

## 学生用電子メールサーバの試験的構築

今 井 慈 郎  
西 平 浩 孝  
安 倍 麻 衣  
本 田 道 夫  
瀬 野 芳 孝  
曾 根 計 俊  
丸 山 久 美 子

### 1. はじめに

香川大学キャンパス情報ネットワークが構築され、香川大学も学内 LAN の整備とインターネットへの接続という新しいネットワーク時代へ移行し始めている。学内 LAN は、

1. 100 Mbps の高速 FDDI 光ループの幹線
2. 10 Mbps のイーサネットワークの支線
3. 情報処理センターの UNIX [ 1 ], [ 2 ] およびパソコン環境
4. 教職員の UNIX およびパソコン環境

などの要素から構成されている。

一方、インターネットへの接続は、SINET インターネットバックボーンへの IP 接続により実現されており、比較的安価な運用経費や少ない作業負担でかなり大きな効果が期待できる [ 3 ] ~ [ 7 ], [ 10 ]。その接続効果は、

1. 岡山大学内の SINET ノードと香川大学とが 1.5 Mbps の高速専用回線で接続され、リアルタイム性や高い信頼性など多様なメリットを持つ情報通

信網を利用した情報交換

2. telnet/ftp など TCP/IP [ 8 ] レベルの各種アプリケーションが利用できることによる情報通信環境の向上と他大学や研究機関などとの共同研究の可能性

などがあり、潜在的にもより先進的な研究環境が入手できたことになる。ほとんど毎年のように専用回線のスピードアップが認められたため、当初の回線速度 64 Kbps から 512 Kbps を経て、現在の高速回線も情報化推進の国策によってさらに高速化が期待でき、地理的条件を意識しなくても学外組織との間で様々な研究や実験が共同で行える環境が実現できるようになってきた。

学内 LAN の整備とインターネットへの接続という環境の変化によって実現可能となるサービスは少なくない。その内でも電子メールサービスの充実の特筆すべきである。それまでの UUCP 接続から IP 接続へ移行したことで、これまで以上に信頼性と高速性が期待できる。かつて一部の研究者のみが利用していた電子メールサービスも、実績を重ね広く市民権を得る存在となってきた。教職員のみならず、香川大学の学生に対しても電子メールサービスの拡大を前向きに検討し始める時期が到来していると思われる。新聞紙上でも氷河期と形容される就職戦線においても『パソコンと電子メール』がキーワードになりつつある [ 9 ]。パソコンは情報処理環境の、電子メールは情報通信環境のそれぞれ代名詞として使用されているとも考えられる。買い手市場の就職戦線において、雇用側が学生の能力の差別化を図る目的で利用し始めていることは自明であろう。学生にどのような形式で情報処理教育や情報通信教育を行うか、など大学全体の情報教育の取り組み方が問われる問題とも言えよう。

本稿では、主として電子メールサービスを行う上での環境をどのように整備するかという観点から、香川大学の現状に適した分散型電子メールサービス環境の仕様、電子メールサーバの設定と運用の手順、試験利用と問題点のまとめなどを行う。学生用メールサーバを試験運用することで、学生全員を対象とする規模の大きな電子メールサービスをどのように実現していくかを検討する一助としたい。

## 2. 分散型電子メールサービス環境

香川大学キャンパス情報ネットワークを有効に活用し、電子メールサービスを効率よく実現するには電子メールサーバを分散配置し、個々のマシン管理をそれぞれの管理者に移譲する方針が適していると思われる。本稿では、電子メールサーバの分散配置と電子メールサービスの独立運用を総称する意味でこのようなサービス体制を分散型電子メールサービスと呼ぶことにする。物理的に分散配置された個々のメールサーバと論理的存在であるサブドメインを適切に対応づけることで、分散型電子メールサービス環境が構築されている。現在、この形態を発展させることで、UNIX マシンを中心とした情報処理センターの計算機環境も分散型情報サーバ体制の実現へと進んでいる。

まず、サブドメインと分散配置されたメールサーバとの対応関係を具体的に説明する。香川大学が JPNIC<sup>(1)</sup>から与えられたドメイン名 (kagawa-u. ac. jp) の下に

- 情報処理センター : cc. kagawa-u. ac. jp
- 教育学部 : ed. kagawa-u. ac. jp
- 法学部 : jl. kagawa-u. ac. jp
- 経済学部 : ec. kagawa-u. ac. jp
- 農学部 : ag. kagawa-u. ac. jp
- 事務部 : ao. kagawa-u. ac. jp
- 学生組織 : stu. kagawa-u. ac. jp

などのサブドメインを設定している。与えられたドメイン内でさらにサブドメインを設定すること自身は JPNIC から一任されているが、同時に学外からも電子メールの送信先に指定できるメールサイトとして認識されるような処置<sup>(2)</sup>を行っている。

(1) JaPan Network Information Center の略称。国際的調整を行う InterNIC と協議しながら、日本国内におけるインターネットアドレス割り当て業務などを担当する組織。

(2) ネームサーバへの設定などの処置を意味している。

以下では経済学部電子メールサーバである `fourier` を実例として説明する。`fourier` は正式なドメイン名 (FQDN: Fully Qualified Domain Name) である `fourier.ec.kagawa-u.ac.jp` でインターネット上の総てのサイトからアクセス可能であり、これが `fourier` の正式なサイト名となる。一方、他のサイトから電子メールを経済学部のメールサイト名である `ec.kagawa-u.ac.jp` に向けて送信する場合には、第一受信先メールサーバとして `fourier` が位置付けられている。従って、`fourier` に登録されているユーザは、電子メールの宛て名として、

ユーザ名@ec.kagawa-u.ac.jp

または

ユーザ名@fourier.ec.kagawa-u.ac.jp

で受信できるよう設定されている。敢えて識別すれば、前者は経済学部の正式メールサイト名に基づいたアドレスであり、後者は `fourier` の FQDN に基づいたアドレスと言える。ネームサーバへの登録によって、FQDN (サブドメイン名と関連) とメールサイト (メールサーバ名との関連) とを論理的に関係づけられ、電子メールサービスとも関連づけられている。

このようにメールサイトに対応させる形式でメールサーバを分散配置する方式の利点として、

- 全学的にユーザ名を指定する場合に比べてユーザ名の衝突が低減できる
- メールサーバの管理業務が個々のマシンに局所化できる

などが挙げられる。複数のメールサーバが存在するという学内 LAN の繁雑さは否定できないが、大学全体のメールサービス体制から見れば、総てのマシンが同時にダウンする可能性は低いので全体の serviceability は確実に向上することになる。

メールサーバの分散配置という体制が個々のメールサーバ管理者の様々な努力を要して実現できたことを受けて、各種情報サーバを個々に立ち上げ、機能を分散配置して実現するという環境が整備されたと言える [11]。メールサーバ

