうです。

筆者はTSEのデモを見て、「これは一般的な組み込みシステムの専門家からは出てこない発想だ」と感じました。専門家は往々にして自分のスキルによって問題を解決することを重視しますし、またスキルに欠ける技術者を軽蔑しがちです。もちろん専門家が専門家である以上、スキルアップや勉強は不可欠ですが、一方で「スキルに欠ける人でもこの作業ができるようになるためには？」という発想は、やはり技術の熟成や普及のために不可欠といえるでしょう。

あくまで個別対応が求められる組み込みシステムの世界では、開発ツールもまたそれぞれに個性をもった発展を遂げるようです。この連載では今後、組み込みLinuxシステム製品だけではなく、開発ツールも積極的に取り上げてゆく予定です。



タクシア代表取締役 関信彦氏に聞く

Interview



はじめまして。御社の製品を拝見していて感じるのは、発想がエンジニア的ではないな.....ということなのですが。

関：そうかもしれませんね。私はもともと、文化人類学出身なんです。システム理論、それも特に社会システムや情報伝達の研究をしてたんです。今の言葉で言うと、情報環境学みたいなもんですね。昔は周りから「文化人類サラリーマン」とか呼ばれてました（笑）。

この業界では珍しいですね。

関：ええ。もともとはバリ島でネットワーク型社会の研究をされていて。1980年代前半のことなのですが。

バリ島とネットワーク？ どういったことだったのでしょうか？

関：社会のサンプルやプロトタイプとして、狭い場所のほうがわかりやすいということがあるんです。それで、文化人類学の研究者は好んでバリ島で研究を行っていましたね。文化人類学者に限らず、先端的な研究者はバリ島みたいなところを好



写真 関信彦氏（せきのぶひこ）

1960年生まれ。早稲田大学卒業。在学中、'83、'84年にバリ島マルヘンドラダッタ大学へ社会政治学研究のため留学。当時よりネットワーク社会の出現を予見・提唱。某自動車メーカーのモータースポーツ部門から国内エンジニアリング企業での情報・エネルギー分野での開発のマネージメントを経て、'96年よりドイツのインターネット技術を日本市場で展開。'99年よりLinuxベースの組み込みOSの事業化のためにインフォマテックジャパン株式会社を設立。2001年3月よりタクシア株式会社に社名変更、現在代表取締役。

ネットワークコンピューティングを基盤としたユニークな製品を多数開発・販売しているTUXIAでは、コンピュータの今とこれからをどのように考えているのでしょうか。今回は、TUXIAの日本法人の代表取締役である関信彦氏にお話を伺いました。

んでフィールドワークの場所として選ぶ傾向があるんです。

そこでのネットワーク型社会と言いますと？

関：当時（1980年代前半）私は貨幣経済が変化していくのと同時に社会そのものがネットワーク型社会に移行するんだと言ってたんですが、当然ですけど誰も信じてくれなかった（笑）。もちろん、まだネットワークの実体としては何もなかったわけです。インターネットも実験段階でしたし。そこで、ヒューマンネットワークをベースとしたプロジェクトをやっていたんです。

そういうことなのですね。そのヒューマンネットワークへのご関心から、ネットワークコンピューティングの世界に入られたのでしょうか？

関：関心が移ったというより、私自身の中にはもともとデジタルとかアナログとかの区別はないんです。ただ、ネットワークにはそれなりの特性と言うか、自律的な文法みたいなものがあって「この部分をきちんとしないといけない」といった思いはあります。そういう意味では、ネットワークやネットワークの中で起こることは私にとって大きな環境問題なんですよ。

その後はインターネットの発展とともに歩んでこられたわけですね。

関：ええ。'93年から'94年ごろ、日本でもインターネットが利用され始めるようになったところですが、初期のインターネットプロジェクトの立ち上げに関わりました。実現しませんでしたけど、都市博にも関わりましたね。

青島都政で中止された都市博ですね（笑）。こちらにもインターネットを利用した企画があったのでしょうか？

関：今となっては昔話ですが、慶応の藤沢キャンパスの学生と一緒に「バーチャルシティ」という企画をやりました。ヘッドマウントディスプレイを着けたプレイヤーが、インターネットの仮想都市上で別のプレイヤーと出会いながら街を作っていくという、チャットとシムシティを合わせたかなりバブリーなゲーム感覚の展示でした（笑）。ドリームトレイン

インターネット (DTI) を立ち上げた石田君なんかがすごく若いころの話です。

そうしますと、ずっとネットワークを追いかけてこられたという感じですね。ヒューマンネットワークからインターネットへと。

関：そして目指すのは24世紀です。

と言いますと？

関：将来はスタートレックみたいになるだろうと思っているんです。現在、ネットワークが一般的になってはいますけれども、サーバのパワーがフルに使われているとはいえませんしね。

そうですね。マシンパワーが有効に利用されているとはいえない場面はまますね。

関：それに、まだ数の問題としてもコンピュータ自体が普及の途上でしょう。

比較的最近の統計で、アメリカ人の30%くらいが「現在も将来もコンピュータを利用する予定はない」と答えていたのが私は記憶に残っています。

関：1984年にジョブスが「Computers for rest of us」と言ったのですが、その状況は現在もそれほど大きくは変わっていないわけです。

17年も経過して、PCはそれなりに一般社会に普及したと思えますが。

関：現在、Windowsが世界のうち8億人くらいに使われているわけですが、世界人口を62億人とすると、80%くらいの人、つまり48億人くらいの人がかこれから将来にかけてコンピュータを使うようになるんじゃないかなと思います。

PCユーザーは、まだ現在のところ数においてはマイノリ

ティということですね。

関：ええ。今のPCのままではそんなに多くの人ができるシロモノじゃないってことです。そこで私どもは「残りの48億人のためのコンピューティング」というモットーを掲げているんですよ。

ここで話題をちょっと変えさせてください。TUXIAさんはワールドワイドに拠点を持たれ、事業展開されていますが、先ほどお話を伺っていると、たとえば「都市博」など日本独自のお仕事にも関わっていらっしやいますね。

関：ええ。現在ドイツを始めとして8カ国に拠点があるんですが、開発そのものはVPNを通してドイツのCVS上でやっています。

ワールドワイドに協調されているわけですね。

関：ただし、マーケティングに関してはやっとなら日本市場の特殊性というか、特性が認められたので、今後日本向けの新しい戦略を打ち出していく予定です。

「各地域ではその地域に合わせたビジネスを」ということですね。ところで、TUXIAさんの社員数は全世界で120人程度ということで、正直なところ「たったそれだけの人数で……」と感じてしまいます。

関：しかも、そのうち技術者は50人程度です。ですから「あれもこれも……」ではなく、開発ターゲットを絞る必要があります。ただし、今後は中国科学技術大学にも開発センターを置いて、日本向けの開発をしてゆく予定です。

たとえばTASTEでリアルタイム性にこだわらないという方向性も、そういった人数の制約から出てくる面がありますか？

関：そうですね。また開発のスピードも制約されざるをえないです。たとえば、今年はTSEの中国語・日本語対応、WebPAD (インターネット端末) の出荷、PDAへのポータリング、SuperHへの対応といったあたりが目標となっているのですが、もちろん開発人員がもっと多ければ、これをもっと速く行うことができますし、またもっと多くのプロジェクトを進めることができます。

敢えてそれを目指すことはしないわけですね。

関：ええ。私どもはあくまで、「ブラウザのあるシステム」に特化する方向性を取っています。

私は以前、大手の情報通信機器メーカーにいましたので、「何でもあります」「何でもできます」をセールスポイントにする方向性に「洗脳」されてしまっているんですが(笑)。

関：それは日本人のメンタリティでもありますね。日本人は「何でもできる」「どのサーバにもつながる」といったことを好みます。

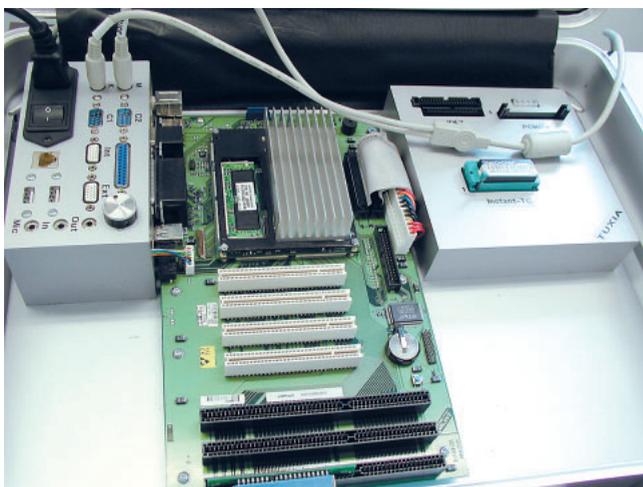
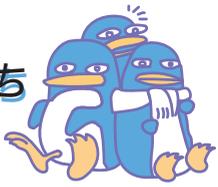


写真3 TSE Toaster内部。3種類のインターフェイスがあり、コンパクトフラッシュなどのFlash ROMイメージをコピーできる。



そして余分な機能が増えるわけですね。

関：ドイツ人は違うんです。削ってゆくことや、特化してゆくことが好きなんです。サーバへの接続に関しても、必要なサーバに接続して業務に必要な通信が行えればよいという感じですよ。

それが形になったのが、TUXIAさんのインターネット・アプライアンスということですか。

関：そうですね。あくまでも現場の方の声を重視しながら、私どもは「ブラウザのあるシステム」に特化してゆきたいと考えています。「インターネット・アプライアンスのため」とびきり最高」を提供するというのが私どもの姿勢です。リアルタイム性など、通常の組み込みLinuxで要求されることはほかの多数の企業さんが追求していらっしゃるから、私どもが何も同じことをすることはないです。

日本法人の方向性にもそれは反映されていますか？

関：もちろんです。たとえば、私どもが「ネットワークコンピューティング」と言いまして、ソニーさんのように製品展開を幅広く行ってゆくことはできないわけです。開発人員の問題もあります。ですから、私どもでは自分たちに適した行き方として、製品というよりはプロジェクトに特化した活動を行っています。

どういったプロジェクトでしょう？

関：NTTさんと一緒に郵政省のTAOのプロジェクトに参加しています。昨年は、学校向けにインターネット環境を構築するというプロジェクトで、サーバと生徒さんの実習用の端末を納入しました。端末側で設定が必要だと、設定に時間を取られて実習ができないということにもなりますので、電源オンですぐ使えて、サーバに接続できて、その上でアプリケーションを動かすという仕組みを作りました。

まさに、学校現場に求められるソリューションですね。

関：このときもブラウザは端末側に載せたんです。ブラウザは重いので、サーバのリソースを食わないようにと考えまして。

納得です。ところで現在の一般的なコンピュータについてはどうお考えになりますか？

関：現在のコンピュータは、火打ち石か、せいぜい木を使った火おこし器の段階ですね。マッチやライターにはなっていない。ですから、WindowsのようなベースのしっかりしていないIOSが世界の90%に利用されてしまうんです（笑）。

かといって、Linuxの一般ユーザーへの普及にはまだ障壁が多数ありますね。

関：ええ。Linuxのユーザーインターフェイスはもっと発展する必要がありますね。でも、OSを作るのは難しい作業で

すよ。MacOS Xが5年かかってまだ未完成でしょう？ 使いやすいIOSを作り込むのは大変な作業ですよ。

まったくそのとおりですね。一ユーザーとして意見を言うのは簡単ですけども。

関：ただ、Linuxはオープンソースであるということに期待が持てます。科学技術は結局、オープンにすることでしか発展しないものですし。先ほど、「目指すのは24世紀、スタートレックの世界」と申し上げましたが、たとえば 星系に宇宙船が出て行ったとしましてね、何かコンピュータに不具合が起こったとしますでしょうか？

あまり想像したくない状況ですね。

関：そこでOSがWindows NTだったとすると、地球と交信して「じゃ、サービスパックを送るから3カ月待ってね」となって、間に合わなくて全員死んでしまう（笑）。だけどオープンソースだったら、自分たちで直せるかもしれないんです。

ソースコードを読める人がいれば、可能性はありますね。

関：そういう面で、オープンソースのOSやソフトウェアには、将来的に希望が持てると思っています。ぶらっとホームの本田さんともよくお話しするんですけど、オープンソースで面白いことをやって、世の中を変えてゆきたいですね。

誰かが面白いことを始めると、評価する人や模倣する人も出てきますしね。

関：最近もリーナス・トーバルズの「それが僕には楽しかったから」が日本で出版されましたが、あそこに出てくるリーナスの自信というか確信って凄いですよ。オープンソースの世界で仕事をしてゆくからには、リーナスのような自信や確信をもって事業を進めてゆきたいと思っています。

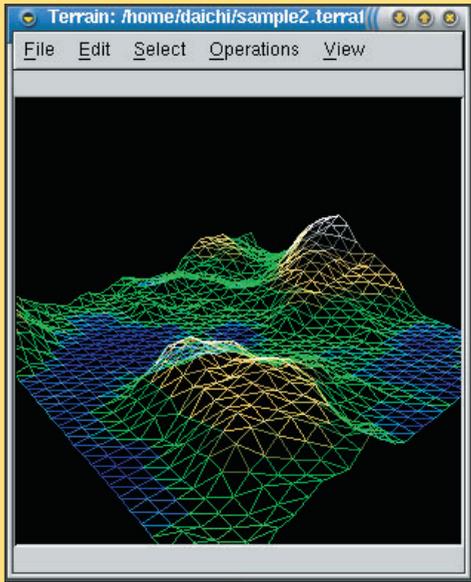
楽しみですね。今日はいろいろと興味深いお話を聞かせていただき、ありがとうございました。



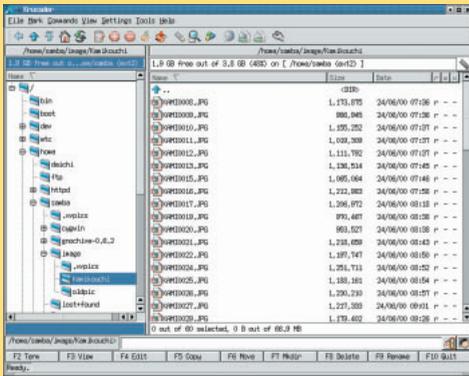
写真4 TUXIAのシン・クライアント。郵政省の「TAO」プロジェクトにも使用された。

Free Application Showcase

文：出井 一
Text : Hajime Dei



Terraform P.118



Krusader P.111



The Blue Mango Quest P.120

-  **111**
 KDE2用の高機能ファイルマネージャ
Krusader
-  **114**
 CD-Rや音楽CDのカバーを作成・印刷する
KCDLabel
-  **116**
 Xのルートウィンドウなどに画像を表示
ChBg
-  **118**
 リアルな3次元の地形を創造する
Terraform
-  **120**
 パックマン風の3Dアクションゲーム
The Blue Mango Quest
-  **122**
 ペンギンが活躍するタイピングゲーム
TuxTyping
-  **123**
 クールなグラフィックのパイプゲーム
PipeNightDeadms
-  **124**
 計算過程が残る四則演算電卓
gtapecalc
-  **125**
 プロセス情報をコンソールに一覧表示
ncps

紹介したソフトは、すべて付録CD-ROMに収録されています。

KDE2用の高機能ファイルマネージャ

Krusader

バージョン : 0.95

ライセンス : GPL

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

ビルドとインストール

Krusaderは、tarボールとRPMバイナリ、ソースパッケージが配布されているので、ディストリビューションに合わせて選択しよう。

tarボールを利用する場合は、「./configure」「make」でビルドし、「su」でrootになった状態で「make install」としてインストールする一般的な手順でOKだ。

一方、RPMバイナリパッケージはMandrake用なので、国産ディストリビューションでは利用できない。以下の手順でソースパッケージに含まれるSPECファイルを修正して、バイナリパッケージを作成しよう。

まず、「su -」としてrootになった状態で、「rpm -i krusader-0.95-1.src.rpm」として、ソースパッケージをインストールする。

続いて、/usr/src/redhat/SPECSにインストールされたSPECファイル(krusader.spec)を適当なエディタで編集し、11行目に「BuildRoot: /var/tmp/krusader-%{version}-root」という行を挿入する。続いて、30行目付近の「--with-qt-dir=」で始まる行をまるごと削除する。これでSPECファイルの修正は完了だ。

あとは、「rpm -bb krusader.spec」とすると、修正したSPECファイルに基づくバイナリパッケージが作成されるので、「rpm -Uvh /usr/src/redhat/RPMS/i386/krusader-0.95-1.i386.rpm」としてインストールする。

起動と初期設定

ktermなどのコマンドラインで「krusader&」とするか、KDEメニューの[アプリケーション] - [Krusader]を選択すると起動する。

最初にKrusaderを起動した際には、アーカイブ系のコマンド(tarやgzipなど)を自動検索した後、各種設定を行うためのKonfiguratorダイアログ(画面1)が開く。

まずは、[Startup]ページで、左右のパネルの初期ディレクトリを設定する。一方をユーザーのホームディレクトリ、もう一方をワークディレクトリに設定するのがお勧めだ(画面2)。

また、[General]ページの[Temp Directory]の設定は、初期設定の「/tmp/krusader.tmp」ではなく、ユーザーごとの独立したディレクトリ(ホームディレクトリ以下のtmpなど)に変更したほうがいい。ここで設定したディレクトリの、読み込み・書き込み・実行許可を持っていないと、後述する仮想ファイルシステムの動作に支障をきたすので注意しよう。

Konfiguratorは、Krusaderからいつでも開けるので、残りの設定はあとで行えばいい。[OK]ボタンを押してダイアログを閉じれば設定完了だ。

さまざまな用途を持つツインパネル

Krusaderのウィンドウは左右に2つのパネルを持つツインパネル構成で、2つのディレクトリのファイル一覧を同時に表示できる(画面3)。なお、初期

Krusaderは、KDE2用の高機能なファイルマネージャだ。2つのディレクトリを同時に表示できるツインパネルを持ち、ドラッグ&ドロップをサポートする。各種アーカイブやRPMパッケージ、FTP / Sambaなどを透過的に扱う仮想ファイルシステム(VFS)が最大の特徴だ。このほか、マウントマネージャやブックマークマネージャ、クイックビューパネルなどを備えている。動作にはKDE2 / Qt2が必要だ。

設定では「.」で始まる隠しファイルリストに表示される。これを非表示にするには、[View] - [Show Hidden Files]のチェックを外せばいい。

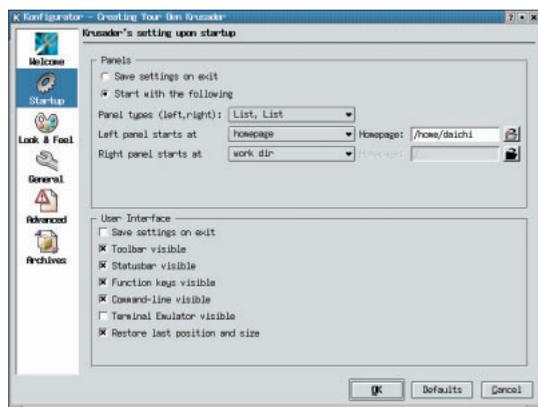
また、[View] - [Executables]を選択すると実行属性を持つファイル、[View] - [Custom]では指定したパターン(画面4)にマッチするファイルを対象として、ファイルの表示が絞り込まれる。これらの設定は左右のパネルで独立しており、フォーカスを持つ(上下の部分が水色反転表示の)パネルの設定が変更される。

一方のパネルをファイル一覧以外の用途に使うことも可能だ。たとえば、[View] - [Tree]を選択すると、フォーカスを持つパネルの表示がディレクトリツリーに切り替わる(画面5)。この状態で、ツリー中のディレクトリをダブルクリックすると、そのディレクトリのファイル一覧がもう一方のパネルに表示される。

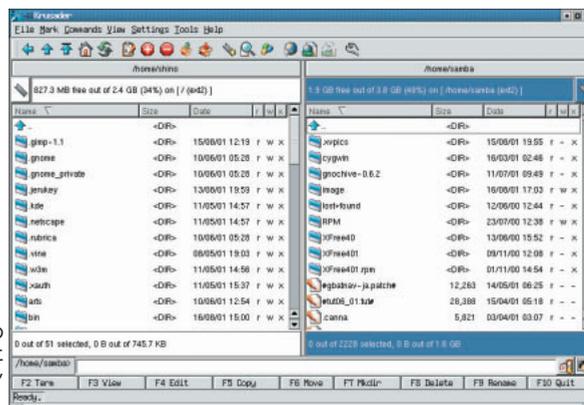
また、[View] - [Quickview Panel]を選択すると、今度はフォーカスを持つパネルが画像ビューアに切り替わり、



画面1 Krusaderの設定をGUIで行うKonfiguratorダイアログ。



画面2 起動時に左右のパネルに表示するディレクトリなどを設定する。



画面3 Krusaderのウィンドウは左右にパネルを持つツインパネル構成だ。

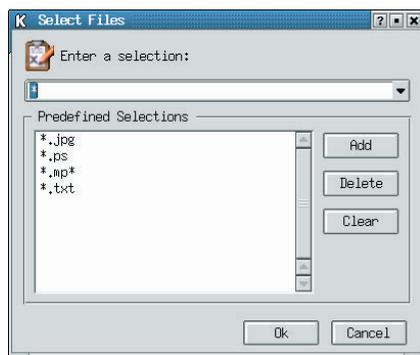
もう一方のパネルでクリックしたファイルの画像が表示される(画面6)。このように、2つのパネルの用途を柔軟に切り替えられることがKrusaderの大きな特長だ。

ドラッグ&ドロップでコピー

ファイルのコピーや移動は、ドラッグ&ドロップで行う。左右どちらかのパネルから目的のファイルをドラッグして、もう一方のパネル上でドロップすればいい。ポップアップメニューが開くので、コピーなら[Copy Here]、移動なら[Move Here]を選択する。リンクを張ることも可能だ。

なお、Shift - クリックやCtrl - クリックによって複数のファイルを選択してからドラッグすれば、それらのファイルをまとめて処理できる。

ファイルの削除やリネーム、閲覧や編集などを行うには、対象となるファイルを選択してから、ウィンドウ下部



画面4 ファイル表示を絞り込むパターンを設定するダイアログ。

のファンクションキーをクリックするか、対応するキーを押せばいい。閲覧にはKViewer、編集にはKwrite(変更可)が使われる。

また、ファイルをダブルクリックすると、KDEのMIMEタイプ設定に基づいたソフトが起動する。たとえば、画像ファイルの場合はPixie、MP3 / WAVE / MPEGファイルならNoatun、テキストファイルならKwriteといった具合だ。KDEアプリの統一感とあいまって、かなり使い心地がいい。

仮想ファイルシステムを体験

Krusaderの最大の特徴は、各種アーカイブファイルやRPMパッケージをはじめ、FTP接続やNFS、Sambaに至るまで、通常のディレクトリとほぼ同様に扱える「仮想ファイルシステム(VFS)」にある。

まずは、アーカイブとRPMパッケージを使って仮想ファイルシステムを体験してみよう。tarやZIP、LHAなどのアーカイブファイルをダブルクリックすると、アーカイブに含まれるファイル一覧が通常のディレクトリのようにパネルに表示される。ファイルの閲覧なども可能だ。

なお、UNIX系OSで広く使われているいわゆるtarボールは、tarでアーカイブされたのち、gzip(あるいはbzip2)でさらに圧縮されているため、tar内部

のファイル一覧を表示するには2段階の操作が必要となる。

RPMパッケージの場合も同様で、最初の段階ではファイルリスト、ヘッダー、cpioアーカイブの3つが表示され、cpioアーカイブをダブルクリックするとパッケージに含まれるファイル一覧が表示される。

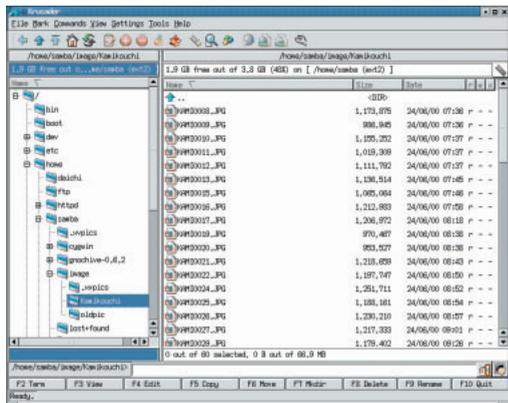
リモート接続を行う

Krusaderには、FTPやSambaのクライアント機能も用意されており、通常のディレクトリとほぼ同様の感覚で、FTPサーバやSamba共有ディレクトリのファイルを扱える。

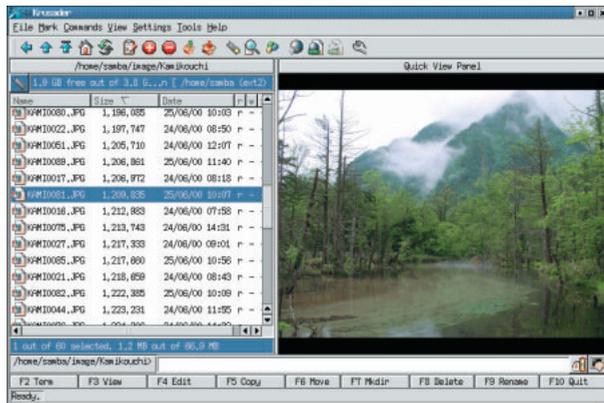
ツールバー右から4つめの[FTP Connect]ボタンを押して、リモート接続を管理するRemoteManダイアログを開いてみよう(画面7)。

RemoteManダイアログを初めて使う場合は、[New Connection]ボタンを押して、新たな接続セッションを登録する。ホストのURL設定の先頭部分を「ftp://」にするとFTP、「smb://」にするとSambaになる。このほか、ユーザー名やパスワード(アノニマス接続用の設定も可能)、リモートディレクトリなどを設定しよう。

設定終了後、[Connect]ボタンを押すと接続処理が行われ、取得したファイル情報に基づいて、フォーカスを持つパネルにリモートディレクトリのフ



画面5 片方のパネルをディレクトリツリー表示に切り替えることも可能。



画面6 クリックした画像がクイックビューのパネル内に表示される。

ファイル一覧が表示される。

次回からは、RemoteManダイアログのツリーにセッションが登録されているので、それを選択して[Connect]ボタンを押すだけでいい。

こうしたリモートディレクトリに対しても、通常のディレクトリと同様に、ドラッグ&ドロップによるファイルのコピーや移動、ファンクションキーを使った閲覧などが可能だ。

ただし、リモートディレクトリ上のアーカイブをダブルクリックしても、内容は表示されない。ローカルディレクトリにコピーする必要がある。

FTP / Sambaのリモート接続を終了するには、ツールバー右から2番目の[FTP Disconnect]ボタンを押せばいい。パネルは元のローカルディレクトリの表示に戻る。

マウントマネージャを利用する
ツールバー右から5番目の

[Mount Man]ボタンを押すと、MountManダイアログが開く(画面8)。このダイアログには、/etc/fstabに記述されているファイルシステムの一覧が表示されており、CD-ROMやフロッピーのマウント操作などが可能だ。

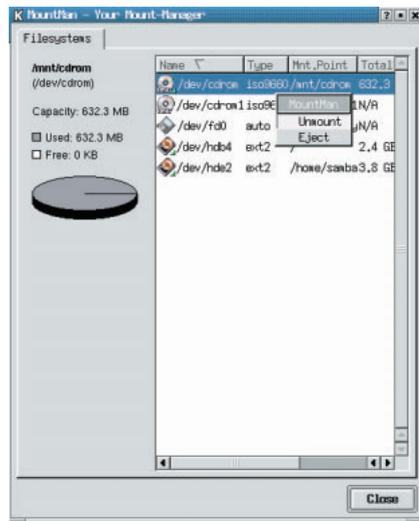
ただし、一般ユーザーでもCD-ROMなどをマウントできるよう、あらかじめ/etc/fstabの該当行にuserオプションなどを設定しておく必要がある。

CD-ROMをマウントするには、CD-ROMをドライブに挿入後、リスト中の「/dev/cdrom」をダブルクリックすればいい。マウントした状態でダブルクリックすると、アンマウントされる。右クリックメニューからマウントやイジェクトも可能だ。

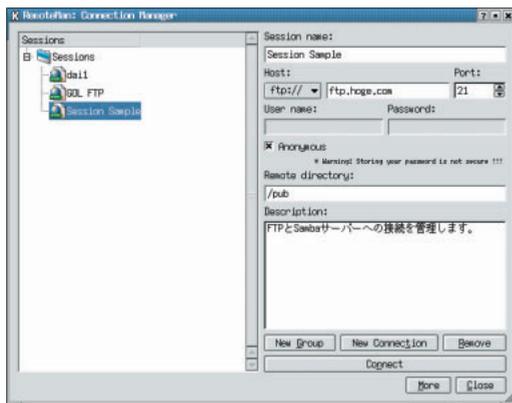
このほか、よく参照するディレクトリを登録するブックマーク機能も用意されている。ツールバー右から7番目の[Add the current path to your bookmarks]ボタンを押すと、現在のデ

ィレクトリがブックマークに追加される(画面9)。

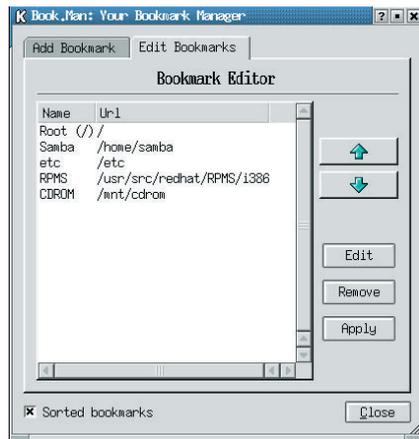
その後、パネルの上部にある「しおり」をクリックし、リストから目的のブックマークを選択すれば、ディレクトリが切り替わる。



画面8 MountManダイアログでは、CD-ROMのマウントをマウスで行える。



画面7 FTP/Sambaサーバーへの接続を管理するRemoteManダイアログ。



画面9 よく参照するディレクトリをブックマークに登録しておく。

CD-Rや音楽CDのカバーを作成・印刷する

KCDLabel

バージョン: 2.7

ライセンス: GPL

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

KCDLabelは、CDケース用のカバーやブックレット、CDに貼るステッカーなどを、画面上でイメージを確認しながら作成できるソフトだ。単にテキストや画像を並べるだけでなく、CD-Rのファイルツリー表示や、CDDDBを利用した音楽CDの曲名表示、テキストの変形や背景のグラデーションなど強力な機能を持つ。作成したラベルはKCDLabelから直接印刷できる。動作にはQt2.2.4 / KDE2.1以降が必要だ。

ビルドから起動まで

KCDLabelの最新版は、ソース一式をtar + gzipしたtarボールのみ配布されている。「./configure」「make」でビルドし、「su」でrootになった状態で「make install」でインストールする一般的な手順だ。

ktermなどのコマンドラインで「kcdlabel&」とするか、KDEメニューの[アプリケーション] - [KCDLabel]を選択して起動すると、ウィンドウが開いて白紙のラベルが表示される。

ラベルを構成する3種類のページ

ラベルには、CDケースの裏側に入れる「バックページ」、表側に入れる「フロントページ」、CDレーベル面用の「ラウンドページ」が含まれる。これらのページに、テキスト / 画像 / ファイル一覧 (ボディ) / 背景を組み合わせで配置できる。

ページの種類は、ウィンドウ内に表示される境界線の形状や、右下のページ番号で区別できる。たとえば、起動

直後に表示されるバックページの場合、左右に「スライド」(背表紙)が付き、ページ番号は「0」だ(画面1)。

キーを押すと、スライド部分のないフロントページに切り替わり、ページ番号が「1」になる。フロントページは複数(最大998ページ)用意されているので、歌詞カードのようなブックレットも作成できる。 / キーで前後のページに切り替え可能だ。

さらに、[Page] - [Go to round page]を選択すると、CDの形をしたラウンドページに切り替わる(画面2)。ページ番号は「999」だ。

ページにテキストを配置

ページに配置できるテキストには、タイトルなどに適した「単独行」と、曲目一覧などに適した「複数行」、凹凸などの変形が可能な「湾曲」の3種類がある(画面3)。

ツールバー右の3つのボタンがこれらに対応しており、配置したい文字列をダイアログに入力すればいい。なお、

日本語も入力できるが、ページに表示する時点で削除されてしまう。

これらのダイアログでは、フォントの設定や、文字色・背景色(透明可)を変更できる。さらに、複数行テキストでは行間スペース、湾曲テキストでは変形の種類(凸形・凹形・円形)なども設定可能だ。

配置後のテキストをダブルクリックすると、選択状態を表わす緑色の矩形が周囲に表示され、ドラッグで位置を変更できる。また、右クリックメニューなどにより、設定内容の変更や削除、センタリングなども可能だ。

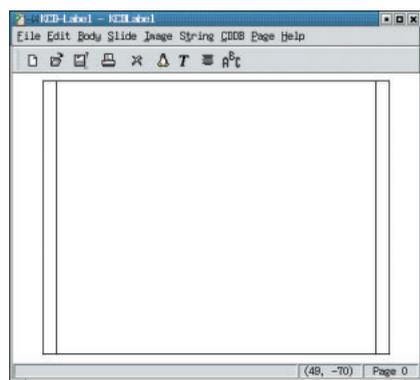
なお、バックページの左右にあるスライドについては、[Slide]メニューに設定が独立しており、単独行テキストのみ配置できる。

画像の配置と背景の設定

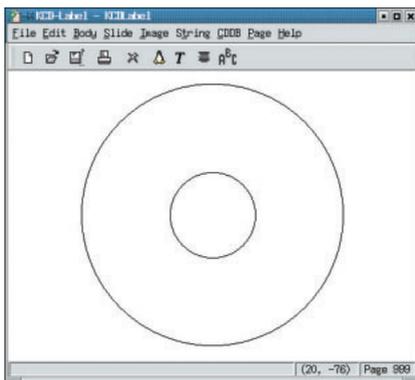
JPEG / PNG / XPMなど、Qtが対応する形式の画像もページ内に配置できる。ツールバー右から4番目のボタンを押して画像ファイルを選択すればいい。ページ内に複数の画像を配置することも可能だ(画面4)。

配置後の画像は、テキストと同様にダブルクリックで選択できる。右クリックメニューを利用して、サイズの変更やタイル表示、背景の透明化の切り替えといった操作が可能だ。

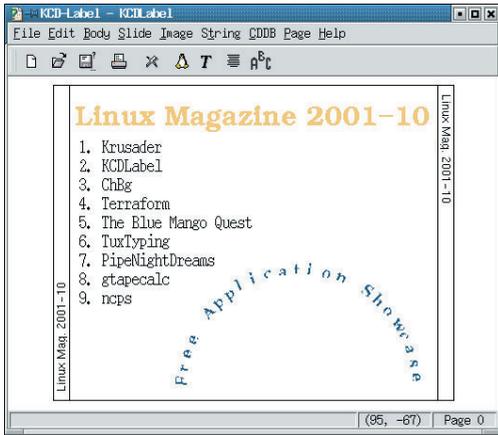
また、ページの背景を単色やグラデーションで塗りつぶすこともできる。[Page] - [Set page background]を選択し、グラデーションの種類(8種類)



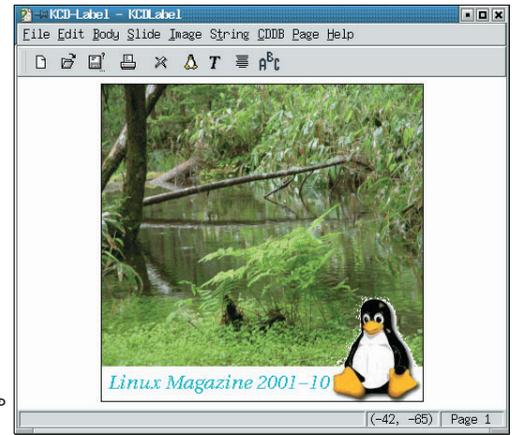
画面1 起動時にはケースの裏側に入れるバックページが表示される。



画面2 CDのレーベル面用のラウンドページも用意されている。



画面3 単独行・複数行・湾曲の各テキストをバックページに配置する。



画面4 各ページには、背景やロゴなどの画像も配置できる。

や使用する色を選択すればいい。背景の設定はページごとに独立している。

ファイルツリーを表示する

KCDLabelには、指定したディレクトリ（初期値は/mnt/cdrom）以下のファイル構成をツリー表示する「ボディ」も用意されており、CD-ROMやCD-Rなどの内容を確認するのに便利だ。もちろん、テキストや画像、背景を重ね合わせることもできる。

CD-ROMなどをマウントした状態で、[Body] - [Read content]でファイル情報を取り込む。このままだとボディは表示されないのので、[Body] - [Number of columns]を[1]に変更しよう。これで、フロントページの2ページ目以降（変更可）に、CD-ROMなどのファイル一覧がツリー表示される（画面5）。

2段や3段表示に変更したり、水平線

やツリー表示の線を消すことも可能だ。また、ページ数が膨大になる場合は、表示するディレクトリの深さを制限するとよい。

音楽CDの曲目を取り込む

このほか、音楽CDのタイトルやアーティスト名、曲目情報をインターネット上のCDDDBサーバから取り込みむ機能も用意されている。

まずは、音楽CDをCD-ROMドライブに入れた状態で[CDDDB] - [Insert a CDDDB entry]を選択する。ダイアログが開いたら、[Disk ID]ボタンを押してディスク情報を読み込み、[OK]ボタンでCDDDBサーバに接続する。

検索が終了すると、マッチした音楽CD情報の一覧が表示される。どれかひとつを選択すると、フロントページ（1ページ目）にタイトルとアーティスト

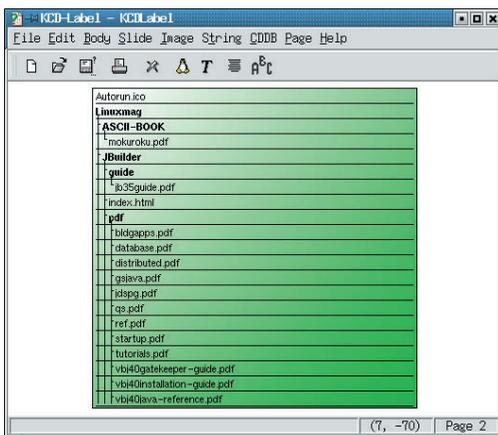
名、バックページには曲目情報などが自動的にテキストとして配置される（画面6）。あとは、好みに応じてフォントの変更などを行えばいい。

完成したラベルを印刷する

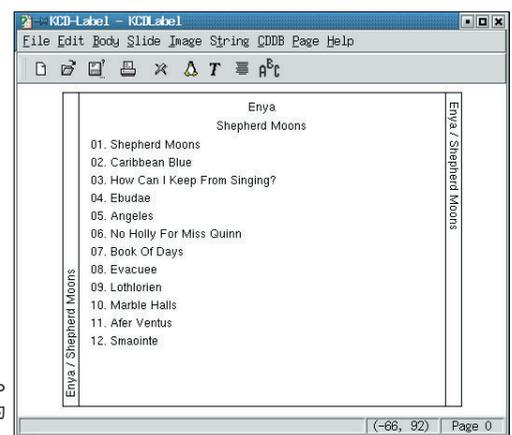
完成したラベルは、[File] - [Save]で独自形式（拡張子.kcd）で保存できるほか、KCDLabelから直接プリンタで印刷したり、EPS形式のファイルとして出力することも可能だ。

フロントページやバックページの印刷の際は、1枚の紙に2ページ分のラベルがまとめて出力されるので、切り取って使えばいい。

一方、CDのレーベル面に貼り付ける（あるいは直接印刷する）ためのラウンドページの印刷は、他のページとは独立している。上下・左右のマージンも細かく設定可能だ。



画面5 ファイルツリーを表示する「ボディ」を配置。多段表示も可能だ。



画面6 音楽CDのタイトルや曲目情報を取り込んで自動的に配置する。

Xのルートウィンドウなどに画像を表示

ChBg

バージョン: 1.5

ライセンス: GPL

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

ChBgは、Windowsにおける「壁紙チェンジャ」のようなソフトで、Xの背景（ルートウィンドウ）に登録した複数の画像を切り替えて表示する。imlibやgdk_pixbuf、imlib2などのイメージライブラリに対応しており、ほとんどの形式の画像を表示可能だ。また、フラクタルなどの計算に基づく画像をシェーディングして表示することもできる。実行にはGTK+といずれかのイメージライブラリが必要だ。

ビルドとインストール

ChBgは、tarボールとRPMソース、バイナリパッケージが配布されているので、ディストリビューションに応じて選択しよう。

tarボールの場合は、「./configure」

「make」でビルドし、「su」でrootになってから「make install」でインストールする一般的な手順だ。使用するイメージライブラリはconfigureスクリプトによって、自動的に選択されるが、「./configure」起動時の--with-image-libraryオプションで指定することもできる。

一方、RPMのバイナリパッケージは、imlibを使用するパッケージとgdk_pixbufを使用するパッケージがそれぞれ配布されているので、いずれか一方をインストールする。ソースパッケージからリビルドする場合は、tarボールと同様にconfigureスクリプトが自動選択してくれる。

セットアップウィンドウを開く

ktermなどのコマンドラインで「chbg&」とするか、GNOMEメニューの[プログラム] - [グラフィック] - [ChBg]を選択すると、ChBgのセットアップウィンドウが開く。設定項目がジャンル別にページ分けされ、左側のタブで切り替え可能だ。付属の日本語カタログにより、日本語環境ではウィンドウ内の表示やチップヘルプに日本語が使われる（画面1）。

まずは、[セットアップ]ページで表示モードを設定しよう。ChBgは、ルートウィンドウに画像を表示するだけでなく、通常のウィンドウに表示したり、スクリーンセーバとして実行することもできる。

画像ファイルを登録する

それでは、複数の画像ファイルを切り替えて表示する方法を説明しよう。まず[画像リスト]ページに切り替え、[画像追加]ボタンを押してダイアログ

を開く（画面2）。

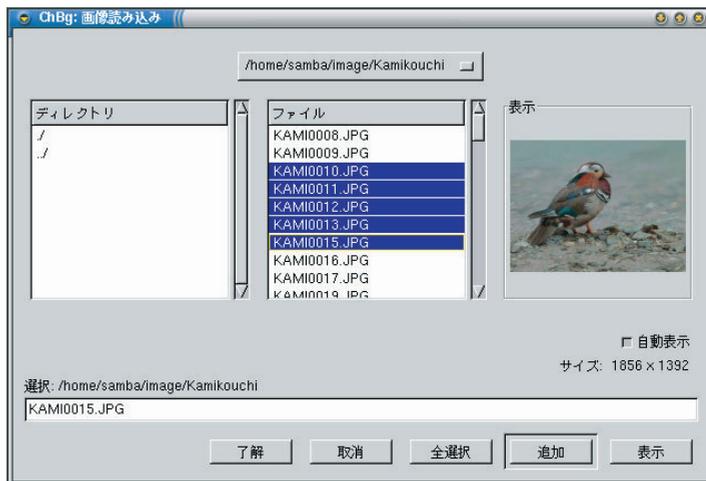
ここでは、ChBgで表示させたい画像をクリックで選択する。複数選択も可能だ。右下の[自動表示]をチェックしておけば、ファイルを選択した時点でサムネイル（縮小イメージ）が右側に表示される。ディレクトリ以下の画像ファイルを一括登録することも可能だ。[追加]ボタンを押すと、選択したファイルがセットアップウィンドウの背景画像リストに登録される。

背景画像リストの画像ファイルに対しては、右クリックメニューで選択や削除、画像の参照（ウィンドウ表示）、全画面表示、サムネイル表示などの操作を行える。

特に、独立したウィンドウに表示されるサムネイル（画面3）からは、画像の表示はもちろん、スライドショーやルートウィンドウへの表示も行える。つまり、ChBgを一種の画像ビューアとして利用できるわけだ。



画面1 ChBgの各種設定を行うセットアップウィンドウ。



画面2 このダイアログで画像を追加する。プレビューも可能だ。



画面3 登録画像を独立したウィンドウにサムネイル表示する。

画像ファイルを切り替えて表示

登録した画像ファイルを時間の経過とともに切り替えて表示するには、[プロパティ]ページで、切り替え後の待ち時間と、画像の描画方法を設定する(画面4)。

切り替え後の待ち時間は、分・秒単位で設定する(初期値5分)。「30秒から2分の間」のように、待ち時間を可変にすることも可能だ。

描画方法は、「敷き詰める」「最大化」「中央寄せ」といった基本的なものから、画像を反転させながら敷き詰める「ミラー」、画面にきっちり収まるように縮小して敷き詰める「Integer-敷き詰め」など各種揃っている。「画面中央で最適表示」を選択すれば、拡大率や画面内での割合などを細かくカスタマイズ可能だ。

設定が完了したら、ツールバー中央の[現在の設定で実行]ボタンを押して、実際の表示を確認しよう。待ち時間が経過すると次の画像に切り替わるはずだ。なお、待機中に変更した設定は[setupの変更を適用]ボタンを押すか、[画像の更新を停止]ボタンで更新停止するまで影響しない。

画像を切り替える際の画面効果は、



画面4 [プロパティ]ページで切り替えの時間間隔と描画方法を設定。

[エフェクト]ページ上部の[画像変更時のエフェクト]で設定する(画面5)。29種類の切り替えエフェクトが用意されており、ランダムに選択することも可能だ。なお、エフェクトに要する時間は待ち時間に含まれない。

しばらく動かしてみて、好みの設定が決まったら、[シナリオを保存]ボタンで現在の設定をファイルに保存しておこう。続いて、[Setup windowを隠し、現在の設定で実行]ボタンを押す。これで、セットアップウィンドウが消えて、バックグラウンドで画像の切り替えが行われる。

シェーディング画像の表示

ChBgのもうひとつの楽しみ方は、フラクタルなどの計算に基づくシェーディング画像を表示することだ。ChBgには500種類以上のシェーディングエフェクトが用意されており、自动生成されるさまざまな画像を楽しめる。

シェーディング画像を表示するには、まず[セットアップ]ページで[シェーディング画像を連続表示]をチェックする。続いて、[エフェクト]ページでは、[画像変更時のエフェクト]を「なし」、[背景シェーディングエフェクト]を

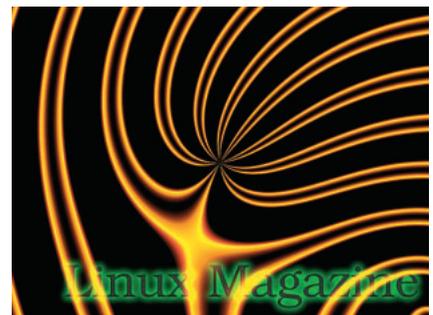


画面5 画像変更時やシェーディングのエフェクトを選択できる。

「ランダム」に変更する。最後に、[背景]ページで[背景の描画方法]を「Gradient」に設定すれば準備は完了だ。

なお、シェーディング画像は画像ファイルの背景として使われるので、シェーディング画像だけを楽しみたいなら、[画像リスト]ページで選択画像のリストを空にしておこう。

ツールバーの[現在の設定で実行]ボタンを押すと、シェーディング画像が生成され、ルートウィンドウに表示される(画面6)。待ち時間の設定などは画像ファイルの場合と同じだ。このほか、[バナー]ページでバナーやロゴなどの小さな画像を登録しておけば、シェーディング画像の中央や片隅にそれらを表示できる。



