

情報通信工学演習Ⅱ(計算機基礎) テキスト

担当: 国島丈生、山内仁

平成 14 年度版

第1章 本演習に関する概説

1.1 まえがき

みなさんは、これまで情報通信工学演習などで UNIX ワークステーションを使ってこられたと思います。

世間でコンピュータといいますと、ほとんどの場合、Windows PC であったり、(あるいは、一部の分野では) Macintosh であったりするわけですが、サーバ系では UNIX が主流ですし、研究開発部門では UNIX WS を使っている場合も多いです。また、最近 PC における新たな技術基盤 (OS) として Linux が注目を集めており、ひょっとすると、みなさんが社会にでるころには、UNIX で組まれたオフィスシステムで仕事をする、などということもあるかもしれません。

また、みなさんはコンピュータの箱 (筐体) の中がどのようなになっているか、見たことがありますか? コンピュータのソフトウェアは CPU をはじめとするハードウェア上で動きますから、メモリ管理であったり、ディスクやバスなどの速度であったり、ソフトウェアの性能を向上させるためにときとしてハードウェアの知識が必要となることがあります。実際にどのようなハードウェアからコンピュータが構成されていて、どのように動いているか、一通り知っておくと、ソフトウェアを書くときでも役に立ちます。

以上のような背景をふまえ、本実験では、UNIX の動くコンピュータを部品から組み立て、最終的に UNIX による簡単な通信ネットワークを組むまでを課題とします。

1.2 概要

2~3 人程度のグループに分かれ、グループごとに以下のような実験をやってもらいます。

計算機ハードウェア PC を部品から組み立てることを通して、計算機ハードウェアの基本的な構成や役割について理解する。

UNIX 環境 先に組み立てた PC に UNIX(FreeBSD) を導入し、環境設定やフリーウェアの導入を行う。これにより UNIX に対する理解を深める。

ネットワーク PC を同一 LAN(Local Area Network) に接続し、WWW やメールなど、インターネットを構成する基盤技術について理解を深める。

UNIX の内部技術 オペレーティングシステム (OS) の基礎について、UNIX を題材として理解を深める。

1.3 諸注意

1.3.1 レポート

各章の最後に課題をつけてある。これらの課題について、各自で(一人ずつ)レポートを提出してもらおう。出席状況とレポートとによって成績点をつける。なお、特別な事情がない限り、一回でも欠席した場合は単位は認定しない。

様式 用紙サイズ A4。そのほかは自由。手書きも可。ただし手書きの場合は読みやすく丁寧に書くこと。

提出期限 原則として、演習を行った翌週の火曜 17:00 とする。祝日等の場合は別途指示する。

レポート内容について他人と相談するのはかまわないが、レポート自体は独力で書くこと。他人のレポートを丸写しすることは認めない。丸写ししたと思われるレポートが見つかった場合、こちらではどちらがオリジナルか判断できないので、両者とも 0 点とする可能性がある。

1.3.2 実験に関する注意

PC の部品 (マザーボード、CPU、ハードディスク、各種ボードなど) は精密部品が多いので、取り扱いには慎重にしてください。1 つ 1 万円程度はする部品が多いので、部品の破損が多発した場合、(予算的に) 部品の補充ができなくなることもあります。詳しい注意点は実験の時に述べます。

また、第 2 週以降の実験では、これまでの演習等で学んだ UNIX に関するさまざまな知識が必要になります。実験中も必要に応じて再確認したり、新規の事項を説明してきますが、これまでの演習のテキストや演習室に備え付けてある書籍などを用いて適宜確認を行ってください。

第2章 計算機ハードウェア

2.1 準備

まず、以下の部品があることを確認してください。グループによって異なる部品を使う場合があるので、注意すること。

ケース (ATX ミドルタワーまたはミニタワー) 各グループ 1 個。電源ケーブル 1 本、コネクタ金具・ネジ類が付属しています。

ディスプレイ 各グループ 1 台。電源ケーブル 1 本、接続用ケーブル 1 本が付属しています。

キーボード 各グループ 1 枚。

マウス 3 ボタンタイプ。各グループ 1 個。

マザーボード 各グループ 1 枚。

IDE ケーブル 1 本

CPU 各グループ 1 個。

メモリ DIMM1 枚。

グラフィックボード 各グループ 1 枚。

イーサネットボード 各グループ 1 枚。

ハードディスクドライブ 各グループ 1 台。

フロッピーディスクドライブ 各グループ 1 台。

CD-ROM ドライブ 各グループ 1 台。

工具 プラスドライバ各グループ 1 本。

2.2 組み立て手順

以下の手順は、この順で組み立てを行わなければならないわけではない。最終的にこれらのステップはすべて行えばよく、ケースの形状などによって、作業手順を変えた方がよい場合もある。

また、組み立てる最中、ケースの縁などで手を切ることがよくある。そのため、軍手を装着して作業を行うことが望ましい。軍手がない場合は、手を切らないよう、くれぐれも注意すること。



2.2.1 ケースの分解

ケース側面をスライドさせてはずす。はずし方はケースにより異なるが、ケース前面でもネジ止めされている場合もある。

マザーボードを取り付ける金具をケースから外せる場合は、外しておいたほうが後の作業がやりやすい。

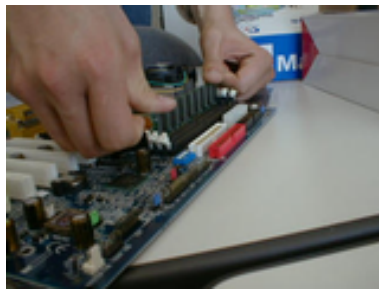
2.2.2 CPU・メモリの装着

マザーボードを取り出し、以下の作業を行う。

CPU をマザーボードの CPU ソケットにはめ込む。まずマザーボードの CPU ソケットの横のレバーを充分上げておく。CPU ソケット・CPU の双方とも一カ所だけ角が欠けているので、そこを合わせてはめこみ、横のレバーを下ろす。精密部品なので、足を折ったりしないよう、慎重に行うこと。

CPU と CPU クーラ (冷却器) が別々になっている場合は、さらに CPU クーラを装着する。CPU ソケットの出っ張りにクーラの留め具をひっかけて固定する形状になっているので、まず片方の留め具を力を入れて押し、反対側の出っ張りにひっかける。このとき、手だけでは押し切れないほど固い場合は、ドライバなどで押すことになるが、マザーボードを傷つけないよう、充分注意すること。

メモリ (DIMM) は、ソケットの両側のロックを外側に倒した状態で、切り欠きに合わせてメモリを装着する。その後、ソケットのロックを DIMM の切り欠きに入るまで押し込む。



2.2.3 マザーボードの設定

マザーボードは、取りつける CPU の種類などによって、ディップスイッチやジャンパの設定を変更する必要がある場合がある。マザーボードのマニュアルを見ながら、ジャンパが正しく設定されているか確認する。とくに CPU の種類とあっているか確認すること。