---

(3) 通常、E-mail アドレス (Electronic Mail Address) と呼ばれる。

以外の情報サーバとしては、学内 BBS サーバ、インターネットニュースサーバ、Anonymous FTP サーバ、Gopher サーバ、WWW サーバあるいはネームサーバなどが想定される。平成 8 年 1 月 1 日現在、香川大学としては Gopher サーバ<sup>(4)</sup>を除いて情報サーバのラインアップが一通りの完成をみている。本稿で用いている情報サーバという記述に必ずしも明確な定義が存在する訳ではない。しかし、計算機を用途に応じてカテゴリに分類する時、CPU サービスの提供を主とする情報処理目的の計算機群と、情報交換などの通信サービスの提供を主とする情報通信目的の計算機群とに大別できる。ここでは情報通信目的の計算機群を一括して情報サーバと表現している。最近、情報処理のための研究および教育環境を実現する目的で、高速処理機能や多様な言語でのプログラミング機能を提供する CPU サーバや、多くのユーザを登録でき大容量なデータ領域を提供できるファイルサーバなどに対する需要が相対的に低下している。替わって、情報サーバへの依存度が増加する傾向にある。パソコンが高性能 CPU を搭載していながら、プログラミングを目的とした利用方法よりも、ワープロ、表計算あるいはデータベースに加えて通信ソフトウェアなどを標準装備する利用方法が一般的になりはじめたことと明らかに符合している。

電子メールサービスは情報通信機能の代表的サービスの 1 つである。前述したように、学生の就職戦線にキーワードとして登場するという象徴的な事態もさることながら、電子メールのアドレスが名刺や年賀状に記されていたり、連絡先として学会の名簿などに電話番号、FAX 番号などと共に併記されていたりすることからも、基本的な情報通信手段の 1 つとして市民権を得てきた結果と言っても過言ではない。インターネットが新聞紙上を賑わせ、マルチメディアとか WWW/Mosaic/Netscape (正式名称: Netscape Navigator) など派手な存在が注目されていることは間違いない。しかし、電子メールサービス自身は、派手さこそないものの、インターネットの有用性を高めている重要なプロモータ役であると断言できる。個々の WWW サーバへアクセスすることは正

(4) Gopher サーバが提供する機能は基本的に WWW サーバに包含されていると考えられるので、香川大学で立ち上げる予定はない。

直言って飽きられる傾向にあるが、インターネット加入者にとって電子メールサービスは今後ますます重要度を増すと思われる。

電子メールサービスの利便性を簡単にまとめてみると、次の5つのキーワードに代表されると思われる。すなわち、

- 即時性
- 非同期性
- 機密性
- 再利用性
- マルチメディア対応

である[5]。即時性とは信頼性の高いリアルタイムな情報交換が可能であることを示している。電子メールが通常の郵便メールなどに比べて高速であり、海外であっても数十秒から数十分の所要時間で情報伝達ができ、即座に返事も得られるなど特筆すべきである。即時性だけなら電話の方が期待できそうであるが、相手が不在でも他と電話中であっても送信でき、受信者と同期をとる必要がないのも利便性を高めている。この点ではFAXが競合しそうであるが、個人利用でかつファクシミリ機が設置されている場所に鍵がかかっている場合を除いて、電子メールのパスワード保護機構の方が機密性に優れ、手渡しに準じるプライバシーの確保が実現されている。また、施錠されているファクシミリ機と比較しても、電子メールでは出張や休暇などで地理的に隔たった場所に居てもtelnet機能やメール転送機能などを利用することでメールを読める可能性が高く、この点でもFAX以上のaccessibilityが期待できる。伝達されるメディアが計算機のファイル形式であるため、再利用性が保証されている。単なるコピーとは異なり、送信あるいは受信したメールの内容を記録することが可能である。全文または一部分を引用したり、修正したりする広義の再利用が容易であるなど、他の情報伝達手段と格段の差があると言える。電子メールとして扱わ

---

(5) 個人的な見解であり、一般的には今でもインターネットの代表的なアプリケーションであることは否定できない。HTMLのバージョンアップやJavaなど新しい記述仕様が提案され、進化し続けていることも事実である。

れるファイル中に画像情報や音声情報を添付することが可能であるため、マルチメディアへの対応性に至っては既存の情報伝達手段では比肩すべき存在を見出すことができない[12]。

電子メールサービスを実現する電子メールシステムは大きく2つの機能、すなわち

- メールを配送管理する機能
- メールを読み書きする機能

から構成される。前者をメールシステムのサーバ機能、後者をクライアント機能と見ることも可能である。サーバ機能とクライアント機能とは明確に分離して実装可能である。従来、これら2つの機能を単一のメールサーバ(主としてUNIXマシン)上に実装する場合が一般的であった。現在でもインターネット上では、8割以上のサーバ機能がUNIXマシン上に実装されていると思われる。電子メールシステムのサーバ機能[14]を実装しているUNIXマシンを通常メールサーバと呼び慣わしているのはこのためである。一方、メールを読み書きする機能はユーザインターフェースを司っており、電子メールシステムのクライアント機能を担っている。この機能もかつてはUNIXマシン上に実装されるのが一般的であった。UNIXマシン上に実装されたクライアント機能を、現在でも利用しているユーザは少なくない。しかし、高機能パソコン(WindowsPCやMacintoshなど)が低価格で市販され始めるに従って、ユーザインターフェースに優れたパソコン上にメールを読み書きする機能を実現する動きが活発化し、結果としてクライアント機能の実装比率はパソコンが主力となっている。これは分散処理という観点からみても有効な実装手法であり、電子メールシステムのサーバ機能はUNIXマシンを中心とした24時間稼動可能なサーバマシン上に実装し、クライアント機能は個々のユーザの手元近くにあるWindowsPCやMacintoshなどのパソコン上に実装するのが利便性からみて

(6) 現時点では総ての電子メール利用者が同一形式のマルチメディア対応の電子メールサービスを利用している訳ではないので、筆者としては必ずしも積極的にマルチメディア対応機能の使用を推奨しているものではない。

も適しており、今後ますます一般的となりそうな傾向と言える。

電子メールシステムを実現するソフトウェア群全体も、サーバ機能を実現するソフトウェアとクライアント機能を実現するソフトウェアとから構成される。記述を簡潔にするため、本稿では

1. メールを配送管理するためのソフトウェア … MTA (Mail Transfer Agent)
2. メールを読み書きするためのソフトウェア … MUA (Mail User Agent) という省略表記を採用する[5]。次節では、試験的に運用を開始している学生用電子メールサーバを例に取り上げ、主としてメールサーバのソフトウェア群の設定を中心に説明する。

### 3. 学生用電子メールサーバの構成

前述したように各種情報サーバの立ち上げが一応完成した現在において、教職員サイドからみれば学内 LAN の利用は個々のユーザのスキルや作業環境に依存するものの、環境上の大きな問題点は存在しないと言える。一方、香川大学全体で4,000名以上の人数を擁する学生にとって電子メールサービス[1][5][13]を利用できる環境は残念ながら整備されていない。問題はいくつか存在する。メールサーバをどのように構成する[14]か、メールサーバをどのように運用するか、など今後メールサービスの実現を検討していく過程で至急に解決しなければならない技術的な問題もあった。そこで、学生用メールサーバのモデルを試験的に構成し、実験的な運用を開始し、問題点の抽出、対策の検討などを行っている。本節では試験モデルとなったメールサーバ leibniz の具体的な仕様や構成手順などについて述べる。

#### 3.1 マシンの基本構成

まず、メールサーバ leibniz のマシン単体としてのハードウェア仕様を述べる。leibniz は比較的小さな UNIX マシン (NEC EWS4800/320VX) であり、

- CPU R4400SC (100 MHz)



- 主記憶                    32 MB
- ハードディスク        1.05 GB
- OS                        UNIX SVR4.2 (EWS-UX/V Rel4.2MP)

