

第 1 回 Linux 開発アプリケーションゼミ

ゼミ担当者 : 荒久田博士, 折戸俊彦, 市川親司
指導院生 : 片浦哲平, 斉藤宏樹, 谷口義樹
開催日 : 2003 年 5 月 13 日

ゼミ内容: 第一回目では, 様々なディストリビューション, X windows System について理解し「Cygwin」や「Debian GNU/Linux」のインストール方法を紹介する。それにより, Linux に関する基礎的な知識を習得する。

1 はじめに

「Linux」とは厳密には比較的小さく, 重要な核心部分のソフトウェアであるカーネル部分を意味する。現在ではこの Linux カーネルにさまざまなソフトウェアを加えたディストリビューションが各ベンダーから配布・販売されている。

ここではまず, Linux に触れる上で重要なディストリビューション, X Window System について説明する。そして, ディストリビューションの 1 つ「Debian GNU/Linux」のインストール方法を簡単に説明する。また, Windows 上で UNIX 環境を構築する cygwin についても簡単に説明する。

2 Linux カーネルと Linux ディストリビューション

Linux カーネルは OS の中核部分として, ファイルやプロセスの管理, 入出力といった基本機能を提供する。Linux カーネルだけではファイルの一覧を出すことや HD をフォーマットすること, メールを読み書きすることもできない。それらはツールとしてのソフトウェアやアプリケーションプログラムの処理として提供される。Linux カーネルだけでは実用的ではないため, OS の核となる Linux カーネルにさまざまなアプリケーションやライブラリを組み合わせたものが配布されるようになった。これが Linux ディストリビューションである (Fig. 1 参照)。

Linux は誰でも自由に再配布ができ, 関連ソフトのほとんどが GPL で配布されているので, カーネルとソフトウェアの組み合わせ次第で, さまざまなパッケージ構成が可能になる。世の中には以下に示す多種多様の Linux ディストリビューション (distribution: 配布) が存在している。

2.1 Debian GNU/Linux

2.1.1 Debian の 3 つのバージョン

当研究室において利用されているディストリビューションであり, Debian Project というボランティア

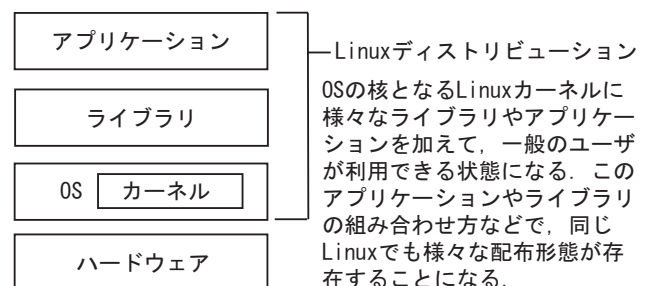


Fig. 1 カーネルとディストリビューションの構造

グループが開発する Linux ディストリビューションである。バージョンとして以下の 3 種類がある。

- stable
安定版としてリリースされたバージョン。現在バージョンのコード名は「woody」である。
- testing
開発版のパッケージの中で安定動作し, かつ動作が不安定なパッケージに依存しないパッケージで構成される。コード名は「sarge」である。
- unstable
開発版としてリリースされたバージョン。現在バージョンのコード名は「sid」である。

2.1.2 Debian の特徴

- フルパッケージ無料
「Linux はボランティアでオープンに開発されているのに, ディストリビューションは企業で閉鎖的に開発されている。ディストリビューションも Linux と同様, オープンに開発出来るはずだ」という Ian 氏¹の考えより, フルパッケージ無料となっている。パッケージとは, ある 1 つのアプリケーションに関連するプログラムや設定ファイル, ドキュメントなどをひとまとめにしたものである。

