A
Northern Illinois University
Academic Computing Services
Workshop

UNIX Basics for Superusers

Michael G. Prais
Swen Parson 120
753-1057

Communications

UNIX users can communicate with the system
through a terminal or another system;
they can communicate what is done on the system to a printer;
or they can communicate with one another using electronic mail.
With the exception of electronic mail between users on the same system,
UNIX communications requires
either *circuit-switched* (serial, RS232C, or RS422)
or *packet-switched* (Ethernet or IEEE 802.3) *network connections*.
The connections through these *ports* require physical wiring
as well as software device drivers.

Sun has provided quite a bit of information about these devices
in the chapter on Adding Hardware to Your System
in the Sun System Administration Procedures Manual.
Configuration information is found in chapters 4 (devices)
and 5 (file formats) of the SunOS Reference Manual.

Electronic Mail

Messages can be exchanged with users on the same system or
with users on connected systems using the *mail* or *mailtool* commands.
Messages from these commands are placed in /var/spool/mqueue
and then routed by the /usr/lib/sendmail daemon.

Sendmail can send mail to an internet address directly
where sendmail on the remote system accepts the mail.
It can also send mail to a uunet address using uux on the local system
and /bin/rmail as an interface to sendmail on the remote system.
The */usr/ucb/mail* program also distributes mail
received by sendmail on the local system
by placing mail in a /var/spool/mail/*username* file.

ls -l /var/spool/mail	Lists the mailfiles for users.

The permissions for these directories must be 700
for the user to access mail.

more /var/spool/mail/*username*	Lists the mailfile for *username*.

The /usr/etc/in.comsat daemon announces the existence of mail
to each user.

Sendmail can also append mail to a file when given an absolute pathname,
or send mail to a process when given a pipe.

Sendmail is configured from /etc/sendmail.cf.

more /etc/sendmail.cf	Displays the routing instructions given to sendmail.

DM*domainname*	Identifies outgoing domainname.
CM*domainname*	Identifies incoming domainname.
DD*domainname*	Identifies outgoing subnet domainname.
CD*domainname*	Identifies incoming subnet domainname.
DUedu	Identifies the education subuniverse.
CV*hostnames*	Identifies local uucp connections.
DMether	Identifies ethernet connection to relayhost.
DMuucp	Identifies uucp connection to relayhost.
DR*hostname*	Identifies outgoing relayhost name.
CR*hostname*	Identifies incoming name for relayhost.
OPPostmaster	Identifies recipient of undeliverable mail.

The system that receives and distributes mail is a *mailhost*.
This system can be the local system or a remote system,
but must be defined in /etc/hosts.

```
more /etc/hosts
```

Displays the known hosts on the network.

Systems that collect mail and redistribute mail are called *relayhosts*
to other systems.
Subsidiary mail systems receive and maintain mail files
for users serviced by the *mailhost*.
The local mailhost must have /usr/lib/sendmail.main.cf as sendmail.cf,
while a local system with a remote mailhost must have
/usr/lib/sendmail.subsidary.cf as sendmail.cf.

```
comm -12 /usr/lib/sendmail.main.cf /etc/sendmail.cf
```

Displays common lines in files.

```
#cp /usr/lib/sendmail.main.cf /etc/sendmail.cf
```

Installs sendmail.cf for a local mailhost.

```
comm -12 /usr/lib/sendmail.subsidary.cf /etc/sendmail.cf
```

Displays common lines in files.

```
#cp /usr/lib/sendmail.subsidary.cf /etc/sendmail.cf
```

Installs sendmail.cf for a remote mailhost.

The default is to install sendmail.subsidary.cf.

```
#/usr/lib/sendmail -v < /dev/null ip_address
```

Tests sendmail connections
with local and remote systems.

The *mail* command uses the /etc/passwd file
to recognize local mail recipients.
Sendmail identifies mail recipients
from information in /etc/aliases.pag and /etc/aliases.dir.
These file are created from the file /etc/aliases.

```
more /etc/aliases              Lists the mail aliases for the local system.
```

A representation for /etc/ aliases is displayed below.

```
            #comment
            mailname: address, ...
```

Aliases can be absolute pathnames or users on other systems.
The /etc/aliases file can be used to redirect mail to root or other users
to the primary user of a workstation.

```
      Postmaster: root
      root: username
```

```
#newaliases                    Recreates the system /etc/aliases files.
```

The *newaliases* command is a link to *sendmail*.
Sendmail will forward mail for a user when it finds a .forward file
containing an address in the user's home directory.

