

JUNET への接続手順と設定

寺元貴幸* 河合雅弘* 中尾三徳† 岡田正*

The process and setting for a connection to the JUNET

Takayuki TERAMOTO*, Masahiro KAWAI*,
Mitsunori NAKAO† and Tadashi OKADA*

The JUNET is one of the most valuable computer network system in the university and research environments of Japan. At January of 1992, about 600 organizations including universities, colleges and private enterprise were connected by mainly UUCP protocols. We participated in the JUNET to use for research work and education. In this paper we report a process and setting for the connection to the JUNET. The connecting procedure using the UUCP and the implementation of a mail or news system are described in detail for the purpose of reference.

1. はじめに

津山高専では現在学内ネットワークの設置が進行しており、すでに情報工学棟¹⁾と図書館棟はそれぞれのネットワークを形成し、研究・教育に十分な能力を発揮しつつある。また、図書館事務室には文献検索用のデータベース「JICST」に接続するためのネットワーク「JOIS」も導入され、文献検索において大きな成果を上げている。さらに将来的には、本館も含めた全校ネットワークが計画されており、これが実現すれば学内の情報交換環境は非常に充実したものになることが予想される。

学内ネットワークの充実が続いて、津山高専のネットワークを学内から学外へ広げることが重要である。ネットワークの利用価値は、その規模が大きければ大きいほど飛躍的に増大し、その波及効果も期待できるからである。そのため、学内だけで閉じているネットワークから抜け出し、全国規模のネットワークにアクセスすることで、より価値の高いネットワークが実現できる。この目的に最も適しているネットワークが「JUNET」²⁾であると考えられる。現在日本全国に広がるJUNETは、研究・教育にとって欠くことのできない存在になっており、最先端の情報がここを幹線として縦横無人に飛び交っている。1992年1月現在で国内の加入組織は600を数え、JUNETから海外へのアクセスが可能なサイトを経由すれば、世界的には加入組織44,000、アカウント9,729,000、読者2,128,000が推測されている。(ただし、海外へのアクセスが可能なサイトで、通信を行うための権利を取得する必要がある。)

つまりJUNETにアクセスすることができれば、世界の情報がほぼリアルタイムで入手でき、しかもそれに直接

参加することが可能となる。また、電子メールのシステムを使えば、全国の研究者に直接電子メールを出すことにより、意見の交換・討論さらにはファイルの転送などが可能となる。これらの効果は地方の高専が抱える地理的に不利な条件をかなり軽減し、さらに情報のネットワークを世界にまで拡大してくれる可能性を秘めている。

この論文は、わが校がJUNETに接続し運用を開始するまでの経過を、できるだけ詳細に記述し報告することを目的としている。そして今後の運用やシステムの改良時の資料となり、また、これからJUNETに参加を希望する学校(もしくは企業)に役立つことを配慮して書かれている。

2. 接続を始めるまでの準備

2.1 JUNETとは

JUNETに接続するためには、その主旨を組織内の人に理解してもらうことが必要である。JUNETを個人的なレベルで運用することはほとんど不可能に近いので、どうしても組織的な施設や資金を利用する必要が生じる。そのためにも、JUNETの利用目的・価値を広く組織の人にまず知ってもらわなければならない。JUNETの主旨を理解するためには「JUNET利用の手引き」³⁾を入手するのがもっとも簡単である。これは、現在JUNETに接続しているドメインに連絡すれば簡単に入手できる。また、元のファイルがL^AT_EXの形式であるから、JUNET以外のネットワーク等からも簡単に手に入れることが可能となっている。さらにこの「JUNETの手引き」には、実際の利用方法や細かな管理方法なども示されているので、接続前に必ず一通り目を通す必要があるだろう。

「JUNETの手引き」の中に次のような一文があり、これがJUNETそのものを端的に表していると思われるで紹介する。

*情報工学科

†電子計算機室

『JUNETはボランティアネットワークですから、参加している以上はそれに応じた義務を負います。例えばメール、ニュースを隣接サイトへ正しく送る等です。各ドメインの管理者はこの手引きを熟読し、それに沿った運営を行うことが要求されます。また、システムのトラブル等に関しても自分で解決することが原則です。(省略)各組織の管理者は、自分自身もJUNETの一部を担っているのだということを見習いましょう。』

このように、JUNETはボランティアによって支えられていることを常に忘れてはならない。

2.2 接続に必要な機器

JUNETの主旨を理解したところで、組織内の合意をとりつける必要がある。JUNETはボランティアに支えられるネットワークとはいえ、やはり学校(大学もしくは企業)同士のコンピュータを電話回線で結び、直接通信をするのであるから、組織内の公の委員会で合意をとりつける必要がある。また、このように公の機関で承認されれば、接続のための技術的なサポートも非常に受けやすくなり、以後の調整が容易となる。津山高専の場合は、「電子計算機システム運営委員会」で正式な合意を得ている。

この合意を得るためには、必要な物品・予算の見積が必要となる。参考とするために、津山高専の例を述べる。JUNETに接続し、活動を行うために次の機器を準備した。

ワークステーション	SUN SPARCstation1 (SunOS 4.0.3)
モデム	Telebit TrailBreiser T2500
ケーブル	RS-232C
電話回線	外線専用の電話回線

また、運営していく上で重要となるのが電話代である。運営前の段階では、あくまでも予想にすぎないが、簡単な見積として1日500Kバイトのファイルが通信されると想定すれば、年間約5万円程度が見込まれる。これも、接続ポイントまでの距離と購読するニュースの種類や数に依存しているため、最終的には運営しながら様子を見て、決めなければならない。むしろ、予算に応じて通信量を加減することが最も現実的となろう。運営後の詳しい様子については改めて報告したい。

