

## マルチ OS-PC システムの構築

大矢健一\*・楡井雅巳\*・西村 治\*・村田雅彦\*\*・山下 威\*\*

岡島英男\*\*\*・中澤達夫\*\*\*・堀内征治\*\*\*\*

(平成8年10月31日 受理)

## Construction of Multi OS Personal Computer System

By Ken'ich OHYA, Masami NIREI, Osamu NISHIMURA, Masahiko MURATA,  
Takeshi YAMASHITA, Hideo OKAJIMA, Tatsuo NAKAZAWA, Seiji HORIUCHI

## 1. はじめに

電子情報工学科ではこれまでの約6年間に渡り、学科設立時に導入されたMS-DOSをOSとするパーソナルコンピュータ(IBM/PS-55)システムを用いて情報処理教育を行ってきた。我々は、とくに高学年の学生についてはUNIXの環境を中心として教育を行う方針を打ち出しており、これを実現するためにパソコンをキャラクタ端末として用い、サーバとなるUNIXワークステーションにシリアル回線のターミナルサーバを介して結んで使用していた。しかし、このシステムでは、例えばUNIXの大きな特徴の一つであるX Window Systemも利用できず、また、WWWを中心とするインターネットの利用にも大きな制限があるなど、今となっては非常に旧式なものとなっていた。このため、機器のリプレースを要望してきたところ、昨年度後半になって予算が認められ、新しいシステムの導入が可能となった。

現在の一般的なPCの代表的な仕様はWindows95を基本ソフトとして使用するものであろう。この最新のOSを利用すれば、ネットワークへの接続が従来に比べて簡単であるなどの利点がある。しかし、既に述べたように我々の学科の主要な教育環境としてUNIXを利用する方針であることから、PC用UNIXをOSとするシステム構築を行うことを前提にして仕様を考えた。しかし、UNIXは低学年の学生には必ずしも取っ付き易いものではないことや、これまでに購入した(ライセンスを持っている)DOS用のアプリケーションソフト資源を今後も有効に利用することも考えて、1台のPCに複数のOSを載せたマルチOS-PCシステムを構築することを計画した。

このシステムの概要については、既に他で報告<sup>1)</sup>しているが、本報告では今後同様のシステム構築を行う際に参考になるよう考慮して、システム構築の実際についてできるだけ詳細に記述した。

## 2. システムの構築

## 2-1 導入したシステムの仕様

本システムでは、以下のような環境が要求される。

\* 電子情報工学科助手, \*\* 電子情報工学科技官,  
\*\*\* 電子情報工学科助教授, \*\*\*\* 電子情報工学科教授

- (1) UNIX が利用できること。
  - (2) 従来利用していた DOS ベースのアプリケーションが利用できること。
  - (3) 1 年生が利用している情報教育センタと互換性のある Windows が利用できること。
- 1、2 についてはカリキュラムの関係から必須であり、3 については 2 の代替案であり、また 1 年生時に利用する環境との継続性からの要求であった。

最終的に決定した仕様 (機器) を表に示す。

導入された機器		
PC (44 台)	CPU:	Pentium 100MHz
	メモリ:	16MB
	HDD:	850MB
キーボード	ASCII 配列	
マウス	3 ボタン	
ディスプレイ	17 インチカラー	
ハブ (2 台)	10BaseT, 24 ポート	
ソフトウェア	FreeBSD	
	Windows95	
	Visual BASIC	

PC 本体は、最近の PC の例に漏れず Windows95 がプリインストールされた機種である。しかし、既に述べたように中心として使用する OS である UNIX に Free Software である FreeBSD を採用するため、OS のインストールを初めからやり直す必要が生じた。ここでは、boot manager を利用して 3 種類の OS (UNIX, MS-DOS, Windows95) を同時に載せ、起動時に必要とする OS を選択してブートするシステムを実現した。今回のシステムのその他の特徴として、ディスプレイを 17 インチのものにしたことが挙げられる。これは、コスト的にはかなり高むことになったが、UNIX の X Window System をある程度快適に利用するために必要であると判断した。

以下、導入されたハードウェアを実際に教育に使用するために行ったシステム構築について詳細に述べる。

## 2-2 3 種類のシステムの共存

DOS + UNIX、UNIX + Windows 3.1 あるいは Windows3.1 + Windows95 など、2 種類の OS を共存させたシステムはこれまでも例が挙げられているが、DOS と UNIX および Windows95 の 3 種類の OS の共存については適切な資料が見当たらず、ネットニュースなどを参考にしながら、手探りでの環境構築となった。

今回導入した OS は、これまで利用している (ライセンスを持っている) IBM DOS J5.0/V、新規導入の PC にプレインストールされている Windows95、および、PC-UNIX (FreeBSD 2.1.0R) である。今回 FreeBSD の導入を選択したのは、無償で入手できる (FreeSoftware) ソフトである点と、システム管理が比較的やりやすいという点とを考慮したためである。

### 2-3 ハードウェアとの関連

まず PC-UNIX を導入する時に問題になるのは、ハードウェアの仕様である。どのハードウェアが実際に使用可能かという自由度は、導入する UNIX システムによって異なり、それぞれのバージョンなどによって確認が必要である。

一般に PC-UNIX の導入時に生じる問題点は、ネットワークボード、グラフィックボードとマウスに関連していることがほとんどである。我々が導入することになった PC は、Windows95 をプレインストールして販売されるごく一般的なタイプのものであり、ネットワークボードはオプションであった。ネットワークボードについては後述するが、初めに業者より提案されたボードは、結局今回の環境では正常に動作せず、利用できなかった。

次に DOS と Windows95 の共存についても、ハードウェアの制約がある。DOS と Windows95 は、共にブートドライブが C である必要があり、インストール時に注意が必要である。

さらにマニュアル類の入手の問題がある。従来 IBM クローン機は、ハードウェアの情報を全て公開しており、購入時には個々の部品の仕様書、取り扱い説明書が添付されるのが一般的である。しかし最近の国産 PC (いわゆる DOS/V 機) は、購入時にはハードウェアに関するマニュアルが添付されておらず、メーカーに直接問い合わせてもその情報を手に入れることが困難である。我々の場合にもマニュアル類の入手にかなりの時間を有し、その間に手探りで不必要な作業を強いられ、多大な時間を浪費する結果となった。

### 2-4 3つの OS のインストール

OS のインストールは、基本的には個々のインストール手順に従って行うことになるが、先に述べたように、複数の OS が共存することによって生じる問題がある。ここでは、問題点を挙げながらインストール方法を紹介する。

