実践で学ぶ

一歩進んだサーバー構築・運用術

written by 仙石 浩明

第4回 UUCPの活用

PPP（Point-to-Point Protocol）の普及に従って，UUCP接続はあまり使われなくなりました。しかし，ある程度たまったデータを短時間に一気にやりとりするUUCPでのデータ転送は，ダイヤルアップ接続に向くものです。UUCPを活用して，通信コストの削減を実現しましょう。

大企業から転職して，はや2カ月。毎週のように面接を行っては人を増やしている今日このごろです。思いがけず，かつての同僚から応募の打診がありました。同僚の中でも飛び切り優秀な人です。即座に連絡を取って会うことにしました。彼いわく大企業でのビジネスの進め方に限界を感じた，自分の技術を生かせる企業へ移りたいとのこと。これはもう雇うしかありません。

ところが2次面接で年俸の額を提示したところ，北欧の超有名メーカーからそれをかなり上回る額の提示があった*1ことを打ち明けられがくぜんとしました。相手は世界的な大企業，こっちは国内でさえあまり知られていないベンチャ，しかも年俸は向うの方が上，研究開発環境も向こうの方が良いに決まってます。これでは，彼がどちらを選択するかは一目瞭然かもしれません。

半ばあきらめつつも，小さい企業の方が提案が通りやすい*2し，もし利益が出ればダイレクトに報酬に反映される，と説得作戦を展開しました。そして，どちらを選ぶかは一生の一大事，ゆっくり考えてくださいと言って別れました。

十中八九だめだろうと思いつつ迎えた，彼が返事をくれると言った弊社の会社説明会の日，担当者の話によると彼は現れたものの何も言わずに帰ってしまったそうです。ここに至って私は完全にあきらめました。

ところがその1週間後，彼から就職を希望する旨のiモード・メールが届いたのでした。いわく，会社説明会で改めてリスクの高さを再認識した。しかし自分の技術を実用化したいのであえてそのリスクを選ぶ，と。

表1 UUCPの主なコマンド

uucp	マシン間でファイルを転送する
uux	リモート・マシン上でコマンドを実行する
cu	リモート・マシンへログインする

```
% uucp file ozenji\!/var/spool/uucppublic/ ↵
```

↵このマークで改行

図1 UUCPでのファイル転送
ファイルはキューにためられたあと，一括送信されます。

```
% rcp file ozenji:/var/spool/uucppublic/ ↵
```

図2 rcpコマンドでのファイル転送
UUCPでの転送の場合と操作方法が似ています。

ダイヤルアップ接続に向くUUCP

UUCPとは，「Unix to Unix CoPy」の略で，UNIX マシン間でデータ転送を行うためのコマンド群の総称です。主なプログラムには**表1**のものがあります。

例えば，マシンasaoからマシンozenjiのディレクトリ「/var/spool/uucppublic」へファイル「file」を送るときは，**図1**のように実行します。rcpコマンドでfileを送るとき（**図2**）とよく似ています。uucpでは，「:」の代わりに「!」が使われている点だけが異なります。「!」の前の「\」は，シェルにとって「!」が特別な意味を持っているため，それを打ち消すために必要になるものです。

*1 携帯電話の分野で日本に出し抜かれたことに危機感を抱き，日本に研究拠点を築こうとしているようです。
*2 なにせ数人の同僚さえ説得できれば，どんな研究開発でも可能です。大企業だと資金を出す（技術には詳しくない）部署を説得するのが，研究開発自体より大変だったりします。

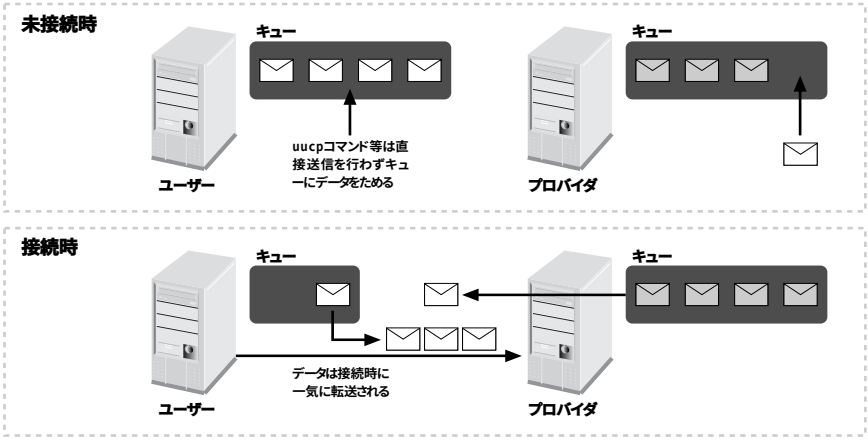

図3　UUCPの仕組み

同様に，uuxコマンドはrshコマンドによく似ています。rcpやrshが使える環境なら，UUCPは不要であると言えるでしょう。事実，インターネットの普及と共に次第にUUCPの活躍の場が減ってきました。