When a user invokes *mail*,
the program processes the system-wide settings in /usr/lib/Mail.rc
and then the user's settings in .mailrc.
The file /usr/lib/Mailrc provides a prototype for .mailrc.

```
more /usr/lib/Mailrc           Lists system-wide mail settings.
```

The runtime configuration often sets up mail to collect copies of
outgoing mail in the .record file.

```
ls -l .record                  Displays the ownership and permissions
                               for the .record file.
```

It is a security risk to allow the .record file to be readable or writeable by the group or outsiders.
Mail sent to the .record file or any file can be read by the *mail* command one message at a time.

mail -f .record	Displays the messages in the .record file.

The .record file should be cleared regularly.

```
cat /dev/null > .record
```

Circuit-Switched Communications

Circuit-switched connections use DB25, DB9, or RJ45 connectors for the physical hardware.
An RS232C connection uses the following wires between a system acting as Data Terminating Equipment (DTE) and a modem acting as Data Communication Equipment (DCE).
The standard pin connections for a DB25 connector are given below.

1	Shield
2	DTE Transmit
3	DTE Receive
4	DTE Request to Send
5	DCE Clear to Send
6	DCE Data Set Ready
7	Ground
8	DCE Carrier Detect
20	DTE Data Terminal Ready
22	DCE Ring Detect

These connections are often made using shielded twisted pair telephone cable.
The shield is connected to pin 1 AT ONE END ONLY to reduce electrical interference and avoid ground currents.
One twisted pair is connected to pins 2 and 7 at both ends while the other twisted pair is connected to pins 3 and 7 at both ends.
The twisted pair reduces magnetic interference.

It is a security risk to connect a modem to a system
without straight-through connections between pins 6, 8, and 20
on both system and modem.
Since these lines are static lines,
they do not need to be twisted with a ground.

When a modem recognizes a carrier signal over a phone line,
the modem drives the Carrier Detect line high.
UNIX expects a Carrier Detect in order for a *getty* to start on that line.
When a user closes a UNIX session, the system drops
the Data Terminal Ready line and the modem drops its carrier signal.
When the Carrier Detect drops, as when the connection is dropped,
the *init* process hangs up the device until another Carrier Detect.
Since a Carrier Detect on a line prevents calling out,
the default system setting is to ignore the Carrier Detect
since the lines are originally considered *call-out* lines.
The blocking of the hardware Carrier Detect is a function of software.

eeprom	Displays the system parameters.

```
ttya-mode=9600,8,n,1,-
ttya-rts-dtr-off=false
ttya-ignore-cd=true
```

Connections between two systems without a modem (with a *null modem*)
require pin 2 of one system to be connected to pin 3 of the other.

Simple connections can be made with shielded, four-wire telephone cable.
At each system pins 6 and 8 are connected to pin 20
so that the DTE Data Terminal Ready on pin 20 sets
both the DCE Data Set Ready on pin 6 and the DCE Carrier Detect on pin 8.

At each system pins 4 and 5 are also connected together
so that the DTE Request To Send on pin 4 substitutes for
the DCE Clear To Send on pin 5.
Many systems can be set to ignore *Requests To Send* and *Clear To Send*.

These physical connections are identified in the software as special files.

ls -li /dev/cul* Lists the traditional *call UNIX* lines that were
used for communications.

ls -li /dev/cua* Lists the traditional *call UNIX* lines
that were used to control the autodialers.

ls -li /dev/tty[a-d]* Lists the existing communications devices.
Note any identical device numbers.

These lines are usually set up for calling out.
It is possible to provide lines that can be used to call in or call out.
The traditional names for these lines are /dev/cua0 for the *call-in* line
and /dev/ttyd0 for the *call-out* line associated with /dev/ttya.
I suggest the more recognizable names /dev/ttyai and /dev/ttyao.

#mv *tty_device tty_device_o* Relabels the call out line.

#ls -l *tty_device_o* Displays the device numbers
of the call-out line.

To provide a call in line that presents a *login:* prompt,
create another device with the same major device number
and a minor device number that is 128 greater than the original.
The lower minor number allows calling out
whenever the device does not register a Carrier Detect
and the higher minor number allows calling in to a *login:* prompt
just after the Carrier Detect is registered.

#mknod *tty_device_i* c *major minor+128*

Creates a new device file.

#chmod a=rw *tty_device_i tty_device_o*

Allows all users to read and write
on this device.

#eeprom ttya_ignore_cd=false — Allows Carrier Detect on this line to be recognized for control.