組織内の同意が得られたならば、運営責任者(津山高専の場合は電算機室室長)が接続先(津山高専の場合は岡山大学工学部情報工学科)に出向き正式な申請を行う。このとき、自分のところのサイト名を申請し、他のサイトとの重複がないか確認してもらうことが必要となる。つまり、この時までには正式なサイト名を決めておく必要がある。実際には次のような手順で行った。

「日本ドメイン名の割り当てについて」「組織ドメイン名選択のためのガイド」等を参考にしてドメイン名 tsuyama-ct.ac.jp を決定し、上位サイトである岡大の JUNET 管理者に申し出る。そこを經由して name-admin@nic.ad.jp に申請し、1992/02/21 付けのサイト名承認書を受領した。

さらにこの後、junet-admin@junet.ad.jp宛の登録申請をしなければならないが、そのタイミングについては、岡山大学との接続を確認してから相談の上決定した。

なお、接続までの手続きは、1992年5月より「JUNET協会」が発足したため、現在ではここに述べたことと異なる点があることをお断りしておく。

3. UUCP 接続までの手順

組織的な準備が一応整えば、次にワークステーション(WS)の技術的な設定に入る。これ以降は、かなり機種依存性(もしくはOS依存性)が強いと考えられる。我々が用いたシステムはSPARCstation1(SunOS 4.0.3)であり、これ以外のシステムを使用する場合には、ここに記述した手順がそのまま適用できるとは限らない。むしろ、そのまま使える場合のほうがまれで、同じSPARCstation1でもOSのバージョンが異なればファイル名やディレクトリの構造に変化が生じる。しかし、設定や手順の大きな流れに差異はなく、どんな場合においても何らかの参考になることを期待している。また、特定の機能に関しては資料の不足により、若干説明不足が生じている箇所もある。その場合は参考文献を参照して頂きたい。

3.1 tip コマンドを使用可能にする

設定の最初に tip コマンドを使用可能にする。それは tip コマンドが通信エミュレータとしての命令をもち、これを使って通信を行うためである。さらに重要な役割として、Telebit T2500のように、設定のためのディップスイッチがないモデムでは、この tip コマンドを使ってモデムに AT コマンドを送り出すことによってモデムの初期設定を行うことがある。その手順を以下に示す。

最初にデバイス cua0 の作成について説明する。RS-232C のポート 1 を cua0 として使う場合の設定は、次のように行う。まず、スーパーユーザになり

```
# cd /dev
# mknod cua0 c 12 128
# chown uucp cua0
```

と入力して、デバイスを設定する。ここで mknod コマンドのパラメータの 12 と 128 は通常このままでよいが、念のため、ttya のデバイスナンバーが 12 と 0 であることを確認すること。なお、これ以降の設定は、それぞれの場合によってスーパーユーザ・news・一般ユーザを使い分ける必要が生じる。以後それぞれの場合をプロンプトで区別する。スーパーユーザのプロンプトは #、一般ユーザのプロンプトは %、news のプロンプトは news% とする。

次に /etc/remoted の修正について説明する。今回使うモデムを UNIX 上で使用できるようにするには、いくつかのファイルを修正・追加する必要がある。その最初のファイルがこの /etc/remoted である。まず、オリジナルの /etc/remoted を remote.org として保存し、新しく /etc/remoted に次のアンダーラインの行を追加する。またこれ以後、特に断らない限りファイルの修正・変更を行う場合は、元のオリジナ

ルファイルを別名（*.org等）で保存しておくほうが安全である。

```
cua0:dv=dev/cua0:br # 19200
dialup1—Dial-up system: ¥
.
.
dialers: ¥
:dv =/dev/cua0:
```

ここで、br#直後の数値は転送速度のボーレートを表し、モデムの種類によっては、もう少し低速に設定しなければならない場合もある。

次に/etc/ttytabファイルの中から「ttya」の行を探し、次のように修正する。

```
ttya "/usr/etc/getty TrailBrazier"
dialup off remote secure
```

ここで、TrailBrazierは適当に付けた名前で、/etc/gettytab以外ならどの様な名前でもかまわない。remoteも同様にlocal以外ならなんでも良い。

そして最後に、上記の設定ファイルの内容を有効にする。これは、OSのバージョンがSunOS 4.0.3の場合、eepromコマンドを用いて設定する。⁴⁾

```
# eeprom ttya-ignore-cd false
```

EEPROMの「ttya-ignore-cd」のフィールドは既定値がtrueなので、このままではソフトウェアキャリアディテクトである。そこでこれを変更することで、ハードウェアキャリアディテクトとすることができる。ただし、この設定を有効にするには、一度システムをリブートしなければならない。またOSがSunOS4.1の場合は

```
# kill -HUP 1
# /usr/etc/ttysoftcar -a
```

とする。

3.2 モデムの設定とテスト

次に、モデム「Telebit T2500」の設定をソフトウェア的に行う。これは、tipコマンドを用いてモデムにATコマンドを送信することによって実現する。以下の説明は、Telebit 2000シリーズであれば、レジスタの番号にごく一部違いがあるほか、どの機種でもほぼ同じである。

モデムの設定を行うために、コマンドラインから次のように入力する。ここでイタリック文字が入力すべきコマンドで、それ以外の通常の文字はモデムからの応答を示している。

```
# tip cua0
CONNECT
at
OK
at & F Q6 S50=255,S52=2 S53=1 S54=3
S58=2 S111=30 & W
[ EOT ]
(ここで [~] [.] を入力して tip を終了する。)
#
```