2.2.4 マザーボードの取り付け

マザーボード取り付け金具にマザーボードをねじ止めし、ケースに装着する。できあがりの状態を想定し、位置や必要な部品を確認しながら作業をすすめること。

マザーボードと取り付け金具の位置あわせを行い、どの部分をネジで固定するか、あらかじめ決めておく。加工の精度の問題で、多少うまく合わない場合があるので、丁寧に位置あわせを行うようにすること。続いて、マザーボードを取り付け金具に固定する。ワッシャがケースの付属品にあ





る場合は、ワッシャをはめるのを忘れないこと。また、精度の問題でうまくネジが止められないときは、無理に止めないこと。全部で4個所以上止まっていれば、充分固定できていると考えてよい。ケースへ装着する際、固定用金具の形状に注意し、必要なら新たに穴を開ける。

2.2.5 各種ボードの取り付け

マザーボードに、グラフィックボードとネットワークボードを取付ける。ネットワークボードはすべて PCI スロット (白) を使用する。また、グラフィックボードは AGP スロット (マザーボード中央辺りの茶色のスロット) のものと PCI スロットのものとの2通りがある。

そのスロットに合う位置の取り付け金具を外し、ボードをはめ込み、ケースにねじ止めする。スロットが固い場合があるが、基盤面そのものには手を触れないように作業を行うこと。

2.2.6 FDD・HDD・CD-ROM ドライブの取付け

本体にフロッピーディスクドライブ (FDD)、ハードディスクドライブ (HDD)、CD-ROM ドライブを取り付ける。

HDD や FDD を装着する金具 (3.5 インチベイ) は、多くの場合ケースから取り外すことができる。外してからドライブを装着したほうが作業がしやすい。ただし、FDD はケース前面との位置合わせが必要になる。ケースの前面パネルを取り付けた状態でドライブを入れ、ネジ穴の位置決めをする。その後、ベイの両面からネジを締め、ドライブを固定する。CD-ROM ドライブも同様に位置決めが必要になるが、横にガイド用金具を取りつけてからケース本体に固定する場合もある。この場合は位置決めは不要である。



HDD は、基盤面を下側にする (ほこり防止のため)。

なお、後述のように接続ケーブルの向きの指定が本体に書かれているので、それを確認してから取り付けを行うこと。

2.2.7 各種配線

配線すべきものは次のようなものがある。

- マザーボードの電源ケーブル
- CPU ファンの電源ケーブル
- 本体前面の各種スイッチ・スピーカなどの配線 (色のついているほうが+)
- HDD, FDD, CD-ROM ドライブなどの電源ケーブル
- HDD, FDD, CD-ROM ドライブなどとマザーボードとの接続ケーブル

マザーボードとの配線については場所や向きを間違えないよう、マニュアルを見ながら行うこと。また、HDD, FDD, CD-ROM ドライブなどとの配線も逆向きに差しがちである。次のような目安にしたがい、配線を間違えないようにすること。

- 本体の方に接続ケーブルの 0 番 (赤い側) の場所が指定されているので、それに合わせて接続ケーブルをつなぐ。
- FDD の接続ケーブルは途中でねじれている方を FDD 側に接続する。また、両端を使う (途中のコネクタは FDD を 2 台以上接続する際に使う)

- HDD はマザーボードの IDE1 に、また CD-ROM ドライブはマザーボードの IDE2 にそれぞれ接続。

2.2.8 電源投入・BIOS の確認

ケースの側面を閉める前に、この状態で電源を入れ、正常に動作するか確認しておく。

各種接続がうまくできているかももう一度確認した後、本体の電源ケーブル、ディスプレイとの接続ケーブル、ディスプレイの電源ケーブル、キーボード、マウスを接続し、電源を投入します。OS 起動用フロッピーディスクを要求するメッセージが出れば、とりあえずは成功である。

一般に PC の場合は、この段階で BIOS (Basic Input Output System) の設定確認を行う必要がある。BIOS は、PC の電源を投入したりリセットを行ったときの最初の起動、基本的な入出力機能、マザーボード等のハードウェアの基本設定を行うプログラムであり、この設定が間違っていると、ハードウェア的には正しく組み上がっていても PC は動作しない。

BIOS の設定画面への入り方はマザーボードによって異なる。今回使用するマザーボードでは、次の 2 通りがある。

- 画面に「PRESS DEL TO ENTER SETUP」というメッセージが表示されたときに **DEL** キーを押す
- 起動直後に表示されるメーカーのロゴ画面表示中に **F2** キーを押す

BIOS 設定画面にはさまざまなチェック項目があるが、CPU の種類・クロック数、HDD の接続先、起動順 (FDD や CD-ROM を優先させるなど) などをチェックしておく必要がある。チェック項目は演習実施時に別途指示する。

課題

1. 実験の過程で直面したトラブルについて、その症状、原因の究明過程、原因、解決方法などについて (自分で調べるなどして) 記述せよ。
2. マザーボード、グラフィックボード、ネットワークボード、ハードディスクドライブ、フロッピーディスクドライブ、CD-ROM ドライブについて、使用したものの型番を記せ。
3. マザーボード、グラフィックボード、ネットワークボードについて、IC の配置がわかるように概略をスケッチせよ。また、主要な IC について、チップに印刷されている型番を記録せよ。

コラム: IDE

今回使用している HDD や CD-ROM ドライブは IDE という規格に基づいています。通常のマザーボードでは、IDE のデバイスを 4 つまで接続することができます (Primary, Secondary 二系統にそれぞれ Master, Slave がある)。Primary Master に接続されているデバイスから最初に起動しようと試みますので、ここに通常 HDD を接続します。
HDD や CD-ROM ドライブの規格には、ほかに SCSI が有名です。

4. CPU(中央演算処理装置)、メモリ (一次記憶)、ハードディスク (二次記憶) について、それぞれの役割、関連を調べよ。

第3章 UNIX環境構築

3.1 まえがき

以前はUNIXというと、演習室にあるようなワークステーションでしか動作しないものでした。ところが、最近、PCで動作するUNIX(PC-UNIXと呼ばれることがあります)が登場し、UNIXを取り巻く環境が大きく変わってきています。特に、Linux (<http://www.linux.or.jp/>) は、一般ユーザや多くのソフトウェアベンダーを巻き込んで大きなブームを巻き起こしています。

今回の演習では、PC-UNIXの中からFreeBSD (<http://www.jp.freebsd.org/>) というものを取り上げ、これをインストールすることでUNIX環境を構築することにします。FreeBSDは、Redhat, TurboLinux, Vine LinuxなどのLinuxに比べて以下のような特徴があり、今回の演習の目的に合っていると考えられます。今回の演習では、FreeBSD 4.2-RELEASE¹ を用いることとします。

- カーネルと呼ばれるUNIXの中核部分、各種コマンドなどが、そのソース(C言語)を含めて統一的に管理されている。
- 余分なアプリケーションを入れずにインストールできる。

