Using A Practical Peripherals 14400FXSA Modem With A Sun™ SPARCstation™.

Syed Zaeem Hosain
Z Consulting Group
P. O. Box 610097
San Jose, CA 95161-0097
email: szh@zcon.com

1. Introduction

This document describes procedures for installing and using a Practical Peripherals 14400FXSA fax/modem on a Sun Microsystems™ SPARCstation for dial-in and dial-out access and a UUCP connection to another Sun workstation (or other UNIX™ system that uses a compatible UUCP package).

These procedures use standard operating system software and commands available on Sun workstations. There is no coverage of the fax capability of the modem, as faxing requires extra software that is not standard in SunOS™, and complete coverage of applications is beyond the scope of this document.

1.1. Operating System Version

Although the procedures are explained using a particular model—the Sun SPARCstation IPX™—they can be used with any Sun Workstation™. This document also assumes that the workstations are running SunOS 4.1.1 through 4.1.3 (also known as Solaris™ 1.x). Users running Solaris 2.0 and higher (also known as SunOS 5.x) will need to modify these instructions. The instructions for Solaris 2.x are effectively covered in the menu-driven system administration program, and will not be repeated here.

1.2. Verbatim Text

In this document, all system prompts and responses to user entries will be displayed in a regular Courier font, and all verbatim text to be entered by the user will be displayed in a bold Courier font. Since the ASCII version of this document cannot display fonts, the verbatim text will be offset from the left margin, and a system prompt “zodiac{root}> ” will also be shown. When displaying such interactions, certain text may be displayed in italic Times font and also enclosed in parentheses—these provide instructions and information that is unique to the users environment, and is not to be entered verbatim.

If the final character on any verbatim text line is the backslash character “\”, this means that the text has been continued onto the next line as it was too long to fit within the page boundaries of this document—however, it should still be entered as one long, continuous line at the system prompt.

1.3. UNIX Commands

In UNIX, if the system fails to find the command, it is likely that the directory containing the executable for the command is not in your executables search path. Check the full path (directory location) of any UNIX command by examining the on-line manual pages for further information and add the appropriate directory to your path before proceeding further.

Most UNIX commands to install and set up a Sun workstation require super-user *root* access. These are privileged operations in a multi-user, multi-tasking environment such as UNIX. Therefore, you must login as *root*, or change to the *root* account by issuing an *su*, before proceeding:

```
zodiac{user}> su
Password: (Enter the superuser password)
zodiac{root}>
```

The system prompt shown above has been modified to show that the commands are being acted upon as the *root* account—this prompt may be quite different in your environment.

2. Sun Serial Lines

2.1. Built-in Ports

There are two built-in serial ports typically available on all Sun workstations, referred to as *ttya* and *ttyb*. For most of this document, *ttya* will be used for the instructions and examples, but *ttyb* can just as easily be setup and specified.

On some models (such as the Sun SLC and Classic), the two serial ports are on the same physical connector on the back of the computer chassis, and a special adapter must be used to separate out the two serial ports—furthermore, on some of the models that require this adapter, one of the ports cannot provide full modem control signals and should not be used with a modem. On the Sun SPARCstation 1, 1+, 2, IPX, and 10, this is not an issue, as full modem control is provided on both built-in serial ports. For details regarding other SPARCstations, please refer to the system manuals or contact Sun Microsystems.

2.2. Transmission Rates

The Sun built-in serial ports can operate up to a maximum of 38,400 bps. However, the CPU interrupt rate is high for rates greater than 19,200 bps, and is usually a burden to the system. This is not a problem for most individual machine users, but many multi-user environments tend to restrict port speeds to 19,200 bps—this can be a waste of the capabilities of high-speed V.32*bis* modems like the PPI. For the rest of this document, connections using a rate of 38,400 bps will be assumed, although this can be lowered if needed.

