

改訂

Linux

● 基礎から学ぶ

LINUX

Linux

SCC



はじめに

ミニコンの世界では従来からUNIXというOSが一般的に利用されており、IT社会ではUNIXの技術ニーズに依然高いものがある。LinuxはUNIX互換のOSであることから、Linuxを学習することは、直接UNIXの学習にもつながるといえる。

Linuxは、一般的に次のような特徴を持つ。

- ・オープンソースである
- ・無料又は比較的安価に使用できる
- ・安定性が高い

安価で安定性があり、バグが見つかったも対応が早いなどのメリットを持つことから、近年ではUNIXや汎用コンピュータに代わるサーバ用OSとして用いられることが多い。動作するハードウェアも高価なミニコンや汎用コンピュータではなく、パソコンやパソコンをベースとするサーバ機で済む経済性も特徴といえる。

一方、クライアントOSとしては、業務用アプリケーションがまだ充実していないことやビジネスで使用するにはGUIや業務用アプリケーションの安定性に問題が残っていること、またWindowsとのデータ互換などの問題を抱えており、まだ一般に普及しているとはいえない状況である。しかし、海外の国や日本でも国や地方公共団体におけるオープンソース活用の政策が進められており、Windowsに代わるクライアントOSとして利用されるようになる日も近いだろう。

本書はLinuxが扱えるようになるための入門書である

本書では主として「Linuxでサーバを構築し、構築したサーバを運用管理する」というシーンを想定して、これに必要な知識を学習する。さらに、クライアントOSとしての用途も今後は無視できなくなってきていることを考慮して、GUIに関する知識を合わせて学習する。

本書は全10章で構成されている。

第1章から第6章までは、Linuxの基本的な知識について学習する。第7章以降は、第6章までに学習した知識を生かしながら、Linuxのシステム管理の基礎を学習する。

Linuxには、様々な企業や団体からリリースされている種類やバージョンがあるが、本書ではVine Linux 2.6を例に使用して解説を行っている。Linuxの種類が違えば、画面も異なることがあることに留意願いたい。

【本書の記法について】

コマンドの書式について、以下の記法に従う。

```
rm [オプション] ファイル名
①      ②      ③
```

- ① \$ (ダラー) 記号又は# (シャープ、井げた) 記号に続いて入力するコマンドを示す。この例では①のrmコマンドの書式を説明している。コマンド以降の入力項目②、③をパラメータという。
 - ② []内は省略可能もしくは必須入力ではないパラメータであることを示す。この例では必要に応じて、決められているいずれかのオプションを入力すればよいことを示している。オプションが[-abcd]と記述されている場合は、オプションを記述する場合には、-a、-b、-c、-dのいずれかまたは複数を指定することを示す。
 - ③ []で囲まれていない項目は必須入力のパラメータであることを示す。この例では、ファイル名の指定が必須であることを示している。
- ②や③において()で囲まれた項目は、()内の|で区切られたもののうちのいずれかひとつを意味する。(abc|def)はabcまたはdefを表す。
- また、... が続くものは繰り返し指定可能項目を意味する。

本書では、学習の便宜上、主な書式を学習する。コマンドの意味や書式は、manコマンドで調べることができる。コマンドの完全な説明や書式を知りたいときには、各自がmanコマンドで調べて見てほしい。

【man (マン : manualの略) コマンドの書式】 man コマンド名

コマンドの入力サンプルでは、下線部が入力した項目を、下線部のないものがシステムから表示されたものであることを示す。

```
[user1@edc]$ ls -l
合計 332
-rw-r--r--  1 hoge  hoge      282673  1月 29  2007 DSCN0295.JPG
-rw-r--r--  1 hoge  hoge        1714  1月 29  2005 mf_wrapper
-rw-r--r--  1 hoge  hoge         283  1月 16  2003 ntp.conf
```

目次

はじめに

第 1 章 Linux の概要	1
1.1 UNIX と Linux の系譜	2
1.2 UNIX の特徴	5
1.3 パソコンで動く UNIX	9
1.4 Linux を使う	11
第 2 章 プロセス	23
2.1 プロセスの概要	24
2.2 親プロセスと子プロセス	26
2.3 マルチタスクの仕組み	27
2.4 プロセスの管理	28
第 3 章 ファイルシステム	33
3.1 ファイルシステムの概要	34
3.2 階層ディレクトリ構造	35
3.3 ディレクトリの操作 (1)	36
3.4 ドットファイル	38
3.5 ファイル名/ディレクトリ名のルール	39
3.6 ファイルやディレクトリの操作 (2)	40
3.7 パーミッション	45
3.8 シンボリックリンク	50
3.9 ディレクトリパスの指定	51
3.10 マウント	53
3.11 Linux のディレクトリ構成	56
第 4 章 シェル	61
4.1 シェルとは	62
4.2 シェルの種類	63
4.3 シェルの機能	64
4.4 シェルスクリプト	78