モデムのレジスタ値には、製品の出荷された段階で最も一般的な値があらかじめ入力されている。ここでは、ほとんどその既定値を利用したため、ほとんどのレジスタは改めて設定する必要がない。しかし、別のメーカーのモデムを使う場合などに役立つために、すべてのレジスタ値を以下に示しておく。

S00 = 001	S01 = 000	S02 = 043	S03 = 013
S04 = 010	S05 = 008	S06 = 003	S07 = 059
S08 = 002	S09 = 006	S10 = 007	S11 = 070
S12 = 050	S18 = 000	S25 = 005	S26 = 000
S38 = 000	S41 = 000	S45 = 000	S47 = 004
S48 = 000	S49 = 000	S50 = 255	S51 = 005
S52 = 002	S54 = 003	S55 = 000	S56 = 017
S57 = 019	S58 = 002	S59 = 000	S61 = 150
S62 = 003	S63 = 001	S64 = 001	S65 = 000
S66 = 000	S67 = 000	S68 = 255	S69 = 000
S90 = 000	S91 = 000	S92 = 000	S93 = 008
S94 = 001	S95 = 000	S96 = 001	S97 = 000
S98 = 003	S100 = 000	S101 = 000	S102 = 000
S104 = 000	S105 = 001	S106 = 000	S107 = 020
S110 = 255	S111 = 030	S112 = 001	S121 = 000
S130 = 001	S150 = 000	S151 = 004	S152 = 001
S153 = 001	S154 = 000	S155 = 000	S157 = 000
S158 = 000	S160 = 010	S161 = 020	S162 = 002
S163 = 003	S164 = 007	S169 = 000	S255 = 000

ただし、Telebit 2500ではS53がS131に置き換えられる。

モデムの設定が終了したところで、モデムのテストを行う。テストは手でモデムを回線に接続し、目的のサイトとの通信を確立することで行う。呼出先のホスト名が「upersite」、電話番号が「0123456789」の場合はつぎのように行う。

```
# tip cua0
CONNECT
at
OK
ATDP0123456789
CONNECT FAST
upersite login: (このプロンプトが出れば接続完了)
[ EOT ] ( [~] [.] を押して tip を終了する )
#
```

この時、接続先の電話番号に内線番号(かりに100番)が必要な場合は「ATDP0123456789,,,,,t100」と入力する。通常代表番号から内線に切り替える処理をするために、交換器がいくらか時間を必要とする。そのために待ち時間を入れるための命令が「,」である。このときコマンドは、1個に

つき2秒間の待ちを意味し、5個並べれば10秒の待ち時間を表す。また回線がトーンダイヤルの場合は「t」を付けることで実現する。

3.3 UUCP 関連のファイルの設定

次に UUCP で通信を行うために、種々のファイルの設定を行う。これらのファイルは、それぞれのサイトによって多少異なっている可能性があるため、注意する必要がある。とくにそのファイルの存在するディレクトリは、パスの設定やシンボリックリンクの関係で異なってくるので注意する。

まずファイル `/etc/uucp/L-devices` の修正を行い、UUCP で使用するデバイスを設定する。オリジナルファイルを別名で保存し、次の1行を追加し、`cua0` の使用を可能にする。

```
DIR cua0 0 19200
```

次に `/etc/uucp/L.sys` を設定する。目的とするマシンに接続し、必要な通信を開始するためには、ある一定の手順が必要である。この手順を自動的に行うためには、定められたシーケンスを記述しておくことが欠かせない。そのためのシーケンスを示すのが、このファイルの目的である。ファイルは次のように作成する。

```
uppersite Any cua0 19200 cua0 "" at ¥ r ¥ c OK-AT
¥ r ¥ c-OK a tdp0123456789,,,,,,,,, ¥ r ¥ c
CONNECT-¥ c-CONNECT ¥ c FAST-¥ c-FAST
¥ r ¥ c ogin:-EOT-ogin: tsuyama assword: xxxxxx
```

このシーケンスは、最初にモデムの設定を行い、上位サイトに電話をかけ、接続を試みる。そして接続を確認した後、目的のマシンにログインするという一連の動作を記述している。この時、必要な文字列を送信して、相手側から返信された反応の文字列が求める文字列と一致すれば、接続の次のステップに進むようになっている。返信された文字の認識は、最初の1文字を除いて行っている。つまり「login」を「ogin」、「password」を「assword」としている。これは、通信条件が多少不安定な接続初期において、文字化け等が発生しても安定に接続することを目的とした工夫で、多少の効果が期待できる。また上の記述では、1行文字数の関係で、適当な所で改行しているが、実際のファイルでは改行せず1行に納める必要がある。さらに xxxxxx の部分は、ログイン時のパスワードであり、上位サイトに依頼し適切に設定してもらう必要がある。

その他のファイルで、設定が必要なものについて、ファイル名とその内容を以下に示す。

ファイル `/etc/uucp/L.cmds`

```
rmail
rnews
lpr
uusend
```

ファイル `/etc/uucp/USERFILE`

```
, dsun01 /
, /usr/spool/uucppublic /var/spool/uucppublic
```

3.4 UUCP の接続テスト

UUCP の設定が終われば通常の接続を行う。ただしその前に、正常な接続が確立されるかどうか、デバッグモード (-x オプション) による実験を行ってもよい。接続そのものの内容は変わらないが、接続時の情報を冗長に伝えるのがデバッグモードである。