画面6 シェーディング画像とバナー画像を組み合わせて表示。

リアルな3次元の地形を創造する

Terraform

バージョン: 0.8.2

ライセンス: GPL/QPL

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

Terraformは、フラクタルを利用したリアルな3次元地形図を自動的に生成するソフトだ。数値地図 (DEM) や各種画像ファイルを地形図に変換することもできる。作成した地形は、3Dワイヤフレームや2D平面図など各種表示モードで眺めつつ、侵食などのフィルタを加えてさらに変形できる。フリーのレイトレーシングソフト「POV-Ray」と組み合わせれば、さらに美しい出力を得ることも可能だ。

ビルドとインストール

Terraformは、ソース一式をtar + gzipしたtarボールのみ配布されている。「./configure」「make」でビルドし、「su」でrootになってから「make install」でインストールする一般的な手順だ。

続いて、レイトレーシングソフトのPOV-Rayをインストールする。ソースからビルドすることも可能だが、Linux用にビルドされたバイナリとデータ類をまとめたtarボール (povlinux.tgz) が配布されているので、以下ではこれを利用する。

tarボールを展開したら、「su」としてrootになってから「./install」とすると、実行ファイルなどがインストールされる。

続いて、POV-Rayを利用するユーザーのホームディレクトリに、設定ファイル (.povrayrc) を作成する。tarボール付属のサンプル (povray.ini) を利用する場合は、「cp povray.ini

./povrayrc」とすればいい。

起動と初期設定

ktermなどのコマンドラインで「terraform&」として起動すると、小型のメインウィンドウが開く (画面1)。

まずは、[File] - [Options]でオプションダイアログを開いて、初期設定を行う。POV-Rayの実行ファイル名が「povray」となっているが、実際にはx-povray (X用) とs-povray (SVG A用) というファイル名だ。TerraformではX用を利用するので、ダイアログの設定を「x-povray」に修正しよう。

3次元地形図を自動生成する

3次元地形図を自動生成するには、メインウィンドウの[File] - [New]以下の項目を選択する。初めは[Random]を選択すればいい。面倒な手法やパラメータの設定が自動的に行われる。その他の項目を選択した場合は、ダイアログが開いて、手法ごとのパラメータ

を細かく設定可能だ (画面2)。

いずれにせよ、指定された手法とパラメータに基づくフラクタル地形が自動生成され、地形図ウィンドウが新たに開いて、2次元平面図が表示される (画面3)。

このほか、市販やフリー配布の数値地図 (DEM形式など) を読み込んだり、各種形式の画像ファイルを読み込んで、3次元地形図に変換することも可能だ。画像ファイルの場合は、明度に応じた地形が作成される。

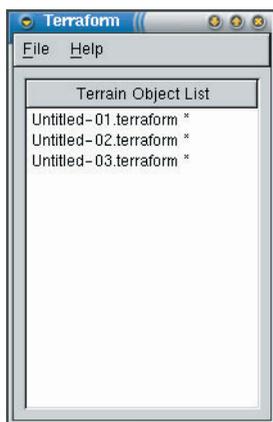
なお、Terraformでは、一度に複数の3次元地形図を扱うことができ、メインウィンドウには、作成した地形図のリストが表示される。

3次元地形図を眺める

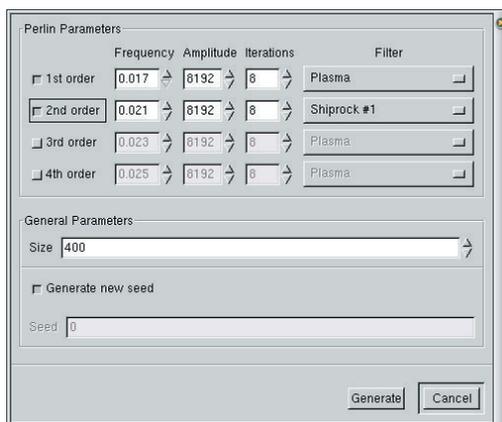
地形図ウィンドウの[View] - [Type]以下には、地形図の表示形式を変更する項目が用意されている。初期設定の「2D Plane」(平面図)のほか、「2D Contour」(等高図)や「3D Wire」(ワイヤフレーム)などに切り替え可能だ (画面4)。

ワイヤフレームの状態では、マウスのドラッグで地形を回転させ、好きなアングルから眺められる。また、[View] - [Type] - [Auto Rotate]をチェックして、自動的に地形を回転させることも可能だ。

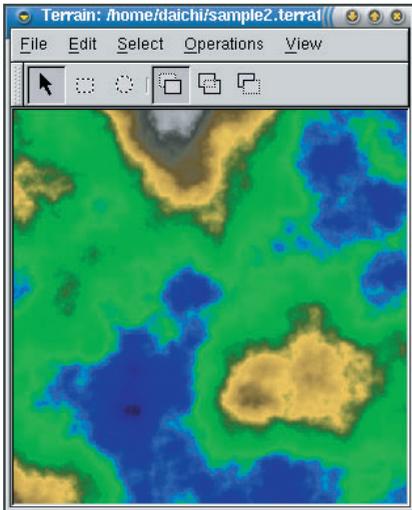
なお、3次元の表示形式には、ワイヤフレームより表現力の高い「3D Height」や「3D Light」も用意されているが、これらは回転させて眺める



画面1 小型のメインウィンドウには、地形図のリストが表示される。



画面2 地形図の生成に使われるパラメータを制御することも可能だ。



画面3 生成された地形の2次元平面図が別ウィンドウに表示される。

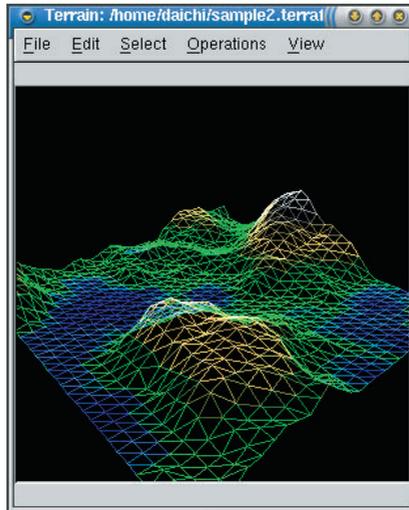
ことはできない。

また、表示に使われるカラーマップも、[View] - [Colormap]以下の項目で変更できる。たとえば、初期設定の「Land」から「Desert」に切り替えると、砂漠風の表示になる（画面5）。

3次元地形図を変形させる

Terraformでは、生成した地形をそのまま使うだけでなく、移動や反転などの基本操作や、クレーターや侵食（Erode）といったフィルタを利用して地形を変形できる。また、地形の一部領域だけを対象として、こうした操作を加えることも可能だ。

地形の選択はツールバーのボタン、基本操作は[Operations] - [Basic]以下、



画面4 好きなアングルから眺められる3次元ワイヤーフレーム表示。

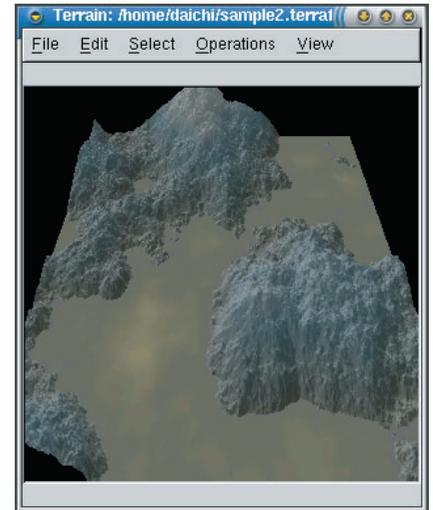
フィルタ類は[Operations] - [Filter]以下のメニューに用意されている。フィルタの場合は、細かなパラメータをダイアログで設定可能だ。

なお、こうした操作の結果が気に入らなければ、[Edit] - [Undo]でダイアログを開いて、過去5回の操作まで取り消せる（画面6）。

気に入った地形ができれば、[File] - [Save As]で名前を付けてファイルに保存しよう。Terraform専用形式の場合、拡張子は「.terraform」になる。BMP形式などの2次元画像や、VRML形式で保存することも可能だ。

POV-Rayを利用する

美しい3次元地形図を得るには、



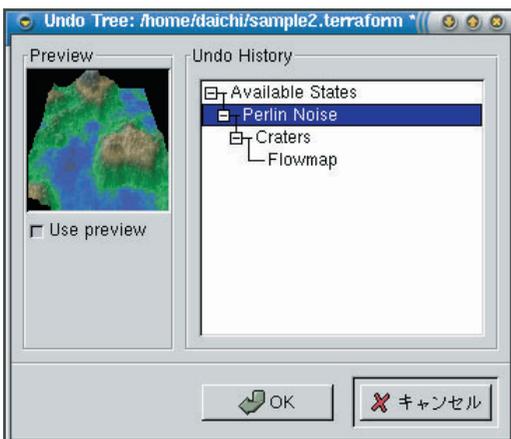
画面5 カラーマップを「Desert」に切り替えると、砂漠風の景色になる。

POV-Rayの助けを借りてレイトレーシングを行う必要がある。

まずは、地形図ウィンドウの[Edit] - [Options]で設定ダイアログを開き、[POV-Ray]ページで、地形のテーマと照明レベルを設定する。Terraformのカラーマップとは独立していることに注意しよう。

続いて、地形図ウィンドウの[File] - [Render] - [POV-Ray]を選択すると、POV-Rayによりレイトレーシングされた美しい画像が別ウィンドウに表示される（画面7）。

なお、照明レベルの設定によっては、表示が完了するまでにかかり時間がかることがあるので、注意しよう。



画面6 過去5回の操作をプレビューで確認しつつ取り消すことが可能だ。



画面7 POV-Rayを利用してレイトレーシングした美しい3次元地形図。

バックマン風の3Dアクションゲーム

The Blue Mango Quest

バージョン : 0.6.1

ライセンス : GPL

<http://mangoquest.sourceforge.net/>
[http://www.libsdl.org/\(SDL\)](http://www.libsdl.org/(SDL))

ビルドとインストール

まず最初に、Simple DirectMedia Layer (SDL) と、それを補助するSDL_image、SDL_ttfをインストールする。いずれもtarボールとRPMパッケージが配布されている。

SDLのRPMバイナリパッケージを利用する場合は、「su -」としてrootになってから「rpm -Uvh SDL-* 1.2.2-1.i386.rpm」とする。「SDL-」の後に「*」を付けるのは、SDL-develパッケージ (SDL-devel-1.2.2-1.i386.rpm) も一緒にインストールするためだ。

一方、SDLのtarボールを利用する場合は、「./configure」「make」でビルドし、「su」でrootになった状態で「make install」でインストールする一般的な手順でOKだ。

続いて、SDL_imageとSDL_ttfについても、同様の手順でビルドとインストールを行う。バイナリパッケージを利用する場合は、SDL_image-devel、SDL_ttf-develパッケージもインストールすることを忘れずに。

以上の準備が完了したら、Mango

Questのビルドにとりかかる。Mango Questはソース一式をtar + bzip2したtarボールのみ配布されており、ビルドとインストールの手順は、SDLのtarボールの場合とまったく同じだ。

3次元迷路を駆け抜ける

ktermなどのコマンドラインで「mangoquest&」とすると、マップファイルなどが読み込まれてから3D迷路のウィンドウが開く。スペースキーを押すとプレイ開始だ。画面表示はQuakeなどのFPS系ゲームに共通の形式で、プレイヤーキャラの視点で周囲の風景が表示される (画面1)。

操作にはキーボードを使用する。基本的には、カーソルキーでプレイヤーキャラを前後左右に移動させるだけでいい。このほか、Page Up / Downキーで視線の移動、Endキーで視線の水平位置への復帰が可能だ。

ゲームの目的は、地面に落ちているガム (DituBox) をすべて取ることだ。しかし、迷路にはモンスター (Shmollux) が何匹も徘徊しており

The Blue Mango Quest (以下Mango Quest) は、モンスターを避けながら3次元迷路をさまようバックマン風の3Dアクションゲームだ。迷路上にはさまざまなアイテムが落ちており、プレイヤーを助けたり、惑わせたりしてくれる。付属のエディタを使って、ユーザーがオリジナルレベルを作成することも可能だ。実行にはマルチメディアライブラリのSDLと、その関連ライブラリSDL_image、SDL_ttfが必要となる。

(画面2)、プレイヤーキャラがモンスターに襲われると、画面右下のライフが1つ減ってしまう。

モンスターに反撃するには、通路上に点在するパワーエサ (ShmixGum) を取る必要がある (画面3)。このアイテムに触れると、しばらくの間モンスターを食べることができるのだ。

画面左下のレーダーには、プレイヤーキャラ (赤) をはじめ、ガム (黄) やパワーエサ (青)、アイテム (緑)、モンスター (黒) などが色分け表示されるので、常にチェックしよう。

迷路上のすべてのガムを回収すれば、そのレベル (面) はクリアだ。逆に、プレイヤーキャラのライフがなくなるとゲームオーバーになる。

さまざまなアイテムを活用しよう

このゲームを面白くしているのが、迷路上に配置されたさまざまなアイテムの存在だ。これらは、プレイヤーキャラを助ける効果を持つ「ボーナスアイテム」と、困らせる効果を持つ「マリユスアイテム」に分類できる。



画面1 3次元迷路の中でモンスターとの戦いが始まる。



画面2 プレイヤーキャラに襲い掛かるモンスター (この顔は...)



画面3 トゲトゲのパワーエサを取れば、しばらくの間は無敵状態に。

ボーナスアイテムには、前述のパワーエサ (ShmiXGum) のほか、1つライフを増やすLife、モンスターに食べられなくなるShieldなど、ありがたいものが揃っている。これらは、取った瞬間に効果が現れ、しばらく経つと元の状態に戻る。

これに対し、マップ上のモンスターを一時的に全滅させるArmageddon、周囲のモンスターのみ全滅させるShock wave、プレイヤーキャラの動きが速くなるHigh speed、しばらくモンスターが動きを止めるTime stopperの4つは、画面下にストックでき、F1~F4キーでいつでも利用できる。

一方、マリユスアイテムについても、迷路が深い霧に覆われるDeep fogや、天井に張り付いてしまうCeiling、動く方向が逆になるInvert keysなど、バリエーション豊かだ。

アイテムについての詳細は、作者のWebページ「Let's talk about Bonus

and Malus」 (<http://mangoquest.sourceforge.net/doc/bonus.htm>) を参照してほしい (画面4)。

オリジナルレベルをデザインする

Mango Questには、Mango Peelerと呼ばれるレベル (面) エディタが付属しており、オリジナルレベルをデザインできる。

まずは、各ユーザーのマップファイルが作成されるディレクトリを用意する。「mkdir -p ./mangoquest/map」とすればいい (「.」に注意)。

ktermなどのコマンドラインで「mangopeeler hoge.shx&」とすると、hoge.shxというマップファイルが上記のディレクトリに作成される (すでにhoge.shxが存在する場合は、その内容が読み込まれる)。また、「mangopeeler&」としてマップファイル名の指定を省略すると、default.shxというマップファイルが使われる。



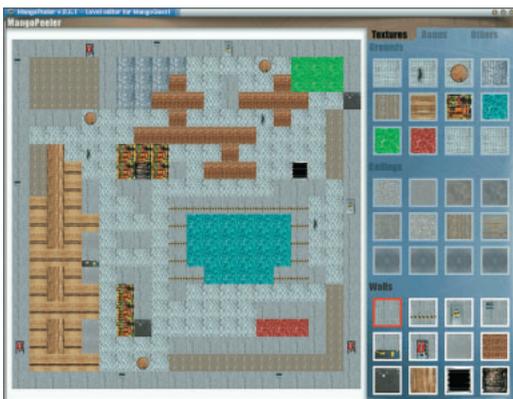
画面4 ゲーム中に登場するさまざまなアイテムを解説しているWebページ。

最初に行うのは、地面・天井・壁のデザインだ。12種類ずつ用意されたテクスチャをクリックで選択し、平面マップの一部をドラッグで範囲指定して配置する (画面5)。

続いて、右上の[Bonus]タブをクリックして画面を切り替え、アイテムを配置する (画面6)。各種アイテムのアイコンを選択し、平面マップをクリックして配置しよう (削除は右クリック)。また、アイテムの有効時間などは、/ キーで変更できる。

最後に、[Others]タブをクリックして画面を切り替え、プレイヤーキャラのスタート位置と向き、モンスターの出現位置などを設定する。

F2キーを押してマップファイルを保存したら、「mangoquest hoge.shx&」のように、そのファイル名をMango Questのコマンドラインで指定すると、自分でデザインしたオリジナルレベルをプレイできる。



画面5 地面と天井、壁のテクスチャを選択して迷路をデザインする。



画面6 さまざまな効果を持つアイテムを平面マップ上に配置する。

ペンギンが活躍するタイピングゲーム

Tux Typing

バージョン: 1.0

ライセンス: GPL

<http://www.geekcomix.com/dm/tuxtype/>
<http://www.libsdl.org/> (SDL)

ビルドとインストール

まず、SDLやその関連ライブラリをインストールしておこう。詳しい手順は、前ページの「The Blue Mango Quest」の記事に書かれている。SDL_mixerとSMPEGも同様の手順でインストールする。

TuxTypingは、tarボールとRPMバイナリパッケージの両方が配布されている。RPMパッケージを利用するにはrpm 3.0.5以降が必要だ。tarボールの



画面1 軽快なBGMとともにタイトル画面が表示される。

ほうは、configureスクリプトを利用する一般的な手順でビルドとインストールを行える。

落下する魚を受け止める

ktermなどのコマンドラインで「tuxype&」として起動すると、フルスクリーン画面に切り替わってタイトル画面が表示される(画面1)。ウィンドウ表示の状態遊ぶには、「tuxype-w&」と-wオプションを付けて起動すればいい。

タイトル画面のメニューでは、文字単位の「Key Cascade」か、単語単位の「Word Cascade」のいずれかをマウスで選択する。続いて、ゲーム難易度(Easy / Medium / Hardの3種類)を選択するとプレイ開始だ。

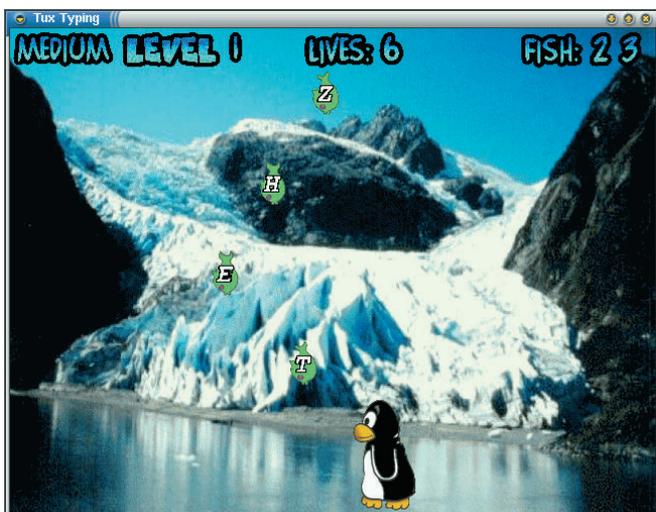
画面の下では、われらがTux君が魚を待ち受けている。操作は簡単で、落ちてくる魚に書かれた文字を順番にキ

ー入力すればいい。キーを間違えた場合はもう一度正しいキーを押そう。入力を受け付けると、Tux君が自動的に左右に移動し、落ちてきた魚を食べてくれる(画面2)。

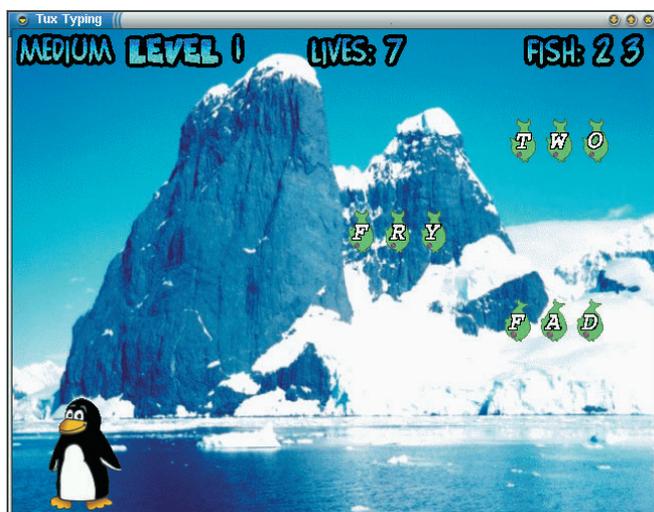
魚を食べ終わる前に次のキーを入力してはいけない。Tux君が別の位置に移動して、元の魚を下に落としてしまうからだ(LIFEが1つ減る)。Tux君のLIFEが0になるとゲームオーバーだ。一方、決められた数の魚を食べるとクリアで、次の面に切り替わる。

「Word Cascade」では、単語の文字数分の魚が並んで落ちてくるので、それらのキーを続けて押す(画面3)。最初は3文字の単語だけだが、面が進むに連れてもっと長い単語が現れる。

また、難易度を「Hard」にした場合は、魚ごとに落下スピードが変わるので、どの魚が先に下に落ちてくるのかの判断が重要になる。



画面2 画面上部から落ちてくる魚に書かれた文字のキーを押そう。



画面3 「Word Cascade」では単語を間違いなく入力しなければならない。

クールなグラフィックのパイプゲーム

PipeNightDeadms

バージョン : 0.8.0

ライセンス : GPL

<http://www.libsdl.org/projects/pipenightdreams/>

ビルドとインストール

はじめに、SDLとSDL_imageをインストールしておこう。詳しい手順は、2ページ前の「The Blue Mango Quest」の記事を参照されたい。

PipeNightDreamsは、tarボールとRPMソース、バイナリパッケージが配布されている。tarボールからのビルドとインストールはconfigureスクリプトを利用する一般的な手順だ。

バイナリパッケージが依存関係により導入できない場合は、「rpm --

rebuild pipenightdreams-0.8.0-1.src.rpm」としてソースパッケージからリビルドする。途中で「File to patch:」と表示されたら、「src/main.cpp」と入力すること。これで、バイナリパッケージが/usr/src/redhat/RPMS/i386以下に作成されるので、あとは通常の手順でインストールすればいい。

入口から出口までの経路を作る

ktermなどから「pipenightdreams&」として起動すると、クールなタイトル画面が表示される(画面1)。Enterキーを押すとプレイ開始だ。

画面上のフィールドには、パイプの入口と出口だけが配置されている。左上に次々に用意されるパイプをフィールド上に配置して、入口から出口までパイプを接続することがこのゲームの目的だ。

操作にはキーボードを利用する。赤

い線で書かれたカーソルをカーソルキーで上下左右に移動させ、画面左上(3つ書かれた一番下)に示されたパイプをスペースキーで配置する。

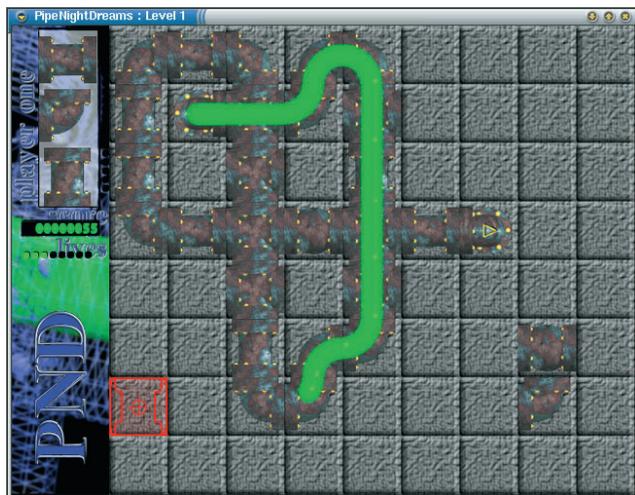
しばらくすると、入口から緑色の液体が湧き出して、少しずつパイプを進み始めるので、パイプからこぼさないように気をつけよう(画面2)。なお、この液体がまだ通過していなければ、パイプの置き換えも可能だ。

緑色の液体を無事に入口から出口まで到達させることができればそのレベルはクリアとなる。なお、出口までの経路を確保できたら、Enterキーを押すと液体が一気に進む。

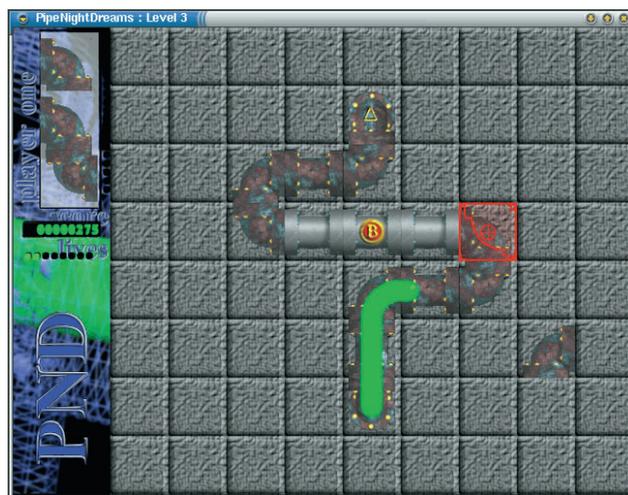
パイプを1つ通るたびに5点が加算され、ループを作って交差させると2倍になる。また、レベルが進むにつれ、液体を通すとボーナス点がもらえるボーナスパイプなどがフィールド上に登場する(画面3)。



画面1 クールなタイトル画面。Enterキーを押せばプレイ開始だ。



画面2 さまざまな形のパイプをフィールドに配置して出口までの経路を作る。



画面3 最初から配置された「B」のパイプを通してボーナス点をゲット。

計算過程が残る四則演算電卓

gtapecalc

バージョン: 0.3.10

ライセンス: GPL

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

ビルドとインストール

gtapecalcは、tarボールとRPMバイナリ、ソースパッケージが配布されている。RPMパッケージを利用するにはrpm 3.0.5以降が必要だ。

一方、tarボールのほうは、configureスクリプトを利用する一般的な手順でビルドとインストールを行える。その際、tarボール展開先のgtapecalc/gtapecalc.cの33行目「gtk_init (...);」の前に、「gtk_set_locale ();」という行を挿入してからビルドすれば、テープに付ける説明に日本語が使えるようになる。

計算過程が逐一テープに残る

ktermなどで「gtapecalc&」として起動すると、左側に電卓風のボタンと表示窓、右側にテープを持つウィンドウが開く(画面1)。

使い方は簡単で、[1]や[+]といったボタンをクリックして数値や演算子を入力していけばいい。もちろん、キーボードを利用して数値や演算子を直接入力することも可能だ。四則演算のほか、[%]ボタンによるパーセンテージの計算や、カッコを利用した計算順序の変更などを行える。

入力値は左上の表示窓のほか、演算

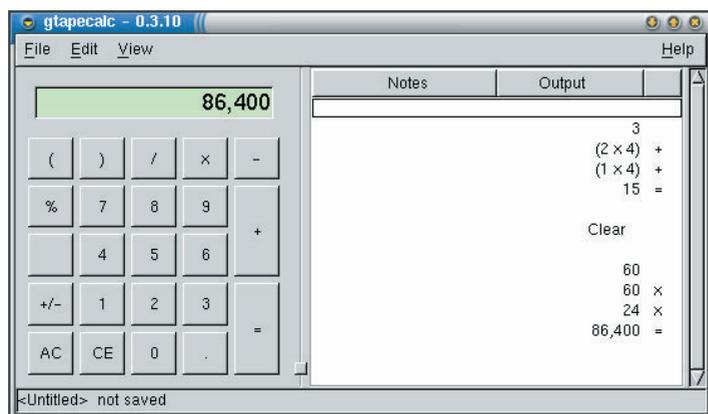
gtapecalcは、計算過程が記録される電卓ソフトだ。入力した数値や演算子の種類、計算結果などが、ウィンドウ右側の「テープ」に逐一表示されるので、入力ミスなどを簡単にチェックできる。入力値や計算値に対して説明文を付けることも可能だ。このほか、テープの内容をファイルに保存して、後でもう一度読み込んだり、プリンタで印刷することもできる。動作にはGTK+が必要だ。

子を入力した時点で右側のテープにも表示される。このテープによって過去に入力した数値や演算子、計算結果を簡単に確認でき、入力ミスなどをチェックできるわけだ。

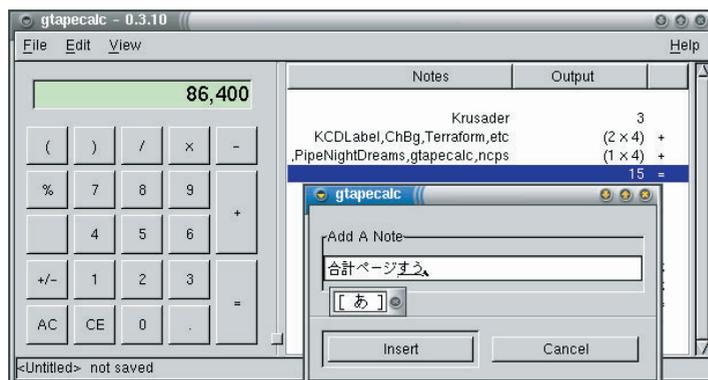
さらに、テープの各行をダブルクリックすると、ノート(説明文)を入力するためのダイアログが開いて、入力値や計算値の説明を書ける。前節の説明通りにソースを書き換えてからビルドすれば、日本語も入力可能だ(画面2)。ノートの削除などの操作は右クリックメニューで行う。

大切な計算を行ったテープは、[File] - [Save As]で名前を付けて保存しておけば、後で読み込んで再利用できる。ファイルはテキスト形式なので他のソフトにも流用しやすい。

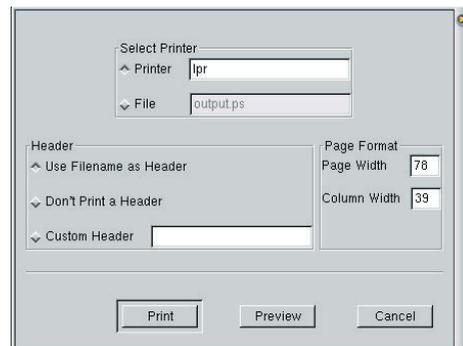
また、[File] - [Print]で開くダイアログから、プリンタに直接印刷したり、PostScript形式でファイルに出力することも可能だ(画面3)。ただし、現状では日本語のノートが文字化けしまうので注意されたい。



画面1 入力値や演算子、計算値が右側の「テープ」に記録される。



画面2 入力値などに付けるノート(説明文)を入力する(日本語対応)。



画面3 テープの内容をプリンタで印刷したり、PS形式で保存できる。

プロセス情報をコンソールに一覧表示

ncps

バージョン: 0.494

ライセンス: GPL

<http://www.math.ohio-state.edu/pschan/ncps/index.html>

ビルドから起動まで

ncpsは、ソース一式をtar + gzipしたtarボールのみ配布されている。configureスクリプトは含まれていないので、そのまま「make」でビルドする。インストールは、「su」としてrootになった状態で、「install -s ncps /usr/local/bin」とすればいい。

ncpsを起動するには、Xを起動していない状態（コンソール）か、X上の端末画面で「ncps」とする。なお、デフォルトのままのktermでは文字色が見にくいいため、Xではrxvtを利用するといいたいだろう。

プロセス情報の表示

画面には、ユーザーが所有するプロセスの一覧が表示される（画面1）。他のユーザーが所有するプロセスも表示するには、起動時に-aオプションを付けなければならない。

プロセス情報の項目は、左からユーザー名、端末、プロセスID、CPU使用率、メモリ使用率、nice値、コマンドラインの内容となっている。1~7キー

```

user  tty  PID  CPU  MEM  nice  command
daichi pts/0 1065  0.0  0.2  0    bash
daichi pts/0 1064  1.4  0.5  0    bash +sb
daichi tty1  1011  0.0  1.5  0    tasklist_applet --activate-goad-server tasklist_applet --goad-fd 13
daichi ?  1009  0.0  1.0  0    /usr/lib/gimp/1.1/plugin-script-fu -gimp 8
daichi ?  1008  0.0  3.7  0    gimp
daichi ?  1005  0.2  1.2  0    asclock_applet --activate-goad-server asclock_applet
daichi tty0  984  0.0  0.3  0    bash
daichi tty0  982  0.0  0.3  0    bash
daichi ?  981  1.3  1.5  0    gkrellm --sm-config-prefix /gkrellm-XFRUN/gnome-name-service
daichi ?  979  0.0  0.4  0    gnome-name-service
daichi ?  973  0.1  2.1  0    panel --sm-config-prefix /panel.d/default-DL
daichi ?  971  0.0  1.1  0    /usr/bin/sawmill --sm-client-id 117f0000100
daichi ?  959  0.0  0.8  0    kinout2 -canna -cannaserver localhost
daichi tty1  945  0.0  1.4  0    /usr/bin/gnome-session
daichi tty1  942  0.0  0.2  0    xinit /usr/X11R6/lib/X11/xinit/xinitrc --
daichi tty1  941  0.0  0.3  0    sh /usr/X11R6/bin/startx
daichi tty1  677  8.2  0.3  0    -bash
    
```

画面1 topコマンド風にプロセス情報が表示される。ソートも可能だ。

を押すと、番号に対応する項目を基準として降順・昇順にソートされる。また、0（ゼロ）キーを押すと、プロセスの親子関係を反映したツリー表示に切り替わる（画面2）。

プロセスが多すぎて画面に表示しきれない場合は、キーでカーソル（空色反転）を移動させればスクロールする。また、表示するプロセスをsキーで絞り込んだり（キーで以前の状態に戻る）fキーでプロセスを検索することも可能だ。

プロセスの強制終了

プロセスを強制終了させるには、目

画面3 複数のプロセスを選択しておけば、まとめて強制終了できる。

ncpsは、コンソールや端末画面にプロセスを一覧表示し、選択したプロセスにシグナルを送信して強制終了などを行うソフトだ。プロセスIDやCPU使用率などを基準としたソート、文字列による絞り込みや検索、プロセスの親子関係を反映したツリー表示への切り替えなども可能だ。また、複数のプロセスを選択しておき、同じシグナルをまとめて送信することもできる。

的のプロセスまで / キーでカーソルを移動させてから、kキーを押せばいい。画面右上に表示されているシグナルが送信される。

シグナルの初期値はSIGHUPだが、tキーを押すことで、SIGHUP SIGKILL SIGABRT SIGQUIT SIGINT SIGTERMと順次切り替わる。

また、複数のプロセスをまとめて強制終了させたいときは、それらをスペースキーで選択状態にする（画面3）。この状態でkキーを押せば、選択中のプロセスすべてに対して同じシグナルが送られる。

```

user  tty  PID  CPU  MEM  nice  command
daichi ?  1011  0.0  1.5  0    |-tasklist_applet SIGKILL
daichi ?  1008  0.0  3.7  0    |-gimp
daichi ?  1009  0.0  1.0  0    |-script-fu
daichi ?  1005  0.2  1.2  0    |-asclock_applet
daichi tty0  984  0.0  0.3  0    |@+bash
daichi tty0  1064  1.4  0.5  0    |@+rxvt
daichi pts/0 1065  1.0  0.3  0    |@+bash
daichi pts/0 1076  0.0  0.2  0    |@+ncps
daichi tty0  982  0.0  0.3  0    |-bash
daichi ?  981  1.3  1.5  0    |-gkrellm
daichi ?  979  0.0  0.4  0    |-gnome-name-serv
daichi ?  973  0.1  2.1  0    |-panel
daichi ?  971  0.0  1.1  0    |-sawmill
daichi ?  959  0.0  0.8  0    |-kinout2
daichi tty1  677  8.2  0.3  0    |@+bash
daichi tty1  941  0.0  0.3  0    |@+startx
daichi tty1  942  0.0  0.2  0    |@+xinit
daichi tty1  945  0.0  1.4  0    |'-gnome-session
    
```

```

user  tty  PID  CPU  MEM  nice  command
daichi ?  1011  0.0  1.5  0    |-tasklist_applet SIGKILL
daichi ?  1008  0.0  3.7  0    |@+gimp
daichi ?  1009  0.0  1.0  0    |-script-fu
daichi ?  1005  0.2  1.2  0    |-asclock_applet
daichi tty0  984  0.0  0.3  0    |@+bash
daichi tty0  1064  1.4  0.5  0    |@+rxvt
daichi pts/0 1065  1.0  0.3  0    |@+bash
daichi pts/0 1076  0.0  0.2  0    |'-ncps
daichi tty0  982  0.0  0.3  0    |-bash
daichi ?  981  1.3  1.5  0    |-gkrellm
daichi ?  979  0.0  0.4  0    |-gnome-name-serv
daichi ?  973  0.1  2.1  0    |-panel
daichi ?  971  0.0  1.1  0    |-sawmill
daichi ?  959  0.0  0.8  0    |-kinout2
daichi tty1  677  8.2  0.3  0    |@+bash
daichi tty1  941  0.0  0.3  0    |@+startx
daichi tty1  942  0.0  0.2  0    |@+xinit
daichi tty1  945  0.0  1.4  0    |'-gnome-session
    
```

画面2 0（ゼロ）キーを押すと、親子関係を反映したツリー表示に切り替わる。

「UNIXとPDP-11」

伝説のマシンを眺めながらUNIXについて考えた

文：榊 正憲

Text : Masanori Sakaki



2001年3月に、情報処理学会の併設展示として、「情報技術のエポック展」という催しがあった。日本の大学や企業で使われていた古いコンピュータやモジュールなどを展示するというものだ。ここでは、そのなかからUNIXに関連するものをいくつか紹介しつつ、かつてのUNIXの哲学と古典的名機PDP-11について簡単に解説しよう（はっきりいって、写真と本文はほとんど関係ありません）。

掲載の写真は、(社)情報処理学会「情報技術のエポック展」展示機器より。

PDP-11システム（写真左）
中央のラックがシステムの本体である。一番上にあるのは紙テープリーダ、中央にあるのがCPU本体、その下にはリムーバブルハードディスクがある。右側の印字式端末は、DEC Writerというコンソール。左側にCRT端末がある。

PDP-11のリムーバブルディスク（写真中）
このような脱着式のディスクカートリッジを使用できた。

PDP-11の紙テープリーダ（写真右）
紙テープにパンチされたプログラム、データなどを読み込む装置。このリーダは、ロール状に巻いたテープではなく、折り畳んだテープを読むようになっている。



のには、いくつかの理由がある。現在のLinuxの人気は、安価なこと、安定していることも大きいだろうが、本質はもっと別のところにある。何だろうか？ ソースが公開されていることだろうか？ オープンソースは、フリーのUNIX互換システムの大きな「売り」のひとつであることは確かだ。しかし、MicrosoftやNovellやIBMが成しえなかった、より本質的なものがUNIXにはある。

LinuxやFreeBSDなどの互換システムも含めて、UNIXオペレーティングシステムが長年にわたって多くの層に受け入れられている

今も昔も、ユーザーが求めることを、UNIXは快適に実現できる。

Column

ソースの公開

最近、ソースの公開がトレンドとなっているが、昔は現在とは別の意味合いもあったようだ。なんのことはない、「何か問題があったら、自分で調べて適当に直してね」という責任回避の手段だったのである。同様に、回路図などを公開しているシステムも珍しくなかった。

いまでは、権利がどうのといった理由で非公開になっている情報が多いが、オープンソースは企業にとってもメリットがある。マメなユーザーがごいねいにバグの原因を探し出してきて、運がよければ対策方法まで調べてくれるのである。「Windows系のオペレーティングシステムも、ソースを公開すればより安定したオペレーティングシステムになるのに」と思っている人間は筆者だけではないはずだ。

非常に簡単なことのように思えるが、Microsoftのような巨大企業であっても、これは成しえていない。むしろ、大きな市場シェアをもつ大企業には成しえないことなのである。理由は簡単だ。オペレーティングシステムやプログラムというリソースを、ユーザーが自由にいじれるようにはしていないからである。一方、Linuxのようなフリーソフトは、ユーザーがリソースを自由にできる（オープンソースは、これを実現する一手段にすぎない）。だから、ユーザーに力と時間があれば、自由にシステムを作り直し、改良することができる。これこそがUNIXの基本哲学なのである。

たとえプログラマーでなくとも、UNIXは自由度が高い。単純なツールが多数揃っていて、ユーザーはそれを組み合わせたり簡単なスクリプトを書いたりして、限られたリソースでもさまざまなことができる。

Windowsなどは、リソースはたくさんあっても公開されていない。だから、メーカーが

想定しているとおりの使い方しかできない。機能を増やすということは、メニューやダイアログボックスのボタンが増えるということであり、組み合わせの自由度が高まることではないのだ。

UNIXはもともと、少数のプログラマーが、自分たちが快適に使えるように作ったシステムだった。万人が簡単に使える（ような気になる）とか、マーケティングリサーチの結果に基づいているのではない。ある意味、非常に自己本位なシステムだが、これが同じような嗜好のユーザープログラマーやコンピュータの研究者たちに受けたのである。UNIXシステムは、UNIXの創造者たちに共感をいただく多数のプログラマーによって、自由に拡張されていった。これを実現するための方法論として、ソースが（一定条件のもとで）公開されていたのである。

UNIXの派生的な改良の成果が、UCB（カリフォルニア大学バークレー校）におけるBSD UNIXである。現在、BSD系UNIXのシ



各種ミニコンピュータ（写真左）
左側がPDP-11、中央奥は沖電気のOKITAC 4300、右側は日立のHITAC 10。

SONYのワークステーション（写真中）
SONYのNEWSシリーズワークステーションは、最初は Station という名称だった。NEWS-800シリーズは、モトローラ社の68020、68030などのCPUを使っていた。

Micro VAX（写真右）
VAXシリーズコンピュータをLSI化し、ワークステーションやサーバとして使えるようにした製品。

Column

ミニコンピュータ

半導体技術の進歩にともない、1970年ごろから、低機能のコンピュータであればかなり小型のものを作れるようになった。ひとことで言ってしまうと、これがミニコンピュータである。

本体の重量はせいぜい数十キログラム程度で、システム一式でも500キログラムないし1トンくらいのものだ。当時の中型コンピュータともなれば、一式で数トン以上、空調なども含めれば10トンを超えただろうから、それからするとやはりミニサイズである。

エアは、System V系のLinuxの影で細々としたものであるが、Linux上のツールやネットワーク機能の多くがBSD UNIXに由来しているという事実は動かしがたい。

AT&TがUNIXの商品価値に気付いてからは、ソースの入手にかなりの費用がかかるようになったが、それでも公開するという原則に変わりはなかった。その後、公開されているUNIXの仕様に基づいて、AT&Tのライセンスに抵触しないフリーのUNIX互換システムが登場したことは、ここで説明するまでもないだろう。LinuxなどのUNIX互換システムは、UNIXという名前こそついていないが、UNIXの基本哲学は本家よりも正しく受け継いでいるのである。

PDPシリーズミニコンピュータ

UNIXの歴史において欠くことができないのが、DEC(Digital Equipment Corporation、Compaqに吸収された)のPDPシリーズのミニコンピュータと、その後継であるVAXシリーズである。

ミニコンピュータというのは、マイクロプ

ロセッサベースのコンピュータの登場以前に使われていた小型コンピュータシステムである。当時、メインフレーム(大型コンピュータ)は使用料金が高価であり、また、巨大でデリケートな機械だったので、研究者やプログラマーなどが完全に自由に使えるものではなかった(専任の保守要員がいて、まさに計算処理のために使われていたのである)。それに対して、安価なミニコンピュータは、個人ないし数人の研究室のレベルで、自由に使うことができた。オペレーティングシステムの開発のために、気軽にクラッシュさせることができたのである。また、ハードウェア仕様なども公開されていたので、必要な周辺機器を作ることもできた。

UNIXは、このようなミニコンピュータ上で、プログラマーが快適に使える環境を構築するために生まれたのである。

1970年代は、半導体のIC化が進み、コンピュータの小型化、低価格化が始まった時代である。各社がミニコンピュータを開発、販売していた。そのなかで、コンピュータサイエンスやプログラミング教育の現場で特に好

計算機内部の配線(写真左、中)
大型のプリント基板を使わず、小さなモジュールを組み合わせた計算機の場合、各モジュールを相互に接続するために、膨大な量の配線が必要だった。

トランジスタ計算機の内部(写真右)
小さなモジュールに、トランジスタやダイオードで作られた論理ゲートが収められている。このようなモジュールを何百個も組み合わせて、コンピュータが作られていた。



Column

スーパーミニコンピュータ

半導体技術が発展し、ミニコンはさらに高性能になった。そして、ミニコンピュータのアーキテクチャを継承し、さらに仮想記憶などの機能を追加して、使いやすさとミニコンのソフトウェア資産を武器に、中型コンピュータの市場に投入されたのが

スーパーミニコンピュータと呼ばれる製品群である。DECのVAXシリーズ、Data GeneralのEclipseシリーズ（これはNovaというミニコンの後継である）などが代表的な製品だった。UNIXは、このクラスのプラットフォームに移行したことで、本格的なマルチユーザーオペレーティングシステムとなったのである。

まれていたのがDECのPDPシリーズだった。

PDPシリーズ、特にPDP-11ファミリは、スタック機構を備え、豊富なアドレッシングモードと使いやすい命令セットにより、教科書に書かれているアルゴリズムや技法を使ったプログラムを作成するのが非常に容易だったのである（もちろんこれは、アセンブリ言語レベルの話だ）。

命令セット

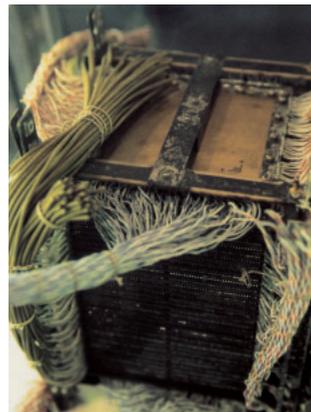
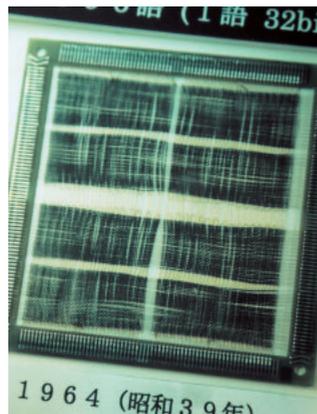
いろいろなCPUのマシンコードを分析したことがあればわかると思うが、命令の種類とアドレッシングモードを豊富に用意し、なおかつ必要なすべての命令でなんの制限もなくアドレッシングモードを自由に組み合わせて使えるような命令セットというものは、なかなか作れるものではない。

命令の語長を長くしたり、命令（オペコード）を可変長にすればさほど難しくはないのだが、たとえばすべての基本命令を16ビットに収める（もちろん、オペランドは別だ）という限られた条件のもとに実現するのは非常

に難しい。嘘だと思ふなら、自分で命令セットを考えてみればいい。16ビットにはまず収まらないだろう。

結局のところ、16ビットに収めようと思ったら、レジスタの数を少なくするとか、汎用レジスタという考え方をやめて、専用レジスタを使うか、あるいはアドレッシングモードの種類を減らすといった選択をすることになる。あるいは、16ビット命令と32ビット命令の両方を用意するなど、命令語長を拡張することにしてしまうかもしれない。

8ビットCPUは、オペコードが8ビットしかなかったため、すべての条件を満たす命令セットなどは作りようがなかった。Motorola社の6800系は、アドレッシングモードが多少充実していたが、レジスタが少ししかなかった。Intelの8080は、レジスタの数は多かったものの、すべて専用レジスタであり、アドレッシングモードという考え方はなかった。その後、Z-80などのように機能が拡張されたプロセッサは、オペコードを長くするという形で、新たな機能を追加するしかなかった。



真空管式計算機の配線（写真左）
真空管式計算機の配線面。真空管はこの裏側にある。この時代は、真空管だけでなく個々の部品も大きかったので、計算機は巨大なものとなった。