Since the PPI modem is capable of operating its serial port DTE rate independently of the modem-to-modem DCE rate, using 38,400 bps will provide the best use of the modem data transfer speeds. However, users should note that the Sun design of the built-in serial ports does not properly support hardware handshaking for *incoming* data, and data rates above 9,600 bps may cause loss of data, if not handled by a properly configured communication program. In this situation, the serial port transmission rates may need to be limited to 9,600 bps.

On the larger Sun servers, multi-port serial cards can be installed, allowing many serial ports at high transmission rates without significant CPU overhead. These cards are available from Sun Microsystems and others. Other manufacturers provide high-performance serial port cards that can run at high transmission rates without being performance hogs. For example, the author of this document uses an Sbus I/O card from Magma Systems, where one serial port to the modem is operated at 57,600 bps, and the other serial port is connected to an IBM PC at 115,200 bps.

3. Setting Up Dial-In Access

3.1. Connecting The Modem

The modem should be connected to the Sun serial port using a standard 25 pin RS-232 modem cable. The connector on the modem end of the cable is usually a D-25 male plug that connects to a D-25 female socket on the back of the PPI modem.

On the workstation end, the cable plug will depend on the Sun model. On a Sun IPX or IPC™, each serial port socket is a mini DIN-8 connector, and an adapter cable assembly is provided by Sun to convert this to a 25 pin female socket. Thus a 25 pin male-to-male "straight-through" cable is needed.

Other vendors may provide different connectors—for example, the Magma Systems serial card provides 25 pin mail sockets, requiring a female-to-male cable. On Sun SPARCstations 1, 1++, and 2, a 25 pin female socket is provided and a 25 pin male-to-male cable can be used directly without any adapter.

The 14400FXSA modem should be connected to an operational phone line using the connector cable provided by PPI. The instructions for this connection are described in the PPI modem users manual.

3.2. Enabling RTS/CTS Handshake

Regardless of the type of serial port used, RTS/CTS hardware handshaking must be enabled to prevent data overruns, although, as mentioned earlier, the Sun built-in serial ports may still lose incoming data at high data rates. The method for establishing RTS/CTS hardware handshaking on serial ports other than the two built-in ones cannot be described in this document, as they are very specific to the vendor, but the built-in serial port *ttya* can be set up easily by entering the following:

```
zodiac{root}> eeprom ttya-ignore-cd=false
zodiac{root}> eeprom ttya-rts-dtr-off=true
```

If both built-in ports are to be used with modems, the *ttyb* port should also be modified similarly. After modifying the eeprom for the serial port values, the workstation must be re-booted for these new characteristics to be enabled:

```
zodiac{root}> fastboot
```

This will enable the modem control signals on the built-in serial port *ttya*.

3.3. Setting Modem Characteristics

Once the modem has been connected and the serial port handshaking defined, the modem characteristics should be set for proper operation. For the PPI 14400FXSA modem, the basic set up for use on IBM™ PC compatible computers will work well:

```
zodiac{root}> (stty 38400 ; echo "AT&F&C1&D2S0=1&W") > /dev/ttya
```

This enables the PPI modem factory defaults (selected by "&F"), RTS/CTS flow control ("&C1&D2"), auto-answer on one ring for in-coming calls ("S0=1"), and stores the selected configuration into the modem non-volatile memory ("&W"). It also sets the serial port DTE rate to 38,400 bps—independent of the modem DCE rate—since the PPI 14400FXSA sets its serial port rate upon receiving an "AT" command.