というデスクトップサイズのワークステーションである。登録されて利用できるユーザ数も常識的には最大 100 名程度と思われる。試験モデルとは言え、とても全学生を登録できる規模のマシンではない。しかし、UNIX マシンとしてはより大きな規模のマシンであっても基本機能は同様と考えられ、MTA や MUA をインストールしたり、テスト運用したりする対象としては十分に価値があると言える。電子メールサービスの提供が主たる目的であるが、システム管理などのため必要となる他のソフトウェアやシステム設定などをチェックする必要があった。また、本格的な運用の準備を行ったり、予め問題となりそうなデータ等を可能な限り収集しておく意味でも試験モデルからのフィードバックは重要である。

システムを設定したり、試験運用をある程度円滑に行ったりするため、UNIX として初期インストールされているソフトウェア<sup>(7)</sup>以外に、基本的なフリーウェアをインストールしている（表 3-1 を参照）。

表3-1 leibniz にインストールされている基本的フリーウェア一覧

ファイル名	機 能	サイズ(Byte)	格納ディレクトリ
a2ps	PS プリンタ出力	18744	/usr/local/bin
fu	ファイル管理	126792	/usr/local/bin
lha	圧縮解凍ツール	175600	/usr/local/bin
nkf	漢字コード変換	14444	/usr/local/bin
mule	エディタ	2331892	/usr/local/bin/GNU
Mosaic	WWW ブラウザ	1833680	/usr/local/bin/X11
showaudio	音声ビューア	6580	/usr/local/bin/X11

Mosaic および showaudio などを予めインストールしているのは将来のイ

(7) 後述する sendmail が UNIX の初期インストールソフトウェアに MTA として含まれている。

インターネット利用を想定して動作チェックをしておくためであり、必ずしも試験的運用の当初から学生が自由に利用することを意図したものではない。Mule は統合環境型エディタ Emacs[15]の多国語対応版であり、バージョンは 1.1 と少し古いですが、MUA の機能もマクロ (Rmail.el など)により実現されており、UNIX 上でメールサービスの総てをチェックする場合に有効である。

学生用メールサーバであるため、ネットワーク環境の設定は不可欠である。イーサネットで接続するため、IP アドレスが必要となる。leibniz の IP アドレスは試験的に、

133.92.3.206

と定めており、FQDN も情報処理センター管理下のマシンであることを明示するよう

leibniz.cc.kagawa-u.ac.jp

と定義されている。また、メールサイトとしての正式な電子メールアドレスは、登録されているユーザ名が s92e745 であれば、

s92e745@stu.kagawa-u.ac.jp

(あるいは s92e745@leibniz.cc.kagawa-u.ac.jp でも可)

となるよう設定している。このような定義がデータベース化され、他のサイトのメールサーバからも情報が適切に取得できるよう、香川大学のネームサーバ<sup>(8)</sup>にも登録している。

ネットワーク(学内 LAN およびインターネットを含む)を介してアクセスしたいホスト計算機の IP アドレスを取得するため、ネームサーバへの問い合わせが発生する。このようなネームサーバ参照をスムーズに行うため、DNS (Domain Name Service) の設定が必要となる。設定ファイルは /etc/resolv.conf であり、leibniz では

```
domain    kagawa-u.ac.jp
nameserver 133.92.6.17
```

(8) 平成 8 年 1 月 1 日現在での、香川大学プライマリネームサーバは ns1.kagawa-u.ac.jp であり、IP アドレスは 133.92.6.17 である。

```
nameserver 133.92.6.18
nameserver 133.92.3.19
```

という内容になっている。IP アドレスが 133.92.6.17 および 133.92.6.18 のマシンはそれぞれ香川大学のプライマリおよびセカンダリのネームサーバである。一方、IP アドレスが 133.92.3.19 のマシンは試験的に運用されている補助的ネームサーバであり、一般ユーザがパソコンなどの DNS として設定すべき設定になっていない。これ以外にも学内 LAN 内の主要なノードマシンの IP アドレスは /etc/resolv.conf でなく、/etc/hosts にデータベース化され蓄積されている。UNIX では通常、身近なマシンへのアクセス頻度が高い確率であるとみなされ、まず、/etc/hosts が参照され、該当マシンの IP アドレスが取得できなければ改めて /etc/resolv.conf に記述されたネームサーバを参照することになる[11]。DNS 以外のネットワーク環境の設定として、leibniz ではネットワークプリンタの利用を想定しているため、プリンタ環境の設定を行っている。設定ファイルは /etc/printcap である。主要な部分の記述のみ示すと、

```
# 1995/09/04
# Information Processing Center LPC 8 JPS / PN-105
ccp 1 | remote lp name: ¥
    : lp= : ¥
    : rm=npcc 1: ¥
    : rp=ccp 1: ¥
    : sd=/var/spool/lpr/rlp:
ccp 1 ps | remote lp name: ¥
    : lp= : ¥
    : rm=npcc 1: ¥
    : rp=ccp 1 ps: ¥
    : sd=/var/spool/lpr/rlp:
```

となっている。ここで、npcc 1 はネットワークプリンタを制御するプリンタバッファであるネットポート名であり、IP アドレスは /etc/hosts に登録されていなければならない。ちなみに、UNIX コマンドの grep で確認すると、

```
7 leibniz root> grep npcc 1 /etc/hosts
133.92.2.2          npcc 1          # Netport
```

となっている。これにより、前述した a2ps コマンドと組み合わせて、

```
a2ps ファイル名 | lpr -Pccp 1 ps
```

などとコマンド行にタイプすることで、“ファイル名”で指定されたファイルの内容をネットワークプリンタに印刷することができる。他にもネットワーク関係で設定する部分が存在するが、本稿では割愛する。

### 3.2 ユーザ登録と環境設定

香川大学の情報通信環境はこれまで、情報処理環境と同様に課題番号によるユーザ登録をおこなっていた。しかし、情報通信環境は基本的に相手を明示的に指定して情報交換などを行うため、一意に認識できることが必要となっている。そのため、情報通信環境を提供するメールサーバなどでは、課題番号でのユーザ登録から固有名詞でのユーザ登録へと移行している。すなわち、学籍番号が 92E702 で経済学部へ 1992 年度に入学した学生の場合、登録されるユーザ名は

```
s92e702
```

であり、学籍番号の先頭に student の “s” を付加している(総てが小文字)。次にホームディレクトリの構成を示す。ユーザ名に学部名や入学年度などが明示的に表現されているので、現時点では個々の学生ユーザのホームディレクトリを学部別にグループ分けを行って簡単な分類方式を採用している。UNIX の場合、学部別、学科別、学年(入学年度)別などのより詳細な分類も可能であり、既にファイルサーバではこの詳細な分類方式を採用している。

個々のユーザがそれぞれ環境設定を個別に行えるのが UNIX の特徴であるが、基本設定を登録時に行っておくのも管理者側の義務であろう。ログイン時

(9) 教職員の場合は姓名、学生の場合は学籍番号をベースとした文字列となっている。学生の場合、同一学年や同一学部で同姓が存在する可能性が高いため、学籍番号とせざるを得ない状況である。

(10) /home/STUD/Edu が教育学部、/home/STUD/Law が法学部、/home/STUD/Eco が経済学部、そして/home/STUD/Agr が農学部のグループである。しかし、農学部のメールサーバは他の学部のそれと比較しても大容量を有しており、学生ユーザ数も少ないので教官と同一マシン内に登録することも可能である。