**2-4-1 ディスク領域の分割** PC-UNIX など複数の OS の導入時には、ハードディスクを OS 毎に領域を分割する。既に DOS や Windows95 などを利用している場合には、"derag"、"fips"といったユーティリティによって分割を行うが、今回は初期状態からのインストールであったので、ブートマネージャによって領域の分割を行った。DOS、Windows95、FreeBSSD それぞれの領域のサイズは、50M、200M、560M バイトとした。DOS 領域が極端に小さいのは、今後新しいアプリケーションの導入が見込まれず、現状の環境のみで良いと判断したためである。ハードディスクをフォーマットした後、ブートマネージャをインストールする。ブートマネージャは、DOS と Windows95 を共存させるため、仮想的に2つの C ドライブを扱えるものである。D ドライブは FreeBSD の領域とした。ハードディスクのフォーマットは、BIOS がフォーマットなどのハードディスクユーティリティを提供していない簡易版であったため、ブートマネージャで全領域を削除することで領域のクリアをし、実際にはフォーマットは行っていない。

**2-4-2 DOS のインストール** ブートマネージャによって、DOS 用の C ドライブをアクティブにした後、通常に DOS をインストールする。DOS 用アプリケーション群は、所有していた他の PC のハードディスク上に環境を整備し、それを slave のディスクとして認識させ、内容をコピーした。DOS 用アプリケーションは、従来から利用してきたソフトウェア (エディタ、C コンパイラ、BASIC など) である。

**2-4-3 Windows95 のインストール** ブートマネージャによって、Windows95 用の C ドライブをアクティブにし、通常に CD-ROM よりインストールした。インストール後、プラグ・アンド・プレイ (PAP) の機能は停止させた。これは、我々の過去の経験から PC-UNIX と Windows などの PAP 機能との相性が良くないと判断しているためである。事実 PAP 機能が働いている場合には、実際には使用されていない 300h の I/O アドレスがフックされてしまったり、ネットワークボードの I/O アドレスが 260h などに割り付けられてしまい、FreeBSD の導入の妨げとなった。Windows95 のアプリケーションとしては、新たに購入した Visual BASIC をインストールした。

**2-4-4 FreeBSD のインストール** FreeBSD のインストールは、PC 付属の CD-ROM から行なった。PC 付属の CD-ROM は、E-ID の CD-ROM であり、ATAPI 対応のブートイメージを使うことによってインストールができた。CD-ROM からのインストールとした理由には、ネットワークボードの問題もあった。業者より提案されたネットワークボードは Fujitsu のチップを使ったものであり、FreeBSD 2.1.0R のブートイメージではサポートされておらず、認識できないという問題があり、ネットワークインストールが不可能であった。幸いにも、我々の一人が FreeBSD が収納されている CD を所有していたため、それを利用することができた。ブートイメージのロード後、Partition の設定などを行なう。今回のインストールでハードディスクに関連した問題も生じている。FreeBSD 側のハードディスクの geometry の認識が誤っているというものであった。このまま作業を進めると、これまでにインストールした DOS や Windows95 の領域までもが破壊されてしまう。通常は FreeBSD が自動的に認識する geometry で問題はないのであるが、今回はセクタ数が実際の半分で認識され、シリンダ数が 2 倍の数で認識されていた。これを確認するためにはフォーマットユーティリティが必要であったが、BIOS がこの機能を提供していないため、ハードディスクを外し、所有していた他の PC でフォーマットを行ない確認を行なった。もしこの時点でマニュアル類が提供されていれば、すぐに確認が可能であったであろう。また、ハードディスクの仕様に関連するかは不明であるが、デバイスの生成時に、必ず 1 度はデバイスの生成に失敗する現象が現れている。

geometry を正しく修正し、デバイス生成が行われた後は、ソフトウェアのインストールを行う。インストール内容は、システム、カーネルのソース、X Window System である。カーネルのソースは、ネットワークボードやマウスを認識させるため、カーネルの再構築を行なうために必要である。X Window System のアプリケーション群は、NFS マウントして利用することを構想していたため、個々にインストールされる必要がなく、ここでは導入していない。

インストールの完了後、カーネルの再構築を行う。再構築の対象は、ネットワークボードおよびマウスである。ネットワークボードは Fujitsu のチップを使った ISA バスのものであり、マウスは Logitech の PS/2 マウスである。それぞれソースに含まれる LINT 内の fe0 と psm0 のエントリを使うことで認識できた。しかしネットワークボードについては、認識はするもののハード的にバスの速度に追従できず、タイムアウトが生じた。このネットワークボードの現象は全体の約 9 割に確認されたため、ボードを 3Com 3C509B のものに全数交換を行なうことにより改善した。3Com のネットワークボードは、FreeBSD 標準の ed0 のデバイスで認識できている。

X Window System は、S3 torio64 のグラフィックアクセラレータに、gendac のチップセットを選択することで、1280X1024 の解像度が得られた。本システムでは、1280x1024 では小さ過ぎると判断し、1024X768 とした。一般的にワークステーションなどで使われている 1054X900 の解像度はディスプレイ側でサポートされておらず、実現できなかった。解像度についても、ディスプレイのマニュアルが入手されたことで確認できたことである。

## 2-5 本システムの利用方法

以上述べたように、本システムは 3 つの OS を導入したマルチ OS-PC システムである。本システムでは、電源投入後ブートマネージャによる OS の選択画面が現れ、利用者が必要なシステムを起動できるようになっている。利用者は、希望する OS を自由に選択できるが、DOS 以外はソフト的にシステムを終了させる作業が必要であり、適切な手順で終了させずに電源を切ってしまうとシステムの破壊の危険がある。

## 3. UNIX システムの構築

PC に FreeBSD をインストールした後は、実際にそれを学生が使えるようにセットアップする必要がある。この節ではその方法について具体的に述べる。

セットアップ作業は想像していた以上にハードなものであった。朝から深夜 2~3 時までの作業が約 3 週間にわたって続いた。今後の同様な作業の軽減に本稿が役立てば幸いである。

### 3-1 どのようなシステムにすべきか

入札により機種が決定した後、実際に PC が長野高専に持ち込まれるのは 3 月末になる。4 月の新学期までの準備期間は実質一週間しか見込めないため、実際に PC が長野高専に納入される前に決められることはなるべく決めておく必要があった。

