
Linux の導入と運用

佐藤 和弘^{*}

Unixはマルチユーザ・マルチタスク環境を実現したすぐれたOSである。しかしほんの少し前までは、Unixといえば高価なワークステーション（WS）の世界であり、一般ユーザとは縁遠いものであった。それが最近のコンピュータの急速な性能向上と低価格化とにより、普及クラスのPCでも、Unix互換OSをストレスなしに動作させることが可能になった。とくに90年代初頭に開発が始まったLinuxは、再配布がフリーであること、インストールが容易であることなどから、最近PCユーザの間に急速に普及してきている。当初はIntelとその互換プロセッサ（i386アーキテクチャ）を搭載するAT互換機が動作の対象となっていたが、現在では他の多くのプロセッサ（PPC、Alpha、SPARC、MIPSなど）を搭載するマシンへも移植が行われている。LinuxというUnix互換OSをPCへ導入することにより、コンピュータの基本的な動作原理を学んだり、Unix用に開発された豊富なアプリケーションを利用したり、Cを始めとするプログラミング言語の学習をしたり、サーバとしての運用方法を学んだり、コンピュータ活用の可能性は一気に拡大する。この論文ではAT互換機へのLinuxの導入法と、重要なUnixアプリケーションのいくつかを紹介することにする。

1 Linuxの導入

はじめにLinuxを導入するための基本的な準備について解説する。Linuxには現在多くの配布パッケージ（distribution）があり、初心者をやや混乱させている面がある。

1-1 Linuxとは

LinuxはPCで動作するUnix互換のOSである。そのソースコードはTorvaldsによりまったく独自に書かれたので、AT&Tが所有するUnixのライセンスに抵触することなく自由に配布することができる（オープンソフト）。現在もTorvaldsが中心になり、世界中のハッカーたち（コンピュータの高度な専門家の意）が共同でLinuxの開発（バージョンアップ）を続けている。1996年にはPOSIX仕様（Unix系OSとしての国際規格）を満たし、Unix用に開発された多くのアプリケーションが問題なく動作する。

PCユーザがLinuxをインストールするには、distributionという配布パッケージを利用するのが普通である。代表的なdistributionにはSlackware、RedHat、Caldera、Debian、SuSEなどがある。いずれもLinuxシステムの構築に必要な一連のソフトをパッケージ化し、パッケージを管理するツールを付け、CD-ROMとして配布している（サイトからftp経由でダウンロードすることも可能である）。最近のインストーラは、ハードディスクをフォーマットし、ディレクトリを作成し、そこへ該当するバイナリファイルを展開する。OSの基本部分であるkernelと各種ドライバ、システム運用に必要な各種ユーティリティと豊富なアプリケーション（GPLに基づくフリーソフト）、さらにGUIを提供するX Windowシステムが一気にインストールされるだけでなく、システムの基本的な設定まで完了してくれる。インストールに要する時間は30分～1時間ほどである。

Distributionの種類は多く、日本で入手できるものに限っても10種類近くある。どれを選ぶか迷うところだが、初心者はRedHat系のdistributionがよい。インストールの簡単さ、デバイス自動認識力の高さ、環境設定ツールと豊富なアプリケーションなどに共通の特徴がある。最近XのWindow Maneger（後述）はKDEかGNOMEが標準となり、デスクトップ環境がMacやWindows系OS並に使いやすくなった。日本語環境にもそれ相応の配慮がなされている。Distributionには商用アプリケーションやユーザサポートがついた製品版（1万円前後）もあるが、Linuxを扱った雑誌か書籍を購入すれば、連動するdistributionのftp版がCD-ROMとして添付されているので、インストールにはそれを利用するとよい。解説を読めばインストールの詳細から、実際の運用に必要な知識まで得ることができる。

1-2 ハードウェアの要件

PCのハードウェアがLinuxの導入に必要な基本要件（スペック）を満たしているかどうかには注意が必要である。WindowsやMacマシンのように、OSの動作が保証されたパーツだけで構成されているわけではないので、マザーボードの仕様、CPUの種類、ビデオカードやネットワークカードの種類によっては、Linuxをインストールできなったり、インストールはできてもLinuxの機能をフルに生かしきれない場合がある。しかし最近のLinuxは、各種デバイスへの対応が進み、標準的な構成のPCであればほぼ問題なくインストールできるようになった。極端に古くもなく、かといって最新でもない、いわゆる「枯れた」パーツで構成されたPCへLinuxを導入するのが、一番トラブルが少ないようである。

すでにWindows系OSが搭載されているPCに、Linuxをデュアルインストールする場合は、ハードディスク（HD）にLinux用の領域を別途確保しなければならない（HDに未使用の領域がない場合はかなり面倒なことになる）。できればLinux用に新しいHDを一台増設することである。最近では20Gクラスのドライブでも低価格で提供されているし、読み書きも高速になっている。Linuxにはliloというブートプログラムがあるので、これをインストールすればWindowsとLinuxのデュアルブートが可能になる。

1-3 インストール時の設定

インストール用CDをセットして起動すれば、Linuxのインストールはほぼ自動的に進んでいく。しかし以下に挙げるいくつかの設定をユーザ自身が行なう必要がある。

- 1 パーティションの設定
- 2 マウントするデバイスとマウントポイントの指定
- 3 インストールするLinuxシステムの選択
- 4 LILO（ブートローダ）設定
- 5 X Window関係の設定
- 6 Window Managerの選択
- 7 IPアドレスとホスト名の設定
- 8 サービス設定
- 9 ルートパスワードの入力

次節でこれらの設定の意味を一部明らかにするが、詳細はマニュアル本を参照していただきたい。OSのインストールは納得いくまで何回かやり直すのが普通である。Xを含まない最小構成で約200MB、パッケージをフルインストールすると1GBを越えることがある。プログラム開発ツールを含む標準的な構成で約600～800MBである。

2 Linuxシステムの特徴

LinuxはUnix互換OSであるから、マルチユーザ環境で使用する場合はもちろんのこと、スタンドアローンで使用する場合も、Windows系OSなどとは似ていて異なるところがある。ここではLinuxシステムの特徴を解説する。

2-1 マルチタスク

一台のCPUを用いて、複数のプロセスを（見かけ上）同時に実行するのがマルチタスクという機能であり、その実現はOSにとって非常に難しい課題である。効率のよいマルチタスクを実現しながら、Linux（Unix系OS）が安定したOSであるといわれるのは、OSの核となるkernelという特別なプログラムの働きによる（なお狭義のLinuxとはkernel本体のこと）。

Kernelの第一の仕事は、起動された複数のプロセスに対して、コンピュータ資源を公平に配分することである。そのためにkernelは、各プロセスへの（仮想）メモリの割り当て、保護と開放、各プロセスの実行、実行待機、実行待ちの制御、タイムスライスによるプロセスの切り替えと実行優先順位の制御（動的スケジューリング）、データ転送、割り込み、例外処理