第5章 エディタ	101
5.1 エディタの種類	102
5.2 vi エディタ	103
第6章 GUI	117
6.1 X ウィンドウシステム	118
6.2 ウィンドウマネージャ	120
6.3 X ウィンドウシステムを起動する	124
6.4 X ウィンドウシステムの操作	130
第7章 プログラムをインストールする	137
7.1 バイナリ供給とソース供給	138
7.2 ソースコードからのインストール	140
7.3 パッケージ管理システム	145
第8章 ネットワーク	153
8.1 Linux の基本ネットワーク機能	154
8.2 サーバ機能	166
8.3 ネットワークトラブル解決のコツ	174
第9章 スーパーユーザの役割	181
9.1 スーパーユーザとは	182
9.2 スーパーユーザとして作業する	184
9.3 ユーザ管理	185
9.4 システム設定	189
9.5 システムの保守	193
第10章 セキュリティ	201
10.1 何を守るのか	202
10.2 脅威とは	203
10.3 不正行為とその対策	204
10.4 誤操作やシステム障害に対する対策	224
付録	231
付録1 主要コマンド一覧	232
付録2 参考資料	262
付録3 主な記号の読み方	265
索引	267

第 1 章

Linuxの概要

本章では、Linuxを概観する。

最初にUNIXとLinuxの歴史的な系譜を学習する。次に、UNIXとLinuxのOSとしての特徴を学び、最後にとりあえずLinuxを動かしてみる。ここでは、早速いくつかの重要なコマンドが出てくるので、しっかり覚えてもらいたい。

1.1 UNIX と Linux の系譜

UNIXは米AT&Tのベル研究所でケン・トンプソン(K.Thompson)らによって開発されたミニコン向けのOS(Unics)を起源とし、UNIX TSSからUNIX System V R4(SVR4)に至るまで30年を超える歴史を持つ。UNIXは発展の過程で大きく2つの系列(SystemV4系列とBSD系列)に分けられる(図1.1)。

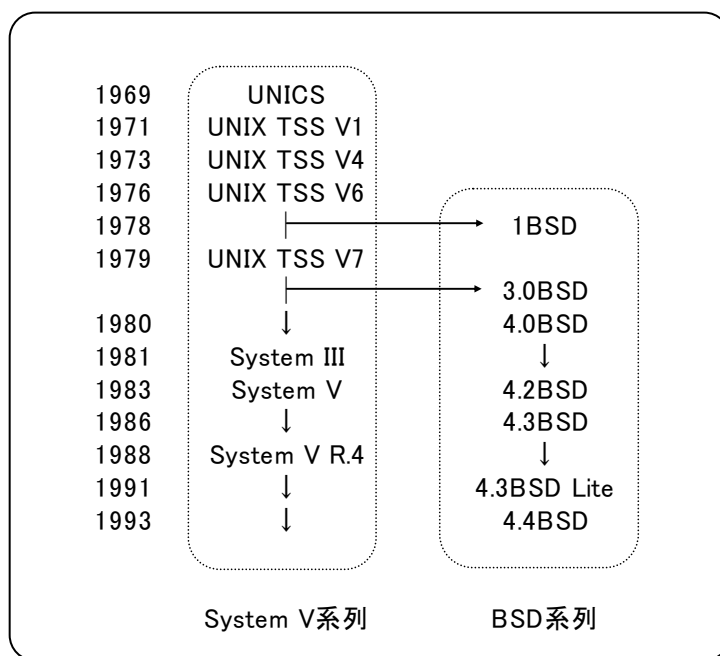


図1.1 UNIXの系図

【UNIX (ユニックス)】

【Linux (リナックス)】

リヌクスなど他にもいくつかの読み方があるが、現在はリナックスと読むのが一般的。Linuxの生みの親であるリーナス・トーバルズ (Linus Torvalds) 氏の名前に由来するといわれている。

1.1.1 System V (システム ファイブ) 系列

System V系列は、最初に開発したAT&Tによる系列である。UNIX TSS V3まではアセンブラで記述されていたが、V4でC言語の開発で知られるリッチ(D.Ritchie)により大部分のコードがC言語で書き換えられ移植性が向上した。また最初に公開されたV6では、ゆるやかなライセンス条件でソースコードを公開していたためBSDをはじめとする多くの派生UNIXが誕生した。V7後、UNIXを商品としてまとめたSystem IIIがリリースされ、System Vに至る。System Vの最後のバージョンであるリリース4はSVR4と呼ばれデファクトスタンダードとなった。現在ではSVR4そのものは使われていないが、商用UNIXの大半(Unixware、Solaris、HP-UX、AIXなど)はSVR4を基にしている。