3 月は担当教官たちの出張などが多い時期であり、システム構築の相談のために実際に集まって会議を持つことは難しいと考えられるため、ネットワークを利用して打合せができるように、新 PC システムについての議論のための `new-pc@ei.nagano-nct.ac.jp` というメーリングリストを新たに作成した。このメーリングリストの構成員は電子情報工学科のメンバーにより構成される `all-ei@ei.nagano-nct.ac.jp` と全く同じであったが、名称を変えておくことにより、新 PC システム専用の議論の場であることが明確に意識できるという利点がある。このメーリングリストを作成したのが 3 月 11 日 (月) のことであった。

メーリングリストにおいて、まず最初に議論すべき以下のようなテーマが提示された。

- 二階実習室をどのように使うか (従来の X 端末の位置付け、既存 PC をどうするか)
- 新 PC をどのようなシステムにするか
  - Windows95 に何 MB のパーティションを割り当てるか
  - PC-UNIX に何をを用いるか
  - NFS サーバ機はどうするか、何台あればよいか、ミラーリングはどうするか
  - バックアップはどうするか
  - NIS をどうするか
  - どのようなツールが必要か
  - どのような手順でインストールしていけばよいか

－ X のウィンドウマネージャは何を用いるか

- 教育内容と方法 (ソフト系実習の見直し)

これを見てわかるように、システム構築の面から見て未定事項が多いのは PC-UNIX に関することである。PC-UNIX はカスタマイズが非常に柔軟に行なえるため、学生用の環境を構築しようとする場合、特に一つの基準と言えるようなものを構築しようとする場合には、多くの選択肢の中から絞りこんでいくという作業が発生する。

### 3-2 PC-UNIX の選択

コンピュータ教育を UNIX で行なう利点は計りしれない。代表的なものを列挙すると以下のようになる。

- 各種言語による開発環境がフリーで構築可能
- さまざまな有用なツールがフリーで入手可能
- OS がユーザ領域とシステム領域とに分けられているため、ユーザがうっかり誤操作等をして OS が落ちる心配がほとんどない
- OS そのもののソースを見ることさえ可能 (フリーの PC-UNIX の場合) であるため、OS そのものに対する疑問点を解決することが可能
- 強力なウィンドウシステムである X Window System が動作

以上のような利点のある UNIX であるが、従来は比較的高価なワークステーションでのみ動作する OS であったが、近年の PC の性能向上により PC 上でも何ら問題なく動作するようになった。

フリーの PC-UNIX として代表的なものは、Linux と FreeBSD である。どちらを載せたらいいのか悩みは尽きなかったが、以下のような理由により今回は FreeBSD を選択した。

- 従来、電子情報工学科では学生用の UNIX マシンに SunOS が使用されていたが、FreeBSD はこの SunOS より近かった
- Linux はちょうどバージョンアップになった時期であり、日本語パッケージである JE がまだリリースされていなかった

### 3-3 UNIX を使う上での PC のハード的な仕様

3 年生以上の学生が PC を使うときには UNIX をメインに使うことを念頭におき、PC のハード的な仕様を以下に示すようにそれを意識したものにした。

- 17 インチディスプレイ (トリニトロン管)
- X Window System が 1024x768 のサイズで動作すること
- マウスは 3 ボタンのもの
- Ethernet

### 3-4 PC-UNIX ネットワーク設計

PC44 台によるシステムを設計する上で何が最重要かを熟考し、とくに以下の点に留意してシステム構築を行っていくことにした。

- どのマシンに login しても、メールの読み書きなども含め、全く同等に使えること
- シャットダウン方法が容易であること

- 学生が従来使用していた UNIX システムと違和感がないように使用可能なこと
- 管理者側からは、ツールのアップデートなどが容易なこと

これを満たすために以下のような設計方針にした。

- NIS によるネットワーク管理
- NFS によるネットワーク管理
- 容易なシャットダウン方法の実現
- 各種ツールの徹底的な初期設定

### 3-5 NIS によるネットワーク管理

数十台の UNIX システムにおいては、さまざまな情報をネットワーク的に共有化するためのツールとして、NIS (Network Information System) というものが通常利用される。NIS を用いることにより、各ホストの情報やユーザ情報のパスワード管理などをネットワーク的に一元的に行なえるようになる。

NIS によるシステムは、一台以上の NIS サーバ機とそれ以外のクライアントから構成される。よって、NIS を用いるためには、まず NIS サーバ機を構築する必要がある。電子情報工学科内ではそれまで NIS を使用していなかったため、全く新たに NIS サーバ機を構築する必要があった。

当初は 44 台のうちの 1~2 台を NIS や NFS のサーバ機にしようとしていたが、44 台全部が Dos や Windows95 などに用いられるときに不都合が生じることに気づいた。すなわち、全台数が Dos や Windows95 などに用いられているとき自体には NIS や NFS のサーバ機は必要ないのであるが、Dos や Windows95 を終えた後にどれかのマシンを UNIX として立ち上げる瞬間にサーバ機が存在しなくなってしまう。

よって、NIS サーバを既存ワークステーション上に構築する必要があった。従来学生用ワークステーションとして稼働していたマシン (Sun IPX, 64MB) を NIS サーバ機にあてることとした。

NIS はブロードキャストを利用するために、NIS サーバとクライアントとが同じサブネットに属する必要がある。当時、ATM LAN が稼働しようとしており、新 PC システムは ATM LAN の方に接続したかったが、従来学生用ワークステーションは既存 LAN 上にあったため、新 PC システムも既存 LAN の方に接続することにした。

従来学生用ワークステーションは 3 台ありそのうち 2 台が Sun のワークステーションであった。一台を NIS のサーバ、もう一台を NIS のスレーブサーバとした。スレーブサーバとは、何らかの理由でサーバが機能していないときにサーバの機能をするものである。

NIS のサーバとスレーブサーバの設定作業は、月刊誌「UNIX Magazine」<sup>2)</sup>の設定例を参考にして行なったが、これは古い記述であったためにこのままでは動作しなかった。そこで Sun のシステム管理のマニュアル<sup>3)</sup>を参考にして、ようやく NIS のサーバとスレーブサーバが動作することとなった。

さて、NIS の設定をする際に気づいたことであるが、`/etc/passwd` の共有化のために、学生がどのマシンに login してもホームディレクトリのパス名が同じである必要があった。このため、学生と教職員のホームディレクトリを、

```
学生    /home/Students
教職員  /home/Teachers
```