の初期設定, パス指定や共通する別名 (alias) の設定などを行うドットファイル (.login/.cshrc/.emacs など) を予め規定している。以下に .emacs ファイルについて一部具体例を示す。

```
;; Display Time and Date
;; (display-time)
;; Kanji Code
;; (set-kanji-input-code 'JIS)
;; (set-kanji-display-code 'EUC)
;; (set-kanji-display-code 'shift-JIS)
;; (set-kanji-input-code 'EUC)
;; Customization
;; (global-unset-key "⌘-h")
;; (global-set-key "⌘-h" 'delete-backward-char)
;; (global-unset-key "⌘-v")
;; (global-set-key "⌘-v" 'scroll-up) ;; this is very extraordinary!!
;; (global-unset-key "⌘-z")
;; (global-set-key "⌘-z" 'scroll-down)
;; (global-set-key "⌘-i" 'indent-for-tab-command)
;; (global-set-key "⌘-x⌘-h" 'info)
;; (global-set-key "⌘-x⌘-z" 'suspend-emacs)
;; GNUS for NetNews
;; (autoload 'gnus "gnus" "Read network news." t)
;; (autoload 'gnus-post-news "gnuspost" "Post a news." t)
;; (define-service-kanji-code "nntp" nil 2)
;; (setq gnus-interactive-post t)
;; (setq gnus-nntp-service "nntp")
;; (setq news-inews-program "/usr/lib/news/inews")
;; (setq gnus-your-domain "stu.kagawa-u.ac.jp")
;; (setq gnus-your-organization "Kagawa University, Takamatsu, JAPAN")
;; (setq gnus-signature-file "~/signature")
```

.emacs ファイル内の総ての設定は予めコメントアウトされている。各ユーザは必要に応じてコメントを解除したり, 新しい記述を追加することが可能である。例えば, GNUS と呼ばれるニュースリーダを利用したければ, 下9行分を

有効にすればよいことになる。

### 3.3 電子メールの配送管理を行う MTA の設定

UNIX-UNIX 間で電子メールの配送管理を行う MTA として、UNIX の基本ソフトウェアの一部に組み込まれているのが sendmail と呼ばれるサーバソフトウェアである [14]。leibniz の場合も、MX に対応するか、しないかの 2 種類の sendmail が予め用意されている。MX に対応するとは、ネームサーバを参照して、電子メールの送信先の IP アドレス等を取得する段階で、指定されたメール送信先のメールサイトに対応する IP アドレスが存在しなくても、MX (Mail eXchange) レコードと呼ばれるメール送信先を指定する記述が存在すれば、そこにメールを送信する機能を有することを意味する。leibniz では試験的意図からもより機能の高い MX に対応した sendmail を使用するよう設定している。leibniz のような SVR4.2 系の UNIX マシンでは、/etc/rc2.d/S82sendmail なるファイル内の指定で sendmail が初期起動される [2]。そのファイルの一部を示すと、

```
# If sendmail is already executing, don't re-execute it.
PATH=/usr/bin:/bin
if [ -f /usr/ucblib/sendmail -a -f /usr/ucblib/sendmail.cf ]
then
    (cd /usr/spool/mqueue; rm -f nf* lf*)
    /usr/ucblib/sendmail -bd -qlh & >/dev/console
fi
```

となっている。そこで、このファイルの記述を直接変更するか、/usr/ucblib/sendmail を MX 対応のバイナリにシンボリックリンクするかの 2 つの方法が考えられる。一般的にシンボリックリンクで設定する方が柔軟性があるので、今回の設定では MX 対応の sendmail.mx を /usr/ucblib/sendmail として利用できるようにシンボリックリンクする方法を採用している。

次に、sendmail がメールサイトの仕様に適応した機能を発揮するよう、sendmail の働きをチューニングする作業が必要となる。学生用メールサイトの仕様

(すなわち、メールサーバ leibniz の仕様であるが)は、メールの送信時と受信時との2つの部分から成り立っている。以下にそれぞれの仕様を列挙する。

#### ◇送信時の仕様

1. 24 時間運用で送信可能
2. 学内 LAN およびインターネットを介して IP ダイレクトなメール送信が可能
3. ネームサーバの参照と A レコード (存在すれば MX レコード) の取得
4. サイト内 (leibniz 内のローカルな) ユーザへの略式メール発信 (ユーザ名のみを指定して送信) 可能

#### ◇受信時の仕様

1. 24 時間運用で受信可能
2. 学内 LAN およびインターネットを介する IP ダイレクトなメール受信が可能
3. 正式な電子メールアドレスを ユーザ名@stu.kagawa-u.ac.jp と指定
4. ユーザ名@leibniz.cc.kagawa-u.ac.jp でもメール受信可能
5. サイト内であればユーザ名のみをメールの宛て先として指定可能

このような仕様を満たすべく sendmail のチューニングを行う設定ファイルが sendmail.cf である。sendmail.cf の記述は、とても通常の間人が手作業で作成するには適していないフォーマットを採用している。一般には、既存の sendmail.cf を入手し、それを修正しながら目的の仕様に適した sendmail.cf を作成するという手順が用いられている [11][14]。leibniz の場合もこの手順に従って作成している。

MTA の設定に関して、重要な問題がもう 1 つ存在する。それは、WindowsPC や Macintosh などのパソコンユーザへの配慮であり、パソコン上でメールの読み書きを行う MUA への対策である。sendmail 自身は電子メールの送信および受信を SMTP (Simple Mail Transfer Protocol) に基づきメール送受信の作業を一手に引き受けて処理するソフトウェアである。24 時間稼動が可能な UNIX マシン上で実行させるには最適と思われる仕様をもったソフト

ウェアの1つである。しかし、SMTPに従ったメール受信ができない個々のパソコンへ、UNIX上で受信したメールを転送するなどの処理にはsendmailの機能は適していない。この点を補完してくれるのがpopperである。これはPOP3プロトコル<sup>(11)</sup>に対応してUNIX上からWindowsPCのメールリーダーなどへメールを転送するフリーウェアである。sendmailとは異なり、システム管理者がインストールを行う必要がある。一方、ソフトウェアの機能が簡潔であるため、sendmailのようなチューニング作業が必要でない点はインストール時の負担を軽くしている。

バイナリを作成した後のインストール作業をまとめておく。通常、実体はディレクトリ/usr/local/etcの下に置かれ、inetデーモンとして起動できるように、

1. /etc/inetd.confへinetd対応のソフトウェアとして起動時の仕様やバイナリのパスなどを登録

2. /etc/servicesへTCPでのアクセス時のポート番号を登録

などを行うことになる。ちなみに、/etc/inetd.confと/etc/servicesとの設定の追加記述は

```
26 leibniz root> grep popper /etc/inetd.conf
pop2  stream tcp    nowait root    /usr/local/etc/popper    popper
pop3  stream tcp    nowait root    /usr/local/etc/popper    popper
```

および

```
27 leibniz root> grep pop /etc/services
pop2          109/tcp      # Post Office
pop3          110/tcp      # Post Office
```