1.1.2 BSD (Berkeley Software Distribution : ビー エス ディー)系列

BSD系列は、公開されていたUNIX TSSのソースコードを元に米カルフォルニア大学バークレイ校(UCB)で機能拡張が行われたUNIXである。BSDは「バークレイによって配布」されたUNIXという意味である。BSDはその高機能さから多くの研究機関で用いられた。インターネットの元になったARPANETはBSDによって構築されており、TCP/IPを最初に実装したのは4.2BSDである。商用UNIXとしてはサンマイクロシステムズ(Sun Microsystems)のSunOSやSonyのNewsOSなどがBSDを元にしていた。AT&TがUNIXを商用化していく過程でSystem Vがデファクトスタンダードになった結果、今日では大手商用UNIXの中にBSD直系のものは残っていないが、BSD由来のコードは現在の*BSDなどのPC-UNIXに受け継がれているほか、System V系のUNIXにも多く取り込まれている。

【Solaris (ソラリス)】: サンマイクロ社のミニコン用UNIXの商品名

【ARPANET (アーパネット)】

1969年に米国防総省の高等研究計画局(ARPA)が導入したコンピュータネットワーク

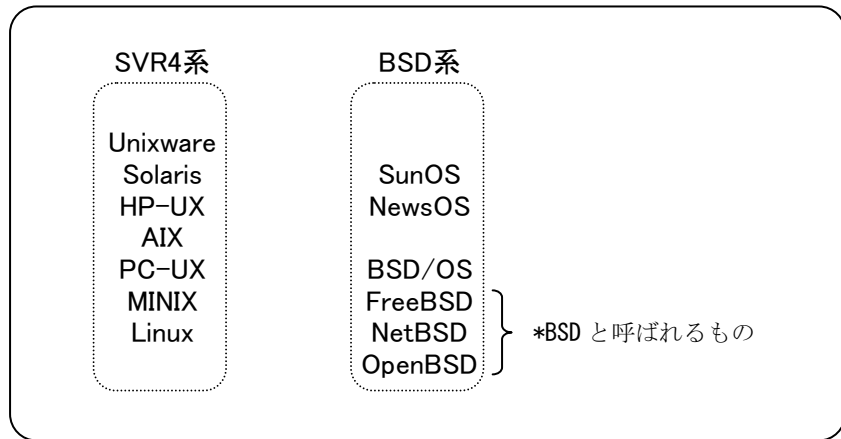


図1.2 主要UNIX

1.2 UNIXの特徴

UNIXには、マルチタスクといったような技術的な特徴の他に、オープンアーキテクチャというベンダーやメーカー固有の仕組みから脱却するための重要な開発思想が含まれている。Linuxにもこの思想は受け継がれており、さらに現在ではオープンソースといった重要な概念が浸透し始めている。

WindowsとUNIXでは表1.1のような違いがあり、これらの内容について学習していく。

表1.1 WindowsとUNIX

	UNIX	Windows
アーキテクチャ	オープン	クローズ(独自)
ソースコード	公開	非公開
ディレクトリ構造	単一のツリー構造	各ドライブごとのツリー構造
コマンドインタプリタ	sh、csh、bash など	command.com、cmd.exe

1.2.1 オープンアーキテクチャ

UNIXの最大の特徴は、オープンアーキテクチャである。オープンアーキテクチャとは、UNIXの仕様がすべて公開されているという意味である。ソースコードが公開されている(オープンソース)ため誰でもソースコードから、仕様を読み取ることができる(各社からリリースされている現在の商用UNIXのソースは非公開である)。ソースコードが公開されているため、様々なハードウェアへの移植性に優れ、同時に必要な機能の取捨選択(OSの再構成)が容易である。

オープンソースであることは、世界中の技術者の知恵を結集できるため次のようなメリットがあるといわれている。

- 開発が速い
- 信頼性が高い
- 安全性が高い

1.2.2 標準化

オープンソースであることは、様々な改造版が誕生することにもつながるが、それによる混乱を防ぐためUNIXの仕様の標準化が行われている。UNIXの標準化は2つの側面から行われている。

ひとつはUNIXのシステムとしての標準仕様を定めることで、これはThe Open Group(TOG)という団体によって行われている。TOGではUNIXの名称を登録商標として管理しており、UNIXという名称を使用するにはTOGの仕様を満たしていることの認定を受けなければならない。

もうひとつ、ソフトウェアの移植性や相互運用のため標準を定めたPOSIX(Portable Operating System Interface)という国際標準規格がある。POSIXではC言語のライブラリAPI(Application Programming Interface)とシェルやユーティリティなどの仕様を定めている。POSIXはインタフェース仕様のみを定めているのでPOSIXに準拠することが即UNIXではない。例えばLinuxはPOSIXに準拠しているが、TOGの認定を受けていないので厳密な意味ではUNIX「互換」OSである。

