12 *Using UUCP*

One of the most powerful facilities on a computer is electronic data transferÐwhether it's transferring files or sending electronic mail. On a NEXTSTEP computer, as on other UNIX computers, UUCP is one of the facilities used for electronic data transfer over serial communication lines. This chapter describes the configuration required for correct operation of UUCP on a NEXTSTEP computer.

Note: UUCP refers to the facility as a whole, and **uucp** refers to a specific program (part of the overall facility) for copying files between computers.

Planning Your UUCP Configuration

Setting up UUCP can be somewhat involved. As with most things, it's a good idea to take the time to make a plan before you begin the actual work. Here are some decisions you'll need to make.

Dial-in or Dial-out

When you use UUCP to communicate, you can choose to set up your system to initiate calls to other sites, to receive calls from other computers, or to do both. Two considerations that may influence your decision are:

·   SecurityÐAllowing dial-in access to your computer is more of a security risk than just dialing out.

·   FinanceÐSomebody has to pay for the phone call.

The choice between initiating and receiving calls doesn't affect the end result; either way, the mail is transferred, the files copied, or the commands executed.

Remote Sites

A UUCP configuration is always a cooperative undertaking. If you set up your computer to call another system, the remote site must be set up to receive your calls. If your system will receive calls from a remote site, some computer must be prepared to initiate calls. Your UUCP partner might be a remote computer within your own organization, or a computer at another organization. Either way, the administrator at the other site must set up the remote computer to communicate with yours.

Of course, you're not limited to a single partner; you might want to exchange data with dozens of other computers. In addition, transferring data to a remote computer doesn't mean you can only communicate with the computers attached to that remote network. By connecting with the remote computer, you have access to all of its UUCP partners as well. For example, a mail message transferred with UUCP might travel through several different systems before arriving at its final destination. A global network of such connections has been established.

Functions

UUCP can be used to transfer files, execute remote commands, and transfer mail. An overwhelming majority of UUCP connections are used solely to transfer mail. Although the procedures in this chapter apply to all kinds of UUCP communications, the primary focus is on mail transfer.

Calling

When a user takes some action that involves UUCP, such as sending a mail message to a remote user or queuing a command to be executed remotely, the action isn't completed until the remote connection is made. You can make a call to a remote computer by executing a UNIX command, or you can just wait until the remote computer calls you. You can also configure your system so that calls are made automatically at specific times, or whenever a mail message is ready to be delivered.

When deciding how you'll initiate remote connections, consider how many transactions you expect to make, and how long you're willing to wait before the transaction is completed. If your users will be sending hundreds of mail messages to remote users on a daily basis, you probably don't want to initiate a call every time a mail message is sent. On the other hand, you want calls made frequently enough that users won't have to wait a long time

before receiving mail.

User Accounts

When one computer makes a UUCP connection with another, it logs in using a user account. You'll need to know the account name and password on the remote computer before configuring your system for dial-out connections. If other computers will be calling you, you'll need to set up an account for each remote system. See ªConfiguring Dialing Inº later in this chapter.

Setting Up UUCP

When you set up your system to communicate with UUCP, several steps are involved. You need to set up your hardware, modify configuration files, and test connections.

Configuring the Modem

Follow these steps to configure your modem for use with UUCP:

1. Connect the modem to one of the serial ports using an appropriate cable. For more information about cabling, see Appendix B, ªCable and Modem Configuration.º

2. To set up your modem for dialing out, edit the file **/etc/remote** so that it includes an entry similar to the following:

```
a2400|Dial out on cua at 2400 baud:\
     :dv=/dev/cua:br#2400:el=^C^S^Q^U^D:ie=%$:oe=^D:
```

Replace *a2400* with the name you want to use to reference the modem when using **tip**; replace both instances of *cua* with the appropriate device file name (**cua** or **cub** for dialing out without hardware flow control, **cufa** or **cufb** for dialing out with hardware flow control); replace both instances of *2400* by the appropriate baud rate. For more information, see Chapter 7, ªAttaching Peripherals,º and the UNIX manual page for **remote**. Note that there may already be an entry in **/etc/remote** that will meet your needs.