3.4. Preparing For A *getty* Process

After the modem has been installed, a *getty* process must be started to monitor the serial line for in-coming calls—*getty* waits for a connection and starts a login session with the operating system. To specify the information for the *getty* process, edit the terminal control file */etc/ttytab*, using *vi*, or your favorite editor:

```
zodiac{root}> vi /etc/ttytab
```

This file contains definitions for each port—one per line. The line for the serial port device */dev/ttya* should be changed from:

```
ttya  "/usr/etc/getty std.9600"    unknown   off local secure
```

to read as follows:

```
ttya  "/usr/etc/getty std.38400"   vt100     on remote secure
```

where the changed information is shown in bold. The keyword *secure* allows people to log in as the superuser *root* on that line. If this is not desirable, e.g. when the line is used for general access by the outside world (common on a modem connected to the phone system), leave out the keyword *secure*:

```
ttya  "/usr/etc/getty std.38400"   vt100     on remote
```

The terminal type may also be changed as needed—this above example shows the selection of a vt100 as the default terminal characteristics to apply to that serial line when anyone logs in. Other terminal types can be selected as desired—the */etc/termcap* file shows the available types of terminals that may be used:

```
zodiac{root}> more /etc/termcap
```

3.5. Changing Port Characteristics

The terminal process started will set 7 data bits, 1 stop bit, and EVEN parity for the incoming connection. This tends to be confusing for people using standard dumb terminals or PC terminal emulation programs. Therefore, the characteristics of the line may be changed to 8 data bits, 1 stop bit, and NO parity, by editing the */etc/gettytab* terminal configuration database:

```
zodiac{root}> vi /etc/gettytab
```

and changing the line associated with the selected transmission rate (in our example, 38,400 bps) from:

```
h|std.38400|38400-baud:\
          :sp#38400:
```

to read as follows:

```
h|std.38400|38400-baud:\
         :sp#38400:p8:
```

where the addition is shown in bold text. This enables 8 bit processing on all 38,400 bps lines for *getty*.

Users should note the characters “std.38400” is the same as specified in */etc/ttytab* for enabling the serial port—this is not a coincidence! In particular, the in-coming and out-going transmission rates will both be set to 38,400 bps. This is convenient, as it avoids the problems of detecting the transmission rate used for communication (autobaud), and selects an optimum value for all operations. Note that if the modem is used

for some other purpose, it is important to ensure that its DTE rate is reset for 38,400 bps, else in-coming calls will fail—the modem will attempt to login at a DTE transmission rate to which *getty* will not respond.

3.6. Restart The *getty* Process

After the terminal configuration file has been modified, the *getty* process should be re-started using kill, and proper *getty* operation should be verified. The following shows the specific response from my computer—it will be different for other systems:

```
zodiac{root}> kill -1 1
zodiac{root}> ps axww | grep getty
root    181 p1 S    0:00 grep getty
root   5385 m0 IW   0:00 - std.38400 ttya (getty)
```

The key item to be noted is the last line above, which shows a *getty* process monitoring serial port *ttya*. That is it! It is now possible to dial and log into the system using the PPI 14400FXSA modem!

4. Setting Up Dial-Out Access

4.1. Communications Software

It should be noted that the two dial-out communication programs provided in SunOS by Sun Microsystems, *tip* and *cu*, are fairly limited in their capability, and users should obtain other programs to get better performance. This is beyond the scope of this document—users are encouraged to obtain *kermit* from Usenet archives or via anonymous ftp from Columbia University (it may also be available on the UNIXFORUM forum on Compuserve), or some other program—*pcomm* has been widely recommended as a reasonable choice, and is available at various Internet ftp sites.

In this section of the document, the procedure for setting up *tip* will be described. The procedures for setting up *cu* are similar to those for UUCP, and will also be described.

4.2. Dial-Out Devices

In a UNIX environment, where dial-in and dial-out using the same modem and serial port is a fairly common occurrence, a mechanism must be provided to prevent the serial line from being accessed for both purposes simultaneously. Furthermore, the *getty* process that watches for in-coming calls must be able to distinguish between these calls and data being received by an out-going communication session. Basically, *getty* must be disabled automatically during a dial-out session.