1.2.3 マルチユーザ／マルチタスク

利用者としてみた場合のUNIXの特徴は、マルチユーザ／マルチタスクなシステムということである。マルチユーザとは、複数のユーザが互いに独立して作業できること、マルチタスクとは複数の処理が同時並行で処理されることを指す。Windows 2000やWindows XPでもマルチユーザ・マルチタスクが実現されているが、WindowsではWindows NTが登場するまでユーザの概念がなく、従ってマルチユーザの概念がなかったのである。

またUNIXではローカルコンソールとネットワーク経由のいずれからでもすべての機能が利用できるが、Windowsは2000やXPにおいてもすべての機能が利用できるのはローカルコンソールからだけでネットワーク経由のユーザができることは限られている。

【オープンソース】

オープンソースは著作権フリーという意味ではない。オープンソースとして公開されているソフトウェアでも、著作権などの権利は原作者に存在し、改変や再配布などは定められたライセンス契約の下に公開される。

1.2.4 単純な階層ディレクトリ構造のファイルシステム

UNIXのファイルシステムは非常に単純な階層ディレクトリ構造をしている。階層ディレクトリ構造というと、今日ではDOSやWindowsでよく知られているが、これは元々UNIXで採用されたものである。UNIXにおいては外部入出力装置もこの階層ディレクトリ構造の中に位置付けられ、ファイルとして扱うことができる。これによって入出力先を意識する必要がなくなり、プログラミングが単純化されるメリットを生んでいる。

UNIXやLinuxにおけるディレクトリとファイルシステムについては、第3章で学習する。

1.2.5 高機能なコマンドインタプリタ

高機能なコマンドインタプリタを持っていることもUNIXの大きな特徴である。OSを動かすための1つの指令をコマンドという。一般的なコマンド入力の方法は、コンソールのキーボードから1行ずつ入力してやる方法である。

コマンドインタプリタとは、入力されたコマンドを解釈し、OSのコア部 (UNIXではカーネルという) に動作指示を与える役割を果たすプログラムである。DOSやWindowsにおけるコマンドインタプリタは、command.comやcmd.exeといったプログラムによって実現されているが、UNIXのコマンドインタプリタはシェル (貝殻という意味) と呼ばれ、これらより高機能である。コマンドインタプリタの詳細は、第4章で学習する。

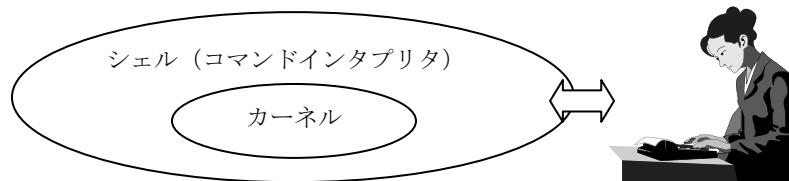


図1.3 Linuxの構造

1.2.6 インターネットとの高い親和性

インターネットの起源となったARPANETがBSD版UNIXで開発されたことは既に述べたが、その後もインターネットの発展とUNIXの歴史は密接に関連している。

現在では、インターネットで利用されているサーバの多くは、Linuxを含めたUNIX系列のOSが利用されている。中でもインターネットで使用されるサーバの1つであるウェブサーバでは、全世界の半数以上がApache(アパッチ)というサーバソフトを使っているとされるが、ここで使用されるOSはUNIX系列と思って差し支えないだろう。その他のサーバソフトについても、インターネット用サーバとして実績が豊富なものが多く、イントラネットへの流用なども行われており、いわゆる定番ソフトと言われるものも多い。

1.2.7 セキュリティと堅牢性

UNIXは30年近くもの間、ソースコードが数多くの技術者の目に晒されてきたことによって数々の不具合が修正されて非常に安定性の高いシステムとなっている。もちろん動作の不具合が見つければ、ソースコードを確認することによって解決策を見つけた技術者の誰もが修正コードを発表することができる。このことはシステムの安定性とともセキュリティの面でも非常に有効である。

1.3 パソコンで動く UNIX

パソコンで動くUNIX (PC-UNIX)としてはLinuxが広く知られている。PC-UNIXの代名詞と呼べるほどに普及してきているLinuxだが、「Linux」とは、本来kernel (カーネル: OSの中核)のみを指す名称である。いわゆるLinuxの配布物は、このkernelにGNU(GNU's Not UNIX:グヌー)と呼ばれるUNIX互換のコマンド群などを組み合わせることでUNIX互換のシステムを形作って、様々な企業や組織から配布される。このときのLinuxの配布物をディストリビューションという。

1.3.1 Linuxの生い立ち

Linuxは、大学生だったリーナス・トーバルズ(Linus Torvalds)が学内でUNIXの教育用に使われていたMINIXというパソコン用UNIXに触発されて、UNIXとのソースコード互換を目標に開発したパソコン用OSである。リーナス独自で開発したものであったが、その成果(バイナリコード、ソースコード)をインターネットで公開したことにより、世界各地の協力者を得て開発が続けられている。