The new device names must be identified in /etc/ttytab
and a *getty* must be set up on the call in line.

more +/*tty_device* /etc/ttytab — Lists the activities of the communications devices.

A representation of /etc/ttytab is displayed below.

```
#comment
tty_device  "/etc/getty gettytab_entry" termcap_entry  \
                        on  secure
```

The entries in /etc/ttytab depend upon entries in /etc/gettytab and
in /etc/termcap.
This device provides call-in access (on) through the *getty* command
to anyone but the superuser (secure).

It is a security risk to leave any lines marked as *secure*
unless they are in a physically secure area.
If the console is marked as *secure*, the system can be rebooted
into a single user state without a password prompt.

more +/*gettytab_entry* /etc/gettytab — Lists the initial configuration of the devices using this *gettytab_entry*.

A representation of /etc/gettytab is displayed below.

```
#comment
gettytab_entry|getty_name:\
:sp#transmission_speed:\
:im=initial_message\n:lm=appended_login_message:\
:nx=next_gettytab_entry_on_break:\
:tc=continuation_entry:
```

The default terminal type is set to the *termcap_entry*.

more +/*termcap_entry* /etc/termcap

Lists the known capacities of the terminal described by the *termcap_entry.*

A representation of /etc/termcap is displayed below.

```
#comment
termcap_entry|termcap_name:\
:...key_definitions...:\
:...feature_definitions...:\
:tc=continuation_entry:
```

Since there are many termcap entries,
searches complete faster when the most common ones for your system are placed near the start of the file.

Once another system, terminal, or printer is connected,
the connection can be tested.

#stty -a > /dev/*tty_device*

Lists the port settings of the device.
BSD sets the standard ouput device and
System V sets the standard input device.

#echo hello > /dev/*tty_device*

Displays *hello* at the device.

It is possible that the configuration of the remote device does not match the configuration of the port on your system.
The following are some options to the *stty* command to change the configuration of the local device.

speed	1200, 2400, or 9600 bits/second	
cs7	7 bits/character	
cs8	8 bits/character	
-cstopb	one stop bit	
-parenb	disable parity	
parodd	odd parity	
-parodd	even parity	
clocal	ignore Carrier Detect (null modem)	
-clocal	hang up on loss of Carrier	(modem)
hupcl	hang up on close (logout)	
-hupcl	do not hang up on close (logout)	
0	hang up now	

#stty *options* > /dev/*tty_device* Sets the port.

The entry for TERMIO in section 4 of the Reference Manual describes all options and the default settings.
In addition to the local settings,
there are also control, input, and output settings.

This configuration should allow communications with this device.
A device that signals the status of the RS232C lines with a LEDs,
or a Breakout Box, and configurable terminal with a monitor mode
are useful devices for testing the operation of a communications line.

Calling out of a communications port requires a modem and terminal emulation software.
Most UNIX systems provide *cu* and *tip*
to emulate the simplest of terminals.
The *cu* (call unix) command is the more primitive
and is used to test *uucp* connections.
Both require configuration information in several files.

To use *tip*, edit /etc/ttytab changing *on* to *off* for the *tty_device* that you will call out on.

```
        tty_device  "/etc/getty gettytab_entry" termcap_entry \
        off  secure
```

more /etc/remote	Lists the hosts available for connection through particular *tty_devices* .

The information in /etc/remote need not be used.

setenv REMOTE remote	Identifies a personal host-device file for tip.
vi remote	Create your host-device file for tip.

```
# Device Characteristics
tip0:\ Default device name for tip
    :dv=/dev/tty_device:\  Device to use for communications
    :br#2400:\             Baud rate (bits/second)
    :du:at=hayes:          Dialup and use Hayes autocall type
tip2400:\                  Device name for tip -2400
    :tc=tip0:              Continuation with tip0
#Host Characteristics
micom:\                    Host name for tip micom
    :pn=753-3000:\         Phone number
    :cm=Space:\            Connect message (gives micom menu)
    :tc=tip0:              Continuation with tip0
myhost:\                   Host name for tip when $HOST is set
    :pn=@:\                Phone number in /etc/phones or $PHONES
    :tc=tip2400:           Continuation with tip2400
```

setenv HOST myhost	Identifies a default host for tip.
more /etc/phones	Lists phone numbers for tip.
setenv PHONES phones	Identifies a personal phone number file for tip.

vi phones	Creates your phone number file.

#comment	
myhostTab3-3000	First number to try.
myhostTab753-3000	Second number to try.

The file /etc/phones should be unreadable to secure its information.

Tip has several internal commands that govern its operation.