3. Configure the modem parameters as follows:

· Set DCD (data carrier detect) when a remote carrier signal is sensed.

· Hang up when the DTR (data terminal ready) signal is dropped, and answer an incoming call only when DTR is high.

· Turn off software flow control. You can enable hardware flow control if your modem supports it and you're using a 68040-based NeXT computer. Be sure you have used the appropriate device name in **/etc/remote**.

· If your modem has a guard time associated with the escape sequence, set it to at least 3 seconds, if possible.

See Appendix B for several configuration examples.

Configuring Calling Out

To set up your system to call out, you need to modify a few UUCP configuration files. All the files are located in **/etc/uucp**, and you must be logged in as **root** in order to modify them. You might also need to change the permissions on a file before you can edit it. Be sure to change the permissions back to their original setting when you're done.

Modifying L.sys

The **L.sys** file contains information about the systems with which you will communicate. Follow these steps to set up **L.sys**:

1. Gather the information necessary to connect with the remote site:

· Host name of the remote computer
· Device name and baud rate for the modem
· Phone number for the remote computer
· Login name and password on the remote computer

2. Open **L.sys** and create an appropriate entry. The following section provides some examples.

Sample L.sys Entries

Here's an example of an entry in **L.sys**:

```
remote Any DIR 2400 cub \
    "" ATDT95557878 \
    ogin: loginname \
    ssword: password
```

Note that this entry is on separate lines solely for readability. In order to continue the entry on multiple lines, each ends with a space followed by a backslash (\).

Here's a breakdown of the various fields:

· *remote*ÐHost name of the remote computer.

· AnyÐCalls can be made at any time.

· DIRÐCall through a direct connection.

· *2400*ÐCall at 2400 baud.

· *cub*ÐCall through the device **/dev/cub**. There must be an entry in **L-devices** for any device named in an **L.sys** entry. See the next section for details.

The rest of the entry is a series of ªexpect/sendº pairs, which specify what the computer should send in response to receiving the indicated character strings.

· "" ATDT*95557878*ÐWithout waiting for anything, send the modem dialing sequence. Note that this example is for a Hayes-compatible modem; yours may use a different dialing sequence.

· ogin: *loginname*ÐWhen the string ªogin:º is received, send the user name of the account on the remote computer.

· ssword: *password*ÐWhen the characters ªssword:º are received, send the password of the remote account.

Here's an example of an entry that includes more advanced features:

```
remote Any DIR 19200 cufa \
    ABORT BUSY \
    "" AT \
    OK~3-AT-OK~3 ATx3s50=255 \
    OK~3 ATDT95552424 \
    CONNECT~80 PAUSE \
    ogin:~15-CR-ogin:~10-BREAK-ogin:~5-BREAK-ogin:~5 loginname \
    ssword:~8 password
```

The first line of this entry is similar to the first line of the previous example. The main differences are in the expect/send pairs:

- ABORT BUSYÐTells UUCP to abort the call if it receives the string BUSY at any time during the call. ABORT is a special keyword; BUSY is just the character string the modem returns when it gets a busy signal.

- "" ATÐWithout waiting for anything, send the attention sequence to the modem.

- OK~3-AT-OK~3 ATx3s50=255ÐExpect the string OK. If it's not received within 3 seconds (~3), send AT again, then wait for up to 3 seconds for OK. Once OK is received, send ATx3s50=255, which sets the modem result codes (this example assumes a Telebit T2500 configured as shown in Appendix B). A string in the form $~n$ sets the timeout to n seconds, overriding the default of 45 seconds.

- OK~3 ATDT*95552424*ÐExpect OK within 3 seconds, then dial the phone number.

- CONNECT~80 PAUSEÐWait for up to 80 seconds to receive CONNECT, then pause for 3 seconds.