パソコンで動くUNIXには、今日ではLinuxの他にBSD版UNIXの流れを引くFreeBSDなどの*BSDがある。これらの多くは世界中のボランティア技術者によって開発され無償で公開されていることから、近年のインターネットの普及にともなって大きな広がりを見せつつある。

1.3.2 ディストリビューション

Linuxのディストリビューションは、組み合わせるコマンドや独自に開発した管理ツールなどによって差別化され、様々な企業や組織からリリース(配布)されている。ディストリビューションは、大きくRedhat系、Slackware系、Debian系に分類することができる。代表的なディストリビューションを表1.2に示す。

表1.2 Linuxの系列とディストリビューション

Redhat系	Slackware系	debian系
Redhat	Slackware	debian/GNU
Fedora		
Turbo		
Laser5	plamo	Omoikane/GNU
miracle		
Vine		

※点線より下は日本生まれのディストリビューションである。

【GPL (ジーピーエル)】

GNU 一般公有使用許諾書(GNU GPL : GNU General Public Lisenca)とは GNU ソフトウェアに定められた使用ライセンス条件であり、Linux (Kernel) もこの GPL に基づいて公開されている。このライセンスは一般的なソフトウェアライセンスと大きく異なり、ユーザの権利が最大限認められている。GPL においてはユーザは利用に関して制限を受けず、ソースコードの入手について保証される。その一方で改変物もオリジナルと同じ条件で公開することが義務づけられる。

1.4 Linux を使う

Linuxを操作する方法にはいくつかあるが、スタンドアロン環境で動作しているLinuxパソコンを想定して、操作してみよう。パソコンの電源スイッチを入れ、Linuxをインストールしたパソコンを起動しよう。

1.4.1 ログイン

Linuxを操作するためには、正当な利用者であることを確認するためのログインという操作をしなければならない。Windowsでは、同様の操作をログオンと呼んでいる。

図1.4は、パソコンの電源投入直後など、ログイン待ち状態の画面である。ログインを促すプロンプト(ログインプロンプト)が表示され、入力待ち状態になっている。プロンプトとは、操作者に入力を促すための短いメッセージのことである。図1.4はCUIベースのログイン画面であるが、GUI環境が構築されている場合には、GUIからログイン可能なようにも設定できる。LinuxのGUI機能については、第6章で学習する。

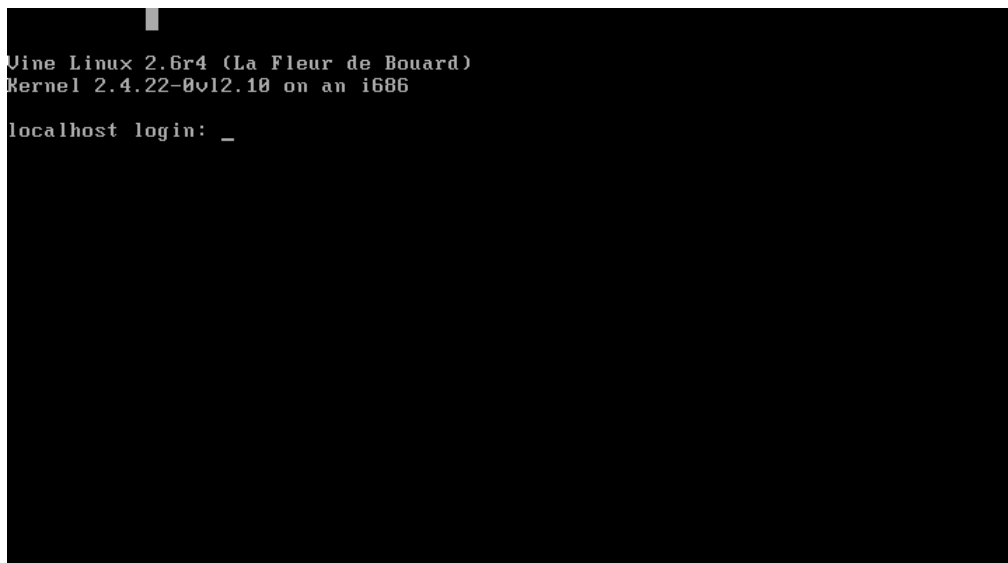


図1.4 ログイン画面

UNIXはマルチユーザOSなので、利用者が誰なのかを区別する必要がある。このための操作として、ユーザ名を入力して利用者を明らかにするのがログインである。また、この際パスワードを確認することによって第三者による不正利用を防いでいる。すなわち、Linuxパソコンを利用するには、ユーザ名とパスワードを事前に登録しておく必要がある。このような考え方は、Windows 2000やWindows XPでも同じである。