なお、本来、PC-UNIXをインストールするには、PCのハードウェアやUNIXに関する知識がいろいろと必要になりますが、これを詳しく説明すると一冊の本が書けてしまいます。今回は、インストールに必要な最低限の説明にとどめることにします。さらに詳しく知りたい人や自分でもインストールしてみたい人は、適宜書籍を参照してください。

— 2 種類の UNIX —

一口に「UNIX」といっても、Solaris(Sun Microsystems 社、演習室のワークステーションに搭載されている)、HP-UX(Hewlett Packard 社)、IRIX(Silicon Graphics 社)、Linux、FreeBSD など、実際には多くの種類が存在します。そしてこれらは、カリフォルニア大学バークレイ校で開発が進められた「BSD」系列と、AT&T ベル研究所で開発が進められた「System V」系列とに大きく分けることができます。Solaris はどちらかという System V (System V Release 4)、FreeBSD はどちらかという BSD に分類することができます。ただし、最近は、BSD 系列と System V 系列とで機能的な差はほとんど無くなってきており、その境界は曖昧になってきています。

¹2002 年 11 月現在の最新リリースは 4.7-RELEASE。

3.2 基礎知識

3.2.1 オペレーティングシステム

オペレーティングシステム (Operating System, OS) は各種デバイスの制御のための機能の提供、応用プログラムの実行制御など、計算機が実際に動作するために必要な基本機能を提供するソフトウェアである²。各種 Windows や Linux・UNIX, MacOS などすべてオペレーティングシステムである。

オペレーティングシステムが異なると実行制御の方法が異なるため、CPU が同じであってもプログラムに互換性はない。例えば、Windows の Microsoft Office が Linux 上で動作しないのは、対象としているオペレーティングシステムが異なるためである。

3.2.2 複数オペレーティングシステムの共存

PC アーキテクチャでは、ディスク上に区画 (スライス³, slice) を作成し、区画内にオペレーティングシステムを導入する方式を採用している。一つのディスクには複数の区画が作成できるため、複数のオペレーティングシステムを一つのディスク上に共存させ、ブートマネージャ⁴ と呼ばれるソフトウェアを利用して起動時に切り替えて使用することが可能となっている。

3.2.3 UNIX

UNIX にはさまざまな種類があるが、その実行制御方法は似かよっている。Linux も、実行制御という点からみれば UNIX の一種と言える。

UNIX は、同時に複数のプログラムが動作することのできる OS (マルチタスク OS) であり、その構成は次のようになっている。

- カーネル (kernel): 複数プログラムの実行制御、デバイス制御を行う。
- 各種常駐プログラム (デーモン, daemon)、ライブラリなどの基本部分
- 応用プログラム。Emacs や Netscape などこれに相当する。

また、UNIX ではパーティション内をさらに複数の区分 (パーティション⁵, partition) に分割し、区分ごとに用途を変えることがある。

3.3 準備

今回の演習では必要ありませんが、通常は以下のようなことをあらかじめ準備しておく必要があります。

²詳細は「システムプログラム」の講義を参照。

³FreeBSD での用語。一般にはパーティションと呼ばれることが多い。

⁴FreeBSD での用語。

⁵FreeBSD での用語。他の UNIX では異なる用語を用いることがある。

3.3.1 使用するデバイスの整理・情報収集

現状の PC-UNIX では、安定動作するハードウェアデバイスの種類が Windows よりも少ない。また、動作する場合でも OS の初期値ではうまく動作せず、使用条件に応じてパラメータを調整する必要があることもある。

あらかじめ WWW 等で、インストールしようと思っている PC-UNIX でサポートされているデバイスを調査し、それに基づいて PC の機種を選んだり部品をそろえたりしなければならない。今回は、あらかじめ、FreeBSD で動作する部品ばかりをそろえてある。

3.3.2 インストールメディア

OS のインストール CD-ROM やブートフロッピーディスクを適宜用意する必要があります。FreeBSD でも、CD-ROM から起動してインストールを行うことができますが、現状ではこの場合、英語でインストールメニューが出ます。そのため、今回は、フロッピーディスクから日本語版インストーラを起動し、配布する CD-R からファイルをコピーするという方法をとります。

3.3.3 確認

BIOS で、フロッピードライブ (A) からの起動になっていることを確認。また、イーサネットケーブルが PC に接続されていることを確認しておく。

3.4 インストール手順

3.4.1 概要

1. ブートフロッピーディスクから起動
2. 必要なら各種ハードウェアのパラメータ設定 (今回は必要なし)
3. 区画 (slice) の作成
4. 区画内の区分 (partition) の作成
5. インストールする配布物の選択
6. インストール
7. インストール後の設定

FreeBSD では、3. より後の作業は `sysinstall` と呼ばれるプログラムから行う。

3.4.2 ブートフロッピーディスクからの起動

FreeBSD 4.2-RELEASE のブートフロッピーディスクは kern.flp, mfsroot.flp の 2 枚から成っています。

1. kern.flp をフロッピーディスクドライブに挿入して PC を起動
2. 画面の指示にしたがってフロッピーディスクを mfsroot.flp に入れ替える
3. 配布した CD-R を CD-ROM ドライブにセット

こののち、インストール用の FreeBSD カーネルが起動し、インストーラが立ち上がります。

3.4.3 各種ハードウェアのパラメータ設定: Kernel Configuration Menu

ハードウェアのデバイス情報 (IRQ, I/O ポートアドレスなど) を編集する画面。これを用いることで、標準で立ち上がるインストール用カーネルの設定情報を変更し、標準状態では正しく動作しないハードウェアでもインストールに用いることができます。

今回は特に設定する必要はありませんので、キーを用いて「Skip kernel configuration and continue with installation」へカーソルを持っていき、**Enter** キーを押します。ここからは GUI (Graphical User Interface) ベースのインストーラ (sysinstall) が起動します。

3.4.4 sysinstall

sysinstall では、画面に出てくる指示にしたがってインストールを進める。やや操作方法が独特で間違えやすいので、以下の点に注意して操作すること。

- 項目を移動 項目の選択 OK かキャンセルかを選択 実行という流れになる。
- 場合によっては、Exit 項目を選択して上位メニューに戻らなければならないこともある。
- 項目間の移動は矢印キーまたはタブキー
- 項目の選択が **SPACE** キー。サブメニューに入る場合と、[X] マークがついて選択される場合がある。
- OK かキャンセルかの選択は **TAB** キー
- 項目の実行が **ENTER** キー (OK, キャンセルとも)

Localization Menu

インストーラの用いる言語の選択。英語、日本語、韓国語、ロシア語、中国語が使用できる。以下の説明は「日本語」を選択したとして仮定している。

メインメニュー

一連のインストール作業は”Standard” (標準インストール) で行うことができる。また、インストールオプションの選択、インストール後の設定などに入ることができる。

スライスの設定: FDISK パーティションエディタ

スライスの作成を行う。今回は FreeBSD しかインストールしないので、「ディスク全体を使用 (A)」を実行した後、「終了 (Q)」を実行。「本来のパーティションテーブル」は保持しておくようにする。