コアメモリ（写真中）
半導体メモリが主流になる前は、コアメモリが広く使われていた。フェライト製のビーズのような磁気コアを電線で編んだものである。1つのコアが1ビットの情報を記憶する。写真に写っているものは、4096ビット分のコアメモリである。

コアメモリスタック（写真右）
平面状に編まれたコアメモリを重ねたものがコアスタックだ。このひと塊で16Kバイトのメモリとなる。

16ビットマイクロプロセッサは、ビット数が増えたので、だいぶましになった。Motorolaの68000は、完全にミニコンやスーパーミニコンを意識し、8ビットCPUからの互換性よりも命令セットの洗練を目指した。しかしIntelの8086は、8080との互換性を重視したため、アドレッシングモードは多少充実したものの、ほとんど専用レジスタに近い考え方が残り、すっきりした命令にはならなかった（その後の32ビット化などで、さらに温泉旅館的な拡張が行われたので、命令アーキテクチャとしてはめちゃくちゃになってしまった）。当時、68000と8086の能力差は明らかだったが、市場で勝利を収めたのはインテルだった。

アドレッシングモード

アドレッシングモードは、プログラムがメモリ中のデータにアクセスする方式を指定するものだ。

高級言語であれば、構造体、ポインタ、配列、リスト、FIFO、LIFO、辞書など、さまざまなデータ構造を使うことができるが、こういったデータ構造は、マシンコードプログラムがさまざまな方法でメモリにアクセスすることによって実現されている。

たとえば、高級言語で単純変数にアクセスするという場合であっても、変数がグローバル変数であるか、ローカル変数であるかに応じて、実際に実行するマシンコードレベルのロード/ストア命令は変わってくる。グロー

バル変数であれば、適当なメモリアドレスからデータをロードすることになるが、ローカル変数なら、スタック上からロードしなければならない。構造体のメンバにアクセスする場合は、構造体の先頭アドレスに対するオフセットを指定することになる。また、変数をポインタとして使う場合、配列にアクセスする場合など、メモリアクセスにはさまざまなバリエーションがある。

高級言語を使っていれば、CPUのアーキテクチャなど何も考えずにコーディングができる。こういったマシンコードやアドレッシングモードの違いは、すべてコンパイラが吸収してくれる。しかし、アセンブリ言語を使ってマシンコードレベルのプログラミングを行う場合は、CPUがどのようなアドレッシングモードをサポートしているかによって、使いやすいものと使いにくいものの差が出てくるのだ。実用面では、今さらアセンブラプログラミングという状況はほとんどないので、もはやどうでもいいという話もあるが……。

PDP-11の命令セットとアドレッシングモード

8ビットや16ビットのマイクロプロセッサよりはるか前に発表されたPDP-11は、あとから登場したプロセッサよりもすっきりした命令セットを持っていたとされるが、実際にはどのようになっていたのだろうか？

まず、PDP-11のオペコード構成を見てみよう（図1）。PDP-11は基本的に2オペランドの命令体系である。2オペランドというの

IMSAI（写真左）

IMSAIのビデオディスプレイモデル。IMSAIは、S-100という標準バスを使って8ビットCPUを中心にしたシステムを組むことができた。標準的なIMSAIは、ミニコンタイプのもだったが、この製品は、ビデオディスプレイ、キーボード、フロッピーディスクを備えた端末型の構成である。



Intelの歴代プロセッサ（写真中）

4ビットCPUの4004以来、Intelは数多くのプロセッサを世に送り出してきた。



ディスク駆動装置（写真右）

大型コンピュータなどに使われていたリムーバブルディスク装置。上部のカバーを開けて、ディスクを脱着する。



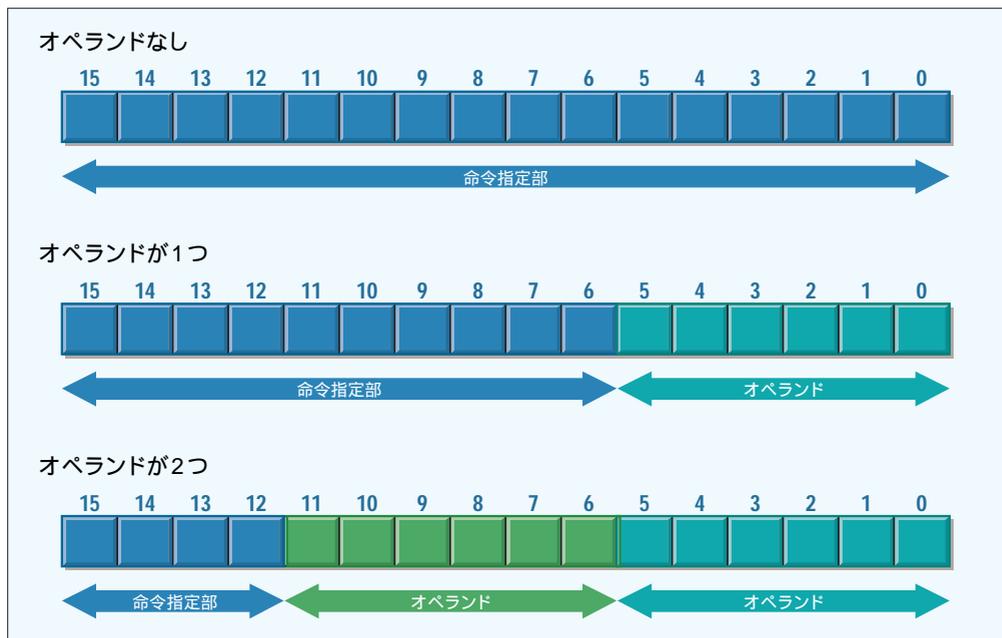


図1 PDP-11の命令

は、演算処理などを行う際に、2つのオペランドで指定された値を使用し、結果をどちらかのオペランドで指定された領域（レジスタやメモリ）に格納するという体系である。たとえば、高級言語で記述した「 $c = a + b$; 」といった代入演算を行う場合を考えてみるとわかりやすい。2オペランドのプロセッサは1つの命令で3つの変数を扱うことができないので、次のように処理される。

```
c = a;
c + = b;
```

もちろん、命令の種類によっては、オペランドが1つのもの、オペランドがないものもある。しかし、加減算や論理演算などの2項演算、データを転送する命令などは、オペランドを2つ使う。インクリメント/デクリメント演算などの単項演算であれば、オペランドが1つでいい。制御命令などはオペランドは不要だ。ただし、一部の命令はこれらとは異なるビットフィールド割り当てのものもある。

命令中のオペランド部は、6ビットで表現される。PDP-11のオペコードは16ビットな

ので、オペランドを2つ指定すると命令指定部は4ビットとなり、2オペランド命令は最大でも16しか定義できないことになる。2項演算命令やデータの移動命令はこのような構成になっている。しかし1オペランド命令は、命令指定部として10ビットを使用できる（上位4ビットは、2オペランド命令に割り当てられていない値を使う）。オペランドのない命令なら、16ビットすべてを命令指定部にできる。これらをうまく組み合わせ、16ビットという限られたビット数で、いろいろな命令を定義しているのである。

オペランド指定の下位3ビットは、使用する汎用レジスタ（R0からR7）を指定する。上位3ビットは、どのようなアドレッシングモードを使うかを指定する（表1）。上位3ビットのうち下位1ビットは、間接指定モードを指定する。たとえば“000”なら、レジスタが収めている内容は直接オペランドとして処理されるが、“001”ならレジスタの内容で示されるアドレスの内容がオペランドとして処理される。

PDP-11の秘密

PDP-11は、1970年代初期という時代の設

計でありながら、いまだに伝説的に語られている豊富な機能とすっきりしている命令セットを実現している。これには、ちょっとした秘密がある。聞いてしまえばなんだというアイデアなのだが、考えれば考えるほどすばらしいのだ（他社がこのアイデアを使えなかったのは、特許のためらしい）。

PDP-11には、R0からR7の8個の汎用レジスタがあるが、R7は汎用レジスタであると同時に、プログラムカウンタになっているのである（ちなみに、R6はシステムスタックポインタである）。これはすごいアイデアだ。R7を対象とするレジスタへのロード命令は、ジャンプ命令になるのだ。R7に対して加減算を行えば、相対ジャンプ命令である。もち

ろん、R7をインデックスレジスタとして使うこともできる。

最後に

現在のコンピュータの設計では、命令セットをきれいにするとか、豊富なアドレッシングモードを提供するといったことに、エネルギーはつぎ込まれていない。いかにプログラムの実行を高速にするかということだけである。命令セットがすっきりしているかどうかとか、どんなアドレッシングモードを使うかといったことは、コンパイラ屋さんが知っていればいいのである。だが、たまには昔に思いを馳せて、古いコンピュータを思い出すのも悪くないのではないか。

上位3ビット	動作	説明
0 0 0	Rx	レジスタを直接指定する
0 1 0	* Rx++	レジスタをポインタとして使用し、あとでインクリメントする(ポストインクリメント)
1 0 0	* --Rx	先にデクリメントしてから、レジスタをポインタとして使う(プレデクリメント)
1 1 0	Rx->disp	インデックス指定
0 0 1	* Rx	レジスタをポインタとして使う
0 1 1	* * Rx++	ポストインクリメントを行う二重ポインタ
1 0 1	* * --Rx	プレデクリメントを行う二重ポインタ
1 1 1	*(Rx->disp)	ポインタをインデックスで指定

表1 アドレッシングモード
下位3ビットはレジスタを指定する。

Column

Branch and Linkとスタック

プログラミングをかじっている読者であれば、スタックというデータ構造がどのようなものか知っているだろう。現在ではごく当たり前に使っているスタックだが、70年代のころは、どのコンピュータでも当たり前というわけではなかった。

プログラミングの教科書を読むと、わりと最初のあたりで、サブルーチンという考え方が出てくる。サブルーチンの呼び出しで、引数と実行結果の受け渡しにスタックを使うのは、プログラマーの基礎知識である。

当時、スタックを持たないアーキテクチャのコンピュータでは、サブルーチンの呼び出しを行うために、Branch and Linkという命令（BAL命令）を使った。これは、指定したアドレスに分岐する命令なのだが（Branchは、指定したアドレスにジャンプする命令）、その際に、PC（プログラムカウンタ）

の内容を、適当な領域に保存するのである。PCは、次に実行する命令のアドレスを収めている。この内容をどうにかして保存しておけば、あとからBAL命令に続く部分の実行を再開できるのである。具体的には、BAL命令で保存した内容をPCにロードする間接ジャンプ命令を実行すればいいのだ。

一般的なBAL命令は、PCの内容を汎用レジスタかメモリ中の特定の領域に保存しておいてリターンするとき使うようになっている。

この方式の欠点は、サブルーチンの再帰呼び出しができないということである。リターンアドレスは、毎回同じところに保存される。再帰呼び出しを行うと、最後のリターンアドレスしか保存されないのである。このようなアーキテクチャで再帰呼び出しをしたり、スタックを使う処理（引数の受け渡しやローカル変数などの利用）をしようと思ったら、自分でスタックデータ構造をソフトウェア的に実現しなければならなかった。

隠喩としてのLinuxコミュニティ

「我働く、故に我あり」の日本における系譜

文：豊福 剛
Text : Tsuyoshi Toyofuku
illustration : hnm

『リナックスの革命』（河出書房新社）には、ポスト資本主義の精神というものがどのようなものであり、それとハッカー倫理とはどのような関係にあるのか、その考察を期待したのだが、期待外れの内容だった。ハッカー倫理をウェーバーが指摘した資本主義の精神と対比させて特徴づけようとする意図は興味深いのだけれども、資本主義の精神とプロテスタンティズムの倫理を、同じものであると誤解しているように思われる記述が散見されるのは気になった。ハッカー倫理の特徴をいくつかの側面から記述し、それらを手際良くまとめてある点は評価できるのだけれども、ポスト資本主義がどういうものなのかについての考察がごっそり抜け落ちているため、ハッカー倫理の延長でポスト資本主義の精神を示そうとする試みは、残念ながら失敗に終わっているように思う。

意地悪な読み方をすれば、本書は、自分が楽しめることを徹底的に追求しよう、その成果を公開してみんなと共有できるようにすれば、もっと楽しくなれるだろう、ということを行っているだけのように思える。そうすると、いわゆる自己啓発本の類と同じなのではないか。

ハッカー倫理と動機づけ

ところで、ハッカー倫理とは、まず何よりも、プログラミングに対する動機づけであると解釈できるだろう。それでは、そもそも動機づけとは何なのか。

本書のプロローグで、リーナス・トーヴァルズは、人間の行動の動機づけには、生き残り、社会生活、娯楽、この3つのカテゴリーがある（リーナスの法則）と言っている。Appleの創始者の一人であるスティーブ・ウォズニアックも「人は幸福になることを目的に行動する。そして幸福は、食べもの、楽しみ、友だちがそろったところにある」という考えを披露している。本書の著者ペッカ・ヒマネンが指摘するように、このウォズニアックの幸福観は、リーナスの法則に対応づけられる。生き残り = 食べもの、社会生活 = 友だち、娯楽 = 楽しみ、というわけだ。

こうした考え方は少しも珍しいものではない。ヒマネンが紹介しているように、心理学者のエイブラハム・マズローの5段階欲求説によれば、動機づけの最低段階には生理的欲求があり、次に安心感を求める欲求（第2段階）、社会的な帰属や愛情を求める欲求（第3段階）、社会的評価

を求める欲求（第4段階）と続き、自己実現（第5段階）が最高の動機とされる。これらの5段階は、自己実現を頂点とするピラミッド構造をなすと考えられている。

このような動機づけや欲求は、たしかに自分のものであるように思われる。しかし、どこまでほんとうに自分のものといえるだろうか。社会生活を求め、コミュニケーションを求めるとき、ほんとうに語るべき何かがあってそうしているだろうか。ほんとうにそれを楽しんでいるだろうか。むしろ、つかの間の暇つぶしのために、空疎なお喋りを楽しんでいるのではないか。

人間は、絶食を欲することもあるし、極端な場合には、自らの命を自らの手で絶つことを欲することもある。他人との一切の関わりを絶ち、あえて孤独を欲することもある。このような欲求は、例外的な状況における例外的な欲求といえるだろう。それでは、同じように、例外的でない欲求についても、ある状況に依存しているのではないのだろうか。

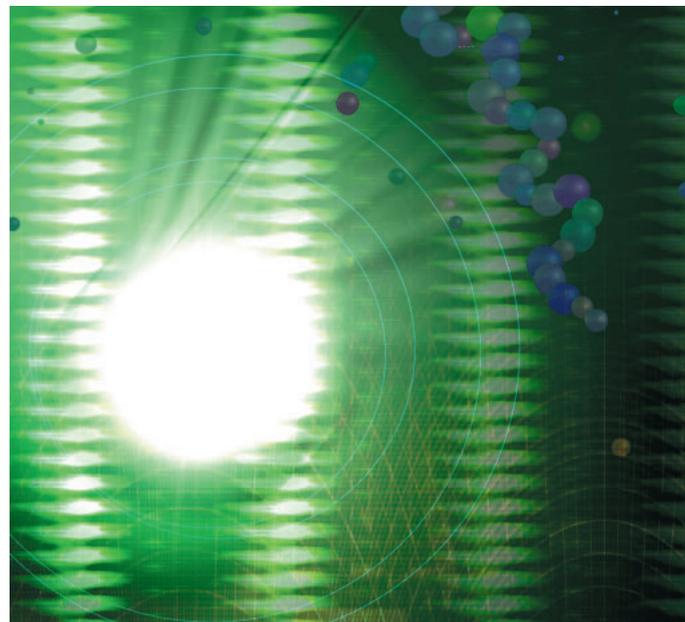
この考えを拡張していくと、人間は自らの欲求に従って行動しているように思っているが、実は、歴史的に形成された制度、文化、宗教、社会関係といった諸々の構造によって動かされているのではないか、という発想に行きつく。

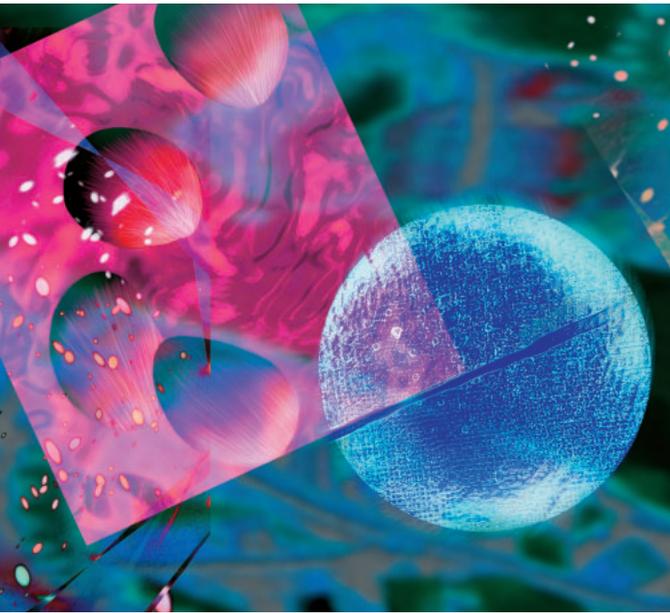
マックス・ウェーバーの説によれば、資本主義以前における職業観の背景には、宗教的な規範があった。プロテスタントイズムにおいては、職業は神から任命された天職（Beruf）であり、それをまっとうすることが神への信仰の証しであるとされた。また、労働の結果として獲得された営利についても、神の意志に添うものとして正当化された。この段階においては、労働は神への信仰という目的に対する手段だった。しかし、資本主義が誕生した背景に、そのような労働観があったのだとしても、いったん資本主義が成立したあとにおいては、もはや、そうした目的は失われてしまい、営利の獲得そのものが目的になってしまう。

日本資本主義の精神について

ところで、資本主義の精神を準備した宗教的な倫理規範の存在は、西欧におけるプロテスタントイズムだけに限定されるわけではないのかもしれない。山本七平は『日本資本主義の精神』で、徳川時代の鈴木正三と石田梅岩の思想に、労働に対する日本独得の宗教的な倫理規範を求めている。

鈴木正三は「世俗の業務は、宗教的修行であり、それを





一心不乱に行なえば成仏できる」という労働観を唱えた。つまり、キリスト教の神への信仰に対応するのが、仏法にしたがって生きること、仏行にはげむことであったというわけだ。それだけでなく、商人が一心不乱に働いた結果として生じる利潤についても、けして否定しなかった。鈴木正三の思想的背景が仏教（禅宗）にあったのに対して、石田梅岩の思想的背景は儒教（朱子学）にあるが、両者の世界観・労働観はほぼ同じであり、表現が違っていたただけだと山本七平は論じている。

徳川時代に鈴木正三や石田梅岩という偉い思想家がいて、彼らの思想がそこそこの影響力をもっていたことは確かだとしても、彼らの思想が日本独得の「資本主義の精神」を生み出したわけではないだろう。そうではなく、その時代に新しく台頭しつつあった経済勢力を正当化する思想として受け入れられたのだろう。そして、もともとは思想や宗教と一体であったはずの行動様式は、いったん正当化され、人々の間に浸透していく中で、その行動様式だけが継承され、それと一体であったはずの思想や宗教は、都合よく忘れられてしまうように思う。

プロテスタンティズムの背景から誕生した資本主義と、儒教や仏教を背景とした職業倫理から誕生した資本主義とでは、資本主義のメカニズムそのものには違いがないとしても、そこで働く人々の行動様式や文化には、大きな違いがある。

日本の資本主義を考えるうえで、いまさら指摘するまでもなく、多くの日本人にとって会社は一種独特の存在である。日本の会社は、機能集団であるだけでなく、同時に共同体として機能してきた、と山本七平は論じている。ここで、山本七平がいう共同体を、コミュニティと言い換えても、その意味するところは、ほとんど同じだといえるだろう。

これに対して欧米はどうかというと、共同体というのは地縁や血縁をもとに形成されるコミュニティであって、会社はあくまで機能集団にすぎず、会社が同時に共同体に転化するといったことは、まずありえない。もちろん、欧米の会社においても、多少は共同体的な側面はあるように思うのだが、しかし、日本の会社のように、共同体的な関係性に全面的に関わることを強いるほどのものではないだろう。

しかし、山本七平の説くような機能集団 = 共同体として

の日本の会社のあり方は、1990年代以降、大きな転換期にあることは言うまでもない。かつて日本的経営の柱であった、正社員に対する終身雇用制と年功序列制は、もはや制度的に維持できない状況に置かれている。そして、機能集団 = 共同体としての日本の会社のあり方が、戦後の復興期 ~ 高度成長期に形成されたものであったとしても、その背景にある職業観が、山本七平が説くように徳川時代に形成されて以来ずっと継承されてきたものだとすると、1990年代以降の日本が迎えている転換期においては、徳川時代以来の伝統的システムをどのようにして乗り越えられるかが問われているように思われる。

ハッカーのコミュニティは機能集団である

たとえば、機能集団 = 共同体である日本の会社においては、「実際に仕事がなくとも忙しく振る舞っていないと、本人が心理的に不安になるという状態を引き起し、また、共同体から疎外されているという不安を感じ、そのために、否応なく多忙を演じる」といった状況が、けして珍しくないように思う。こうした状況は、本人にとっても、まわりにとっても、けして幸福なものとは思えないのだが、忙しく振る舞っていれば他人に認めてもらえる、暇にしていると他人に認めてもらえない、という心理が働いているのだろう。こうした心情は、これはこれで、リーナスの法則という社会生活にもとづく動機づけなのだろう。

興味深いのは、ハッカーのコミュニティというのは、同じコミュニティといっても、日本的な共同体（村社会といってもいいだろう）とは対極にある、徹底した機能集団であり、それこそ能力主義で評価される社会であるように思える点だ。そこからは、会社が共同体として機能するとき日本人が感じていたような精神的な充足感は、少しも得られないように思う。そもそも、インターネットでの共同作業というのは、顔を突き合わせながら一緒に働き、腹をわって本音でとことん話し合う、といったようなベタな日本の社会生活とは、徹底して無縁である。

たぶん、この先日本人は、機能集団を共同体に転化するだけで、会社以外の自立的な共同体を形成することを棚上げしてきたツケを、長い年月をかけて穴埋めしていかなければならないようだ。そして、もしこの穴埋めに失敗したとしたら、不幸な歴史が繰り返されることになるのだろう。

Profile

とよふく つよし

1962年東京生まれ。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。

あるOSの死

文：安田幸弘
Text: Yukihiko Yasuda

10年という時間

Linuxが誕生したのは、今からちょうど10年前だった。そしてちょうどその頃、シリコンバレーの一角にBe, Inc.というパソコンメーカーが活動を開始した。

それから10年。Linuxはマイクロソフトを脅かすほどのシステムに成長した反面、同時期にスタートしたBeは苦戦を続け、ついに今年の8月には会社の清算を宣言、Beの資産はPalmに売却されることになった。Beの資産を買ったPalmが、BeOSをどうするつもりなのかはわからないが、おそらくBeが持っていたマルチメディアやマルチタスクの技術がPalm OSに組み込まれることになるのだろう。現在のところデスクトップOSとしてのBeOSは、Be発足から10年にあたる2000年の夏にリリースされた、バージョン5を最後に消滅することになっている。

猛烈なスピードで変化するコンピュータの世界で、10年という時間は長い。湾岸戦争が起き、ソ連が消滅し、バブル崩壊から日本経済の墜落が始まった10年前は、まだDOSの時代だ。その頃、マイクロソフトは世界中のパソコンユーザーの期待を一身に集めて、Win32 APIをベースとする新しいOSの開発を始め、それに対してIBMとアップルはPINKと呼ばれる高性能OSの開発を宣言した。ずいぶん昔の話のような気がするけれど、ちょうど10年前だった。

この間、PINKは途中で解体、IBMがマイクロソフトから引き取り、磨きをかけて市場に投入したOS/2 Warpは、Windows 95 / NTよりはるかによくできたOSだったのに、結局市場の獲得に失敗してメンテナンスモードに入ってしまった。個人的にはBeOSは非常に好きなOSだっただけに、BeOSの消滅はとても残念なニュースだが、その反面、よくここまで持ちこたえたもんだ、なんて思ったりもする。

Beの軌跡

BeOSは優秀なOSだったと思う。デュアルCPUを前提に、マイクロカーネル的な手法で開発されたマルチスレッド、マルチタスクのカーネルは効率が高く、マルチメディアのような重いアプリケーションも軽快に実行することができ、しかも実に安定していた。

だがコンピュータの世界では、良い製品が売れるとは限らない。IBMのOS/2 Warpにしても、ビジネス的には失敗だった。極論すれば、コンピュータのユーザーにとって、OSの技術やポテンシャルなんてどうでもいいのである。目の前のアプリケーションがサクサク動けば、ユーザーは幸せなのだ。技術的には平凡なUNIXもどきでしかないLinuxの成功にしても、結局、目の前のUNIXソフトウェアがサクサク動き、しかも岩のように安定していたという事実が最大のポイントだったのだろう。Beにとっての不幸は、動かすべきアプリケーションがなかったことだ。

熱狂的なBeOSユーザーを中心に、「BeOSのメンテナンスを続けてくれ」という署名運動などが起きているが、今回のBeの売却は、BeOS、あるいはBeIAというプロダクトの売却ではなく、Beが持つ技術の売却であるということがユーザーには悲しい。つまり、秋葉原のジャンク店でよく見かける「部品取り用」マシンのように、BeOS / BeIAは部品取り用システムとして売られてしまったわけだ。将来のPalmがBeの技術を生かした高性能なマルチメディア携帯端末をリリースするとしても、少なくとも自分のデスクトップでBeOSを動かすという目はなくなってしまった。

Beはスタート以来、迷走を繰り返してきた。もともと、BeBoxと呼ばれる2つのPowerPCを搭載する専用ハードウェアとOSがセットになったシステムとして計画されていたのだが、ハードウェア部門の収益が見込めないと、同じPowerPC

チップを搭載するPowerMac用のOS専業に転身したものの、サードパーティの参入が低調だった。その後、デベロッパーを呼び込むために、開発環境を含むIntelマシンでも動作するバージョンをリリースしたものの、BeOS用のアプリケーションは増えず、周辺機器への対応も進まなかった。

Linuxをはじめとする高性能な無償のオープンソースOSが普及したのも、Beの誤算だったようだ。当初Beは「Linuxはサーバ用OS、ウチはデスクトップのOS」と、棲み分けが可能だと強調していた。確かにビジネスユーザーはLinuxをサーバ用OSだと考えたかもしれない。しかし、BeOSのユーザー層と重なる個人のマニア層は、やはり少からず自分のハードディスクにLinuxのパーティションを持っていたはずだ。結局、OS無償化の波には逆らえず、Beもデスクトップ用のBeOSを無料配付にせざるを得ず、その代わりにインターネット端末用のBeIAに活路を求めることになったのだろう。BeIAは、コンパックやソニーなどへの売り込みに成功したものの、そもそもインターネット端末そのものが絶不調という状態で、とうとう進退極まったらしい。

BeOSが、生き残りの方策を求めて試行錯誤を繰り返している間、ユーザーの間からは何度もオープンソース化を求める声が上がった。おそらくBeもオープンソース化で収益をあげる方策について検討はしたはずだが、売却は以前からオプションのひとつだったらしい。オープンソースでビジネスを建て直すというのであればともかく、売却ということを考えると、オープンソース化は難しかったのだろう。主力商品がオープンソースのソフトウェアでは、売却は困難だったに違いない。

「知的財産」は誰のものなのか

赤字続きだったBeの債権者にしてみれば、売却で少しでも投資を回収したいと思うのは無理もないが、BeOSのファンにしてみれば、ハシゴを外さ

れたような気分だ。ぼく自身は、BeOSに大した貢献をすることができなかったが、それでもBeがこれまで事業を継続できたのも、BeOSを愛する世界中のGeekたちがいたからじゃないか、と言いたい。Beに投資した投資家にしても、Geekのためのシステムに投資したのではなかったのかと尋ねたい。

PalmがBeの技術をどのような形で製品に組み込んでいくのか、あるいは特別なメンテナンスを行うつもりがあるのかどうか、今はまだわからない。しかし、OSというのは、一般のアプリケーションとは全く異なるソフトウェアだ。大げさに言えば社会的責任ってものだって発生する。だからこそIBMは、現在もOS/2 Warpのメンテナンスを続けているのだらうし、マイクロソフトは市場の独占に対する審判を受けなければならなかった。

OSを規制しろなんてことになれば、それはそれで問題かもしれないが、少なくともOSを売るビジネスには、OSは経営者が勝手に売り買いできるモノとは違って、ユーザーに対する責任や、社会的な責任があることを自覚してもらいたい。それじゃなければ、安心してシステムを使うことなんてできやしない。

もし、どうしても事業の継続ができないのであれば、オープンソース化して公共のメンテナンスに期待するという方法を取るべきではなかったのだろうか。あるいは、ユーザーや各方面からの寄付を募って、BeOSをオープンソース化し、メンテナンスを続けるための財団を作るといったこともできたのではなかったのだろうか。

GNUが主張する、ソフトウェアは人類の共通財産だ、という言葉の意味を改めて考えてみたいものである。

Profile

やすだ ゆきひろ

生業はテクニカルライター。原稿書きのかたわら、(株)市民電子情報網のボランティア社長兼技術スタッフとして、NGO向けプロバイダのネットワーク運営に携わる。

ドクターShiodaの

ギョーカイ SnapShot

PC業界今月の話題

文：塩田紳二
Text：Shinji Shioda

- 7・19 Code Red 猛威をふるう
- 7・27 Palm、OSライセンスを子会社化
- 7・30 マイクロソフトWindows XP日本語版の発売日を11月16日と発表
- 8・7 マイクロソフト、米最高裁判所に上告
- 8・12 IBM PC発売20周年

今年、インテルのデベロッパーフォーラム（IDF）とLinuxWorldがばっちり重なってしまいました。昨年は、1週間ずれていたの、お盆の最中に米国に2週間取材というスケジュールでしたが、今回は、もっと辛くて、サンフランシスコで開催されるLinuxWorldとサンノゼのIDFを両方取材せねばなりません。ここで問題なのは、サンフランシスコとサンノゼって、車で行くか、電車（カルトレイン）で行くかぐらいしか方法がなくて、米国は、電車が結構不便なので、車で往復しなくちゃいけないこと。実は、ほとんどペーパードライバーなんです。いちおう、車もっているんですが、女房を運転手にして「やってちょうだい。パーカー」って感じで助手席専門なもので。去年1回、米国で車借りて運転したんだけどね。地理が全然わかんなくて大変でした。大丈夫かなあ。

Code Redの恐怖

7月に登場したCode Redワームは、マイクロソフトのHTTPサーバであるInternet Information Server（IIS）をターゲットにしたもの。さらにバックドアを残す変種なんかも登場して、かなり猛威をふるったよう。多くの管理者がセキュリティパッチを当てていなかったのが一番の原因。そんなサーバ使わなければいいとは思いますが、世の中には、さまざまな事情で、自分の意にそぐわないソフトウェアを使わされる管理者も多いもの。やはり、セキュリティパッチは、丹念に当てていかないとひどい目に遭うという教訓のサンプルみたいな事件。

しかも、今回、マイクロソフト系の電子メールサービスHotMailのサーバもCode Redに感染したらしい。なんだよ、自分のところもかあ。なんだか心配だねえ。

日本でも一部のプロバイダがCode Redが原因でサービスが停止したらしい。また、個人ユーザーでもADSLなどの定額接続、常時接続のサイトでやられたところがあるのだとか。なにせ、Windows NTや2000のサーバ版をインストールすると自動的にIISがインストールされちゃうので、動いているのを認識していない人もいたのだとか。

筆者のところは、Windows 2000、Windows NTサーバともにあるけれども、インターネット側からはHTTPではアクセスできないようになっているので、被害はなし。でも、なんかネットワークが混んでたって感じにはなった。OCNとかの常時接続サービスを使っているユーザーの中には、セキュリティが甘いところが少なくないという話なので、それでやられたところが多かったかも。そういえば、マイクロソフトからのニュースレターにも、Code Redに対する警告が結構入っていたようだけど、みんな見てなかったのかな？

みんなが同じシステムを使って、1つのバイナリが共有できて便利のだけど、こういう事態も起きうるってことだ。はたしてドットNETは大丈夫なのかな？ 便利になるのはいいけど、またワームが出て、被害続出してことになると、サーバ防衛、最良の方法は、マイクロソフトのサーバを使わないこと、なんてなったりしてね。

裁判も進捗あり

さて、控訴審で差し戻しが言い渡されたマイクロソフト対司法省の裁判だが、マイクロソフトは結果を不服として最高裁に上告した。地裁で再審理となったのだが、マイクロソフトが独禁法違反という点については認定されてしまったからである。しかし、この上

告は、棄却される見通しだという。というのは、控訴審の裁判官全員一致の結果として、これ以上この件を議論する余地がないからだという。

つまり、この裁判は、マイクロソフトは有罪、だけでも2分割案は不相当という答えなのである。なので、差し戻されても、議論されるのは是正措置をどうするかということであって、マイクロソフトが違反しているかどうかではないのである。マイクロソフトの上告は、時間稼ぎだという観測も多い。というのは、もうWindows XPの出荷が迫っており、出荷前に地裁で再審理が始まってしまうと、XPの出荷に影響が出る可能性があるからだ(たとえば、仮処分申請というやり方で、XPの出荷を一時的に差し止める)。

そのXPだが、日本でも発表が行われ、出荷は11月16日になるという。米国での発売予定が10月25日なので、約3週間という時差である。OS自体は、シングルバイナリといいつつも、実際には附属ツールなどに日本語化が必要だし、なにより米国の最終マスターをベースに作業が行われる関係上、どうしても時差がついてしまうのである。まあ、昔は、1年も差があったり、日本では発売されないバージョンなんてのもあったのに比べると、時差は縮まったとはいえるが、これでシングルバイナリといえるのかねえ？

こうして登場しているXPがすでにいくつの火種を生んでいる。まずは、AOLとのデスクトップアイコンを巡る争い。XPのデスクトップに何を置くのかは、原則ライセンスを受けるメーカーの自由になったのだが、AOLがアクセスのためのアイコンを置いてもらおうと、メーカー各社と交渉していたところ、マイクロソフトは、なにもアイコンを置かないか、あるいはIE、MSN

などのアイコンを置くかのどちらかを選ぶようにOEMメーカーに要望したらしい。つまり、AOLのアイコンを置きたいのならMSNなどのアイコンも置かねばならないのである。

もともとAOLはマイクロソフトと対決姿勢にある。同社のMSNなどが競合しているからである。それで、デスクトップにMSNのアイコンだけが置かれることを警戒しているのだ。

Netscapeを買収したAOLは、企業規模などを考えると、ネットワーク対応を進めていくマイクロソフトとはいずれ本格的に激突する可能性がある。さて、これからどうなるでしょうかね。

インテルは、新I/Oアーキテクチャ

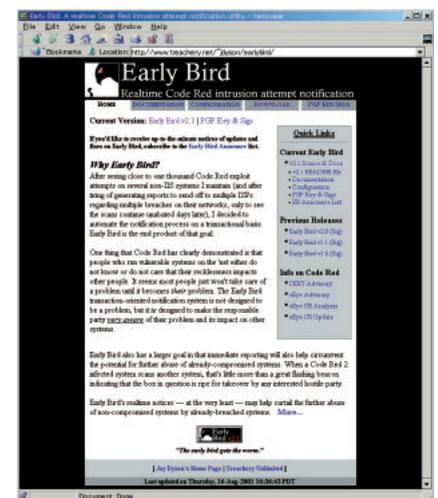
インテルは、8月末のIDFで、新しいI/OアーキテクチャであるArapahoe(アラパホーと読む)を発表する予定だが、現在広く使われているPCIバスを管理しているPCI-SIGは、このArapahoeを支持することにした。このArapahoe今年2月末のIDFで計画が公開されたものの、詳細は8月に発表と、ほとんど実体なしで発表された。そんなに急いだのは、AMDがHyperTransportというI/Oアーキテクチャを広めるべく動き始めたからである。

AMDは、TransmetaやNVIDIA、Sun Microsystems、Apple、Cisco Systemsなどを集め、今年7月にはHyperTransportテクノロジーコンソーシアムを設立。これに対抗してインテルは8月にArapahoeの開発で、Compaq、Dell、IBM、Microsoft、そしてPCI-SIGと協力関係を結ぶ。つまり困り込みが始まったわけである。とはいえ、HyperTransportを使ってもPCIとの互換性はあり、アプリケーション側は、従来のPCIバスと同様にデ

バイスを扱うことができる。ただ、PCIはもともとインテルが開発し、これを引き受けるかたちでPCI-SIGが結成されたわけで、インテルの管理下にあるといってもよく、それでArapahoe陣営に名を連ねたわけである。

ただし、よく見ると、どちらにもチップセットメーカーが入っていない。チップセットメーカーは、これを取引材料に今後のインテルとのライセンス契約などを有利にしたいと思っているため、簡単には話に乗らないのである。AMD、インテルともにこれらのメーカーとは交渉中と思われるが、実際のところ、AMDはインテルとの交渉の「ダシ」に使われるという可能性もある。

さて、互換CPUから出発したAMDとインテルの争いだが、ここに来て、アーキテクチャがらみの競争になってきた。これが本格的になってくると、いままで単一だったPCアーキテクチャ、つまりIBM PC/ATをベースにしたPCという世界も変わってくるのかもしれない。そういえば、今年は、最初のIBM PCが出荷されて20年目。そろそろ、アーキテクチャとしての終焉を迎えつつあるのかもしれない。



Code RedはApacheには無害だが、1日に数百回の攻撃を受けることも珍しくない。感染したIISのドメイン管理者に警告メールを送るプログラムもある。
<http://www.treachery.net/~jdjyson/earlybird/>

初級Linuxer養成講座

エイリアスの使い方

シェルのコマンドラインに慣れてくると、毎回同じようなコマンドを入力するのが面倒になることがある。このようなときに、長いコマンド名に短い別名をつけたり、コマンドのオプションを省略できるようにする「エイリアス」という機能がある。一見難しそうな機能だが、意外と簡単で便利なものなので試してみよう。

文：竹田善太郎
Text：Zentarō Takeda

水不足が騒がれるほど雨が少ない今年の首都圏だが、今現在、外は台風11号の豪雨が降っている。このようなときには車で外出するのも大変だが、雨の日の運転でなによりもうっとうしいのが、フロントガラスの油膜だろう。カー用品店には棚いっぱいいろいろなウォッシャー液が売られているのだが、これといった製品に行き当たったことがないのは、単に運が悪いからなのだろうか？「撥水コート」も試してみたことがあるが、ワイパーを動かしたときに微小な水滴がガラス面いっぱいに広がって視界を遮り、夜の市街地ではかえって危険なので、すぐに剥がしてしまった（撥水コートをきれいに剥がすのは大変な作業である）。

ウォッシャー液といっても、実体は単なるアルコール入り石けん液のはずなので、別に水道水 + アルコール + 中性洗剤でも問題はないと思うのだが、塗装を痛めるとかノズルが詰まるなどといった脅し文句に乗せられて、ついつい市販のウォッシャー液を買ってしまうのだ。しかし、「油膜をとる」という宣伝文句の割には、効果がうすいものばかり

だ。効果がうすいどころか、ウォッシャー液をかければかけるほど、かえって油膜がひどくなるものすらある。かといって、水道水 + 中性洗剤の自家製ウォッシャー液でも、やはり同じような状態になるので、こんなものかとあきらめていた。

ある日、山間部にドライブに行った際、走行中にウォッシャー液が切れてしまった。そこで、緊急用ということで沢の水をウォッシャー液代わりにタンクに入れたことがある。ところが、この天然ウォッシャー液の具合が、とても良かったのである。油膜がかかりそうになったときでも、わずかな量を噴射するだけで、視界がすっきりするのである。晴天時などに汚れを落とすときに噴射しても、市販のウォッシャー液よりもはるかに汚れ落ちが良い。それではと、洗剤を入れない水道水を試してみると、こちらはまったくだめである。噴射すればするほど油膜がついてしまう。

フロントガラスの「油膜」は、いわゆるあぶらではなく、水道水の中にも含まれているような物質（たとえば、カルシウムなどのミネラル成分）が原因なのかもしれない。そう思って、今

度は、スーパーで安く売られていた、ペットボトル入りの「天然水」を試してみた。これが大当たりで、沢水と同じように、とても汚れ落ちの良い高性能ウォッシャー液になったのである。ご存じの人も多いだろうが、日本で売られている「天然水」や「ミネラルウォーター」と呼ばれているものは、ほとんどが、ミネラル成分の少ない「軟水」である。このような軟水なら、ガラス面に着くミネラル分も少なく、また、着いてしまったミネラル分を洗い流す能力も高いのだろう。

こんなことがあってから、ウォッシャー液には市販の「天然水」を使い続けている。人間様が飲む水を、ウォッシャー液などに使うとはなんともばちあたりな気もするが、「天然水」は安売りなら2リッターで110円前後で買えるのに対して、カー用品店の「ウォッシャー液」は数百円もする。「天然水」のほうが性能の面でも価格の面でも勝っているのである。洗剤を使わなければ環境にも良いし、噴射する量も少なくてすむのだから、あながちむだづかいでもないだろう。車に常備しておけば、のどが渇いたときに人間が飲むこともできるし、きれいな水なので、

ラジエーター液やバッテリー液の緊急補充用にも使える。

ウォッシャー液にする場合、気温が氷点下になる冬にどうするかが問題だが、薬局で売っている消毒用アルコールを少量混ぜれば問題ないだろう。フロントガラスの油膜でお悩みなら、ぜひ一度試してみしてほしい。

面倒なコマンド入力を省略したい

シェルのコマンドラインを日常的に使うようになると、同じようなコマンドやコマンドのオプションを繰り返して使うようになってくる。たとえば、lsコマンドを使ってファイルの一覧を見る場合、どんなファイルがあるのかを大まかに知りたいような場合には、

```
$ ls
```

のように、何のオプションも付けませんが、ファイルの日付やサイズなどの詳細な情報を知りたいときには、

```
$ ls -l
```

のように、-lオプションを付けることが多い。また、シェルやアプリケーションなどの設定ファイルは.emacs

や.bashrcのように、ファイル名の先頭が「.」(ピリオド)で始まるファイルになっていることが多いが、このようなファイルをlsコマンドで表示させたい場合には、-aオプションが必要になる(画面1)。

さらに、lsコマンドには、ファイルの種類、すなわち、ディレクトリなのか、単なるファイルなのか、「リンク」なのか、あるいは実行可能なコマンドファイルなのかを区別して表示する-Fオプションがある(画面2)。

lsコマンドを使うときは、これらのオプションをいろいろと組み合わせて使うことになるわけだが、これをいちいちキーボードから入力するのは、ちよつと面倒に感じるかもしれない。たとえば、ディレクトリ中のすべてのファイルを、ファイルの種別の印をつけて表示させたい場合には、

```
$ ls -aF
```

と入力することになる。さらに、ワイルドカードを指定して、表示するファイルの範囲を狭くしたい場合なら、

```
$ ls -aF .ba*
```

のように、コマンドラインはどんどん

長くなってゆく。

lsコマンドのような単純なコマンドであっても、使う人それぞれに特有のくせのようなものがあるはずだ。たとえば、いつも-aオプションを付けないと落ち着かない人がいれば、-Fオプションなど使ったこともないという人もいるだろう。

ちょっと話は脱線するが、打ち合わせなどの場で、他人がパソコンを操作しているのを間近に見る機会があると、それぞれが個性的な流儀のようなものを持っているのがわかっておもしろい。今触れているlsコマンドの使い方もそうだし、Linuxに限らず、Windowsのメニュー操作やショートカットキーの使い方にしても、いろいろな癖を目にする。また、そうやって他人が操作しているのを見て、新たな技を覚えることも多い。筆者自身も、自分では気づいていなくても、他人が見たらおかしく思うような癖を持っていることだろう。

話を元に戻すと、lsコマンドのオプションのように、常用するコマンド形式がある場合、毎回毎回オプションを入力するのは無駄に思える。このようなときに、オプションを省略できるようにする方法はいくつかある。

たとえば、次のような内容のテキスト

```

K ~ term
[zen-t@zen06 tvmaster]$ ls
Desktop          bin                mbox               recorded
_200103261224_NHK.avi  dead.letter      procmailrc        spool
_200103261228_NHK-EDU.avi  dot.forward     public_html
[zen-t@zen06 tvmaster]$ ls -a
.                  .canna           .less             .xauth            dead.letter
..                 .elvisrc        .qt118nrc        .xemacs           dot.forward
.Xdefaults        .emacs          .rhosts          .xsession        mbox
.bash_history     .exrc           .screenrc        Desktop          procmailrc
.bash_logout     .inputrc       .tcshrc          _200103261224_NHK.avi  public_html
.bash_profile    .kde            .vimrc           _200103261228_NHK-EDU.avi  recorded
.bashrc          .lang           .wl              bin               spool
[zen-t@zen06 tvmaster]$

```

画面1 lsとls -a

```

K ~ term
[tvmaster@zen06 tvmaster]$ ls
Desktop          bin                mbox               public_html
_200103261224_NHK.avi  dead.letter      oldhome           recorded
_200103261228_NHK-EDU.avi  dot.forward     procmailrc        spool
[tvmaster@zen06 tvmaster]$ ls -F
Desktop/        bin/              mbox              public_html/
_200103261224_NHK.avi  dead.letter      oldhome@         recorded/
_200103261228_NHK-EDU.avi  dot.forward     procmailrc*     spool/
[tvmaster@zen06 tvmaster]$

```

画面2 ls -Fオプションの効果

トファイルを作成して「ll」というような名前で作成しておく。

```
ls -l
```

この「ll」ファイルは「chmod +x ll」のようにして実行可能属性をつけて、/usr/local/bin/ディレクトリにでも置いておけば、コマンドラインで、

```
$ ll
```

と実行するだけで、「ls -l」と同じ結果が得られる(画面3)。

しかし、bashには、このような面倒なことをしなくても、コマンドを省略できるしくみがある。それが今回とりあげるエイリアス(alias)という機能である。

エイリアスとは何か

エイリアスとは、英語では別名とか偽名というような意味になるのだが、Linux(bash)の世界では、コマンドの省略形とかコマンドの別名というような意味を持っている。長くて複雑なコマンド名やコマンドラインに、短い「別名」を付けて、その別名を入力するだけで元のコマ

ンドを実行できるようにするものである(図1)。先ほど例にあげた「llファイル」のような「シェルスクリプト」でも、似たようなことができるのだが、エイリアスは簡単なコマンドを実行するだけで、必要なときに必要な数だけ別名を作成できる点が便利である。

bashでエイリアスを作成する(もう少し難しい言葉で言えば「エイリアスを定義する」)には、bashの内部コマンドであるaliasコマンドを使用する。aliasコマンドの使い方は、次のようになる。

```
$ alias ll='ls -l'
```

aliasコマンドの引数には、まず新たに定義したいエイリアスの名前(すなわち、新しいコマンド名)を記述し、「=」に続いて、そのエイリアスが使われたときに、実際に実行したいコマンドを記述する。このとき、「'」などの引用符を使って、どこからどこまでが割り当てたいコマンド文字列なのかを明確に示す必要がある。

上のようにして、「ls -l」に「ll」という別名を割り当ててやれば、次からは、

```
$ ll
```

と入力するだけで、ls -lと同じ実行結果を得られるようになる。「llファイル」を作ったときのように、ファイルを編集したり、/usr/local/bin/ディレクトリなどに配置したり、chmodコマンドで実行属性を追加したりするような手間はいっさい必要ない。

aliasコマンドは、必要なときにいつでもエイリアスを追加できる。たとえば、あるコマンドを使った作業を繰り返し行っているようなとき、一時的に簡単なエイリアスを定義して使うことができる。