¹1993 年から, Debian GNU/Linux の開発を始めた人

Table 1 apt の基本的な使い方

apt コマンド	機能
apt-get install <パッケージ名>	パッケージをインストール
apt-get remove <パッケージ名>	パッケージを削除
apt-get -purge remove <パッケージ名>	設定ファイルを含めてパッケージを削除
apt-get update	パッケージデータベースを更新
apt-get upgrade	アップデートパッケージをインストール
apt-get dist-upgrade	ディストリビューションをアップグレード
apt-cache search <文字列>	文字列からパッケージを検索
apt-cache -names-only search <パッケージ名>	パッケージ名からパッケージを検索
apt-cache show <パッケージ名>	パッケージの概要を表示
apt-cache depends <パッケージ>	パッケージの依存関係を表示
apt-setup	apt のセットアップ

● 独自のパッケージ管理方式

一般的に Linux ディストリビューションにおいては RPM というパッケージ管理方式が使われているが、Debian では独自の dpkg というパッケージ管理方式が使われている。しかし最近では、dpkg コマンドではなく apt ツールの利用によりパッケージの管理が行われている。apt²は、パッケージ間の依存関係を考慮しつつパッケージのインストールや削除を行うツールである。apt の基本的な使い方は、Table 1 のとおりである。

2.2 Slackware

老舗のディストリビューションで、米 Walnut Creek Inc. 社がメンテナンスしている。Slackware の日本語化のために用いられてきた PJE (Project Japanese Extensions) に携わったスタッフの多くは現在「Project Vine」に移行して活動している。

2.3 Red Hat Linux

世界で最大シェアを持つ Linux ディストリビューションで、Red Hat Inc. 社がメンテナンスしている。グラフィカルなインストーラと、RPM と呼ばれる高度なパッケージ管理は、多くのユーザに衝撃を与えた。インターネット上で無償で公開されているが、商用ソフトウェアとユーザサポートを含めた商用版が特に企業にとって魅力になっている。

2.4 Turbolinux

日本語インストーラや日本語対応のアプリケーションなど、日本語環境の構築が容易なディストリビューションで、turbolinux.com 社がメンテナンスしている。日本語環境に重点を置いていて、初心者向けに配慮しているため、日本ではかなりユーザ数の多い Linux ディスト

リビューションである。Red Hat Linux をベースにしている。

2.5 VineLinux

使いやすい日本語環境を目指したディストリビューション。日本人だけで構成される「Project Vine」によってメンテナンスされており、Red Hat Linux をベースにしている。日本語インストーラ、デフォルトの日本語設定、Vine Tools と呼ばれている日本語対応のメールクライアントやエディタなど、使いやすさを重視している。

上記したとおり、ディストリビューションには多くの種類がある。つまりひと言に Linux といっても、ディストリビューションによって、細かい違いがあることになる。いくつかの Linux ディストリビューションには、市販されている製品版とフリーで配布されている FTP 版の 2 つがある。基本的には両者とも同じものであるが、製品版にはマニュアルや商用アプリケーションが付属していたり、メーカーサポートが提供されているなどの付加価値がある。FTP 版では、メーカーからのサポートなどは基本的には受けられない。

3 X Window System について

3.1 X Window System とは

X Window System は Unix で用いられる GUI で、コマンドを使わずにマウス操作で Linux を使うことができるものである。X Window System は通常「X」と呼ばれている。X は Project Athena の一環としてアメリカの MIT で開発されたものであり、現在では事実上の業界標準の位置を占めている。現在は X コンソーシアムが X の開発・配布を行っているが、自由なライセンス制度があるのでフリーのものや商用のものがある。X は Microsoft の Windows などとはまったく異なるウィンドウシステムである。Windows が OS 部分とウイン

²Advanced Package Tool

ドウシステムの部分を切り離すことができないシステムであるのに対して、X は OS から独立したウィンドウシステムとして存在している。

3.2 X の開発と特徴

X は次のような目標を持って設計、開発された。

- 移植性の高いシステムにする
- 業界の標準となるようなものを目指す
- ネットワークの使用を前提とする
- 拡張可能なプロトコルを設計する
- ハードウェアや OS には依存しない
- 誰でも入手可能とする