など、非常に多岐にわたる処理を行っている。簡単にいえばkernelはプロセスごとに仮想的なマシン（virtual machine）を提供しているわけである。

さらにシステム保護のためkernelは非常に強い特権を持ったCPUモードで動作する。仮想メモリ割り当てを変更したり、入出力を制御できるのはkernelモードだけである。プロセスは、原則的にハードウェアを直接制御することはできず、必要がある場合はいったんkernelモードに処理を渡す。したがってあるプロセスが他のプロセスに影響を与えたり、kernelそのものの動作に影響を及ぼすことはない。かりにあるプロセスが暴走しても、そのプロセスだけを安全に強制終了できる。またkernelプログラムがロードされる（仮想）メモリ領域は、ユーザプロセスのメモリ領域とは完全に分離され、プロセスからは参照できないように保護されている。

同じマルチタスクを実現しながらWindows95/98がしばしば不安定になるのは、メモリ保護が不完全だったり、DOSとの互換性を保障するためにメモリ使用に制限が設けられているからである。

2-2 ファイルシステム

Unix系OSのファイルシステムは、ルートディレクトリから派生するtree状の階層構造を取っている。DOSやMacのファイルシステムはUnixのファイルシステムを模倣したものである。ルート以外のパーティションや、CD-ROMといったHD以外のデバイス（記憶媒体）も、Unixのファイルシステムではルート以下の適当なディレクトリのように扱われる。これらのデバイスの（異なる）ファイルシステムを利用するときは、マウントポイント（空のディレクトリ）を作って、そこへデバイスをマウントする。マウント後は、もはやファイルシステムに区別なく、自由に参照することができる。マウントポイントは階層のどこにでも自由に設定できる。さらに自分のコンピュータに装備されたデバイスだけではなく、他のコンピュータのデバイスも、ネットワークを経由してマウントすることができる（nfsという仕組み）。コンピュータのリソース（資源）を共有するために、Unix系OSではこのようなファイルシステムを実現している。

2-3 X Window システム

Unix系OSのGUIの基礎となるのがX Window システム（Xと略記）である。Xではデスクトップに複数のwindowを開くことができ、それぞれのwindow内では独立なプロセスが走っている。マルチタスク環境で快適に仕事をしようとするれば、Xは必須である。

XはOS部分とはまったく独立している。MacやWindows系OSでは、GUIはOSの一部であり分離することはできない。Xは画像表示のためのライブラリ（Xlib）を持つが、それは点を打ったり線を引いたり領域を塗りつぶすといった基本的な作図命令だけであり、メニューやボタンなどのGUI要素さえいちいちプログラムを組んで作らなければならない。これでは能

率が悪いので、GUIの要素（widgetという）を簡単に作成するための専用Xツールキットが開発された。

Xlibやツールキットを用いてプログラムされたソフトをXアプリケーションという。Xアプリケーションの中でもっとも重要なのがWindow Manager（WM）である。WMは、複数のwindowの配置、重なり具合、サイズなどを管理し、またwindowのデザイン、メニューやアイコンの操作性などを統一することによって、デスクトップ環境のlook & feelを決定している。WMのデザインや操作感は好みがあるので、どれがよいとは一概に言えないが、最近は一種の「デスクトップ統合環境」であるKDEやGNOMEが事実上の標準となってきた。これらはファイルマネージャ、端末エミュレータ、簡易エディタ、各種ユーティリティがセットになっており、デスクトップはMacやWindows並に操作性がよい。参考に図1にKDEのデスクトップの例を示す。

なおXはクライアント・サーバモデルである。サーバとクライアントは独立なプロセスとして動作しており、両者の間は通信（Xプロトコル）がとりもっている。したがってXプロトコルさえ確立されていれば、サーバとクライアントとが別々のコンピュータの上で動作していてもかまわない。