となっている。

### 3.4 メールの読み書きを支援する MUA の設定

MTAの設定は、sendmailの設定およびチューニングとpopperのインストールとが終了した時点で一応の完成とみなせる。そこで、受信したメールを

(11) POP (Post Office Protocol) はバージョンの違いでPOP2/3が存在するが、総称してPOP3と記す。



読み出したり、送信すべきメールを作成するなどのユーザインターフェースを司る MUA<sup>(12)</sup> の設定などについて述べる。MUA も大きく 2 つに分類される。すなわち、UNIX 上のメイラとパソコン上のメールリーダ<sup>(13)</sup> とである。MUA として UNIX 上のメイラを利用するか、パソコン上のメールリーダを利用するかで利用する日本語処理環境も異なる。UNIX 上であれば Wnn または Canna と呼ばれる日本語 FEP が一般的である。leibniz では Canna が標準装備されている。一方、パソコン上であればワープロなどと連動するいろいろな日本語 FEP が利用可能である。実際には、この日本語 FEP の好みに基づいて MUA を選択するユーザもかなりの数に達すると思われる。

leibniz で利用可能な MUA を UNIX 上のメイラから順次説明する。Mule (Emacs)-Rmail と Canna という組み合わせが考えられる。Mule は前述した通り著名なエディタであるが、統合環境型ソフトウェアでもあり、Emacs-lisp で書かれたマクロを読み込むことで様々なカスタマイズが可能となる[15]。メイラやニュースリーダなどを実現するマクロパッケージが数多く存在するのも大きな特徴である。Rmail マクロも Emacs を用いてメイラを実現するため Emacs-lisp で書かれたパッケージである。

```
% mule -e rmail
```

などとコマンド行でタイプすることで Mule をメイラとして起動できる。図 3-1 はメイラとしての Mule 起動画面を示している。より詳細な利用方法については参考文献[15]などを参照して欲しい。日本語 FEP である Canna は Mule との親和性に優れており、簡単なコマンドで Mule 内から利用できる。

WindowsPC 上で POP3 対応のメールリーダとして AL-Mail や Chameleon が提供するメールリーダなどが利用できる。AL-Mail は当初フリーウェアであったため、多くのユーザを獲得し、香川大学情報処理センターのパソコ

(12) MUA は伝統的にメイラ (mailer) あるいはメールリーダ (mail reader) と呼ばれる場合がある。

(13) メイラと同義語であるが、パソコンユーザの間ではこちらの呼称が一般的のようである。

(14) 実際にはコマンド行 `mule -fn 7x14 -fk k14 -e rmail &` で X Window Emulator 上に表示させた画面である。

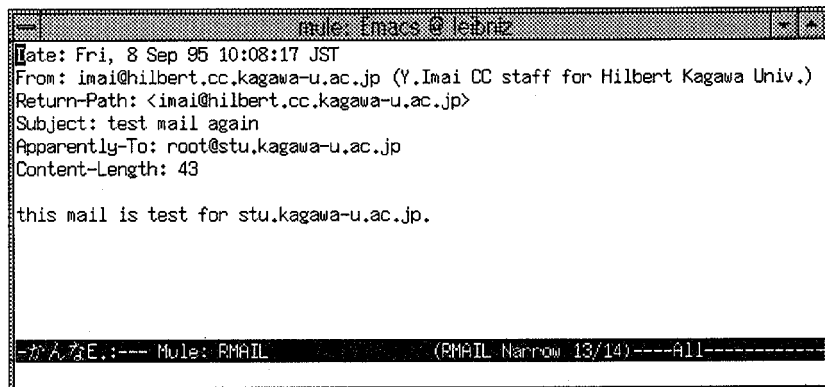
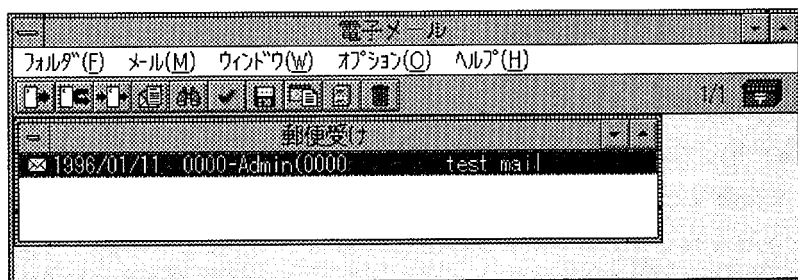
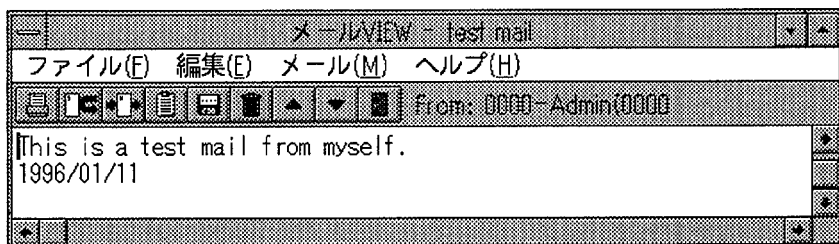


図3-1 メイラとしての Mule の起動画面



(a) メールサーバの受信メールを受け取った画面

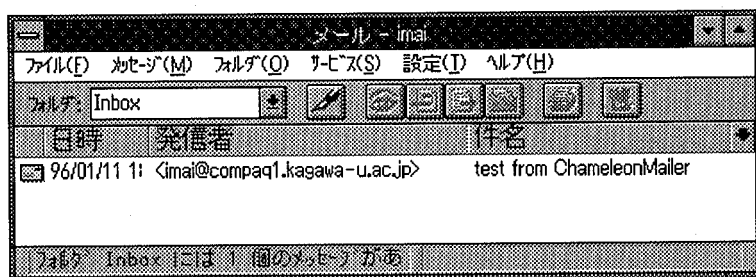


(b) PC上で受信メールを読み出した画面

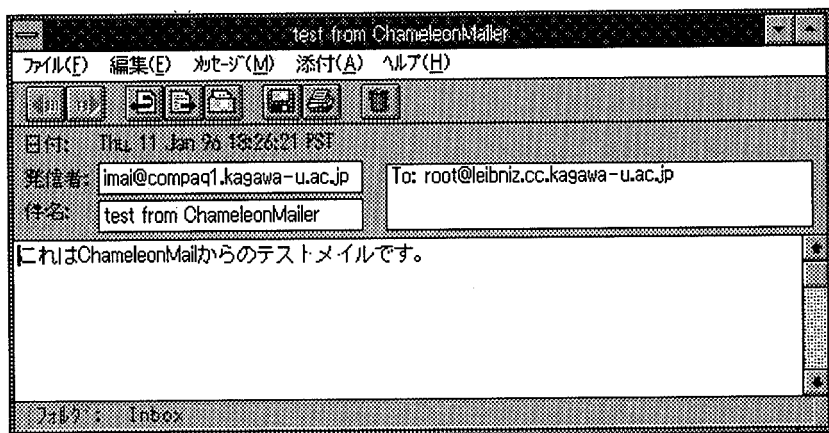
図3-2 AL-mailでのメール読み出し画面

ンルームに配置されている WindowsPC 上にもインストールされている。一方、市販の WindowsPC 用 TCP/IP 対応の通信ソフトウェアとして Chameleon が実績を持っていた。これも情報処理センターのパソコンの総てにインストールされている。Chameleon のパッケージには TCP/IP のプロトコルスタックや TCP/IP 応用プログラムとして telnet/ftp などのコマンド群が標準実装され、メールリーダーやニュースリーダーなどの実用ソフトウェアも組み込まれている。以下、本稿では Chameleon パッケージに含まれるメールリーダーを ChameleonMail と呼ぶ。

