

1. はじめに

長崎大学情報処理センターを利用するユーザ間での電子メールの送受信を実現するコマンド、`mail`を開発した。

電子メールは、コンピュータネットワークを利用した情報交換システムのひとつであり、郵便や電話と並んで、情報交換の重要な手段となりつつある。

電子メールの特長としては、まず、ユーザの都合のよいときにメールの発信、受信ができることがあげられる。電話のように受信者と発信者が同時に電話口に立つ必要がなく、情報交換が成立しやすい。また、複数の受信者にメールを出すことや、受信したメールの返事を出すことが簡単に行えることも特筆される。さらに、メールをコンピュータ上に保管できるので、メールの再利用が簡単にできる。等々、電子メールの有効性は、枚挙に暇がない。

既存の電子メールシステムとしては、UNIX上のメールシステムが広く知られている。今回のプログラムは、UNIXのメールシステム中でも、最もスタンダードであると思われる`Mail`コマンドの動作を参考にした。しかし、プログラムの動作する環境が異なるため（かたやUNIX、かたやFACOM OS IV）、全く同じ動作を実現するわけにはいかず、若干、機能を縮小した部分もあり、あると便利と思われる機能をつけ加えた部分もある。

たとえば、外部ネットワーク間との電子メールの送受信はサポートしておらず、可能であるのは学内の電子メールだけである。この問題には、時節が到来次第、取り組むつもりである。

拡張した部分としては、制限付きながら日本語データを扱えるようにしたこと、及び、メールに書留モードを導入したことがあげられる。

以下では、この`mail`コマンドの動作、使用法、及び、プログラムの概要について、紹介する。

この`mail`コマンドによって、長崎大のコンピュータユーザが、より円滑に、よりアクティブに情報交換ができるようになり、より有機的なつながりが持てるようになれば幸いである。

2. `mail`コマンドの動作

このメールシステムでは、各ユーザのメールボックスをシステムの管理するデータセット中に保持している。

メールを発信するとは、所定の受信者用のメールボックスの最後尾にメールをアペンドする事である。メールには、発信者が伝えたいと思うメッセージに、メールが發送された日時など数種の情報がヘッダとして自動的に付加される。Subjectフィールドは、メールの発信者によって、意図的にヘッダに加えられる情報である。このフィールドはメールを分類するに有効な項目となるから、発信者は必ず、メールに適当なSubjectフィールドを付加するべきである。Subjectフィールドの付加の仕方については後述する。

メールを受信するとは、システムが管理するデータセット中の自分用のメールボックスをオープンし、メールを読みだす、あるいは、様々な作業をすることである。通常の動作では、mailコマンドの終了直前までに参照したメールは、受信者の管理するデータセット中のmbox, textの最後尾にアペンドされ、システムの管理するメールボックスからは削除される。参照されなかったメールは引続きシステムの管理するメールボックス中にとどまる。書留のメールを参照したときは（書留のメールの出し方は後述）、受信者にその書留メールに対する返答をシステムが強要する。システムは直ちに、その返答を書留メールを発信したユーザに送り返す。書留メールを参照せずにmailコマンドを終了しようとしたときは、システムが自動的に書留メールを表示し、受信者に返答を強要し、その返答を發送した後、mailコマンドを終了する。

3. mailコマンドの使用法

メールの發送、受信の仕方は、ほぼ、UNIX 4.2BSDのMailコマンドに準じている。しかし、長崎大学内には、大文字端末が無視できない数だけ設置されており、大文字に慣れたユーザも相当数いると考えられるため、コマンドについては、大文字小文字の区別をしないことにした。そのしわ寄せで、一部、UNIXのMailとはコマンドのキーが異なっているものもある。

3. 1 メールシステムへの登録

現段階では、メールシステムに参加するために、情報処理センターでの事務的な手続きが必要である。これは、将来、端末からその意志を表明するだけで済むようにしたいものである