として統一した。すなわち、/home に全て統一するということにより各マシンにおける /home はマウントポイントへの単なるリンクとなる。PC 各マシンに FreeBSD がインストールされる際に、システム領域以外のディレクトリは通常 home という名称になるが、それを /home0 とし、PC 各マシンの /home が指す実体は、NFS サーバのディレクトリとした。そこに全てのユーザのホームディレクトリが置かれることになる。

さて、実際に NIS を用いてホスト情報やユーザ情報を設定し PC 側から login しようとするが、login することができなかった。FreeBSD では /etc/passwd における暗号化されたパスワードをセキュリティの点から見えないようにしているが、そのために NIS がうまく動作しないものと最初は思われた。

過去の NetNews<sup>5)</sup>などをしばらく調べているうちに、SunOS と FreeBSD ではユーザのパスワード情報の暗号化が異なることがわかった。すなわち、SunOS では DES を用いているのに対し、FreeBSD では MD5 を用いているのであった。この問題に対してはすでに同様なことをしている人が多くいたようであり、NetNews<sup>5)</sup>などに解決法がすでに記されていた。すなわち、FreeBSD の方で DES を用いるようにすればよいのである。FreeBSD の DES のライブラリも公開されており ftp で入手可能なのであるが、規制のためにアメリカからは ftp できないという注意点がある。よって、アメリカ以外の ftp サイトから ftp して取ってくればよいのであった。FreeBSD に DES をインストールする際に注意する点が他にもあり、パスワードの暗号化が変わってしまうために、従来のパスワードが無効になってしまうということである。すなわち、ルートのパスワードを解除しないで DES をインストールするとルートで login できないので大変なことになる。実は白状すると筆者はこの大変な事態に遭遇してしまったのが、FreeBSD をシングルユーザモードで起動することによりなんとか危機から脱出することができた。DES のインストール自体は以下のようにして行なえた。

```
cat des.aa | tar --unlink -xpfz - -C /
```

その後、NIS による基本的な機能の動作確認をし、それから細かな設定に入っていった。基本的な機能の動作確認における問題点として、/etc/services の共有がうまく動作しないというのがあった。いろいろ調べはしたものの結局原因がわからず、このファイルについては残念ながら NIS を用いるのではなく各 PC においてローカルに持つこととした。

次の問題点として、NIS における netgroup がうまく機能しないということがあった。/etc/hosts.equiv などのファイルにおいて各 PC を一つの netgroup として名称を付けて NIS により管理しようとしていたのであるが、“ワークステーション ⇄ FreeBSD ⇄ FreeBSD” という三者間における rsh, rcp が同時にはうまくいかなかった。ワークステーションや PC それぞれにおいて、/etc/host.conf における hosts, bind, nis の優先順位があり、そのために三者間で同時にはうまくいかないのだろうか、同時にうまくいく解はないものか、などと考えた。論理的に考えてうまくいくと思われる方法はどれもうまくいかなかったため、結局、あらゆる全ての組み合わせを試すこととなったが、それもうまくは

いかなかった。この作業には 2 日間ほど要した。/etc/hosts.equiv には結局全 PC のホスト名をそのまま記述することとした。

### 3-6 NFS によるネットワーク管理

システムの構築環境として、まず、NFS サーバ用 PC 一台 (以降、親機) と NFS クライアント用 PC 一台 (以降、子機) から成る PC 一対のモデルを構築した。この最小モデルがネットワーク的に問題なく動作すれば、あとは親機の内容を最終的な NFS サーバであるワークステーション側に移動し、子機の内容を全 PC に次々とコピーすれば PC44 台のシステムができあがる。

さて、NFS であるが、PC44 台ものクライアントを経験したことがなかったため、ネットワークのトラフィックがどのくらいになるか心配であった。とはいえ、管理する側から考えればこれしか方策はなかった。実際に稼働させて不具合があれば、NFS サーバを複数にすることで解決を計ろうと考えていた。複数の NFS サーバを用いるときにはその複数のサーバ同士の整合性を取るためのミラーリングなどが必要となり、これはこれでまた大変なことなのであった。

NFS を用いるにあたって、具体的にサーバ機をどうするか、という問題が出た。ネットワーク設計当初は、あまっている PC (Pentium 75MHz) を NFS サーバ機に仕立てようと思っていたが、よく考慮し直すと NFS サーバ機としてはかなり苛酷な条件であることに気づき、それまで学生用ワークステーションとして稼働していたマシン (Sun IPX, 64MB) を NFS サーバ機にあてることとした。1 Sun と FreeBSD とはバイナリが異なるため、ワークステーション側には、親機で作成した NFS サーバ用のファイルを置くだけとなる。

各 PC からは /etc/fstab により以下のような NFS リンクをはることにした。これにより、ワークステーション側の 4 つのパーティションを後々まで自由に使えることとなる。

PC	Sun (ホスト名:wakame)
/mnt/wakame/home1	/home1
/mnt/wakame/home2	/home2
/mnt/wakame/home3	/home3
/mnt/wakame/home4	/home4

次に、各 PC におけるファイルのリンクを以下に示す。

リンク元	リンク先
/home	/mnt/wakame/home2/home
/home0	/usr/home0
/tmp	/usr/tmp
/var	/usr/var
/usr/Tara	/mnt/wakame/home1/Tara
/usr/X11	/usr/X11R6
/usr/X11R6	/usr/Tara/X11R6
/usr/local	/usr/Tara/local
/usr/share	/usr/Tara/share

以上で示されているように、変更が比較的頻繁に行なわれることが予想される /usr/X11R6、/usr/local、/usr/share は全てワークステーション側の /usr/Tara 内に実体が置かれることとなる。これにより有用なツールのアップデートなどが容易に行なえる。

/usr/X11 というリンクを作成しているのは、Netscape Navigator がこのディレクトリの存在を前提としているためである。

また、/tmp や /var の 2 つのディレクトリは PC 内のより広いパーティションである /usr へ移動させた。

### 3-7 容易なシャットダウン方法の実現

PC に PC-UNIX をインストールした場合、終了時にシャットダウンを行なう必要が生じる。通常、シャットダウンはルートに su してから行われるが、学生に su させるわけにもいかず、かといってセキュリティに穴が開かないような工夫が必要となる。

試行錯誤の結果、今回の設定では、NIS によるアカウントにシャットダウン専用のアカウントを作成し、そのアカウントで login するとシャットダウンが行われるようにした。NIS を用いているためにシャットダウン専用アカウントのパスワード変更などが容易に行なえるという利点がある。