- ogin:~15-CR-ogin:~10-BREAK-ogin:~5-BREAK-ogin:~5 *loginname*ÐWait for 15 seconds for ªogin:º; if not received, send a carriage return (CR), then wait 10 seconds for ªogin:º. If still not received, send a break signal (BREAK), then wait another 5 seconds; if still not received, send another break signal, then wait 5 more seconds. Once ªogin:º is received, send the user name.

- ssword:~8 *password*ÐWait up to 8 seconds for ªssword:º, then send the password for the remote account.

For more information, see the UNIX manual page for **L.sys**.

Other Configuration Files

In addition to setting up entries in **L.sys**, you also need to modify **L-devices** and possibly **L.aliases**.

1. Modify **L-devices** to contain an entry indicating the device to be used when calling out:

   ```
   DIR cua unused 19200 none
   ```

   Replace *cua* with the appropriate device name and *19200* with the correct baud rate. The device specified in the **L.sys** entry must have an entry in **L-devices**. More information can be found in the UNIX manual page for **L-devices**.

2. Set up **L.aliases** if you want to recognize the remote computer by something other than its host name. If you do create an alias, you must also modify **L.sys** to contain an entry similar to the following:

   ```
   alias Never none 19200 none
   ```

   Replace *alias* with the alias in **L.aliases** and *19200* with the correct baud rate. For more information, see the UNIX manual page for **L.aliases**.

3. Make sure the owner, group, and permissions for the configuration files are set correctly. The output from the

command **ls -lg /etc/uucp** should look like this:

```
-rw-------   1    uucp    daemon    380    Apr 17 17:06    L-devices
-rw-------   1    uucp    daemon    300    Apr 13 15:28    L-dialcodes
-rw-------   1    uucp    daemon     56    Jun 5 15:18     L.aliases
-rw-------   1    uucp    daemon    907    Jul 5  1989     L.cmds
-rw-------   1    uucp    daemon   2195    Jul 10 13:58    L.sys
-rw-------   1    uucp    daemon    414    Jul 10 13:59    USERFILE
```

Warning: A great deal of sensitive information is in the UUCP configuration files, including account passwords for remote sites. *Only* the account **uucp** should be able to read and write to these files. Setting the ownerships or permissions to anything else poses a *very* serious security risk not only for your site, but also for the remote sites.

Testing Dialing Out

Now that you've configured dialing out, it's a good idea to run a test:

1. Use the **uupoll** command to test dialing out. Enter the following command in a shell window, replacing *rhino* with the host name of a remote site that is set up to accept incoming calls from your computer:

   uupoll *rhino*

   The **uupoll** command initiates a call to the remote site, according to the information stored in **L.sys** and **L-devices**.

2. Examine the messages generated during the call by entering the following into a shell window:

   tail -f /usr/spool/uucp/LOGFILE

   Messages from **uupoll** are sent to **/usr/spool/uucp/LOGFILE**. The **tail** command with the **f** option examines the contents of the file as it is created. The output should look something like this:

   ```
   frank rhino (7/18-06:55-4088) SUCCEEDED (call to rhino)
   frank rhino (7/18-06:55-4088) OK (startup cufa 19200 baud)
   frank rhino (7/18-06:55-4088) OK (conversation complete)
   ```

   The first line shows that the remote login was successful. The second indicates successful communication with **rhino**, and the third that the call was terminated successfully. If you find any other messages, see the ªTroubleshootingº section at the end of this chapter.

3. Press Control-c to interrupt the **tail** command.

Automating Dialing Out

You can enter the **uupoll** command in a shell window whenever you want to initiate a call to a remote site. However, you can also set up your system to initiate calls automatically.

Initiating a Call when Mail Is Queued

The **sendmail** configuration file determines how to deal with mail that is sent through UUCP. As shipped, mail messages are queued until one site polls the other to check for waiting messages. If you want mail to be delivered as soon as it is queued, you need to modify the configuration files.