これらの目標はほぼ達成され、現在の X の特徴となっている。

3.3 X の構造

3.3.1 クライアント/サーバシステム

X は、クライアント/サーバモデルに基づいて作られている。クライアント/サーバモデルとは、同じ計算機資源を使う多くのプロセスがある場合、計算機資源の管理を一手に引き受けるプロセス（サーバ）を設けて交通整理をさせることにより、計算機資源の獲得に関する競合を解消しようとするモデルである（Fig. 2）。すなわち、計算機資源を利用したいプロセス（クライアント）はサーバに対して割り当て要求を出し、要求を受け取ったサーバは受け取った順に要求を処理して、結果をクライアントに返すものである。

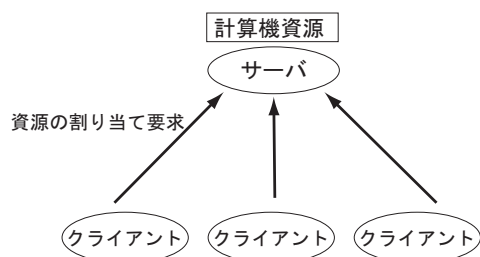


Fig. 2 クライアント/サーバモデル

X は、マウスやキーボードからの入力や画面への出力を受け持つ「X サーバー」と言われるソフトと、実際の作業を行う「X クライアント」と言われるソフトから構成されている。その概念図は Fig. 3 のとおりである。X サーバと X クライアントは「X プロトコル」と言われる規約に従って通信を行う。X サーバがコントロールできる X クライアントは 1 つに限らず、複数の X クライアントからの要求を処理することができる。X の場合、ウィンドウを利用するアプリケーション・プログラムはすべてクライアントとなる。

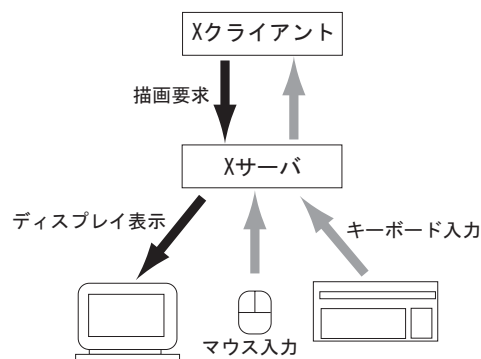


Fig. 3 X サーバと X クライアント

3.3.2 ネットワーク透過性

X クライアントと X サーバのやり取りには、ネットワークを用いるようになっている。つまり、X クライアントと X サーバの間にはネットワークがあればいいので、必ずしも同じ機械で動いている必要はないのである。もちろん同一のワークステーションで X クライアントと X サーバを動かすこともでき、また UNIX 系のワークステーションから、別の UNIX 系のワークステーションにログインして、ログインした先のワークステーションのデスクトップを表示させることもできる。つまり、隣の部屋にあるワークステーションのアプリケーションを、自分の席で利用したりすることができるようになる。

3.3.3 実際の作業

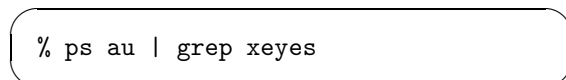
1. 自分の PC にログインし、xeyes を実行する。実行すると、ディスプレイに Fig. 4 が表示される。



Fig. 4 xeyes の実行画面

xeyes と同様「xcalc」「xclock」というコマンドがある。xeyes ではなく、これらのコマンドを実行した際は Fig. 5, Fig. 6 のような画面がディスプレイに表示される。