Various flavors of UNIX attempt to deal with this problem in different ways. In SunOS, this is accomplished by using different logical "devices" for the same physical serial port. Thus */dev/ttya* is a serial in-coming logical device and the corresponding out-going device is usually named */dev/cua0*. For the in-coming device */dev/ttyb*, the out-going device is usually named */dev/cua1*. These devices must be "created", as they are usually not installed as part of the SunOS installation process.

First determine some information about the existing serial devices:

```
zodiac{root}> cd /dev
zodiac{root}> ls -l ttya ttyb
crw-rw-rw-   1  root   12,   0 Dec 11 00:13 ttya
crw-rw-rw-   1  root   12,   1 Dec 11 00:13 ttyb
```

The information returned above should be checked for two important numbers: the 12 and the 0 (or 1) following "root" (these are the typical numbers for a SunOS 4.1.3 system, but may be different in your environment). These are the major (12) and minor (0 or 1) indexes into the UNIX kernel device tables, and must be used to build the devices for the out-going serial ports. In SunOS, the corresponding out-going devices to these in-coming lines must have the same major number, but the minor number is calculated by adding 128 to the *ttya* or *ttyb* minor numbers.

Thus, the next step is to check to see if the */dev/cua0* and */dev/cua1* devices already exist and have the correct major and minor numbers. If they do exist, and the device numbers are correct, then you can skip to the next section, otherwise these devices need to be removed (if the device numbers are wrong), and re-created. The following shows normal correct entries (based on the *ttya* and *ttyb* numbers of 12 and 0 or 1).

```
zodiac{root}> ls -l cua*
crw-rw-rw-   1  root   12,   128 Dec 11 00:13 cua0
crw-rw-rw-   1  root   12,   129 Dec 11 00:13 cua1
```

If the device numbers are not correct, the current entries should be removed before other steps are taken:

```
zodiac{root}> rm cua0 cua1
```

Then, to create (or re-create) the out-going devices for the standard Sun built-in serial ports, enter:

```
zodiac{root}> mknod cua0 c 12 128
zodiac{root}> mknod cua1 c 12 129
```

The letter *c* in the above mknod commands is used to specify character-mode devices (such as serial ports) as opposed to block devices (such as disk drives) and must be explicitly entered as shown. Both out-going devices may be built as shown, even if you are not using both ports for dial-out, as the existence of an unused device in the */dev* directory is not a problem. Entering:

```
zodiac{root}> ls -l cua*
crw-rw-rw-   1  root   12, 128 Feb 28 12:13 cua0
crw-rw-rw-   1  root   12, 129 Feb 28 12:13 cua1
```

will display the new devices that have been created. If the file privileges are not as shown above, they should be modified once:

```
zodiac{root}> chmod 555 cua0 cua1
```

After this single modification is performed once, they will be automatically handled by the *cu* and *uucico* (main routine for UUCP operation) commands during normal operation.

4.3. The Remote Host Description File

After the dial-out devices have been created, the remote host description file */etc/remote* must be edited for use with *tip,* using *vi,* or another editor:

```
zodiac{root}> vi /etc/remote
```

Copy the two-line entry beginning with the characters "UNIX-1200" into another location using "cut-and-paste" editing procedures, rather than manually re-typing the characters, since these lines contain some special control characters that are difficult to simply insert by typing them. After copying the two lines, various fields in the new lines should be replaced for proper communications with the modem. The characters "UNIX-1200" should be replaced with the characters "tip38400"; the transmission rate

information following the characters “br#” should be replaced with the new serial port rate, and the modem type should be changed from “ventel” to “hayes”.

In other words, the *newly copied lines* should be changed from:

```
UNIX-1200:\
      el=^D^U^C^S^Q^O@:du:at=ventel:ie=#$%:oe=^D:br#1200:tc=dialers
```

to read as follows:

```
tip38400:\
      el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#38400:tc=dialers
```

where the necessary changes are shown in bold. The modem is now configured for dialing out using *tip*.