UUCPは非常に古いデータ転送方法です。UUCPが生まれてから実に4半世紀が経過しています*3。現在ではもうUUCPは不要と考える方も多いでしょう。しかし，そうではありません。確かにインターネットは爆発的に普及したのですが，多くのユーザーはダイヤルアップでインターネットに接続しているだけなので，まだまだUUCPが有用である場合が多いのです。

多くの人，特にWindowsユーザーの大半が，PPPでプロバイダにダイヤルアップ接続し，SMTP（Simple Mail Transfer Protocol）でメールを送信し，POP（Post Office Protocol）でメールを受信し，そしてNNTP（NetNews Transfer Protocol）でニュースを読み書きしているのではないかと思います。しかし，SMTP，POP，NNTPのいずれも常時接続を前提としたプロトコルであり，ダイヤルアップ接続で使うのは変則的と言えるでしょう。

これらのプロトコルは常時接続が前提ですから，通信時間を短くしようとする工夫は一切入っていません。それどころか，サーバーの過負荷を回避するために，集中的にデータが流れないように（つまり通信時間が長くなるように）する仕掛けすらあります。ダイヤルアップで使うのは，はなはだ不適当と言えるでしょう。日本のように電話代が高い国ではなおさらです。

一方，UUCPは元々ダイヤルアップで使うことが前提です。前述したuucpコマンドやuuxコマンドは転送すべきデータをキュー*にためるだけで，実際の転送は行いません。そして，転送に適した条件が整ったとき（例えば，データがある程度たまったとき，あるいは電話代が安い時間帯，あるいはサーバーの負荷が軽いとき）ダイヤルアップして一気に転送し，転送が終わると速やかに接続を切ります（**図3**）。

実際の転送を行うのはuucicoプログラムの役目です。uucicoは相手先ホストにダイヤルアップし，キューにたまっていたデータを一気に転送します。相手先ホストのキューに自分あてのデータがたまっている場合は，同時に受信も行います。

通信中の双方のホストの仕事は，単にファイルを転送することだけですから，多少マシンの負荷が高いときでも転送速度は落ちません。さらに，ネット・ニュースの場合は，あらかじめ複数記事をまとめて圧縮しておくので転送するデータの量が半分以下で済みます。

例えば，主に日本語が使われるニュース・グループ群として有名なfjを購読するだけなら，1日に10分程度UUCPでダイヤルアップするだけで済みます。NNTPだと少なくともこの10倍くらいの時間がかかるでしょう。サーバーの負荷が高い場合は，もっとかかるかも知れません*4。

このようにダイヤルアップ環境にとってUUCPは最適であるにもかかわらず，UUCPのユーザーはPPPのユーザーに比べると圧倒的に少数です。UUCPでは，インターネット普及の原動力となったWebを利用できないことを割り引いて考えても，少な過ぎるように思います。これは，多くの人がUUCPに敷居の高さを感じているためではないでしょうか。

UUCPの敷居が高く感じられる原因は次の2点でしょう。

（1）Windows標準でない

PPPもWindows95に採用される前は敷居が高いものでした。Windowsのようなシェアの高いOSで追加インストールせずに標準で使えることの意義は絶大で，Windows95の発売をきっかけに，PPPユーザーは一気に増えました。これに対し，UUCPはWindowsでは標準サポートされていません。

（2）UUCP対応のプロバイダが少ない

UUCPに興味があっても，接続先がなければ実験してみることさえままなりません。だからUUCPのユーザーが増えないのは当然とも言えます。ユーザー数が少なくなるに従ってUUCPが採算に乗らないサービスになり，UUCPサービスを廃止するプロバイダも出てくるといった悪循環が生じています。

逆に言うと，UUCPが何となく難しく感じられてしまうのはこの2点だけが原因で，UUCP自体は極めてシンプルな仕掛けです。そして，Linuxユーザーであれば，この2点は次のように容易に克服可能です。