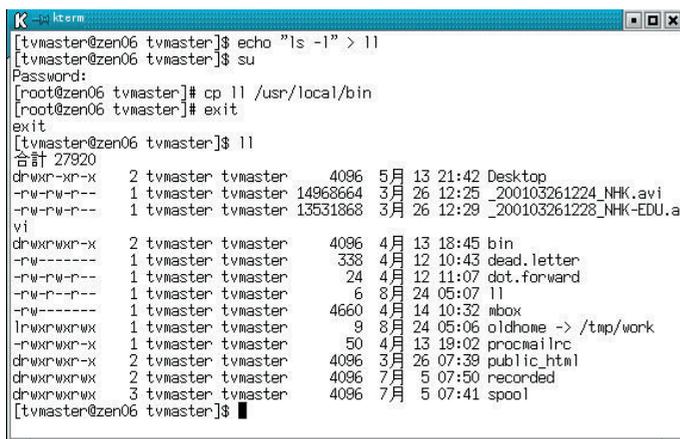
```
$ alias tcf='tar cf /tmp/collection.tar'
$ tcf *.jpg
$ cd /home/user1/image
$ tcf *.jpg
:
```

上のように、「あるtarファイルにファイルを追加するtarコマンド」をエイリアスとして定義しておけば、長いtarコマンドを入力しなくても、「tcf ファイル名」という簡単な形式で、さまざまなディレクトリに分散しているファイルを、手動で1つのtarファイルに収集するような作業が簡単にできるようになる。

/.bashrcで定義する

そのときだけ必要なコマンドなら、先ほどのように使うたびにaliasコマンドでエイリアスを定義すればよいのだが、「ls -l」のエイリアスのように、頻繁に使うと思われるようなものについては、Linuxを起動したらいつでも使えるようになっていると便利である。

そのようなときは、aliasコマンドによる設定を、ユーザーのホームディレ



```
[tvmaster@zen06 tvmaster]$ echo "ls -l" > ll
[tvmaster@zen06 tvmaster]$ su
Password:
[root@zen06 tvmaster]# cp ll /usr/local/bin
[root@zen06 tvmaster]# exit
exit
[tvmaster@zen06 tvmaster]$ ll
合計 27920
drwxr-xr-x  2 tvmaster tvmaster  4096  5月 13 21:42 Desktop
-rw-rw-r--  1 tvmaster tvmaster 14968664 3月 26 12:25 _200103261224_NHK.avi
-rw-rw-r--  1 tvmaster tvmaster 13531868 3月 26 12:29 _200103261228_NHK-EDU.avi
vi
drwxr-xr-x  2 tvmaster tvmaster  4096  4月 13 18:45 bin
-rw-----  1 tvmaster tvmaster   338  4月 12 10:43 dead_letter
-rw-rw-r--  1 tvmaster tvmaster    24  4月 12 11:07 dot_forward
-rw-rw-r--  1 tvmaster tvmaster    6  8月 24 05:07 ll
-rw-----  1 tvmaster tvmaster  4660  4月 14 10:32 mbox
lrwxrwxrwx  1 tvmaster tvmaster    9  8月 24 05:06 oldhome -> /tmp/work
-rwxrwxr-x  1 tvmaster tvmaster    50  4月 13 19:02 procmailrc
drwxrwxr-x  2 tvmaster tvmaster  4096  3月 26 07:39 public_html
drwxrwxrwx  2 tvmaster tvmaster  4096  7月 5 07:50 recorded
drwxrwxrwx  3 tvmaster tvmaster  4096  7月 5 07:41 spool
[tvmaster@zen06 tvmaster]$
```

画面3 「ll」シェルスクリプト

クトリの.bashrcファイルで行うようにすればよい。bashrcファイルは、ユーザーがログインしてbashが起動したときに最初に行われるようになっていて、シェルのプロンプトでいつでも同じエイリアスが使えるようになる。

実は、Linuxディストリビューションの多くでは、インストールした状態ですでに、.bashrcファイルでいくつかのエイリアスが定義されている。たとえば、Turbolinux Workstation 6.0では、次のようなエイリアスが定義されている。

```
alias cp='cp -i'
alias mv='mv -i'
alias ..='cd ..'
alias ll='ls -l'
alias la='ls -aF'
alias ls='ls -NF'
```

まさにこの記事でいままで例にして

きたll (ls -l) も、しっかり定義されている。また、「ls -aF」のほか、lsというコマンド自身もエイリアスになっていて、「ls -NF」というコマンドが実行されるようになっている。ちなみにlsの「-N」オプションは、「制御文字などが含まれたファイル名もそのまま表示する」という意味である。日本語のファイル名やWindowsなどの別のOSで作成したへんなファイル名も表示できるようにという配慮だろう。

ユーザーが自分のエイリアスを使えるようにしたいのであれば、.bashrcファイルにaliasコマンドの行を追加してやればよい。ちなみに筆者はlsコマンドに関しては、次のようなエイリアスを追加して使っている。

```
alias lal='ls -aF'
```

話は少し脱線するが、上の例のTurboLinuxで定義されているalias ..='cd ..'というの、なかなかおも

しろいエイリアスである。これは、1つ上のディレクトリに移動する「cd ..」というコマンドを、単に「..」と入力するだけで実行できるようにするものだ。1つ上の親ディレクトリに移動するというコマンドは、ひょっとするとコマンドラインでもっとも頻繁に実行するコマンドかもしれない。これが「..」だけで済むようになれば、確かに便利である。

もしこれと同じようなことを、エイリアスではなくシェルスクリプトなどのコマンドラインで行おうとしても、不可能である。なぜなら、Linuxのファイルシステムでは、.や..という名前のファイルを作成することができないからだ。より正確に言えば、すべてのディレクトリには「.」と「..」という名前の実体のないファイルが、あらかじめ作成されていて、それぞれ、自分自身のディレクトリ、親ディレクトリを指す、一種の「リンク」として機能しているのだ。

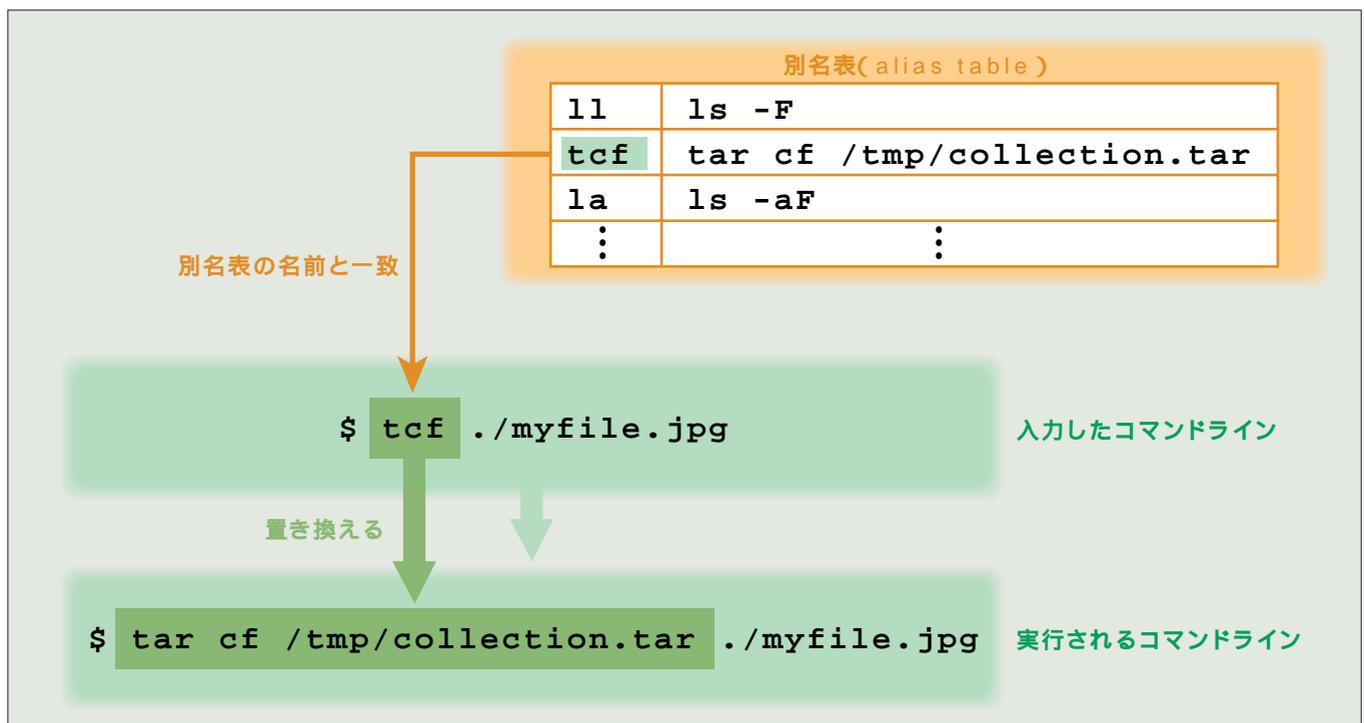


図1 コマンド名の「別名」の仕組み

ついでなので、

```
alias ...='cd ../..'
alias ....='cd ../../..'
alias .....='cd ../../../..'
```

といった具合に、2つ上、3つ上、4つ上...といったように、さらに上のレベルのディレクトリに移動するエイリアスがあっても便利だろう。

ところで、もし「/tmp/test」というように、ルートディレクトリから2つのレベル下にあるディレクトリで、2つ以上上のディレクトリに移動するようなコマンドを実行すると、おかしなことになるのでは？と心配する人がいるかもしれないが、これは問題ない。Linuxのファイルシステム(ext2)では、ルートディレクトリ中で「..」ディレクトリを指定すると、ルートディレクトリそのものを示すようになっているからだ。

エイリアスの作法？

エイリアスの使い方に規則のようなものがあるわけではない。強いていえば、既存の重要なコマンドを使えなくするようなエイリアスは避けるとか、ファイルを削

除してしまうようなエイリアスを定義するときは、操作ミスをしないように注意するといった程度の注意事項しかないだろう。

たとえば、「alias」という名前のエイリアスを作ってしまうこともできる。

```
$ alias alias='ls'
```

すると、コマンドラインで「alias」と入力すると、実際にはlsコマンドが実行されて、ファイルの一覧が表示されるようになってしまう(画面4)。この状態で、別のエイリアスを定義しようとしても、

```
$ alias lal='ls -alF'
ls: lal=ls -alF: No such file or directory
```

というように、そんなファイルはないというlsコマンドのエラーが発生することになる。

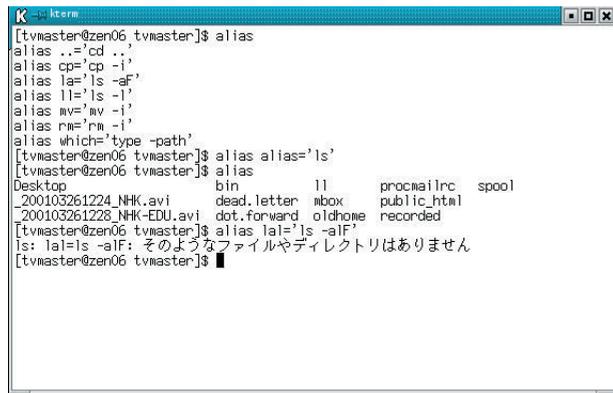
間違っってこのようにしてしまった場合には、後述する方法でエイリアスを削除することができるので、たいした問題にはならないのだが、たとえば、システム管理に関するような重要なコマンドを、無意味なエイリアスで上書きしてしまうと、とんでもない障害が

発生する可能性もないわけではない。

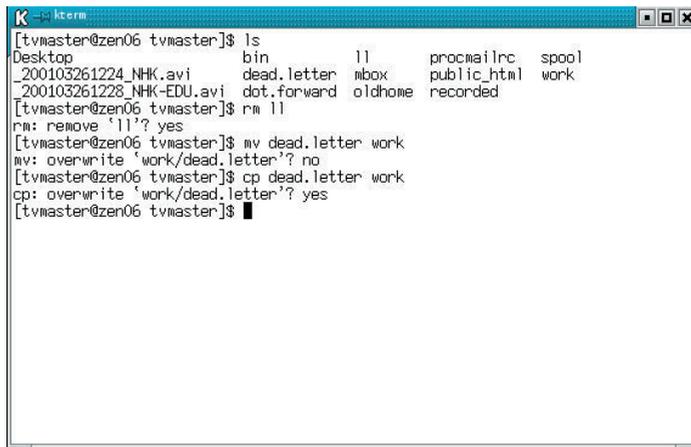
前出のTurbolinuxのデフォルトのエイリアス定義には、「mv」コマンドに関するエイリアスが含まれている。mvコマンドを実行すると「-iオプション」をつけるようにするというものだ。-iオプションは、「ファイルの上書きをする場合には、ユーザーに確認する」という意味なのだが、昔からのUNIXユーザーの中には、このようなエイリアスの使い方に神経質になる人も少なくない。

初心者ユーザーが、mvコマンドにこのようなエイリアスが定義されていることを知らずに使い続けてしまうと、mvコマンドはファイルを上書きするときにはかならず確認してくれるという勘違いをしてしまう危険性がある。このようなユーザーが、mvにエイリアスを定義していないような環境でUNIXやLinuxを使うと、不用意にmvコマンドを使って重要なファイルを上書きしてしまう危険性があるというのである。

同じ理由から問題視されるエイリアスには、「rm -i」(ファイルを削除する前に確認する)や「cp -i」(ファイルを上書きする前に確認する)もあり、こちらについても多くのディストリビ



画面4 「alias」エイリアス
ちなみに、「正しい状態」(?)なら、「alias」を実行すると、その時点で定義されているエイリアスの一覧が表示される。



画面5 mv -iやcp -iをそれぞれ「mv」、「cp」にエイリアスしている場合

ューションで、とくにrootユーザーの.bashrcファイルであらかじめ設定されていることが多い(画面5)。

このようなエイリアス設定を毛嫌いする理由は十分理解できるし、複数のユーザーが1台のマシンを共有していたような時代なら、1人のユーザーのミスは多数のユーザーにつながることもあったので、**死活問題**でもあったわけだ。

しかし、現在のように、1人が1台(あるいは何台もの)Linuxマシンを**専用**に使う時代になってからは、たとえ自分がミスしても、**その害を被るのは自分1人だけ**なので、あまり神経質になる必要はないだろう。

最近のLinuxディストリビューションが、「rm -i」や「cp -i」などのエイリアスを設定しているのは、WindowsやMacintoshなどのOSでは、ファイルの上書きを確認するようになっているのとも関係があるのかもしれない(画面6)。これらのOSのユーザーがLinuxを使ったときに、違和感なく使えるようにという配慮なのだろう。

このような配慮といえば、

```
$ alias dir='ls -l'
```

というように、lsコマンドにdirというエイリアスを定義しているディストリビューションも見かける。これなどは、まさに、Windows(DOS)ユーザーの利便(のみ)を考えてのことと

いってよいだろう(逆に、MS-DOSに「LS.BAT」というバッチファイルを仕掛けて、つかの間のUNIX気分を味わっていたことがあったのも思い出す)。

エイリアスを無効にする方法

さて、エイリアスはこのように簡単に使える便利な機能なので、形式にとらわれずに、便利なエイリアスをどんどん**開発**してみしてほしい。これは便利だというものは、/.bashrcにどんどん追加してしまえばよい。

しかし、先ほどのaliasエイリアスのように、間違っただけで作成してしまったものは、エイリアスを無効にしなければならない。

まず、コマンドラインを実行するときに、**一時的に無効にしたい**場合を考えてみよう。たとえば、lsというエイリアスに「ls -F」というコマンドが割り当てられている場合、コマンドラインでlsコマンドを実行すると次のようになる。

```
$ ls
file1.txt command.sh* directory1/
```

このように、ファイル名の末尾に、そのファイルの種類を表す記号文字が付けられることになる。しかし、たとえばlsコマンドの結果を別のコマンドで処理したり、テキストファイルとし

て残しておいたりしたいとき、この文字が邪魔になることがある。そのためだけに、いちいちエイリアスの定義を変えたりしては面倒なので、あるコマンドラインの中だけでエイリアスを無効にする方法がある。

次のように、コマンド名の先頭に¥(円記号、英語フォントの場合は、半角の\バックスラッシュ)を付けてみよう。

```
$ \ls
file1.txt command.sh directory1
```

このように、エイリアスの設定は無視されて、lsコマンドがいわば**生の状態**で実行されるようになる。

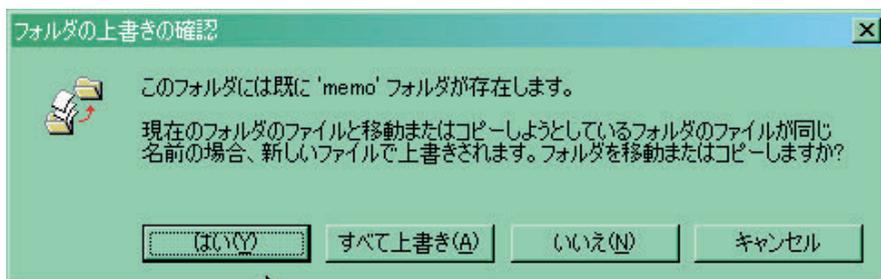
一時的に無効にするのではなく、使用中のシェルを終了する(あるいはログアウトする)までの間、ずっとエイリアスを無効にしたい場合には、**unalias**コマンドを使う。unaliasの使い方は簡単である。

```
$ unalias ls
```

これで、「ls」という名前に割り当てられていたエイリアスは無効になるので、これ以降、「lsコマンド」を入力すると、「生のlsコマンド」が実行されることになる。

もちろん、/.bashrcの中で定義されているエイリアスを、**未来永劫金輪際使わない**ということであれば、/.bashrcをテキストエディタで編集して、そのエイリアスを定義している部分をコメントアウト(先頭に#を付ける)か、削除してしまえばよい。

エイリアスの使い方には、これ以外にも多少高度な話題があるのだが、それについては別の機会に触れることにしたい。



画面6 Windowsではファイルのコピーや移動をするときに、必ず上書きするかどうかを確認してくる



彼方からの 手紙

— IMAP4メール環境で遊ぶ —

第1回 とりあえず「基本的なことなど」

文：塩田紳二
Text : Shinji Shioda
illustration : M.T

今回から始まりますこの企画、簡単に申せば、電子メールの仕組みを使ってスクリプトを作って、いろいろやってみようというもの。皆さまが毎日お使いの電子メールにちょっとした便利な機能を付け加えてみようという試みでございます。と申しても、真面目一本槍ではないために、もしかしたら失笑を買ってしまうようなものがあるかもしれません。とはいえ、今後とも最良に願いたく、まずは、始まりの口上など。さて今回は、初回ということもあり、電子メールの復習でございます。

なぜメールで？

電子メールは、みなさん、すでにお使いのことと思います。特に常時インターネットに接続している環境であれば、メールクライアントは常に動いていて、刻々とやってくるメールをお読みになっているのではないのでしょうか。我々が受け取る電子メールのなかには、いろいろなメッセージが入っております。重要な会議の通知や飲み会のお誘い、くだらないメール、SPAM、いろいろなニュースレターやメーリングリスト、などなど。多い人は、日に数百のメールを受け取るともいわれています。

電子メールはいまや日常的に使う道具で、ケータイ電話同様の、生活必需品といえましょう。このような電子メールの仕組みをちょっと使って、日々の生活が少しラクになるのなら、というのがこの連載の企画意図です。

目的のひとつは、電子メール生活の環境向上です。多く

のメールを受け取ることを考えると、いかにして「重要なメール」を取り出すか、あるいはいかにして「不要なメール」を排除するか、が課題のひとつとなるでしょう。ただし、メールの価値を判断するのはあくまでも人間。となると、機械的な方法であらかじめメールを分類しておくことが有効です。また、受信メールをすべて1つのフォルダに入れておくなんてことは、あとからの検索にも時間がかかるので、整理して保存したいところ。と考えると、いわゆるメッセージの「振り分け」についてもなんらかのテを考えないといけないでしょう。

メールなんて、読んだら消しちゃうよ、という方は、もしかしたらこの記事とは相性が悪いかもしれません。どちらかという、メールをいつまでも保存している人のほうがフィーリングが合いそうです。

昔のメールなんて、メールソフトが受け取ったマシンでしか読めないじゃん？と思われた方、この記事では、どこからでもメールにアクセスできるようにIMAP4サーバを使います。このため、利用しているメールソフトの違いはほとんど問題にならないはずですが（もっとも、IMAP4が使えないメールソフトだったら問題があるのですが……）。

もうひとつの目的は、電子メールシステムの応用です。たとえば、ちょっとしたメモを自分あてのメールとして送ったことはありませんか？電子メールシステムはテキストのデータベースでもあり、ここに置いておけば、自分で消さないかぎりなくなってしまうことはありません。また、単純なテキストファイルと違って検索も容易だし、なにし

ろ、毎日目にしているところにあるという安心感があります。皆さまも、大事なメモをポストイットでディスプレイに貼り付けてませんか？ このように、かならず目に入るところに置いておくことが重要なのではないのでしょうか。

そんなわけで、この連載では少々プログラミングをいたします。といっても、Perlでスクリプトを組むだけのことです。本来なら、自分の環境や目的に合わせてこれを改造して使ってほしいわけですが、まあ、IMAP4が動いている環境なら、そのまま動かすこともできるので、出来合いのプログラムとしてそのまま使っていただいても結構です。



連載の流れなど

今回は、初回としてちょっとした解説をしますが、次回よりスクリプトを紹介していきます。なお、プログラムを作るのは私ではなく、とりあえず現時点では匿名希望のプログラマーS氏であります。職業的プログラマーであり、組み込み系のお仕事をされていて、「原稿は1行も書きたくないが、プログラムならいくらでも書く」と、本記事担当者に宣言したため、強制的にこの企画に組み込まれてしまいました（その意味でも彼は組み込みプログラマーなのです）。筆者の場合、「原稿ならいくらでも書くが、仕事でプログラムを書くのはいや」というほうなので、組み合わせとしてはちょうどいいのかも。

これからは毎月、基本的な「お題」としてなんらかの目



的が示され、それを実現するプログラムを作ります。あまり、大きな機能のプログラムにはしない予定ですが、機能や仕様を考えるのは、プログラマーではなく私なので、過大な要求を出してしまうかもしれません。そうすると、1回で終わる予定が2回、3回となってしまいますが、まあ、それも「ちょっとは便利なものができる」とご了承ください。

あと、毎回プログラムの解説ってのもなんですので、お勉強コーナーとして、そのプログラムにまつわるいろいろな解説を適宜入れていきます。



電子メールのキホン

さて、ここからは解説です。今後の記事で、毎回は細かく解説できないので、ここで電子メールのキホンをおさらいしておこうというわけです。ちょっと不安なら、電子メール関連の書籍などを入手しておいたほうがいいかもしれません。

電子メールといっても本当はいろいろな方式があるので、ここで取り上げるのはインターネットで使われている電子メール（以下、電子メールといえばインターネットで使われているものを指すことにしておきます）。つまり、RFC822形式（最近、RFC2822に改訂された）で決められているテキストベースの本文をメッセージとして、SMTPで転送するものです。もっとも、最近ではほとんどMIME（RFC2045～2049など）に準拠して拡張され、画像やプログラムなどテキスト以外のデータも取り扱えます。

もうひとつ重要な点として、DNSがあります。メールを送信するのに先だって、まず、DNSを使って宛先ドメインでメールを受け取ってくれるサーバを調べる必要があるのです。この情報は実際には、宛先ドメインのDNSサーバにMXレコードとして登録されています。

多くの電子メールの解説では、メールの読み書きに使うメールソフトなどをMUA（Mail User Agent）と呼び、マシン間でメールを転送する仕組みをMTA（Mail Transfer Agent）といいます。たとえば、sendmailやqmailはMTAになるわけです。

ですが、かつてのUNIXのメールシステムでは、どこかのディレクトリにsendmailが着信したメッセージが保存されていて、ユーザーはそのマシンにログインしてメッセージを直接読んだり、sendmailを直接呼び出して送信したりしましたが、パソコンがこれだけ普及すると、電子メールソフトはPOPやIMAP4を使ってサーバからメールを取り出し、送信もSMTPを使ってサーバに中継を依頼す

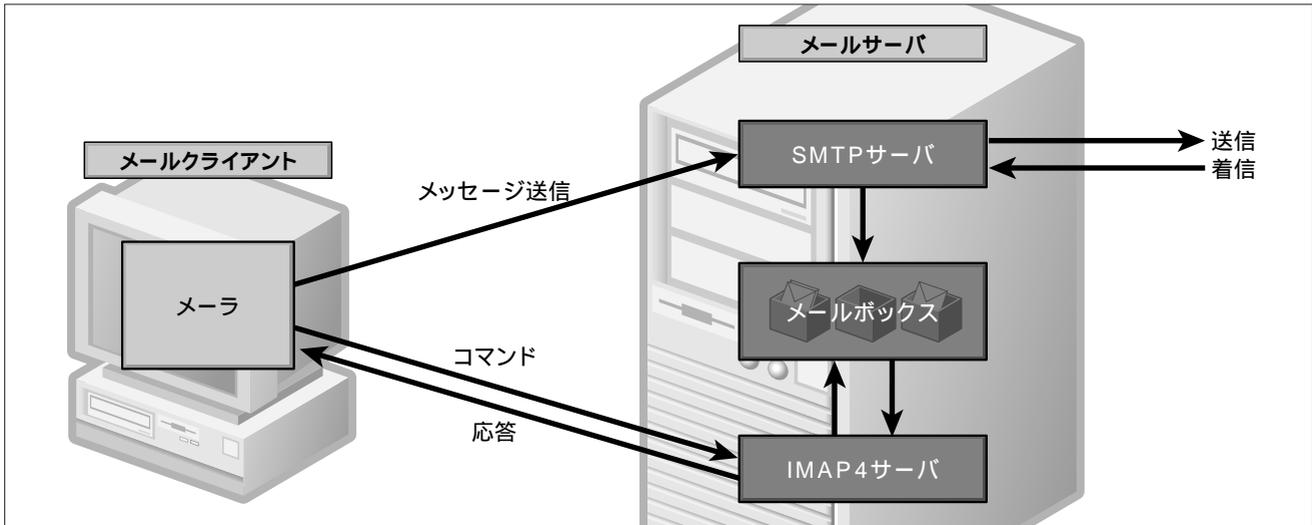


図1 IMAP4を使ったメール環境

るという状況になってきました(図1)。これらのメールソフトは、ある意味、MTAの機能の一部を持っているともいえます。またかつては、DNSでMXレコードを引いて、直接SMTPで送信できるWindows用のメールソフトもありました。

そんなわけで、この連載では、MUAやMTAという用語にはあまりこだわらないことにしました。ユーザーがメールの読み書きに使うプログラムを単に「メールソフト」と呼ぶことにします。

さて、この記事で重要になるのは、IMAP4です。まだ、お使いになっていない方もいらっしゃると思うので、ここでちょっとIMAP4について解説しておきましょう。

IMAP4とは？

最近では、電子メールは受信者にサービスをしているメールサーバまで配送され、受信者は手元のマシンでメールソフトを動かしてメールサーバにアクセスし、メールを受け取るようになってきました。というのも、家庭からISP経由でインターネットに接続している場合、ダイヤルアップ接続がほとんどで、届いたメールをどこかに保存しておいてもらわないと、インターネットにつながってなくてメールを受け取れない時間があるからです。そこで最もよく使われているのがPOP(Post Office Protocol)です。簡単にいえば、POPというのは、郵便局に行って自分のメールを受け取るようなものです。「私書箱」という宛先がありますが、これは郵便局にある郵便受けです。毎月郵便局にお金を払うと、この私書箱を使うことができます。ちょっと大きな郵便局に行くと、壁に小さなロッカーのようなものが並んでいますが、これが私書箱です。

このPOPを使う場合、メールソフトで取り込んだメールは、そのメールソフトを使って自分で管理します。このため、メールの保存にはメールソフトにとって都合がいいようなファイルフォーマットが使われています。そうすると、ほかのプログラムでメールを扱うのはやっかいになります。一番問題となるのが、外出時にノートパソコンを使ってメールを使うような場合です。この場合、今やってきたメールは受信可能ですが、すでにデスクトップマシンで受信してしまったメールを読むことはできません。このためには、デスクトップマシンとノートパソコンで同じソフトを使って、メールフォルダファイルなどをコピーしておかねばなりません(あるいは、テキストファイルにセーブしたり印刷して持ち歩くか)。

これでは不便なので、メッセージ管理の部分をサーバに置いて、いろいろなホストやメールソフトからいつでもメールを取り出せるようにしようというのがIMAP4(Internet Mail Access Protocol version 4 rev1: RFC2060)です。

IMAP4は、簡単にいってしまえば、サーバ側にあるメールフォルダ(メールボックスと呼ぶ)をアクセスするためのプロトコルです。メールボックスを選んで、そこにあるメッセージの一覧を取得したり、特定のメールを読み出したりします。また、メッセージのMIME解析機能があるため、本文部分のみを取り出したり、添付ファイルのみを取り出すといったことも可能です。このほか、検索機能などもあります。

IMAP4は、TCPのポート143番を使って通信します。また、SMTPやPOPに似ていて、テキストベースのコマンドを使います。ただし、複数のコマンドを同時に発行可

能で、このために各コマンドの先頭にタグを付けて識別するようになっています。このタグは英数字の組み合わせならなんでもよく、応答が返ってきたあとなら同じタグを使えます。コマンドの書式は、

タグ コマンド 引数...

という形になります。

これに対して応答の形式は、

* 応答情報

* :
* :

タグ ステータス 文字列

という形になります。もし情報を返す必要がない場合には、最後の行のみが返されます。この場合、ステータスは“OK”、“NO”、“BAD”のいずれかです。OKは、コマンドが正常に実行できた場合、NOはコマンドが失敗した場合、BADはコマンドが間違っているなどのエラーを表します。あとに続く文字列は、その状況を表します。たとえば、以下のような感じです（下線部はクライアントからのコマンド）。

101 CAP間違ったコマンド

101 BAD Command unrecognized/login please: CAP

102 LOGIN user password正しいコマンド

102 OK LOGIN completed

103 select "Mail/_test/test9"コマンド実行不可

103 NO SELECT failed: Can't open mailbox Mail/_test/test9: no such mailbox

IMAP4には、図2のような状態遷移があります。簡単にいえば、接続が行われたあとの認証前の状態（Non-Authenticated）、認証が行われたあとメールボックスを選択しているとき（Selected）、認証が行われたがメールボックスは選択していないとき（Authenticated）の3つです。IMAP4のコマンドには表1のようなものがありますが、コマンドによってどの状態で実行できるかが決まっています。

メッセージを読み出したり操作するためには、まず該当のメールボックスを選択します。そのあと、FETCHやSTORE、COPYなどでメッセージを操作します。メールボックス自体を処理したい場合には、メールボックスを選択せずに直接コマンドを発行します。コマンドの実行対象としては、ホストが受信したメッセージを格納する特殊なメールボックスINBOXと、ユーザーが作成したほかのメールボックスがあります。どちらのメールボックスも、同等に操作することができます。

実際にコマンドを使ってメッセージをアクセスしてみたようですが、155ページの画面1です。ここで注意すること

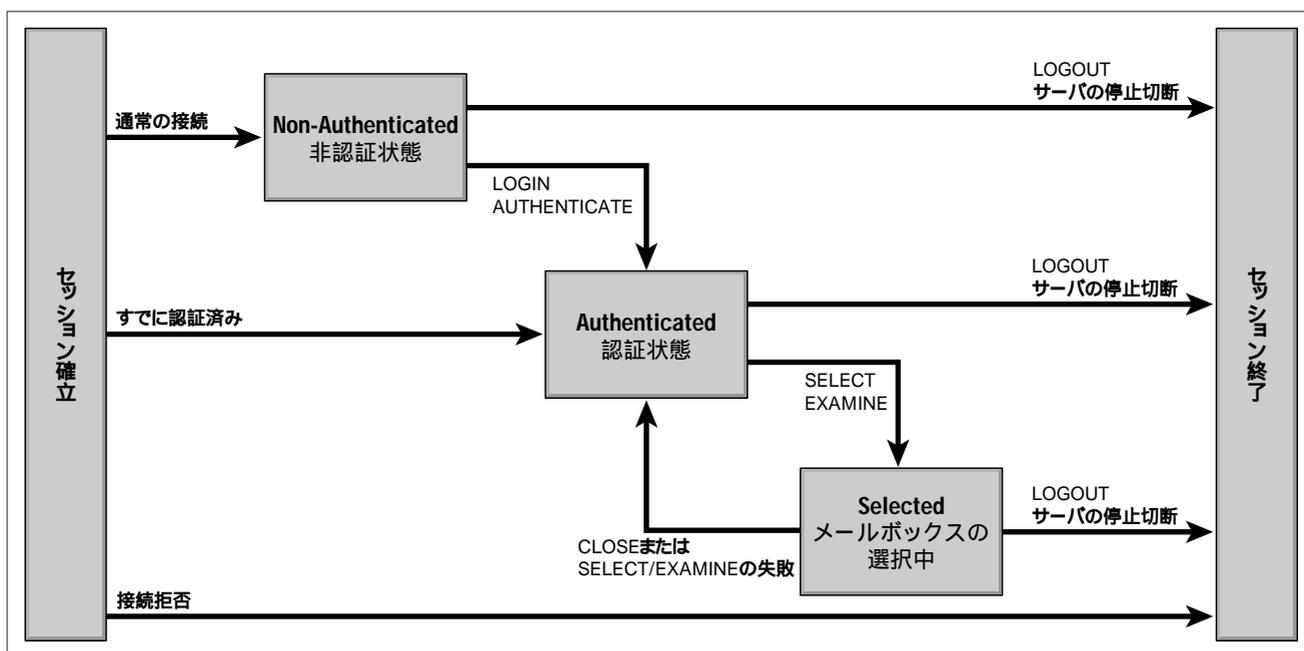


図2 IMAP4の状態遷移

は、FETCH コマンドなどで、目当てのメッセージを指定する方法です。メッセージ番号（1で始まっているメールボックス内の通し番号）は、別のメッセージの削除などで変化してしまいますが、UID コマンドを使って、

タグ UID FETCH UID番号 引数

とすると、常に変わらないUID（サーバが割り当てる Uniq ID）を使ったアクセスが可能になります。逆に、メッセージ番号でアクセスしたい場合には、

タグ FETCH 番号 引数

とします。通常、受信したメッセージは原則として書き変わることがないので、そのメッセージが削除されないかぎり、あとからそのUIDを使って同じメッセージをアクセス可能です。逆に、過去にアクセスしたときと同じUIDであれば、同じメッセージだと考えていいわけです。注意すべきなのは、いったんフォルダ全体が削除されて、また同名のフォルダが作られた場合です。この場合、たとえUIDが同じでも、違うメッセージになっている可能性があります。

これを確認するには、SELECT 時に返される“UID VALIDITY”の値が前回と同じかどうかを確認します。通常、IMAP4対応のメールソフトは、この性質を利用してメッセージリストやメッセージそのものをキャッシュしておき、再度要求があったときでも変更がなければ、本文などをあらかじめ取得せずにキャッシュの情報を使います。

UIDのリストは、FETCH コマンドに引数としてUIDを指定することで得られます。また、メッセージの総数は、SELECT コマンドの応答で返されます（STATUS コマンドでも取得は可能です）。

実際、IMAP4のコマンドを生で使うことは少ないと思いますが、プロトコル自体を理解しておく、今後のスク립トの解説がわかりやすくなると思います。



とりあえずどんな環境がいるの？

さてこの連載の内容は、IMAP4サーバとPerlさえ用意できれば、どんな環境にもあてはまるはずですが、ただし、すべての環境で動作をチェックするというわけにもいかないので、検証用の環境として、次のようなものを用意しました。

	コマンド	書式	解説
常時	CAPABILITY	タグ CAPABILITY	IMAPのバージョン、拡張機能などの取得
	NOOP	タグ NOOP	なにもしない
	LOGOUT	タグ LOGOUT	ログアウト
非認証中	AUTHENTICATE	タグ AUTHENTICATE 使用認証方式	クリアテキスト以外の認証方式の実行
	LOGIN	タグ LOGIN ユーザー名 パスワード	クリアテキストによる認証
認証中	SELECT	タグ SELECT メールボックス名	メールボックスの選択
	EXAMINE	タグ EXAMINE メールボックス名	メールボックスの選択（変更はしない）
	CREATE	タグ CREATE メールボックス名	メールボックスの作成
	DELETE	タグ DELETE メールボックス名	メールボックスの削除
	RENAME	タグ RENAME 旧メールボックス名 新メールボックス	メールボックス名の変更
	SUBSCRIBE	タグ SUBSCRIBE メールボックス名	メールボックスを購読済みに指定
	STATUS	タグ STATUS メールボックス名（ステータス）	メールボックスの情報を取得
	UNSUBSCRIBE	タグ UNSUBSCRIBE メールボックス名	メールボックスを未購読に指定
	LIST	タグ LIST 参照名 メールボックス名	メールボックスのリスト。ワイルドカードとして“*”（パスを含む）と“%”（パスを含まない）が使える
	LSUB	タグ LSUB 参照名 メールボックス名	購読済みのメールボックスのリスト
	APPEND	タグ APPEND メールボックス名 [フラグリスト] [日付/時刻文字列] メッセージ	メッセージの追加
選択中	CHECK	タグ CHECK	チェックポイントの実行
	CLOSE	タグ CLOSE	メールボックス選択の終了
	EXPUNGE	タグ EXPUNGE	¥DELETE フラグのついたメッセージの削除
	SEARCH	タグ SEARCH [文字セット] 検索条件	メッセージの検索
	FETCH	タグ FETCH 対象メッセージ データ項目名	メッセージの取得
	STORE	タグ STORE 対象メッセージ フラグコマンド（フラグリスト）	フラグの設定
	COPY	タグ COPY 対象メッセージ メールボックス名	メールボックス間でのメッセージのコピー
	UID	タグ UID COPY/FETCH/STORE 引数 タグ UID SEARCH UID ユニークな番号...	ユニークな番号によるメッセージの指定を行う
	X(atom)	タグ Xcmd	拡張機能実装用のための接頭辞

表1 IMAP4 コマンド

Red Hat Linux 7.0.1J
imap-4.7c2-12
perl-5.6.0-10

Linuxなら、ディストリビューションやIMAP4サーバのバージョンが多少違っていてもなんとかなると思われま

さて、IMAP4サーバですが、手に入れやすいものとしてワシントン大学で作られたUW-IMAPサーバがあります。このほか、カーネギーメロン大学のCYRUSというサーバもありますが、インストールだけを見ればUW-IMAPのほうが簡単です。というのは、CYRUSはsendmail.cfに手を入れる必要があるからです（CFで設定は可能ですが）。このため、UW-IMAPサーバは、RPMなどのパッケージ形式で多くのディストリビューションに付属しています。これを使えば、インストールは簡単に終わります。なお、以降で必要になるので、開発用のパッケージであるimap-devel-xxxx.rpmなども同時にインストールしておきます。これは、IMAPなどのアクセスを簡単に行うためのライブラリなどがインストールされます。

Red Hat 7Jについては、インストールが終わったらxinetd.dディレクトリのimapファイルを開き、disable行が“no”になっているのを確認します。ディストリビューションによっては、この代わりにinetd.confを使うようになっているかもしれません。この場合は、imapのところの先頭に“#”が付いていないことを確認します。なお、この設定ファイルを書き換えたら、xinetd（またはinetd）にSIGHUPを送ります。

また、tcpd経由で起動するようになっているはずなので、tcpdでローカル環境からのアクセスが可能のように設定しておきます（デフォルトならそのままアクセス可能になっているはずなので、触る必要はないでしょう）。

あとは、IMAP4対応のメールソフト（Windows環境ならOutlook Expressなど）を使って、IMAP4でのアクセスが可能かどうかを確認します。うまく動いたでしょうか？ では次回をお楽しみに。

今回は、簡単な例としてPerlからIMAP4のメッセージを扱うところから始めたいと思っています。さて、何ができるでしょうか。

項目名	説明	備考
ALL	(FLAGS INTERNALDATE RFC822.SIZE ENVELOPE)と同じ	
BODY	拡張部分のないメッセージ構造	(type subtype (parameter-list) id description encoding size [lines])の順
BODYSTRUCTURE	MIMEヘッダを解釈したメッセージ構造	(type subtype (parameter-list) id description encoding size [lines] (nested-body) multipart-subtype Extended-part)の順
BODY [セクション]	本体構造の一部の取り出し	ヘッダ部分は0、本文は1、マルチパートの場合には1～nを指定する。なお、マルチパートのメッセージは、ピリオドで区切って指定する（例：マルチパートの最初のメッセージの本文は1.1と指定）
BODY.PEEK [セクション]	BODY [セクション]と同じ。ただしSeenフラグをセットしない	
ENVELOPE	ヘッダ情報	(date subject from (sender) (reply-to) (to) (cc) (bcc) (in-reply-to) (message-id))
INTERNALDATE	メッセージが最後に転送された日時	SMTP [RFC821]で指定された形式の日時情報
FAST	(FLAGS INTERNALDATE RFC822.SIZE)と同じ	
FLAGS	メッセージにセットされたフラグ	FLAGS (フラグの一覧)
FULL	(FLAGS INTERNALDATE RFC822.SIZE ENVELOPE BODY)と同じ	
INTERNALDATE	サーバが管理するメッセージの内部日付	INTERNALDATE "内部日付"
RFC822	RFC822形式のメッセージ	RFC822形式そのままの情報が返される
RFC822.PEEK	上記と同じ内容。ただし、\Seenフラグをセットしない	
RFC822.HEADER	RFC822形式のヘッダ	
RFC822.HEADER.LINES	ヘッダリスト	ヘッダリストで指定するヘッダ情報
RFC822.HEADER.LINES.NOT	ヘッダリスト	ヘッダリストと一致しないフィールド名のヘッダ
RFC822.TEXT	メッセージの本文	
RFC822.TEXT.PEEK	メッセージの本文。ただし、\Seenフラグをセットしない	
RFC822.SIZE	メッセージのサイズ	サイズがバイト数として返される
UID	メッセージのUID	メッセージシーケンスは、削除などにより変化するが、UIDは変化しないユニークな番号

表 2 FETCHコマンドの引数

```

* OK test.xxx.xx.jp IMAP4rev1 v12.264 server ready
100 capability
* CAPABILITY IMAP4 IMAP4REV1 NAMESPACE IDLE SCAN SORT MAILBOX-REFERRALS LOGIN-REFERRALS
AUTH=LOGIN AUTH=ANONYMOUS THREAD=ORDEREDSUBJECT
100 OK CAPABILITY completed
100 login USER PASSWORD
100 OK LOGIN completed
100 List "Mail/ test" "*"
* LIST (\NoSelect) "/" Mail/_test
* LIST (\NoInferiors) "/" Mail/_test/test3
:
* LIST (\NoInferiors \Marked) "/" Mail/_test/bbbb
100 OK LIST completed
100 select "Mail/ test/test3"
* 8 EXISTS
* NO Trying to get mailbox lock from process 18810
* 0 RECENT
* OK [UIDVALIDITY 965640330] UID validity status
* OK [UIDNEXT 12] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
100 OK [READ-WRITE] SELECT completed
101 fetch 1:8 UID
* 1 FETCH (UID 4)
:
* 8 FETCH (UID 11)
101 OK FETCH completed
115 uid fetch 11 body
* 8 FETCH (UID 11 BODY ("TEXT" "PLAIN" ("CHARSET" "iso-2022-jp") NIL NIL "7BIT"31516 655))
115 OK UID FETCH completed
116 uid fetch 11 rfc822.header
* 8 FETCH (UID 11 RFC822.HEADER {669}
Return-Path: <mail24-reply@ascii24.com>
:
)
116 OK UID FETCH completed
111 FETCH 1 RFC822.TEXT
* 1 FETCH (RFC822.TEXT {12840}
ASCII24電子メールサービス 2001-07-07
:
)
111 OK FETCH completed
120 fetch 8 flags
* 8 FETCH (FLAGS (\Seen))
120 OK FETCH completed
120 uid store 11 +flags (\deleted)
* 8 FETCH (FLAGS (\Seen \Deleted) UID 11)
120 OK UID STORE completed
121 expunge
* 8 EXPUNGE
* 7 EXISTS
* 0 RECENT
121 OK Expunged 1 messages
122 close
122 OK CLOSE completed
123 logout
* BYE test.xxx.xx.jp IMAP4rev1 server terminating connection
123 OK LOGOUT completed
Connection closed by foreign host.