UUCP 接続のテストは、次の手順で行う。自分のサイト(以下津山高専とする)から外にアクセスし、上位サイトにファイルの転送を行うことで、正常な動作の確認を行う。ファイルの転送は、津山高専から上位サイトへの送信と、反対に上位サイトから津山高専の双方向を個別に行う。送信するファイルはここでは「test」という名前とし、簡単なテキストファイルとする。この時、テストのため英数字だけでなく漢字も含めるとよい。ただし、漢字コードにはさまざまな種類があるが、JUNET では JIS コードが基準となっているため、JIS コードを使うべきである。

最初に津山高専から上位サイトにファイルを転送する。まず、`/usr/spool/uucppublic` にテスト用のファイルとして「test」を用意し、接続先のサイト名が `uppersite` の場合は次のように入力すればよい。

```
# cd /usr/spool/uucppublic
# uucp test uppersite ¥ !/usr/spool/uucppublic/test
```

これで津山高専から上位サイトに、自動的にファイルの転送を行ったことになる。ただし、成功したかどうかはこのままでは分からないので、今送ったファイルを逆に上位サイトから津山高専に返送することで、成功を確認する。

上位サイトから津山高専へのファイル転送は、次のように行う。ただし、先ほど送ったファイル「test」と区別するために、受け取るファイルの名前は「testret」とする。

```
# cd /usr/spool/uucppublic
# uucp uppersite ¥ !/usr/spool/uucppublic/test testret
```

モデムの音を聞きながら、様子を見て接続が終了したら、実際にファイルが正しく送られているか `diff` コマンドを使って確認すればよい。

```
# ls
test testret
# diff test testret
#
```

このように、ふたつのファイルにまったく違いがなければ、津山高専と上位サイトとの間で双方向の転送が確認できることになる。ここで、違いが出る可能性として漢字コードの化けが考えられる。しかし、JIS コードでの通信が確立できれば、今後問題は生じないだろう。

3.5 UUCP の自動化

実際に JUNET を運営していく場合、上位サイトのマシンに接続するのに、毎回人間が UUCP を起動していたのでは効率が悪く、実用的でない。そのために UNIX には、指定した時間に指定したファイルを実行する機能がある。ここでは、その機能を使うための設定と手順について述べる。

モデムの初期設定や、電話回線の接続のための一連のシーケンスは L.sys に記述してあるので、ここでは接続のためのコマンドのみを記述すればよい。そのためのファイルを /etc/uucp/uucp.ougw として作成する。

```
rm -f /usr/spool/uucp/STST.uppersite
/usr/lib/uucp/uucico -r1 -suppersite
```

ここで最初に「STST.xxx」を rm で消去するのは、新しく uucico を起動する場合に、古い STST.xxx が残っていると新たに UUCP が起動しないためである。

次に cron の設定を行う。通常 WS の場合、cron 設定は /var/spool/cron/crontabs の下に cron ファイルを記述することで実現できる。ただし SPARCstation1 の場合には、対話的に crontab を修正するようになっている。具体例については 4.2 で述べる。また、接続する設定時間は接続先のシステム管理者と相談して、他のサイトと衝突しないよう決めなければならない。

4. mail システム

4.1 mail システムの設定

接続先が決まり、その管理者の同意がえられた段階で mail システムと news システムのプログラムをストリーマテープ等でコピーする必要がある。今後はこれを使って JUNET の設定を行う。我々が入手した時のバージョンは mail が 6.4、news が 2.11 であった。今後も頻りにバージョンアップがあると考えられるので、バージョンはよく確認し、努めて最新のものを入手する。

これから mail システムのインストール法について説明する。まず、入手したテープをハードディスク上に転送し、圧縮がかかっている場合は展開する。このとき展開するディレクトリに特別な制約はないが、これから先のインストール作業を既定値を使って簡単に進めるためには、/junet がよいと思われる。また、以後の作業はスーパーユーザとして行う。

```
# mkdir /junet/mailconf
# mv /usr/spool/uucppublic/tar_mailconf
    /junet/mailconf
# cd /junet/mailconf
# tar xf tar_mailconf
# ls
DESC/   News   doc_6.2/  src/     tool/
MST/   doc/   doc_6.3/  tar_mailconf
```

この時、展開したファイルの拡張子が「*.Z」の場合は、compress によって圧縮されたことを示しているのので、uncompress で展開しなければならない。

次に必要なファイルの作成ならびに修正を行う。まず、それぞれのサイトに合った mail システムを設定するためのファイルとして /junet/mailconf/DESC/desc.dat を作成する。この設定は mail システムにとって非常に重要であるので、慎重に設定する必要がある*。ただし、このファイルを作成するためには、それぞれのサイトの正確な構成計画と、そのためのドメイン名が必要となる。通常、コピーしたテープの中に参考となるファイルが存在するので、それも参考にするとよい。津山高専の場合は、最初もっともシンプルな形として、ドメイン名を「tsuyama-ct.ac.jp」とし、その下にホスト名「dsun01」の SPARCstation1 が 1 台だけ接続されている形で設定を行った。今後必要に応じて拡張していくかも知れないが、最初であるのでこの形を選択した。ファイルの内容は次の通りである。

```
# This is a description file for
#          '*.tsuyama-ct.ac.jp'
# Control Information
#
$ os:      SunOS4.0          # Our Operating System

$ control: !casesensitive
$ control: junet           # Junet -> Jp
$ control: !fromjunet     # We use 'From:
#          uu@xx.org.ac.jp'
#          rather than 'From:
#          uu@xx.org.junet'
$ control: genericfrom    # no host name in From:
#          field
$ control: !defsysname    # we believe $ w value
#          defined by sendmail
$ control: noyp           # We have YP system
$ control: !useyp         # we do not use YP map
#          (mail.aliases)
$ control: !severe        # forward message toward
#          #unknown domain to the master
$ version: tsuyama1.1    # Local Version
#
# name information for '*.tsuyama-ct.ac.jp'
#
name: junet               option: pseudo, omit
name: ac.jp              option: pseudo, omit
name: go.jp              option: pseudo, omit
name: co.jp              option: pseudo, omit

name: tsuyama-ct.ac.jp
option: omit
#lower_level_name:
#          tsuyama

#name: tsuyama.tsuyama-ct.ac.jp
#option: omit
lower_level_name:
          dsun01

name: *.tsuyama-ct.ac.jp
lower_level_name:
          NONE
```