(1) Linux標準である

ほとんどのディストリビューションにUUCPは標準で含まれています。このUUCPはTaylor UUCPと呼ばれるUUCPの最新バージョンで，現在の事実上の標準と言えます。Linuxだけでなく他のほとんどすべてのUNIX上で使うことができます。

(2) UUCP の接続先を探すためのメーリングリストがある

私が主催している「ご近所のUUCPサイトを探せ」メーリングリストに参加すれば，UUCP接続先を見つけることができます。このメーリングリストについて詳しくはhttp://www.gcd.org/ml/seek-uucp-sites/Welcome.ja.htmlを参照してください。

UUCPでメールを送受信する

UUCP経由でメールを送信するときは，uuxコマンドを用います。例えば，ノード名[*5]がozenjiであるマシンに，sengoku@ozenji.gcd.orgあてのメール「mail.txt」を送る場合は，**図4**のように実行します。

rshコマンドでリモートからコマンドを実行するとき(**図5**)と同様です。つまり，ozenjiで「rmail sengoku@ozenji.gcd.org」を実行し，その標準入力にメール本体(mail.txt)を与えます。もちろん，uuxとrshでは通信プロトコルが全く異なる上に，uuxはコマンド実行時に相手先マシンを呼び出すのではなく，通常は一定時間ごとに呼び出す点が異なります。しかし，コマンドの実行方法というレベルでは両者は似たようなものと考えることができます。

あて先マシンで実行するrmailの標準入力に与えるメール本体は，メール・ヘッダーを含み，先頭に「From 差出人アドレス」行が必要です。例えば**図6**のようなものです。

メールを送るたびにuuxコマンドを手で入力していては大変ですから，通常はsendmailやqmailのようなMTA(Mail Transfer Agent)にuuxを実行させます。例えばqmailの場合だと，「/var/qmail/alias/.qmail-ozenji-default」ファイルなどに，**図7**のような記述を行う[*6]ことになりますし，sendmailの場合だと，「sendmail.cf」ファイル中に，**図8**のように書きます[*7]。詳しくはMTAのマニュアルを参照してください。

```
% uux - ozenji\!rmail sengoku@ozenji.gcd.org < mail.txt ↵
```

図4　uuxコマンドでのメール送信

```
% rsh ozenji "rmail sengoku@ozenji.gcd.org" < mail.txt ↵
```

図5　rshを使ったと仮定すると図4はこうなる

```
From sengoku@gcd.org Sun May 07 04:41:57 2000
Return-Path: <sengoku@gcd.org>
Received: (qmail 8369 invoked from network); 7 May 2000 13:41:57 +0900
Received: from localhost (sengoku@127.0.0.1)
  by localhost with SMTP; 7 May 2000 13:41:57 +0900
To: sengoku@ozenji.gcd.org
Subject: test
Date: Sun, 07 May 2000 13:41:57 +0900 (JST)
From: Hiroaki Sengoku <sengoku@gcd.org>

仙石です。これは UUCP でメールを送るテストです。


#6089.                                                    仙石 浩明
http://www.gcd.org/sengoku/      Hiroaki Sengoku <sengoku@gcd.org>
```

図6　rmailに渡すメールの形式
先頭に「From 差出人アドレス」という行が必要です。

```
|preline -d /usr/bin/uux - -r -gB -z -a $SENDER ozenji!rmail "($EXT2@$HOST)"
```

図7　qmailでUUCPを利用する場合の設定
/var/qmail/alias/.qmail-ozenji-defaultなどにこのような設定を行います。

```
Muucp,  P=/usr/bin/uux, F=msDFMuP, S=30, R=30, M=100000,
        A=uux - -r -gB -z -a $f $h!rmail ($u)
```

図8　sendmailでUUCPを利用する場合の設定
/etc/sendmail.cfファイル中にこのような設定を記述します。

【キュー】　処理を待つデータをためておく場所のこと。先にキューに入ったデータから順に処理が行われる(先入れ先出し方式)。

＊3　UUCPは1976年に生まれ，翌年UNIX Version 7とともに配布されました。当時すでに，インターネットの前身であるARPANET自体は存在していました。UNIXにTCP/IPが実装され，rcpやrshコマンドが登場したのは1982年のことです。