AL-Mail および ChameleonMail の特徴として、



(a) メールサーバの受信メールを受け取った画面



(b) PC上で受信メールを読み出した画面

図3-3 ChameleonMailでのメール読み出し画面

1. WindowsPC で利用できる GUI ベースの電子メールリーダ
2. POP3 に対応し指定された UNIX マシンから自動的に受信メールを転送
3. 受信メールをフォルダと呼ばれるディレクトリに分類して管理
4. Windows で利用される他のアプリと同様の日本語 FEP が利用可能
5. JIS コード-シフト JIS コード間の文字コード変換処理の自動実行
6. 大容量ファイルの自動分割メール送信
7. 受信した分割メールの自動アSEMBル
8. 送信メールの内容にバイナリファイルの添付可能
9. 受信メールに添付されたバイナリファイルの読み出し可能

などが挙げられる。電子メールサービスの詳細にこだわることなく利用できる点が、初心者にとって使い易いメールリーダであると言える。図 3-2 に AL-Mail でのメール読み出し画面を、図 3-3 には ChameleonMail でのメール読み出し画面をそれぞれ示す。

他にも、パソコン上で稼動する telnet や X Window Emulator (現時点では WindowsPC など高機能パソコンでのみ利用可能) を使用することで、パソコンからでも UNIX 上の MUA を利用することも可能である。しかし、電子メールサービスのみを利用したい学生ユーザにとってメールサービスを享受するただけに、コマンドなど UNIX の詳細を理解しなければならないとしたら、条件が厳しいかもしれない。POP3 に対応していれば必ずしも WindowsPC でなくともよく、Macintosh などからでもメールサービスは利用できる。従って、情報処理センターのパソコンルームのみならず、現在構築中の経済学部構内の Macintosh ルームなどからも利用できる状況になろう。

#### 4. 学生用電子メールサーバの動作検証

本節では電子メールサーバの動作チェックを sendmail および popper を中心に行い、サーバとしての leibniz の機能を検証する。

#### 4.1 sendmail の動作例と sendmail.cf のチェック

MTA である sendmail はメールサーバを実現するサーバソフトウェアであり、通常はデーモンとして利用されるが、直接起動してメール配送の動作を確認したり、sendmail の設定ファイルを新規に作成した場合のチェックなどにも利用できる。まず、sendmail がどのようにメール送信しているか、その動作例を以下に示す。sendmail を v (verbose) オプション付きで起動すると、メールを送信すべき相手マシンとのネゴシエーションの様子が詳細に表示される。

```
43 leibniz root> /usr/ucblib/sendmail -v imai@ec.kagawa-u.ac.jp
subject: test mail from leibniz
A Happy New Year!
This is a test mail from leibniz to fourier.
1996/01/09

imai@ec.kagawa-u.ac.jp... Connecting to fourier.ec.kagawa-u.ac.jp (smtp-n)...
220 fourier.ec.kagawa-u.ac.jp Sendmail 5.67+1.6W/6.4J.6-1.00 ready at Tue, 9
Jan
96 21:17:36 JST
>>> HELO leibniz cc.kagawa-u.ac.jp
250 fourier.ec.kagawa-u.ac.jp Hello leibniz.cc.kagawa-u.ac.jp, pleased to meet
you
>>> MAIL From: <root@leibniz.cc.kagawa-u.ac.jp>
250 <root@leibniz.cc.kagawa-u.ac.jp>... Sender ok
>>> RCPT To: <imai@ec.kagawa-u.ac.jp>
250 <imai@ec.kagawa-u.ac.jp>... Recipient ok
>>> DATA
354 Enter mail, end with "." on a line by itself
>>> .
250 Ok
>>> QUIT
221 fourier.ec.kagawa-u.ac.jp closing connection
imai@ec.kagawa-u.ac.jp... Sent
44 leibniz root>
```

このメールが相手マシン(メールサイト, すなわち ec.kagawa-u.ac.jp)である

fourier に届いているかどうか確認するため以下のように fourier に telnet でログインし、受信メールを読み出すと、

```
(fourier) imai[1] mailx
mailx version 4.2 Type ? for help.
"/var/mail/imai": 1 message 1 new
>N 1 0000-Admin(0000) Tue Jan 9 21:17:16 16/639 test mail from leibniz
? 1
Message 1:
From root@leibniz.cc.kagawa-u.ac.jp Tue Jan 9 21:17:37 1996
Received: from leibniz.cc.kagawa-u.ac.jp by fourier.ec.kagawa-u.ac.jp (5.67+1.6W/6.4J 6-1.00)
    id AA01882; Tue, 9 Jan 96 21:17:37 JST
Received: by leibniz.cc.kagawa-u.ac.jp (5.67+1.6W/6.4J 6-1.00)
    id AA02346; Tue, 9 Jan 96 21:19:32 JST
Date: Tue, 9 Jan 96 21:19:32 JST
From: root@leibniz.cc.kagawa-u.ac.jp (0000-Admin(0000))
Return-Path: <root@leibniz.cc.kagawa-u.ac.jp>
Message-Id: <9601091219.AA02346@leibniz.cc.kagawa-u.ac.jp>
Subject: test mail from leibniz
Apparently-To: imai@ec.kagawa-u.ac.jp
Status: R
Content-Length: 76
```

A Happy New Year!

This is a test mail from leibniz to fourier.  
1996/01/09

? q

```
Saved 1 message in /home/PRO/imai/mbox
(fourier) imai[2]
```

となっており、確かにメールが正しく送信できていることが確認できる。

次に、sendmail の設定ファイルの記述が仕様を満たしているかどうかをチェックする手順について具体的に示す。チェックすべき点は、

1. ユーザ名だけを指定してメールを送信すればどこに送信されるか

2. ユーザ名@leibniz を指定してメールを送信すればどこに送信されるか
3. ユーザ名@FQDN (例えば, souri@sh.kantei.go.jp) を指定してメールを送信すればどこに送信されるか
4. ユーザ名@stu.kagawa-u.ac.jp へのメールが正しく leibniz で受信できるか