* 参考として JUNET の記事である「瀬上孝司:JUNET E-mail, 236 @ mcadgw.microcad.co.jp(1990)」を利用した。

```
# route information
#
# Local Area Network
#NEIGHBOR-SITE: MAILER ZZZZZ.XXXXX.junet
dsun01:      local  dsun01.tsuyama-ct.ac.jp
              tsuyama-ct.ac.jp
              ac.jp
              go.jp
              junet

uppersite:  junet-n  pub.it.okayama-u.ac.jp
              it.okayama-u.ac.jp
              okayama-u.ac.jp
              ac.jp
              junet
```

システムを変更する必要がある場合も、このファイルの設定まで戻らなければならないので、このところはよく確認する必要がある。

次にファイル /junet/mailconf/MST/sendmail.mst の一部を修正する。このファイルは内容のごく一部を変更すればよい。変更する場所は、ファイルの 255 行目のあたりで、その場所を次のように変更する。

```
Mlocal, P=/bin/mail, F=rlsDFMmn, S=10, R=20,
              A=mail -d $ u
      ↓
Mlocal, P=/usr/mcad/lib/mail.mcad, F=rlsDFMmn,
S=10, R=20, A=mail -d $ u
```

次にファイル /usr/lib/Mail.rc の作成を行う。

```
set append dot save
set ask
set sendmail=/usr/mcad/lib/sendmail/mcad
```

さらに mail の漢字コードの変換を自動的に行うように設定する。そのためのファイルが /usr/mcad/lib/mail.mcad であり、ここでは漢字コードの変換を行うためのシェルスクリプトを作成する。我々のシステムでは、JIS コード (JUNET 標準) から EUC コード (SUN 標準) に変換を行っている。これもそれぞれのシステムによって、適当なものを選ぶ必要がある。

```
# ! /bin/sh
exec /bin/jistoeuc | /bin/mail $ *
```

そして、これを実行可能にする。

```
# chmod 755 mail.mcad
```

次に mail を送信する場合の漢字コードの変換を /usr/mcad/lib/sendmail.mcad に設定する。ネットワークに出ていくファイルの漢字は必ず JIS コードでなければならないので、そのための変換が必要となる。それをこのファイルにより設定する。

```
# ! /bin/sh
exec /bin/euctojis | /usr/lib/sendmail $ *
```

さらに、このファイルも実行可能にする。ファイルの属性を確認してみると、

```
# ls -lg
-rwxr-xr-x 1 root staff 53 Feb 9 14:39 mail.mcad*
-rwxr-xr-x 1 root staff 61 Feb 9 14:37 sendmail.mcad*
```

となるはずである。

4.2 mail のテストと実行

いま設定した環境が、正常に動作するかどうかのテストを行う。まず mail システムの環境を定義するための mailconf を作成する。

```
# cd /junet/mailconf/src
# make
# mailconf
```

これを実行すると「sendmail.cf」というファイルが自動的に作られる。すこし大きなファイルであるが、テキストファイル形式であるから中を確認してみるとよい。

つぎにこの「sendmail.cf」をテストする。これは、仮の mail 環境をつくり、アドレスの変換を実際に行うものである。

```
# /usr/lib/sendmail -d9 -C./sendmail.cf
# /usr/lib/sendmail -bt -C./sendmail.cf
```

これを実行すると、プロンプトの種類が変わり、入力要求する状態になるので、> junet-n username と入力してみる。ここで username は適当な名前でもかまわない。正確にパス名等が表示されれば成功である。また、> junet-n username@tsuyama-ct.ac.jp などとも試して見る必要がある。

アドレス変換のテストが成功すれば、次にこの「sendmail.cf」を正規の位置である /etc に移し sendmail を再起動する。この時、すでに走っている sendmail の実行を止めなければ、新しい sendmail を実行しても効力を発揮しない。

```
# cp sendmail.cf /etc
# rm /usr/lib/sendmail.cf
# ps ax | grep sendmail
 69 ?   I  0:00 /usr/lib/sendmail -bd plh
4040 p0  S  0:00  grep sendmail
:
# kill -9 69 (それぞれのプロセス ID を参照)
# /usr/lib/sendmail -bz (/etc/sendmail.cf を作成)
# /usr/lib/sendmail -bd -q30m (デーモンとして実行)
```

再起動したプロセスを確認したら、最後は定期的に電話をかけるように cron を設定する。3.5 で述べたように SPARCstation1 での cron 設定は、直接ファイルを書き換えるのではなく対話的に /var/spool/cron/crontabs を作成するようになっている。

まず、次のようなファイルを /etc/uucp/uucp.uppersite として作成する。

```
# !/bin/sh
rm -f /usr/spool/uucp/STST.uppersite
/usr/lib/uucp/uupoll uppersite
```

つぎに cron の設定を行う。この接続時間は上位サイトとの関係で決まる。

```
# crontab -e uucp
55 3,11,15,23 * * *
                /etc/uucp/uucp.uppersite
```

以上が設定が終了すれば、上位サイトに連絡し実際に mail を出して確認する。ただ、この時点では JUNET に正式登録されていないので、直接接続している上位サイトのみにしか mail を出すことが許されていないので注意する。