＊4　もっとも，NNTPを使っているダイヤルアップ・ユーザーのほとんどは，fj全ニュース・グループを購読しようとは思いもしないかも知れませんね。

＊5　UUCPにおけるマシン識別子。TCP/IPにおけるhostname(ホスト名)と似たようなものです。実際，ノード名とホスト名を同一にしてもよく，例えばozenjiはノード名であると同時にホスト名(FQDNはozenji.gcd.org)でもあります。

＊6　このままだと，差出人アドレス($SENDER)に空白文字が含まれる場合，うまくいきません。uuxは空白文字を含むアドレスを正しく扱えないので，空白文字が含まれる場合は，「-a$SENDER」オプションを指定しない方がよいでしょう。

＊6，＊7　あて先アドレスを「( )」でくくるのは，アドレスに「!」が含まれている場合に対応するためです。最近は「!」形式のアドレスは，まず使われませんから「( )」は無くてもいいのかも知れません。

```
% rmail sengoku@ozenji.gcd.org < mail.txt ⏎
```

図9　UUCPでメールを受信するには
あらかじめ図のようにrmailでメールが受信できるようになっている必要があります。

```
% uux - ozenji\!rnews < news.txt ⏎
```

図10　UUCPでニュースを送信

```
#! rnews 203
Path: gcd.org!sengoku
From: Hiroaki Sengoku <sengoku@gcd.org>
Newsgroups: gcd.test
Subject: test 1
Date: 12 Apr 2000 21:40:20 +0900
Message-ID: <8d1qrk$ufv$1@asao.gcd.org>

This is the 1st test article.
#! rnews 204
Path: gcd.org!sengoku
From: Hiroaki Sengoku <sengoku@gcd.org>
Newsgroups: gcd.test
Subject: test 2
Date: 13 Apr 2000 00:13:35 +0900
Message-ID: <8d23qv$3co$1@asao.gcd.org>

This is the 2nd test article.

#! rnews 272
Path: gcd.org!sengoku
From: Hiroaki Sengoku <sengoku@gcd.org>
Newsgroups: gcd.test
Subject: test 2
Date: 13 Apr 2000 16:52:50 +0900
Message-ID: <8d3uci$1a1$1@asao.gcd.org>
References: <8d1qrk$ufv$1@asao.gcd.org> <8d23qv$3co$1@asao.gcd.org>

This is the last test article.
```

図11　ニュース送信用のバッチ・ファイルの例

```
% rnews < news.txt ⏎
```

図12　UUCPでニュースを受信するには
あらかじめ図のようにrnewsコマンドを実行して記事を受け取ることができるよう，ニュース・システムを設定しておく必要があります。

表2　Taylor UUCPの設定ファイル

設定ファイル	内容
config	メイン設定ファイル
sys	接続相手設定ファイル
port	ポート設定ファイル（発呼用）
dial	ダイアル設定ファイル（発呼用）
call	アカウント・ファイル（発呼用）
passwd	アカウント・ファイル（着呼用）

一方，UUCP経由でメールを受信する側は，送信側がuuxコマンドへの引数で指示した通り，rmailコマンドを実行するだけです。つまりmail.txtを受信したメール本体とすれば，**図9**のようにrmailコマンドを実行したとき，正しくメールがあて先に配送されるよう設定しておく必要があります。

なお，通常rmailは/usr/binに置きます。インストールされていない場合は，sendmailに含まれているrmailをインストールしてください。rmailの機能は非常に単純で，単に内部でsendmail（qmailを使っている場合はsendmailラッパー）を呼び出しているだけです。

UUCPでニュースを送受信する

UUCPでネット・ニュースの記事を送信するときも，uuxコマンドを用います。例えば，ozenjiに，記事のバッチ・ファイルnews.txtを送る場合は，**図10**のように実行します。実際には，sendbatchプログラムなどによってバッチ・ファイル生成とuuxコマンドの実行が行われます（INN等のニュース・システムの場合）。

バッチ・ファイルとは，「#! rnews 記事のバイト数」という行で区切られた複数の記事をまとめたファイルです。例えば**図11**のようなものです。これをgzipなどの圧縮プログラムで圧縮して，先頭に「#! cunbatch」という行を付けたファイルを送ることもできます。

一方，UUCP経由でネット・ニュース記事を受信する側は，送信側がuuxコマンドへの引数で指示した通り，rnewsコマンドを実行するだけです。つまりnews.txtを受信したバッチ・ファイルとすれば，**図12**のようにrnewsコマンドを実行したとき，正しく記事を受け取ることができるよう，ニュース・システムを設定しておく必要があります。