~?	List commands.
~#	Send a Break.
~>	Send file to remote host.
~<	Capture file.
~c mydir	Change directory.
~s all	List variables.
~s sc	Start session script in *tip.record*.
~s !sc	Stop session script.
~.	Quit.

Tip can use a .tiprc file for setting its variables
like host, phones, and record.
It records call activity in /var/adm/aculog.

more /var/adm/aculog	Displays the recent tip activity.

Tip does not work, sending a *all ports busy* message, when
the Carrier Detect line is held high,
there is a getty on the port,
or the uucp lock file /var/spool/uucp/LCK..*tty_device*
or /var/spool/locks/*tty_device* exists.

Packet-Switched Communications

Packet-switched communications networks allow
faster transmission speeds and
simultaneous connections to multiple systems
including the local system itself.
Each Sun Sparcstation needs a transceiver with a T-connector
to connect links of RG58 coaxial cable between T-connectors.
To maintain the signal timing,
the coax links should be at least 2.5 meters apart.
The extreme ends of the links should be connected to 50-ohm terminators
and at most 180 meters apart.

Connections of these systems consist of:
an network interface (Ethernet) card with a unique 48-bit address
(XX:XX:XX:XX:XX:XX), an 32-bit internet protocol address (XXX.XXX.XXX.XXX),
a transmission protocol, and a port.
The address, transmission protocol, and port is called a *socket*
The *internet protocol address* consists of a networkaddr and a hostaddr.
The type of the address,
defined in the first digits by the location of the first zero,
identifies the extent of the names in the address.

Address Class	Addresses	Network	#	Host	#
A (0*)	0-127	XXX.	127	XXX.XXX.XXX	16M
B (10*)	128-191	XXX.XXX.	16K	XXX.XXX	64K
C (110*)	192-223	XXX.XXX.XXX.	2M	XXX	256

The addresses 0.0.0.0 and 255.255.255.255 are not available.
Network and host numbers of all zeros or ones should not be used as well.

hostname	Displays the local hostname.
domainname	Displays the local domainname.

The *domainname* is set in /rc.local.
The domainname can be synonmous with the network name.

The /etc/hosts file is used to find various hosts on the network.

more /etc/hosts   Lists known internet addresses
on the local network.

A representation of /etc/hosts is displayed below.

#comment
internet_address hostname alias alias ...

Notice the address for *localhost* (127.0.0.1);
it can be used to connect to the local system.

#arp -a   Displays the current table of hostnames,
addresses, and ethernet addresses.

The /etc/ethers file is also used to find various hosts on the network
by their ethernet adapter address.
It is used by Reverse Address Resolution Protocol (RARP) servers
for diskless workstations.

more /etc/ethers   Lists known ethernet addresses
on the local network.

A representation of /etc/ethers is displayed below.

#comment
ethernet_address hostname alias alias ...

The connections provided by a network allow users
to have access to several systems.
It is possible to provide simple, consistent access
to all these equivalent systems.

more /etc/hosts.equiv   Lists hosts whose users appear in
the local /etc/passwd file
that do not need a password
for *rlogin* and *rsh* commands.

The .rhosts file in a users directory can override this file
providing similar access for an individual user.
It is a security risk to allow outsiders to read these files
and identify which system allow a user to connect without a password.

more /etc/networks	Lists known networks on the local network.

A representation of /etc/networks is displayed below.

```
#comment
network_entry  network_number  aliases
```

Subnets can be defined on larger networks in order to simplify routing.
Subnets are defined by network masks.

more /etc/netmasks	Lists known subnets on the local network.

```
#comment
network_number  network_mask
```

0xFF000000	255.0.0.0	Class A Mask
0xFFFF0000	255.255.0.0	Class B Mask
0xFFFFFF00	255.255.255.0	Class C Mask

The network number has zeros in place of a host number.
The network mask has ones in place of the subnet number and
zeros in place of a host number.
The network mask identifies those systems that are expected to be on
the same cable so that routing through other system is not needed.

more /etc/gateways	Lists distant gateways for routing.
/etc/ifconfig le0	Displays the internet address, the network mask, and the broadcast address of the Lance Ethernet (le) controller.
more /etc/inetd.conf	Lists services provided for Internet requests.

A representation of the information in /etc/inetd.conf is displayed below.

#comments
*services_entry socket_type protocol_entry *
wait_status user_id program

more /etc/protocols Lists network protocols such as Terminal Control Protocol (TCP) for virtual direct connections and the User Datagram Protocol (UDP) for connectionless communication.