ブートマネージャのインストール

OS の起動切り替えを行うプログラムのインストールなど。今回は切り替えの必要がないので”Standard”を選択。

FreeBSD ディスクラベルエディタ

FreeBSD 用の区画 (slice) の中をさらにいくつかの区分 (partition) に分割することができる。今回は「全てをデフォルトに設定 (A)」を実行した後、「終了 (Q)」を実行する。

配布ファイルの選択

インストールするファイル群の指定。あとの演習で必要となるので、以下のようにする。

- ”X-Developer” (全ソース、バイナリ、X Window System) を選択。
- ports コレクション.....インストール
- XFree86 配布物.....以下のもの (デフォルト値がこうなっているはずである)
 - Basic(基本コンポーネント).....bin, cfg, lib, man, prog, set
 - Server(X サーバ).....SVGA, VGA16, Mach64
 - Fonts(フォント).....fnts, non

インストールメディア

インストールする配布物をどこから取得するかを指定。CD-ROM のほか、ネットワーク経由で取得することも可能になっている。今回は”CDROM”を選択。

このあと確認のメッセージが出たあと、インストール作業が始まる。ここまでの作業に間違いがなければ約 15 分程度かかる。時間が短すぎる場合はどこかに間違いのあった可能性がある。

3.5 インストール後の設定

インストールが終了した後、インストール後の設定を行う。下記の項目以外については、表 3.1 を参照のこと。

イーサネット ネットワークインタフェースとして”fxp0” (Intel EtherExpress Pro/100B ethernet device driver) を選択した後、表 3.2 のように設定する。そのあと、インタフェースを有効化する。

コンソール機能のカスタマイズ Yes。キーボードマップのみ設定。「システムコンソールキーマップ」メニューに入り、日本語キーボードの場合は”Japanese 106”とする。英語キーボードの場合は何も設定せず Cancel。

タイムゾーンの設定 Yes。CMOS クロックは UTC(世界標準時) に合わせずに使用。その後、タイムゾーンを”Japan”(省略形 JST) に設定。

マウスの設定 USB 以外のマウスが接続されている。マウスの種類...”Auto” (自動選択)、マウスポート.....PS/2、追加フラグ...なし

システム管理者 (root) のパスワード 本来は厳重に管理すべきものですが、今回は実験の都合上、ホスト名と同じものにしておいてください。

表 3.1: インストール後の設定

項目名	設定内容
末端のホストか	Yes
FTP アクセス	通常ユーザのみ許可
NFS サーバ	No
NFS クライアント	No
セキュリティレベルの設定	No (medium レベルに設定)
Linux バイナリ互換機能	No
X サーバの設定	No(あとで行う)
パッケージコレクション	No (あとで行う)
ユーザアカウントの設定	No (あとで行う)

このあと総合設定メニューに移動し、“Cancel”を実行してメインメニューに戻る。これでインストール作業は終了なので、画面の指示に従い、PC を再起動する。CD-ROM を取り出すのは、再起動が始まってからにすること。

このあと起動中に、各種デバイスが認識された旨のメッセージが出力されるので、よく見ておくこと。これらのメッセージの多くは、“dmesg”というコマンドによって表示させることができる。次のような表示が出れば、インストールは成功である。

Wed Oct 6 09:46:52 JST 1999

FreeBSD/i386 (comx200.c.oka-pu.ac.jp) (ttyv0)

login:

表 3.2: イーサネットオプションの設定一覧

項目名	値
IPv6	使用しない
DHCP	使用しない
ホスト名	演習時に指示
ドメイン名	c.oka-pu.ac.jp
IPv4 ゲートウェイ	163.225.216.254
ネームサーバ	163.225.215.252
IPv4 アドレス	演習時に指示
ネットマスク	255.255.255.0(自動設定される)
ifconfig への特別なオプション	なし

3.6 起動後の設定

3.6.1 ログイン

管理用アカウント (root) でログインしてみます。X ウィンドウもまだ使えませんし、mule(emacs) や netscape など当然まだ動作しませんが、ls, cd, pwd などの基本的な UNIX コマンドは動作しますので、いろいろ探検してみてください。たとえば、/usr/src には、今動いている FreeBSD 自体のソースファイル (C 言語で書かれています) が一式置いてあります。

3.6.2 留意点

ついで、インストール時に行わなかった追加の設定をいくつか行う。ここでは、インストール時に用いた GUI プログラム sysinstall (/stand/sysinstall) を用いるが、以下の点に留意すること。

- sysinstall はシステム管理者権限で実行すること。
- 日本語メニューを使用してから sysinstall を終了すると、キーボードからの入力が端末に出力されない状態におちいることがある。これは sysinstall のバグであり、“stty sane”というコマンドを実行すると復旧する。

3.6.3 ユーザアカウント作成

システム管理者アカウント“root”というユーザは強力な権限を持ち、システムを破壊してしまうことも可能である。したがって、必要のあるとき以外極力使わないようにすることが望ましい。普

段は「一般ユーザ」と呼ばれる、より権限の弱いユーザで作業を行い、必要になったら root の権限を取得する⁶のが、UNIX での一般的な管理方法である。

そこで、まず一般ユーザのアカウントを作成する。

1. sysinstall を起動し、“Configure”を選択。ついで設定メニューで“User Management”を選択。
2. 「ユーザグループをシステムに追加する」を選択。
3. ユーザグループ: users, グループ ID: 1001 を入力し、「OK」を実行。「グループメンバ」は空欄でよい。
4. 「ユーザをシステムに追加する」を選択。
5. ログイン ID: user???, UID: 1??? (???は各ホスト名の下3桁の数字と同じ)、グループ: users、ログインシェル: /bin/csh。パスワードはグループごとに決める。ほかはそのままでよい。
6. 「Exit」を選択し、「FreeBSD 設定メニュー」に移動。
7. 「Exit」を選択し、「/stand/sysinstall メインメニュー」に移動。
8. 「導入終了」を選択。

このあと、作成した一般ユーザで実際にログインできるようになったことを確かめ、さらにシステム管理者権限で/etc/group というファイルを編集する。具体的には、“wheel*:0:root”と書かれた行の最後に“user???”を追加する。たとえば、user201 を追加した後の状態は次のようになる。

```
wheel*:0:root,user201
```

3.6.4 hosts ファイルの設定など

以降の実験でほかのホストにアクセスしたり、適宜スーパーユーザの権限で作業したりするので、そのための設定をしておく。

システム管理者権限で/etc/hosts を編集し、次の内容を末尾に付け加える。すでに書かれている行は書かなくてよい⁷。

163.225.216.200	comx200.c.oka-pu.ac.jp	comx200
163.225.216.201	comx201.c.oka-pu.ac.jp	comx201
163.225.216.202	comx202.c.oka-pu.ac.jp	comx202
163.225.216.203	comx203.c.oka-pu.ac.jp	comx203
163.225.216.204	comx204.c.oka-pu.ac.jp	comx204
163.225.216.205	comx205.c.oka-pu.ac.jp	comx205
163.225.216.206	comx206.c.oka-pu.ac.jp	comx206
163.225.216.207	comx207.c.oka-pu.ac.jp	comx207
163.225.216.208	comx208.c.oka-pu.ac.jp	comx208