とはいえ、NIS のマスターファイルはワークステーション側にあるため、ワークステーション側のファイル起動によって PC 側のルート権限を実行することになる。

### 3-8 各種ツールの徹底的な初期設定

学生が通常用いる各種のツールの中には、ある程度管理者側で初期設定が必要なものがある。学生が最初に触れる環境を構築するのだという重要性を考慮すると一種の規範となるべき環境を構築する必要があり、そのために模索の中で行われたこの初期設定作業は極めて時間がかかるものであった。

**3-8-1 X Window System** FreeBSD 2.1.0R における X Window System は X11R6 であり、XFree86 3.1.2 というバージョンであった。

X サーバにおいては、VRAM が 2MB あったために 1 ピクセルに 16 ビット用いることが可能であった。Sun などの通常のワークステーションでは通常 8 ビットであるため、ワークステーションよりも色数が多いという快適な環境となった。サーバのサイズは 1024x768 である。フォントもいくつか追加し、/etc/XF86Config に加えた。

ウィンドウマネージャは fvwm 1.21z を採用した。fvwm は、X11R4 から X の代表的なウィンドウマネージャとなった twm および Motif 環境下において代表的なウィンドウマネージャである mwm の上位互換とも言えるため、今回はこれを採用した。1.21z というバージョンを採用した理由であるが、このバージョンを境として fvwm は見栄えを重視した多機能化に進んでしまったように思えたからである。1.21z というバージョンでウィンドウマネージャとして重要な機能は全て備えているように思えた。

初期設定としてタイトルバー内のボタンは左に 1 つ右に 3 つ配置し、mwm と Windows95 との上位互換のようにした。これは、Windows95 に慣れている多くの学生に配慮した結果である。

ウィンドウフォーカスの際のタイトルバーの色の変化は、薄いグレイから濃いグレイへの

変化というように地味な配色にした。仮想画面の数はやや多いかもしれないが6とした。アイコンボックスは右端の縦の領域とした。各ウィンドウの枠の幅は5とした。ウィンドウのオートレイズを採用し動作までの時間を0.5秒とした。タイトルバーについては、fvwmのページャやbiffなどはそれぞれと不必要にリサイズが行なわれる可能性があるため、つけないように設定した。タイトルバーが必要ないものはウィンドウの枠も必要ないと思われるため、枠についてもいくつかのものにはつけない設定をした。

ルートウィンドウからのマウス左ボタンにより、xtermをはじめとする代表的なツールを起動するためのメニューが表示されるようにした(図1)。

muleが2つあるのは、フォントのサイズを切り替えて起動するためのものである。通常muleは14ドットのフォントで起動されるが、Mule-16を選択すると16ドットのフォントで起動される。(起動してからフォントの大きさを変更することも可能である)。Toolsにマウスポインタがあるとさらに別のメニューを呼ぶことができるようになっている(図2)。Rloginにより、従来使用していた2台のワークステーションにrloginできるようになっている。また、K-Rloginはktermからrloginするためのものである。

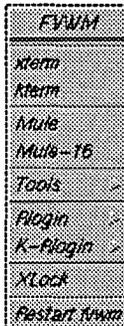


図1 ルートウィンドウからのメニュー

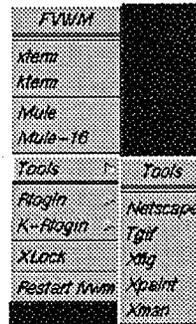


図2 Toolsの選択

ユーザからのloginにはxdmを採用した。xdmについてはデフォルトの設定をそのまま用いた。また、xdmのスタートアップファイルであるXsession内の設定は以下のようにした。

```
xrdb -load /usr/X11R6/lib/X11/xdm/xresources-local
fvwm &
xmodmap /usr/X11R6/lib/X11/xdm/xmodmap-ctrl
/usr/local/bin/xpbiff -xnllanguage ja_JP.JIS7 -geometry 48x48+0+0 &
exec xterm -C -fn 5x8 -sb -sl 256 -geometry 80x5+50+0 -n login
```

xmodmap-ctrlでは、よく行なわれているように、キーボードの「Caps Lock」と「Control」の機能を交換している。これにより、左手小指で「Control」を押すことが極めて容易となる。

メールの着信を知らせるためのツールとしてxpbiffを起動している。学生への連絡に電

子メールを用いることが多いため、このツールを起動しておくことにより、学生は login 直後に誰からメールが届いているかを知ることができる。

最後は、xterm をコンソールとして exec することにより終了している。このウィンドウは小さなフォントを用いているが、それは主にシステムのエラー出力の表示用としているからである。

xdm から login した直後の画面には、左上に xpbiff、コンソールとしての xterm、右下に fvwm のページャである 6 つの仮想ウィンドウ、というように隅に 3 つのウィンドウがあるのみとなっている (図 3)。これは比較的シンプルな初期設定かと思われる。色調はグレイを中心とし目に負担にならないよう配慮した。

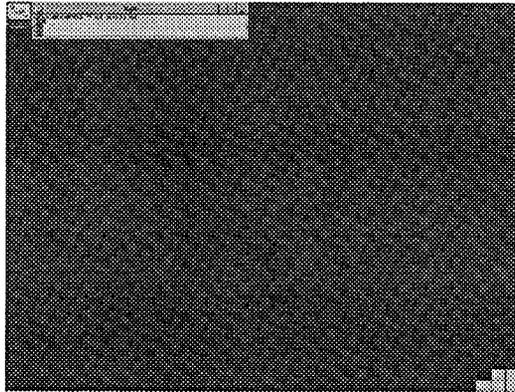


図 3 login 直後の画面

**3-8-2 xpbiff — X の国際化** メール到着を知らせる biff 機能のためのツールとして、xpbiff を採用した。xpbiff は、From: と Subject: とをポップアップして表示してくれるので非常に便利なツールとして世界中で愛用されている。また、MIME エンコードされた日本語をデコードして表示することも可能となっている。

ところが、xpbiff をインストールしたところ、MIME エンコードされた日本語のデコードがうまく動作しなかった。当初は xpbiff の方に問題があるのだろうといろいろ調べたが問題はなく、X のサーバの方に問題があることがわかった。すなわち、FreeBSD 2.1.0R における X は国際化のための `-DXLOCALE` というオプションがない状態で `make` されたものであることがわかった。X を入れ替える必要があったが、`-DXLOCALE` というオプションをつけて X をコンパイルするのは大変なのでネットワーク上を探索したところ、運よくコンパイル済みのものを見つけることができた<sup>4)</sup>。それをインストールして X を総入れ替えすることにより、xpbiff の日本語デコード表示が初めて可能となった。