などである。以下のように bt オプションで sendmail を起動すると、ルールセットと送信先の電子メールアドレスのペアを入力して、仕様通りに送信が可能かどうかチェックすることができる。

```
44 leibniz root> /usr/ucblib/sendmail -bt -C/var/ucblib/sendmail.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
> 0 user
rewrite: ruleset 3 input: "user"
rewrite: ruleset 8 input: "user"
rewrite: ruleset 8 returns: "user"
rewrite: ruleset 3 returns: "user"
rewrite: ruleset 0 input: "user"
rewrite: ruleset 0 returns: "$#" "local" "$:" "user"
> 0 user@leibniz
rewrite: ruleset 3 input: "user" "@" "leibniz"
rewrite: ruleset 8 input: "user" "@" "leibniz"
rewrite: ruleset 8 returns: "user" "@" "leibniz"
rewrite: ruleset 6 input: "user" "<" "@" "leibniz" ">"
rewrite: ruleset 6 returns: "user" "<" "@" "LOCAL" ">"
rewrite: ruleset 3 returns: "user" "<" "@" "LOCAL" ">"
rewrite: ruleset 0 input: "user" "<" "@" "LOCAL" ">"
rewrite: ruleset 9 input: "user"
rewrite: ruleset 3 input: "user"
rewrite: ruleset 8 input: "user"
rewrite: ruleset 8 returns: "user"
rewrite: ruleset 3 returns: "user"
rewrite: ruleset 0 input: "user"
rewrite: ruleset 0 returns: "$#" "local" "$:" "user"
```

```

rewrite: ruleset 9 returns: "$#" "local" "$:" "user"
rewrite: ruleset 0 returns: "$#" "local" "$:" "user"
> 0 souri@sh.kantei.go.jp
rewrite: ruleset 3 input: "souri" "@" "sh" "" "kantei" "" "go" "" "jp"
rewrite: ruleset 8 input: "souri" "@" "sh" "" "kantei" "" "go" "" "jp"
rewrite: ruleset 8 returns: "souri" "@" "sh" "" "kantei" "" "go" "" "jp"
rewrite: ruleset 6 input: "souri" "<" "@" "sh" "" "kantei" "" "go" "" "jp">"
rewrite: ruleset 6 returns: "souri" "<" "@" "sh" "" "kantei" "" "go" "" "JPTOP"
">"
rewrite: ruleset 3 returns: "souri" "<" "@" "sh" "" "kantei" "" "go" "" "JPTOP"
">"
rewrite: ruleset 0 input: "souri" "<" "@" "sh" "" "kantei" "" "go" "" "JPTOP"
">"
rewrite: ruleset 6 input: "souri" "<" "@" "sh" "" "kantei" "" "go" "" "
JPTOP" ">"
rewrite: ruleset 6 returns: "souri" "<" "@" "sh" "" "kantei" "" "go" "" "JPTOP"
">"
rewrite: ruleset 7 input: "souri" "<" "@" "sh" "" "kantei" "" "go" "" "JPTOP"
">"
rewrite: ruleset 7 returns: "<" "@" "MX-look-up" "" "smtp-n" "" "mailer" ">"
"." "souri" "<" "@" "sh" "" "kantei" "" "go" "" "jp" ">"
rewrite: ruleset 27 input: "<" "@" "MX-look-up" "" "smtp-n" "" "mailer" ">"
"." "souri" "<" "@" "sh" "" "kantei" "" "go" "" "jp" ">"
rewrite: ruleset 27 returns: "<" "@" "MX-look-up" "" "smtp-n" "" "mailer" ">"
"." "souri" "<" "@" "sh" "" "kantei" "" "go" "" "jp" ">"
rewrite: ruleset 0 returns: "$#" "smtp-n" "$@" "sh" "" "kantei" "" "go" "" "jp"
"$:" "souri" "@" "sh" "" "kantei" "" "go" "" "jp"
> 0 user@stu.kagawa-u.ac.jp
rewrite: ruleset 3 input: "user" "@" "stu" "" "kagawa-u" "" "ac" "" "jp"
rewrite: ruleset 8 input: "user" "@" "stu" "" "kagawa-u" "" "ac" "" "jp"
rewrite: ruleset 8 returns: "user" "@" "stu" "" "kagawa-u" "" "ac" "" "jp"
rewrite: ruleset 6 input: "user" "<" "@" "stu" "" "kagawa-u" "" "ac" "" "jp"
">"
rewrite: ruleset 6 returns: "user" "<" "@" "LOCALA" ">"
rewrite: ruleset 3 returns: "user" "<" "@" "LOCALA" ">"
rewrite: ruleset 0 input: "user" "<" "@" "LOCALA" ">"

```



```

rewrite: ruleset 9 input: "user"
rewrite: ruleset 3 input: "user"
rewrite: ruleset 8 input: "user"
rewrite: ruleset 8 returns: "user"
rewrite: ruleset 3 returns: "user"
rewrite: ruleset 0 input: "user"
rewrite: ruleset 0 returns: "$#" "local" "$:" "user"
rewrite: ruleset 9 returns: "$#" "local" "$:" "user"
rewrite: ruleset 0 returns: "$#" "local" "$:" "user"
> ^D

```

上記のチェック手順ではプロンプト(>)の後に "0 souri@sh.kantei.go.jp" などと入力している。ここに、ルールセット番号0を指定することで、souri@sh.kantei.go.jp という電子メールアドレスへの送信が、sendmailによってどのように処理されるかを具体的に表示させることが可能である。すなわち、チェックの結果から、

- “MX-look-up” という表示からネームサーバの MX<sup>(15)</sup> レコードを参照して A レコード (IP アドレス) を取得する処理が行われている
- メールを配信するエージェントを示す “\$#” が “smtp-n” なる SMTP を解釈実行するデーモンである
- 配信エージェントがメールを届ける相手先のホスト名を示す “\$@” が “sh.kantei.go.jp” であり、IP ダイレクトに送信可能である
- 配信エージェントがメールを届ける相手先のユーザ名を示す “\$:” が “souri@sh.kantei.go.jp” である

などが検証でき、正確にメールが送信できることが確認できる。

## 4.2 popper の動作チェック

続いてMTAを補完し、パソコンへのメールの送信を可能にする popper の動作確認について述べる。TCP アプリケーションである popper は同じく

(15) /etc/resolv.conf で指定されたネームサーバ(ここでは 133.92.6.17)が照会される。

TCP アプリケーションである telnet コマンドを利用して直接起動し、その動作をチェックすることができる[8]。以下にチェックの手順を述べる。

```
45 leibniz root> mailx root
Subject: test for popper
This is a test mail for popper to process mail.
1996/01/10
EOT
46 leibniz root> mailx
mailx version 4.2mp  Type? for help.
"/var/mail/root": 1 message 1 new
>N 1 0000-Admin(0000)  Wed Jan 10 10:01    13/378  test for popper
? q
Held 1 message in /var/mail/root
47 leibniz root> telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK UCB Pop server (version 1.831 beta+0.8 LFO) at leibniz starting.
user root
+OK Password required for root.
pass xxxxxxxx ←実際はパスワードがエコーバックされる(ここでは xxx. と伏せ字にする)
+OK root has 1 message(s) (388 octets).
stat
+OK 1 388
list
+OK 1 messages (388 octets)
1 388

retr 1
+OK 388 octets
Received: by leibniz cc. kagawa-u. ac. jp (5.67+1.6W/6.4J. 6-1.00)
        id AA02970 ; Wed, 10 Jan 96 10:01:42 JST
Date: Wed, 10 Jan 96 10:01:42 JST
From: root (0000-Admin(0000))
```

```
Message-Id: <9601100101.AA02970@leibniz.cc.kagawa-u.ac.jp>  
To: root  
Subject: test for popper  
Content-Length: 60  
Status: 0
```

```
This is a test mail for popper to process mail  
1996/01/10
```

```
stat  
+OK 1 388  
dele 1  
+OK Message 1 has been deleted.  
stat  
+OK 0 0  
quit  
+OK Pop server at leibniz signing off.  
Connection closed by foreign host.  
48 leibniz root>
```