⁶このための UNIX コマンドが“su”である。

⁷ここまでの作業が正しくできていれば、自分の班のホスト名はすでに書かれているはずである。

163.225.216.209	comx209.c.oka-pu.ac.jp	comx209
163.225.216.210	comx210.c.oka-pu.ac.jp	comx210
163.225.216.211	comx211.c.oka-pu.ac.jp	comx211
163.225.216.212	comx212.c.oka-pu.ac.jp	comx212
163.225.216.213	comx213.c.oka-pu.ac.jp	comx213

3.6.5 パッケージ

次週以降の演習で用いるプログラム類をインストールしておく。なお、ここでは「パッケージ」と呼ばれる、コンパイル済みのプログラム・設定ファイル一式をインストールする。一般には、これらのプログラムはC言語などで書かれたファイル(ソースファイル)で配布されているので、これを自分でコンパイルしなければならない。

1. CDROM をセットした後、/stand/sysinstall を起動して、“Configure”を選択。
2. 設定メニューで“Packages”を選択。
3. 配布先として“CDROM”を選択。
4. japanese カテゴリの以下のパッケージを選択: ja-kterm-6.2.0, ja-netscape-communicator-4.76
5. パッケージ選択メニューに戻り、“Install”を実行。

3.6.6 XFree86 の設定

普段みなさんがなにげなく使っている X ウィンドウシステムだが、Linux や FreeBSD をインストールする場合、最大の関門が X ウィンドウシステムの設定である。それは以下の理由による。

- PC の場合グラフィックボードが多種多様で、それぞれによって X サーバが異なる。
- X サーバを選択するだけでなく、グラフィックボードごとに適切なパラメータを与えて設定を行う必要があるが、この際、グラフィックボードのハードウェアに対する深い知識が必要になる。
- 極端に間違った設定を行うと、特にノート PC や液晶ディスプレイでは、ハードウェアを壊しかねない。

今回は、この部分はあらかじめ作成した設定ファイルを取得してもらうこととした。anonymous ftp で XF86Config.101 (英語キーボードの場合) または XF86Config.106 (日本語キーボードの場合) というファイルを取得し(場所は別途指示する) た後、root でログインして以下の手順を行う。

```
取得したファイルを /etc/XF86Config にコピーする。たとえば:  
# cp XF86Config.101 /etc/XF86Config  
その後、次の作業を行う。  
# cd /usr/X11R6/bin  
# ln -s XF86_Mach64 X  
# rehash  
# startx
```

— UNIX のユーザ管理 —

UNIX では、ユーザ、ユーザの集合である「グループ」の 2 つでユーザを管理している。それぞれ便宜上名前になっているが、システム的には UID(ユーザ ID)、GID(グループ ID) という数字で管理が行われている。ファイルやディレクトリに対するパーミッション (アクセス権管理) は、「持ち主のユーザ」「持ち主の属する (ある) グループ」「それ以外のユーザ」の 3 レベルに分けて管理される。たとえば、`chmod 755 foo.sh` という UNIX コマンドを実行すると、「持ち主のユーザ」は `rwX`(2 進法で 7、読み出し・書き込み・実行可能)、「持ち主の属する (ある) グループ」は `r-X`(2 進法で 5、読み出し・実行可能だが書き込み不可)、「それ以外のユーザ」は `r-X` というようにアクセス権が設定される。

なお、UNIX ではシステム管理者アカウントは `root` という名前になっている。たいていのシステムファイルはシステム管理者の権限でなければ変更することはできないが、システム管理者の権限を得るには次の 2 通りの方法がある。

- アカウント `root` でログインする。
- 一般ユーザでログインしている状態で `su` コマンドを実行する。

— vi 簡易マニュアル —

- `h, j, k, l`.....カーソルをそれぞれ左、下、上、右へ移動
- `i` 文字列 `ESC`.....文字列をカーソル位置から挿入
- `a` 文字列 `ESC`.....文字列をカーソル位置の 1 文字後ろから挿入
- `o` 文字列 `ESC`.....文字列をカーソル位置の次の行に挿入
- `x`.....カーソル位置の文字を 1 文字消去
- `:w!`.....強制書き込み (セーブ)
- `ZZ`.....セーブ後終了
- `:q!`.....セーブしないで終了

— ホスト名と IP アドレス —

インターネットに接続されているすべての計算機は、全世界で一意に決まる番号 (IP アドレス) と、ホスト名とを持っている。たとえば、情報通信工学科のメールサーバの IP アドレスは `163.225.215.253`、ホスト名は `com-server1.c.oka-pu.ac.jp` である。`/etc/hosts` は、ホスト名と IP アドレスの対応表である。また、`/etc/hosts` に書かれていないホストの IP アドレス等の情報を取得するサービスが DNS(Domain Name Service) であり、そのサービスを提供するサーバを俗に「ネームサーバ」と呼ぶ。

これで X ウィンドウが立ち上がれば成功である。

最近のバージョンの XFree86 ではセキュリティが強化されており、このままの状態では一般ユーザで startx が実行できない。この解決策はいくつかあるが、ここでは XDM (X Display Manager) を使った方法を示す。startx が実行できることが確認できたら、以下の作業を行う。

ファイル `/etc/ttys` を編集。途中に次のような行があるので

```
ttyv8    "/usr/X11R6/bin/xdm -nodaemon"  xterm    off secure
```

これを

```
ttyv8    "/usr/X11R6/bin/xdm -nodaemon"  xterm    on  secure
```

と書き換える。その後、再起動を行う。

時間的余裕のある班は、教官もしくは TA の立ち会いのもと、自分たちで XF86Config を作成してもよい。将来 PC-UNIX に自分で触れたいと思っている人は、なるべく XF86Config を自分で作成する作業をやっておいた方が望ましい。

3.6.7 emacs のインストール

emacs はふだん頻繁に用いているツールであるが、OS に標準で含まれているプログラムではない⁸。後の作業のために emacs をインストールし、システム管理作業の一端を体験する。

手順

anonymous ftp を用いて `comx200.c.oka-pu.ac.jp/pub` から `make-3.79.1.tar.gz`, `emacs-20.7.tar.gz` というファイルを取得した後、以下の作業を行う。

Emacs のコンパイル作業には GNU make (gmake) が必要である。そのため、まず gmake をコンパイル・インストールしてから Emacs のコンパイル・インストールを行う。

リスト 1: GNU make のインストール

```
% gzcac make-3.79.1.tar.gz | tar xvf -
(この作業を行うと make-3.79.1 というディレクトリが作られる)
% cd make-3.79.1
% ./configure --program-prefix=g
% make
% su
# make install
# exit
% rehash
```