**3-8-3 X のその他のツール** X Window System のツールの多くは FreeBSD の package を利用したためにコンパイルすることはなかったが、いくつかのツールは環境により細かい設定が異なるため、やはり自前でコンパイルする必要があった。

ghostscript, ghostview はそういうツールの中の代表的なものである。これら 2 つのツールはポストスクリプトファイルをプレビューするためのものである。

これら以外の X のツールとして、tgif, xfig, xgraph, gnuplot, xcal, xmeter,

dclock, xdviなどをインストールした。

3-8-4 mule 電子情報工学科ではUNIXにおけるエディタとして従来 nemacs を用いていたが、今回のシステム更新に伴い nemacs の使用をやめて mule を用いることにした。nemacs も mule も emacs を自然言語的に拡張したものであるが、前者はすでに開発も終わり今ではサポートされておらず、現在広く使われているのはもっぱら後者の mule の方である。PC 納入時点での mule の最新バージョンは 2.3 であったので、そのバージョンのをインストールした。

インストールの際にはかな漢字変換をあらかじめ指定する必要がある。かな漢字変換の代表的なものとして Wnn と Canna とがあるが、両方を指定するとうまくいかないという報告もあったため、今回は、より広く使われていると思われる Wnn を指定することとした。

また、電子情報工学科では SKK というかな漢字変換を従来から奨励していた。SKK は mule 上でも問題なく動作するため、これもインストールすることにした。SKK については mule のインストールの際に指定する必要はなく、mule のインストールが終了した後に追加することが可能である。

以上により、mule の上では Wnn と SKK と両方のかな漢字変換システムが動作することとなった。

次に、mule 起動時に読み込まれる初期設定ファイル (/usr/local/lib/mule/VER/lisp/default.el) を細かく設定する必要がある。やはり一種の規範となるべきものであるので、試行錯誤による慎重な作業となった。

ファイルの内容を以下に示す。

```
; util
(setq next-line-add-newlines nil)
(line-number-mode 1)

; Wnn
(setq jserver-list '("tms" "katsuo" "wakame" "ikura"))
(setq enable-double-n-syntax t)

; hilit
(load-library "hilit19")

; GNUS
(load "gnus-init.el")

; MH
(setq mh-progs "/usr/local/bin/mh/")
(setq mh-lib "/usr/local/lib/mh/")
(setq mh-summary-height 9)
(setq mhl-formfile "mhl.noconv")

; TRR
(autoload 'trr "/usr/local/lib/trr/trr" nil t)

; supercite
(load "sc-default.el")

; ispell-3.0.09
(autoload 'ispell-word "ispell"
  "Check the spelling of word in buffer." t)
(global-set-key "\e$" 'ispell-word)
(autoload 'ispell-region "ispell"
  "Check the spelling of region." t)
(autoload 'ispell-buffer "ispell"
  "Check the spelling of buffer." t)
```

```

(autoload 'ispell-complete-word "ispell"
  "Look up current word in dictionary and try to complete it." t)
(autoload 'ispell-change-dictionary "ispell"
  "Change ispell dictionary." t)

;;;;
;;;; for Students
;;;;
(load "students-global.el")
(load "case-changers.el")
(load "private_c_mode.el")

; skk8.6
(load "skk-init.el")

; TeX environment
(load "mytex2.el")

; mew
(load "mew-default")
(defvar mail-yank-original-default-indentation0 " ")
(defvar mail-yank-original-quoting-indicator "> " " ")

; for less frequent garbage collection
(setq gc-cons-threshold 1000000)

; for tm5.9.1
(load "mime-setup.el")

(setq-default fill-column 60)

; c-mode-hook
(add-hook 'c-mode-hook
  (function (lambda ()
    (setq file-coding-system *sjis*))))
;
; shell
(display-time)
(shell)

```

まず、util における最初の行であるが、カーソルを下へ移動するとき新たに改行コードを入れないようにするためのものである。util の次の行は、モードラインに現在の行数を入れるためのものである。

Wnn の 2 行目では「nn」による「ん」の入力を可能とした。

hilit では、各モードに対応する色づけを行なうようにした。

GNUS の行は、ニュースリーダー GNUS の設定のためのものである。ニュースリーダーとしては、やや重いという弱点はあるものの、従来どおり GNUS を用いることとした。GNUS の設定は多岐にわたるので gnus-init.el という別のファイルにまとめ、それをロードすることとしている。

MH の行では、MH と呼ばれるメールハンドラについての設定がなされている。メールの読み書きは mule 上で MH を用いることとした。mh-rmail の他に mew も用いることができるようにしてある。MH のバージョンは、当時の最新である“MH 6.8.3.JP2c”を用いた。

TRR の行では、タイピング練習ソフトである trr の設定をしている。

supercite の行では、メールの返事を書く際に元のメールに引用符をつけるためのツールである supercite の初期設定ファイルをロードしている。supercite の設定もやはり多岐にわたるため、sc-default.el という別のファイルをロードすることとした。

ispell の行では、スペルチェッカーとして広く使われている ispell についての初期設定が

行なわれている。

for Students の 3 行では、電子情報工学科ローカルの設定を行なっている。これは従来好評であった各種設定をまとめたものである。students-global.el においては、便利なキーバインドや関数、オートセーブの設定などを行なっている。case-changers.el においては、カーソルが置かれている単語の大文字化・小文字化・先頭のみ大文字化などのための関数が定義されている。private\_c\_mode.el においては、C モードのときにさらに便利になるような関数が定義されている。

skk の行においては SKK の初期設定ファイルをロードしている。

TeX environment の行においては、mule から IAT<sub>E</sub>X の処理を行なえるような関数などが定義された mytex2.el というファイルをロードしている。

mew の行ではメールリーダー mew の初期設定ファイルのロードと引用の際に必要な変数を定義している。

garbage collection の行では、割と頻繁に行なわれるガーベージコレクションの回数を少なくするように設定している。

tm5.9.1 の行では、MIME を扱うための関数群である tm の初期設定ファイルをロードしている。

その次の行では、fill モード時の横幅を 60 に設定している。

c-mode-hook の行では、C のソースファイルを開いたときに漢字コードが sjis になるように設定してある。

最後に、モードラインに現在時刻を表示させ、\*shell\*バッファを作成することにより、mule が立ち上がる。

**3-8-5 その他のツール** 文書処理システムとして IAT<sub>E</sub>X をインストールした。IAT<sub>E</sub>X にも様々な種類がありどのバージョンをインストールするか迷ったが、結局は、電子情報工学科で従来用いていたのと同じバージョンである ASCII の jtex1.7 をインストールした。これはソースからのコンパイルとなった。