```

接続直後のメッセージ

メールボックスの一覧

メールボックスの選択

UIDの取得

メールの情報を取得

ヘッダを取得

本文テキストを取得

削除用にフラグをセット

削除の実行

8番目のメールを削除

全部で7つのメール

メールボックスのクローズ

ログアウト

接続が終了 (telnetのメッセージ)

画面1 IMAPによるメッセージアクセス

サーバ百科

基礎から学ぶネットワークの仕組み

RFC やマニュアルを読んだだけではわかりにくいサーバの動作も、自分で基本を確かめてみると理解しやすい。Linux のツールを使ってサーバの仕組みやインターネットのプロトコルを実験してみよう。

第2回 プロトコルの実験

文：安田幸弘

Text: Yukihiko Yasuda

インターネットでは、さまざまな種類のプロトコルが使われている。一般のユーザーが意識するインターネットのプロトコルは、HTTP (WWW) やSMTP (mail)、FTP といった程度、管理者でもふだん意識するプロトコルは数十個程度だろう。しかし、そのようなプロトコルを利用するためのDNSやルーティング、さらに下位のTCP / UDPのようなプロトコルまで含めると、常時数十種類のプロトコルが使われている。ふだん使わないプロトコルまで含めると、インターネットで利用できるプロトコルの数は千種類以上になる。

インターネットのプロトコルとRFC

サーバを運用するうえで、こうした数々のプロトコルについての知識が要求される場面は少なくないが、そんなときに頼りになるのがISOC (Internet Society: インターネット協会) 傘下のIAB (Internet Architecture Board) が発行するRFC (Request for Comments) と呼ばれる文書群だ。

「RFC」という名前を聞いたことがある方や、何かの機会にRFCをお読みになった方も多いと思うが、RFCとはインターネットのプロトコルに関する原典となる文書群の呼び名で、インターネットのプロトコルに関するドキュメントのすべてはこのRFCが出発点だ。原典は英文なので、日本のユーザーがRFCを読みこなすにはやや敷居が高かったが、最近はインターネットを検索すると、重要な文書



についてはボランティア的な翻訳を見つけることもできる。本連載でも、しばしばRFCに言及することになると思うが、何かのプロトコルについて知りたいときには、ぜひRFCを検索してみるといいだろう。

ところで、インターネットに限らず、通信の分野ではこれまで数々のプロトコルが決められ、使われてきた。たとえば、インターネットのユーザーにも馴染みのモデムでは、モデム同士の通信規格としてITU-T (国際電気通信連合電気通信標準化部門) によるV.xxなどの一連のプロトコルに関する勧告がある。

ITU (国際電気通信連合) は、世界各国の政府代表による国際組織で、その中に設置された電気通信標準化部門は、国際的な通信を可能にするために世界共通のプロトコルを作成・勧告する部門だ。もっとも、ここで作成されたプロトコルは、あくまでも「勧告」であり、法的な強制力は持たず、各国それぞれの事情に合わせて国内の標準規格が作られることもある。日本で言えば、JISで制定されるなどである。いずれにしても、公的な機関の官僚によって決定されるトップダウンの規格という色彩が強い規格である。

しかし、インターネットで使われているプロトコルは同じ通信プロトコルでも決定のプロセスが少々異なっている。RFCとしてまとめられているインターネットのプロトコルは、RFC (Request for Comments: コメント求む) という呼称にも現れているように、「規格」というよりは、ユーザーから「こんなプロトコルが使いたいと思ってま

めてみたけど、意見はありませんか」といったような提案文書の形になっている。これは、実験的なネットワークとしてゼロから出発したインターネットの歴史的な事情を反映するもので、理屈で作られた規格にしたがってプログラムを作るのではなく、実際に動くプロトコルについて、インターネットのコミュニティの中で合意を得ながらプロトコルを完成させていこうというインターネットの技術開発ポリシーに基づくものだ。

このようなRFCの原案は、ISOC傘下のIETF (The Internet Engineering Task Force) によって提出されることが多いが、RFCのポリシーにより、IETFに限らず誰でも自分が開発したプロトコルを持ち寄り、一定のディスカッションや査読を経て、RFCとして登録、公開することができる。ただし、無条件でRFCを登録できるわけではなく、提案されたプロトコルなどに関する内容は、具体的に現実的なものであることが求められる。つまりRFCを読めば、そのプロトコルで通信するためのすべての情報が得られるようになっていなければならない。逆にいえば、どんな新しいプロトコルでも、提案が検討の価値ありと認められれば（認めさせることができれば）、ワーキンググループでインターネットドラフトとしてまとめられ、それを元に議論、承認のプロセスを経て公的な標準プロトコルにすることができる。

なお、RFCの手続きの中で文書はInternet Draft (草案) 以外に、Standard Track (標準化過程のドキュメント) としてProposed Standard (標準化提案)、Draft Standard (標準草案)、Standard (標準) があり、さらにExperimental (実験的)、Informational (情報)、Historical (歴史的) などに分類される。

Internet Draftは、RFCの準備段階のもので、6カ月の期限があり、期限が過ぎたInternet Draftは廃棄される。Internet Draftを経てRFCとしてまとめられた標準的なプロトコルに関するドキュメントはStandard Trackとなり、完成度の順にProposed Standard、Draft Standard、Standardとしてまとめられる。最終的なStandardには、STD1から始まる番号が振られ、公式のインターネット標準プロトコルとしての地位を得ることになるが、Draft Standardの段階までくれば、事実上、ほとんど標準的なプロトコルと見なすことができる。また、現在、最も適切な実装に関するドキュメントは、BCP (Best Current Practice) としてまとめられている。

なお、Experimentalトラックのドキュメントは、Internet Draftとして公開するさらに前の段階で、文字通

り開発のための実験的なプロトコルや、企業が自社の技術を提案する場合などに、Experimentalとして文書化される。さらに、現在は使われていなくてもプロトコルの進歩の過程を追うために必要な文書はHistoricalトラックのRFCとして残されることもある。

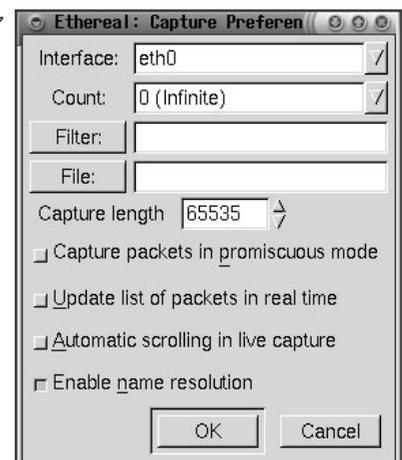
RFCにはすべてが書かれているが、注意すべき点は、RFCは常に改訂されていることだ。現在3000を超えるRFCがあるが、古いRFCの内容は新しいRFCで更新されたり置き換えられたりしている。たとえば、電子メールなどのメッセージフォーマットに関するRFCとしてよく言及されるRFC 822は、いくつかの曖昧な部分が残っており、長い間、それが日本語のメールを実現するうえでの議論の種になってきた。しかし、RFC 822は現在RFC 2822に更新され、問題の部分は解釈の相違が生まれないように明確化されている。

開かれたネットワークであるインターネットは、常に新しい技術が登場し、それにしたがって、RFCも改訂されていく。つまりRFCの内容は、決して固定的なものではなく、技術の開発に伴って常に変更されるものなので、RFCを参照する場合は、できるだけ最新のバージョンを参照するようにしたい。

プロトコルを見る

インターネットのプロトコルのすべてはRFCに記述されていることになっている。したがって、RFCに書かれていることを理解すれば、どんなプロトコルも思いのままに操れるはずだが、実際にはそう簡単にRFCの内容を理解できるというものでもない。また、現実のサイトを運営するうえで管理者を悩ませるのは、RFCに書かれているプロトコルはあくまでも提案であって、現実のシステムが必ずしもRFCに書かれている通りに作られているとは限らないと

画面1 パケットキャプチャの設定



いうことだ。RFCに書かれた通りに作られているはずのサーバが、なぜかうまく動かないということもある。そんなときに、いろいろと調べてみると、クライアントやサーバのバグはともかくとして、クライアントやサーバがRFCと異なる動作をしていたり、RFCの解釈の相違や、サブセットだけしか実装されておらず、動作の互換性が取れていないということもある。

複雑なプロトコルが実際にどのように動作しているのかわかりにくいことがクライアント/サーバシステムの難しさのひとつだが、さまざまなトラブルへの対処はもちろん、インターネットのプロトコルの勉強をする場合にも大いに役に立つのがプロトコルアナライザと呼ばれるツールである。

プロトコルアナライザは、文字通りプロトコルを分析するためのツールである。大規模なネットワークのトラフィックを分析するためには、高価なプロ用のプロトコルアナライザが必要になるが、実験的なネットワークから中小規模のネットワークなら、ソフトウェアによるプロトコルアナライザで十分実用的にトラフィックの分析ができる。

この種のプロトコルアナライザ/スニファは、さまざまな種類のものが公開されているが、フリーソフトウェアのプロトコルアナライザとしては、Ethereal (<http://www.ethereal.com/>) が使いやすく、性能的にも充実している。プロトコルの勉強やネットワーク管理をはじめ、使い方を覚えておくと何かと便利なツールなので、簡単にEtherealの使い方を紹介しておこう。

Etherealの使い方

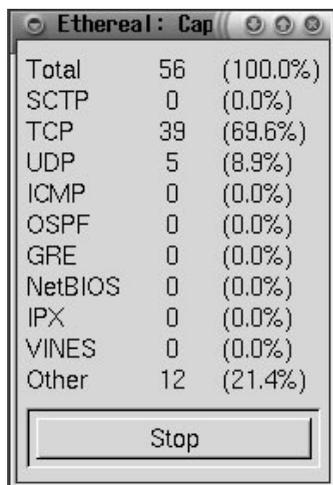
Etherealはほとんどのディストリビューションに含まれている。インストールにはGTK+とlibpcapが必要だが、rpmコマンドでインストールすればすぐに使えるようになる。2001年夏の時点で最新版は0.8.19だが、古いバージョンでも基本的な部分は大きく変わっていない。

コマンドラインから「ethereal &」などのコマンドで起動するが、ディストリビューションによってはネットワークインターフェイスをpromiscuousモードにするためにrootの権限で起動する必要があるかもしれない。

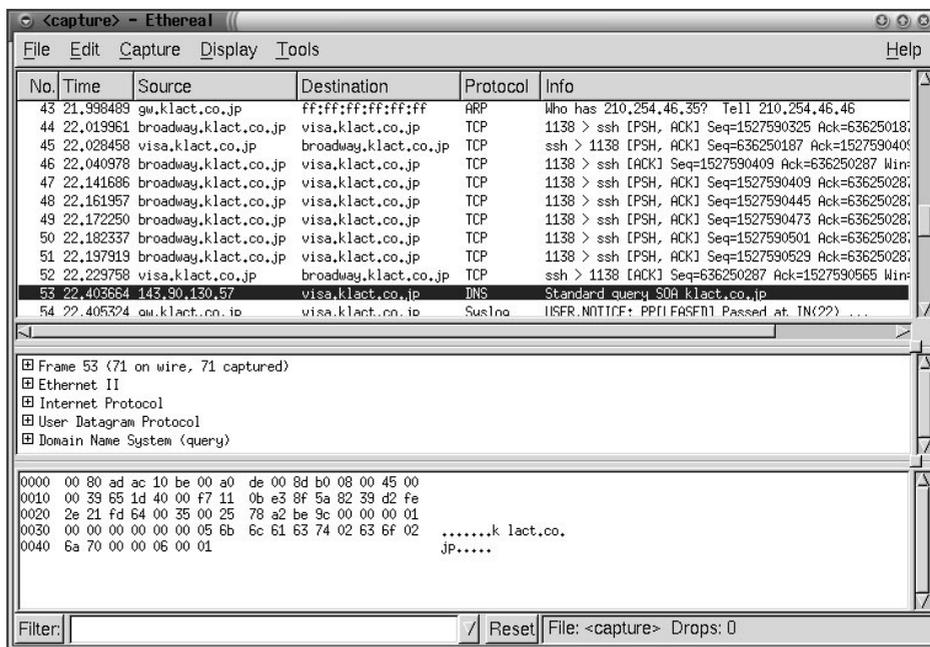
起動後に、CaptureメニューからStartを選び、ダイアログのOKボタンをクリックすると(画面1)、指定されたインターフェイスがつながっているネットワークのパケットの収集を始める(画面2)。適当なところでStopボタンをクリックすると、メインのウィンドウに収集したパケットの情報が表示される(画面3)。上部はパケットの一覧が表示され、中段にはパケットの解析結果、そして一番下にはパケットの16進ダンプが表示される。

各ウィンドウの表示は、特に説明の必要もないと思うが、上段の一覧から選択したパケットの情報が中、下段に表示される。中段は、パケットのブロックごとに階層表示されるので、必要な部分をクリックすると、下段の16進ダンプの対応する部分が強調表示される。これを見れば、そのパケットがどのようなパケットで、どのようなデータを含むのかが一目瞭然だ。

しかし、トラフィックの多いネットワークでの監視や、偶発的なパケットを捕捉するために長時間の監視を行う場



画面2 キャプチャ状況の表示



画面3 キャプチャ結果

合、すべてのパケットを記録していると目的のデータ以外の膨大なパケットを拾ってしまい、画面が見にくくなり、バッファが一杯になって記録できなくなってしまう。たとえば、DNSの問い合わせと回答のパケットを調べようと思っても、問い合わせに対する回答のパケットの間に無数の無関係のパケットや、ほかのDNSの問い合わせが表示されてしまうかもしれない。また、混雑した高速ネットワークに遅いマシンをつないで使うような場合、パケットのロスが発生する可能性もある。

このような場合に、必要なデータだけを表示するディスプレイフィルタの機能、および必要なパケットだけを記録するキャプチャフィルタの機能を利用する。

家庭内のネットワークのようにあまりトラフィックの多くないネットワークなら、とりあえずすべてのパケットを拾い、ディスプレイフィルタで必要な情報だけを表示させるようにするといい。しかし、ある程度のトラフィックがあるネットワークでは、取得するパケット自体をキャプチャフィルタである程度絞り込んでおかなければならない。

ディスプレイフィルタは、画面左下の「Filter :」に必要なフィルタの式を入力するか、EditメニューのDisplay Filtersであらかじめ定義する。

基本的な書式は、単にプロトコルの名称を指定するか、「== (等しい)」、「!= (等しくない)」、「>= (以上)」、「<= (以下)」、「> (大きい)」、「< (小さい)」の比較記号でフィールドの属性を指定する。たとえば、単に「dns」と入力すればDNSのパケット、「telnet」と入力すればtelnetプロトコルのパケットのみを表示する。また、「ip.src == 192.168.0.1」と入力すると、パケットのソースアドレスが192.168.0.1のパケットだけを表示する。さらに、「&& (論理積)」、「|| (論理和)」、「! (否定)」、「^^ (排他的論理和)」の記号で式を結んで「(ip.src == 192.168.0.1) && dns」と入力すると、ソースが192.168.0.1のDNSに関するパケットのみを表示する。フィールドの属性については、Etherealのマニュアルやヘルプなどを参照していただきたい。

次に、キャプチャフィルタだが、EditメニューのCapture Filtersに登録するか、キャプチャを開始するときに表示されるウィンドウのFilterに入力する。

書式はディスプレイフィルタと異なり、pcapライブラリの書式 (tcpdumpのフィルタと同じ書式) を使って指定する。詳しい書式についてはtcpdumpのマニュアルを参照していただきたいが、たとえばソースが192.168.0.1のDNSに関するパケットのみを表示するのであれば、「src

host 192.168.0.1 && port dns」を入力すればよい。

このほか、Etherealにはさまざまなオプション機能が装備されているが、便利な機能のひとつにTCPセッションを追跡してテキスト表示する機能がある。HTTP、NNTP、POP3またはSMTPなどのパケットを選び、ToolsのメニューからFollow TCP Streamを選択すると、サーバとクライアントの間で交換されるテキストのコマンドやデータがすべて表示される。

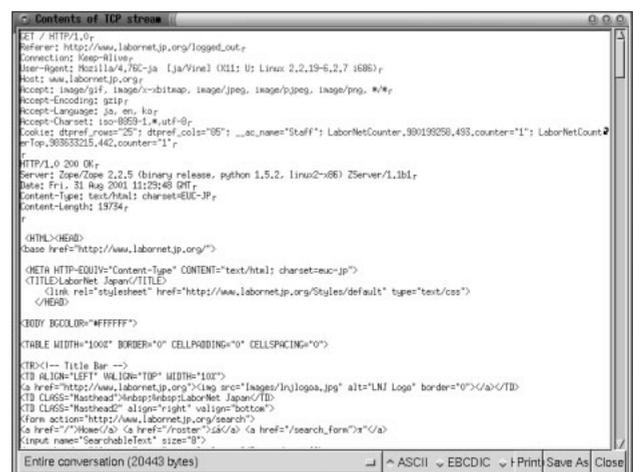
画面4は、EtherealのFollow TCP StreamでWWWの接続を表示させた例である。パケット一覧とTCPストリームの追跡を見ることで、クライアントからのHTTP要求とサーバからのHTTP応答で1つのTCPセッションが構成され、ほかの接続は別のTCPセッションで接続されていることがわかる。

netcatを使う

Etherealのウィンドウでさまざまなパケットを観察するだけでも、いろいろと勉強になるが、Follow TCP StreamでWWWやSMTPなどのコマンドのやりとりを観察していると、インターネットのサーバといってもかなりシンプルな仕組みで動いていることが理解できるのではないだろうか。

前回、telnetを使ってテキストベースのTCPプロトコルを使うサーバに接続して、サーバからの反応を得る例を紹介したが、今回はさらに一歩進んで、netcatを使ってクライアントからのコマンドに反応する簡単なサーバを作り、どのようにプロトコルが組み立てられているのかを実際に調べてみよう。

netcatは小さなツールでインストールも簡単で、コマンド名やパッケージ名はncという名前になっている。なお



画面4 EtherealのFollow TCP Stream機能でセッション内容を表示

現在、netcatの開発は進められていないらしく、数年前からバージョンは1.10のままである。また、Red Hatをはじめとするいくつかのディストリビューションは、netcatのコンパイル時にコマンド実行(-e)オプションが外されている。このオプションはサーバの実験には便利なオプションなのだが、安易に使用するとシステムに大きなセキュリティホールを空けることになるため、標準的な配布では外されているのだろう。そのため、今回は-eオプションを持たないnetcatを前提として説明する。

netcatの主な機能は、TCPおよびUDPで任意のアドレスとポートに任意のデータを送受信することで、ちょうどUNIXのcatコマンドのように、インターネットのサーバにデータを送り込むことができるツールといえる。さらに、netcatはソースルーティングの機能を持ち、送信元アドレスを任意に設定することができるほか、ポートスキャンの機能も持っている。これらの機能は、サーバの動作を調べるときに大いに役に立つ機能で、管理者にとって便利なツールだが、システムのセキュリティを突破して、バックドアを仕掛けるときなどにも大いに役に立ってしまう。もっとも、前に紹介したEtherealやtcpdumpのようなツールのように、ネットワーク管理者に便利なツールはクラッカーにとっても便利なツールであり、netcatが特に危険なツールというわけではないが、使用には十分に注意すること。もちろんこれらのツールの悪用は慎むべきであることはいうまでもない。

さて、netcatの使い方だが、基本的な書式は次の通り。

nc [オプション][ホスト][ポート]

主なオプションには、以下のようなものがある。そのほかのオプションについてはヘルプを参照のこと。

-l	待ち受けモード
-p ポート	接続を受けるローカルポートの指定
-u	UDPによる接続
-v	冗長メッセージの出力
-w 秒数	タイムアウトの設定
-h	ヘルプの表示

具体的な使い方としては、たとえば、example.comのHTTPポートに接続するのであれば、「nc example.com 80」を実行すればよい。また、ローカルマシンのポート80でTCP接続を待ち受ける場合は「nc -l -p 80」というこ

とになる。

TCPサーバの実験：HTTP接続を受け取る

では、実際にnetcatを使ってクライアントからの接続を受け付けてみよう。

サンプルとして、TCPを使ったプロトコルの中でも、最もシンプルなプロトコルのひとつであるHTTPを使うことにする。もっともフルセットのHTTPは、十分に複雑なプロトコルなのだが、前のEtherealの例でもわかる通り、1回のTCPセッションで要求と応答が完結するので応答部分のスク립トが簡単なもので済む。

詳しいHTTPの仕様については、RFC 2616 (HTTP / 1.1)に書かれている。この中で、HTTPの要求の書式はSection 5に、応答はSection 6に書かれている。

Section 5で、HTTP要求は要求行、および0以上のジェネラルヘッダ、リクエストヘッダ、エンティティヘッダに続き、空行(CRLF)を1行入れ、必要であればメッセージボディが続くとされている。また、要求行は「メソッド(空白)リクエストURI(空白)HTTPバージョンCRLF」で構成される。

Section 6のHTTP応答は、HTTP要求の先頭にある要求行がステータス行に代わる以外は要求と同じ形式であ

リスト1 サーバ機能を実現するシェルスク립ト

```
#!/bin/sh
until [ -s exp.tmp ]
do
    sleep 1
done

if grep ^GET exp.tmp >/dev/null 2>&1
then
    FILE=`grep ^GET exp.tmp | awk '{print $2}'`
    echo HTTP/1.0 200 Okey
    if expr $FILE : '.*html$'
    then
        echo Content-type: text/html
    else
        echo Content-type: text/plain
    fi
    echo Content-type: text/html
    echo
    cat /var/www/html$FILE
else
    echo HTTP/1.0 400 Bad Request
    echo Content-type: text/plain
    echo
    echo This server responds only GET.
fi
killall nc
```

る。ただし、ステータス行は、「HTTPバージョン（空白）ステータスコード（空白）理由 CRLF」になる。

これらの仕様から、クライアントから受け取ったHTTP要求のうち、要求行に含まれるメソッドとリクエストURIから、適当なステータス行とともに必要なファイルを送り返すことで、とりあえず必要最低限のサーバ的機能が実現できそうだということがわかる。そこで、リスト1のようなシェルスクリプトを作成してみた。

このスクリプトは、次のようにncコマンドと組み合わせて使う。

```
rm -f exp.tmp; sh exphttp.sh | nc -l -p 2000 > exp.tmp
```

exp.tmpは一時ファイル、exphttp.shは実験スクリプトの名前で、netcatはポート2000で待ち受けている。また、HTMLファイルのルートディレクトリは「/var/www/html」である。さてこのスクリプトが動いているホストに対し、適当なブラウザから「http://ホスト名:2000/index.html」などの要求を送ると、1回だけドキュメントルートのindex.htmlを送り返し、スクリプトは終了する。

簡単なスクリプトなので特に解説するまでもなく、動作は一目瞭然だが、最初の部分ではカレントディレクトリのexp.tmpという一時ファイルがゼロバイトを超えるサイズになるまで、ループを空回ししている。netcatは、exp.tmpに出力をリダイレクトしているので、netcatがポート2000への接続を検出すると、ループから脱出する。そして、exp.tmpの要求行をgrepで調べ、GETなら要求されたドキュメントを送り返し、それ以外であればステータス行でステータスコード400とメッセージと、ボディ部分に書かれたtext/plain形式のメッセージを送り返す。そして最後に「killall nc」でnetcatを終了させ、セッションが切断される。

なお、要求されたファイルが「html」で終わっていれば、MIMEタイプにtext/htmlを、そうでなければtext/plainを指定して、ファイルの内容が適切に表示されるようにしている。画像やその他のファイルタイプが必要であれば、ここに適切な拡張子とMIMEタイプを書き加える。さらに、GET以外のメソッドもサポートしたい場合は、case文などでメソッドを調べればよい。

ただし、この簡単な実験サーバでは、1ページに画像を含み、複数のドキュメントが混在するページは正しく表示されない。ブラウザは、サーバが送り返した応答のハイパ

ーテキストを調べ、埋め込まれたドキュメントに対する要求を自動的に生成するのだが、要求が生成された時点で、この実験サーバは終了してしまっているからだ。これを正しく表示させるためには、サーバの終了時に一時ファイルを一旦削除して、同じスクリプトを繰り返すような工夫が必要になる。埋め込まれた画像への要求も処理できるようなコードを加えて、GET以外のメソッドも使えるようにすれば、ちょっとしたHTTPサーバの完成である。

一見、複雑に見えるHTTPサーバだが、基本的な部分は、小さなツールと簡単なスクリプトだけで実現することができるのである。

UDPデータグラム型プロトコルの実験：DNS要求の送信次に、やはりnetcatを使ってUDPのプロトコルのやりとりを実験してみよう。

DNSでは、簡単な問い合わせと回答はUDPのデータグラムに必要な情報が詰め込まれて送受信される。TCPのプロトコルにもデータグラムを使うものは少なくないが、DNSは問い合わせと回答が一往復で済む。そこで今回は非テキストデータのデータグラム型プロトコルの実験として、DNSを例に実際に問い合わせのデータグラムを作り、ドメイン名を問い合わせしてみることにする。

なお、ここでの実験はスクリプトを使わずに、適当なバイナリエディタを使って手作業でデータグラムを作り、それをnetcatでDNSサーバに送り、回答のデータグラムはEtherealで解読するものとする。回答をEtherealで解読する理由は、データグラム解読のデコーダを作るのが面倒という現実的な理由はともかく、DNSサーバからの応答が必ずしもUDPで返ってくるとは限らないため、TCPとUDPのポートを監視して回答を取得しなければならず、スクリプトが複雑になってしまうからだ。

netcatでUDPデータグラムを送信するのは簡単で、次のようにデータグラムのファイルをパイプでnetcatに送り込めばいい。

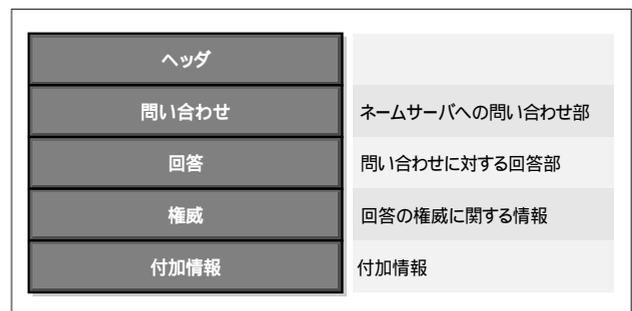


図1 DNSのデータグラム

```
nc -u DNSホスト 53 < datagram
```

問い合わせに対するネームサーバからの回答を捕まえるために、netcatでデータグラムを送る前に、Etherealのキャプチャフィルタに次のルールを加え、パケットのキャプチャを開始しておく。

```
src host DNSホスト && port dns
```

さて、問題はデータグラムの作り方だが、例によってRFCを参照してみよう。DNSに関する仕様はRFC 1035に掲載されている。DNSについての知識がないと少々読みづらいが、データグラムの構造はSection 4のMessagesに書かれている通りだ。RFC 1035によれば、データグラムは問い合わせ、回答とも図1のような形式になっている。

RFCによれば、ヘッダ部は常に必要で、ヘッダにはその他の情報に関するフィールドや、そのデータグラムが問い合わせなのか応答なのかを示す情報などが含まれている。

問い合わせ部には、問い合わせのタイプ(QTYPE)、問い合わせのクラス(QCLASS)、および問い合わせるドメイン名(QNAME)が含まれる。なお、QTYPEはレコードのタイプで、通常のホストのIPアドレスに関するAレコードのほか、メール配信のレコードを表すMX、正式名称のCNAME、ネームサーバのためのNSなどのタイプがある。QCLASSは、ほとんどの場合、インターネットクラスを表すINと考えてよい。また、QNAMEは問い合わせるドメイン名そのものだが、ドットで区切られた部分ごとに長さを示す1バイトの符合なし整数とアスキー文字で構成される。

回答部、権威部と付加情報部は、回答で使われ、その回

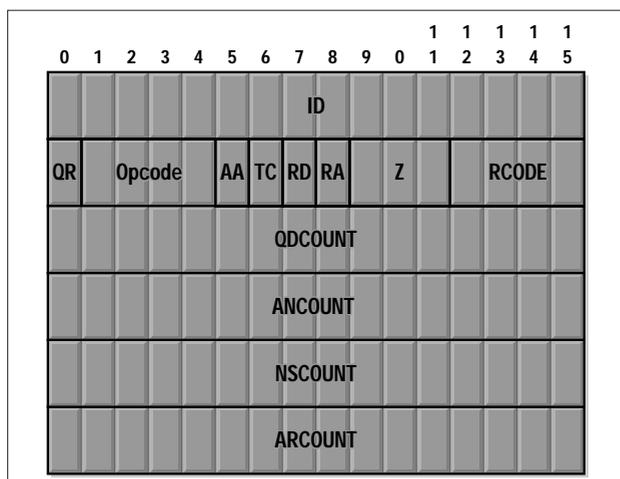


図2 DNS データグラムのヘッダ部

答に関する情報が含まれている。今回作る問い合わせのデータグラムには不要なのでここでは解説は省略するが、興味があれば直接RFC 1035を参照していただきたい。

なおヘッダ部には、問い合わせる情報に関する一種のメタデータが含まれており、その形式は図2の通りである。

ここで、それぞれのフィールドには次のような情報が収められる。

- ID
クライアントが生成する16ビットの識別子で、問い合わせに対する回答のヘッダには同じIDがコピーされ、どの問い合わせに対する回答なのかを識別できるようにする。

- QR
データグラムが問い合わせなら0、回答なら1が入る1ビットのフィールド。

- OPCODE
4ビットのフィールドで、問い合わせの種類に関する情報が含まれる。このフィールドが0ならドメイン名からIPアドレスを検索する問い合わせ、1なら逆引き、2ならサーバのステータスの問い合わせで、3から15までは未使用だ。

- AA
回答のデータグラムで使われるフィールドで、回答を返したネームサーバがそのドメインに対して権威を持っているかどうかを1/0で表す。

- TC
このビットが1ならトランケーションがあることを示す。通常、回答のデータグラムがUDPの制限を超えた場合にセットされる。

- RD
再帰的な問い合わせが必要であれば、このビットがセットされる。ほとんどの場合、ここは1。

- RA
回答時にセットまたはクリアされるビットで、再帰的な問い合わせをサポートするかどうかを示す。

- Z
未使用。常に0でなければならない。

• RCODE

回答で使われるレスポンスコード。意味は以下の通り。

- 0 エラーなし
- 1 問い合わせの書式のエラー
- 2 サーバのエラー
- 3 存在しないドメイン名
- 4 要求された問い合わせは未サポート
- 5 問い合わせを拒否
- 6 ~ 15 予約

• QDCOUNT

問い合わせ部に含まれる問い合わせの数がセットされる。

• ANCOUNT

回答部に含まれる回答のレコード数がセットされる。

• NSCOUNT

権威部のNSレコードの数がセットされる。

• ARCOUNT

付加情報部に含まれる付加レコードの数がセットされる。

これらの情報から、ヘッダと問い合わせ部にどのような情報をセットすればいいのかがわかる。

具体的には、まずIDには適当な16ビットの数を入れる。次にQR、Opcode、AA、TCは0、RDが1、そしてRA、Z、RCODEはそれぞれ0が入る。つまり、この部分は2進数なら00010000、16進数で0x0100になる。

次に、QDCOUNTは問い合わせの数になるが、今回は1つのドメインだけを検索するので1、QDCOUNT以外は問い合わせ時には0をセットすればよい。

そして問い合わせ部に検索したいドメイン、ここではwww.ascii.co.jpをセットすることにしよう。この場合、wwwとascii、co、jpの各部分ごとに長さのアスキーコードをセットし、ヌル(0)で終わる。また、QTYPEにはAレコードを表す1、QCLASSはインターネットクラスを表す1をそれぞれセットする。具体的には、問い合わせるドメインは、16進で「0x03 0x77 0x77 0x77 0x05 0x61 0x73 0x63 0x69 0x69 0x02 0x63 0x6f 0x02 0x6a 0x70

リスト2 作成したDNSデータグラムの16進ダンプ

```
66 66 01 00 00 01 00 00 00 00 00 03 77 77 77
05 61 73 63 69 69 02 63 6f 02 6a 70 00 00 01 00
01
```

0x00」の文字列となり、後続の2つのフィールドはどちらも0x0001となる。

こうして作ったデータグラムの全体をリスト2に示す。この内容をbeavやghex、hexedit、bviなどのバイナリエディタを使って入力し、適当な名前を付けて保存する。あるいはxxdがあれば、リスト2の内容を適当なファイルにそのまま入力し、「xxd -g1 -r -p -s off 入力ファイル名 出力ファイル名」を実行する。

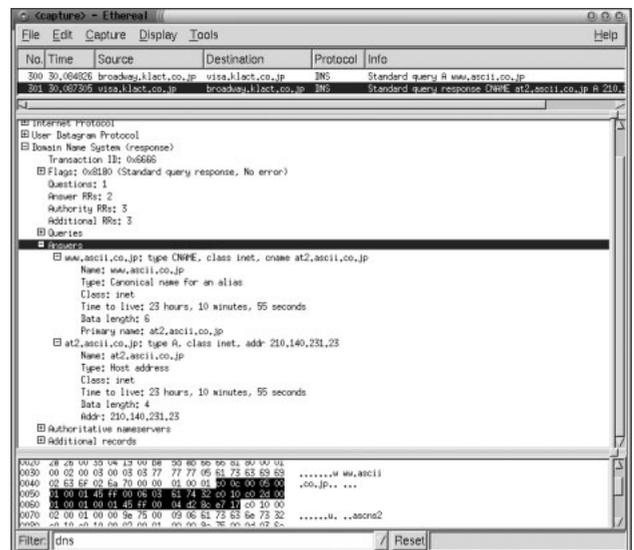
データグラムのパケットができれば、前述したように、Etherealを起動して次のコマンドを実行する。

```
nc -u DNSホスト 53 < ファイル名
```

画面5は、Etherealの画面から、DNSのパケットを探して表示させたものである。実際に、どのようなデータが返ってきているのかが、実際に自分のマシンで確かめていただきたい。

以上、今回はEtherealやnetcatといったネットワーク管理ツールを使って、プロトコルを実際に観察し、ストリーム型プロトコル、データグラム型プロトコルのやりとりを体験した。単にRFCを読むだけ、あるいは解説を読むだけよりも、インターネットのサーバやクライアントが何をしているのか、そしてインターネットのプロトコルとはどういうものなのかを実感として理解できたのではないだろうか。

次回は、サーバの運用のための実際的な設定ファイルに関する基本的な知識について紹介することにしよう。



画面5 DNSサーバからの回答

Oracle8iで作るWebサイト入門

PHPが理解できたところで、いよいよOracle8iデータベースにアクセスします。とはいってもデータベースへのアクセスはOracleに限らずほとんど共通しています。データベースは難しいなんて思っていた方も、この連載を読めば意外と簡単だと感じることでしょ

第4回 PHPでデータベースを使う

文: おもてじゅんいち / かざぐるま

Text: Junichi Omote/Kazaguruma

Oracle8iはRDB(リレーショナルデータベース)というタイプのデータベースで、現在主流となっているのがこのRDBです。DB2やSQL Server、フリーのPostgreSQLなどほとんどがRDBです。RDBは基本的にテーブル(表)の形式でデータを格納します。RDBのR(リレーション)とは複数のテーブル間で関連したデータを持つことができるという意味になります。

Oracle8iのおおまかなデータベース構造は図1のようになっています。データベースサーバはデーモンとして動いており、データ本体とは切り離れていて、サーバが扱うデータベース(データ本体)を切り換えたり、複数のデータベースを同時に動かすこともできます。インストール直後にはorclという名前のデータベースひとつだけが動いていますが、新しいデータベースを作って動かすことができます。/etc/oratabがデータベースの起動設定です。データベースはひとつの目的を持つシステムごとにひとつと考えればよいでしょう。

データベースには複数のユーザーを登録できます。このユーザーがいろいろな権限を持ってデータベースにアクセスします。たとえばユーザーやテーブルなどを新規作成できる管理者ユーザー、データの更新が行えるユーザーやデータを閲覧することしかできないユーザーなどを登録しておけます。データを実際に格納するのはテーブルですが、テーブルは原則的にいずれかのユーザーのスキーマに属します。スキーマとはユーザーごとの領域のようなもので、

そのスキーマ内にテーブルを作成していくことになります。つまりすべてのテーブルにはその所有者(ユーザー)がいることになります。

テーブルは文字どおり表の形をしていると考えてください。テーブルには列(Column)と行(Row)があります。他のデータベースでは列と行をそれぞれフィールドとレコードと呼ぶこともあります。列には必ず列名がついていて、その列に格納されているデータの種類、目的は同じでなければなりません(図2)。つまり表とはいってもそれは概念的なもので、表計算ソフトのように自由にセルを使えるわけではないのです。

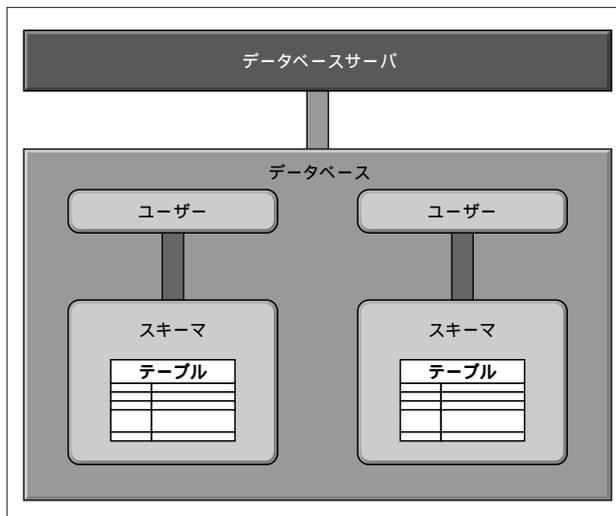


図1 Oracle8iデータベースの構造

SQL

テーブルのデータを読み書きしたり、テーブルを作成するなどデータベースに何らかの命令を与えるためにSQL (Structured Query Language) という言語が使われます。

これはOracle8iに限ったことではなく、ほとんどのRDBで使われている標準言語です。SQLが普及しているおかげで、データベースごとに言語を覚えたりしなくても、一度SQLをマスターすれば、データベースの種類を問わずに扱うことができるようになります。逆に、最近のデータベースを扱う場合、SQLを知らないと困ることがけっこうあります。

SQLにはかなり多くのデータベース操作機能がありますが、ここではとりあえず最低限データを読み書きできるだけの手続きを学習します。Webアプリケーションに限らず、データベースのデータを読み書きするときに必要なのは、SQLの文法でいうと、以下の4つになります。

SELECT.....データの読み出し
 INSERT.....データの新規追加
 UPDATE.....データの更新
 DELETE.....データの削除

もちろんSQLの機能はこれだけではありませんが、とりあえずこれだけ使えるようになれば、データベースを使うことができるようになります。

SQL * Plusを使う

実際にOracle8iを使ってSQLを試してみましょう。本連載では最終的にPHPスクリプトがOracle8iに対してSQLを発行することになりますが、いきなりスクリプトを書かなくても、手でSQL文を入力してOracle8iを動かす方法があります。

列	顧客コード	商品コード	商品名
行	1001	34	月刊アスキー
	1015	17	Linux business
	⋮	⋮	⋮
	⋮	⋮	⋮

図2 テーブルの構造

本連載第1回のOracle8iのインストールのところで、SQL * Plusというツールを使いましたが、このSQL * PlusがOracle8iにSQLを発行するツールなのです。さっそくコマンドラインでsqlplusと入力してSQL * Plusを起動してください。

SQL * Plusを起動すると、起動メッセージのあと、ユーザー名とパスワードを聞いてきます。そこで、

```
ユーザー名    scott
パスワード    tiger
```

と入力してください。

```
SQL>
```

というプロンプトが表示されます。これがSQL文の入力待ちの状態です。ここで、quitと入力するとSQL * Plusは終了します。

このようにSQL * Plusを起動してからユーザーを毎回指定するのもめんどろなので、SQL * Plusを起動するときに、

```
sqlplus scott/tiger@orcl
```

と入力するようにしましょう。これでユーザー名、パスワード、使用するデータベース名を指定できます。

これから使うデータベースはインストールのときに、「CDから既存データベース・ファイルをコピー」したものです。これにはOracle8iに付属のサンプルデータベースが含まれています。scottはそのサンプルデータベースに登録済みのユーザーで、パスワードはtigerです。インストール時に作ったデータベース名はorclでした。つまり、SQL * Plusは、

```
sqlplus (ユーザー名)/(パスワード)@(データベース名)
```

として起動することができるのです。

このようにユーザー名とデータベース名を指定してSQL * Plusを起動すると、すでにデータベースにログオンしている状態ですから、すぐにSQLを発行できます。

「sqlplus scott/tiger@orcl」と入力して、SQL * Plusを起動してください。

SQL*Plus: Release 8.1.6.0.0 - Production
(c) Copyright 1999 Oracle Corporation.

Oracle8i Release 8.1.6.1.0 - Production
JServer Release 8.1.6.0.0 - Production
に接続されました。

のような感じで表示されればOKです。もしエラーメッセージが出たら、Oracle8iが無事にインストールされていないか、データベースが起動していませんから、マニュアルか本連載第1回のインストールと起動の項を確認してみてください。

SELECT文

それではSQLを使ってみましょう。まずはデータを読み出すSELECT文です。SELECT文は、

```
select 列名リスト from テーブル名
      [where .....][order by .....];
```

```
SQL> select empno,ename from emp;

  EMPNO ENAME
-----
   7369 SMITH
   7499 ALLEN
   7521 WARD
   7566 JONES
   7654 MARTIN
   7698 BLAKE
   7782 CLARK
   7788 SCOTT
   7839 KING
   7844 TURNER
   7876 ADAMS

  EMPNO ENAME
-----
   7900 JAMES
   7902 FORD
   7934 MILLER

14行が選択されました。
```

画面1 selectの実行

```
SQL> select * from emp;

  EMPNO ENAME      JOB              MGR HIREDATE          SAL          COMM          DEPTNO
-----
   7369 SMITH      CLERK            7902 80-12-17          800           0              20
   7499 ALLEN      SALESMAN         7698 81-02-20          1600          300            30
   7521 WARD      SALESMAN         7698 81-02-22          1250          500            30
      :
```

画面2 select *の実行

という書式になります。サンプルデータベースには、empというテーブルがあり、empテーブルにはempnoとenameという名前の列がありますから、

```
select empno,ename from emp;
```

と入力します(最後のセミコロンを忘れないように)。どうでしょうか、画面1と同じものが表示されたでしょうか。表示されないときは、SQLに入力ミスがないかをよく調べてください。

このSELECT文は、「empというテーブルからempnoとenameという名前の列のデータを取り出しなさい」という意味になります。列は指定していますが、何件のデータを取り出すかという行方向の指示はしていないので、そのテーブルに含まれるすべてのデータが表示されるのです。行を限定したければ、

```
select empno,ename from emp where empno=7900;
```

と入力してみてください。where句は行を抽出する条件を指定するためのもので、この場合はempnoが7900であるJAMESだけが表示されます。「empno < 7900」というような指定も可能です。

また、データ順を指定するorder by句を使って、

```
select empno,ename from emp order by ename;
```

と入力してみましょう。先ほどはempno順に並んでいたのですが、今度はename順に並びます。

画面1では列名にempnoとenameだけを指定しましたが、ここを「*」にすると、すべての列が表示されます(画面2)。ターミナルの画面によっては途中で改行されるので、ちょっと見にくい表示になることがあります。

INSERT文

データを新規に追加するにはINSERT INTOという文

を使います。テーブルには新しい行が追加されることになります。書式は、

```
insert into テーブル名 (列名リスト)
          values (値のリスト);
```

となります。列名リストと値のリストの個数は同じでなければエラーになります。ではデータを1行追加してみます。

```
insert into emp (empno,ename) values (9001,'JUN');
```

エラーがなければ、「1行が作成されました。」というメッセージが表示されます。結果を確かめるために、

```
select empno,ename from emp;
```

と入力してください。画面3のような表示になればOKです。empno順に表示されますので、最後の行に、

```
9001 JUN
```

が表示されて、データが15行になっています。

(列名リスト)と(値のリスト)のそれぞれの中身の順番を対応させるように気をつけてください。

UPDATE文

テーブル内に存在するデータの内容を更新(修正)するのがUPDATE文です。書式は、次のようになります。

```
UPDATE テーブル名 SET 列名=値 [,列名=値, .....]
                    [where .....];
```

```
SQL> insert into emp (empno,ename) values
(9001,'JUN');
```

1行が作成されました。

```
SQL> select empno,ename from emp;
```

```
EMPNO ENAME
-----
7369 SMITH
7499 ALLEN
   :
7934 MILLER
9001 JUN
```

15行が選択されました。

画面3 insertの実行

UPDATE文には必ずSET句が必要です。where句はオプションですが、where句がない場合、すべての行のデータが更新されてしまいますので、実際にはwhere句といっしょに使う場合がほとんどです。

先ほど追加したデータの内容を更新してみます。

```
update emp set ename='JUNICHI' where empno=9001;
```

つまり、empno=9001の行(データ)のenameの値をJUNICHIに変更せよということになります。実行して確認したものが画面4です。

DELETE文

最後にデータ(行)の削除です。書式は、

```
delete [from] テーブル名 [where .....];
```

です。これもwhere句がないとすべてのデータが削除されてしまいますから、必ずwhere句ありと覚えておくほうがよいでしょう。Oracleの場合fromはあってもなくてもかまいません。ではempno=9001のデータを削除してみます。

```
delete emp where empno=9001;
```

画面5で確認してください。

以上、4種類のSQL文を使ってみました。SELECT、INSERT、UPDATE、DELETEについてもまだまだここで紹介した以上のいろいろな使い方があります。あくまでもここでは、SQLとはどんなものか実際に動かしてみようための説明しかしていません。あとはSQLの入門書や解説書を参照して、ぜひSQLの理解を深めてください。

```
SQL> update emp set ename='JUNICHI' where
empno=9001;
```

1行が更新されました。

```
SQL> select empno,ename from emp;
```

```
EMPNO ENAME
-----
7369 SMITH
7499 ALLEN
   :
7934 MILLER
9001 JUNICHI
```

15行が選択されました。

画面4 updateの実行

PHPでOracle8iを使う

SQLでデータの読み出しや追加ができれば、あとはそれをPHPで実現するだけです。SQL文自体はデータベースが変わってもほとんど共通なのですが、サーバサイドスクリプト(PHP)側とデータベースサーバとのやりとりは、採用するデータベースサーバによって違ってきます。

そのためPHPには、各データベースサーバ用の関数が用意されています。Oracle用には、Oracle7までの関数群とOracle8用の関数群が用意されていますので、ここではOracle8用関数を使います(表1)。

Oracle8用関数はOCI(Oracle Call Interface)関数とも呼ばれ、関数名がOCIから始まります。OCI関数はOracle8i側が用意しているインターフェイスにもとづいて作られていますから、さまざまな機能の関数がありますが、ここではそのうちの、

OCILogon
OCIParse
OCIDefineByName
OCIExecute
OCIFetch
OCIFreeStatement
OCILogOff

を使います。

ではリスト1で実際のスクリプトを見てみましょう。リスト1はSQL*Plusで行った画面1と同じ内容です。実行結果は画面6のようになるはずですが、

OCILogon、OCILogOffは、データベースに対する接続

```
SQL> delete emp where empno=9001;
```

1行が削除されました。

```
SQL> select empno,ename from emp;
```

```
EMPNO ENAME
-----
 7369 SMITH
 7499 ALLEN
      :
 7902 FORD
 7934 MILLER
```

14行が選択されました。

画面5 deleteの実行

を確立/切断します。OCILogonでは、「ユーザー名」「パスワード」「データベース名」を指定します(リスト1の)。前述のSQL*Plus起動時の指定と同じですね。OCILogOffはデータベースを使用しなくなったときに実行します。

データを読み出すためにSQL文を送るのですが、SQLはまずOCIParseで解釈してもらいます(リスト1の)。OCILogonで返ってきたコネクションID(リスト1の \$conn)とSQL文が引数になります。

OCIParseはステートメントID(リスト1の \$stmt)を返します。これはSQL文にエラーがあればFalseになり、エラーがなければ次のOCIExecuteでこのSQLを実行するための手続き番号のようなものとして使われます。

ここでは特にSQL文がエラーだった場合の処理はしていませんが、実運用する場合はエラー処理も書いておきましょう。

SQLでSELECT文を発行する場合のみ、あとで読み出

Oracle8関数	
OCIDefineByName	SELECT実行中、定義用のPHP変数を使用する
OCIBindByName	Oracle プレースホルダにPHP変数をバインドする
OCILogon	Oracleへの接続を確立する
OCIPLogon	持続的接続を使用してOracleデータベースに接続、ログオンする。新しいセッションを返す
OCINLogon	Oracleデータベースに接続し、新規接続にログオン、新規セッションを返す
OCILogOff	Oracleとの接続を切る
OCIExecute	文を実行する
OCICommit	未解決のトランザクションをコミットする
OCIRollback	未解決のトランザクションをロールバックする
OCINewDescriptor	空の新規ディスクリプタLOB/FILE(LOBがデフォルト)を初期化する
OCIRowCount	作用された行の数を得る
OCINumCols	ある文における結果のカラム数を返す
OCIResult	取得された行におけるカラムの値を得る
OCIFetch	結果バッファの次の行を取得する
OCIFetchInto	結果配列の次の行を取得する
OCIFetchStatement	結果データのすべての行を配列に取得する
OCIColumnIsNull	結果カラムがNULLであるかどうかを確認する
OCIColumnSize	結果カラムサイズを返す
OCIServerVersion	サーバのバージョン情報を含む文字列を返す
OCIStatementType	OCI命令の型を返す
OCINewCursor	新規カーソル(命令ハンドル)を返す
OCIFreeStatement	ある文に関連する全リソースを開放する
OCIFreeCursor	カーソルに関連づけられた全リソースを開放する
OCIFreeDesc	ラジオブジェクト記述子を削除する
OCIColumnName	カラムの名前を返す
OCIColumnType	カラムのデータ型を返す
OCIParse	クエリをパースし、ステートメントを返す
OCIError	stmt、conn、globalに関する直近のエラーを返す。エラーが生じていない場合はFALSEを返す

表1 PHPのOracle8関数

した各列のデータを使うための変数の準備をします。それがOCIDefineByName関数です（リスト1の ）。

OCIDefineByNameを使うと、テーブルの列名と同じ名前の変数でデータを参照することができるようになります。

```
OCIDefineByName($stmt, "EMPNO", &$empno);
```