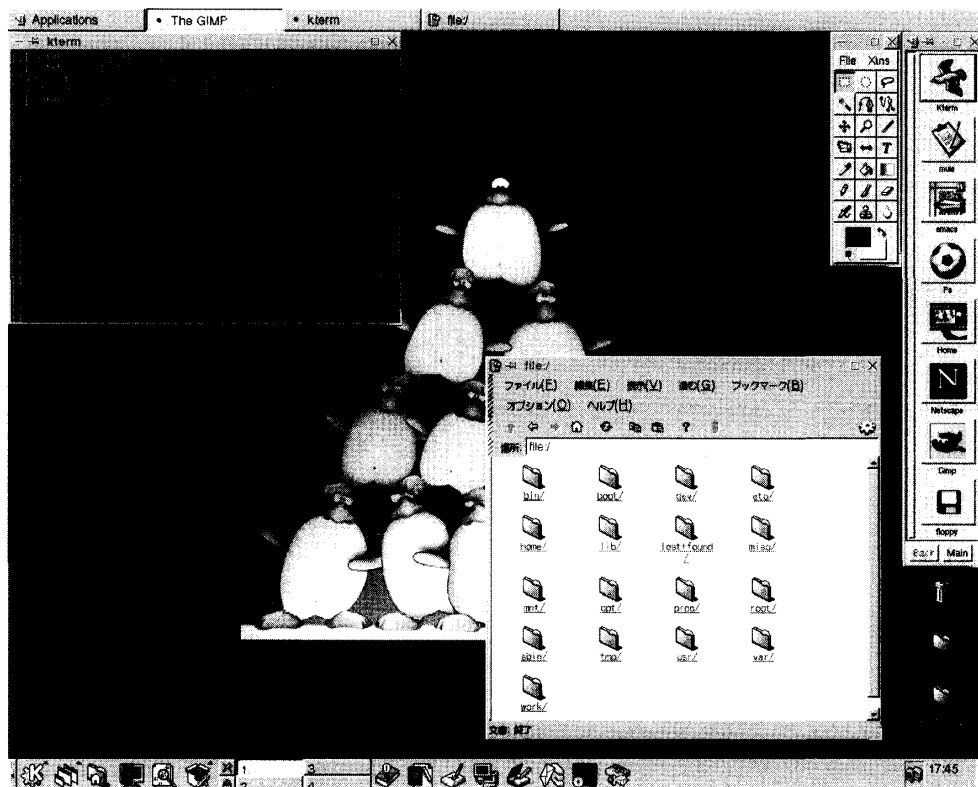


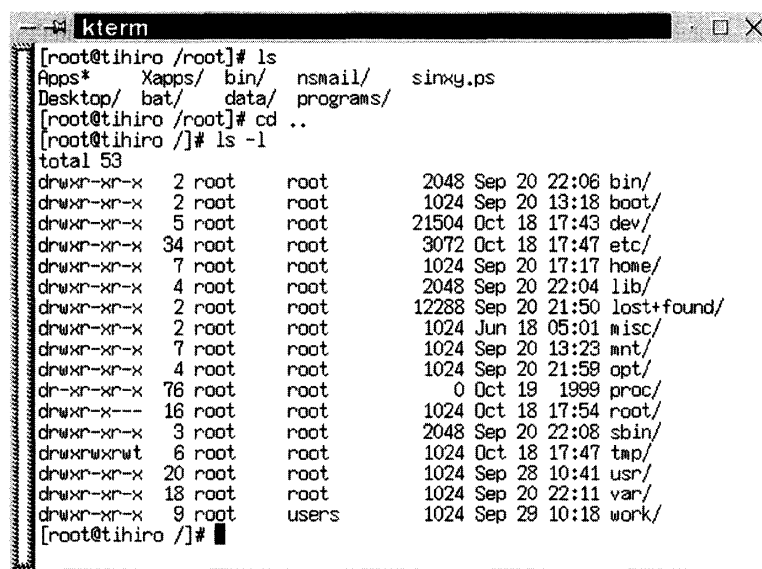
図1 Window Manager KDE

Linuxは好みに応じてWindow Managerを選択することができる。この図はKDEのデスクトップの例である。KDEは端末エミュレータ、ファイルマネージャ、エディタなどがセットになった「統合デスクトップ環境」を実現している。

2-4 Shell

コマンド入力によるコンピュータの操作（CUI）は、今でも Unix 系 OS の基本である。ユーザが入力したコマンドを解釈し、要求された動作を実行してくれるのは shell（これもプログラム）の働きである。Shell は入力コマンドに応じて適切なプログラムを起動し、それを子プロセスとして実行する（shell 自身の機能として実行することもある）。

Shell が解釈できるコマンドは基本的なものだけでも 200 種類以上あり、一つ一つのコマンドの機能は単純である。ただし Unix 系 OS では、これらのコマンドを自在に組み合わせることができるよう設計されており、それによってかなり複雑な処理をも実現できるのである（Small is beautiful は Unix の精神）。X 環境では kterm などの端末エミュレータを起動してコマンド操作を行う（図2 参照）。



```
[root@tihiro /root]# ls
Apps*   Xapps/  bin/    nsmail/  sinxy.ps
Desktop/ bat/    data/   programs/
[root@tihiro /root]# cd ..
[root@tihiro /]# ls -l
total 53
drwxr-xr-x  2 root  root    2048 Sep 20 22:06 bin/
drwxr-xr-x  2 root  root    1024 Sep 20 13:18 boot/
drwxr-xr-x  5 root  root   21504 Oct 18 17:43 dev/
drwxr-xr-x 34 root  root    3072 Oct 18 17:47 etc/
drwxr-xr-x  7 root  root    1024 Sep 20 17:17 home/
drwxr-xr-x  4 root  root    2048 Sep 20 22:04 lib/
drwxr-xr-x  2 root  root   12288 Sep 20 21:50 lost+found/
drwxr-xr-x  2 root  root    1024 Jun 18 05:01 misc/
drwxr-xr-x  7 root  root    1024 Sep 20 13:23 mnt/
drwxr-xr-x  4 root  root    1024 Sep 20 21:59 opt/
dr-xr-xr-x 76 root  root      0 Oct 19 1999 proc/
drwxr-xr-x 16 root  root    1024 Oct 18 17:54 root/
drwxr-xr-x  3 root  root    2048 Sep 20 22:08/sbin/
drwxrwxrwt  6 root  root    1024 Oct 18 17:47 tmp/
drwxr-xr-x 20 root  root    1024 Sep 28 10:41 usr/
drwxr-xr-x 18 root  root    1024 Sep 20 22:11 var/
drwxr-xr-x  9 root  users   1024 Sep 29 10:18 work/
[root@tihiro /]#
```

図2 日本語対応端末エミュレータ kterm

X から shell を起動するには端末エミュレータを使う。Kterm は日本語対応の端末エミュレータである。図は ls コマンドでルートディレクトリを一覧表示したところ。

さらに重要なのは、コマンドをファイルの中に記述しておき、このファイルを shell に読み込ませる（入力リダイレクション、あるいはファイルに実行許可を与える）ことで、コマンドを自動的に実行できることである。for 文や if 文を使った制御構造を用いることもできる（インタープリタ型プログラミング、いわゆる shell スクリプト）。このように Unix 系 OS は、X による GUI と shell による CUI とが、よい意味で「渾然一体」となっている。

2-5 ネットワーク

マルチユーザ環境においては、ユーザがネットワーク経由でホストコンピュータにリモートログインしてくる。そのために Unix 系 OS にはネットワーク機能が実装されている。

ネットワークはいわば「公道」である。この公道の上を、デジタル化されたデータが小さなパケット（小包）となって流れている。パケットは、あて先、送り元、データの種別、パケット番号などを示す「荷札」の部分と、データの本体とからなる。パケットのサイズは決まっており、大きなデータは複数のパケットに分割して発送される。ネットワークに接続されたコンピュータ間で、実際にパケットを交換する際には、パケットの経路などを決めておかなければならないし、送信側と受信側とで通信方式を統一しておかなければならない。通信の物理層とネットワーク層とでこれを定めたのが、イーサネット方式とTCP/IPというプロトコル（通信規約）である。なおTCP/IPはBSD系Unixを中心として開発が進められた。

パケットのあて先はIPアドレスという32bitの数値で決められている。このアドレスさえ指定すれば、世界中どこにでもパケットを送ることができる。しかし数字の羅列ではなじみないので、ホストには具体的な「名前」を付けることになっている。この名前のことを「ホスト名」あるいは「ドメイン名」と呼ぶ。ホスト名とIPアドレスとの対応付け（名前解決という）はそれぞれのLANに設置されたDNSサーバが行う。DNSサーバは自分の管理化にあるホスト群だけでなく、ある範囲をカバーする名前解決用のデータベースを持っている。しかし自己のデータベースでは名前解決ができなかったとき、DNSサーバはDNSネットワークを通して最上位のサーバ（ルートサーバ）に問い合わせを行う。ルートサーバは下位にあるDNSサーバに順次問い合わせを行って、名前解決を試みる。原理的にはこういう手続きを経て、世界中に4000万台以上（99年の概算）あるといわれるホストの中から、ただ一つのホストのIPアドレスを特定できるようになっている。

名前解決が終わると、ルータ（ゲートウェイともいう）は最寄のルータに向けてパケットを送信する。ルータは近傍のルータと交信して経路制御を動的に行っている（時々刻々経路情報をアップデートしている）ので、どこかのルータに障害が発生したり、ある回線のトラフィックが極端に悪くなれば、それが経路情報に動的に反映されて適当な迂回路が選ばれる。これを分散型ネットワークにおける到達可能性という。