4.4. Dialing Out Using *tip*

To use the modem for dialing out, the following will establish a connection to a particular phone number:

```
zodiac{root}> tip -38400 555-9999
dialing...connected
...(May have to hit return a few times for the remote system to respond)
...(Much display information not shown)
~.    (the tilde-dot exits from tip)
Closed connection
```

In this example, the *tip* program has been asked to utilize a serial port configured for 38,400 bps and dial the (imaginary) number 555-9999. Selecting other transmission rates will not function properly, as the remote host configuration file was not set for use with other baud rates. It is, of course, possible to have done so, but this would not provide optimal use of the modem DTE rate-independence capabilities.

For more information, users should refer to the on-line manual page for *tip*.

5. Setting Up A UUCP Connection

Using the PPI modem for a UUCP connection is a relatively easy process. It is important to remember that UUCP is a co-operative process between two workstations, and both systems must be prepared correctly to handle the communications. The specific UUCP communication protocols (how UUCP works) are beyond the scope of this document—interested users are referred to *“Sun System & Network Manager's Guide”* delivered by Sun with each system, and the excellent *“Managing UUCP And Usenet”* by Tim O'Reilly and Grace Todino.

However, specifics regarding the use of the PPI modem for UUCP will be described here—in particular, only those changes that are specific to the local system, as the remote UUCP capabilities will be established by the System Administrator for that remote system and remote modem combination.

5.1. Selecting A UUCP Device

The device used for the connection is specified in the file */etc/uucp/Devices* in the SunOS UUCP package, which is based on HoneyDanBer UUCP, and is also known as Basic Networking Utilities or BNU. Note that this filename is specific to SunOS 4.x and higher (including Solaris 2.x)—older versions of the OS used a different UUCP package, with different file names, and will not be discussed here.

Therefore, edit the file:

```
zodiac{root}> vi /etc/uucp/Devices
```

and add the following two lines at the end of the file:

```
ACU cua0 – 2400 ppi2400
ACU cua0 – 38400 ppi
```

These two lines are intended to provide information to the UUCP system (and *cu*) on what device is to be used for outgoing communications for the requested transmission rate. It is possible to use multiple modems, with different transmission rates, on different serial ports and specify these accordingly.

It should be noted that a special entry is being created for 2400 bps operation. The characteristics for this transmission rate will be set later to allow connection to old 2400 bps modems that do not perform any data-compression or error-correction—i.e., modems that do not provide V.42 error correction or V.42*bis* data compression.

When calling in to such modems, feature negotiation must be turned off on the PPI modems (just like all other similar high-performance V.32 and V.32*bis* modems) to allow the connection to proceed normally. If it is anticipated that no such old non-error correcting 2400 bps modems will ever be called, the two lines shown above can be replaced by the single line:

```
ACU cua0 – Any ppi
```

The final words on each line, *ppi* and *ppi2400*, correspond to the entries to be accessed in the Dialers file (discussed shortly). Note that many texts on UUCP will recommend using a pre-defined entry *hayes* for this purpose. This is usually correct and will work for the PPI 14400FXSA, but the pre-defined entry does not cleanly reset the modem to the default stored characteristics. If the modem is used for any other purposes, and its characteristics are changed, the pre-defined *hayes* entry may fail to work till the modem is reset. For this reason, a new entry *ppi* is shown and recommended in this document, and correctly set up for more consistent UUCP connections.

It is important that the ownership of the */etc/uucp/Devices* file be set to the *uucp* account, and file privileges be set accordingly. Therefore enter the following to ensure this:

```
zodiac{root}> chown uucp.uucp /etc/uucp/Devices
zodiac{root}> chmod 444 /etc/uucp/Devices
```