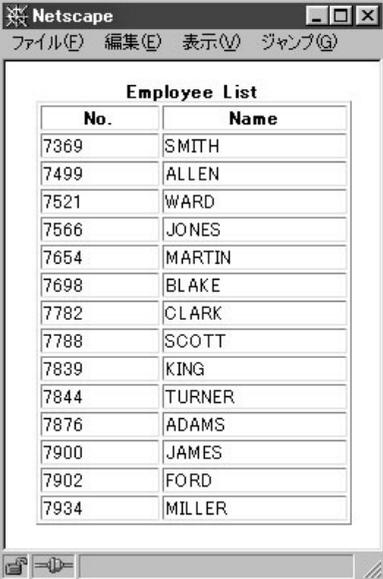
とすれば、\$empnoという変数の内容が、empnoという列のデータと同じになっています。ただしSELECT文なので、参照するだけで、テーブルのデータを更新することはできません。最初の引数はOCIParseで返されたステートメントIDです。

ここまでの準備が終わったら、OCIExecuteでSQLを実行します。SQL文自体はOCIParseで前もって渡してありますから、引数はステートメントID（\$stmt）だけです。

今発行したSQLは、前述のSQL * Plusの操作での画面1と同じでしたから、結果は14行のデータが返ってくるは

ずです。ではその14行のデータはどうやって受け取ればよいのでしょうか。

リスト1 のOCIFetchは、SELECT文によってデータ



No.	Name
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

画面6 select.php3の実行結果

リスト1 select.php3

```
<?php
    $conn = OCIlogon("scott","tiger","orcl"); .....
?>
<html>
<head>
<title>Employee List</title>
</head>
<body>
<center>
<b>Employee List<b><br>
<table border=1 width="50%">
<tr>
<th>No.</th><th>Name</th>
</tr>
<?php
    $stmt = OCIparse($conn,"select empno,ename from emp"); .....
    OCIdefinebyname($stmt,"EMPNO",&$empno); .....
    OCIdefinebyname($stmt,"ENAME",&$ename);
    OCIexecute($stmt); .....
    while(OCIFetch($stmt)) { .....
        print("<tr><td> $empno </td><td> $ename </td></tr>\n");
    }
    OCIfreestatement($stmt);
?>
</table>
</center>
</body>
</html>
<?php
    OCIlogoff($conn);
?>
```

ベースが読み出してくれたデータを1行ずつ受け取る関数です。受け取ったデータは、OCIDefineByNameで設定した列名と同名の変数に格納されています。変数は各列1個ずつしか用意してありませんから、OCIFetchは1回に1行分のデータしか読み込まないようになっています。

しかし、OCIFetchはもう一度実行すると次の行のデータを読み込むというように、次々とデータを読み込んでいってくれますから、最後の行が終わるまでループすれば、すべての行のデータを読み出すことができます。読み出す行がなくなるとOCIFetchはFalseを返しますので、

```
while(OCIFetch($stmt)) { .....
```

とすることで、すべての行を読み出してくれるようになります。ここでは、このループの各回で、各行のデータをHTMLの表の形にして出力しています。

SQLで指示した内容が完了したら、ステートメントIDは必要なくなりますから、OCIFreeStatementでステートメントIDを破棄します。後始末も忘れないようにしましょう。そしてスクリプトの最後でデータベースとの接続を切断しています。

リスト2 insert.php3

```
<?php
    $conn = OCIlogon("scott","tiger","orcl");
?>
<html>
<head>
<title>Employee List</title>
</head>
<body>
<center>
<b>Employee List<b><br>
<?php
    $stmt = OCIparse($conn,"insert into emp
(empno,ename) values (9001,'JUN')");
    if ($stmt) {
        OCIexecute($stmt);
        echo("OK<br>");
    }
    else {
        echo("Error<br>");
    }
    OCIfreestatement($stmt);
?>
</center>
</body>
</html>
<?php
    OCIlogoff($conn);
?>
```

PHPでのINSERT、UPDATE、DELETE

INSERT、UPDATE、DELETE文については、SELECT文よりも単純です。これらの文はデータの値自体をSQL文に含めて発行するため、SELECT文でのOCIDefineByNameのような準備をする必要がないからです。

リスト2がINSERTの場合です。SQL文自体はSQL * Plusのときとまったく同じですから、

```
OCIparse
OCIexecute
```

を呼び出すだけで、SQLが実行されます。ここでは、SQL文にエラーがあったときのエラー処理を行っています。SQL文にエラーがあったときはOCIparseが返すステートメントIDがFalseになっていますから、その場合はエラーメッセージを表示します。Falseでないときは「OK」と表示しています(リスト2のと)。

リスト2を実行したあと、もう一度リスト1を実行した結果が画面7です。データが1行追加されているのがわかります。ここではSQL文を固定して書き込んでいますが、実際にはこんなことはあまりなく、場合に応じてSQL文自体を変化させる必要があります。たとえば、フォームなどで入力されたデータを新規にテーブルに追加する場合は、フォームデータを変数に格納して、そのデータを含んだSQLを発行することになります。

フォームで入力された社員番号と名前がそれぞれ、\$f_no、\$f_nameという変数に格納されている場合、それ



No.	Name
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER
9001	JUN

画面7 insert後のselect.php3の実行結果

を新規追加するSQL文は、リスト3のように合成できます。 ” ” 中にある \$f_no、\$f_name は変数として認識されて、値に展開されます。

このように、 ” ” 内で変化する部分は変数名を書いておけば、場合に応じたSQL文を簡単に作ることができます。ただし注意しなければならないことは、もとのSQL文の値リストの部分が、

```
insert ..... values (9001,'JUN')
```

となっているので、\$f_no=9001、\$f_name='JUN'だとすると、

```
insert ..... values ($f_no,$f_name)
```

と書いてしまいそうになるのですが、これだと変数が展開されたときに、

```
insert ..... values (9001,JUN)
```

リスト3 SQL文を合成する

```
$sql = "insert into emp (empno,ename) values ($f_no,'$f_name')";
$stmt = OCIparse($conn,$sql);
```

リスト4 update.php3

```
<?php
    $conn = OCIlogon("scott","tiger","orcl");
?>
<html>
<head>
<title>Employee List</title>
</head>
<body>
<center>
<b>Employee List<b><br>

<?php
    $stmt = OCIparse($conn,"update emp set
ename='JUNICHI' where empno=9001");
    if ($stmt) {
        OCIexecute($stmt);
        echo("OK<br>");
    }
    else {
        echo("Error<br>");
    }
    OCIfreestatement($stmt);
?>
</center>
</body>
</html>
<?php
    OCIlogoff($conn);
?>
```

となってしまいます。「JUN」の部分がクォーテーションでくくられていませんね。これだとこのSQL文はエラーになります。正しくは、

```
insert ..... values ($f_no,'$f_name')
```

というように、変数自体をクォーテーションでくくっておかなければなりません。変数展開はあくまでも値(ここではJUN)のみに展開されるのです。

UPDATEとDELETEもSQL文を除いてはINSERTと変わりません(リスト4、リスト5)。ただこれらのスクリプトを実行してみれば感じるのですが、UPDATEにしてもDELETEにしてもけっこうあっけなく実行されてしまいます。更新や削除といった危険な作業にもかかわらずです。SQL文で操作を行う場合には、データベース側の機能としては親切に確認メッセージを出したりしませんから、更新や削除の場合には、スクリプトで確認メッセージを出すなどしてやり直しや中止ができるようにしたいものです。

リスト5 delete.php3

```
<?php
    $conn = OCIlogon("scott","tiger","orcl");
?>
<html>
<head>
<title>Employee List</title>
</head>
<body>
<center>
<b>Employee List<b><br>

<?php
    $stmt = OCIparse($conn,"delete emp where
empno=9001");
    if ($stmt) {
        $res = OCIexecute($stmt);
        echo("OK<br>");
        echo($res . "<br>");
    }
    else {
        echo("Error<br>");
    }
    OCIfreestatement($stmt);
?>
</center>
</body>
</html>
<?php
    OCIlogoff($conn);
?>
```

プログラミング工房

今月は、サーバ管理システムのまとめとして、ディスクパーティションやディスククォータの設定など、今まで解説しきれなかった部分について説明する。

第22回 サーバ管理プログラム(5)

文：藤沢敏喜
Text: Toshiki Fujisawa

ディスクパーティションの設定

最初に、ディスクパーティションの割り当てとクォータ設定について、少し詳しく解説しよう。ここでは、Vine Linux 2.0を使った場合について説明することにする。

今回のシステムでは、設定ファイルやデータファイルなど、すべてのファイルを/smbに置くことにしている。このディレクトリは、専用のパーティションとして構成するのが望ましいので、ここではIDEディスクの2番目のドライブである/dev/hdb1に割り当てる。まず、スーパーユーザーでログインし、マウントするディレクトリを、

```
# mkdir /smb
```

として作成する。次に、linuxconfを実行し、ローカルドライブへのアクセスを選択し、[追加]を選ぶ。そして、画面1のように入力して、[マウントを行う]を選択する。最後に、[変更]を実行してから、linuxconfを終了する。

この設定を行ったあとには、/etc/fstabの最後の行に、

```
/dev/hdb1 /smb ext2
noexec,nodev,nosuid,rw,grpquota,bsdgroups 1 1
```

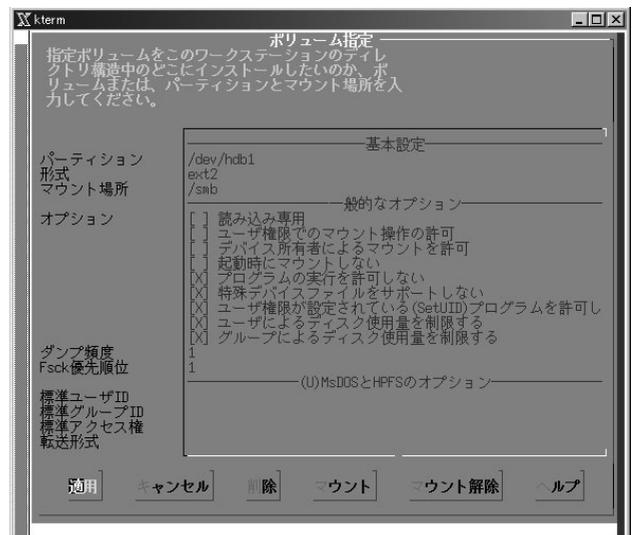
が追加されているはずである。



続いて、dfコマンドで/smbに/dev/hdb1がマウントされていることを確認し、

```
# touch /smb/quota.user /smb/quota.group
# /sbin/quotacheck -v -R -a
```

を実行する。リポートすると、/smbへのグループクォータが有効になり、ブートの途中で初期化作業が行われる。初期化作業にはパーティションのサイズとその中身によって多少時間がかかる場合があるが、正常に初期化が終了した場合は/var/log/messagesファイルに、



画面1 linuxconfの設定画面

```
rc.sysinit: Turning on user and group quotas for
local filesystems succeeded
```

というログが残っているはずなので確認しよう。

/smb ディレクトリの構築

ここまでで、/smb ディレクトリのマウントが終了し、ディスククォータの準備ができた。次は、各種の設定ファイルやユーザーのホームディレクトリなどを作成する。

構築には sedusr コマンドを使用するので、スーパーユーザーでログインし、付録CD-ROMをマウントする。そして、本連載のディレクトリに移動し、makeを実行する。

さて、/smb ディレクトリの構築である。これは、

```
# sedusr -x init
```

を実行するだけである。このコマンドを入力すると、**画面2**のような内容が実行される。この初期化では、システムに必要なディレクトリやファイルを作成し、その所有者と許可モードを適切に設定している。

```
mkdir -p /smb/home
chown sedusr.sedusr /smb/home
mkdir -p /smb/data
chown sedusr.sedusr /smb/data
mkdir -p /smb/logon
chown sedusr.sedusr /smb/logon
mkdir -p /smb/dust
chown sedusr.sedusr /smb/dust
chmod 700 /smb/dust
mkdir -p /smb/etc
chown sedusr.sedusr /smb/etc
chmod 700 /smb/etc
mkdir -p /smb/mail
chown sedusr.sedusr /smb/mail
mkdir -p /smb/ml
chown sedusr.sedusr /smb/ml
chmod 700 /smb/ml
touch /smb/etc/rpw.txt
chown sedusr.sedusr /smb/etc/rpw.txt
chmod 600 /smb/etc/rpw.txt
touch /smb/etc/alias.txt
chown sedusr.sedusr /smb/etc/alias.txt
chmod 600 /smb/etc/alias.txt
touch /smb/etc/dir_name.txt
chown sedusr.sedusr /smb/etc/dir_name.txt
chmod 600 /smb/etc/dir_name.txt
printf 'net time   \\\vm /set /yes\r\n' >
/smb/logon/STD.BAT
printf 'net use n: \\\vm\n\r\n' >>
/smb/logon/STD.BAT
chown sedusr.sedusr /smb/logon/STD.BAT
```

画面2 sedusr -x init で実行される内容

なお、Windows NT ログオン時に起動されるログオンスクリプトの生成もここでやっているが、ログオンスクリプトで指定しているサーバ名は、sedusr コマンドを実行したマシンのホスト名から自動的にサーバ名を抽出している。マシンのホスト名とこのサーバ名は同じほうが混乱が少なく望ましいが、もし異なる名前にしたい場合は、エディタで書き換える必要がある。

画面2は、手動でも実行できる内容ではあるが、数十台のマシンに設定する場合などにも設定ミスがなくなり便利である。また、数カ月後に新しいマシンを追加するといった場合には、前回設定した内容を忘れてることが多いが、このようなコマンドにしておく、筆者のようにもの忘れが激しい人間でも設定項目を忘れることがなくなる。

なお、Web ツールから sedusr コマンドを呼び出す場合は、管理者 (sedusr) 用パスワードが必要になるので、上記の「sedusr -x init」を実行したあとに、

```
# passwd sedusr
```

を実行して、「sedusr」という名前の管理者ユーザーのパスワードを設定しておく必要がある。

共有ディレクトリの作成

design という共有ディレクトリのグループIDを30002として作成するには、

```
# sedusr add_dir design,30002 -x
```

というコマンドを実行する。第1回目のコラムで解説したように、グループIDを決定するには、あとあとのことを考えて綿密な計画を立ててから決定するようにしておく。なお、このコマンドを「-x」オプションなしで実行すると、

```
groupadd -g 30002 design
echo design >> /smb/etc/dir_name.txt
mkdir -p /smb/data/design
chown sedusr.design /smb/data/design
chmod 700 /smb/data/design
```

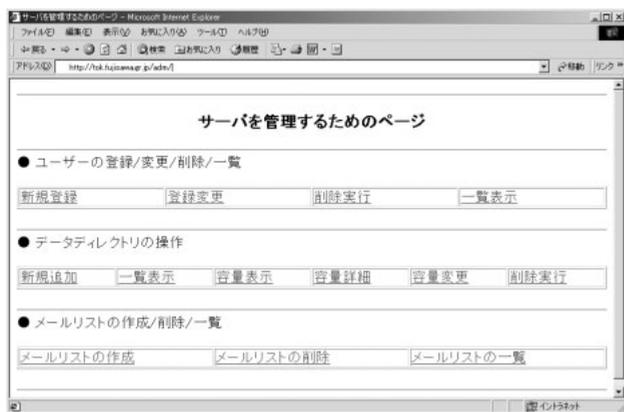
というように、実行されるコマンドが表示される。ここでは、説明の都合上、**画面3**に示すスクリプトで共有ディレ

rm \$TEMP

であり、sed コマンドを用いて制限容量の数値を書き換えている。なお、ここでは同時に複数実行された場合でも問題が発生しないように、プロセス番号に展開される「\$\$」を用いて、ユニークな中間ファイルに出力している。また、MBYTE という環境変数に制限容量を M バイト単位で渡し、sed コマンドで2行目 (2s) のイコール記号「=」とスペース「 」と数字 [0-9] の1文字以上の繰り返しを、\$MBYTE + 000 という文字列にグローバル変換 (s/旧文字列/新文字列/g) している。

ユーザーの登録と変更

今回のシステムでは、ユーザーの登録を Web ブラウザ

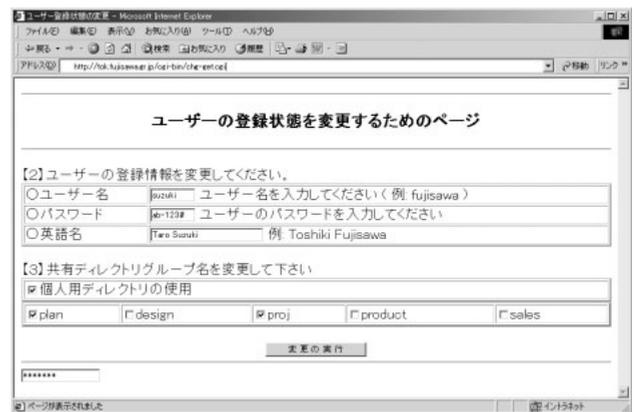


画面5 管理ページ

から行えるようになっている。今月号のプログラムでは、デフォルトの制限容量は25M バイトとなっているが、もしデフォルトの容量を変えたい場合は、DEFAULT_QUOTA_LIMIT_MBYTE というマクロを変更するといいだろう。

登録したユーザーが改姓した場合や、異動のためアクセスできるディレクトリを変更したい場合には、管理ページ (画面5) からユーザーの「登録変更」を選択する。そして、ユーザー変更の入力ページを選び、管理者のパスワードと、変更するユーザーのログイン名を入力する。さらに、ユーザー情報変更ページ (画面6) で、ユーザーの名前やアクセスできるディレクトリ名のチェックを行う。

なお、以前述べたように、管理の都合上登録と変更時にはユーザーの生パスワードをファイルに保存するようにしてある。これは、多数のユーザーを管理するうえでは便利



画面6 ユーザー情報変更ページ

Column

サーバ構築の実験

容量制限を実現するためのディスククォータを設定するには、パーティションを分けたりして、いろいろな実験が必要である。しかし、パーティションの変更をしたり、設定を間違ったりすると、システムが起動しなくなることもある。したがって、実験を行う場合は、壊れても大丈夫な専用のマシンを用意するほうがよいだろう。

なお、このような実験を行うには、PC/AT 互換機エミュレータである VMware を使うと便利である。VMware では、ボタンをクリックするだけで新しいパーティションを用意できる。また、そのパーティシ

ョンはホストとなる OS から見ると単なるファイルなので、簡単にオリジナルをバックアップしたり、パーティションの変更内容を書き換えないようにできる。そのため、OS の設定をいろいろと変えて実験するには都合がよい。

今回の記事を書くにあたっては、Windows 2000 が動作しているノート PC 上で、VMware を2つ起動して実験した。プログラムと原稿は、VMware で動作する FreeBSD 上の Emacs と、Windows 2000 上の X サーバを使って書き、Samba サーバや Web インターフェイスの実験は、もう1つの VMware 上で動作する Vine Linux 2.0 で行っている。

実験中には、/etc/fstab の中で/dev/hdb1

と書くべきところを/dev/hdb1として、Linux が起動しなくなったことがあった。通常的环境であれば、シングルユーザーモードで/etc/fstab を編集するなどの面倒な作業が必要になるが、今回は VMware なので Windows 2000 上で VMware の Linux パーティションファイルをコピーし直すだけで元に戻すことができた。

実は今、この原稿はハワイへ向かう飛行機の中で、ノイズキャンセラー内蔵のヘッドフォンで MP3 の音楽を聞きながら書いている。こういった状況でも、Windows から Samba 接続実験や、IE (Microsoft Internet Explorer) を使った Web ツールの確認実験などができるのはたいへん便利である。

Column

Sambaの代わりにFTPを使う

Sambaを使ったファイル共有は便利だが、次のように不十分な場合もある。

- (1) UNIX系OSやMacOSなどからのアクセス
- (2) セキュリティ上問題のあるネットワークでの使用
- (3) 遅い回線を使った場合の共有

(1)の問題は、Linuxのsmbmountやsharity-lightを使うことにより、UNIX系OSからアクセスできるようになるし、CAPを使うことによりMacOSなどからのアクセスも可能にはなるものの、面倒であることは確かである。

(2)と(3)は、特にインターネット上でのファイル共有を実現する場合に問題となる。盗聴は仕方ないとしても、Sambaサーバへの攻撃が気になる。

このような問題を解決するためには、Sambaではなく、FTPによるファイル共有を考えるとよいだろう。

FTPプロトコルは、多くの会社や学校のファイアウォールを通過できるので、複数の会社が参加するプロジェクトなどのファイル共有も可能である。また、もともと細い回線でも問題なく動作するように作られているし、古くからあるプロトコルなのでOSの標準ツールを用いて転送を行うこともできる。

そして何よりも、最近のブラウザではあたたかもローカルのハードディスクと同じような感覚でファイル転送を行うことができ

るので便利である。Internet Explorerのバージョン5.50では、

- ・ [ツール]メニューから [インターネットオプション] を選択する
- ・ [詳細設定] タブをクリックする
- ・ ブラウザの中の [FTPサイト用のフォルダビューを使用する] をオンにする
- ・ ブラウザの中の [ファイアウォールおよびDSLモデムとの互換性を保つためにPASVモードをサポートするFTPを使用する] をオンにする

という設定を行うことにより、ファイアウォールの中からでもファイルのアップロードやダウンロードが可能になる。

たとえば、IPアドレスが10.3.2.200のFTPサーバの/tmpというディレクトリに、fujisawaというユーザー名でログインしたい場合は、URLとして、

`ftp://fujisawa@10.3.2.200/tmp`

と入力すると、パスワード入力画面が現れる(画面7)。パスワードを入力すると、画面8のように普通のドライブにあるフォルダとまったく同じようなウィンドウが表示される。もちろん見かけだけでなく、コピーや削除、リネーム、移動などもファイルと同じ操作で行えるため、一般的なWindowsユーザーでもまったく問題なく使用できるはずだ。

また、ftp.fujisawa.gr.jpというホスト名のマシンにアクセスする場合、

`ftp://ユーザー名:パスワード@ftp.fuji`

`sawa.gr.jp/tmp`

というように、URLにパスワードを指定することもできる。この場合、このURLを送ることにより、だれでも簡単にアクセスできる(同じファイルをhttpでアクセスできるようにになっていることも多いだろう)。

なお、Proxyタイプのファイアウォールの場合は、

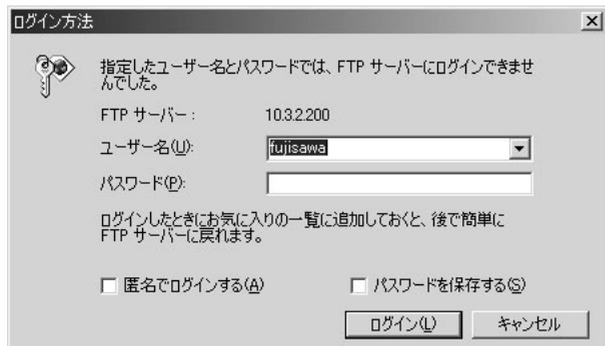
`ftp://ユーザー名%40ftp.fujisawa.gr.jp@proxyホスト名`

というように、最初の「@」(アスキーコードで0x40)を「%40」に置き換えることにより、アクセスができるはずだ。

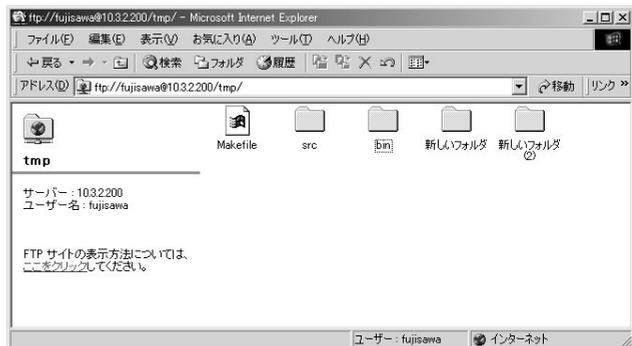
最近では、プロバイダに加入すると自動的にFTPできるWebスペースが数Mバイト付いてくるし、無料のWebサービスも多いため、FTPできる場所は無数にある。

機密の問題に関して、以前この連載(2000年2月号)で作成したWindows用GnuPGなどで、暗号化してからアップロードすればまったく問題はない。何よりも、日ごろからトラブルを多数抱えるサーバ管理者にとって、自分でFTPサーバを構築する必要がなく、クラックの心配やディスククラッシュに脅えなくてすむというのは実に魅力的である。

Sambaには上記のような問題があるし、設定や運用が難しい面もある。そのままファイルを作成できたり開いたりできるのは便利だが、必要のないゴミファイルも増えやすい。FTPサーバを使うのが得策であるケースはかなり多いはずだ。



画面7 IEのパスワード入力画面



画面8 FTPサイトのディレクトリを開いた画面

(ないと困る)ではあるが、サーバの目的によっては機密上の問題が発生する。

セキュリティを重視するなら、生パスワードの保存をやめるか、GnuPGなどで生パスワードを暗号化して保存するようにプログラムを変更するとよいだろう。

ちなみにGNUの暗号化ツールであるGnuPGでは、

```
# gpg --key-gen
```

として、管理者ユーザーの公開鍵を作成し、inpというファイル名に生パスワードを入れ、

```
# gpg --batch -o - -a -r 管理者ユーザー -e inp >out
```

とすると、outというファイルに暗号化することができる(詳しくは、Linux magazine No.2のGPGの記事を参照。<http://linux.ascii24.com/linux/allascii/>からPDFファイルをダウンロード可能)。

秘密鍵をサーバとは別のコンピュータに保存しておけば、セキュリティ面では万全になるものの、管理上は不便になる。安全性と利便性は相反するので、使用する環境に応じて適当な方法を使い分けるとよいだろう。

ユーザーの一覧表示

登録されているユーザーの一覧を得るためには、

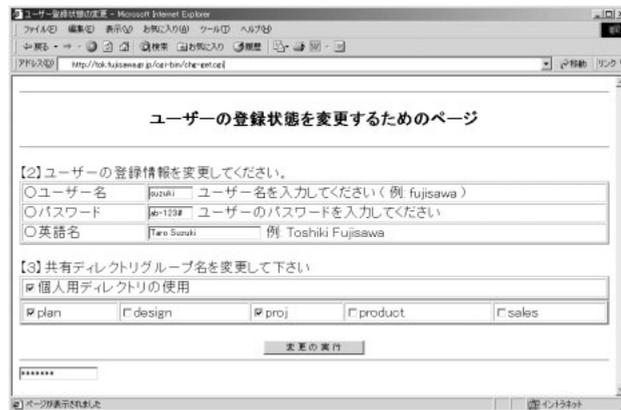
```
# sedusr listup
```

を実行する。このコマンドの出力は、

```
fujisawa,,ERR_NORAW,Toshiki Fujisawa,,
suzuki,,ab-123#,Taro Suzuki,MYDIR,plan/design
tanaka,,K23Ab-4,Ichiro Tanaka,MYDIR,plan/sales
aoki,,16#sWak,Toshio Aoki,,plan/design/sales
ito,,qugA9@,Jiro Ito,,plan/design/sales
ueno,,HeqkG9#,Saburo Ueno,UENO,design/sales
ohno,,[#!#33,Shirou Ohno,OHNO,sales
```

というCSV形式である。また、管理ページからユーザーの一覧表示を選択して、管理者用パスワードを入力すると、登録されているユーザーの一覧が表示される(画面9)。

なお、この中で表示される個人フォルダ名は、



画面9 ユーザーの一覧表示ページ

リスト1 パスワードチェック部分

```
rpw = get_rpw(uname ); /* 生パスワードを取得 */
if( rpw == NULL ){
    rpw = "ERR_NORAW";
}else{
    if( (md5 = get_master_md5(uname) ) == NULL ){
        rpw = "ERR_NOREG";
    }else{
        if( strcmp(crypt(rpw, md5 ), md5 ) == 0 ){
            if( !( *mydir == '\0' && *glist == '\0' ) ){
                if( (md4 = get_smbpw(uname)) == NULL ){
                    rpw = "ERR_NOSMB";
                }else{
                    if( strcmp(md4,calc_smb_md4(rpw))!=0 ){
                        rpw = "ERR_MD4";
                    }
                }
            }
        }else{
            rpw = "ERR_MD5";
        }
    }
}
}
```

- (1) 個人専用ディレクトリはなし
- (2) MYDIR という名称のディレクトリ
- (3) ログインユーザー名と同じ文字列のディレクトリ

の3種類がある。グループ用ディレクトリは、Windowsでマウントされた、N:ドライブ直下に現れるが、このディレクトリと同じ階層に個人のディレクトリを作成されると見苦しいため、N:ドライブ直下のディレクトリ階層には書き込みができない設定になっている。

そのため、個人専用のファイルは、上記の(2)または(3)ディレクトリの中に作成するようにする。この個人専用のディレクトリは他人からは見えないので、自分だけのファイルを置くことが可能であるが、グループでの共有ファイルだけ見ればよいというユーザーのために、個人専用ディレクトリがないような設定もできる。

なお、この画面9の一覧には、fujisawaのようにこのシステムの管理下でないユーザーも表示される。このようなユーザーの場合、生パスワードが存在しないので、パスワードの欄はERR_NORAWという文字列が表示される。今回のシステムでは、メール用パスワードとSamba用パスワードを同一であることにしているが、この一覧表示では、両方のパスワードが生パスワードと等しいかのチェックをしている(リスト1)。

もし、メールパスワードが生パスワードと一致しない場合はERR_MD5が表示され、Samba用パスワードが生パスワードと一致しない場合にはERR_MD4が表示される。

Sambaで使用するMD4パスワードを計算する適当なライブラリを見つけることができなかったため、Sambaのソースコードから該当部分を抜き出し、Sambaとは独立して使用できるようにcalc_smb_md4という関数を作成してある。

なお、WindowsからSambaパスワードを変更されると、このシステムで管理している生パスワードとの整合がとれ

なくなる。このため、完璧を期すのであれば、Sambaを改造してWindowsからのパスワード変更をできないようにし、パスワードの変更はWebツールのみから設定できるようにするとよいだろう。

ユーザーの削除

ユーザーを削除するためには、

```
# sedusr del 'ログイン名,ユーザーの英語名'
```

を実行する。ここでは誤って消さないように、ユーザーの英語名を指定するようにしてある。また、Webからは画面10のようにして、削除できるようになっている。

なお、この削除を行うと、たとえば、

```
mkdir -p /smb/dust/2001-0805-08:38.55-suzuki
```

が実行され、/smb/dustに削除した時間と削除するユーザー名のディレクトリが作成される。そして、

```
/smb/home/suzuki
/smb/mail/suzuki
/smb/logon/suzuki.bat
/var/mail/suzuki
```

といったファイルが、このディレクトリに移動するようになっている。つまり、ユーザーを消しても、そのユーザーのデータはあとから戻せるようにしてあるのだ。ディスク容量が少なくなってきたら、/smb/dustの中を定期的に掃除するとよいだろう。

おわりに

5カ月に渡りサーバ管理に必要なプログラムの作成方法と、運用方法について解説した。ひと口にサーバといっても、必要とされる機能はさまざまであり、使用する場所によって運用方法は千差万別である。また、今回例題として作成したsedusr コマンドは、あくまでサンプルであり、そのまま使用することはできない。このプログラムを参考にして、使いやすいプログラムを作成していただければと思う。



画面 10 ユーザーの削除のページ

ステップアップC言語

メモリアクセスとvolatile宣言

サーバを運用していると、さまざまなトラブルが発生する。特に、管理しているマシンの台数が多い場合は、ハードウェア障害も頻繁に発生する。ハードウェア障害で最初に思い付くのはハードディスクのクラッシュであるが、メモリ関係のトラブルも無視できないくらい多い。

ある日の早朝、ファイルサーバが見えなくなったとたたき起こされた。調べてみると、DNSを提供するnamedプログラムがsignal 11、つまりSIG SEGVで落ちていた。このSEGV (SEGmentation Violation) は、アクセスできないメモリにアクセスを試みたときに発生する割り込みである。

このシグナルは、たとえば、

```
int main(void)
{
    char *p;

    p = (char *)0x12345678;
    *p = 0;
    return 0;
}
```

というように、プログラムにバグがあってデタラメなアドレスにアクセスしたときに発生する。

しかしながら、namedのようなよく使われているプログラムが、signal 11で落ちることはまずあり得ない。signal 11が発生するのは、メモリ不良であることが多いのである。とりあえずリブートしてようすを見たのだが、念のためサーバを置き換える準備をしておいた。案の定、夕方にはnamedだけでなく、sendmailなどがsignal 11で落ち始めたので、即座にサーバごと交換した。このようなメモリ不良の場合、メモリ単体の不良だけでなく、マザーボードや電源電圧の低下などの問題もあるため、マシンごと交換するのが正解である。

最近のSDRAMはクロック周波数が高いため、グラウンドへの配線が数センチ遠かつ

たりすると誤動作する。バスが32ビットもあると、データが書き変わるときに流れる電流もかなりのもので、データバスが0x00000000から0xFFFFFFFFに変わるときの突入電流にグラウンドが耐えられないこともある、したがって、SDRAMのチップ構造を考慮に入れたRAMチェックプログラムを書かないと、メモリ不良を検出することはできない。このような不良を検出するためには、

```
#define SDRAM_TOP \
    ((unsigned long *)0x40000000)
#define SDRAM_END \
    ((unsigned long *)0x41000000)

volatile unsigned long *p;

for(p = SDRAM_TOP; p<SDRAM_END; p++){
    *p = 0xffffffff;
    if( *p != 0xffffffff ){
        error();
    }
    *p = 0x00000000;
    if( *p != 0x00000000 ){
        error();
    }
}
```

というプログラムを書くことになる。上記の最初にメモリに書き込んでいる0xffffffffを0xf0f0f0f0のように、0から1に変化するビットを減らすと流れる電流が少ないのでエラーにならないが、32ビット全部が変化するとビット化けが起きるのである。

さて、上記のプログラムではvolatileというキーワードを使用している。volatileは「揮発性の、一時的な、はかない」という意味を持つが、この修飾子を付けることによりコンパイラの最適化を抑制することができる。もし、上記のプログラムをvolatile修飾なしで、

```
unsigned long *p;
```

として宣言し、最適化オプション「-O」を付けてgccでコンパイルすると、

```
.L7:
    movl $0,(%eax)
    addl $4,%eax
    cmpl $1090519039,%eax
    jbe .L7
```

というアセンブラ出力が得られる。つまり、最初にメモリに書いた0xffffffffは、意味がないので最適化のため取り去られ、あとに書く0x00000000の書き込みだけが実行される。またif文は必ず成立するので、意味のないerror()関数の呼び出しもなくなってしまふ。一方、

```
volatile unsigned long *p;
```

のようにvolatile修飾子を付けた場合は、

```
.L7:
    movl $-1,(%ebx)
    movl (%ebx),%eax
    cmpl $-1,%eax
    je .L8
    call error

.L8:
    movl $0,(%ebx)
    movl (%ebx),%eax
    testl %eax,%eax
    je .L6
    call error

.L6:
    addl $4,%ebx
    cmpl $1090519039,%ebx
    jbe .L7
```

というようにコンパイラの最適化を抑制し、望みどおりの結果が得られる。

Perlスクリプト入門

すぐにできる実践Perl

もともと汎用的な言語として普及したPerlですが、今一番よく使われているのがCGIでしょう。Webブラウザを入出力端末として使えるので、プログラミングの実習環境としても便利です。なにより、気のきいたWebサイトには不可欠です。

第8回 実践CGI

文: おもてじゅんいち / かざぐるま
Text: Junich Omote/Kazaguruma

ふだんにげなく見ているWebですが、その仕組みを考えてみましょう。まず、ブラウザはWebサーバに対して、URLで指定した文書ファイルなどを送るよう要求します。次に、Webサーバは指定されたファイルをブラウザに返します。そして、ブラウザは受け取ったファイルを画面に表示しているのです。このとき送られるページは一般的に、HTMLというフォーマットで作られたファイルです。HTMLファイルはあらかじめWebサーバ側に用意されています。

この仕組みの場合、常に変わらない固定された内容のページなどはよいのですが、ブラウザ側で入力されたフォームのデータによってページの内容を臨機応変に変化させたり、サーバ内にあるデータベースのデータをユーザーの要求に応じて表示したりすることができません。

CGIとは

CGI (Common Gateway Interface) は、Webサーバ内で動く小さなプログラムのことで、HTMLファイルと同じようにブラウザからURLで指定されることによって動きます。HTMLファイルと違ってプログラムですから、データを受け取って何らかの処理を行い、その場合に応じた動的なページを送り返すことができます。

このとき、送り返すデータをHTMLの形にしておけば、受け取ったブラウザにしてみれば単なるHTMLデータを

受け取ったときと同じように表示すればよいということになります。

CGIは、サーバ上で動くプログラムならどんな言語で作られたものでもよいのですが、最近はPerlで作られることが多いようです。また逆に、Perlが使われる目的も最近はいよいよCGIのためのようです。

リスト1は一番簡単なCGIスクリプトです。print文だけしか使っていません。CGIの目的はHTMLデータをブラウザに返すことですが、CGI側からふつうのprint文で文字列を出力してやれば、あとはWebサーバがHTMLデータとしてブラウザに返してくれます。ですから、CGIで

リスト1 一番簡単なCGIの例

```
#!/usr/bin/perl

print "Content-type:text/html\n\n";
print "<html>\n<head></head>\n<body>\n";

printf "これはPerl CGIです。<br>\n";

print "</body>\n</html>\n";
```

リスト2 リスト1が出力したHTMLソース

```
<html>
<head></head>
<body>
これはPerl CGIです。<br>
</body>
</html>
```

はブラウザとのやりとりをほとんど意識せずに、返したいデータをprint文で出力するだけですみます。

リスト1の出力結果、すなわちブラウザに送られたデータはリスト2のようになります。見くらべてみれば、print文で出力されたものがブラウザに届いているのがわかるでしょう。ただ、ここでひとつ決まっていることがあります。それはデータの先頭に、「Content-type:text/html」というヘッダ文と、それに続けて改行を2回送らなければならないことです。ブラウザにはHTMLページのほかにGIFやJPEGイメージ、そのほかいろいろな形式のデータが送られてきます。このため、ブラウザがそれを判別するために、データの先頭にMIMEタイプと呼ばれるデータの形式を表すヘッダを付けなければならないのです。このCGIで送りたいのはHTMLデータですから、MIMEタイプ「text/html」ですよというヘッダを送ります。

リスト1の最初のprint文が、MIMEヘッダの出力になります。リスト2を見てもらえばわかるとは思いますが、このMIMEヘッダ部分はブラウザが解釈するためのものなので、表示されるHTMLデータには現れません。2つめのprint文以降の出力結果が、そのままリスト2のHTMLソースになっています。

CGIスクリプトファイルも、HTMLファイルと同じようにWebサーバのデータディレクトリ内に置いておきます。ブラウザからCGIを起動するときにはURLを指定すればよいのですが、サーバ側で少し設定が必要になります。

WebサーバはApacheを使うものとしませんが、Apacheの設定ファイル/etc/httpd/conf/httpd.confの中の各ディレクトリ記述で、CGIを動かしたいディレクトリに、

```
Options ExecCGI
```

という記述がなければなりません。Apacheではインストール後、/var/wwwディレクトリ下にhtmlとcgi-binというディレクトリが作られていますが、デフォルトではcgi-binディレクトリでCGIが動き、htmlディレクトリではCGIが動かない設定になっています。オープンなWebサイトではセキュリティの関係上、どこでもCGIが動いてしまうのはよくないということでこのような設定になっているのですが、今は練習ということでhtmlディレクトリでもCGIが動くように設定してみましょう。

それでは、httpd.confの中の、

```
<Directory /var/www/html>
```

から始まる記述を見てください。デフォルトでは、

```
Options Indexes Includes FollowSymLinks
```

という記述があると思います。そこでこの記述を、

```
Options Indexes Includes FollowSymLinkss ExecCGI
```

にします。次に、

```
#AddHandler cgi-script .cgi
```

という行のコメント(#)を外し、Webサーバを再起動します。これで、/var/www/htmlディレクトリでもCGIを動かすことができるようになりました。もし、/var/www/htmlディレクトリ全体でCGIを動かせるようにするのはなく、特定のディレクトリのみで動かせるようにするのなら、httpd.conf内には、

```
AllowOverride All
```

と書き、CGIを動かしたいディレクトリに.htaccessというファイルを作り、その中に、

```
Options ExecCGI
```

と記述しておく方法もあります。詳しくはApacheのマニュアルや解説本を参照してください。

さて、これでCGIを動かす環境ができましたので、リスト1のスクリプトをtest.cgiという名前で/var/www/htmlに置いて、ブラウザから、

```
http://<ホスト名>/test.cgi
```

とアクセスしてみましょう。しかし、このままでは「Forbidden Error」が出てしまいます。CGIはプログラムとして起動しなければならないのですが、test.cgiを作っただけでは、実行許可のパーミッションがありません。そこで、

```
$ chmod 755 test.cgi
```

として、test.cgiに誰でも実行できるパーミッションを与

えてください。これでブラウザから起動できます。

ブラウザにたった1行、

これはPerl CGIです。

と表示されれば成功です。ブラウザのソースを見てみれば、リスト2のようにになっているはずです。

時刻で変わるページを作る

リスト1のように、単に決まったHTMLを出力するだけならもちろんCGIを使う必要はありません。場合や要求

リスト3 時刻で変化するCGI

```
#!/usr/bin/perl

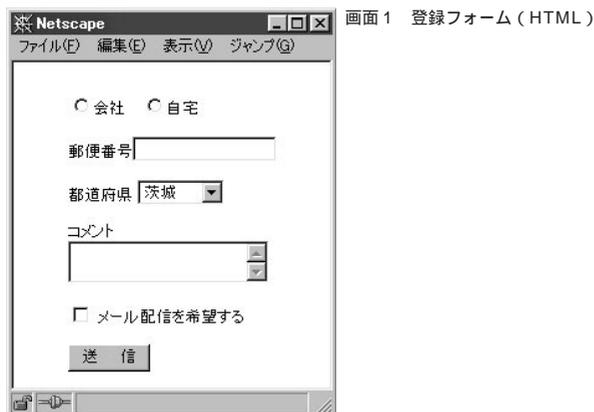
#HTMLヘッダ
print "Content-type:text/html\n\n";
print "<html><head></head><body>\n";

#現在時刻を取得
($sec,$min,$hour) = localtime();

#時刻の表示
printf "%02d:%02d:%02d<br>",$hour,$min,$sec;

#時刻にあわせたあいさつ
if ($hour < 12) {
    print "おはようございます。<br>";
}
elsif ($hour < 18) {
    print "こんにちは。<br>";
}
else {
    print "こんばんは。<br>";
}

#HTMLフッタ
print "</body></html>\n";
```



画面1 登録フォーム (HTML)

に応じて変化するページをブラウザに表示させるのがCGIです。まずは、アクセスした時刻によって表示メッセージを変化させるCGIを作ってみます。

時刻を取得するにはPerlの組み込み関数localtime()を使います。localtime()は、

```
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday)
= localtime($time);
```

といった使い方をするもので、引数の\$timeを省略すれば、現在の時間を返します。戻り値はそれぞれ表のようになります。

localtime()の戻り値はリストですからすべての変数を書く必要はなく、時刻だけなら、

```
($sec,$min,$hour) = localtime();
```

と書くこともできます。今回のCGIでは時刻だけ必要なので、このようにして使います。

リスト3が時刻に合わせてあいさつをするCGIです。現在時刻が午前0～12時なら「おはようございます」、12～18時なら「こんにちは」、それ以降は「こんばんは」と表示します。localtime()関数以外は今までに学んだことばかりですから、スクリプトをじっくり理解して、動かしてみ

戻り値	内容
\$sec	秒 (0 ~ 59)
\$min	分 (0 ~ 59)
\$hour	時 (0 ~ 23)
\$mday	日 (1 ~ 31)
\$mon	月 (0 ~ 11) 実際より1小さい値
\$year	年 (西暦マイナス1900)
\$wday	曜日 (0 ~ 6)
\$yday	元日からの日数

表 localtime()関数の戻り値



画面2 登録確認画面

リスト4 登録フォーム(画面1)のHTMLソース

```

<html>
<head></head>
<body>
<form action="entry.cgi">
<input type="radio" name="addr" value="com">会社
<input type="radio" name="addr" value="home">自宅
<br><br>
郵便番号<input type="text" name="zipcode">
<br><br>
都道府県
<select name="pref">
  <option value="0">茨城
  <option value="1">群馬
  <option value="2">山梨
  <option value="3">埼玉
  <option value="4">東京
  <option value="5">千葉
  <option value="6">神奈川
</select>
<br><br>
コメント<br>
<textarea name="comment">
</textarea>
<br><br>
<input type="checkbox" name="mailreq">
メール配信を希望する
<br><br>
<input type="submit" value="送 信">
</form>
</body>
</html>

```

リスト5 確認画面のHTMLソース

```

<html>
<head></head>
<body>
<center>

登録情報<br><br>
<table border=1>
<tr><td>会社 / 自宅</td><td>会社</td></tr>
<tr><td>郵便番号</td><td>151-8024</td></tr>
<tr><td>都道府県</td><td>東京</td></tr>
<tr><td>コメント</td><td>Linux Business 発売中</td></tr>
<tr><td>メール配信</td><td>希望する</td></tr>
</table>

<form action="register.cgi">
<input type="submit" value="登 録">
</form>
</center>
</body>
</html>

```

ましよう。

フォーム入力処理するCGI

次はWebでもおなじみのフォーム入力です。フォーム自体はHTMLで作れるのですが、入力されたデータを受け付けて処理するにはCGIが必要になります。ここでは画面1のようなフォームで入力されたデータを画面2のように確認画面に表示するCGIを考えてみます。

リスト4は画面1のHTMLソースです。これはCGIではありませんから、HTMLファイルとしてサーバに置きます。すべてHTMLのタグでできているのですが、CGIと連携するために気をつけなければならないことは、<input> や <select> といったフォームの各アイテムに名前を付けておかなければならないことです。それぞれのタグに「name=」という記述があり、各々別の名前が付けられています。

フォームデータを受け取るCGI側では、これらのデータを取り出すために「name=」で指定された名前を使います。CGIに送りたいデータのタグには必ず、しかも重複しないように名前を付けてください。

フォームの場合、CGIはsubmitボタンを押したときに起動するようフォームタグに、

```
<form action = "entry.cgi">
```

というように「action=(CGIファイル名)」と記述しておきます。こうすれば、submitボタンを押したときに、このCGIファイルが起動します。

次にデータを受け取るCGI側を作ります。このCGIはフォームのデータを受け取った後、画面2のような画面をHTMLとして出力することになります。ではまずその画面のHTMLのひな型を作ります(リスト5)。

各欄のデータはとりあえずダミーのものを書いておきます。うまくHTMLができれば、どの部分をCGIによって変化させればよいかを考えます。ここではリスト5の網掛け部分に変化する部分になります。つまり、フォームから受け取った各データをこの部分に出力すればよいことになります。

ライブラリ cgi-lib.pl

リスト6がそのCGIです。フォームから送られてくるデータを受け取るのは、実際には複雑な処理なので、サブル

ーテン化されているものを使います。そのサブルーチンは、作者の Steven E. Brenner 氏によって、cgi-lib.pl というライブラリとして提供されています。入手するには、「cgi-lib」などで検索するか、作者の運営する Web サイト (<http://cgi-lib.berkeley.edu/>) を見てください。cgi-lib.pl が入手できたら、CGI ファイルと同じディレクトリに置きます。