1. Determine which **sendmail** configuration files to modify. On a mail server, you should edit the file **sendmail.mailhost.cf**. The file **sendmail.sharedsubsidiary.cf** is used by network mail clients, and standalone computers use the file **sendmail.subsidiary.cf**. All the files are in **/etc/sendmail**.

2. Open the appropriate configuration file and find the following entry:

   ```
   Muucp, P=/usr/bin/uux, F=msDFMhuU, S=13, R=23, E=\n
   A=uux - -r $h!rmail ($u)
   ```

3. Remove **-r** from the entry so that it looks like this:

   ```
   Muucp, P=/usr/bin/uux, F=msDFMhuU, S=13, R=23, E=\n
   A=uux - $h!rmail ($u)
   ```

   Without the **-r** flag, **uux** queues the request and then immediately executes **uucico** to initiate a call.

4. Save the file.

Initiating Calls at Specific Times

Another way to automate UUCP connections is to have calls placed periodically at specific times. You can use the **cron** facility to set up such an arrangement.

· Open **/etc/crontab.local** (create it, if needed) and add an entry similar to the following, replacing *remotesite1* and *remotesite2* with the host names of the remote computers:

   ```
   3  7-20 * * 1-5  uucp  uupoll -n remotesite1
   3  7-20 * * 1-5  uucp  uupoll -n remotesite2
   5  7-20 * * 1-5  uucp  /usr/lib/uucp/uucico -r1
   ```

This entry executes **uupoll** once an hour for each remote site, then runs **uucico** separately. This scheme invokes **uucico** only once, after the last **uupoll** is run. If you prefer, you can eliminate the **-n** option to the **uupoll** entries and delete the final line executing **uucico**.

For more information, see the UNIX manual pages for **cron** and **crontab**.

Polling at Boot Time

You might want to initiate calls to the remote sites each time the computer boots. You do this by modifying one of the startup scripts.

1. Open **/etc/rc.local** and add the following lines:

```
if [ -f /etc/rc.uucp ]; then
        sh /etc/rc.uucp $1
fi
```

These lines will execute the **rc.uucp** script, which in turn executes **uucico**.

2. If you also want to force an explicit poll of a remote site, such as a central mail routing hub, insert a line similar to the following above the lines that execute **rc.uucp**:

```
/usr/bin/uupoll -n rhino
```

Configuring Dial-in

If one or more remote sites will be calling your system, you need to configure it for dial-in:

1. Create a user account for each of the remote sites with UserManager. Even though you usually shouldn't have two accounts with the same user ID, you might assign the same user ID to all of the UUCP accountsÐthe number 5 is recommended. Don't use the same user ID as the **uucp** account, which is 4. Be sure to assign a password to each of these accounts.

2. Open **/etc/uucp/USERFILE** and modify it to list the directories you want to make available to remote users. Entries look something like this:

```
Urhino,rhino /usr/spool/uucppublic
```

This entry permits the user **Urhino** (an account on the local computer) and the remote host **rhino** (the same name used in **L.sys**) to access **/usr/spool/uucppublic**. You can also include default entries to be used by any users or hosts not specifically listed. For more information, see the UNIX manual page for **USERFILE**.

Warning: To avoid a serious security risk, make sure that each user or host is only granted access to the directories necessary. Be particularly careful with directories that are accessible by any host or user. In general, no user or host should have access to the root directory from UUCP.

3. Edit **/etc/uucp/L.cmds** so that it only includes those commands that you want to be executed by remote hosts. Be sure to include **rmail** so that mail transfer will work. Be careful when you select the commands to be executed by remote users, and only include those that you're sure they'll need.

4. If you want a remote site to access your computer by something other than its host name, put the alias in **/etc/uucp/UUCPNAME**.