まず、MUA の 1 つである UNIX コマンド mailx でテスト用のメールを作成し、ユーザ名 root へ送信する。MUA である mailx は受信メールを読むことも可能であるが、ここでは受信していることだけを確認している。受信メールを読み出すには popper を利用する。コマンド telnet で leibniz 自身にアクセスする。通常 telnet はサービスポート番号 23 にアクセスするが、popper のテストのためポート番号 110 にアクセスしている。このように popper を直接起動することで、動作を確認することが可能である。パスワードなどを入力して正当な root ユーザであることを示し、受信メールが存在することを確認する。その後、UNIX 上の受信メールをパソコン上に回収・検索 (retrieve) し、UNIX 上から削除して終了している。この結果、受信メールは UNIX マシンからパソコンへ転送されたことになる。但し、上記テストは leibniz 上で実行されている。popper は TCP アプリケーションとしてアクセスされた時点で起動され、呼

び出されたユーザとの間で必要な情報交換を行った後、即座に終了することができる。ソフトウェア自身も小さいが、複数ユーザから利用されても、時系列的アクセスであれば UNIX マシンへの負担も少ない。1 台のメールサーバを多数の利用者が任意の時間にアクセスするというサービス構造には適しており、パソコンユーザが電子メールサービスを利用する上で重要な役割を果たす。しかし、ユーザのパスワードをエコーバックしているなど機密保護機能が乏しい点もある。恐らく今後も改良が試みられる可能性が高いと思われる。popper に対する主なコマンドを表 4-1 に示す。

表4-1 popper の主要コマンド一覧

コマンド名	機 能 説 明	備 考
user	ユーザ名の文字列を入力	
pass	ユーザのパスワードの文字列を入力	エコーバックに注意
stat	状態(受信メールの存在など)の確認	
list	受信メールの個数やサイズなどを表示	
retr	番号を指定してメイン内容を回収・検索	retr と dele とを組み合わせて転送を実現
dele	番号を指定して受信メールを UNIX から削除	
quit	popper の終了	ログインも終了
help	オンライン help メッセージの表示	

## 5. おわりに

既に述べたように、学部毎にメールサーバが置かれ、事務系職員を対象としたメールサーバも稼動し始めた現在、教職員と同様、学生全体にも電子メールサービスを提供する時期を真剣に検討すべき段階に来ていると判断される。学生用メールサーバの試験を開始したのはこのような検討を想定した時、技術的側面からの資料作りを行なうためであった。一方、試験的メールサーバと、将来構築する必要のある実用的メールサーバとは規模の点で大きな違いが存在する。試験的メールサーバで得られた情報は、テクニカルな事項が中心であり、運用時に発生すると思われる問題点の収集が今後の課題である。ユーザ登録な

ど一部のシステム管理を学生に任せていることで、教官負担を軽減する試みも結果がまだでていない。確かにシステムは動作し、サービス自体は実現できたが、利用者を100名規模に拡充して再度調査してみたいと思っている。

leibniz 自体が試験的運用を開始したばかりであるため、利用状況などの調査は十分ではない。学生がどのような情報交換を行うのか、どのような指導が必要か、教育効果はあるのか、など学生用メールサーバの利用形態について考察しておく必要がある。学生がメールサービスを利用することで、

#### 1) 個人対個人の情報交換

教 官 対 学 生…小規模な資料(メモや数頁の文書など)配布に便利、  
質疑応答、意見交換

事 務 官 対 学 生…小規模な資料(メモや数頁の文書など)配布に便利、  
呼び出し

学 生 対 学 生…約束や相談に便利 リアルタイムな情報交換、プライベートな利用も可

#### 2) 複数(個人)対複数

教 官 対 ゼ ミ 生…授業やイベントの連絡、予定の確認や意見交換

顧問対クラブ学生…クラブのスケジュール調整、ミーティングの設定

事 務 官 対 学 生…小規模な資料の一斉配布に便利、呼び出し

メイリングリストの構成…申請のあったユーザのグルーピングとメール配送  
の管理

ゼ ミ 生 同 士…ゼミの先輩(OB, OG)からの情報入手、プライベートな  
利用も可

ク ラ ブ 生 同 士…クラブの先輩(OB, OG)からの情報入手、プライベート  
な利用も可

などの様々な形態の情報交換が実現できる。電子メールサービスのプライベートな利用も許可するかどうかを議論すべきであるが、学生の自主性を伸ばすためには管理者組織を構成してある程度の権限を委譲すべきとも考えられる。香川大学全体としても情報化推進が叫ばれている現在、中長期的には学生が情報

通信環境を積極的に利用する状況は間違いなく到来するであろう。その時点までに整備すべき問題も多い。

学生のみならず教職員をも包括できる相互教育指導体制も検討すべき課題であろう。情報通信環境の現状を適切に理解し、調和のとれた利用状況が望まれる。そのためには、例えば、

- MUA の利用方法の指導体制
- 学生ユーザに対する技術教育としつけ教育
- 研究室(ゼミ)配属後の学生とそれ以前の学生への指導・責任体制
- 就職活動と電子メールの取り扱い
- 大学院生(特に社会人大学院生)と学部学生との線引き

など対応が急務であろう。選抜された学生を対象としたユーザ登録から、学生全員のユーザ登録へ移行する過程で注意すべき問題も多数ある。今後は追跡調査などの試みも必要となろう。

## 謝 辞

経済学部教授、中村邦彦先生には MTA および MUA の設定や利用方法など様々な点について唆に富んだご助言を賜った。経済学部情報管理学科の3年生の市川玲君、伏見修君、村口征応君などには学生用メールサーバの運用形態に関して貴重な情報を提供していただいている。記して感謝の意を表します。

## 参 考 文 献

### ★UNIX 全般に関して

- [1] 山口和紀監修『The UNIX Super Text(上)(下)』技術評論社, 1992
- [2] Fielder/Hunter/Smith(中原紀訳)『UNIX SVR4 システム管理』HBJ 出版局, 1993

### ★ネットワーク/LAN/インターネットに関して

- [3] 石田晴久『コンピュータ・ネットワーク』岩波新書, 1991
- [4] 石田晴久『はやわかりインターネット』共立出版, 1994
- [5] 村井純/吉村伸監修『インターネット参加ノ手引き 1994 年度版』共立出版, 1994
- [6] 吉田茂樹, 森秀和, 杉岡隆司『インターネット漂流記』オーム社, 1994

- [7] 松田徹『初めてのインターネット』西東社, 1995
- [8] M. R. Arick(竹内俊博訳)『TCP/IP ビギナーズガイド』トッパン, 1994
- [9] グループまたたび『インターネット情報生活入門』技術評論社, 1994
- [10] 日経コミュニケーション編『企業ユーザのためのインターネットハンドブック』日経 BP社, 1994
- [11] Hunt(村井監訳)『UNIX システム管理者のための TCP/IP ネットワーク管理』オーム社, 1994
- [12] 瀬川至朗他『これからのメディアとネットワークがわかる事典』日本実業社, 1995

★電子メールに関して

- [1] 山口和紀監修『The UNIX Super Text(上)(下)』技術評論社, 1992
- [5] 村井純/吉村伸監修『インターネット参加ノ手引き 1994 年度版』共立出版, 1994
- [13] 山本和彦『ハッピー・ネットワーキング—新入生のためのインターネット入門』アスキー, 1994
- [14] Costales/Allman/Rickert(村井監訳)『sendmail 解説』オーム社, 1994
- [15] Schoonover/Bowie/Arnold(小畑他訳)『GNU Emacs 完全ガイド(上)(下)』トッパン, 1992