以上で津山高専上での mail システムの設定は終了したことになる。これ以降は JUNET からの正式承認を待って、全国のサイトと mail の交換が可能となる。

5. news システム

つぎに news システムのインストール方法^{5,6)}、並びにその運用手順を示す。mail システムが個人的な活動であるのに対し、news システムはより一般的性が高く、不特定多数の人に向けて記事が出されるから、まったく面識のない人の意見や知識にアクセスできる。これは、非常に有意義なことで、地方にいながらにして最先端の情報が入手可能となる。また「…について分からないのですが教えて下さい。」風の記事も多く、その利用価値は計り知れないものがある。ここでは、そのインストールと各ファイルの設定について説明する。

5.1 ディレクトリの作成

news システムをインストールする前に、いくつか準備をしておかなければならないことがある。まず、news を管理する者を決めてそのアカウントを「news」とし、またグループとして「news」も作成する。この場合 /etc/passwd と /etc/group に登録すればよい。通常はすでに登録されている場合が多く、/etc/passwd ファイルの中で password の欄が * で login 出来ない状態にしてあることが多い。この場合は * のままでよく、設定の必要はない。news で作業を行うときは、root から su コマンドでユーザを切り替えて行えばよい。また news システムをインストールして運用して行くために、いくつか必要なディレクトリがあるので、以下のように作成する。

ラベル	内容・ディレクトリ
LIBDIR	news システムの実行ファイルやデータファイルならびに設定ファイル等を置く場所。通常 /usr/lib/news とする。
SPOOLDIR	送られてくるニュースをためて置くディレクトリで、通常 /usr/spool/news とする。
BATCHDIR	他のホストとの記事の交換を行う時に使用するディレクトリで、通常 /usr/spool/batch とする。

ここで注意すべきことは、これらのディレクトリの所有者及びグループは news になっていなければならないことである。よく root で作業をしていると、これらの所有者が root になってしまい、運用上都合が悪くなるがあるので、十分注意する必要がある。もし所有者が root になっていると、必要な時に自動的なファイル変更ができなくなり、news システムが正常に機能しなくなる恐れがある。また、ファイルのモードも所有者とグループが読み・書き・実行が可能で、その他が読み・書きが可能の状態にしておく。

また SPOOLDIR は、news システムの運用を開始すると1日あたり1 Mbyte程度のファイルが毎日蓄積されていくので、十分容量のあるデバイスがマウントされている必要がある。

5.2 news システムのインストール

最初に mail システムと同様に、テープから必要なファイルを取り出し展開する。この時、展開するディレクトリは、先ほどの LIBDIR とは別のディレクトリを用いる。必要なファイルの転送は、この後の設定により自動的に行われる。

まず、それぞれのサイト特有の情報を含んだ専用の localize.sh を作成しなければならない。mail システムの場合の desc.dat ファイル同様に非常に重要なファイルで、設定はさらに複雑となる。設定が困難な理由は、新しい設定が正常かどうか、news システムを実際に運用するまで表面化しない場合があるからである。そのため必要により、運営後もたびたび訂正する必要がある。ここに、今回津山高専が設定した localize.sh を示す。

```
rm -f Makefile
cp Makefile.dst Makefile
chmod u+w Makefile
ed - Makefile <<'EOF'
/^UUXFLAGS/s/-r -z/-r -z -n -gd/
g/^#V7 /s///
g/^#BSD4_2 /s///
g/^#USG /d
g/^#VMS /d
g/^#BSD4_1 /d
g/^#NOTVMS/s/^#NOTVMS.*//
/^HOME=s/^/# /
/^SPOOLDIR/s;/usr/spool/news;/home1/spool/news;
/^BATCHDIR/s;/usr/spool/batch;/home1/spool/batch;
/^SPOOL_DISK/s;/usr/spool;/home1;
```

```

/~LNRNEWS/s/ln/ln -s/
/~COMMANDS/s/$/ jnews/
/~lint:/s/$/ jlint/
$a

# vnews for KANJI
#
JVOBJECTS = readnews.o rfuncs.o rfuncs2.o
            reextern.o process.o rpathinit.o \
            $(OBJECTS) jvisual.o jvirterm.o
JVSRCs = readnews.c rfuncs.c rfuncs2.c
            reextern.c process.c $(SRCS) \
            jvisual.c jvirterm.c
jlint: defs.h params.h rparams.h header.h
        $(JVSRCs) pathinit.c
        lint $(LINTFLAGS) $(JVSRCs)
        -DREAD pathinit.c

jnews: $(JVOBJECTS)
        $(CC) $(LFLAGS) $(JVOBJECTS) $(TERMLIB)
        $(LIBS) -o $@
jvisual.o: jvisual.c rparams.h defs.h params.h
            ndir.h header.h Makefile
        $(CC) $(CFLAGS) -c jvisual.c
#jvirterm.o:
# $(CC) $(CFLAGS) -DONLYSJ -c jvirterm.c
#
# content of next line must be only one period: '.'
.
w
q
EOF

rm -f defs.h
cp defs.dist defs.h
chmod u+w defs.h
ed - defs.h <<'EOF'
/ROOTID/s/10/6/
/N_UMASK/s/000/002/
/NOTIFY/s/usenet/news/
/DFTXMIT/s/-z/-z -gd/
/UXMIT/s/-c/-gd/
/INTERNET/s;/\* ;;
/MYDOMAIN/s;".*";".tsuyama-ct.ac.jp";
/GHNAME/s;/\* ;;
/DOXREFS/s;/\* ;;
/BSD4_2/s;/\* ;;
/ALWAYSALIAS/s;/\* ;;
/SENDMAIL/s;/\* ;;
/MYORG/s;".*";"/usr/lib/news/organization";
/NICENESS/s;/\* ;;
/LOCKF/s;/\* ;;
/VOID_SIGNALS/s;/\* ;;
/LINES/s/512/1024/
/SNLN/s/8/32/
$a
#define JUNET
/* enable domestic modification
 *
 * by kato@titech */

#ifdef JUNET
/* Specify KANJI code conversion filters.
 * If your Unix is not Kanji version, keep
 * undefined both macros. NTOHK : JUNET
 * standard JIS to your local code
 * HTONK : Your local code to JUNET standard JIS