5.2. Setting The Dialers

The next step is to set the dialer characteristics available to the system. This is an index into */etc/uucp/Dialers* that specifies the characteristics and login sequence to use when the UUCP system is connecting to another site. The file */etc/uucp/Dialers* should be edited:

```
zodiac{root}> vi /etc/uucp/Dialers
```

and the following lines should be inserted at the end of the file:

```
ppi =,–, "" \dA\pTZ\r\c OK\r ATS2=255\r\c OK\r \EATDT\T\r\c CONNECT
ppi2400 =,–, "" \dA\pTZ\r\c OK\r ATS2=255&Q6S37=6\r\c OK\r EATDT\T\r\c CONNECT
```

As mentioned earlier, the *ppi2400* entry is designed for connecting to old 2400 bps modems that do not perform error-correction or data-compression and do not negotiate these features during initial dial-in. This

second entry for *ppi2400* may be deleted from the above file, if these old modems are not going to be used at any of the remote sites that are being dialed.

Once again, it is important that the *uucp* account own this file and that it have the correct access privileges. Therefore, enter the following to ensure this:

```
zodiac{root}> chown uucp.uucp /etc/uucp/Dialers
zodiac{root}> chmod 444 /etc/uucp/Dialers
```

5.3. Setting The Remote System Information

The next step is to establish the information pertaining to the remote system, such as the login name and password to use, the telephone number to dial, the login chat sequence to use for establishing a connection. This is unique for each system—thus this document will only describe a typical Sun workstation setup.

For the purposes of this discussion, assume a remote machine name of *sunset*, whose system administrator has defined a UUCP login account of *uzodiac*, with a password *uzodpass*. This remote machine is accessed via the telephone number *123-4567*. Thus, edit the local file */etc/uucp/Systems*:

```
zodiac{root}> vi /etc/uucp/Systems
```

and enter the following line at the end of the file:

```
sunset Any ACU 38400 1234567 "" \r in:--in: uzodiac ssword: uzodpass
```

This reads as follows: connection to the machine *sunset* is to be established by using Any ACU device that can be run at 38,400 bps, by dialing the number *1234567*, expect nothing as an initial response and send a carriage-return, wait for the letters *in:* (typically the last characters for a *login:* prompt), enter the characters *uzodiac*, wait for the characters *ssword* (typically the last characters for a *Password:* prompt), and enter the characters *uzodpass*. To enter numbers that contain secondary dial-tones, use an equal sign after the initial dial-tone access prefix. For example, the number may be entered as *9=1234567*. This would dial 9, then wait for a secondary dial-tone for an outside line, before dialing the rest of the number.

This is generally sufficient to establish the connection. On the remote machine, the system administrator will have set up the *uzodiac* account to automatically run the UUCP communication program *uucico*.

Once again, the file ownership must be set to the *uucp* account. Enter the following to establish this:

```
zodiac{root}> chown uucp.uucp /etc/uucp/Systems
```

The Systems file also contains sensitive information, such as account names and passwords, and should be protected from general read access by other users. Therefore, change the file protection as follows:

```
zodiac{root}> chmod 400 /etc/uucp/Systems
```

5.4. Setting UUCP Permissions

When using UUCP, it is common to limit access to commands and directories. Since UUCP is very often used for only receiving electronic mail and Usenet News feeds, such connections can be cleanly restricted. If file copying is to be allowed, typically between trusted machines, some restrictions need to be lifted (up to a point). The file */etc/uucp/Permissions* is used to establish these restrictions.

There are two type of entries in this file:

- LOGNAME entries allow control of specific permissions used for uucp login id's when remote systems call your machine.

- MACHINE entries allow control of specific permissions used for uucp login id's when your machine calls remote systems.

For the purposes of the discussion in this section, we will use the same hypothetical remote machine *sunset* mentioned earlier, and various levels of permissions will be described, although a complete discussion of all the possible options is beyond the scope of this document.

Thus, */etc/uucp/Permissions* should be edited:

```
zodiac{root}> vi /etc/uucp/Permissions
```

and entries should be made for each system that is dialed by the UUCP software.