メールシステムへの登録を完了したユーザは、長崎大学内でユニークなアドレスを獲得し、システムが管理するデータセット中に自分専用のメールボックスを持つことになる。

3. 2 メールのおし方

メールを発送するためのコマンド形式は、

```
mail 受信者名の並び オプションの並び
```

である。

通常、システムは、

```
Subject :
```

と、聞いてくるので、メールの表題として適当と思われる文字列を入力する（リターンキーで入力終了）。その後、伝えたいメッセージを入力する。メッセージは複数行にわたってよく、空行、あるいは、行頭の” ”によって入力が終了する。

受信者名は、システムに登録してある名前か、発信者が別名登録機能（後述）によって定義した名前であればならない。受信者名は複数にわたってもよく、その場合、一つ以上のブラックで区切る。受信者名の並び中に、メールシステムに登録されていない名前を見つけた場合は、メールはdead. textにセーブされる。

オプションとして、”-a l l”を指定すると、システムに登録されている全ユーザにメールを発送する。”-f =ファイル名”を指定すると、端末から入力するメッセージのかわりに、ファイルの内容がメールとして発送される。”-s =文字列”を与えると、その文字列がメールの表題になる。このとき、システムは、メールの表題の入力を促さない。”-m =r”は、メールを書留にするオプションである。

たとえば、ユーザ全員に、test. textファイルの内容を、”test”という表題で、さらに、書留モードにて発送したい場合は、

```
mail -a l l -f=test. text -s=test -m=r
```

とする。

3. 3 メールを受取り方

メールを受信する処理を行なうには、

```
mail
```

を実行する。

メールボックス中に届いているメールがなければ、たとえば、

```
No mail for kimura@kyoyo
```

を出力してメールコマンドは終了する。

そうでない場合、メールボックス内のメールの一覧表を出力した後、プロンプト (&) を出して、ユーザの入力を待っている。たとえば、次のようなものである。

```
>1 U suzuki@kyoyo 02/17/88 21:00:00 a
  2 N sato@kougaku 02/18/88 22:00:00 b
  3 R tanga@igaku 02/18/88 23:00:00 c
&
```

1 番目のメールは、Unresolved、すなわち、以前、mail を実行した時すでにメールボックス中にあったが参照しなかったメールであり、教養の鈴木氏から、88年2月17日21時00分00秒に届いた、a という表題のメールであることを示している。以下、同様であるが、N は New、すなわち、新着のメールであることを表わし、R は新着であり、かつ、書留 (Registered) であることを表わしている。

"1" の左の ">" は、リターンコマンド (後述) により、次に参照されるメールを指している。

リターンコマンド等により参照が行なわれたメールの場合、この一覧表から、U、N、もしくは、R の表示は消える。

d コマンド (後述) により削除されたメールのエントリは、この一覧表から消える。

以下で、メールの受信時のコマンドについて説明する。

リターン

メールボックスから次のメールを取り出し、表示する。

数字

その番号のついたメールを取り出し、表示する。

d

直前に参照したメールをメールボックスから消去する。

f 受信者名の並び

直前に参照したメールを他の受信者に転送する。現在のバージョンでは、受信者名に別名(後述)は使えない。転送するメールは書留にできない。

h

メールボックス中のメールの一覧を表示する。メールはdコマンドによって削除されていないものに限る。

m 受信者名の並び オプションの並び

メールを送信する。別名(後述)の使用可。

p

直前に表示したメールをもう一度表示する。

q

mailコマンドを終了する。参照がすんだメールは、システムの管理するメールボックスから削除され、通常は、受信者のデータセット中のmb ox. t e x t にアペンドされる。参照されなかった書留メールがあった場合、それを表示した後、受信者に返答を強要し、その返答を書留メールの発信者に自動的に送信する。現在のバージョンでは、返答のメールは書留にできない。