```

```

* These are executed by popen(2) or system(3),
* you can also specify arguments in the
* definition. */
#define NTOHK "/usr/bin/jistoeuc"
        /* Network to Host Kanji */
#define HTONK "/usr/bin/euctojis"
        /* Host to Network Kanji */
#endif
w
q
EOF
echo "Make sure that /usr/new is in
PATH in /usr/lib/uucp/L.cmds;
read ../misc/L.cmds"

```

通常、テープの中に news システムをコピーしてきたサイトの localize.sh が残っているので、それを参考にして作成する。この時、次のことに注意する必要がある。

設定の中で compress の設定がある。これは UUCP で通信する場合に、通信費を節約する目的で compress を使用する場合に必要となる。通常、news を通信する場合、compress で圧縮をかけて転送する。これにより、最高で2倍の通信効率が得られ、それだけ電話代も安くすむ。ただし、この設定は SPARCstation1 の場合で、その他の WS ではそれぞれのシステムに合った設定が必要になる。

また、ファイルの最後のあたりにネットワークとホストの漢字コードの変換を設定する場所がある。SUN は EUC コードを使用しているが、JUNET ではネットワーク上の漢字は常に JIS コードで通信する決まりがあるために、このような変換が必要になる。

さらに、このファイルの中に「/SPOOLNEWS/s;/¥* ;」の設定をいれると、news システムを運用しても送られてきた news が /usr/spool/news/.rnews の中に貯まるだけで、いつまで経っても news がよめない状態に陥る。

次に、この localize.sh を基に Makefile を作成する。いま作成した localize.sh をシェルとして実行すると、news システムのコンパイルに必要な Makefile をシステムが自動的に作成する。この時、sh にパスが通っていないため、次のように入力する。

```
/bin/sh localize.sh
```

実行後に新しい Makefile が作成されているか確認し、ファイルの内容も一通り目を通し、間違いが無いかチェックする。また、このとき同時に「defs.h」なるヘッダファイルが作成されるので、これもチェックする。

確認が終了すれば、次の下線部分の変更を行う。

```

# define NOTIFY "usenet" /* Tell him about certain
ctl messages */
↓
# define NOTIFY "news" /* Tell him about certain
ctl messages */

```

またページャとして less があれば、more を less に変更する。ただ news を読むプログラム内には、less と同様な機能も付属している。

以上の設定が出来れば、Makefile をもとにコンパイル・リンクを行う。ただ、この操作によって作成されたファイルの所有者が root になった場合は、後でそれを news に変更する必要があるため、あらかじめ news となって作業を行うほうがよい。

```
# su news
news% make
news% make install
news% make update
```

これらの処理に少し時間を必要とする。終了したらディレクトリを /usr/lib/news に移動し、新しくファイルが生成されていることを確かめる。正しく、インストールできていれば、約 20 余りのファイルが新しく作成される。

次にこのファイルのいくつかを、それぞれのサイトに合わせて修正・作成する。なお、news システムの再構築の必要が生じた場合は、/usr/lib/news の下のファイルすべてと、/home1/spool/news のファイルを全て消去してから作業を行う。一見無駄のようであるが、実際はこの方が早く作業できる。

最初に /usr/lib/news/distribution を訂正する。このファイルはヘッダの部分に現れる Distribution: で指定する配布範囲の一覧表となる。配布範囲と名前の説明を 1 行ごとに書くことができる。次の例を参考にして設定を行う。

```
local      Local to this site
organize   Everywhere in this organization
fj         Everywhere in Japan
world      Everywhere on Usenet in the world
```

次に /usr/lib/news/mailpaths を設定する。このファイルは、メールのあて先に関する情報を記述するものであり、通常は次のように設定する。

```
backbone  %s
internet  %s
```

次に、/usr/lib/news/sys を修正する。このファイルは、自分や隣接するネットワーク、ニュース運営ホストの扱うニュースグループや、属する配布範囲、ニュースの送信手段を記述する。1 行が 1 ホストに対応し、コロンで区切られた 4 つのフィールドからなる。それぞれのフィールドは第 1 フィールドがシステム名、第 2 フィールドが news グループと配布範囲そして、第 3 と 4 フィールドは記事の送信手段を表す。第 3 フィールドの F は送信の際にファイルを compress することを表す。

```
dsun01:world,fj, comp,sci,news,rec,soc,
                talk,misc,local,to:
uppersite:world, fj,comp,sci,news,rec,soc,
                talk,misc,to.ougw:F:
```

news システムは、必要に応じて管理者に mail で通報を行うように設計されている。この設定を行うのが、ファイル /usr/lib/news/notify である。そのメールの発送をどの範囲で誰に送るのかの設定は、通常はつぎの通りでよい。

```
all news
```

最後に、ニュースのヘッダに現れる組織名として /usr/lib/news/organization を設定する。これについて、定められたフォーマットは存在しない。しかし、news の発信元が簡単に特定でき、漢字の使用を避けて次に例のようにする。

```
Tsuyama National College of Technology,
Okayama, Japan
```