なお，通常rnewsは/usr/binに置きます。INNなどのニュース・システムをインストールすればrnewsもインストールされるはずです。

Taylor UUCPの設定例

ozenjiという名前のノード（ホスト）から，ノード名がgcdであるGCD[*8]のゲートウエイへUUCP接続する場合を例に，Taylor UUCPの設定方法を説明します。設定ファイルを**表2**に示します。設定ファイルを置くディレクトリは，コンパイル時の設定で変更可能ですが，通常は/etc/uucpか/usr/conf/uucpであることが多いようです。configファイル（メイン設定ファイル）以

外の設定ファイルは，configファイルでパスを設定可能です。また，configファイルは実行時オプションで変更することもできます。

発呼側（ozenji）の設定

UUCPで発信するマシンに必要な設定を，各設定ファイルごとに解説します。

●config

メイン設定ファイルです。自ノード名，作業用ディレクトリ，他の設定ファイルのパス名，ログ・ファイル名など，UUCPシステム全体に関係する設定を行います。ほとんどはデフォルトの設定をそのまま使えばよいので，ここでは，自分のノード名だけを設定しています（**図13**）。

●sys

接続する相手ノードごとの設定を行うファイルです（**図14**）。「system」で始まっている行から次の「system」行までが，それぞれの相手ノードごとの設定で，最初の「system」行までは，すべてのノードに共通するデフォルト設定です。

この例では，相手ノードとして「gcd」を定義しています。接続用のポート（「port」行）として「ACU*9」（後述するportファイルで定義）を用い，電話番号は「044-XXX-XXXX」（一部伏せ字）としています。

ダイヤルアップしたときのログイン方法を「chat」行で指定しています。この指定では，まず改行（\r）を送信し，「login:」プロンプトが返ってくるのを待って，アカウントID（\L）を送信します。さらに「Password:」プロンプトが返ってくるのを待って，パスワード（\P）を送信します。

ここで「\L」，「\P」で送信する実際の文字列は，それぞれ「call-login」行，「call-password」行で指定します。ですが，アカウントIDはともかくパスワードをsysファイルに直に書いてしまうと，この設定ファイルを一般ユーザーから読めない設定にしなければならないので，通常はここで示した例のように「*」を指定しておいて，アカウントIDとパスワードはcallファイル（発呼用アカウント・ファイル）に書くようにした方がよいでしょう。

callファイルは**図15**に示すように，接続する相手ノードごとに，ノード名，アカウントID，パスワード*10を空白で区切った行を列挙します（この例では接続先が1つだけなので1行だけです）。

●port

接続するためのポートを定義するファイルです（**図16**）。「port」行から次の「port」行までが，それぞれのポートごとの設定です。

ポート「ACU」はモデム型の装置で，データの入出力には「/dev/ttyS0」を使用し，転送速度は115200ビット/秒，ダイアル方法は「hayes*11」（後述するdialファイルで定義）であるという意味です。

ポート「ttyS0」の「type direct」とは，ダイアル装置が不要な直接接続という意味ですが，実際には/dev/ttyS0は前述したようにモデムに接続されているわけで，ここでは

```
nodename ozenji
```

図13　configファイルの設定例
自分のノード名以外は，ほとんどデフォルトのままで構いません。

```
call-login      *
call-password   *
time            Any
port            ACU
chat            "" \r\c ogin:-\r\c-ogin:-\r\r\c-ogin: \L word: \P
commands        rnews rmail

system          gcd
alias           gcd.org
phone           044-XXX-XXXX

system          ttyS0
port            ttyS0
```

図14　sysファイルの設定例

```
gcd     Uozenji     h=7kjiZe
```

図15　callファイルの設定例
接続する相手ノードごとに，ノード名，アカウントID，パスワードを設定します。

```
speed  115200

port   ACU
type   modem
device /dev/ttyS0
dialer hayes

port   ttyS0
type   direct
device /dev/ttyS0
```

図16　portファイルの設定例

＊8　筆者が運営する任意団体です。個人で引いたOCNエコノミーの費用の一部を賄うために，さまざまなサービスを提供しています。本連載では，このGCDでのサーバー構築，運用法を例に挙げています。

＊9　Automatic Call Unit（自動発呼装置）の略。モデム（変調器復調器）とは独立してダイアル用の装置があった（つまりモデムにはダイアルの機能がなかった）時代の名残。ここでは「モデム」と同義。