リスト 2: emacs のインストール

⁸多くの Linux ディストリビューションや MacOS X など、標準で含まれている UNIX も増えてきている。

anonymous ftp(匿名 ftp)

ftp というコマンドは、ftp をかける側の計算機にもユーザアカウントを持っていないと実行できない。これでは、フリーウェアなどをインターネットで公開する場合、ユーザアカウントの管理がたいへんである。そこで、anonymous というユーザ名でだれでも ftp ができるようにしている場合がある。この場合、パスワードは自分のメールアドレスを入力することがマナーになっている。WWW ブラウザで、"ftp://..." という URL にアクセスすることがあるが、これは上記の anonymous ftp の処理をブラウザを通して行っているのである。

anonymous ftp は次のような手順で行う。**psi.c.oka-pu.ac.jp** というホストに **anonymous ftp** を行っている例を示す。なお、一般ユーザで書いているが、root で行ってもかまわない。

```
% ftp psi.c.oka-pu.ac.jp
Connected to psi.c.oka-pu.ac.jp.
...
Name (psi.c.oka-pu.ac.jp:user201): anonymous
...
Password: (自分のメールアドレス)
...
ftp> cd ディレクトリ
...
ftp> get 取得したいファイル名
...
ftp> quit
...
%
```

途中で出力されるメッセージを良くみて、正しくファイルが取得できたことを確認しておくこと。とくに、ftp を実行したディレクトリに書き込み権限がないと、ファイルを取得できない。


```
% gzcac emacs-20.7.tar.gz | tar xvf -  
  (この作業を行うと、emacs-20.7 というディレクトリが作られる)  
% cd emacs-20.7  
% ./configure  
% gmake  
% su  
# gmake install  
# exit  
% rehash
```

この作業を終えると emacs が使えるようになる。

3.6.8 UNIX の停止方法・再起動の方法

緊急の事態を除き、PC の電源を落とす場合は UNIX システムに悪影響を与えないような手順を踏む必要があります。

電源停止の場合は次の手順をシステム管理者権限で実行してください。

1. shutdown -h now という UNIX コマンドを実行する。
2. 「電源を落としても良い」旨のメッセージ (英語) が出力されたら、電源を切る。コンセントを引き抜くなどの手荒なことはしないこと。また、メッセージが出力された段階でキーボードのキーを押してしまうと PC が再起動するので、注意すること。

なお、今回実験に用いた PC によっては電源ボタンを押すだけでは電源の切れないものがある。この場合は、キーボードのキーを適当に押し、再起動が始まった瞬間に電源ボタンを押すとよいようである。

3. 電源を切った後、必要に応じてディスプレイの電源を切る。

課題

1. 実験の過程で直面したトラブルについて、その症状、原因の究明過程、原因、解決方法などについて (自分で調べるなどして) 記述せよ。
2. PC の起動時に表示されたメッセージを dmesg コマンドを使って確認し、それらをレポートに添付しなさい。その中で自分達が使った部品がどこに表示されているか、わかる範囲で明記しなさい。

GNU make

gmake (GNU make) は Free Software Foundation で作られた make コマンドであり、Solaris に標準で付属している make よりもひじょうに多くの機能を持つ。また、UNIX であればたいていコンパイルすることができるので、GNU make の使用を前提にすれば OS を気にすることなく Makefile を記述することができる。そのため、Emacs や tcsh など、最近のフリーウェアの make は GNU make を使うことを前提としていることが多い。

3. ディスク領域を複数のスライスに分割することのメリット・デメリット、および一つのスライスを複数のパーティションに分割することのメリット・デメリットを考察しなさい。
4. OS はどのようにして起動されるのか、ハードウェアに電源が入ってから過程を調べて書け。

— configure と xmkmf —

UNIX にもさまざまな種類があり、それぞれでヘッダやライブラリなどが微妙に異なっている。したがって、どの UNIX でも動作するプログラムを書くのは、プログラムの規模が大きくなるにつれ、難しくなる。

このような汎用的なプログラムを書くための支援ツールとしてよく知られているのが GNU configure と xmkmf である。これらはいずれも OS に用意されているライブラリなどを動的にチェックし、環境に適したコンパイラオプションやヘッダファイルを生成してくれるコマンドである。

第4章 LAN

4.1 WWW

4.1.1 WWW の仕組み

WWW(World Wide Web)はサーバ・クライアント型のシステムであり、「WWW サーバ」と「WWW クライアント」との通信によって動作している。

WWW クライアントは、いわゆる WWW ブラウザである。一方、WWW サーバはコンテンツ(ホームページのソースファイルや画像など)を保持し、WWW クライアントからの要求に応じてコンテンツを送信する。この送受信のプロトコル¹がHTTP(HyperText Transfer Protocol)と呼ばれるものである²。WWW クライアントで受信されたコンテンツは、WWW クライアント側で整形され画面に表示される。

WWW サーバと WWW クライアントの通信の様子を見てもらうため、各 PC に WWW サーバと WWW クライアントをインストールし、実際にネットワークを介してアクセスしてもらう。

4.1.2 必要な設定の確認

この章の課題を行うには、これまでの課題・設定が正しく行えている必要がある。次の点に注意しておくこと。

- netscape が導入できていない.....第2回の「FreeBSD のインストール」の際に XFree86 のパッケージがうまくインストールできておらず、netscape パッケージの導入に失敗した可能性が高い。配布した CD-ROM をセットし、XFree86 を正しく導入した後、/stand/sysinstall を用いて netscape を導入する。
- /usr/ports が無い、もしくはその下に何も無い.....上と同じく、OS のインストール時に選択を誤っている。同じく、/stand/sysinstall を用いて ports を導入する。
- /etc/hosts の設定.....OS のパッケージを追加インストールした場合、選択を誤ったなどの理由で/etc/の下にファイルが上書きされている場合がある。特に/etc/hosts に自班・他班のホスト名がすべて書かれている必要がある。

¹データのやり取りの方式や手順に関する規約。

²URL の先頭の“http:”は、HTTP でコンテンツの送信を行うという意味である。

4.1.3 テキスト型 WWW ブラウザ w3m

最近の UNIX 環境では、使いやすさ、バージョンアップのしやすさなどを考慮して、パッケージ管理ツールが用意されていることが多い。Linux の rpm や apt (Debian)、FreeBSD などの ports、Solaris の pkg.info などがある。パッケージ管理ツールでは、おおむね次のような機能が提供されている。

- どのファイルがどのパッケージに属するものを記録
- ファイルの一括追加・削除
- インストール時の設定の支援機能
- パッケージのバージョンアップ

ここでは、FreeBSD で提供されている ports というパッケージ作成・管理支援ツールを用い、w3m というテキストベースの WWW ブラウザをインストールする。

comx200:/pub から w3m-0.1.11-pre.tar.gz, w3m-0.1.11-pre-kokb17.patch.gz, gc5.0alpha4.tar.gz を取得し、/usr/ports/distfiles/に置く。/usr/ports 以下に何もファイルがない場合は、第 2 回目の OS のインストールに失敗しているため、/stand/sysinstall を用いて ports を追加インストールする。

```
% su -  
# cd /usr/ports/japanese/w3m  
# make install  
# exit  
%
```