以降では、事前に登録されているユーザ“user1”を使用して操作を行うことにする。

画面の「login:」プロンプトに続けてユーザ名をタイプしEnterキーで入力する。「password:」プロンプトが表示されたらパスワードを入力する。パスワードは入力しても表示されない。めでたくユーザ名とパスワードが正当なものと確認されると、図1.5のように表示されて次のキー入力待ちとなる。

Linuxのディストリビューションや個別の設定によって、表示内容が変わることがあるが、最終的には図1.5のような入力待ちになる。

```
Umine Linux 2.6r4 (La Fleur de Bouard)
Kernel 2.4.22-0v12.10 on an i686

localhost login: user1
Password:
[user1@localhost user1]$ _
```

図1.5 ログイン直後の画面

ログイン後のコマンド入力待ちを示すプロンプトを、シェルプロンプト(以下、単にプロンプトと表記した場合はシェルプロンプトを指す)と呼ぶ。これ以降は、コマンド(指令、命令)を入力することでLinuxを使っていくことになる。

【root (ルート) という名のユーザ】

Linux には必ずユーザ ID が 0 (ゼロ) のユーザが存在する。通常このユーザ ID 0 のユーザは root という名で呼ばれる。以下、root といったときにはユーザ ID 0 のユーザを指すものとする。この root というユーザは極めて特別なユーザである。マルチユーザ OS において一般のユーザは、特別に許可されない限り、他のユーザの所有しているファイルを操作することや、他のユーザが実行中のプログラム(これをプロセスといい次章で学習する)を操作することはできない。しかし、root にはこれらの制約が一切なく、いかなる操作も可能なのである。また、新規ユーザを作成することができるのは root のみである。このように、root は一般的なユーザではなく、Linux システムの最高管理者といえることから、そのパスワードは厳重に管理する必要がある。

1.4.2 ログアウトと電源の切断

Linuxの利用を終了するためにはログアウトの操作を行う。同じ操作ではあるが、Windowsではログオフという。

ログアウトするには、logoutコマンドを入力する。

```
$ logout
```

ログアウトすると、コンソールではログインプロンプトに戻る。

ログアウトと同時に、Linuxそのものを終了しパソコンの電源を落とす場合は、次のようにshutdownコマンドを使用する。

```
$ shutdown -h now
```

これによりLinuxは終了し、パソコンの電源が切れる状態になる。

shutdownコマンドは、管理者以外のユーザが勝手にパソコン電源を切れないようにrootユーザのみが使用可能なコマンドである。なぜなら、LinuxはマルチユーザOSのため、他にログイン中のユーザがいるかもしれないからである。従って、rootユーザ以外でログインした場合には、このコマンドは使用できない。

【logout (ログアウト) の書式】 logout

類似のコマンドに exit がある。

【exit (イグジット：出口、抜ける) の書式】 exit

【shutdown (シャットダウン) の書式】 shutdown [オプション] 時間

(オプション) -h シャットダウン後システムを停止する

 -r シャットダウン後システムをリポートする

(時間) now すぐにシャットダウンする

 now の他に hh:mm 形式などで時間が指定できる

1.4.3 私は誰？

さて、改めてuser1でログインしてみよう。ここでは、Linuxの特徴として説明したマルチユーザについて実際に見てみることにする。

図1.5のプロンプト行の先頭に表示される「user1@」のuser1はログインしたユーザ名を表している。このように、どのユーザ名でログインしているかはプロンプトに表示されるようになっていることが多いが、プロンプトの表示形式は変更可能なので、確認できない場合もある。確実にログイン中のユーザ名を知るためには、whoamiコマンドを使えばよい。

```
$ whoami
user1
```

文字どおり「私は誰？」というわけである。

似たような名前のコマンドにwhoコマンドというものがある。これは、その時点でそのLinuxパソコンを使用しているユーザの一覧を見るコマンドである。

```
$ who
user1      tty1      Jun 11 14:12
user2      tty2      Jun 11 14:30
```

表示結果を見ると、ユーザ名とともにどこからいつログインしたかが表示され、今このLinuxパソコンには私(user1)の他にuser2という別のユーザがログインしていることがわかる。同時に複数のユーザがログインできるのはマルチユーザ・マルチタスクOSならではのことで、このuser2はどこからログインしているのだろうか？

【whoami（フーアムアイ：私は誰？）の書式】 whoami

【who（フー：誰か）の書式】 who

who コマンドには、いくつかのオプションがあるがここでは説明しない

1.4.4 リモートログイン

パソコン自体に接続されたキーボードとディスプレイのことをローカルコンソールと呼ぶ。物理的にローカルコンソールはひとつしかないので、これを使えるのはひとりだけである。Windowsの場合はローカルコンソールを使う以外に方法がないので、同時に複数のユーザが操作することは不可能である。