＊10　もちろん，このパスワードは説明用にランダムに生成した文字列で，実際のパスワードとは異なります。念のため。
＊11　米Hayes社が作成したモデム操作のためのATコマンド体系を指します。最近のモデム（に限らずISDNのTAやディジタル式携帯電話など）のほとんどすべてが，このATコマンド体系を採用しています。

```
chat-fail BUSY
chat-fail NO\sCARRIER

dialer hayes
chat "" AT OK ATD\T CONNECT
```

図17　dialファイルの設定例

```
U*              uucp    @       /usr/libexec/uucp/uucico -l -u @
/AutoPPP/       uucp    ppp     /usr/local/etc/paplogin
*               -       -       /bin/login @
```

図19　login.configの設定例
mgettyの設定ファイル「login.config」で，アカウントIDごとに起動するプログラムを設定することができます。

```
% cu gcd ↵
login: Uozenji ↵
Password:h=7kjiZe ↵
Shere=gcd
```

図18　発呼側の動作確認
接続後，アカウントIDとパスワードを手で入力して「Shere=相手ノード名」が出るか確認します。

```
commands        rnews rmail

system          ozenji
alias           ozenji.gcd.org
called-login    Uozenji
```

図20　着呼側のsysファイルの設定例

デバッグ等の目的でモデムを直接操作するために定義しています。

例えば，cuコマンドはsysファイルで設定した相手ノードへ接続するためのコマンドですが，

```
% cu ttyS0 ↵
```

と実行することによりモデムにつながり，ATコマンドを入力してモデムを直接操作することができます。

●dial

モデム型の装置においてダイアルする方法を定義するファイルです(**図17**)。「dialer」行から次の「dialer」行までが，それぞれのダイアル方法の設定です。

ダイアル方法「hayes」は，まず「AT」を送信し，モデムから「OK」が返ってきたら，次に「ATD」に続けて電話番号(\T)を送信する，モデムから「CONNECT」が返ってきたら成功，「BUSY」または「NO CARRIER」が返ってきたら失敗，という意味です。

発呼側(ozenji)の動作確認

以上で，発呼側のノードozenjiで必要な設定ファイルがすべてそろいました。まずcuコマンドで相手ノードgcdへ接続してみます。gcdの「login:」プロンプトが出れば成功です。続いてアカウントIDとパスワードを手で入力してみます。「Shere=相手ノード名」が出るか確認します(**図18**)。

次にuucicoを実行してみます。ログ・ファイル(通常は/var/spool/uucp/Log)で，UUCP接続が正常に行われたか確認します。

```
% uucico -s gcd ↵
```

もしエラーが生じている場合は，--debugオプションを使ってデバッグ・レベルを上げ，詳細なログを出力させるとよいでしょう。

着呼側(gcd)の設定

着呼側は，かかってくる電話を受けるだけですから，ダイアル方法などを細かく指定する必要がある発呼側に比べればTaylor UUCP自体の設定は簡単です。半面，電話を受けてからuucicoへ制御を渡すまでが少々複雑です。

電話を受けるのがUUCP接続の場合だけであれば単純ですが，UUCPだけのために電話番号を1つ割り当てるのはもったいないので，通常は複数のサービスに対応させる必要があるからです。例えばGCDのゲートウエイの場合，1つの電話番号で

・通常のlogin
・UUCP接続
・PPP接続
・FAX

を受付けています。接続のタイプによって適切なコマンドを起動するためにmgetty+sendfax[*12]を使っています。

mgetty+sendfaxに含まれるmgettyプログラムは，login.config設定ファイルで，アカウントIDごとに起動するプログラムを設定できます。GCDのゲートウエイの場合，**図19**のように設定しています。

つまり，FAXモデムからの信号をmgettyプログラムが監視していて，データ着信があるとlogin:プロンプトを出します。1文字目が「U」であるアカウントID(「U*」の行で指定)を受信すればuucicoを実行し，PPPフレーム(「/AutoPPP/」の行で指定)を受信すればPPPのPAP認証を行います。それ以外のアカウントID(「*」の行で指定)を受信した場合は，loginプログラムを実行します。FAX着信の場合は，mgettyはFAXデータを受信します。

したがって，アカウントID「Uozenji」を受信した場合は，「uucico -l -u Uozenji」が実行されます。-lオプションは，uucicoに「login:」および「Password:」プロンプトを表示させるためのものです。ただし，「-u アカウントID」オプションを付けると「login:」プロンプトをスキップします。