2-6 各種サーバ機能

Linuxは各種のサーバソフトを標準で備えており、それらをインストールしたマシンは本格的なネットワークサーバとして運用することができる。しかし家庭内で高々数台のマシンをピア・ツー・ピア接続しているLANの場合は、必要最小限のサーバ機能だけを選んでサービスを開始しておけばよい。すなわちnetwork、inetd、nfsfs、nfs、httpdなどのサービスである。各マシンにはプライベートアドレス（192.168.1.1から始まる）を設定しておく。最近では100BTX対応のネットワークカードやスイッチングハブが安価になり、これらを基盤にしたLANにおいては、HD並の速度でファイルを転送することができる。なお将来インターネットに接続する場合は、ドメイン名（とIPアドレス）を取得した上で、DNSやルータなどの設定が必要になる。

3 代表的なアプリケーション

LinuxにはUnixで開発された多くの強力なアプリケーションが移植されている。また各種のプログラム開発言語が標準で搭載されている。ここではそれらの中でとくに重要なものを簡単に紹介する。

3-1 エディタ

テキスト文書を作成するソフトは一般に「エディタ」と呼ばれる。ワードプロセッサと違い文書を「見たまま」方式では編集できないが、編集に必要な機能が一通り揃った上で、動作が軽快である。Unix系OSの定番エディタといえば、GNUが開発したEmacsとそのX版であるXEmacsがある（図3参照）。

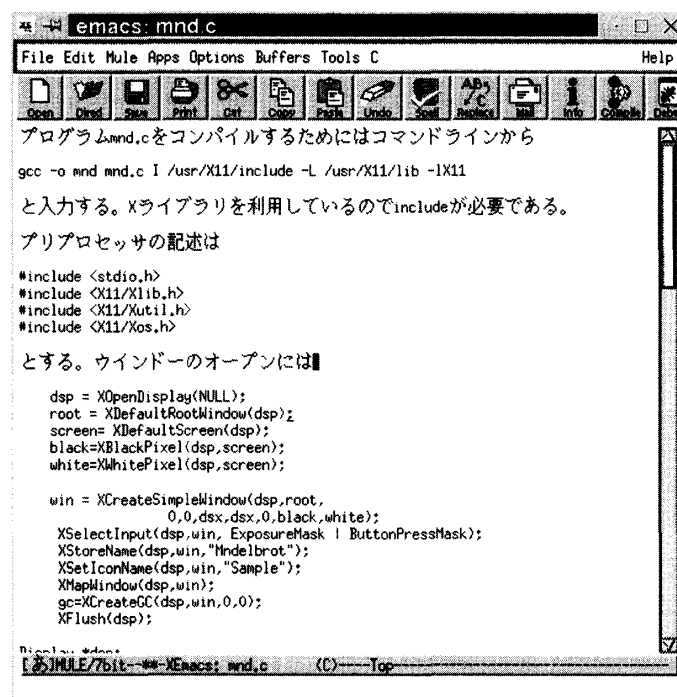


図3 X対応エディタ XEmacs

XEmacsは高機能エディタである。かな漢字変換サーバと連携すれば日本語入力も可能。

Unixは本来ASCIIコード（1バイト言語）を標準としているが、マルチバイト言語への対応（国際化）も進んでいる。Linuxもその恩恵を受けて、今では日本語を使用することが特別ではなくなった。日本語の入力にはクライアント・サーバ方式が採られている。X環境でかな文字の入力処理するのがXIM（X input method）である。普通は日本語入力のためにXIMと変換サーバの両方を起動しておく必要があるが、Emacsは何種類かの変換サーバと直接通信する機能を持つので、単独でも日本語の入力が可能である。よく知られているように日本語の文字コードには三種類あるが、Emacsはそれらを自動判別する機能もついている。

3-2 LaTeX

Linux (Unix) にはLaTeXという文書整形プログラムがあり、高い品位をもった印刷文書を作成することができる。印刷される文書にはおのずと決まったスタイル (文書構造) というものがあるが、LaTeXはこうした文書構造の「雛型」を定めておき、その大枠に沿って文書を整形するプログラムである。整形は「マークアップ」によって行う。マークの多くは論理的なもの (ここはタイトルであるとか章の始まりであるとか脚注であるとかいう「意味」を指定する) である。さらにフォントの指定であるとか、図の挿入であるとか、細かい指示もマークすることができる。なおLaTeXの原典であるTeXは、多彩な数式を含む科学 (とくに数学) 論文を、著者自身の意図通りに印刷する目的で開発されたものである。

LaTeXの文法に従って書かれたソースファイルを文書整形プログラムにかけると中間ファイルが作成される。この中間ファイルを表示用のプログラムで開くと、文書の印刷イメージをプレビューすることができる。具合の悪いところがあればソースをエディタで修正し、再度プレビューする。これを繰り返すことによって校正を進めていく。実際にプリンタで印刷する方法はポストスクリプトの項で述べる。

3-3 グラフィックス

作図 (作画) 用のソフトは、ペイント系とドロー系とに分けられる。Linuxにはペイント系としてXpaint、ドロー系としてTgifというフリーのソフトがついている。スキャナやデジタルカメラから取り込んだ画像の処理を目的とするソフトも数が多い。LinuxにはGIMPというフリーのソフトがある。GIMPはまたさまざまな画像フォーマット (BMP、JPEG、GIF、PS などなど) に対応しているので、フォーマットの変換用としても重宝する。図4にスキャナで取り込んだ画像をGIMPを用いてエンボス処理した例を示す。

関数やデータを与えると、それを自動的にグラフ化してくれるgnuplotというツールもある。二次元グラフ、三次元グラフなどを、簡単なコマンドを入力することによって描くことができる。オプションが豊富に準備されており、グラフの仕上がりを細かく指定することができる。三次元グラフをgnuplotで作成した例を図5に示す。

3-4 ポストスクリプト

文書や画像をプリンタで出力するとき、300～1200dpiの解像度を持つプリンタの性能を生かし、またデバイス依存性をなくすために開発されたのがページ記述言語である。現在はポストスクリプト (PS) が主流である。PSは基本図形 (点、線、円、多角形など) の出力を言語によって記述するので、PS対応プリンタであれば、何を使ってもほとんど同一の仕上がりとなる。PSはアウトラインフォントも使用することができる。アウトラインフォントは文字や記号の輪郭を数式で表現しているので、どのようなサイズの文字を出力する場合でも (コンピュータが計算処理を行うので) ふちが滑らかである。