5. Make sure that there is an entry in **/etc/uucp/L.aliases** for every remote site. If there is a site that your computer doesn't call, but calls your system, you need a special entry for that remote site. Add an entry to **L.aliases** similar to the following, replacing *remote* with the host name of the remote computer and *19200* with the appropriate baud rate:

```
remote Never none 19200 none
```

6. Edit **/etc/ttys** so that remote users can log in through your modem. Find the appropriate line for your modem portÐ**ttyda**, **ttydb**, **ttydfa**, or **ttydfb**. Change the baud rate, if necessary; change **unknown** to **dialup**, and **off** to **on**. When you're done, the line will look something like this:

```
ttydb "/usr/etc/getty D19200" dialup on
```

Warning: Don't include the **secure** flag on lines for dial-up ports, as this allows remote **root** loginÐa serious security hole.

7. Make sure that **/etc/gettytab** contains an appropriate entry for your modem port. The existing entries are probably sufficient, but you might need an additional entry. For example, if your modem support speeds up to 19200 baud, you'll need to add an entry in the section of the file labeled ªFast dialup terminalsº similar to the following:

```
D19200:nx=D9600:tc=19200-baud:
```

Notice that the label for this entry (**D19200**) must match the argument to **getty** specified in the line for your modem port in **/etc/ttys**. The **nx** field lists the next entry to be used if a BREAK signal is received; in this example, **D9600**. For more information, see the UNIX manual page for **gettytab**.

Testing Dial-in

Once you've set up your system to accept remote logins, it's a good idea to test the configuration.

1. Have someone at the remote site set up and test their connection following the instructions in ªConfiguring Calling Outº earlier in this chapter.

2. If the remote site calls your system by using **uucico** with debugging turned on, you can find debugging output in **/usr/spool/uucp/AUDIT/**_remotesite_, where _remotesite_ is the host name of the remote computer.

Ongoing UUCP Management

Once your UUCP connections are functioning properly, you'll probably want to keep track of the activity. Several log files and an activity report are available to help you do this.

Log Files

You can examine the following log files to keep track of UUCP activity:

· **/usr/spool/uucp/LOGFILE**ÐLog of incoming and outgoing calls
· **/usr/spool/uucp/SYSLOG**ÐLog of files transferred
· **/usr/spool/mqueue/syslog**ÐLog of mail messages; maintained by **sendmail**

Each of these files is trimmed periodically to prevent them from growing indefinitely.

If you want to keep a record of remote logins, you can set up an additional log:

1. Open **/etc/syslog.conf** and add the following line:

```
auth.debug  /usr/adm/auth.log
```

Remote logins will now be recorded in **/usr/adm/auth.log**. For additional information, see the UNIX manual page for **syslogd**.

2. To prevent this log file from growing indefinitely, edit **/usr/adm/weekly**. Open the file and find the following line:

```
for i in hourly.log daily.log weekly.log monthly.log lpd-errs aculog
```

3. Add a space to the end of this line, followed by **auth.log**. Now the log file will be trimmed once a week when **cron** runs **/usr/adm/weekly**.

Activity Reports

If you want to see detailed information about all UUCP activity, you can use the shell script **/etc/uucp/uucp.day.sh**.

1. Set up a mail alias to receive the activity data in NetInfo. An appropriate name for the alias might be **uucpstats**. The value of the **members** property for the alias should be:

   ```
   | /etc/uucp/uucp.day.sh -l
   ```

   For information on setting up mail aliases, see Chapter 6, ªManaging Electronic Mail.º

2. Open **uucp.day.sh** and find the following line (it's almost halfway through the file):

   ```
   STATSPERSON=uucpstats@localhost.com
   ```

   This line defines the user that will receive mail containing the raw activity data. The value of this variable should be the name of an alias.

3. Modify the line to indicate the mail alias you created, then save the file.

4. Open **/etc/crontab.local** (or create it, if it doesn't exist), and add a line like the following:

   ```
   20 23 * * * su uucp -c /usr/lib/uucp/uucp.day.sh
   ```

   This will run the script every evening, and the activity report will be stored in **/usr/spool/uucp/STATS**.