図 4.1: w3m のインストール

このあと、課題 1 を行う。

4.1.4 WWW サーバ～apache

ついで WWW サーバをインストールする。今回は、フリーウェアの WWW サーバでもっとも有名な Apache (<http://www.apache.org/>) というソフトウェアを使うことにする。これも ports でインス

ports

w3m のインストールにおいて、“make install”とただでアプリケーションがインストールされたことに注意されたい。FreeBSD の ports は、make を用いてアプリケーションのソースファイルの取得、展開、configure 等のコマンドの実行、コンパイルなどを自動で制御している。

ツールを行う。comx200:/pub から apache-1.3.??tar.gz というファイルを取得し、/usr/ports/distfiles/に置く。そののち、図 4.2 に示す手順でインストールを行う。

```
# cd /usr/ports/www/apache13
# make install
# cd /usr/local/etc/apache
# cp access.conf.default access.conf (もし access.conf がなければ)
# cp httpd.conf.default httpd.conf (もし httpd.conf がなければ)
# cp magic.default magic (もし magic がなければ)
# cp mime.types.default mime.types (もし mime.types がなければ)
# cp srm.conf.default srm.conf (もし srm.conf がなければ)
# /usr/local/etc/rc.d/apache.sh start
```

図 4.2: apache のインストール

netscape で http://comx2??/ にアクセスし、インディアンの羽根の絵のついたページが表示されれば成功である。

4.1.5 ホームページを立ち上げる

一般ユーザ (user2??) でログインし、実際にホームページを作成する。

1. ホームディレクトリに public.html というディレクトリを作る。
2. その下に index.html という HTML ファイルを置く。HTML ファイルの記述方法はコラム「HTML」を参照のこと。
3. netscape で http://comx2??/~user2??/index.html にアクセスしてみる。
4. 他班のページにもアクセスしてみる。

こののち、課題 2、課題 3 を行う。

4.2 インターネットプロトコルの仕組み

HTTP は TCP/IP (Transmission Control Protocol/Internet Protocol) と呼ばれるインターネット標準プロトコルの上に構成された通信プロトコルであり、そのやり取りの様子を telnet という TCP/IP のための汎用通信コマンドで手動で模倣することができる。HTTP の場合は、“GET” というコマンドを用いて WWW サーバ上のコンテンツを取得することができる。

自班の WWW サーバに対して図 4.3 のように telnet で接続し、通信の様子を模倣してみよ。こののち、課題 4 を行う。

HTML

HTML(HyperText Markup Language) は WWW のコンテンツを記述するためのもっとも標準的なマークアップ言語である。細かい文法は国際規格によって定められているが、おおむね次のように記述する。

```
<html>
  <head>
    <title>ここにタイトルを書く</title>
  </head>
  <body>
    <h1>大見出し</h1>
    <p>本文はここに書く。</p>
    <ul>
      <li>LaTeX の itemize 環境に相当する書き方</li>
    </ul>
    <ol>
      <li>LaTeX の enumerate 環境に相当する書き方</li>
    </ol>
  </body>
</html>
```

```
% telnet localhost 80
(メッセージが表示され、入力待ち状態になる)
GET /index.html
(http://localhost/index.html の内容が取得できる)
%
```

図 4.3: telnet による WWW サーバとの接続

4.3 CGI

プログラムで実行した結果を WWW ページに反映させる手段がいくつか用意されているが、そのうちもっとも普及しているものがサーバの CGI (Common Gateway Interface) 機能を利用する方法である。

CGI を利用する場合、WWW サーバに設定を行う必要のある場合がある。今回利用する Apache の標準設定では、`/usr/local/www/cgi-bin/` の下に CGI スクリプトを置くことで CGI を利用することができる³。たとえば、`test` という CGI スクリプトをおいたとしたら、その URL は `http://???/cgi-bin/test` となる。

もっとも簡単な CGI スクリプトをリスト 3 に示す。これは現在の日付・時間を表示するものである。

リスト 3: 現在の日付・時間を出力する CGI スクリプト

```
#!/bin/sh

echo "Content-Type: text/html"
echo ""

echo "<html>"
echo "<head>"
echo "<title>test</title>"
echo "</head>"
echo "<body>"
date
echo "</body>"
echo "</html>"
```

この例からわかるように、CGI スクリプトは以下の条件を満たすようにプログラミングすればよい。

- 最初に `"Content-Type: text/html"`、ついで一行空行を出力する。
- そのあと、出力結果が HTML ファイルとなるように出力を行う。

この条件さえ満たせば、C で書いても Perl で書いてもシェルスクリプトで書いてもよい⁴。

CGI について課題 5、課題 6(発展課題)を用意してあるので、行うこと。

課題

課題 0

実験の過程で直面したトラブルについて、その症状、原因の究明過程、原因、解決方法などについて(自分で調べるなどして)記述せよ。

³一般ユーザの `public_html` ディレクトリなどに CGI を置いて利用する方法に興味がある人は別途質問されたい。

⁴世の中で Perl が CGI スクリプト言語の主流になっているのは、移植性が高いという理由でしかない。

課題 1

自分でいくつかの WWW ページを選び、w3m と netscape で表示結果を見比べよ。表示結果があまり異ならないページと大きく異なるページを挙げ、その理由について、HTML の整形 (レンダリング) という観点から理由を調べよ。

課題 2

WWW サーバのアクセス記録 (ログ) が `/var/log/httpd-access.log` というファイルに残っているはずである。この中身を書き記し、記述されている内容について考察せよ。4.2 節の内容も参考のこと。(参考資料: `/usr/local/share/doc/apache/logs.html`, <http://httpd.apache.org/docs/logs.html>)

課題 3

`comx200:/pub/access.log` は、ある週の言語・ソフトウェア工学研究室の WWW サーバのアクセス記録である。このログについて、以下の事柄を UNIX 標準コマンドのみを用いて調べよ。調べた手順も書け。

1. 全アクセス数
2. ユーザ `kunishi` の公開しているファイルへのアクセス数

課題 4

図 4.3 での入力待ち状態でのメッセージを書き記せ。また、上記の方法で、4.1.5 節で作成したページの内容を取得し、実際のファイルの内容と比較してみよ。

課題 5

リスト 3 のスクリプトを `currentdate.cgi` というファイルとして作成し、パーミッションを 755 にして `/usr/local/www/cgi-bin/` に置き、`http://localhost/cgi-bin/currentdate.cgi` にアクセスして CGI が動作していることを確かめよ。

そののち、CGI プログラムは WWW ブラウザに対してどのような出力を渡すか、作成した CGI プログラムと 4.2 節で示した `telnet` を用いる方法で得られた出力結果を示し、考察せよ。

課題 6

CGI プログラムを C で書いてみて、実行してみよ。たとえば「簡易版カウンタ」を書いてみよ。カウンタ数はファイル `count` 中に保存することとし、CGI スクリプトにアクセスしたら、`count` から読み込んできた値に 1 加えた値を HTML として出力し、その値を `count` に書き戻す。(発展課題。時間がなければ、省略してよい。)