以上の作業が終了すれば、再度ファイルの所有者とグループ名を調べ、すべて news であることを確認する。

5.3 ニュースのテスト

以上で基本的な news システムの設定は終了である。次に機能が完全に動作するかそのテストを行う。テストは自分のサイト（津山高専）だけで有効なローカルニュースグループを作成し、そのニュースグループにニュースをポスト（投稿）して、その後そのニュースを読むという手順で行う。

まず、ニュースグループの作成を行う。ここでは、津山高専だけのローカルなテストを行うために、「local.test」というニュースグループを作成する。

```
# su news
news % cd
news % set path=(/usr/lib/news $ path)
news % cd /usr/lib/news
news % inews -C local.test < /dev/null
/usr/lib/news/inews -n local.test.ctl -c newsgroup
local.test -d local -l +news@dsun01.tsuyama.ac.jp *
```

ここでエラーの応答があれば、その指示に従って訂正しなければならない。エラーの表示がなされない場合でも、同じディレクトリ中のファイル log とファイル erlog、さらに /usr/spool/news/control/1 のファイルの内容をチェックして、全てにエラーが出ていないことを確認する。また /usr/lib/news/active の最後の行に、local.test がうまく転送されたいことも確認する。また news 宛に、新しくニュースグループが作成されたことの報告の mail が届くので確認する。

つぎに記事のポストを行う。通常の記事の投稿と同様であるが、唯一違うのは記事の配布先を local.test にすることである。

```
% rehash
% postnews
Is this message in response to some other message?
Subject: test posting
Language [ Japanese ] ?
Kanji or Pomaji [ kanji ] ?
Keywords:
:
:
For a list of newsgroups,type?
> local.test
>
:
:
```

この後スクリーンエディットの vi が自動的に立ち上がるので、記事の内容を作成・編集してセーブ終了する。するとシステムから、記事を本当に送るかどうか確かめてくれるので、s を入力して送付する。

この記事を読む場合は、次のように入力する。これも通常の運用とまったく同様である。

```
% vnews -n local.test
```

これで正常に記事が読めれば成功である。日本語を使用した場合に、vnews では漢字が化ける可能性があるので、ニュースリーダとして jnews を使うと良い。それでも漢字が正しく表示されない場合は、送付時の漢字コードが違っている可能性があるため、その漢字コードをチェックする。

5.4 学外へのニュースの送付

これまでの作業で、ローカルなニュースグループの作成が可能となったので、次は学外とのニュースのやりとりを行う。まず、上位サイトから最新の active ファイル（存在するニュースグループの一覧表）を mail で転送してもらい、その中からどのような記事を購読するかを決める。購読する記事が決まれば、上位サイトにそのことを連絡し、記事を送付する準備を申請する。津山高専側では、送られてきた active ファイルの内容を編集しカウンタをリセットする（例 fj.junet 000000 000000 y）。この時 awk を用いて

```
% awk ' { print $1,"000000 000000", $4 } '
/tmp/active > /usr/lib/news/acitve
```

とすれば便利である。また news のバージョンにより、カウンタの桁数が違うようなので注意する。

さらに、津山高専から外へ向けて news を送付するために、sendbatch を実行する⁶⁾。これは上位サイトとの接続を行う少し前（10分程度）に、毎回実行するものなので、cron に登録することで実現する。

```
# cd /var/spool/cron/crontabs # crontab -e news
45 3,11,15,23 * * * /usr/lib/news/sendbatch -c uppersite
```

この設定は、接続の10分前にスプールにためるようにしている。正常に設定できていれば上位サイトとの news の

交換が可能となり、すべてのニュースを購読した場合1日あたり1M程度のファイルが転送されてくる。

以上で news システムの初期設定はすべて終了である。

6. あとがき

本報告では、UUCP 接続から mail,news の設定まで、JUNET の基本的な設定を詳しく述べた。JUNET への接続が決まってから、通常の運用に入るまで実際には4ヶ月程度必要とした。そのうちシステムの構築には1ヶ月半ほどかかり、かなりの手間を要した。しかし、ネットワークはその設定を行っただけでは、健全な運用を行うことはできない。通常の運用体制に移行した後も、ほとんど毎日のようにシステムをチェックし、メンテナンスを行わなければならない。そのための手順については、稿を改めて報告したい。現在では通常の運営状態に移行し、毎日1Mbyte程度の news が購読され、個人的な mail 交換も行われるようになってきている。本報告書を参考に新しい JUNET サイトが増えるとともに、今後はますます JUNET の利用者が増え、研究や教育に役立てて頂くよう心から願っている。

謝 辞

JUNET の接続では上位サイトの存在が不可欠で、快く許可していただいた岡山大学工学部情報工学科の松山隆司教授にお礼を申し上げます。また正常な運用にいたるまでには岡山大学工学部情報工学科の中田晴久先生にはたいへんお世話になり、この場をかりて感謝の意を表します。また、本校が JUNET に接続するという方針を認めていただき、援助いただいた電子計算機システム運営委員会（矢野健三委員長）のみなさまにも謝意を表します。

文 献

- 1) 岡田・河合:津山高専紀要, 28 (1990)17.
- 2) 村井:bit,19-8(1987)15.
- 3) JUNET 利用の手引き作成委員会: JUNET 利用の手引き (第1版), (1988).
- 4) 山下・城谷:SUN システム管理, アスキー, (1991).
- 5) 斉藤・山口:UNIX MAGAZINE,4-7(1989-7)74.
- 6) 斉藤・山口:UNIX MAGAZINE,4-8(1989-8)40.