図4 画像処理ソフトGIMP
 スキャナで取り込んだ画像をエンボス処理したところ（西村豊著「ヤマネ」講談社より）。

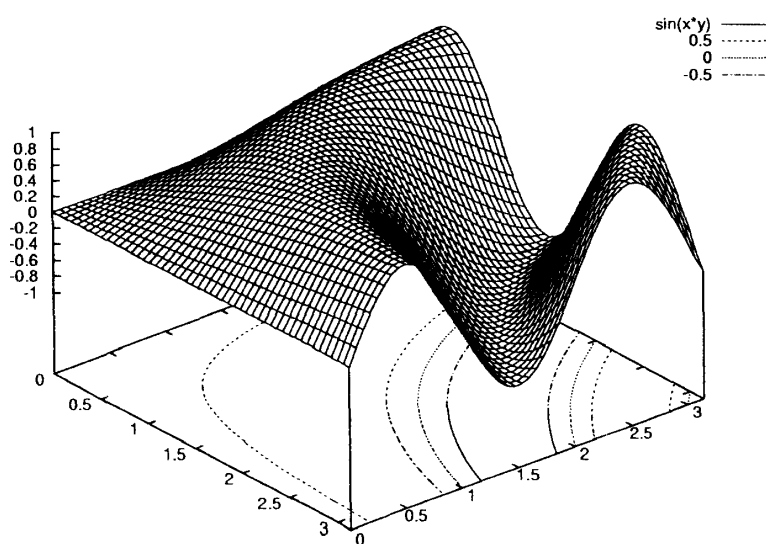


図5 グラフ作図ツールgnuplot
 2変数の関数 $z = \sin(xy)$ を3Dグラフに表示したところ。陰線処理をし、等高線も描いている。

文書あるいは画像をプリンタで出力する場合は、事前にPSファイルに変換しておく必要がある。LaTeX 文書の場合は、先に述べた中間ファイルをさらに変換プログラムにかけてPSファイルを作成する。Tgifやgnuplotで作った図やグラフは、保存の際にPSファイル形式を指定しておく。画像ファイルはGIMPを利用してPSファイルに変換することができる。LaTeXはPS形式で保存した図や画像を、文書の任意の場所に張りつけることができる。

PSファイルはPS対応プリンタ（CPUを内蔵しPS言語を直接解釈できる）で出力するのが普通だが、非PS対応プリンタを用いて印刷するためのエミュレータ（ghostscript）もある。LaTeX文書の非PSレーザープリンタによる印刷例を図6に示す（マークアップ方式で書かれたソースコードも示す）。この文書はGIMPによる画像とgnuplotによるグラフを含んでいる。

図6 a)

<pre> \documentstyle[a4j,12pt,epsbox,twocolumn]{article} \setlength{\textwidth}{160mm} \begin{document} \title{LaTeXの演習} \author{佐藤和弘} \maketitle \section{LaTeXの特徴} \begin{itemize} \item マークアップ言語 \item 抜群な数式の表現力 \item 画像ファイルの取り込み \item PSプリンタによる高品位な印刷 \end{itemize} \section{LaTeX文書の例} \subsection{テキスト} 免疫になじみのない読者のために説明をする。免疫系の 主役は免疫担当細胞、すなわちリンパ球（T細胞、B細胞）とマクロファージ（食細胞）である。これらはいずれも骨髄中の造血幹細胞が分化成熟したものである（以下略）。 </pre>	<pre> \subsection{PS画像の例} スキャナで取り込んだ画像をPSファイルに変換して表示した例です。 \begin{flushright} \psbox[scale=0.5]{pen2.ps} \end{flushright} \subsection{gnuplotによる3Dグラフの例} \begin{center} \psbox[scale=0.5]{sinxy.ps} \end{center} \subsection{数式の例} \begin{equation} f(t)=\sum_{j=1}^m a_j \exp i \lambda_j t \end{equation} \subsection{まとめ} このようにできるわけです。 \end{document} </pre>
---	--

図6 b)

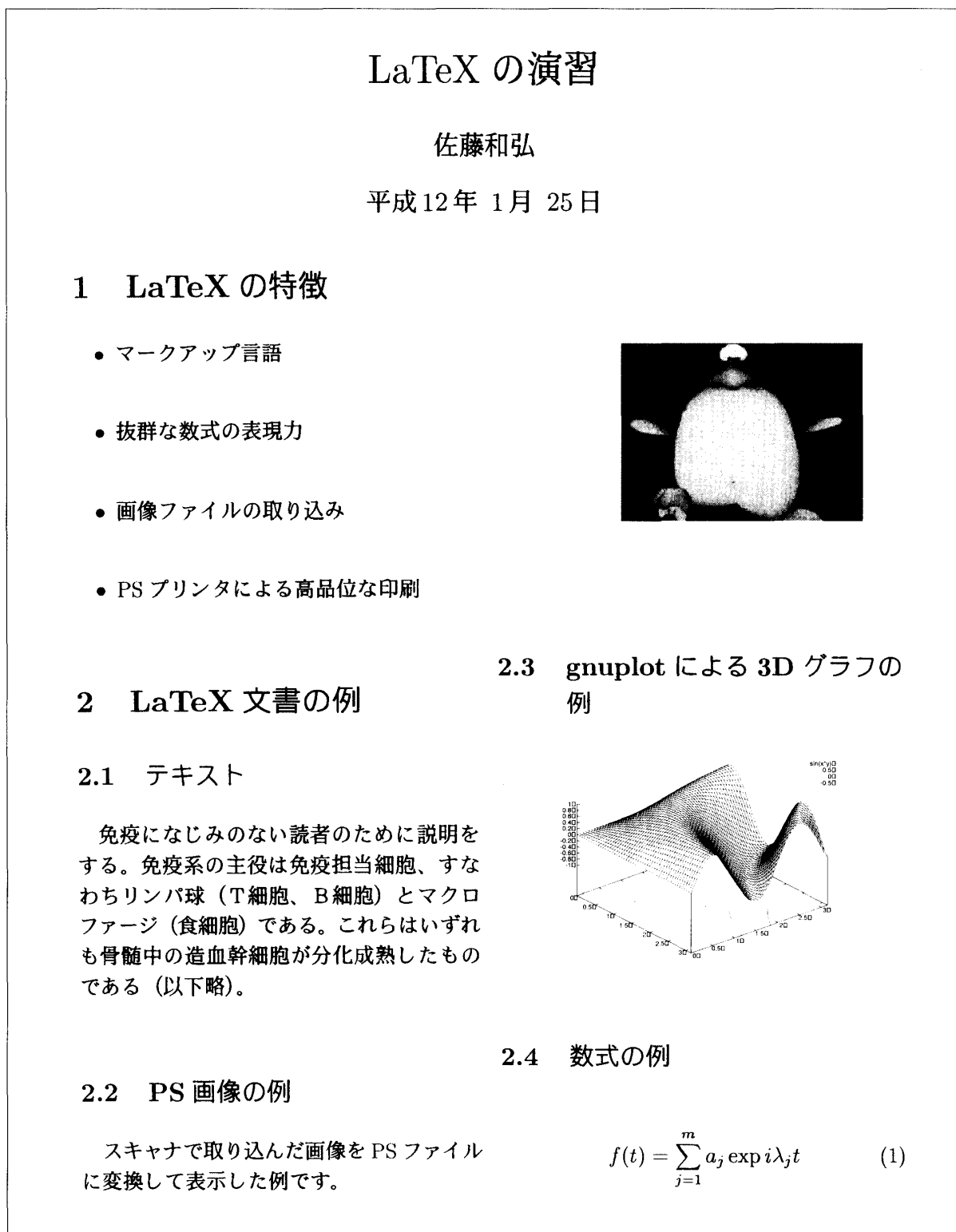


図6 LaTeX文書の例

- a) マークアップで書かれたLaTeXの原稿。
- b) プリンタによる出力。節の番号などは自動で振られる。図は適当な場所に挿入される。LaTeXはもともと記号であふれる数学論文をDTPするために開発された文書整形ソフトであるから、数式の表現力はすばらしい。