このようなライブラリを使うときは、require を使って、

```
require 'cgi-lib.pl';
```

と書いておきます (リスト6の2行目)。そして、フォーム入力を前処理してくれるサブルーチン ReadParse を呼び出します (3行目)。これでフォームの各アイテムに入力されたデータが簡単に取り出せるようになります。

各アイテムのデータは、

```
$in{'(アイテム名)'};
```

というハッシュ変数に格納されています。ですから、リスト4の各アイテムのデータはそれぞれ、

```
$in{'addr'} $in{'zipcode'} $in{'pref'}
$in{'comment'} $in{'mailreq'};
```

という変数に格納されているのです。

となれば、リスト5のHTMLを元に、これらの変数の値を埋め込んで出力すればよいのですから、このCGI スクリプトはリスト6のようになります。

リスト6では固定したHTML部分をヒアドキュメントを使って出力しています。こうすれば大量のprint文を書かなくて済み、スクリプトも見やすくなります。

ハッシュ変数

ハッシュ変数は、配列のようにリストデータを格納するもので、連想配列とも呼ばれます。配列では各要素を呼び出すときにインデックス番号を使いますが、ハッシュ変数では名前で呼び出せます。たとえば、HTMLの色番号を配列に格納した場合、

```
@clist = ("#000000", "#888888", "#ffffff");
```

となり、各要素は、

リスト6 確認画面 CGI ソース (entry.cgi)

```
#!/usr/bin/perl
require 'cgi-lib.pl';
&ReadParse;

print <<'EOD1';
Content-type:text/html

<html><head></head><body>
<center>登録情報<br><br>
<table border=1>
EOD1

print "<tr><td>会社 / 自宅</td><td>$in{'addr'}</td></tr>\n";
print "<tr><td>郵便番号</td><td>$in{'zipcode'}</td></tr>\n";
print "<tr><td>都道府県</td><td>$in{'pref'}</td></tr>\n";
print "<tr><td>コメント</td><td>$in{'comment'}</td></tr>\n";
print "<tr><td>メール配信</td><td>$in{'mailreq'}</td></tr>\n";

print <<'EOD2';
</table>
<form action="register.cgi">
<input type=submit value="登録">
</form>
</center>
</body>
</html>
EOD2
```

ヒアドキュメント (固定HTML部分)

スクリプトによって変化する部分

ヒアドキュメント (固定HTML部分)

```
$clist[0]
$clist[1]
$clist[2]
```

のように使うことができます。

配列変数は@ ですが、ハッシュ変数名には%が付きます。
ハッシュ変数の初期化は、

とアクセスしなければなりません、ハッシュ変数を使った場合だと、

```
%hlist = {'black'=>'#000000' , 'gray'=>'#888888'
          , 'white'=>'#ffffff'};
```

```
$hlist{'black'}
$hlist{'gray'}
$hlist{'white'}
```

というように、名前(キー名)と値をペアにして指定します。前述したcgi-lib.plのReadParseを使うと、フォームデータがそれぞれ、各アイテム名をキーとするハッシュ変

リスト7 フォームデータを加工して表示する CGI (entry.cgi)

```
#!/usr/bin/perl
require 'cgi-lib.pl';
&ReadParse;

print <<'EOD1';
Content-type:text/html

<html><head></head><body>
<center>登録情報<br><br>
<table border=1>
EOD1

if ($in{'addr'} eq 'com') { $addr = "会社"; }
else { $addr = "自宅"; }

@plist = ('茨城', '群馬', '山梨', '埼玉',
          , '東京', '千葉', '神奈川');

$prno = $in{'pref'};

$comm = $in{'comment'};
$comm =~ s/\n/<br>/g;

print "<tr><td>会社 / 自宅</td><td>$addr</td></tr>\n";
print "<tr><td>郵便番号</td><td>$in{'zipcode'}</td></tr>\n";
print "<tr><td>都道府県</td><td>$plist[$prno]</td></tr>\n";
print "<tr><td>コメント</td><td>$comm</td></tr>\n";

if ($in{'mailreq'}) {
    print "<tr><td>メール配信</td><td>希望する</td></tr>\n";
}
else {
    print "<tr><td>メール配信</td><td>希望しない</td></tr>\n";
}

print <<'EOD2';
</table>
<form action="register.cgi">
<input type=submit value="登録">
</form>
</center>
</body>
</html>
EOD2
```

フォームデータaddrが「com」なら\$addrは「会社」に、「home」なら「自宅」にする

県名のリストを配列@plistに用意して、\$prnoにフォームデータの数値を格納すれば、\$plist[\$prno]で県名を取得できる

コメントデータ内の改行を
に置換する

zipcodeはフォームデータのまま

フォームデータmailreqがチェックされていたら(真なら)「希望する」を表示、チェックされていないば(偽なら)「希望しない」を表示

数に格納されてくるというわけです。

データの加工

リスト6を実行した結果が画面3です。ちょっとおかしいですね。「会社/自宅」「都道府県」「メール配信」の欄が思ったものとは違う表示になっています。フォームデータをそのまま表示するとこうなったのですが、実際には画面2のような表示になるように加工しなければなりません。

リスト7はその処理を加えたものです。addrはif文を使って、「com」「home」を「会社」「自宅」に変えています。prefは県名を配列に用意して、フォームから送られてくる数値をインデックス番号として使います。commentは特に文字を変換する必要はないのですが、フォームで入力された改行がHTMLではそのまま改行として扱われないために、
タグに置き換えています。mailreqはprint文ごと2種類を用意して、フォームで「メール配信を希望する」がチェックされたときとそうでないときを判断しています。

ここではとりあえず、できるだけいろいろな処理を紹介する意味で使っておりま。実際には一番適していて、できるだけシンプルなスクリプトになるような処理を行えばよいでしょう。

リスト8は、フォームに入力されたデータをファイルに保存するスクリプトです。これをリスト7に付け加えれば

リスト8 登録されたデータをファイルに保存（リスト7の最後に追加）

```
open OUT,">>entry.dat";

print OUT "$addr,$in{'zipcode'},$plist[$pno],$comm,";

if ($in{'mailreq'}){ print OUT "希望する\n"; }
else                { print OUT "希望しない\n"; }

close OUT;
```



画面3 フォームデータをそのまま表示

データを保管することができます。その結果、どのようなファイルが作成されるかは、実際に動かして確認してください。ただし、保存するデータファイルentry.datには666のパーミッションが必要です。

Hit&Blow ゲームを作る

最後にCGIでゲームを作ってみます。Hit&Blowという数あてゲームです。ご存じの方もおられると思いますが、1から9までを使った4桁の数字を入力すると、数字も桁位置も合っていたときにHit、数字は合っているが桁位置が合っていないときにBlowとして、「1 Hit 2 Blow」というような判定を返します。ユーザーはその結果を元にまた4桁の数字を入力して行き、最終的に相手の数字を4桁とも当てるのです。ここではサーバ(CGI)側が問題の数字を考えて、ユーザーの入力に対してHit&Blowの判定を下すようにします(画面4)。

最初に問題の数字がランダムに発生しなければゲームになりませんから、乱数を発生するrand()関数を使います。rand()関数は0から引数までの範囲でランダムな数字(実数)を1つ返します。たとえば、

```
$a = rand(3);
```

とすれば\$aには、「1.6235268」などの数字が返ります。このままでは小数部分があって使いにくいので、普通は、

```
$a = int(rand(3));
```

というint()関数を使って整数に直します。こうすれば、\$aには0、1、2のいずれかがランダムに格納されます。リスト9では、で1から9までの乱数を発生しています。そ



画面4 Hit&Blow ゲーム

れを1桁ずつ変数\$ansに追加して行って、4桁になるまで繰り返すのですが、同じ数字が出てこないように、すでに\$ansに含まれていた場合はもう一度乱数を発生させています。\$ansが4桁になればループを抜けます。

なお、このCGIは自分で自分を呼び出す構造になっています。ゲームは何度も繰り返されるものですし、そのたびに入力された数字の判定という同じ処理を行いますから、1つのCGIを繰り返し起動してやればよいのです。

CGIが出力するHTMLにフォームを含めて、そのformタグの「action=」に自分自身を記述すると、再びそのCGIが呼び出され、同時にフォームデータも渡されます。あとはその繰り返しです(リスト9の)。

最初にこのCGIをブラウザから直接起動したときだけは、入力されたフォームデータがないことを検出して問題の数字を作り、スタート画面を表示します(リスト9の)。

フォームデータを受け取ったとき、HitおよびBlowの判定をするために、4桁の数字を1桁ずつに分解する必要がありますが、ここではパターンマッチを使ってちょっと変わった方法を採用しています。元の文字列は4桁の数字ですから、

```
/\d\d\d\d/
```

にマッチします(¥d は [0-9]と同じ)。1つの¥dに1桁の数字がマッチしていますから、パターンを、

```
/(\d)(\d)(\d)(\d)/
```

にすれば、それぞれの桁を後方参照することができます。つまり、各桁の数字はパターンマッチの結果\$1、\$2、\$3、\$4にうまくあいに格納されることになるのです。その後、

```
@alist = ($1,$2,$3,$4);
```

というように配列に移してやれば、@alistの各要素は、

```
$alist[0] = (1桁目の数字)
$alist[1] = (2桁目の数字)
$alist[2] = (3桁目の数字)
$alist[3] = (4桁目の数字)
```

となります。@nlistも同様です。

あとは、今までに学習した知識で理解できるものばかりです。スクリプトをじっくり理解するのもいいですが、とにかく実際に動かしてから自分で書き換えてみましょう。

リスト9 Hit&Blowゲーム(hitblow.cgi)

```
#!/usr/bin/perl
require 'cgi-lib.pl';
&ReadParse;

print <<'EOD1';
Content-type:text/html

<html><head></head><body><center>
<h2>Hit&Blow</h2>
EOD1

if ($in{'num'}) { #フォームデータがあるか
    # 1桁ずつに分解
    $ans = $in{'ans'};
    $ans =~ /(\d)(\d)(\d)(\d)/;
    @alist = ($1,$2,$3,$4);
    $in{'num'} =~ /(\d)(\d)(\d)(\d)/;
    @nlist = ($1,$2,$3,$4);
    #BLOWを判定
    $b = 0;
    foreach $n (@nlist) {
        if ( $ans =~ /$n/ ) { $b++; }
        # @nlistの各文字が$ansに含まれるかどうか
    }
    #HITを判定
    $h = 0;
    for($i=0; $i<4; $i++) {
        if ($nlist[$i] == $alist[$i]) { $h++; }
    }
    $b -= $h; # BLOWにはHITが重複しているため
    if ( $h == 4 ) {
        print "<b>$h</b> HIT<br>";
        print "Congratulations !!<br>";
        print "<b>$ans</b><br>";
    }
    else {
        print "<b>$h</b> HIT/<b>$b</b> BLOW<br>";
    }
}
else { #フォームデータがないとき
    $ans = "";
    # $ansが4桁になるまで
    while(length($ans) < 4) {
        $r = int(rand(9)) + 1; # 1-9の乱数
        if ( $ans =~ /$r/ ) { redo; }
        # $ansにあればやり直し
        $ans .= $r; # $ansに追加
    }
    print "START!!<br>\n";
}

print <<"EOD2";
<form action='hitblow.cgi' method='post'>
4桁の数字(1-9)<br>
<input type=hidden name='ans' value='$ans'>
<input type=text size=4 name='num'><br><br>
<input type=submit value='送信'><br>
</form></center></body></html>
EOD2
```

[超]入門シェルスクリプト

今回は、bashのバージョン2以降で利用できるシェル変数の配列機能に注目してみよう。配列の要素に値を設定する方法や、設定した内容を取り出す方法などについて説明したのち、配列を利用してカレントディレクトリを対話的に変更するシェル関数を作成する。

第12回 シェル変数の配列を活用する

文：大池浩一
Text: Koichi Oike

bashのベースとなったBourneシェルには、PerlやRubyといったスクリプト言語と違って、複数の要素を数値（インデックス）で制御する「配列」の仕組みが用意されていない。その代わりに、空白文字（スペース・タブ・改行）や「:」（コロン）などの区切り文字を使って、複数の要素を1つのシェル変数にまとめて格納する方法が用いられる。たとえば、環境変数PATHやMANPATHには、「:」（コロン）を区切り文字として複数のディレクトリが格納される。

この方法の欠点は2つある。ひとつは、要素を操作するのが面倒なことだ。先頭や末尾以外の位置への要素の挿入や、既存の要素の削除、内容の変更などを行おうとすると、数行から十数行のスクリプトを書かなくてはならない。もうひとつの欠点は、区切り文字自体を含んだ要素を正しく扱えないことだ。たとえば、「:」を含むディレクトリをPATHに追加することを考えてみよう。

先月号の本連載で紹介した組み込みコマンドevalを利用すれば、配列によく似た仕組み（擬似配列）を実現して、複数のシェル変数に情報を分散して格納できる。ただし、evalを利用したスクリプトはかなり読みにくしいし、要素数などを別のシェル変数で管理する必要がある。

そこで、bashのバージョン2以降には、シェル変数の配列機能が新たに用意された。それぞれの要素は通常のシェル変数と同様に扱えるうえ、複数の要素に同時に値を設定したり、要素数を簡単に参照できるため、多数の情報を効

率よく処理することができる。

bash2ではシェル変数の配列を活用しよう

今回紹介するシェル変数の配列機能は、bashのバージョン2系（最新版は2.04）でのみ利用できる。ところが、Red Hat 6ベースのディストリビューション（Vine Linuxなど）では、標準シェル（/bin/sh、/bin/bash）としてバージョン1系のbashが使われている。こうしたディストリビューションは、バージョン2系のbashを別パッケージで配布しているのだから、これを利用しよう。バージョン1系とは別の名前（/bin/bash2）でインストールされる。

一方、Red Hat 7.1やKondara 2.0などでは、最初からバージョン2系のbashが標準シェルとして採用されているので、新たにインストールする必要はない。これらのディストリビューションでは、/bin/bashと/bin/sh、/bin/bash2はすべて同じ内容だ。以下では、

- ・従来の方法の利点と欠点
- ・配列の要素に値を設定する方法
- ・配列の要素の内容を参照する方法

などについて説明したのち、カレントディレクトリを対話的な操作で変更するシェル関数「icd」を、シェル変数の

配列機能を利用して構築する。

1つのシェル変数に複数の要素を格納する

配列を利用するとどのような点が便利になるのかを明らかにするために、まずは複数の要素を1つのシェル変数にまとめて格納する従来の方法について説明する。

典型的な例として、コマンドの検索パス(ディレクトリ)が設定された環境変数PATHを取り上げよう。PATHには、複数のディレクトリが「:」(コロン)を区切り文字として格納されている。たとえば、

```
$ echo $PATH
/bin:/usr/bin:/usr/X11R6/bin:/usr/local/bin
```

といった具合だ。

シェルのプロンプトに入力された(あるいはシェルスクリプトに記述された)コマンドは、PATHの先頭のディレクトリ(上の例では「/bin」)から順番に検索され、最初に見つかったコマンドが実行される。

PATHの末尾に、新たなディレクトリ(たとえば「/home/daichi/bin」)を追加してみよう。現在のPATHの内容は「\$PATH」で参照できるから、

```
$ PATH=$PATH:/home/daichi/bin
```

とすればいい。「\$PATH」と「/home/daichi/bin」の間には区切り文字の「:」が必要だ。これを忘れると、PATHの末尾が「/usr/local/bin/home/daichi/bin」となって、コマンドの検索に失敗してしまう。

一方、PATHの先頭に追加するなら、ディレクトリ(たとえば「/usr/kde2/bin」)を「\$PATH」の前に置い

リスト1 指定されたディレクトリを環境変数PATHから取り除くシェル関数 delpath

```
1: function delpath
2: {
3:   ifs="$IFS"
4:   IFS=:
5:   path=""
6:   for f in $PATH; do
7:     if [ "$f" != "$1" ]; then
8:       path=$path${path:+}$f
9:     fi
10:  done
11:   IFS="$ifs"
12:   PATH=$path
13: }
```

て、

```
$ PATH=/usr/kde2/bin:$PATH
```

とすればいい。

このように、新たな要素を先頭や末尾に追加するだけなら、従来の方法でも簡単に操作できる。しかし、要素を1つずつ取り出す必要がある場合は、数行から十数行のスク립トを書く必要がある。

たとえば、現在のPATHに格納されたディレクトリを、1行に1つずつ表示するには、以下のようなスクリプトを書かなければならない。

```
#!/bin/sh
IFS=:
for f in $PATH; do
    echo $f
done
```

まず、環境変数IFS(インプット・フィールド・セパレータ)の内容を、初期値の空白文字(スペース・タブ・改行)から「:」に変更する。これで、PATHの内容を展開する際に、「:」を区切り文字としてワード分割が行われるようになる。そこで、for制御構造を使ってPATHに含まれるディレクトリを1つずつループ変数fに取り出し、echoで出力すればいい。

スクリプトを少し変更して、PATHに含まれるディレクトリの数を数えることもできる。

```
#!/bin/sh
IFS=:
count=0
for f in $PATH; do
    count=`expr $count + 1`
done
echo $count
```

ディレクトリの数を格納するシェル変数countを0に初期化しておき、for制御構造でPATH内のディレクトリを1つ取り出すたびに、countの値を1つ増やす。最後にechoでcountの値を出力すればいい。

PATHの内容を変化させる操作、たとえばディレクトリの挿入や削除などを行う場合は、シェル関数を利用する必

要がある。通常、シェルスクリプトはサブシェルで実行されるので、PATHの内容を変更しても元のシェルのPATHには影響しないのだ。なお、シェル関数の詳細については、本連載の第9回（2001年7月号）を参照されたい。

例として、関数実行時のコマンドラインで指定したディレクトリをPATHから削除するシェル関数「delpath」を作成した（リスト1）。delpathを定義後、たとえば、

```
$ delpath /usr/local/bin
```

とすると、現在のPATHから「/usr/local/bin」を取り除いたものが新たなPATHとなる。

内容を説明しよう。まず、PATHに含まれるディレクトリをワード分割するため、環境変数IFSに「:」を格納する必要がある（4行目）。ただし、関数終了後もIFSが「:」のままでは、ほかのコマンドやスクリプトの動作に悪影響を及ぼす可能性がある。そこで、あらかじめIFSの内容をシェル変数ifsに退避しておき（3行目）for制御構造を抜けた後で元の内容に戻す（11行目）。なお、変数名の大文字/小文字は区別されるので、IFSとifsはまったく別の変数だ。

続いて、for制御構造を利用して、PATHに含まれるディレクトリを1つずつループ変数fに格納し（6行目）、その内容を関数実行時の最初のコマンドライン引数（\$1で参照）と比較する（7行目）。両者が等しくないなら、あらかじめ空文字列に初期化しておいたpath（PATHとは別の変数）の末尾に「:」とループ変数fの内容を追加する（8行目）。両者が等しい場合はpathは変更されないの、最終的にpathは指定したディレクトリだけが削除された状態になる。あとは、PATHにpathの内容を設定すればいい（12行目）。

注意してほしいのは、8行目で区切り文字の「:」をそのまま書く代わりに、「\${path:+:}」としている点だ。「:」をそのまま書くと、最初のディレクトリの前にも「:」が付いてしまう。そこで、変数の内容が空でない場合に指定した文字列で置換される「\${変数名+文字列}」という表現を利用する。つまり、すでにpathにディレクトリが設定されている場合にだけ「:」が付くわけだ。

さらに、ディレクトリをPATHの途中に挿入するシェル関数「inspath」も作成してみた（リスト2）。

```
$ inspath /usr/local/bin 2
```

とすると、PATHの先頭のディレクトリと2番目のディレクトリ間に「/usr/local/bin」が挿入される。

delpathと異なる部分だけ説明しよう。まず、ディレクトリの数を保持するシェル変数countを用意して、0に初期化しておく（3行目）。その後、PATHからディレクトリを取り出すたびに1つずつ増やし（8行目）countの値が関数実行時の2番目のコマンドライン引数（\$2で参照）と等しいなら、現在のpathの末尾に「:」と最初のコマンドライン引数（\$1で参照）を追加する（9～11行目）。それとは別に、現在のpathの末尾に「:」とループ変数fの内容を追加する（12行目）。これで、PATHに設定されたディレクトリ間に、指定したディレクトリが挿入されるわけだ。

この節での要点をまとめると、

- 空白文字や「:」などの区切り文字により、複数の要素が1つのシェル変数にまとめて格納される。
- 先頭や末尾への追加は簡単に行える。
- シェル変数の内容を操作するには、数行から十数行のスクリプトやシェル関数を書く必要がある。

ということになる。

シェル変数の配列に値を格納する

「複数の要素を1つのシェル変数にまとめて格納する」という従来の方法では、要素を挿入、削除、内容変更する際に、いったんそれぞれの要素を取り出し、なんらかの操作を行ったあとで、再度1つにまとめるといった処理が必要になる。また、要素を区切るために使われる区切り文字が

リスト2 指定されたディレクトリを環境変数PATHの指定位置に挿入するシェル関数inspath

```
1: function inspath
2: {
3:   count=0
4:   ifs="$IFS"
5:   IFS=:
6:   path=""
7:   for f in $PATH; do
8:     count=`expr $count + 1`
9:     if [ $count -eq "$2" ]; then
10:       path=$path${path:+:}$f
11:     fi
12:     path=$path${path:+:}$f
13:   done
14:   IFS="$ifs"
15:   PATH=$path
16: }
```

要素自体に含まれていると、要素を取り出す際に途中で切れてしまうといった問題も起きる。

こうした点を考慮して、バージョン2系のbashでは、シェル変数を配列として利用する機能が新たに用意された。配列として使うシェル変数の名前は、通常のシェル変数と同様に英字と数字（2文字目以降）で構成され、使用前に最大要素数などを宣言する必要はない。

配列の要素は、C言語などと同様の書式、

変数名[インデックス]

で表わせる（インデックスは0以上の整数）。

たとえば、配列用のシェル変数をhogeとすると、配列の要素はhoge[0]、hoge[1]、hoge[2]...と書ける。なお、インデックスの上限は6千億弱なので、日常的な処理で困ることはまずない。

配列の要素に値を設定する方法は2種類用意されている。ひとつは、通常のシェル変数と同様に、

変数名[インデックス]=値

という書式で、要素ごとに値を設定する方法だ。たとえば、

```
$ hoge[0]='Linux magazine'
$ hoge[1]='*** 2001/10 ***'
$ hoge[2]=[超]入門シェルスクリプト
```

とすると、hoge[0]には「Linux magazine」、hoge[1]には「*** 2001/10 ***」、hoge[2]には「[超]入門シェルスクリプト」という文字列がそれぞれ設定される。なお、設定値に空白文字や特殊文字（「*」や「?」など）が含まれる場合は、両端を「'」または「"」で囲む必要がある。

一方、もうひとつの方法では、

変数名=(値 値 値 ...)

という書式で、配列の複数の要素に値を設定できる。カッコ内の最初の値はインデックスが0の要素、次の値はインデックスが1の要素、...と順番に設定される。

たとえば、上の例と同じ内容を設定するなら、

```
$ hoge=('Linux magazine' '*** 2001/10 ***' [超]入門
シェルスクリプト)
```

とすればいい。

ただし、こちらの方法では、それまで配列で使われていた要素の内容がすべて破棄されてしまう。このため、初めてその配列を利用する場合や、あるいは以前の内容を破棄して利用したい場合に限って使うべきだ。

コマンドの出力を配列に取り込みたいときには、カッコ内でコマンド置換を使えばいい。たとえば、

```
$ dirs=(IFS=: ; echo $PATH)
```

とすると、「`」で囲まれた部分がサブシェルで実行され、実行結果が配列dirsに格納される。具体的には、PATHに含まれるディレクトリが「:」を区切り文字としてワード分割されてから出力されるので、dirs[0]、dirs[1]、dirs[2]...に順番に格納されるわけだ。

通常はインデックスが0の要素から順番に利用するが、特殊な使い方として不連続なインデックスの要素だけを利用することもできる。このような場合、複数の要素にまとめて値を設定するには、

```
$ cpu=([8086]=Intel [6502]=MOS [6809]=Motorola)
```

のように、「[インデックス]=値」という書式で、値を格納するインデックスを指定する必要がある。

この節での要点をまとめると、

- bashのバージョン2以降では、シェル変数を配列として利用できる。使用前に宣言は必要ない。
- 配列の要素は「変数名[インデックス]」で表わす。
- 各要素に個別に値を設定だけでなく、複数の要素にまとめて値を設定することもできる。
- 要素のインデックスは不連続でもいい。

ということになる。

シェル変数の配列の要素を参照する

シェル変数の内容を参照するには、変数名の前に「\$」を付けて「\$変数名」と書く（簡易形）。また、変数名を中カッコで囲って「\${変数名}」と書いてもいい（標準形）。たとえば、「\$foo」や「\${foo}」といった具合だ。

一方、シェル変数の配列の要素を参照する際には、標準形しか使えないので注意しよう。「\${変数名[インデックス]}」のように、インデックス部分も含めて中カッコで囲まむ必


```
$ (IFS=, ; echo "${hoge[*]}")
```

Linux magazine,*** 2001/10 ***,[超]入門シェルスクリプト

のように、全要素が「,」で区切られるようになる。なお、IFSの変更が以後のコマンドの動作に影響を及ぼさないよう、カッコで囲んでサブシェルで実行している。

この節での要点をまとめると、

- ・配列の要素は「`${変数名[インデックス]}`」で参照できる（中カッコは必須）。
- ・インデックスに「@」を使って「`${変数名[@]}`」とすると、配列のすべての要素の並びをそれぞれ独立したワードとして参照できる（区切り文字はスペース）。
- ・配列の要素数は「`${#変数名[@]}`」で参照できる。
- ・インデックスに「*」を使って「`${変数名[*]}`」とすると、配列のすべての要素の並びを1つのワードとして参照できる（IFSで区切り文字を変更可能）。

ということになる。

シェル変数の配列を使ったスクリプト

ひと通り説明が済んだところで、配列を使った簡単なスクリプトをお見せしよう。先月号の本連載では、標準入力の内容を最後の行から逆に表示するスクリプト「mytac」を作成した（リスト3）。mytacは、evalによる擬似配列を駆使して作られているので、今回は本当の配列を使った「mytac2」を作成してみた（リスト4）。

この両者を比較してみよう。どちらも、

(1)前準備を行う

(2)標準入力の内容を（擬似）配列に確保

(3)（擬似）配列の内容を逆順に出力

リスト3 evalによる擬似配列を利用して、標準入力を逆順に出力するスクリプトmytac

```
1: #!/bin/sh
2: idx=0
3: IFS=''
4: while eval read lines_$idx; do
5:   idx=`expr $idx + 1`
6: done
7: while [ $idx -gt 0 ]; do
8:   idx=`expr $idx - 1`
9:   eval echo \${lines_$idx}
10: done
```

という構成で作られている。

(1)前準備を行う

mytacでは、擬似配列のインデックスを制御するシェル変数idxを用意して0に初期化する（2行目）。また、行頭の空白文字が出力時に削除されるのを防ぐため、環境変数IFSに空文字列を設定する（3行目）。

一方、配列を使うmytac2では、インデックスは自動的に制御されるので別のシェル変数を用意する必要はない。2～3行目では、環境変数IFSに改行のみを設定している。その理由はすぐあとで説明しよう。

(2)標準入力の内容を（擬似）配列に確保

mytacでは、while制御構造による繰り返し処理で標準入力の内容を取り込む（4～6行目）。readで標準入力を1行ずつ擬似配列の要素に格納し、インデックスを制御するidxの値を1ずつ増やす。evalを利用した擬似配列についての詳細は先月号の本連載を参照されたい。

これに対し、mytac2ではたった1行、

```
lines=(`cat`)
```

だけで同じ処理が済んでしまう（4行目）。配列の強力がよくわかる場面だ。

まず、catがサブシェルで実行され、標準入力の内容がそのまま標準出力に出力される。コマンド置換により、その結果はカッコ内に展開され、配列の要素に順番に格納される。前準備で環境変数IFSを改行のみに変更しておいたので、展開時に1行ずつワード分割される。つまり、lines[0]には1行目、lines[1]には2行目、...といった具合に配列linesの要素に格納されるわけだ。

リスト4 配列を利用して、標準入力を逆順に出力するスクリプトmytac2

```
1: #!/bin/bash2
2: IFS='
3: '
4: lines=(`cat`)
5: idx=${#lines[@]}
6: while [ $idx -gt 0 ]; do
7:   idx=$((idx-1))
8:   echo "${lines[idx]}"
9: done
```

(3) (擬似) 配列の内容を逆順に出力

mytacでは、while制御構造による繰り返し処理で、インデックスを制御するidxの値を1ずつ減らしながら、擬似配列に格納された内容を出力する(7~10行目)。擬似配列の要素の内容を出力する9行目は、evalの使い方を知っていないと理解しづらいかもしれない。

mytac2では、まずシェル変数idxに配列linesの要素数(「\${#lines[@]}」で参照)を格納する(5行目)。その後の処理はmytacとほとんど同じだ。ただし、idxの値を1ずつ減らす処理では、bashのみで利用可能な\${(...)}構文を利用している(7行目)。また、配列の要素の内容を出力する8行目は、mytacよりかなり直感的な記述だ。

今月のスクリプト

後半は、与えられたテーマを実現するスクリプトの作成手順を説明する。今月は、bashのバージョン2以降で利用できるシェル変数の配列を使って、

・カレントディレクトリを対話的な操作で変更するシェル関数icd

を作成する。カレントディレクトリのすべてのサブディレクトリや「..」(親ディレクトリ)を配列の要素に格納し、select制御構造を使って対話的な選択処理を行う。

findを使ってサブディレクトリの一覧を取得

カレントディレクトリに含まれるサブディレクトリの一覧は、UNIX系OSの基本コマンドのひとつfindを使って取得する。findでは、検索開始ディレクトリ、検索条件(判別式)、検索ファイルの処理(アクション)を「find <ディレクトリ> <判別式> <アクション>」という書式で指定する。

今回の場合、ディレクトリには「.(カレントディレクトリ)、判別式は「-type d」(ディレクトリのみマッチ)、アクションは「-print」(ファイル名を表示)を指定すればいい。サブディレクトリ以下のファイルを検索しないオプション「-maxdepth 1」も必要だ。たとえば、

```
$ find . -maxdepth 1 -type d -print
```

とすると、カレントディレクトリに含まれるすべてのディレクトリが、1行に1つつ出力される。カレントディレ

クトリと「-print」の指定は省略できるので、

```
$ find -maxdepth 1 -type d
```

としても結果は同じだ。

findの出力のうち、「.(カレントディレクトリ)以外のディレクトリは先頭に「./」が付いている。これを取り除くには、sedを「|」(パイプ)で接続して、

```
$ find -maxdepth 1 -type d | sed 's!^\./!!'
```

とすればいい。

sedの引数に指定した「s!^\./!!」は、入力行の先頭が「./」ならばその部分を削除せよ、という意味のスクリプトだ。これで、「./src」や「./GNUstep」といったfindの出力が、「src」や「GNUstep」に加工される。

シェル関数icdのプロトタイプを作成する

基本的な機能を実装したicdのプロトタイプを作成してみた(リスト5)。3行目では、上記のコマンドラインで得られた複数のディレクトリを、コマンド置換によって配列dirsの要素にまとめて格納している。

このとき、dirs[0]には「.」が格納されるが、カレントディレクトリの変更先としては意味がない。そこで、4行目で「..」(親ディレクトリ)に変更する。

あとは、5行目のselect制御構造に、配列に格納したすべてのディレクトリを「\${dirs[@]}」として与えると、それらのディレクトリが選択肢として番号付きで一覧表示され、ユーザーに番号の入力を求めてくる。

入力した番号が選択肢の範囲内なら、シェル変数ansに対応するディレクトリが格納される。すると、6行目のif制御構造の条件判断が成立し、cdによるカレントディレク

リスト5 シェル関数icd(プロトタイプ)

```
1: function icd
2: {
3:   dirs=(`find -maxdepth 1 -type d | sed
   's!^\./!!')
4:   dirs[0]='..'
5:   select ans in "${dirs[@]}"; do
6:     if [ -n "$ans" ]; then
7:       cd "$ans"
8:       break
9:     fi
10:  done
11: }
```

トリの変更と、breakによるselectからの脱出が行われたあと、シェル関数を終了する。

リスト5のシェル関数icdを適当なファイル(たとえばicdfunc)に記述し、

```
$ . icdfunc
```

とすると、現在のシェルにicdが定義される。

```
$ icd
```

として実行すると、

```
1) ..
2) GNUstep
3) src
#?
```

のように、「..」とサブディレクトリの一覧が番号付きで表示される。あとは、変更したいディレクトリに対応する番号を入力すればいい(中断はCtrl-Cキー)。

シェル関数icdを完成させる

プロトタイプicdにはいくつか問題がある。まず、シェル関数内で使用されるシェル変数と同名の変数がすでに使われていると、関数の実行によりその内容が失われてしまう。また、select制御構造が入力を求める際のプロンプトも、初期値のままの「#?」より、もっとわかりやすい文字列に変更したほうがいい。機能的な面では、カレントディレクトリ変更後に終了しないで、再度変更できると便利だろう。こうした点を改良して、シェル関数icdを完成させた(リスト6)。内容を見ていこう。

まず、dirsやansなどのシェル変数を、シェル関数内でのみ有効なローカル変数として宣言する(3行目)。これで、同名のシェル変数がすでに使われていたとしても、その内容にはいっさい影響を与えない。

また、select制御構造のプロンプトは環境変数PS3で変更できる。そこで、現在の内容を別のシェル変数ps3に退避しておき(4行目)「select number(0:exit)?」という文字列に変更する(5行目)。また、関数を終了する前に元の内容に戻しておくことを忘れずに(16行目)。

続いては、カレントディレクトリの変更を繰り返し行う処理だ。残念ながら、select制御構造による繰り返しの中

で配列dirsの内容を変更しても、選択肢の一覧は変化しない。select実行時に指定された選択肢の内容は、独自に管理されているからだ。

そこで、selectの外側にwhile制御構造を追加して、ディレクトリを配列に設定する処理と、選択肢の中からディレクトリを選択する処理を繰り返すようにした。これらの処理は、選択したディレクトリを格納するansの内容が空でない間繰り返される。なお、最初はansが空文字列なので、ダミーの文字列を設定しておく(6行目)。

whileにより繰り返される処理を見ていこう。ディレクトリを配列に設定する部分(8~9行目)はプロトタイプと同じだ。また、連続して変更できることを考慮して、その時点でのカレントディレクトリを表示する(10行目)。

select制御構造による選択処理は、プロトタイプとはかなり異なる。if制御構造を排除して、変数の内容が空の場合にだけ指定した文字列で置換される「\${変数名:文字列}」という表現を利用して、cdの実行時に場合分けを行っているからだ(12行目)。

入力された番号が選択肢の範囲内なら、対応するディレクトリがansに格納され、cdによりカレントディレクトリが変更される。一方、番号が範囲外(「0」など)の場合はansが空文字列になる。すると、「\${ans:-.}」は「.」に置換され、「cd .」が実行されるので、カレントディレクトリは変わらない。

どちらの場合も、13行目のbreakでいったんselectの繰り返し処理から抜け、7行目のwhile制御構造の条件判断に戻る。ansが空でなければ処理が繰り返され、空の場合はシェル関数を終了する。

リスト6 シェル関数icd(完成版)

```
1: function icd
2: {
3:   local dirs ans ps3
4:   ps3="$PS3"
5:   PS3='select number(0:exit)? '
6:   ans=dummy
7:   while [ -n "$ans" ]; do
8:     dirs=(`find -maxdepth 1 -type d | sed
9:     's!^\./!!'`)
10:    echo "cwd is `pwd`"
11:    select ans in "${dirs[@]}"; do
12:      cd "${ans:-.}"
13:      break
14:    done
15:  done
16:  PS3="$ps3"
17: }
```

Linux 日記

第25回 メール配送 (12)

長く続いてきたメール配送の話もいよいよ大詰め。今回は、メールの送受信の際にユーザーが直接利用するMUA (Mail User Agent) についてお話ししよう。

文： 榊 正憲

Text : Masanori Sakaki

筆者は夏が大好きである。別に海で遊ぶとか山に行くとか、そういうことではない。小学校以来の伝統というか、夏になると何か工作や自由研究をしたくなるのである。一昨年は、久しぶりにプログラムを作って、古いワンボードマイコンをWindows上で復活させた。昨年は、レゴブロックで大きなクレーンを作った。最終的には、自由度11で完全電動、最大揚程は2m以上という大作になった。どちらも、一部の好事家の間でかなり受けた。今年はちよいと忙しいため、気合いの入った工作はできそうにない(温めているアイデアはあるのだが)。

科学教材社

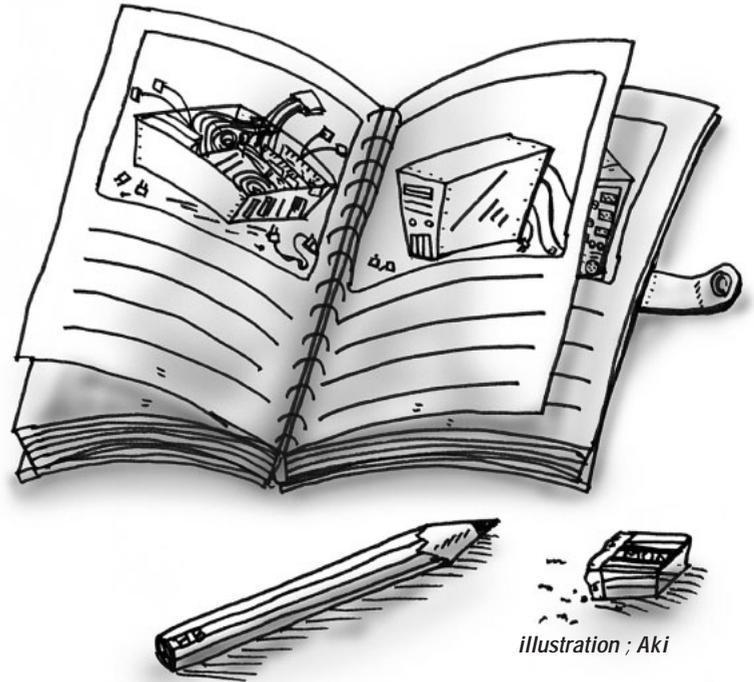
現在はだいぶ衰退してしまったようだが、筆者が子供だった高度成長期の末期ないしそのしばらく後ぐらいまでは、男の子の遊びにおいて、電子工作や模型工作は、確固たる地位を誇っていた。そういう理工系子供たちが読んでいた雑誌が、『子供の科学』や『模

型とラジオ』であった。これらの雑誌には、巻末に模型や電子工作のキットや部品、素材の通信販売のページがあった。この通信販売を一手に行っていたのが東京神田にある科学教材社である。しばらく前に、知り合いから科学教材社のWebページがあると聞いて(<http://www.kagakukyozaisha.co.jp/>) さっそく眺めてみた。さすがに時代相応に取扱商品は変わっていたが、本質はあまり変わっていないようで、鉄道模型やラジオコン、電子工作キットなどを相変わらず扱っているのが妙にうれしくなり、出かけてみた。最後に訪れたのは20年ほど前で、その時に比べると、お店はすっかりきれいになり、残念ながら当時のダンジョン的雰囲気はなくなっていた。

とはいっても、このようなお店がまだあったこと自体がうれしくて、何となく6石スーパーラジオの組立キットを買ってきた。ついでに、秋葉原の工具屋で、ANTEXのハンダごてやハンダ吸い取り器なども20年ぶりに新調し

た。工具のほうはラジオのキットより高くついてしまったが、まあ、よくあることだ。結局、これが今年の夏休みの工作となってしまった。

工作といっても、たかが6石スーパーのキットである。プリント基板に適当に部品をハンダ付けすれば、数時間でできあがりだ。と思って、なめてかかったのが失敗であった。組み立ては問題なく完了。受信感度調整でヘマをやった。我が家では、コンピュータが多数動いている。とにかくノイズだらけなのだ。そんなところで調整をやっているうちに、中間周波数トランスの調整用コアを回しすぎて、内部の電線を切ってしまった。もちろん、これではラジオは鳴らない。トランスといっても、1cm角程度のシールドケースに収められた直径5mm、長さ5mmほどのコイルである。このコイルに巻かれた0.1mm程度の銅線を1ターンだけほどいて、端子にハンダ付けしなめという作業を、ルーペとピンセットを使ってやるはめになった。老眼が始まった



(しかし老眼鏡は持っていない)筆者には恐ろしく辛い作業であった。

それでもとりあえず修復に成功し、ラジオは無事に鳴るようになった。何度やっても、自分で組み立てたラジオから音が出る瞬間というのはいずれのものである。少なくとも、組み立てたコンピュータでBIOSメッセージを見るよりは、はるかに感動的だ。

さて、電子メールの話である。しばらくsendmailから離れ、今回はMUAの話だ。

UNIXワークステーションを使う場合

ここまで解説してきたメールシステムの構成は、多少の例外はあるが、sendmailが動作するUNIXホスト上で、ucbMailなどのローカルMUAを使うという形だった。つまり、MUAはMTAであるsendmailを直接呼び出して実行し、メールを送信することができる。また、sendmailが受信し、binMailがスプールしたメールボックスファイルは、ローカルファイルシステム上にあるので、MUAから直接アクセスすることができる。

ユーザーが各種UNIXを使っていて、sendmailのようなMTAをローカルホスト上で運用し、DNSのMXレコードをきちんと設定し、さらにマシンを24時間運転していれば、完全なメール配信を行うことができるようになる。だが、これが非現実的であることは明らかだ。このように運用するためには、常時接続環境、独自ドメイン名の取得、ネームサーバの運用(あるいはアウトソーシング)、固有のグローバルIPアドレスの取得といった条件も求められる。組織規模では珍しくもないが、個人でこれを実現するのは、かなり大変だ。最近では、ドメイン名の取得要件が緩和され、実質的な常時接続の費用も安価になったが、それでもこの環境を維持するためには、かなりのパワーが必要だろう。

こういった環境が揃って、組織内、あるいは家庭でUNIXを使うとしても、ユーザーのワークステーション上でMTAを運用するというのは現実的ではない。たとえば、linux01.writers.takobeya.comというワークステーションでユーザーmasaがメールを

使うとしよう。メールの送信は問題なくできる。linux01上でsendmailを稼働させ、sendmail.cfを適当に記述すれば、masa@takobeya.comという発信アドレスでインターネット上にメールを送ることができる。では、受信はどうするか? takobeya.comのMXをlinux01.writers.takobeya.comにすれば、masa@takobeya.com宛のメールを受信できるようになるが、別のワークステーションを使っている別のユーザー宛のメール、たとえば、linux02.editors.takobeya.comを使っているユーザーyamada宛のメール(yamada@takobeya.com宛)までlinux01に届いてしまう。各ユーザーのメールアドレスを、user@takobeya.comという形式ではなく、ワークステーション名まで含むmasa@linux01.writers.takobeya.comという形式にし、それぞれのワークステーション向けのMXレコードを用意すれば受信できるようになるが、誰もこんな長ったらしいメールアドレスを使いたくないだろう。

やはりどうしても(少なくとも受信のためには)ワークステーションとは別に、組織を代表するハブサーバ(たとえば、user@takobeya.comという形式の宛て先アドレスを受理するmail.takobeya.com)を運用することになる。もちろん、takobeya.comのMXはこのハブサーバを指すようにする。これで、user@takobeya.comの形式のメールは、ハブサーバが受信できるようになる。mail.takobeya.comまで届いたメールを各ユーザーのワークステーションに配信する方法は、前回説明した通りだ。たとえば、リスト1のようにエイリアスを設定することによって、masa@takobeya.comという宛て先のメールを、masa@linux01.writers.takobeya.com宛に転送することができる。

Column

ダンジョンのお店

一部業種のお店というのは、長年営業を続けていくうちに、迷宮的な要素を帯びてくるものである。最新の製品と、いつのものかわからない製品がまぜこぜにあり、通路を通るだけでも一苦労、目的のものがどこにあるのかさっぱりわからないといったお店である。筆者の行動範囲では、ジャンク屋や模型屋がこのような傾向になることが多いよう

である。ダンジョンなので、奥深くまで進入するためには、お客のほうにもそれなりの資質が求められる。ある程度の経験値がないと、買い物という目的が果たせないのである。しかしダンジョンゆえに、お宝が転がっていることも多い(もちろん、宝物を判定するためには、それなりの経験値が必要だ)。ジャンク屋にしる模型屋にしる、最近はこのようなお店がかなり減ってしまい、さみしい思いをしている。

リスト1 mail.takobeya.com上のエイリアス

```
masa: masa@linux01.writers.takobeya.com
yamada: yamada@linux02.editors.takobeya.com
```



この方法を使えば、とりあえず各ユーザーのワークステーションでメールの送受信ができるようになる。メール送信は各ユーザーのワークステーションが直接行う。あるいはハブサーバに中継させてもいい。受信はmail.takobeya.comという代表サーバが行い、それが各ユーザーのワークステーションに中継するという形だ。

この方式では、受信メールが各ユーザーのワークステーションでしか読めないという欠点がある。宛て先ユーザー名に応じて、特定のワークステーションに配信されるからだ。ちょっと別のマシンからメールを見るといったことができない。また、ワークステーションは原則24時間運転にするとすると、サーバマシンほど管理の手間をかけないだろうし、設定変更やアップグレードなどで、何のためらいもなく止めてしまうことだろう。SMTP受信エージェントの長時間の停止は、エラーとして処理される。場合によっては、メールが発信者にエラーとして送り返されてしまうし、そこまで至らなくても、途中のサーバ（この場合はハブサーバ）で保留されることになる。好ましい状態でないことは確かだ。

管理上の手間もかかる。管理者は、各ユーザーのワークステーションすべてのsendmail.cfを管理しなければならない。ワークステーションのsendmail.cfはすべて同じものにできるかもしれ

ないが、管理しなければならないという点は同じだ。したがって、各ワークステーションでMTAを稼働させるというアプローチは、技術的に実現可能な構成ではあるが、実用的な構成とは言えない。

MUAとMTAの分離

個人のワークステーションで、常時稼働を前提としたsendmailのようなMTAを動かすというのは、あまり現実的ではなさそうだ。かといって、各ユーザーがtelnetなどでメールサーバホストにログインして、その上でサーバ上のローカルMUAを動かしてメールの読み書きを行うというのも、今や現実的ではない。特にユーザーが多い場合はほとんど不可能である。プロバイダのメールサーバのように、非常に多くのユーザーに対してメールサービスだけを提供するホストの場合、普通のシェル環境にログインするというのは冗長だし負担が大きすぎる。メールの保管のためのディスク容量も半端ではない。プロバイダ側のセキュリティの問題もある。ユーザーから見た場合も、ログインするのは避けたいはずだ。多くのユーザーはWindowsやMac、あるいはLinuxなどの環境を手元で動かし、メールもそれらのマシン上で読み書きできるに越したことはない。

結局のところ、ごく例外的な環境を除いて、現在の使用形態では、

sendmailなどのMTAが動作しているメールサーバ上でローカルMUAを直接動かすという使い方はまずしない。メールサーバではメールの送受信のみを行い、サーバ上ではユーザーのオペレーションは行わないのである。

メールクライアントプログラム

そもそも、MUAを稼働させる環境でMTAを運用できるのは、ワークステーションとしてUNIXを使っているからであり、MacやWindowsでは、はなから難しい相談だ。MacやWindowsには、sendmailのようなMTAはない。つまり、受信デーモンとしても動作し、発信時にはMUAから呼び出されるといった形のプログラムはないのである。たとえこのようなMTAを稼働させることができたとしても（技術的に、動かすことは不可能ではないし、サーバ運用向けとしては用意されている）、全マシンを24時間稼働にするというのは、UNIX以上に非現実的だろう。