IDとパスワードは，passwdファイル(着呼用アカウント・ファイル)に，「Uozenji

h=7kjiZe」のように書いておきます。

configファイルとsysファイルは，発呼側とほぼ同様です。例えば，configファイルは「nodename gcd」のように自分のノード名を設定し，sysファイルは，相手ノードであるozenjiを定義します（**図20**）。ozenjiがgcdに接続するときのアカウントIDをUozenjiに限定するために，「called-login」行を指定します。

着呼側（gcd）の動作確認

適当な端末エミュレータ（cuコマンド，kermitコマンド，あるいはWindows95標準の「ハイパーターミナル」など）でgcdへ電話をかけてみます。「login:」プロンプトが返ってきたら，「Uozenji」を入力し，「Password:」プロンプトが返ってきたらパスワードを入力してみます。「Shere=gcd」が返ってくれば成功です。後は，発呼側でuucicoを実行してUUCP接続を確立するだけです。

UUCP over TCP/IP

一般に遠距離だと電話代が高くなります。UUCPだと短時間の接続で済むため，よほど大量のメールやニュースを送るのでなければ，多少遠距離でもあまり問題にはならないのですが，それでも安ければそれに越したことはありません。

インターネットはデジタル・データなら何でも送れますから，通信路の途中をインターネットで置き換えて通信料を節約しようというのが最近の流行りです。インターネット電話が有名ですし，企業間ネットワークをインターネット上に構築する，いわゆるエクストラネットも盛んです。

当然，UUCPもインターネット上で利用することができます。これを「UUCP over TCP/IP」と呼び，インターネットに接続しているホスト同士でUUCP接続します。設定方法は簡単で，発呼側は相手ノードの電話番号を指定する代りにホスト名を指定するだけです。具体的には，sysファイルの「port」行で，ACUの代わりにTCPを指定し，「address」行でTCP/IPのホスト名を指定します（**図21**）。

さらにportファイルで，ポート「TCP」の設定をします（**図22**）。

発呼側の設定はこれだけです。図18と同様に，cuコマンドを使って動作確認してください。ダイヤルアップ環境であれば，まず最寄りのプロバイダのアクセス・ポイントへPPP接続する必要があります。アカウントIDとパスワードを手で入力して，「Shere=相手ノード名」が出れば成功です。あとはuucicoを実行するだけです。電話回線だけでUUCP接続する場合に比べ，アクセス・ポイントまでの市内通話料だけで済みます。

着呼側は，TCP/IPのポート540番に接続があったらuucicoを実行するようにtcpserver＊を実行しておきます（**図23**）。

図24のようにtelnetコマンドを使って動作確認をします。

ただし，このままだとインターネット上をパスワードが平文で流れてしまいます。

UUCP接続のたびに同じパスワードが流れますから，それだけ盗み読みされる危険が高いと言えるでしょう。

拙作stone＊を使ってUUCP接続全体を暗号化してしまえば安心ですが，双方のホストにstoneをインストールする必要があり少々面倒です。この方法についてはstoneの解説（次号予定）をするとき一緒に説明するとして，今回は次善の策として，特定の

＊12　FAXを受信することができるgetty（UNIX等でユーザーからのログインを受け付けるためのプログラム。ユーザーからのアカウントIDの入力を受け付けた後，loginプログラムを実行する）です。詳しくはhttp://alpha.greenie.net/mgetty/を参照してください。
【tcpserver】　TCP/IP用ツール群「ucspi-tcp」に含まれるフィルタリング用ツール。inetdとTCP_Wrapperを合わせたような機能を持ちます。inetdよりもDoS攻撃に強く，不要な機能を排すことでTCP_Wrapperよりも安全性を向上させています。
【stone】　筆者が作成したTCP&UDPパケット・リピータ（http://www.gcd.org/sengoku/stone/Welcome.ja.html）です。任意のプロトコルをSSL（Secure Sockets Layer）で暗号化して通信させることができるほか，POPとAPOPの変換機能なども持ちます。

```
port    TCP
type    tcp
```

図22　UUCP over TCP/IPを利用する場合のportファイルの設定例

```
system          gcd
alias           gcd.org
port            TCP
address         uucp.gcd.org
```