r

直前に参照したメールの送信者に返事を出す。現在のバージョンでは、返事を書留にはできない。

s ファイル名

直前に参照したメールをファイルにセーブする。ファイル名が与えられなかった場合には、mb ox. t e x t ファイルにアペンドする。

t

メールボックスの先頭に戻る。

w ファイル名

直前に参照したメールの文書部分のみをファイルにセーブする。ファイル名が与えられなかった場合には、mb ox. t e x t ファイルにアペンドする。

x

メールボックスを更新せずに、mailコマンドを終了する。mbox, textファイルへの書き出しも行なわない。

!文字列

文字列をFACOM OS IVのコマンドとして実行する。コマンドが終了次第、mailコマンドに復帰する。

?

ヘルプメッセージを表示する。

3. 4 mailコマンドの環境設定

mailコマンドは、起動時にmailコマンドを発行したユーザのデータセット中にmailrc, dataファイルを探す。もし、あれば、そのファイルに書き込まれた情報に従ってmailコマンド自身の環境を設定する。

3. 4. 1 別名登録機能

mailrc, dataファイル中に、

```
alias a kimura@kyoyo  
alias b uchimoto@center  
alias c a b
```

があれば、メール発送時の、

```
mail c
```

と、

```
mail kimura@kyoyo uchimoto@center
```

は、等価である。つまり、mailコマンドは、“c”を、“a b”に、“a”、“b”をそれぞれ、“kimura@kyoyo”、“uchimoto@center”に置き換える。

置き換えによる無限ループが生じないようにユーザは注意すべきである。

3. 4. 2 その他

mailrc. data中に、

```
set ask
```

があれば、メールの発送時に、自動的に、"Subject:"を聞いてくる。これを抑制するには、"set noask"を代わりにいれておく。

```
set hold
```

は、参照しなかったメールをシステムのメールボックス内にとどめるモードに設定する。とどめたくない場合には、"set nohold"を指定する。

```
set reply
```

は、qコマンドによる終了時に、参照していない書留メールがあれば、受信者に知らせ、返事を出すモードに設定する。"set noreply"は、それを抑制する。

mailコマンドのデフォルト値は、

```
set ask
```

```
set hold
```

```
set reply
```

である。

3. 5 日本語の扱いについて

このメールシステムでは、メッセージ部ばかりでなく、Subjectフィールドにも日本語を使うことができる（これは、JUNET等、外部のネットワークとの接続を考える場合には問題となるかも知れないが）。

パソコン端末で日本語メールの送受信を行う場合には、漢字コードを旧JISにて送受信できるような端末プログラムが必要である。日本語フロントエンドプロセッサにATOK6、端末プログラムはCTERMを用いて実験してみたが、うまくいくようである。

センターの日本語端末を使用する場合、日本語メールの受取には、ほとんど、何も意識する必要はない。日本語メールを発信するには、日本語文書をファイルに作成する必要がある。たとえば、ODMコマンドで日本語文書を作成し、ラインナンバなしでセーブする。そのファイルを、“-i=”オプションで指定するとうまくいく。

その他の組み合わせについては、各ユーザで試してほしい。

4. 作業環境

このプログラムは、C言語で記述した。

大型コンピュータ上のCコンパイラには、ソースコードのインクルードができないという、かなりの痛手があった。さらに、日本語の扱いに関して、マニュアルに記載してある動作と異なる動作をする部分もあった。何と言っても、エディタの使い心地が悪く、結局、使い慣れたパソコンでプログラムを開発することにした。

マイコン上で、エディット、コンパイル、テストラン、デバッグを行い、出来上がったソースコードを、一部、大型コンピュータ用に書き換えた後、電話回線で大型コンピュータへ送った。そのコードを、大型コンピュータでコンパイルして、最終的に、大型コンピュータ上で動作するmailコマンドを得た。