2. 現在、実行されているプロセスを確認する。



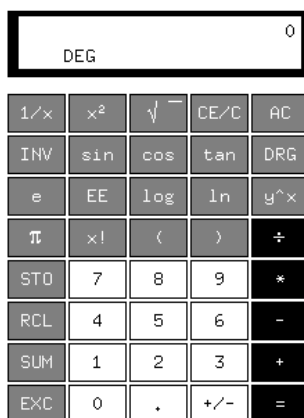


Fig. 5 xcalc の実行画面



Fig. 6 xclock の実行画面

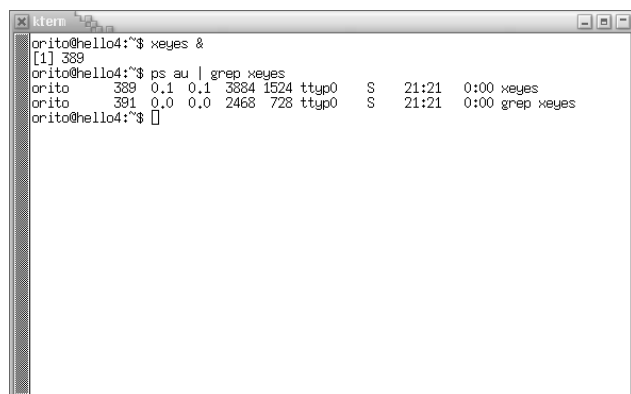


Fig. 7 プロセスの確認

3. 自分以外の PC (ここでは 192.168.6.121) にログインする . このとき , アカウント (ここでは orito) の Password が必要となる .

```
% ssh -X orito@192.168.6.121
```

4. ログイン先で xeyes を実行する .

```
% xeyes &
```

5. 自分の PC , 相手の PC の双方で実行中のプロセスを確認する .

以上の操作を行うと , ログイン先でプログラムを実行した場合 , 相手の PC で動作し , 自分の PC では動作していないことがわかる . つまり , UNIX 系のワークステーションでは , 離れた場所にある PC 資源を使用することが可能である .

3.3.4 Windows における X server

Windows での X server として代表的なものとして , アステック・プロダクツ社の「ASTEC-X」がある³ . これを用いることにより , UNIX で動いている X のアプリケーションを , Windows で動いている X server で動かすことができる .

4 Debian/Linux のインストール方法

実際に Debian をインストールする方法を記載する . Debian のインストール方法は , museion の管理者関連の技術報告をクリックし , Linux インストールマニュアルの項目の , [8] Debian GNU/Linux3.0 インストールマニュアルからダウンロードする .

5 cygwin

5.1 Cygwin とは

Cygwin は , 当初 GNU の開発環境を Win32 に移植する事を目的として開発される . しかし , Cygwin にはコンパイラやデバッガ等の開発ツールのみならず , シェルを始めとする UNIX/Linux の一般的なコマンドラインツール群が含まれているため , Cygwin をセットアップすることで Win32 上で UNIX コマンドライン環境を構築できる . インストールは Windows 上へ行うため , Windows を消したり , 再インストールする必要がない .

5.2 Cygwin で何が出来るか

開発者ではない多くの人にとっては , Windows 上で UNIX と同様の高度なコマンドライン環境を構築できることが Cygwin の最も大きな利点である . もちろん , 開発者は gcc を始めとする GNU の開発環境を使うことで ,

³<http://www.astec.co.jp/products/ASTECX/astecx.html>

Windows アプリケーションの開発を行うこともできる。また、ソースが公開されている UNIX アプリケーションを大幅なソースの変更なしにコンパイルし、Win32 上で使用することができるようになる。

cygwin は、<http://www.cygwin.com/> で配布されている。

5.3 cygwin のインストール

1. <http://cygwin.com/> にアクセスし、cygwin のインストーラをダウンロードする。
2. インストーラを起動すると Fig. 8 のような画面が出てくる。
3. "Install from Internet" を選択する。