これに対してマルチユーザ・マルチタスクOSであるLinux(UNIX全般に同じ)では、ログインする方法が複数用意されている。

- ① ローカルコンソール
- ② リモートコンソール (telnetなど)
- ③ 仮想コンソール

この内、②はネットワーク経由で利用する方法で、telnetというアプリケーションプログラムを使ってログインする方法が代表的である。WindowsでもLinuxでも、ログインしたいLinuxパソコンと同じネットワークに接続した他のパソコンがあれば、②の方法でログインすることができる。

telnetを使ったリモートログインは、Windowsパソコンではコマンドプロンプトから、また、Linuxならプロンプトから、次のように入力(ログイン先のLinuxパソコンのIPアドレスが192.168.152.131の場合)する。

```
> telnet 192.168.152.131
```

これによって、図1.6のようにログインプロンプトが表示される。ローカルコンソールの場合と同じく、ログインするためには、ユーザ名とパスワードを入力すればよい。

【telnet (テルネット) の書式】 telnet ホスト

ホスト ホスト名または IP アドレス, ホスト名の場合は名前解決ができること。

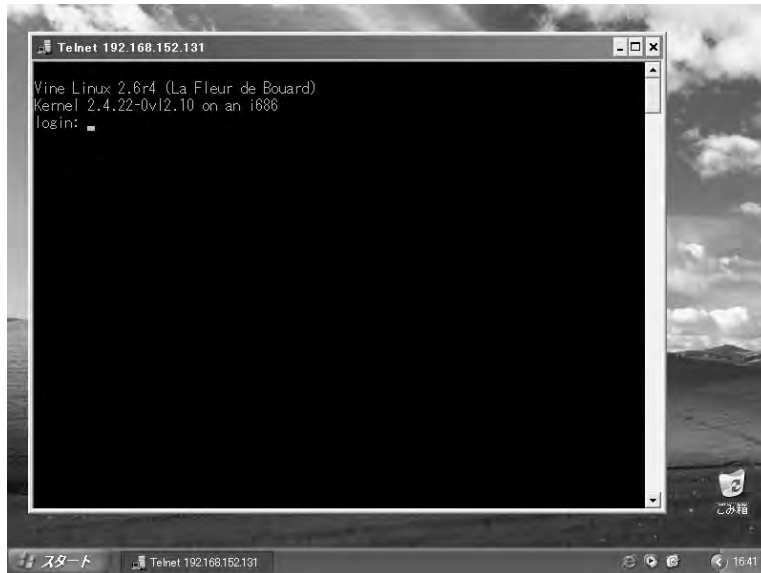


図1.6 windowsパソコンからのログイン画面

telnetを使ったリモートログインでは、ログイン先のホストでTELNETサーバが動作していなければならない。また、セキュリティ確保の観点から、特定のユーザやパソコンからしかログインを許可しない、あるいはリモートログインそのものを許可しないといった設定が行われていることが多い。Linuxのセキュリティに関しては、第10章で学習する。

1.4.5 仮想コンソール

ローカルコンソールだけでも同時に複数のユーザがログインすることができる。

ローカルコンソールのキーボードで[Alt]キーとファンクションキー(F1、F2、...)を同時に押すと、新たなログインプロンプト画面が表示される。これがログイン方法の3番目にあげた仮想コンソールである。一般的に、6~8程度の仮想コンソールを使うことができる。電源投入直後、つまり一般的にローカルコンソールと思っているものはF1に対応する仮想コンソールである。

仮想コンソールを使う場合、同じキーボードとディスプレイを共有するので、厳密には同時に複数のユーザが利用できるわけではないが、ある仮想コンソールで時間のかかる処理を実行中に、別の仮想コンソールに切り替えて別の処理を行う、といった使い方ができる。

WindowsXPのユーザ切り替え機能と類似の機能と見ることができ、WindowsXPでは1ユーザ1ログオンに限定されるのに対して、Linuxの仮想コンソールでは複数のコンソールに同じユーザがログインすることも可能である点などが異なる。

1.4.6 マルチタスクを見る

Linuxの特徴の一つであるマルチタスクについて、複数の処理が同時並行で実行されている様子を紹介する。

Linuxでは、実行中のプログラムをプロセスと呼ぶ。プロセスは、タスクと同じと考えて良い。Linuxでは、同時に数多くのプログラムが実行できるようになっているので、マルチタスクと呼ばれる。

プロセスが動いている状態をリアルタイムに表示させるコマンドとしてtopコマンドがある。

```
$ top
```

【top (トップ: 上位) の書式】 top [オプション]

様々なオプションがあるが省略する。qキーの入力又はctrl+zで表示を終了する。

図1.7に、topコマンドで表示させたプロセスのサンプルを示す。

```
午後05時13分 稼働 1日間, 15時51分, 2 ユーザ, 負荷平均率: 0.00, 0.00, 0.00
プロセス: 合計 46, 待機中 44, 実行中 1, ゾンビ 0, 停止中 1
CPU状態 : ユーザ 0.1%, システム 0.3%, nice 0.0%, 待機99.4%
メモリ   : 合計 126288K, 113924K使用, 12364K空き, 0K共有, 32892Kバッファ
スワップ : 合計 257032K, 3768K使用, 253264K空き 48424Kキャッシュ
```