他に、netscape, perl, flex, mpeg\_play, ispell, mtools, bison, sox, lha, GNU make, gzip, GNU sed, GNU awk, gcc, bash, tcsh などを用いたツールとしてインストールした。

また、メールの送受信のために必要となる /usr/lib/sendmail、/etc/sendmail.cf のファイルは、WIDE のパッチをあてた最新版のものに入れ替えた。

### 3-9 カーネルの再構築

デフォルトの /kernel をそのまま用いると立ち上げ時に存在しないデバイスを見に行こうとするために起動が遅くなってしまいます。そのため、存在しないデバイスを削り新たな /kernel を作成することにより立ち上げを高速化することが可能となる。

### 3-10 子機の複製

以上の一連の作業により親機と子機の一对が完成した後に行なうことは、親機のサーバ部分をワークステーション側にコピーする作業と、子機の内容を他の PC に移植させる作業である。

子機の移植作業の手順は以下のようにした。まず、PC に一台ずつ root で login し、sysconfig, hosts, hosts.equiv, passwd をネットワーク的に使えるようにエディットしてリポートする。後述の「子機作成スクリプト」と子機作成のために必要なファイル一式とを rcp した後に、子機作成スクリプトを実行した。全 44 台にこれを行なうには 6 時間ほど要した。

子機作成のために必要なファイル一式を以下に示す。

```

etc          XF86Config, host.conf, master.passwd, resolv.conf
            spwd.db, passwd, fstab, printcap, sendmail.cf
            group, hosts.equiv, pwd.db, services, rc.local, shells

/usr/lib     sendmail
/           kernel
DES library  des.aa
base file   hosts.BASE, sysconfig.BASE

```

参考のため、子機作成スクリプトを以下に示す。

```

#!/bin/csh
#
#      mk -- make a "child machine"
#
#      by K.Ohya

#      /usr/mk -- this script
#      /usr/Staff -- staff

#      USAGE:
#      # ./mk NUM (NUM: 01--44)
#

if ($#argv == 0) then
    echo "Usage: ./mk NUM"
    exit 1
endif

# hosts, sysconfig
echo 'hosts, sysconfig...'
cd /usr/Staff
sed -e "s/NUM/$1/g" hosts.BASE > hosts
sed -e "s/NUM/$1/g" sysconfig.BASE > sysconfig

# /root
echo '/root...'
cd /root
cp .cshrc .cshrc.ORG
cp .login .login.ORG
cp /usr/Staff/.cshrc .
cp /usr/Staff/.login .

# tcsh
echo 'tcsh...'
cp /usr/Staff/tcsh/bin

# kernel
echo 'kernel...'
/usr/bin/chflags noschg /kernel
mv /kernel /kernel.old
cd /usr/Staff
/usr/bin/install -c -m 555 -o root -g wheel -fschg kernel /

# /
echo '/tmp...'
cd /
tar cvpf /usr/tmp.tar ./tmp
cd /usr

```

```
tar xvpf tmp.tar
chown bin.bin tmp
chmod 1777 tmp
cd /
mv tmp tmp.ORG
ln -s /usr/tmp

echo '/var...'
cd /
tar cvpf /usr/var.tar ./var
cd /usr
tar xvpf var.tar
cd /
mv var var.ORG
ln -s /usr/var

# des
echo 'des...'
cd /usr
cat /usr/Staff/des.aa | tar --unlink -xpf - -C /
cd /etc
cp passwd passwd.ORG
cp master.passwd master.passwd.ORG
cp pwd.db pwd.db.ORG
cp spwd.db spwd.db.ORG
cp /usr/Staff/{passwd,master.passwd,pwd.db,spwd.db}.
chmod 600 spwd.db spwd.db master.passwd
chmod 644 pwd.db
chown root.wheel passwd master.passwd pwd.db spwd.db
touch passwd master.passwd pwd.db spwd.db

# home dir
echo '/home0...'
cd /usr
tar xvpf /usr/Staff/home0.tar
cd /
ln -s /usr/home0

# symbolic links
echo 'symbolic links...'
cd /
mkdir /mnt/wakame
mkdir /mnt/wakame/home1
mkdir /mnt/wakame/home2
mkdir /mnt/wakame/home3
mkdir /mnt/wakame/home4
mkdir /mnt/tara00
mkdir /mnt/tara00/usr
rm -rf home
ln -s /mnt/wakame/home2/home
cd /var
mv mail mail.ORG
ln -s /mnt/wakame/home1/var/spool/mail
cd /usr
ln -s /mnt/wakame/home1/Tara
mv X11R6 X11R6.ORG
ln -s Tara/X11R6
ln -s X11R6 X11
mv local local.ORG
ln -s Tara/local
mv share share.ORG
ln -s Tara/share
cd /usr/lib
ln -s /usr/local/usr.lib.TEX/tex
ln -s /usr/X11R6/lib/X11
cd /usr/include
ln -s /usr/X11R6/include/X11

# printer
```

```

echo 'printer...'
cd /var/spool
mkdir oki tama
chown daemon.daemon oki tama
chmod 2710 oki tama

# etc
echo 'etc...'
cd /usr/Staff
foreach file (XF86Config fstab group host.conf hosts.equiv printcap rc.local\
 resolv.conf sendmail.cf services shells hosts sysconfig)
  if ( -e /etc/$file ) then
    mv /etc/$file /etc/$file.ORG
  endif
  cp $file /etc
end

# sendmail
echo '/usr/sbin/sendmail...'
cd /usr/sbin
mv sendmail sendmail.ORG
cp /usr/Staff/sendmail .
chown root.kmem sendmail
chmod 6555 sendmail

echo 'done'
echo ''
echo 'rebooting...'
sleep 10
/sbin/shutdown -r now