A typical example of the entry that may exist in this file for a highly trusted system is thus (note the use of the back-slash character to denote a continued line that is actually a single line in the file):

```
MACHINE=sunset MYNAME=zodiac COMMANDS=ALL REQUEST=yes SENDFILES=yes \
READ=/usr/spool/uucppublic WRITE=/usr/spool/uucppublic
```

The corresponding entry in the same file on the remote system *sunset* may look like this:

```
LOGNAME=uzodiac MYNAME=sunset MACHINE=zodiac COMMANDS=ALL \
REQUEST=yes SENDFILES=yes READ=/usr/spool/uucppublic \
WRITE=/usr/spool/uucppublic
```

In the above examples, the COMMANDS=ALL is a very “open” and unprotected entry, that will allow quite a lot of access—the remote machine can execute commands locally! This may be a security issue, and should not be done except with extremely well-trusted machines.

An example of a machine that is only connected for the sole purpose of sending electronic-mail may look like this:

```
MACHINE=sunset MYNAME=zodiac COMMANDS=rmail
```

Here, the commands have been restricted to *rmail*, which only allows electronic mail to be sent. In a typical environment that accesses the Usenet News feeds, additional access may be required:

```
MACHINE=sunset MYNAME=zodiac REQUEST=yes SENDFILES=yes \
COMMANDS=rmail:rnews WRITE=/usr/spool/news:/usr/spool/uucppublic
```

Clearly, the correct combination of entries in the */etc/uucp/Permissions* is fairly large, and very dependent on the specific requirements. A detailed description of all possible options is beyond the scope of this document—interested users are, once again, referred to the excellent text *“Managing UUCP And Usenet”*.

Once again, the file ownership and privileges should be properly set:

```
zodiac{root}> chown uucp.uucp /etc/uucp/Permissions
zodiac{root}> chmod 444 /etc/uucp/Permissions
```

5.5. Setting The UUCP Polling

After establishing the device and system characteristics, a polling sequence needs to be setup for every remote machine. This simply means that the system will allow remote machines to be connected at various times during the day to allow the UUCP requests that have been queued up, to actually perform their functions.

To achieve this, edit the file */etc/uucp/Poll*:

```
zodiac{root}> vi /etc/uucp/Poll
```

and enter one line for each remote system. In the following example, we are still working with the sample remote machine named *sunset*, so add the line:

```
sunset 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

Here, we are simply requiring that the machine *sunset* will be setup for polling every hour. The UUCP system may still not actually call till the appropriate processes have been started with an entry in the cron tables (as shown in the next section), but the polling sequence will be established.

As shown many times already, set file ownership and privileges as needed:

```
zodiac{root}> chown uucp.uucp /etc/uucp/Poll
zodiac{root}> chmod 444 /etc/uucp/Poll
```

5.6. Starting cron Processes For UUCP

The UUCP system needs to be executed periodically to perform the tasks that have been queued, as well as to log into the remote system to retrieve work and files, and finally, to perform normal cleanup tasks. For this purpose, Sun provides standard scripts that need to be executed on a schedule that is meaningful for the environment. The *cron* process is used to regularly execute these scripts.

When, and how often to execute the scripts, is a matter of choice for the user. Typically, some balance between the required accesses, the cost of the telephone call (whether it is local or long-distance), the time of day, the amount of data transferred, whether a Usenet News feed is accessed, etc., is needed to allow efficient UUCP usage.

Sun provides a basic crontab control file for use with UUCP. This file is located in */usr/lib/uucp* and is named *uudemon.crontab*. The following are the lines in this example file (for SunOS 4.1.3):

```
zodiac{root}> cat /usr/lib/uucp/uudemon.crontab
#
# ident  "@(#)uudemon.crontab 1.1 88/05/16"
#
48 8,12,16 * * * /etc/uucp/uudemon.admin
45 23 * * * /etc/uucp/uudemon.cleanup
0 * * * * /etc/uucp/uudemon.poll
11,41 * * * * /etc/uucp/uudemon.hour
```

For most users, this will suffice, but the entries in this file can be tweaked as necessary. This file should be installed as the cron control file for the uucp account. For this purpose, enter:

```
zodiac{root}> cd /usr/lib/uucp
zodiac{root}> crontab -e uucp
```

This will start a *vi* edit session of the cron control file for the *uucp* account, and you can then include the file *uudemon.crontab* using the ":r uudemon.crontab" command during the edit. When you exit the editor, this file will be installed as the cron file for the *uucp* account.

The following shows how to check the file for a given user, and also shows the entry for *uucp* on my Sun IPX system:

```
zodiac{root}> crontab -l uucp
#
# ident  "@(#)uudemon.crontab 1.1 88/05/16"
#
48 8,12,16 * * * /etc/uucp/uudemon.admin
45 23 * * * /etc/uucp/uudemon.cleanup
0,30 * * * * /etc/uucp/uudemon.poll
11,41 0-7 * * * /etc/uucp/uudemon.hour
21 8-23 * * * /etc/uucp/uudemon.hour
```

This is similar to the Sun distributed file, except that the */etc/uucp/uudemon.hour* script is only called once per hour during the hours from 8am to 11pm, and the polling is performed every 30 minutes. The remote machines are called at least once an hour as they are local calls. Users may wish to limit how often a remote machine is called, particularly if they wish to use long-distance UUCP calls at night! For more details on the format of the crontab file, please refer to the on-line manual page for the file format:

```
zodiac{root}> man 5 crontab
```

And this it! The UUCP system is now operational.

5.7. Testing The UUCP System

Sun has provided a test program to allow the UUCP configurations to be properly tested for a remote site. To perform this test, enter:

```
zodiac{root}> cd /usr/lib/uucp
zodiac{root}> Uutry sunset
```

The Uutry program will perform a normal UUCP connection operation with the remote system *sunset*. If everything proceeds smoothly, the program will end without errors. However, if errors are encountered, these will need to be debugged carefully. This debugging is beyond the scope of this document—interested users are referred to "Managing UUCP And Usenet".

5.8. Dialing Out Using *cu*

Once the UUCP system is operational, the *cu* program can be used to dial remote systems whose dialing information is available in */etc/uucp/Systems*. For this purpose, the following command may be used:

```
zodiac{root}> cu sunset
(modem dials out to the remote machine)
sunset login: (enter username)
password: (enter password)
Welcome to SunOS 4.1.3 (SUNSET)
sunset{user}>
...(login session details not shown)
sunset{user}> logout
~.
zodiac{root}>
```

The *cu* program can also be used with a specific line and phone number, even if the entry for the system is not included in the */etc/uucp/Systems* file. For more information, please refer to the man page for *cu*.

6. References

6.1. *"Managing UUCP And Usenet"*, by Tim O'Reilly and Grace Todino, published by O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol, CA 95472.

6.2. *"Sun System & Network Manager's Guide"*, by Sun Microsystems, Inc.

6.3. *"SPARCstation IPX Installation Guide"*, by Sun Microsystems, Inc.

6.4. *"Practical Modem 14400FXSA Operating Manual"*, by Practical Peripherals, Inc.

7. Trademarks

Sun Microsystems, *Sun Workstation*, *Sun*, *Sun-3*, *Sun-4*, *IPX*, *IPC*, *NFS*, *SunOS* and *Solaris* are registered and unregistered trademarks of Sun Microsystems, Inc.

UNIX and *OPEN LOOK* are registered trademarks of UNIX Systems Laboratories, Inc. and/or Novell Corporation.

IBM PC is a registered trademark of International Business Machines Corporation.

SPARC is a registered trademark of SPARC International, Inc. Products bearing the SPARC trademark are based on an architecture developed by Sun Microsystems, Inc. *SPARCstation* is a trademark of SPARC International, Inc., licensed exclusively to Sun Microsystems, Inc.

Compuserve is a trademark of Compuserve, Inc.