3-5 WWW

コンピュータに保存された文書間に、あらかじめ関連づけ（リンク）を行い、相互に情報検索できるシステムのことをハイパーテキストという。WWWはハイパーテキストの一種であるが、従来と違うのはリンク先の文書が同一コンピュータ内の文書に限られないことである。リンク機能に通信機能を連携させることにより、リンク先はネットワークに接続されたホストコンピュータ内の文書にまで拡大できる。リンク先を指定するためには、ホストのアドレス、ファイルのパス、転送プロトコルの三つをセットにしたURLという表記法が用いられる。

WWWが対象とするHTML文書は、論理マークアップ方式で書かれたテキストファイルである。文書の中にタグと呼ばれるマークを記述することにより、文書としての体裁を整える（当然ながらタグの文法はLaTeXのそれとよく似ている）。他のファイルや他のサイトへのリンクは、タグの間にパスやURLを指定することによって行う。タグを処理して、整形された文書をコンピュータ画面に表示するのはブラウザ（閲覧ソフト）の仕事である。最近はワードプロセッサ感覚でHTML文書を作成できるソフトが流行しているが、HTMLは本来論理マークアップ言語であり、文書構成ということをまず念頭におかなければならない。検索ロボットとかサーチエンジンは<H1>タグ（ページの見だしにあたる）などを目印に検索とデータベース化を行うので、文書構成と無関係なところでむやみに文字を大きくしたりすることは避けるべきである。

URLが入力されると、ブラウザは指定されたWWWサーバとの間にネットワーク接続を確立し、httpプロトコルにしたがってHTML文書を転送し、それを画面に表示する。インターネットは今や世界の隅々にまで広がっているから、WWWを使えば、世界中のサーバに分散保存されたコンピュータ資源（resource）を、World Wideに共有することができるわけである。

WWWの技術と仕様は、MITなどが中心になるW3Cという団体によって標準化されている。HTMLは現在version 4.0（1997）が標準である。主要なブラウザはあらゆるコンピュータシステムに移植され、現在フリーで提供されている。Linuxに添付されているのはNetscape社のブラウザである。HTML文書（index.html）のソースファイルと、ブラウザによる閲覧の例を図7に示す。

WWWは世界中のコンピュータに分散保存された情報を、インターネットを通して公開し、皆で共有する仕組みである。知識や資料やデータといった人類共通の財産を、誰もが簡単に検索し、閲覧できる手段を与えた意義は大きい。

3-6 C言語

LinuxにはC/C++はもとより、Fortran、Pascal、Lisp、Java、GUIツールのGtk、スクリプト系のPerl、といったさまざまな言語のコンパイラが、豊富に用意されている。コンピュータの真の面白さを体験するのはプログラム言語に出会ってからという場合が多い。

開発言語に取り組もうとするなら、まずCから始めるのがよい。豊富な入門書と専門書が揃

図7 a)

```
<HTML>

<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=x-sjis">
<TITLE> Satoh Lab Homepage </TITLE>
</HEAD>

<BODY TEXT="#000000" LINK="#0000ff" VLINK="#800080"
BGCOLOR="#c0c0c0"BACKGROUND="/gazou/bgi01.jpg">

<center>
<H1>
<font color="#ff0000">Welcome</font>
</h1>
<h3>
<font color="#0000ff"> 佐藤ゼミ・ホームページ </font> </h3>
<p><center><h4><FONT COLOR="#005500">複雑さの科学</FONT></h4></center>

<IMG SRC="/Joh1/valley.jpg"width="300" height="308" vspace="10"><p>
<A HREF="/Fractals/index.html"> マンデルブロ集合 </a>
</center>

<p><multicol cols="2">
<!--段組二段 netscapeにのみ対応>

<p>私の研究テーマである「複雑さの科学」の紹介をしましょう。複雑さの科学とは、そもそも構成が複雑なシステム
や、振舞いが複雑なシステムに対して、複雑さの由来あるいは
複雑さを生み出すメカニズムを明らかにしようという研究です。複雑さの研究には、従来のような学問領域間の境界が
なく、その地平も広大（無限の果てまで続く！）です。なにし
ろスタートして間もない研究なので、今後どのように発展するか予測がつかない、それだけに多くの可能性と発見の楽
しみに満ちた研究分野です。

</multicol>

<hr>
<center>
<p><A HREF="/cmplx.html"> 複雑さとは何だろうか </A>
<p><A HREF="/lect.html"> 「講義・ゼミ関係」へ進む </A>
<p><A HREF="http://www.nebuta.ac.jp"> 青森公立大学ホームへ進む </A>

<p><hr>
メールアドレス：ksatoh@bb.nebuta.ac.jp <p>
<font color="#008800">
このホームページは研究室のWWWサーバを使って公開しています。OSは<A HREF="/linux.html">Linux</A>です。

</font></center>
</BODY>
</HTML>
```

図7 b)



図7 HTML文書の例

- a) マークアップ方式で書かれた原稿。
- b) Netscapeでブラウズしたところ。表示はブラウザまかせなので、文書イメージはブラウザの種類によって若干変わる。

っている。なおCはUnixと深い関わりがある。Unixは当初アセンブラで記述されていたが、機種依存性が強くて他のCPUアーキテクチャへの移植が困難であった。そのためCという高級言語が開発されUnixはCで書き直されたという経緯がある。

Cはコンパイラ方式の言語である。Cは言語が単純で、データ型が豊富で、演算子が豊富で、ポインタが使える、制御構造が明快で、しかも動作が高速である。仮想メモリ機能を持ちメモリアドレス空間は広大である（Linuxの場合アドレス変換法の制限によりi386で約1GBまで）。X Windowによるグラフィックスが使用できる（Cプログラムの中にXlibで提供される基本的なグラフィックス関数を記述すればそのままXプログラムになる）。動作が高速なので負荷の大きなリアルタイム・グラフィック処理にも対応できる。