PID	ユーザ	優先	NI	サイズ	常驻	共有	状態	LIB	%CPU	%MEM	時間	コマンド
13	root	12	0	0	0	0	SW	0	0.1	0.0	0:43	kjournald
9638	user1	15	0	1200	1200	972	R	0	0.1	0.9	0:01	top
1	root	9	0	520	484	456	S	0	0.0	0.3	0:06	init
2	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	keventd
3	root	19	19	0	0	0	SWN	0	0.0	0.0	0:00	ksoftirqd_C
4	root	9	0	0	0	0	SW	0	0.0	0.0	0:10	kswapd
5	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	bdflush
6	root	9	0	0	0	0	SW	0	0.0	0.0	0:13	kupdated
7	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	khubb
137	root	-1	-20	0	0	0	SW<	0	0.0	0.0	0:00	mdrecoveryd
144	root	9	0	0	0	0	SW	0	0.0	0.0	0:00	kjournald
416	root	9	0	828	816	764	S	0	0.0	0.6	0:00	dhcpcd
471	root	9	0	604	600	504	S	0	0.0	0.4	0:01	syslogd
476	root	9	0	1300	872	460	S	0	0.0	0.6	0:00	klogd
488	rpc	9	0	572	480	480	S	0	0.0	0.3	0:00	portmap
510	rpcuser	9	0	772	660	660	S	0	0.0	0.5	0:00	rpc.statd
558	root	9	0	624	532	512	S	0	0.0	0.4	0:00	automount

図1.7 topコマンドを使ったプロセスの表示例

図1.7の表示内容は時々刻々と変わっていく。右端のコマンドというタイトルの下に表示されているのがこのプロセスを生成したコマンドである。この1行1行がプロセスを表している。これだけたくさんプロセスが同時に実行されているのである。

参考として、WindowsXPのプロセスを表示をしたものを図1.8に示す。



図1.8 WindowsXPのプロセスの表示例

第1章で学習したコマンド

logout	ログアウトする
exit	ログアウトする
shutdown	シャットダウンする
whoami	自分のアカウント名を確認する
who	ログイン中のユーザを確認する
telnet	リモートログインする
top	実行中のプロセスの様子を表示する

練習問題

練習 1.1 UNIX の 2 大系列である System V と BSD についての説明文を次のア～ウの中からそれぞれ選びなさい。

- ア UNIX を産み出した AT&T による。リリース 4 は SVR4 と呼ばれデファクトスタンダードとなった。
- イ リーナス・トーバルズ(Linus Torvalds)によるパソコン用 OS である。
- ウ カルフォルニア大学バークレイ校による。インターネットの基盤となっている TCP/IP を最初に実装したのは 4.2BSD である。

練習 1.2 次の文章の空欄①, ②に入る言葉を答えなさい。

UNIX の最大の特徴は、[①]である。その仕様がすべて公開されているという意味である。[②]が公開されているため誰でも入手でき、その結果その仕様は誰でも [②]から読み取ることができるのである。

練習 1.3 次の各文の説明にあてはまる言葉を語群から答えなさい。

- ①UNIX のシステムとしての標準仕様を定め、名称を登録商標として管理する団体。
- ②UNIX ソフトウェアの移植性や相互運用のための国際標準規格。

【語群】 ア kernel イ POSIX ウ GNU エ The Open Group

練習 1.4 次の各文を UNIX の特徴を表す文として完成させなさい。

- (1) [①]ユーザ・[①]タスクなシステムである。
- (2) オープンソースであるため、[②]性に優れ、OS の[③]が容易である。

練習 1.5 次のア～エの中からログインについて正しく説明した文をすべて選びなさい。

- ア Linux では、利用者を識別するために必ずログイン操作が必要である。
- イ システムに登録されたユーザが 1 人だけの場合はログイン操作は不要である。
- ウ ローカルコンソールはログイン操作無しで利用できる
- エ ネットワーク経由でログインすることをリモートログインという

練習 1.6 Linux の利用を終了して、ログインプロンプトに戻る際の操作（コマンド）として正しいものを語群からすべて答えなさい。

【語群】 ア shutdown イ login ウ logout エ 電源スイッチを押す

練習 1.7 Linux のログインについて正しい文をすべて答えなさい。

- ア Linux のすべての機能を利用するためにはローカルコンソールからログインする必要がある
- イ ローカルコンソールでは同時に複数のログインが可能である
- ウ Windows パソコンから Linux にリモートログインすることができる

練習 1.8 次の操作に使用するコマンドを答えなさい。

- ① 自分が何というアカウントでログインしているかを知る
- ② 今ログインしているユーザを知る
- ③ ログアウトする