In addition to creating and archiving activity reports, **uucp.day.sh** cleans up old log files and directories, and queues up a set of jobs to force polling of remote sites. For more information, see the comments in **uucp.day.sh**.

Troubleshooting

UUCP communication involves several pieces of equipment, many programs, and quite a few configuration files. Tracking down a problem can get fairly involved. Here are some suggestions.

Testing

You can make several tests to help isolate a problem:

- Use **uupoll**, following the instructions in ªTesting Dialing Outº earlier in this chapter.

- Use **tip** to attempt to connect with the remote site. You should be able to call the remote site and log in. For more information, see Chapter 7, ªAttaching Peripherals,º and the UNIX manual page for **tip**.

- Run **uucico** in debugging mode. The final message from **uucico** should be

  ```
  OK (conversation complete)
  ```

  For more information, see Appendix G, ªAnnotated **uucico** Debug Output,º and the UNIX manual page for **uucico**.

- Call the remote site using a phone. You should hear a tone from the remote modem. If you don't, you'll know that there's something wrong with the phone number, the phone line, or the remote modem.

Problem Solving

Here are some specific guidelines for solving problems you might encounter while testing communications:

Can't Connect to Modem

If you attempt to connect to the modem and get no response, or see a message such as ªdevice busy,º your problem might be one of the following:

- Cable configured incorrectlyÐMake sure you're using the proper cable. See Appendix B.

- Incorrect deviceÐFor dial-in use, always specify one of **ttyda**, **ttydb**, **ttydfa**, or **ttydfb** in **/etc/ttys**. For outgoing use, always use one of **cua**, **cub**, **cufa**, or **cufb** in the configuration files, including **/etc/remote**. If you're using hardware flow control, make sure you've specified the appropriate device names (the ones containing an ªfº in their name).

- Port lockedÐWhen a port is in use, a file named **LCK..**device, where device is the device name, is created in **/usr/spool/uucp/LCK**. If such a file exists and you're absolutely sure another user isn't accessing the modem, remove the lock file.

For more information, see Chapter 7.

Can't Connect to Remote Site

If you can connect to your modem but can't complete the connection to the remote modem (you don't get a ring, the remote modem doesn't answer), one of these might be the problem:

· Incorrect phone numberÐVerify that you're using the correct number for the remote site.

· Phone line problemsÐTry connecting your modem to a different phone line. If the problem persists, do the same at the remote site.

· Incorrectly configured remote modemÐMake sure the remote modem has been configured to answer.

Can't Log into Remote Site

If you can connect to the remote modem, but can't log in with **tip**, or if you see FAILED (LOGIN) from **uucico**, this might be the problem:

· User account problemÐMake sure the remote site has set up an account for you, and verify the user name and password.

Problems Once Logged In

If the login is successful, but you see garbage output from **tip**, or **uucico** reports TIMEOUT (*hostname*), one of these could be the problem:

· Modem speed mismatchÐMake sure your modem and the remote modem are communicating at compatible baud rates. Make sure the argument to **getty** in **/etc/ttys** matches an entry in **/etc/gettytab**. Make sure the **gettytab** entry cycles through appropriate baud rates. Make sure the **L.sys** entry sends break signals when an expected response isn't received.

· Parity mismatchÐMake sure that you and your remote site are using the same parity scheme. Add the appropriate keyword to the entry in **L.sys**. For details, see the UNIX manual page for **L.sys**.

Problems with uucico

There are two known bugs that will make **uucico** fail:

· Blank line in **L.sys**ÐMake sure there are no blank lines in **L.sys**.

- Missing final ªnoneºÐMake sure that any entries in **L.sy**s similar to the following include the second ªnoneº:

```
remote Never none 19200 none
```

Other Problems

Two other situations can cause problems. These can be identified from **uucico** debugging output:

- Incorrect host namesÐMake sure all host names in the configuration files are spelled correctly.

- Wrong calling timeÐMake sure you aren't attempting to call during a time that's not allowed in the **L.sys** entry.