Linuxが採用しているのはGNUプロジェクトにより開発されたgccというC/C++コンパイラである。図8に点、線、円、四角という基本図形を描くCプログラムの例（一行目はコンパイルの仕方）とその出力結果を示す。Cは扱えるデータ型が豊富なので、文字列処理などに威

図8 a)

```
// gcc -o mnd mnd.c | /usr/X11/include -L /usr/X11/lib -
IX11

#include <stdio.h>
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/Xos.h>
#define BUFMAX 256

Display *dsp;
Window win,root;
GC gc;
//char achBuf[BUFMAX];
unsigned long pixel;
int screen,black,white;

main(void)
{
    XEvent event;
    dsp = XOpenDisplay(NULL);
    root = XDefaultRootWindow(dsp);
    screen= XDefaultScreen(dsp);
    black=XBlackPixel(dsp,screen);
    white=XWhitePixel(dsp,screen);

    win = XCreateSimpleWindow(dsp,root,
                             0,0,500,500,0,black,white);

    XSelectInput(dsp,win, ExposureMask |
    ButtonPressMask);
    XStoreName(dsp,win,"Sample");
    XSetIconName(dsp,win,"Sample");

    XMapWindow(dsp,win);
    gc=XCreateGC(dsp,win,0,0);

    for(;;)
    {
        XNextEvent(dsp, &event);
        if(event.type == Expose)

            Draws();
        else if(event.type == ButtonPress)
            exit(0);
    }

    Draws()
    {
        int i,j;
        j=10;

        for(i=1;i<=j;i++)
        {
            pixel=0xff0000+20*i;
            XSetForeground(dsp,gc,pixel);
            XDrawLine(dsp,win,gc,10,250,450,250+10*i);
            XDrawPoint(dsp,win,gc,10*i,350);

            XDrawRectangle(dsp,win,gc,200+10*i,20,150,75+10*i);
            //
            XFillRectangle(dsp,win,gc,200+10*i,100+10*i,400,300);
            XDrawArc(dsp,win,gc,10*i,10*i,100,100,0,360*64);
            // XFillArc(dsp,win,gc,10*i,10*i,300,300,0,360*64);
        }
    }
}
```

図8 b)

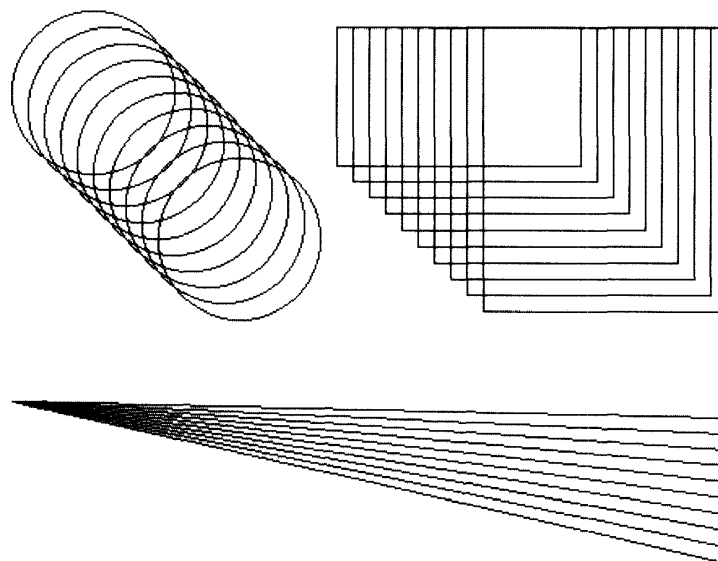


図8 CによるXプログラムの例

- a) プログラムリスト。Xlibを用いて点、線、円、四角を描く。
- b) コンパイルして実行した結果。

力を発揮する。ワードプロセッサやデータベースソフトは通常Cで記述される。表計算ソフトやゲームといったアプリケーションの多くもC (C++) で記述されている。

4 今後の課題と最新の状況

4.1 今後の課題

Linuxはたかだか10年の歴史しかないので、他のUnix系OSと比較するといろいろ改善すべき部分が多い。またdistributionの数が多く、バージョンアップの間隔が短いので、ユーザーをいたずらに混乱させている面がある。これは発展途上にあるシステムには付きもののことで、むしろ今後の発展の可能性を示唆していると言える。

OSの核となるKernelは開発者のTorvaldsが中心になって統括しているが、システム構成に関しては世界中のLinuxコミュニティがそれぞれ独自に開発を進めているので、さまざまな個性を持ったdistributionが今も絶え間なく登場している（これとは対照的にたとえばFreeBSDの開発は特定のメンバーの間で閉じている）。しかもそれぞれが頻繁にバージョンアップを重ねている（ほぼ半年おき！）。このような活力は他のOSの世界では考えられないことであり、まだ当分の間続くであろう。

技術的なことをいうと、Linuxの当面の課題となっているのは、マルチプロセッサへの本格的な対応、アドレス変換法の改善、ファイルシステムの拡張、高負荷状態での安定性強化、ネットワーク機能の最適化などである。OSの機能を最大限引き出すためにはアプリケーション側の対応も必要である。LinuxはUnix系OSとしての歴史が浅く、まだノウハウの部分が足りないのである。むしろ今後運用の実績を積みながら、徐々に洗練されていくことが期待される。

4.2 最新のLinux事情

この論文は駆け足でLinuxの概要を議論してきたが、最後に本文で解説したUnixアプリケーションが一通り揃っていて、すぐに実践的な学習ができるdistributionを紹介する（1999年末現在）。これらは著者が自分のPC（複数台）にインストールして、実際に何ヶ月か使用している。

- 1 Vine Linux 1.0 (ftp版あり)
- 2 Laser5 Linux 6.0 (ftp版あり)
- 3 Kondara MNU Linux 1.0 (製品版のみ)

1は日本語環境への対応を重視している。ただkernelが2.0.36と古い。WMは

WindowMakerが採用されている。2は日本語RedHatLinux6.0として配布されるはずだったものが、諸般の事情でこういう名前になった。kernelは2.2.5、WMはGNOMEが標準だがKDEも選択できる。3は最新のパッケージで構成されていて、kernelは2.2.13、WMの標準はKDEである。ただしインストーラにバグが残っていたり、設定ファイルのいくつかに手を加える必要がある。かな漢字変換サーバはいずれもcannaである。

日本語pLaTeXを除いてもよいのなら、

4 Turbo Linux 4.x (ftp版あり、製品版はpLaTeXも可)

5 Caldera Open Linux 2.3 (ftp版あり)

6 Mandrake Linux 6.1 (ftp版あり)

なども選択肢に挙げられる。4は日本で一番人気があるdistributionで、日本語環境が整っている。Calderaは米国の、Mandrakeはフランスのdistributionだが、日本語パッケージが添付されている。Kernelは順に2.2.9、2.2.10、2.2.13、WMはすべてKDEが標準、ftp版のかな漢字変換サーバはいずれもcanna (4はwnn4も可) である。