メール

メールは、一般にバケツリレー方式で配送が行われる。すなわち、インターネット上にある複数の計算機を次々とメールが配送されていく^a。この配送は、それぞれの計算機で動作している MTA(Mail Transfer Agent) と呼ばれる daemon プログラム^b によって自動的に行われる。一方、ユーザが Mew でメールを送信する場合を考える。このとき、Mew はユーザが作成したメッセージを適当に再整形し、MTA に渡す。受け取った MTA が、上記のバケツリレーを行って目的の計算機まで配送するわけである。ここでいう Mew のように、ユーザとのインタフェースを担うメール読み書きプログラムのことを MUA(Mail User Agent) と呼ぶ。MTA によるメールのバケツリレーの記録は、各メールのヘッダ中の”Received:”フィールドに残っている。自分宛てのメールについて、確認してみるとよい。メールを配送する際に利用されるプロトコルが SMTP (Simple Mail Transfer Protocol)、メールを受信する際によく使われるプロトコルが POP (Post Office Protocol) であり、ともに TCP/IP の上に実装されている。

^a最近では DNS(Domain Name Service) というホスト情報データベースを参照することで直接目的の計算機に配送されることも多いが、バケツリレーの段数が 1 段になったものと理解しておいてほしい。

^bUNIX で、ユーザの利用の如何に関わらず、常に走っているプログラム。

第5章 UNIX 内部

5.1 小規模アプリケーション: cat

ls や cat など、標準的に UNIX 環境で用意されているコマンドも C 言語で書かれている。ソースファイルが /usr/src の下にあるので、ls や cat など、普段使い慣れているコマンドのソースを実際にコンパイルしたり、内容を読んだりする。

5.1.1 cat

ディレクトリ /usr/src/bin/cat/ の下にある。C 言語のソースは一つだけなので、中身を簡単に眺めた後、図 5.1 のようにコンパイルを試みる。

```
# cc -o cat cat.c
```

図 5.1: cat のコンパイル

こののち、課題 1 を行う。

5.1.2 echo

ディレクトリ /usr/src/bin/echo/ の下にある。C 言語のソースは一つだけ (リスト 4) である。このコマンドに関して、課題 2 を行う。

リスト 4: echo.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int
main(argc, argv)
    int argc;
    char *argv[];
{
    int nflag;

    /* This utility may NOT do getopt(3) option parsing. */
```

```

    if (++argv && !strcmp(*argv, "-n")) {
        ++argv;
        nflag = 1;
    }
    else
        nflag = 0;

    while (argv[0]) {
        int len = strlen(argv[0]);

        if (len >= 2 && !argv[1]
            && argv[0][len - 2] == '\\\''
            && argv[0][len - 1] == 'c') {
            argv[0][len - 2] = '\\0';
            nflag = 1;
        }
        (void)printf("%s", argv[0]);
        if (++argv)
            putchar(' ');
    }
    if (!nflag)
        putchar('\n');
    exit(0);
}

```

5.2 大規模アプリケーション

ここまで見てきたのはソースファイルの数が少ないものばかりであったが、より大規模なアプリケーションになると、ソースファイルの数もひじょうに増え、コンパイル作業もたいへんになる。そのため、make など、コンパイル作業を補助するツールが必要になってくる。

大規模アプリケーションに関して、課題 3・課題 4 を行う。

5.3 OS を壊す実験

最後の実験として、root で”rm -rf /”を実行する実験を行う。

UNIX のファイルシステムは/からはじまる木構造になっている。また、root は強力な権限を持つユーザであり、すべてのファイルやディレクトリを変更・上書き・消去できる権限を持つ。一方、UNIX では、今実行しようとする rm や、各種 daemon プログラム、また UNIX の実行の核であるカーネル(kernel) とよばれる実行ファイルもすべて、一つのファイルとして存在している。

したがって、この実験の目的は、「UNIX で、現在実行中のプログラムのファイルを消去しようとするとなんが起るか確かめる」と言い換えることができる。

作業手順を実験時に指示するので、それにしたがって作業を進めること。また、課題を必ず参照すること。

5.4 後始末

これで実験は終了である。次のグループの実験のために、組み立てた PC を元の状態に分解し、片付ける。部品を壊さないよう、丁寧に扱うこと。また、元の箱・袋に入っているか、ケースのネジの締め忘れがないか、などにも注意しておく。

課題

課題 0

実験の過程で直面したトラブルについて、その症状、原因の究明過程、原因、解決方法などについて (自分で調べるなどして) 記述せよ。

課題 1

`/usr/src/bin/cat` の下でコンパイルして作った実行形式ファイル `cat` が `/bin/cat` と同じ振舞いをするか、具体例を用いて確かめよ。確かめた方法も記述せよ。

課題 2

`echo` というコマンド (`/usr/bin/echo`) について、以下の課題を行え。

1. 実際にコマンドを実行させ、その振舞いを調べよ。
2. リスト 4 の C 言語プログラムを解析し、どういう動作をするコマンドか述べよ。
3. オプションを付けたときの振舞いについて、C 言語プログラムの解析結果をもとにコマンドを実行させ、振舞いを調べよ。

適宜オンラインマニュアルを参照すること。なお、Solaris では `/usr/ucb/echo` がほぼ同じ動作をする。

課題 3

`/usr/src/bin/lis` にて `root` で `make` に関して次のような操作を行い¹、その振舞いを述べよ。

1. `make`
2. `make clean`
3. `make`
4. `rm lis; make`

¹できあがった実行形式ファイルは `lis` である。実際に実行して、振舞いを確認してみるとよい。

5. `rm ls.o; make`

6. `touch util.c; make` (`touch` が何をするコマンドか調べてレポートに記載すること)

上記の結果より、`make` が何を行うコマンドなのかまとめよ。

課題 4

`emacs` のソース (`emacs-20.7.tar.gz`) を展開し、この中に `.c` ファイル、`.h` ファイルがそれぞれいくつ含まれているか調べよ。調べた方法も記載すること。

課題 5

「OS を壊す実験」について、以下の内容をまとめ、報告せよ。

1. 実験結果の予想。かならず実験を始める前に各自で結果の予想を立てよ。
2. 実験手順の報告。`rm` を実行中に試したことがあれば、それについても述べること。
3. 実験結果の報告。
4. 実験結果の考察。予想と合っていたかどうか、なぜこのような結果が出て、自分の予想はどこが違ったのか、など。

実際に何が起こるか、結果を記した WWW ページがある (URL は演習中に指示)。このページを見たり、オペレーティングシステムに関する書籍を参照するなどして、まとめること。

課題 6

来年度以降の参考にしますので、今回の実験を通しての感想を書いてください。

参考文献

1. <http://www.tkcitey.net/~nobusan/hardware/boot.hdd/index.html>
2. <http://www.sra.co.jp/people/katsu/doc/rmrf/>