図21　UUCP over TCP/IPを利用する場合のsysファイルの設定例

```
# tcpserver 0 uucp /usr/libexec/uucp/uucico -l ↵
```

図23　ポート540番にアクセスがあればuucicoを起動
tcpserverを使う場合の設定例です。

```
% telnet uucp.gcd.org 540 ↵
Trying 210.145.125.162...
Connected to asao.gcd.org.
Escape character is '^]'.
login: Uozenji ↵
Password:h=7kjiZe ↵
Shere=gcd
```

図24　telnetコマンドを使った動作確認
「Shere=相手ノード名」の表示が行われるかどうかを確認します。

```
#!/usr/bin/perl
$Uucpd = "/usr/libexec/uucp/uucico";
$Uuxqt = "/usr/libexec/uucp/uuxqt";
$Conf  = "/etc/uucp/config";
$host = $ENV{'TCPREMOTEHOST'};
$ip   = $ENV{'TCPREMOTEIP'};

$site = "";
if ($host =~ /\.ocn\.ne\.jp$/) {
    $site = "ocn";
} elsif ($host =~ /\.odn\.ad\.jp$/) {
    $site = "odn";
}
push(@opts,"-l","--nouuxqt","--nodetach");
if ($site) {
    push(@opts,"-I$Conf.$site");
    system $Uucpd,@opts;
    exec $Uuxqt;
}
print <<EOF;
Your machine ``$host\[$ip\]'' is not registered.
If you are the authorized user, please inform me by e-mail.
EOF
exit 1;
```

図25　UUCP接続を限定するためのPerlスクリプト
作成後は，/usr/libexec/uucp/uucpdとして保存します。

```
# tcpserver 0 uucp /usr/libexec/uucp/uucpd↵
```

図26　uucicoの代わりにuucpdを起動
接続先を限定するために，tcpserverの起動法を変更します。

```
/usr/libexec/uucp/uucico -l --nouuxqt --nodetach -I/etc/uucp/config.ocn
```

図27　指定した条件を満たしたときに実行されるコマンド
接続元ノードがプロバイダOCNにダイアルアップ接続している場合に限り，図のコマンドが実行されます。

```
nodename gcd
sysfile /etc/uucp/sys.ocn
```

図28　/etc/uucp/config.ocnの設定例

```
commands        rnews rmail

system          ozenji
called-login    Uozenji
```

図29　/etc/uucp/sys.ocnの設定例

```
% telnet uucp.gcd.org 540 ↵
Trying 210.145.125.162...
Connected to asao.gcd.org.
Escape character is '^]'.
Your machine ``asao.gcd.org[210.145.125.162]'' is not registered.
If you are the authorized user, please inform me by e-mail.
Connection closed by foreign host.
```

図30　登録していないプロバイダからのアクセスは不許可

プロバイダから，特定のノードからのUUCP接続のみを受け付ける方法を紹介します。

つまり，パスワードの盗み読みに成功したとしても，正規ユーザーになりすましてUUCP接続するには，正規ユーザーと同じプロバイダ経由でアクセスする必要があります。世界中のどこから来るか分からない，なりすまし攻撃に比べれば，特定のプロバイダに限定される分守りやすくなります。

暗号化する場合に比べるとセキュリティ的には弱いのですが，この方法だと着呼側に設定が必要なだけで，発呼側は追加設定の必要がないというメリットがあります。

まず**図25**に示すperlスクリプト「/usr/libexec/uucp/uucpd」を作ります。そして，uucicoの代わりにuucpdをtcpserverから起動します（**図26**）。

このようにすると，接続元ノードのホスト名が「*.ocn.ne.jp」である場合，すなわち接続元ノードがプロバイダOCNにダイアルアップ接続している場合に限り，**図27**のコマンドが実行されます。「-I/etc/uucp/config.ocn」はconfigファイルのパスを，/etc/uucp/config.ocnに変更するためのオプションです。

/etc/uucp/config.ocnに**図28**のような記述をしておいて，/etc/uucp/sys.ocnにはOCNからの接続を許可するノードだけを登録しておきます（**図29**）。

同様に，接続元ノードがODNを利用している場合は，/etc/uucp/config.odnが使われます。接続元ノードが利用するプロバイダごとにconfigファイルを切り替えるようuucpdに追記し，それに対応するconfig.*ファイルとsys.*ファイルを定義しておくと良いでしょう。

uucpdに登録していないプロバイダからアクセスした場合は，**図30**のように表示され接続を切られてしまいます。