なお3と6は、パッケージがPentium (i586以上) に最適化されており、486マシンでは動作しない。また3と5のインストーラはGUIに対応していて、しかも自動化が進んでいるのには驚かされる。インストールだけならWindowsの場合より簡単かも知れない。日本で流通している非PSプリンタへの対応は2と3がよい。

以上総合すると、2のLaser5 Linuxが本論の内容にもっともよく連動しているdistributionのようである。しかしWMの動作が他と比較して「重く」感じられる。著者は現在3のKondara MNUをメインに使用している。まだバージョン1.0のゆえか、部分的にややちぐはぐなところがあり、またインストール後の環境設定が必要であるなど決して初心者向けではないが、内容的にはすぐれている。

終わりにWindows環境の中にLinuxシステムを「同居」させてしまうという、異色のdistributionについて触れておく。LinuxMLDとWinLinux2000がそれである。Linux用に

1) 本論では1999年末の時点としてLinux distributionを紹介しているが、半年たった2000年6月現在、Vine、Laser5、Kondara、Turbo、Calderaの各distributionが(大規模あるいは小規模の)バージョンアップを行っている。Linuxはdistributionの数が多く、バージョンアップの間隔も短いので、タイムラグの大きな活字メディアを通して「最新版」を紹介するのは難しい。なおこれらの最新版(いずれもftp版あり)では、採用されているkernelとXFree86のバージョンに事実上の差がなくなり、WMもGNOMEかKDEが標準となった。デバイスのほとんどが自動認識されるのでインストールは極めて容易である。インストーラもGUIを利用したものが多くなり、前のバージョンに残っていたバグも大部分が取れたため、インストール後の手動による再設定はほとんど必要なくなった。クライアントとして普通に使用する分には(日本語環境も含め)まったく問題ない。とくにVine Linux 2.0は約1年ぶりのバージョンアップであり、日本語環境に関してはTeXへの対応も含めてLaser5同様充実しており、ftp版でも十分推薦できる内容である。

HDのパーティションを用意する必要はなく、今あるWindows領域の中にLinuxシステムをインストールする（その仕組みの詳細は省く）。したがって現行のWindows環境はそのまま継続して使用できるし、Linuxをアンインストールするのも簡単である。WM、ツール、アプリケーション、開発環境なども揃っている。機能に制限があるので本物同様というわけにはいかないが、Linuxシステムの予備的学習のためには役立つし、Windows環境に（Cを始めとする）本格的なプログラム開発環境を導入できる点だけを取っても十分価値がある。割り切って使えばそれなりに評価できるdistributionである。

付記 WWWサーバによる授業支援

たいていのdistributionの場合、Linuxシステムをインストールすると、フリーのWWWサーバソフトであるApache（現在約5割のシェアを持つ）が組み込まれ、httpdサービスが開始されるようになっている。したがって、指定のディレクトリに一連のHTMLファイル（およびリンクされる画像ファイルなど）を保存しておけば、それをホームページとして公開することができる。すでにLANが構築された大学などの環境下では、ネットワークに接続されたコンピュータからこのホームページを自由に閲覧することができる。URLとしてはLinuxマシンのIPアドレスかホスト名を入力する。教員がこのホームページを「授業支援」ために活用することも可能である（学生への授業のアナウンス、HTML文書化した実習ノート、テキスト、教材などの提示）。ftp機能と組み合わせれば、授業支援システムとしては一層充実したものになる。各教員が独自のWWWサーバを立ち上げれば、ホームページのアップデートがこまめにできるし、ネットワーク全体の負荷が分散するというメリットもある。必要であれば大学の公式ホームページから各教員のサーバへリンクを張っておけばよい。最近は大学や高専などで、情報系科目の授業のために開発したテキストを、ホームページで一般公開する教員が増えてきた。

Apacheのデフォルト設定では無制限にアクセスを許可する（世界中から誰でもアクセスできる）ようになっているが、場合によってはアクセスを学内に制限したり、特定のサブドメイン（たとえばある実習室）に制限したほうがよいことがある。そのときは設定ファイル（access.conf）にエディタで手を加えることになる。詳細はネットワーク関係の参考文献を参照されたい。

謝 辞

この研究は、著者がサバティカルで東北大学情報処理教育センターに長期滞在している間に行われた。研究環境を提供してくれた静谷教授と、さまざまなアドバイスを頂いた静谷研究室のメンバーに感謝する。なお文中に記載した会社名および製品名は、それぞれ各社の登録商標または商標である。図中に登場したペンギンの絵は（著者が敬愛する）Stewart Moskowitzの作品である。

(2000年6月13日受理)

参考文献

- キャンベラ-ケリー、アスプレイ：「コンピュータ200年史」 海文堂（1999）
久野靖：「UNIXの基礎概念」 アスキー出版局（1995）
稲垣高宏：「初めての人によくわかるUNIX」 西東社（1997）
Libes & Ressler：「Life with UNIX」 アスキー出版局（1990）
小山・斎藤・佐々木・中込：「UNIX入門」 トッパン（1996）
小山・斎藤・佐々木・中込：「LINUX入門」 トッパン（1996）
小山・斎藤他：「LINUXネットワーク」 トッパン（1998）
安中悟：「入門Linux」 DART（1998）
大津真：「TurboLinuxによるUNIX入門」 日経BP社（1999）
エリック・レイモンド：「伽藍とバザール」 光芒社（1999）
クリフ・ミラー：「Linux革命」 ソフトバンク（1999）
脇英世：「LINUXがWindowsを超える日」 日経BP社（1999）
DiBona, Ockman & Stone 編：「オープンソースソフトウェア」 O'REILLY（1999）
所真理雄：「計算システム入門」 岩波書店（1988）
大木敦夫：「FreeBSDカーネル入門（改）」 アスキー出版局（1998）
時乗洋昭：「X Window System」 技術評論社（1998）
インプレス編集部編：「できるLinuxサーバー構築編」 インプレス（1999）
山口亮：「Linuxネットワーク構築術」 ナツメ社（1998）
久野靖：「UNIXによる計算機科学入門」 丸善（1997）
臼田・葭谷：「パソコンUNIX入門」 森北出版（1998）
佐渡秀治・よしだともこ：「Linux/FreeBSD日本語環境の構築と活用」 ソフトバンク（1997）
野寺隆志：「楽々LATEX 第二版」 共立出版（1994）
和田悟・近藤佐保子：「インターネットコミュニケーション」 培風館（1999）
アंक：「HTMLタグ辞典」 SE（1998）
三田典玄：「入門C言語」 アスキー出版局（1998）
林晴比古：「改訂新C言語入門」 ソフトバンク（1998）
細江亮：「UNIX Xウィンドウプログラミング（Xlib）」 HBJ出版局（1992）
木下凌一：「X-Windowプログラミング」 日刊工業新聞社（1989）