```

### 3-11 現在までに生じた問題点

**3-11-1 X のハング** 現在までに生じた問題点として最大のものは、突然 X がハングしてしまうというものだった。症状は、マウスもキーボードも効かないが、ネットワーク的に外からは何ら問題なく login できるというものであった。症状から見ると単に X が落ちてしまっているようだった。落ちる割合も割と頻繁であり、3 時間の実習で 10 台前後が落ちてしまうため、そのためにシェルを kill する必要があり非常に不便であった。

どういうときに X が落ちるのかを調べようとしたが、何か特殊な操作をしているわけではなく何かの拍子に単にうまくいかなくなるだけのようなようだった。

原因がわからないために対応の取りようがなく困っていたが、過去の NetNews<sup>5)</sup>に同様な質問と回答とがあったことを知った。それによると、どうやらキーボードが原因のようであった。NetNews<sup>5)</sup>の記事にあるように、処置として、syscons.c 中の DELAY の値を全部 10 倍にした新たなカーネルを生成することにより解決した。

**3-11-2 ネットワークトラフィック** NFS を導入したときから心配であったネットワークトラフィックの問題であるが、44 台を全く同時に立ち上げるときには、立ち上がるまでにさすがに数分かかってしまう。しかしながら、通常の実習のように実習室に来た学生から順番に立ち上げていくような場合には何ら問題はなかった。

立ち上げるとき以外にトラフィックを発生させる可能性としては、実習時において同時に同じツールを学生に使わせる場合がある。例えば、Netscape Navigator 3.0b8 のサイズは 5.2MB であり非常に大きい。これを全 PC からほぼ同時に起動させたところ 5 分以上かかった。よって、実習時において同時に同じツールを使わせる場合には、システム起動時にまずそのツールを立ち上げるように指導するなどして起動のタイミングをずらす工夫が必要となる。実際にこの方法を取ることににより、スムーズに実習を開始することができた。

以上、NFS 導入時に心配されたことは、起動のタイミングをずらすことにより問題なく解決することができた。

#### 4. 新しいシステムの利用状況

本システムは、平成8年度当初より使用し始めており、学生からも好評を得ている。ことに、卒業研究の学生は、研究室に設置されているワークステーションよりも本システムの方が高速の UNIX システムとして扱える場合も多いので、盛んに利用しているようである。また、低学年の教育では DOS/V 環境を今までと同様に利用できるため、これまでと変わりなく従前通りの教育が実施できている。高学年の実験ではクラス全員が同時に X 環境を使えるようになったことで、大きなメリットがある。グラフツールなどが手軽に利用できるようになり数値計算の授業などでも視覚的に結果が理解しやすくなった。これにより学生はわかりにくい数値結果にも以前より興味を示すようになったと思われる。さらにシステム自体が使い易くなったため、応用問題としてやや高度な、カオス、電気回路、物体の運動などの課題も出題しやすくなり、いろいろな問題に取り組むことが可能となった。

本システムを利用する授業は、表に示すように実験を中心に数多くある。

本システムを利用する授業	
2年:	「マイクロコンピュータ」(週2時間) 「工学実験」(週3時間)
3年:	「情報処理」(週2時間) 「工学実験実習」(週3時間)
4年:	「情報処理」(週2時間) 「工学実験」(週4時間)
5年:	「ソフトウェア工学」(週2時間) 「工学実験実習」(週4時間)

また、上述のように卒業研究などでも活用されているため、稼働率は非常に高い。なお、1年生については混合学級の一斉教育としての「情報処理」の科目が、本校の情報処理センターで実施されているため、現在のところこの学科独自のシステムを使う授業はない。

このほか、夏期休業中に電子情報工学科が主催して行う公開講座「中学生のためのインターネット入門」などについても、情報教育センターの機器を使用した昨年度<sup>6)</sup> <sup>7)</sup>とは異なり、本年度はこのシステムを利用して実施した。

本システム利用上の問題点としてはまず、システムが複雑になったため、とくに低学年の学生が DOS(Windows) を立ち上げるべきところを、誤って Free BSD を立ち上げてしまい、学生だけでは対処できなくなったりすることがある。また、高学年の学生が Free BSD を立ち上げたまま席をはずし、そのまま次の授業が始まって低学年の学生はどうしてよいかわからないという事態も時々あり、教官が対応に追われることがある。さらに、卒研生が他学年の授業時間中に Screen Lock をかけてリモートで使用(あるいは放置)している場合があり、これは教官でもすぐには対処できず授業に支障を来たすということもあった。これらは、ほとんどが導入教育とモラルの向上で解決できる問題であり、事実、最近ではこの種の支障は少なくなってきた。

運営開始当初の大きな問題の一つに、UNIX システムがやや不安定である点があった。使用中に X Window がハングすることがしばしば起こり、授業中に教官や学生を悩ませた。これは、FreeSoftware を使用しているために生じるシステム運営の情報不足も原因の一つであるが、逆に FreeSoftware であるが故にネットニュースなどでもトラブルに関する同様の事例が討論されており、これらを参考にしてほぼ改善することができている。

## 5. おわりに

本システムが稼働し、実際に授業に活用し始めてから半年強が経過した。これまで述べてきたように、かなりの労力を費やして構築したシステムであるが、幸いにして概ね当初の計画通りの性能を発揮しており、学生にも好評である。また、夏の公開講座および秋の学校祭(工嶺祭)の公開企画でも活用され、中学生を中心に理工系教育の啓蒙活動に効果を上げている。今後さらに適切な運営と有効利用を図るために、一層のシステムの充実と拡張に努めたい。

## 参 考 文 献

- 1) 中澤、大矢 他, “長野高専電子情報工学科の情報処理教育用マルチ OS-PC システムについて”, 「情報処理教育研究発表会論文集」第 16 号 (1996,8 月), pp.120-121.
- 2) “ドン! ときたら困るけど、ネットワーク管理(6)”, 「UNIX Magazine」1993 11 月号, pp.17-36.
- 3) “第 16 章 ネットワーク・インフォメーション・サービス”, 「システムとネットワークの管理」(日本サン・マイクロシステムズ(株)), (1993), pp.513-560.
- 4) <ftp://ftp.noc.titech.ac.jp/pub/tmp/aki/FreeBSD/etc/X312LOCA.tgz>
- 5) <news://tms.ei.nagano-nct.ac.jp/fj.os.bsd.freebsd/>
- 6) 楡井 他, “公開講座「やさしいコンピュータ通信」”, 「長野工業高等専門学校紀要」29 卷 (1995,12 月), pp.101-108.
- 7) “'95 ふれあいサタデープラン報告書”, (1996,3 月).