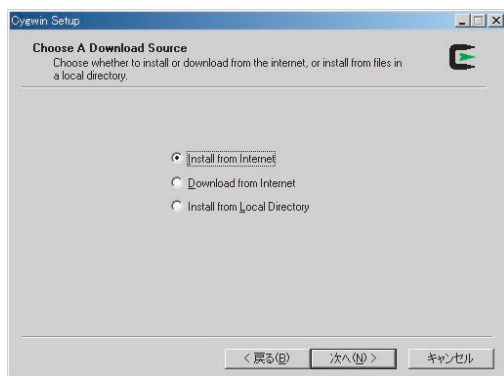


Fig. 8 cygwin のセットアップ画面

4. ディレクトリは、"C:\cygwin-pack" を入力する。(Fig. 9)

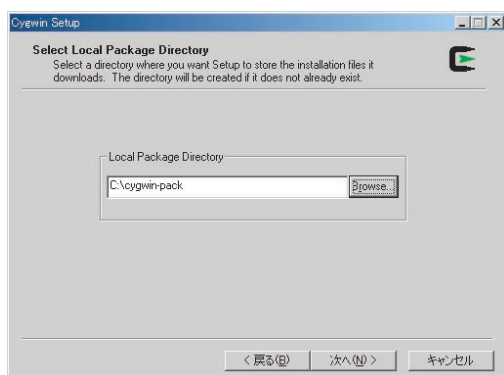


Fig. 9 cygwin のセットアップ画面

5. 接続方法は "Use HTTP/FTP Proxy" を選択し、Proxy Host に "proxyt2.doshisha.ac.jp"、Port に "8080" を入力する (Fig. 10)
6. ダウンロードサイトを選択する (Fig. 11)
7. 必要なパッケージを選択する (Fig. 12) cvs, gcc, perl, ssh, emacs は選択することをお勧めする。

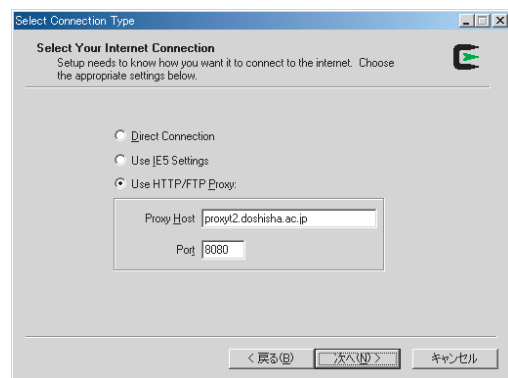


Fig. 10 cygwin のセットアップ画面

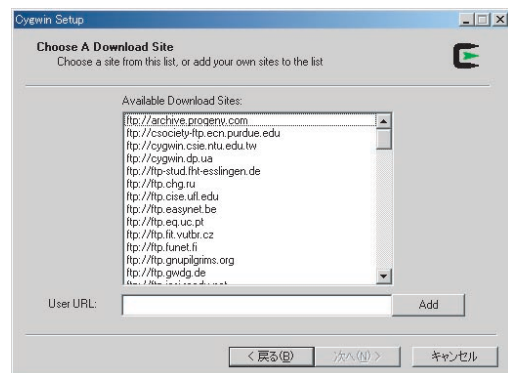


Fig. 11 cygwin のセットアップ画面

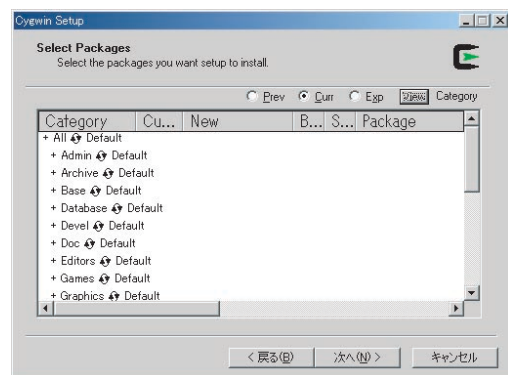


Fig. 12 cygwin のセットアップ画面

参考文献

- 1) Debian JP Project
<http://www.debian.or.jp/>
- 2) Debian GNU/Linux サポートカンパニー
<http://www.yrlinux.co.jp/>
- 3) Cygwin ~ Windows 上で実現する UNIX 環境 ~