5. プログラムの説明

コーディングにあたっては、処理スピードやオブジェクトサイズよりも、移植性の良さと、プログラムの見通しの良さを重視した。

まず、マシンに依存するコード部分をひとつのファイルにまとめた。その中に記述しているのは、プログラムの動作するマシン上で日本語を処理するに必要なデータ構造の定義と、入出力関数など、極めて基本的なものである。

また、プログラム内で扱う抽象データ構造の定義と、そのデータをアクセスする関数を、抽象データ構造ごとにひとつのファイルにまとめ、情報隠べいに心がけた。

メイン関数など、上位の関数は、それらのファイルをインクルードして使うことにして、データ構造の細かいインプリメンテーションを知らずに処理を行えるようにした。これによって、あとあと、データ構造を換えたり、データ構造へのアクセス法を変えても、上位の関数は全く影響を受けずにすむ。

具体的には、以下のようになっている。

mail. c	メールコマンドのメインルーチン
config. h	メールコマンド内で使う定数、大域変数等の宣言
prims. c	システムコール等、マシンに依存する低レベル関数の集まり
text. c	テキスト型、及び、それにアクセスする関数の定義
usr. c	ユーザ型、及び、それにアクセスする関数の定義
header. c	ヘッダ型、及び、それにアクセスする関数の定義
send. c	メールを発送する機能を実現する関数の定義
recv. c	メールを受信する機能を実現する関数の定義
opts. c	メールコマンドのオプションを処理する関数

このプログラムをMS-DOS上のMS-Cで記述し、動作を確かめた。その後、“config. h”と、“prims. c”を、大型コンピュータ用書き換え、すべてのソースコードをひとつのファイルに展開した上で（大型コンピュータのCコンパイラはソースコードのインクルードができない）、大型コンピュータへ転送した。

言い替えば、そのふたつのファイルの内容を、このmailシステムを実現したいマシン用に置き換えるだけで、このプログラムを異なるマシンに移植する作業は完了する。ファイル中に記述してある事柄は非常にシンプルであり、ファイルのサイズも小さく、書き換えずすむ部分も多いので、移植のための労力はかなり少ないはずである。もちろん、ターゲットマシンには、Cコンパイラが動作していることが必要であるし、プログラムの性質上、マルチユーザで稼動できるマシンでなければ意味がない。

このmailコマンドが多くのユーザに使ってもらえ、センターのホストマシンが新しくなっても存続させてもらえるようになったら、また、たくさんのユーザに改良を加えて頂けるものであれば、全くの幸せである。筆者としては、このプログラムのソースコードを公開するつもりであり、このプログラムに対する（なるべく建設的な）ご意見、ご質問等が、このmailコマンドによって筆者に届くようであれば、筆者には2倍の喜びとなるだろう。

6. おわりに

このプログラムを書くにあたって、長崎大学教養部の田井村氏には、漢字コードなどの問題について、よく相談にのって頂いた。また、情報処理センターの教職員の方々にも、筆者の突飛な質問や相談で、たいへんご迷惑をおかけした。特に、阪上、内本の両氏のご協力なくては、

このプログラムを書き上げるにはいたらなかった。紙面をお借りして、深く、感謝の意を表します。

<参考文献>

- 1) 富士通、"FACOM OS IV C言語文法書"、V10L20系用
- 2) B. W. カーニハン、D. M. リッチー著、石田晴久訳、"プログラミング言語C"、共立出版、昭和56年。
- 3) 石田晴久他編、"最新UNIX"、bit臨時増刊、共立出版、1987。
- 4) Jhon Allen、"Anatomy of LISP"、McGraw-Hill、1978。
- 5) Kernighan、B. W. & P. J. Plauger、"Software Tools in Pascal"、Addison-Wesley、1981。
- 6) Aho、A. V. etc、"Data Structures and Algorithms"、Addison-Wesley、1983。
- 7) Comer、D. 、"Operating System Design—the XINU approach—"、Prentice-Hall、1984。