そこで、1台のUNIXホストを多数のユーザーがTSSで使うという構成でもなく、また、各ユーザーのワークステーション上でMTAを稼働させるというものでもないアプローチでメールを使うための仕組みが作られた。メールシステムに関するコンピュータを、クライアントとサーバに分け、クライアントはメールの送受信をサーバに委任するという形にする（図1）。サーバは常時稼働しており、随時メールを受信し、各ユーザー用のメールボックスにスプールする。クライアントは、ネットワークを介して必要な時にサーバにアクセスし、受信メールにアクセスすることができる。メールを送信する場合は、まずクライアントのMUA上でメールメッセージ（いくつかのヘッダを含んだもの）を作成する。MUA

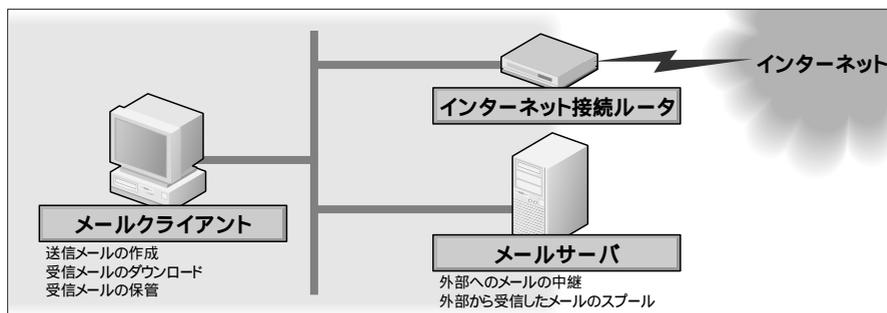


図1 ネットワーク接続されたメールサーバとクライアント

はこれをメールサーバに送信し、メールサーバは必要に応じてこのメールのヘッダ類を整えて、外部に再送信する。

このような形態で運用することにより、ユーザーレベルでは、24時間稼働、インターネット常時接続というMTAを運用する必要はなくなる。組織レベル、あるいはプロバイダで運用するメールサーバは常時稼働/常時接続である必要があるが、ユーザーは必要な時に自分のコンピュータの電源を入れ、随時メールを送信し、また、常時稼働しているメールサーバから自分宛のメールをダウンロードすることができる。

メールの送信

まずはメールクライアントアプリケーション(MUA)を使ってメールを送信することを考えてみよう。

MUAでは、メールの本文などのほかに、MUAでしか用意できないヘッ

ダを作成する必要がある。もっとも重要な情報は、宛て先を指定するTo:やCc:である。また、発信者を識別するFrom:、メールの件名を示すSubject:も指定することになる。Subject:はなくても構わないが、メールで情報を伝えるという観点からは、事実上必須の情報である。これらとは別に、発信日時を示すDate:、個々のメールメッセージを識別するMessage-Id:なども必要であるが、これらはMUAが生成しなくても、メールを配信するMTAが生成してくれる。もちろん、メールの作成時点で、MUA上で生成するのがまっとうなやり方である。

これらのほかに、メール本文の形式を示すヘッダ類も、MUAで生成するヘッダである。MUAで作られたメールは、たとえばリスト2のようになる。

これらのヘッダは、さまざまな方法で指定される。To:やCc:、

Subject:などは、メールを送信する際にユーザーが指定する情報である。Date:やMessage-Id:などは自動生成される。From:で指定する発信者の名前やメールアドレスは、MUAの初期設定パラメータとして登録しておくことになる。

送信されるメールが、実際にどのような形式になるかは、MUA次第である。X-で始まるヘッダは、MUAなどが自由に定義できるものであるし、MimeやContent系のヘッダは、メールの本文に関する情報だ。ここで重要なのは、Message-Id:、Date:、To:、From:、Subject:といったヘッダがMUAのレベルで用意されるということである。

このようにして作成されたメールは、SMTPを使って適当なMTAに送信される。もちろん、宛て先アドレスについてDNSにMXレコードを問い合わせ、宛て先MTAに送るということも不可能ではないが、メールクライアントアプリケーションの場合は、普通は、送信メールの宛て先に関らず、特定のMTAホストに接続して送ることになる。あとは、このMTAが宛て先を評価して、目的のMTAに中継するのである。MUAがメールを送信するホストは、メールクライアントアプリケーション上で、初期設定パラメータとして指定する(MUA上では、SMTPサーバといった名称で示されている)。たとえば、組織でmail.takobeya.comというメールサーバを運用しているのであれば、このMTAホストにメールを送り、中継してもらおうという形になるだろう。プロバイダのメールサーバを使う場合であれば、mail0.provider.ne.jpといったMTAホストに中継してもらおうことになるだろう(図2)。

MTAに達してしまえば、今まで説

リスト2 MUAが作成するメールの例

```

Message-Id: <4.2.0.58.J.20010811154944.00afc5c0@mail.takobeya.com>
X-Sender: masa/mail.takobeya.com@localhost
X-Mailer: QUALCOMM Windows Eudora Pro Version 4.2.0.58.J
Date: Sat, 11 Aug 2001 15:50:08 +0900
To: nobu@ascii.co.jp
From: Masanori Sakaki <masa@takobeya.com>
Subject: Test
Mime-Version: 1.0
Content-Type: text/plain; charset="ISO-2022-JP"; format=flowed
Content-Transfer-Encoding: 7bit
    
```

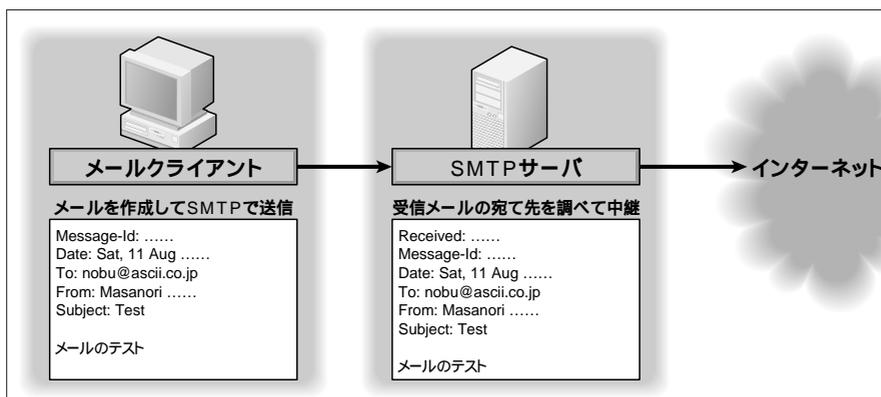


図2 メール送信



明してきたように、MTAがメールの宛て先を適当に評価し、適切な配信エージェントを選択し、メールの配信を行う。そして、宛て先アドレスを管轄するMTAに達したら、ユーザーがそのメールを読めるように、メールボックスにスプールする。MTAから見た場合、最初にメールを作成したMUAは、ほかのMTAと等価なものである。SMTPで接続を行い、適切な情報をやり取りし、メールを受け取っているだけである。実はこの部分に、現在のインターネット電子メールシステムの大きな問題があるのだが、そのことについてはまたしばらくしてから解説しよう。ここでは、MUAで作成されたメールは、SMTPにより指定した中継MTAに送られるということ、メールを最終的な宛て先に届けるという作業は、中継MTAによって行われるということ覚えておいてほしい。

メールの受信

メールの送信処理は、MUAからMTAへの配信にSMTPを使うという点で、今まで説明してきたsendmailの動作と特に変わることはない。以前にtelnetでSMTPコマンドを直接sendmailデーモンに送り、メールを送信するという実験を行ったが、MUAはまさにこれを行っているプログラムなのである。メールを作成するインターフェイスをユーザーに提供し、必要なヘッダを持ったメールメッセージを作成し、適切なSMTPコマンドを使ってこのメールメッセージをMTAに送っているのだ。

一方、受信したメールをユーザーのワークステーション上で閲覧するという処理を行うためには、まったく別のシステムを導入する必要がある。ある宛て先に届いたメールは、その宛て先

ユーザーを管轄するMTAホスト上のメールボックスにスプールされておしまいだからだ。SMTP配信や通常のMTAには、クライアントに受信メールを送るという機能はない。このMTAホスト上で動作するローカルMUAなら、メールボックスにアクセスして受信メールを閲覧することができるが、別のコンピュータ上のユーザーは、メールボックスにアクセスすることができないのである。そのため、ここまで説明してきたMTA間のやり取りやSMTPなどはまったく別の仕組みが必要になる。

これに使われるのが、POP (Post Office Protocol) やIMAP (Internet Message Access Protocol) というプロトコルと、このプロトコルを使って通信を行うサーバである。

POP

ネットワークで接続されたクライアントとサーバがあり、サーバには、ユーザー宛のメールが届いている。ユーザーが、クライアントでこのメールを見る場合、サーバからクライアントにメールをダウンロードする必要がある。これを行うのがPOPという通信プロトコルと、その通信処理を行うサーバだ。そして、POPで通信を行い、受信メールをクライアントにダウンロードするサーバをPOPサーバという。

サーバ上にスプールされたメールをクライアント側にダウンロードするという場合、必要な処理は次のようなものだろう。

・ユーザー認証

メールを誰でも見られるというのは好ましくない。そのため、サーバとクライアントの間でメールをやり取りする前に、ユーザー認証が必要である。

・情報の一部取得

メール全体を見る前に、スプールされているメールの発信者や件名など、一部の情報を見ることができれば便利である。また、実際にダウンロードするためには、メール一覧といった情報が必要になるだろう。

・個々のメールのダウンロード

POPの最大の目的は、サーバ上にスプールされているメールをクライアント側にダウンロードすることである。すべてのメールをダウンロードするというアプローチよりは、一覧情報に基づいて、1つずつダウンロードできるほうが便利だろう。

・メールの削除

受信メールをクライアント側にダウンロードしたら、サーバ上のメールは不要になる。サーバは多くのユーザー

Column

Bcc : による宛て先指定

メールの宛て先は、一般にTo : かCc : で指定するが、Bcc : でも指定できるということ覚えておいてほしい。Bcc : による宛て先指定は、メールボディ中からは削除され、エンベロープ情報にのみ現れるので、受信したユーザーはこの宛て先情報を見ることはで

きない。MUAでBcc : を使って宛て先を指定した場合は、メールをSMTPで中継する際には、MUAとMTAの間でやり取りされるエンベロープ情報として、Bcc : で指定した宛て先が使われる。SMTPで配信されるメールボディには、宛て先情報は含まれていない。したがって、Bcc : による宛て先一覧を隠したメール配信は、メールクライアントを使う場合でも問題なく行うことができる。

が共有するリソースであるから、不要なメールは削除し、ディスクスペースを空ける必要がある。

POPは、こういった処理をネットワークを介して行うためのプロトコルである。POPにはいくつかバージョンがあり、現在広く使われているのはバージョン3 (POP3) である。メールをスプールしているMTA上には、POPによるリクエストを受理し、応答を行うためのデーモン (pop3dなど) が用意されている。POPには表1のようなコマンドが用意されている。

UNIX上でPOPデーモンを動かす場合は、ユーザー認証は通常のUNIXのユーザーアカウントを使用することになる。プロバイダのメールサーバなど、非常に多数のユーザーのメールを管理する専用システムの場合は、独自のユーザーアカウントデータベースに基づいて認証を行うことになるだろう。

MUA側は、メールサーバに接続して、POPデーモンと対話するための機能を持っている。接続するPOPサーバホスト、ユーザー認証で示すユーザー名などは、MUA側の初期設定パラメータで指定しておく。パスワードも登録してしまえば、接続を完全に自動で行うことができるが、セキュリティの面では多少甘くなる。

MUAがPOPデーモンに接続し、認証されれば、サーバ上にスプールされているメールにアクセスできる (図3)。

機能	説明
USER	接続ユーザーの指定
PASS	パスワードの指定
QUIT	終了
LIST	メール一覧
RETR	個々のメールのダウンロード
DELE	スプールからメールを削除
TOP	ヘッダのダウンロード
UIDL	メッセージIDの問い合わせ

表1 POPの主要なコマンド (機能)

MUAの標準的な動作は、スプールされているメールをすべてダウンロードし、その後、サーバ上からメールを削除するというものだ。実際には、より細かな設定を行うことができる。たとえば、メールをダウンロードするが、サーバ上から削除しない、あるいは指定日数が経過してから削除するという設定を行える。また、大きなメール (ファイルが添付されているメールなど) はダウンロードしないといった指定も可能だ。携帯端末などからメールを調べる場合などには、こういった機能を使うことになるだろう。

これらの処理は、ユーザーが指定した時に行うことができるが、通常は、メールクライアントアプリケーションを起動したとき、そしてその後一定間隔おきに行うというのが普通だ。

複数のクライアントがある場合

POPを使ってメールをMUA側にダウンロードするというアプローチは、基本的に、受信メールをMUA上で管理するという考え方に基づいている。POPサーバとして動作するMTAは、メールを一時的に保管する場所であり、メールの最終的な保管場所はMUAが動作しているワークステーションとなる。通常はこの方法で問題ないのだが、ひとりのユーザーが、複数のコンピュータや携帯端末を使ってメールにアクセスする場合は問題となる。たとえば、ふだんは会社のマシンを使ってメール

のやり取りを行うが、外出時や帰宅時は、ノートPCを使って、ダイヤルアップでメールにアクセスするといった場合だ。もちろん、この2台のコンピュータで使うメールアドレスは同一のもので、それを管理するPOPサーバも同一のホストである。

問題は、一方のコンピュータでメールをPOPサーバからダウンロードし、メールをサーバ上から削除してしまうと、他方のコンピュータは、もはやそのメールにアクセスできないということである。たとえば、会社のワークステーションでダウンロードし、削除したメールは、ノートPCからアクセスできない。

このようになる理由は明らかだ。POPは、受信メールを、ダウンロードしたMUA上で保存するという考え方をういているからだ。MUAが複数あれば、分散してしまうのは仕方がない。ダウンロードしたメールを削除しないという方法もあるが、この場合は、複数のMUA上に同じメールのコピーが存在することになるし、POPサーバ上でスプールされているメールの容量も大きくなってしまふ。プロバイダのメールサーバを使う場合は、容量の制限や、古いメールの自動削除なども障害となる。組織のサーバであれば、多少の無理は効くだろうが、多くのユーザーがこのような使い方をしたら、スプールディレクトリはあっという間にあふれてしまうだろう。以下次号……。

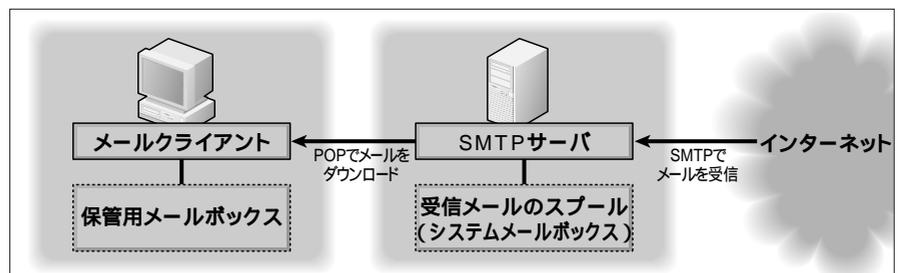


図3 POPによるメールのダウンロード

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18回

- 【秋葉原】(あきはばら)
- 【東京三菱銀行】
(とうきょう・みつびし・ぎんこう)
- 【電気街口】(でんきがいくち)
- 【PCパーツショップ】
(ぴーしー・ぱーつしょっぷ)
- 【リナックスカフェ】(りなっくすかふえ)

秋葉原

【あきはばら】

自作系 Linuxer にとっての聖地。ラジオ・ジャンク族によって開拓され、カデーン人に統治されていた秋葉原だが、西暦 1990 年代初頭、おりからの自作 PC ブームにのり、PC パーツショッ



聖地巡礼

プが雨後の筍のごとく増殖。自作系 Linuxer の祖先にとっては、パピロニアのネブカ DOS/IV ネザル大王による侵略ほか幾多の迫害の舞台となるなど、まさに苦難に満ちた地であった。しかしその後、唯一神・モノリシックカーネルとの契約により、出マハーポーシャを果たした Linuxer たちは、ふたたびチチブ電機集に集い、自分たちの国家を打ち立てたのである。残念ながら、その後、かの地では幾度となく版図が描きかわり、過去の豊穰なる王国は失われてしまったが、現在でも全国に散り散りになった自作系 Linuxer の間では「週に一度は巡礼すべき地」とされている。

長い歴史をもつ都市、秋葉原は、その数奇な運命の中で自作系 Linuxer 以外の宗教にとっても聖地としてあがめられ

るようになった。そのひとつが「アニメオタク」である。偶像を排し、ソースコードのみを信奉する Linuxer と異なり、2D・3D 問わず、あらゆるめがねっ子、猫耳っ子を偶像崇拜する彼らだが、唯一神であるモエ～の降り立った地として秋葉原を尊ぶ点では共通している。

また、新興勢力である「エロゲーマー」の間でも、中古 98 用ソフト、同人ソフト買い込みの拠点として秋葉原を巡礼する習慣が定着している。これら世界の 3 大宗教がなぜ聖地を同じくするのかというナゾは、古代史でもっともミステリアスなテーマとして多くの学説をよんでいる。「自作系 Linuxer = 大なり小なりアニメオタク = 大なり小なりエロゲーマーだから」と主張する「三位一体説」もそのひとつだが、あまりに事実近すぎるとしてクレルモン公会議にて異端審決が出されたことは記憶に新しい。

東京三菱銀行

【とうきょう・みつびし・ぎんこう】

秋葉原中心部に支店を構える、ほぼ唯一の都市銀行。現金を引き出すスポットがなく、“銀行砂漠”とまでいわれる秋葉原において貴重なオアシスとなっている。東京三菱銀行自身にとっても、



銀行前と言えよ？！

激化するいっぽうの金融業界再編の流れのなか、秋葉原支店の稼働率の高さとそこから生み出される手数料収入の多さは無視できないものがあり、ここを核にリテール市場へのテコ入れをはかるという声も行内では強い。

しかしながら、一部パーツショップではデビットカードが普及、そして東芝 PC テクノセンタ脇の郵便貯金キャッシュサービスでは提携金融機関を拡大するなど、徐々にその地

位を脅かす要素がそろいはじめている。これを警戒した東京三菱銀行では、店舗前で行員みずから焼きそば、クレープなどの露店を展開。利用者を少しでも引きとめようと涙ぐましい努力を続けている。読者諸兄も、秋葉原参内のおり彼らが額に汗しているところを見かけたならば、「銀行員もたいへんですね」と労をねぎらってあげていただきたい。

電気街口

【でんきがいくち】

毎土日午後に国内随一の人口密度を記録する、JR秋葉原駅の改札口のひとつ。街自体がひとつの巨大宗教施設となっている聖地・秋葉原の中で、この改札口および手前の長い階段は参道の役割を果たしている。人混みに揉まれるという苦難を味わった巡礼者は、やがて改札口から自由な新世界＝秋葉原へと解放される。

これはまさに胎内 誕生の過程を経る出産を巡礼者に疑似体験させることをねらったものであり、日本的“ハレ”の儀式なのである。儀式を終えた自作系 Linuxer は、開放感に突き動かされてついつい不要なパーツまで買い込んでしまう。これは“ケガレ”をはらう意味があり、その後1週間は無駄な浪費を防げるという効果もある。

PC パーツショップ

【ぴーしー・ぱーつしよっぷ】

「相性問題での返品は受け付けません」と、デキの悪い娘を嫁に出した実家のような看板を掲げている、秋葉原に多い店の一群のこと。

「いらっしやいませ」

「すみません。このあいだ

こちらで購入したネットワークカードなんですけど」

「はいはい。この商品ですね。確かに当店で販売したカードのようですが、どうかしましたか？」

「とりあえず挿して通電してみたら、いきなり火吹いてぶっ飛んじゃったんですよ」

「ははー。そうですかー」

「そうですかーって、あなた火ですよ。火。まあ、とにかく不良品なんで返品しに来たんですけど」

「返品ですか。お客さまの生年月日伺ってよろしいでしょうか？」

「なんで生年月日なんか……。1970年6月30日だけど？」



さるかにバトル

「お調べしますので少々お待ちください……。申しわけありません。当店では相性問題での返品は受け付けておりませんので、ご希望にそいかなるのですが」

「そうですか……。って、火吹いてのになんで相性なんだよ！」

「逆ギレされても当店の規則ですから困ってしまうんですが。落ち着いてくださいよ。当店では相性判断に際して、独自の“真・動物占い”を採用しているんです」

「なんじゃそりゃ！ っていうか“真”ってなんだ！ 真って」

「お客さまの生年月日ですと、真・動物占いでは“サル”にあたるんですよ。サルタイプの性格はですね、“わがままであてんぼう。でも、頭の回転は速いほうかも。逆にそのせいで周りからズル賢いっていわれることもありそう。気をつけて”ですって」

「よけいなお世話だ！ っていうか、サルだからなんだっていうんだ！」

「それがですねえ。お客さまのお買いあげになったカードは Realtek 製ですから、これがもう最悪の相性で……」

「ここまで引っ張っておいて、オチはサルカニ合戦かよ！」

リナックスカフェ

【りなっくすかふえ】

地域活性化のため、千代田区ほか企業が秋葉原地域に設立を予定している施設。

Linux をキーワードに事業を展開する。1階売店では、「Linux まんじゅう」「Linux せんべい」「Linux カステラ」などを販売。カフェという名のとおり、店内での飲食にも対応する。2

階は地元青年団のための交流スペース。カラオケ施設完備で、季節によっては「よさこいソーラン Linux 音頭」の練習場にも使える広さが自慢。千代田区は「Linux は町おこしにピッタリ。いつかは日本一の Linux の町と呼ばれたい」と抱負を語る。

ハアー 俺が町サ Linux がやってきたー（ヨイヨイ）

インストールはしたけれど

近ごろ とんとごぶさただね

近ごろ とんとごぶさただね（ヨイヨイ）

吾作んとこの 女房とおなじさね

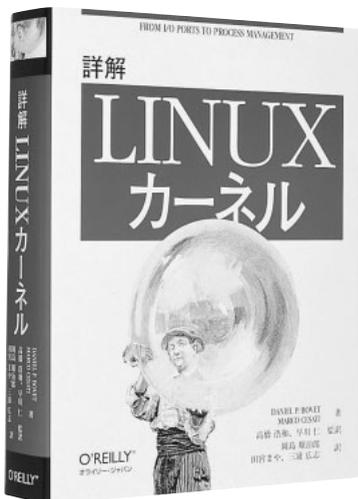
ハア おなじさね ハア おなじさね

（よさこいソーラン Linux 音頭）



そのほかのおみやげ

Books



詳解 Linuxカーネル

Daniel P. Bovet, Marco Cesati 著 / 高橋浩和、早川 仁 監訳 / 岡島 順治郎、田宮まや、三浦広志 訳
オライリー・ジャパン / オーム社
 B5変形判 / 780ページ 本体価格 5800円

フリーソフトウェアであるLinuxの大きな特長は、ソースコードが公開されていて、誰でもそれを閲覧、あるいは改変する自由が保証されていることだ。1980年代までは、本格的なオペレーティングシステム(OS)のソースコードが無償で公開されるなんて夢物語のようだった。とはいえ、ほとんどの人にとって、Linuxカーネルのソースコードを読み解くのは非常に困難な作業だ。

本書は、x86アーキテクチャで動作するLinux 2.2.14を対象に、メモリやプロセス、割り込みの管理、システムコールやシグナルの仕組みなど、Linuxカーネルがどのような仕組みで実装され、動作しているのかを解説するものである。もともとは、イタリアの工科大学でOSの講義を行うための教材として書かれたものなので、解説は非常に丁寧になされている。ソースコードとあわせ、1年くらいの時間をかけてゆっくり読むことをお勧めする。

究極!! X68000エミュレータ

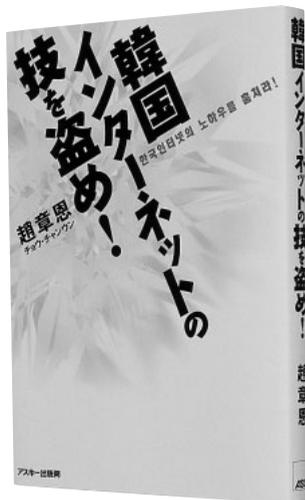
高木啓多 著
 秀和システム

A5判 / 256ページ / CD-ROM1枚付き

本体価格 2500円

X68000といえば、16ビットCPU (Motorola MC68000)、6万5535色同時発色、2ch・8和音FM音源など、当時としては画期的なスペックを誇り、現在でも通用する先鋭的な筐体フォームとあいまってユーザーからの熱い支持を得ていた。そして、その高いスペックを最大限に活かしたのがゲームソフトウェアであった(というか、ゲームくらいでしか活かせなかったのだが)、現在でも、本体がネットオークションで売買されていたり、一部のマニア達がプログラム開発を行っていたりと、根強い人気を保っている。

本書は、そんなX68000のエミュレータ (WindowsとMacに対応) とゲームソフトを紹介する1冊だ。もちろん付属CD-ROMには、それらのソフトが収録されている。はっきり言ってLinuxとはまったく関係ないのだが、なつかしさのあまり取り上げただいである。読者の中にも「ベケロッパ」と聞くと郷愁を誘われる方もいるのでは?



韓国インターネットの技を盗め!

趙章恩 (チョウ・チャンウン) 著
 アスキー

四六判 / 224ページ

本体価格 1400円

「つくる会」の教科書問題、小泉首相の靖国参拝騒動など、最近何かとお隣、韓国との関係がよろしくないようだ。こんなことで来年の世界カップは大丈夫なのかと、サッカー好きの私などは心配になるのだが.....

それはさておき、グルメ、映画に続いてマスコミで取り上げられる機会が増えてきたのが韓国の情報通信事情である。TVのニュース番組や新聞で特集が組まれ、「どうやらインターネットの分野では日本より進んでいるらしい」ことが伝えられるようになった。しかし、韓国版インターネットカフェ「PCパン」でネットゲームに熱中する若者の姿などが紹介される程度で、その全貌が伝えられることは少なかった。本書は、ネットベンチャー企業の分析を中心に、政府の取り組みやユーザーの動向など幅広い視点で韓国のネット事情の実態を捉えている。刺激的なタイトルとは裏腹の落ち着いた語り口にも好感が持てる。



エニャック 世界最初のコンピュータ開発秘話

スコット・マッカートニー 著 / 日暮雅通 訳
 パーソナルメディア
 四六判 / 288ページ

本体価格 1900円

大学の授業や企業の研修でコンピュータについて学んだ経験のある方なら、「エニャック」が何であるのかおわかりだろう。入門テキストの最初のほうで「コンピュータの父」John von Neumannとともに必ず登場するのが、「世界最初のコンピュータ」エニャック (ENIAC: Electronic Numerical Integrator and Computer) である。総重量30トン、床面積1800平方フィート、搭載する真空管1万7468本、消費電力174キロワットという、このモンスターマシンを設計し、開発の指揮をとったのが本書の主人公John W. MauchlyとJ. Presper Eckertである。本書はMauchlyとEckertの生涯を追いながら、2人のコンピュータ開発にかけた夢と情熱をドラマティックに描いている。NeumannとJohn V. Atanasoff (現在のコンピュータの原型となる装置を開発した人物) に対する評価が卒すぎるくらいはあるが、冒頭から一気に読ませるヒューマンドキュメンタリーの力作に仕上がっている。

Linuxファイアウォール



ロバート・L・ジングラー 著 /
 依田研司 訳

ピアノ・エデュケーション

B5変形判 / 548ページ

本体価格 4200円

お気楽Q&A Linux



たなかひろし 著

IDGジャパン

B6判 / 288ページ

本体価格 1500円

Unixバックアップ&リカバリ



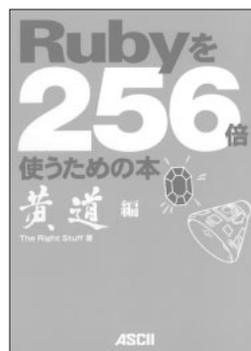
W. Curtis Preston 著
 長原宏治 監訳 / 田和勝 訳

オライリー・ジャパン /
 オーム社

B5変形判 / 760ページ

本体価格 6600円

Rubyを256倍使うための本 横道編



The Right Stuff 著

アスキー

B6判 / 264ページ

本体価格 1200円

読者の声

俺にも
いわせろ!

カレンダーもデスクトップもデパートも、秋一色だというのに、まだまだ暑い日が続きます。

初夏の後悔の涙も乾かぬうちに、今年も真っ黒になるまで焼けてしまいました。そんな私の財布は緩みっぱなしです。ウーロン茶色の肌を見ると、美白用品を買わずにいられない。「美白成分と保湿成分を配合」、「シミ・そばかすを防ぎます」、「うるおいのある肌に」。ああ、なんて魅力的な言葉でしょうか。気がつけば、レジを済ませている私……。

9月号付録CD-ROMについて

今回の付録CD-ROMは日本語インストーラが入っていなかったのですがFTP版だからですか？ メビウスPC-100Jにインストールしようとしたのですがインストーラがエラーを起こし失敗しました。旧バージョンではインストールできました。CD-ROMに問題は無いのでしょうか？ デスクトップ等ではまだ試していません。

(田中 多津男さん)

◎申し訳ありません。9月号の付録CD-ROMでは、インストール時に日本語表示を選択することができませんでした(インストール終了後は日本語環境を使用可能です)。これは、編集部でCD-ROMイメージを作成する際に、設定を間違えたことによるものです。弊誌をお買い上げくださいま

した皆様に、ご迷惑をおかけしたことを深くおわびします。以後、このようなことがないように、十分に注意いたします。

日本語でインストールするためには、起動用フロッピーディスクを作成し、そのフロッピーディスクから起動する必要があります。手順につきましては、弊誌ホームページと今月号の付録解説ページに記載致しましたので、ご参照ください。

コマンドマスター 呪文使いへの道

待ってましたよ、コマンドマスター特集！パワーの低いマシンでLinuxを使用しているので、どうしてもコマンドラインで作業をすることになってしまいます。ですので、今回の特集は非常に有益で助かります。ありがとうございました！

(伊藤真和さん)

Linuxコマンド攻略ガイドは大変参考になりました。コマンドリファレンスは所持していますが、システム中での使用方法とか例など、わかりやすい解説でとても良かったです。

Webブラウザカタログも参考になりました。Linuxをインストールしたら、デフォルトで3~4種類のブラウザがインストールされているので、今回の記事を参考にいろいろテストしてみたいです。今回のような、フリーのアプリケーションについて、使用方法等の解

説を特集していただけないでしょうか？ Linuxの世界が広がってますます抜け出せなくなりそうです。これからも頑張ってください。

(三好哲史さん)

UNIX(Linux)を知るにはまずはコマンドラインから……。ということで余計なGUIは不要！と早や半年。MS-DOSとの違いに苦労して今は起動もしていない状態。そんなこんなで、まだ軽く目を通しただけなのですが、今回の特集は「もう一度」と思わせてもらうのに十分でした。アンケートの自己評価がワンランクアップできるよう、もう一度励んでみたいと思います。

(EAST-MOUNTさん)

今月号の特集は、とてもためになりました。そこいらに売っているコマンド本よりも、より実践的に詳しく解説してあり使用例などもあってわかりやすかったです。できれば特集ではなく、別冊付録の形式であったならより一層便利に利用できたと思います。

(せんたさん)

◎9月号の特集の扉ページは、呪文使いのイメージでいこうということになりました。いくつかのイラスト案の中から扉に使うものを決めるときのこと「女の子が手前で、大きいほうが良いんじゃない？」「女の子

のmantleが透けていたほうが良いんじゃない?」……。エロおやじばかり。

電気代と猛暑とパソコン

最近Linuxマシンをサーバ機として1ヵ月フル稼働した結果、今月の電気代が5000円近く多く請求されました。パソコンの電気代のせい、最近の猛暑によるエアコンのつけ過ぎ、原因がいまいちわかりません。こんなに請求されたのは初めてなので、パソコンの電気代を検証してみようと思います。貴誌でもぜひ! 特集をお願いします。

(宮下直記さん)

暑い日がまだ続きますね。我が家のPCルームは、クーラーを入れていてもなんだか妙に暑いです。なので、クーラー+冷風機をフル稼働させています。今月の電気代が恐ろしい……(^_^;)。皆さんはマシンの熱対策ってどうされているんでしょうか?

(井上 美由紀さん)

④クーラーを入れてもまだ暑いとは、クーラーの効きが悪いのでは……。実は、担当の部屋のクーラーの効きが良くないのです。水漏れしたり、23度以上に設定すると部屋が臭くてたまらないなど。大家さん、エアコンの掃除もしてから部屋を貸し出してもらえませんか?

担当の周りでは、マシンの熱対策は冷房に限るとの意見多数。ここは思い切って、PCルームで過ごすことをあきらめて、リモート操作するというはどうでしょう?

パソコンの電気代を計測して、タスクバーに表示するフリーソフトを見つけました。これを使えば、パソコンの電気代を常に把握できます。でも、ちょっと焦っちゃいますよね。

深夜の研究室

我が研究室にあるマシンのCD-ROMは、深夜に突然開いたり閉ったりします。お祝いしてもらったほうがよいでしょうか?

(泥んこ次郎さん)

④むむ。近くに以前焼き鯖になってしまったマシンがありませんでしたか? マシンは無念の思いから成仏しきれず、魂だけこの世に残ってしまっています。ぜひ、早急にお祝いしていただけてください。

ブラウザ

「特集2」Webブラウザカタログ2001・夏の企画は大変参考になりました。ネスケだけでなく、いろいろなブラウザを試そうと思います。きちんとインストールできたら、の話ですが……。

(遠藤浩二さん)

Mozillaの完成度がかなり上がってきたので、うれしく思う今日この頃ですが、9月号の特集を見て、タブ対応のブラウザというのなかなかおもしろそうだと、気になってしまいました。

(森本 雄一郎さん)

④編集部では、IEコンポーネントを利用したタブ対応ブラウザの「BugBrowser」が人気です(Windows環境での話ですが)、担当も、皆のマネをしてBugBrowserをインストールしてみました。Hotbarを愛用する担当としては、Hotbarの使えないBugBrowserに少々不満あり。でも、タブ機能に心揺られる今日この頃。

システム設定

4月にADSLを導入して、常時接続が

にサーバを構築(それでも設定をまねするだけ)することにしました。

そこでLinux magazineのバックナンバーでサーバ構築に関わる記事をかき集め、さらに付録にあったVine Linuxを利用して、3ヵ月かけてこつこつ設定してきました。ディストリビューションの差異によるものかわかりませんが、うまく設定できないところもあり、これらは数々のメーリングリストを参考にして設定しました。ADSL回線は安定しており、我が家のサーバは順調に稼働中です。ここまでできたのも貴誌のおかげです。ありがとうございます。

今後は、各種サーバの設定を調整していき、Linuxを自由に使えるよう頑張っていきたいと思います(調整方法の記事を組んでいただけると参考にできるのですが……)。

(おがちゃんさん)

サーバをメンテナンスするときに運用担当者は実際に何をするのか、こういうことはほとんど書かれていないような気がする。概論ではなく、具体的に。運用システムの内容にもよるのだろうが、サーバのチェック項目や、それらをどのように記録・管理するのか、実際に使っているコマンド群など。記載されていたら即買いですね。

(野里元宏さん)

④ 自宅のLinux専用マシンで、Linuxを利用されている方や、常時接続のサーバを立てている読者さんがずいぶん増えているようですね。

サーバの運用方法などについて、近いうちに特集する予定です。サーバ運用の記事について、ご要望などがありましたら、ご意見いただければ幸いです。

付録CD-ROMに収録した

LASER5 Linux 7.1の インストール

本誌付録のCD-ROMに収録したLASER5 Linux 7.1はFTP版です。非商用ソフトだけが含まれており、製品版を販売しているレーザーファイブ株式会社からサポートを受けることはできません。

また、CD-ROMのメディアに不良があった場合は、お手数ですが住所、氏名を明記のうえ（linux-cd@ml.ascii.co.jp）宛にご連絡くださるようお願いいたします。なお、Linuxの設定などについてのご質問にはお答えできませんので、あらかじめご了承ください。

インストールの前準備

これからLASER5 Linux 7.1のインストールを始めます。多くのネットワークカード、ビデオカード、モニタは自動認識されますが、自動認識されない場合に備えて、あらかじめハードウェアのマニュアルなどを用意しておく、インストール中にあわてなくて済みます。

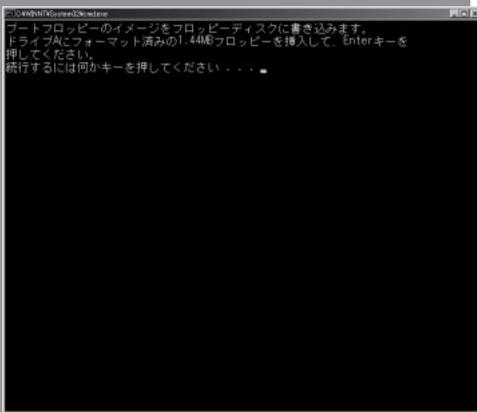
また、レスキュー用のフロッピーを作成するために、空のフロッピーを1枚用意しておいてください。CD-ROMブートができないマシンを使う場合は、別にインストーラ起動用のフロッピーも用意します。

さて、Linuxのインストールには、Linux専用に使えらるディスク領域が必要です。ハードディスクにLinux専用に使えらる領域を作成するか、ディスクを増設してインストールに備えましょう。

ブートディスクの作成

CD-ROMからインストーラを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。

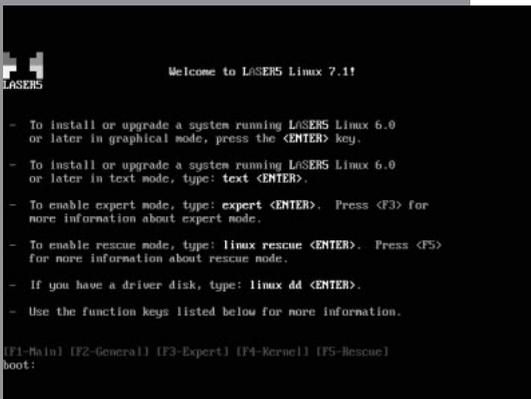
まず、Windowsマシンに、空のフロッピーディスクとCD-ROMのDisc 1をセットして、エクスプローラでCD-ROMのフォルダを開きます。[mkboot]をダブルクリックするとDOS窓が開くので、[Enter]を押します。再度[Enter]を押すと、インストーラ起動用のディスクの作成が始まります。ディスクの作成が終わり、再度[Enter]を押すとDOS窓が閉じます。



インストーラの起動

作成したフロッピーディスク、または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て[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。

グラフィカルなインストール画面がうまく表示されない場合は、[boot:]の箇所ですべて[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。



使用言語の選択

インストーラを英語表示させる場合は [English] を、日本語表示させる場合は [Japanese] を選択します。ここでは、[Japanese] を選択します。

4

キーボードとマウスの設定

デフォルトでは [モデル] に [Japanese 106-key] が、[レイアウト] に [Japanese] が選択されています。日本語キーボードを使用する場合は、このままの状態です。[次] を押します。

次はマウスの設定です。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ト] 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。これは便利な機能なので、チェックしたまま [次] を押します。

インストールタイプの選択

[ワークステーション] [サーバシステム] [ラップトップ] を選択すると、それぞれデスクトップ用、サーバ用、ノートPC用にLinuxがインストールされます。ただし、パーティションやブートローダLILOが自動で設定されるので、マシンをLinux専用にする場合以外は、それらを柔軟に設定できる [カスタムシステム] を選択するのがよいでしょう。[アップグレード] は、すでにインストールされているLASER5 Linuxをアップグレードするための項目です。

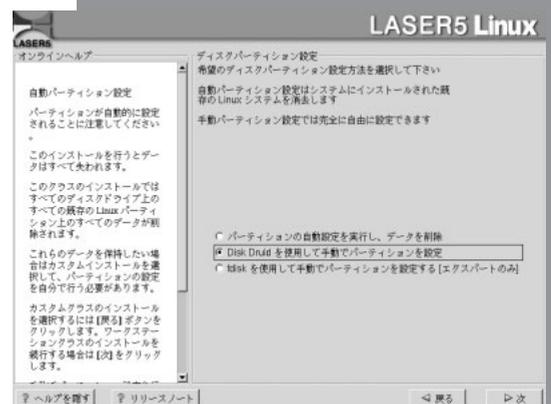
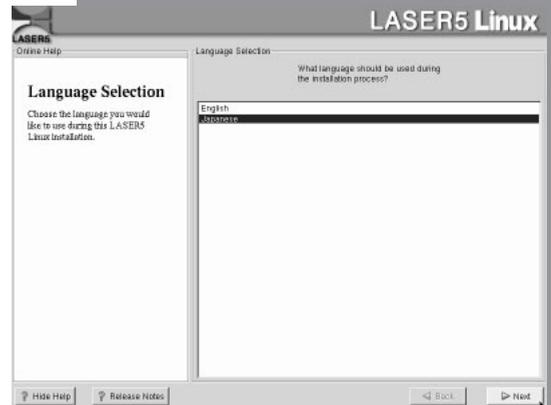
ここでは [カスタムシステム] を選択したものととして、解説を進めます。

パーティション設定ツールの選択

[fdisk] は柔軟にパーティションを設定できますが、コマンドで操作するので初心者にはおすすりできません。

[Disk Druid] はやや柔軟性に欠けるものの、マウスを使ってパーティション設定できるので初心者におすすりです。

ここでは [Disk Druid] を選択したものととして解説を進めます。





パーティションの作成

Linuxのインストールには最低限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。

まずはLinuxシステム用のパーティションを作成します。[追加]を押すと、小さいウィンドウが別が開かれるので、[マウントポイント]に[/]を、[パーティションタイプ]に[Linux native]を選択して、ディスクを約2.3GBバイト消費するので、フルインストールの場合Linuxシステム領域を3~4GB程度作成して[OK]を押します。

次にSwap用パーティションを作成します。[追加]を押して開かれるウィンドウで、[マウントポイント]にはなにも指定せずに、[パーティションタイプ]に[Linux swap]を選択して、マシンが搭載するメモリの1~2倍程度のサイズを指定します。2種類のパーティションを作成したら、[次]を押します。



パーティションのフォーマットとLILOの設定

Linux用にフォーマットするパーティションを選択します。[フォーマット中に不良ブロックをチェック]をチェックすると、ハードディスクに不良箇所がないかどうかを調べながら、パーティションをフォーマットします。この欄はとくにチェックする必要はないでしょう。選択したら[次]を押します。

次にブートローダLILOのインストール先を設定します。まず、IDEタイプのハードディスクを使う場合は、[リニアモードを使用する(一部のSCSIドライブが必要)]のチェックをはずします。マシンをLinux専用を使う場合と、Windows 9x/MeとLinuxをデュアルブートする場合は、LILOのインストール先に[/dev/hda マスターブートレコード(MBR)]を指定します。Windows NT/2000とLinuxを共存させる場合は、LILOをインストールせずにフロッピーディスクからLinuxを起動します。[ブートディスクを作成]は必ずチェックしておきましょう。



ネットワークとファイアウォールの設定

LASER5 LinuxをDHCPサーバからネットワーク情報を取得する環境で使う場合は、[DHCPを使用して設定]を、DHCPサーバのない環境で使う場合は[DHCPを使用して設定]のチェックをはずして、IPアドレスなどのネットワーク情報を入力します。なお、この画面ではダイヤルアップ接続の設定はできません。

次にファイアウォールの設定です。セキュリティの設定に詳しくないユーザーは、デフォルトのまま[次]を押してください。



使用言語とユーザー設定

LASER5 Linuxで使用する言語を選択します。LASER5 Linuxを日本語環境として使う場合は、デフォルトのまま[次]を押します。次の[タイムゾーンの選択]画面もそのまま[次]を押します。

さて、今度はユーザー設定です。Linuxには大きく分けて、ソフトウェアのインストールなどシステム管理を行う特権ユーザー[root]と、Webブラウザやメールの読み書きなど一般的な作業を行う一般ユーザーの2種類が存在します。

まず、Linux管理者のパスワードとして、[root パスワード]と[確認]に同じものをを入力します。この管理者のログイン名は[root]です。

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](=ログイン名)、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。

ユーザー設定を終えたら[次]を押します。[認証設定]もそのまま[次]を押します。

パッケージグループの選択

パッケージがいくつかのグループに分類されています。どのグループを選択してよいのかわからないユーザーは、最下段の [すべて] を選択して、すべてのパッケージをインストールしておくといでしょう。なお、[すべて] を選択した場合は、ハードディスクを2.3Gバイト程度消費します。

ここでは [すべて] を選択したものととして解説を進めます。



ビデオカードとモニタの設定

多くのビデオカードは自動で認識されます。インストーラが自動認識に失敗した場合は、リストの中から使用するビデオカードを選択します。

次はモニタの設定です。モニタの自動認識に失敗した場合は、モニタのマニュアルを参考にしながら、[Generic]の中からモニタのスペックを超えない解像度の [Generic Monitor] を選択します。なお、液晶モニタを使用する場合は、[Generic Laptop Display Panel] から無理のない解像度のものを選択します。

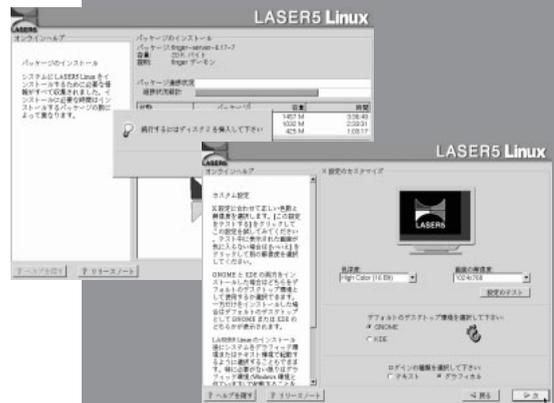


Xの設定とパッケージのインストール

Xを綺麗に表示させるには、[色深度]で [High Color(16Bit)] 以上を選択する必要があります。[画面の解像度]には、モニタに合わせて適当な解像度を指定します。[色深度]と [画面の解像度] を選択したら、[設定のテスト] を押してXの表示テストを行います。GNOMEのデスクトップ画面が表示されれば、Xの設定は成功です。GNOMEのデスクトップ画面が表示されない場合は、[色深度]や [画面の解像度] の数字を低くしてテストしてください。

[デフォルトのデスクトップ環境を選択して下さい]には、ふだん使いたいほうのデスクトップ環境を選択します。[ログインの種類を選択して下さい]で、[テキスト] を選択するとテキストベースのログイン画面が、[グラフィカル] を選択するとグラフィカルなログイン画面が表示されます。ただし、Xの表示テストに成功していないユーザーは [グラフィカル] を選択してはいけません。

[次]を押すとパッケージのインストールが始まります。途中 [続行するにはディスク2を挿入して下さい]と表示されたら、CD-ROMを入れ替えてインストールを続けます。



ブートディスクの作成

ハードディスクからOSを起動できなくなった場合にそなえて、緊急時用の起動ディスクを作成します。空のフロッピーディスクをセットして [次] を押すと、ディスクの作成が始まります。

起動ディスクの作成が終わるとインストール作業は終わりです。フロッピーディスクを抜いて、[終了] を押してください。CD-ROMは自動で排出されます。

お疲れさまでした。