A representation of the information in /etc/protocols is displayed below.

#comment
protocol_entry protocol_number aliases

more /etc/services Lists non-Remote-Program-Call (RPC) network services.

A representation of the information in /etc/services is displayed below.

#comments
service_entry port_number/protocol_entry aliases

There are several tests that can be used to check the availability of the network.

ps -aux l grep ‘d ‘ Lists the daemons on the system.

The inetd daemon must be running to start up the other network daemons.

telnet localhost Provides *remote access* to the local system.

CTRL] quit Exits telnet.

telnet *hostname* Provides *remote access* to the local system.

CTRL] quit Exits telnet.

ping *ip_address*	Tests communication with another system.
netstat	Displays the network status.
netstat -i	Displays Ethernet interface status.
netstat -ian	Displays the status of all Ethernet interfaces.
netstat -s	Displays protocol statistics.
netstat -r	Displays the routing table.
netstat -rs	Displays routing statistics.
traffic	Displays the network activity.
nslookup	Starts an interactive session to interrogate name servers on the network.

Printing

Printing is handled on a UNIX system as a service to all users.
Printing to local printers as well as printing to printers
on remote systems is possible.
The */usr/ucb/lpr* command queues up print requests in /var/spool/*printername*
as data files (df###) and control files (cf###).

ls /var/spool/lp	Lists print files for the lp printer.

The */usr/lib/lpd* print server processes the request as it finds them
creating status files and lock files
to describe and control the printer when in use.
The original *lpd* daemon starts other versions of itself
to service each printer.

ps -ax | egrep "lpd|PID"	Lists the print daemon process status.

The *lpd* daemon also listens to the socket /dev/printer
to service remote print requests.
Sockets allow network connections to be treated as files.
A socket is a network address, a host address, and a TCP port.

Printing can be controlled with the *lpc* command.

lpc help	Lists possible lpc commands.
lpc status all	Displays the status of all printers.
lpc topq *printername job#*	Identifies the next request to print.

Controlling printing on a UNIX system involves
controlling the queues, the daemons, and the printers.

lpc stop *printername*	Stops a printer without disabling its queue.
lpc restart *printername*	Restarts the daemon for a printer.
lpc down *printername message*	Stops a printer, disables its queue from accepting jobs, and terminates a daemon.
lpc up *printername*	Restarts a daemon, queue, and printer.

The available printers are defined in /etc/printcap.

```
more /etc/printcap          Lists the available printers and
                            their characteristics.
```

```
lp|printername:\          Printer names
:lp=tty_device:\          Device driver--nothing if remote
:rm=hostname:\            Remote system for printing
:rp=printername:\         Printer name on remote system
:br#9600:\                Transmission speed for local printer
:ms=+/-modes,...:\        Set/clear local communication modes
:sd=/var/spool/lp:\       Spool directory
:lf=/var/adm/lp/log:\     Log file other than /dev/console
:of=/local/of:\           Output filter (not used if remote)
:tr=\012:\                Trailing Form Feed
:sh                       Suppress header page
```

The /var/spool/lpd.lock file contains the process ID of the process
that controls the printer.

```
more /var/spool/lpd.lock    Displays a process ID.
```

The existence of this file stops printing.

The printername can be used with the -P option or
set in the PRINTER variable.
The spool directories must exist as /var/spool/*printername*.

Network File Service

A UNIX host can provide access to its files (file service)
to a remote client as though the files were local to the client.
The file server provides access to its files
through its mountd and multiple nfsd daemons
and the client gets access through its multiple biod daemons.

On the server, directories (and files) are made accessible (exported) with several options.

ro	Accessible as readonly
root=*hostname*:...	Accessible to root on *hostname* with local superuser privileges
access=*hostname*:...	Accessible to *hostname* only

The *exportfs* command makes these directories (and files) accessible by placing information in the /etc/xtab file.

#exportfs -o *options pathname*	Exports an individual pathname.
more /etc/xtab	Lists accessible directories and files.
#exportfs -u *pathname*	Removes access.

The /etc/exports file (644) maintains a list of directories and files for client access.

more /etc/exports	Displays regularly accessible directories and files.

A representation of /etc/exports is given below.

pathname -option,option=value,option=value:value

During the system startup /etc/rc.local runs exportfs which examines /etc/exports.

#exportfs -a	Makes all directories in /etc/exports accessible.
exportfs	Displays current contents of /etc/xtab.
more /etc/xtab	Displays currently accessible directories.
showmount -e	Lists exports on local host.

The server must have a *mountd* and several *nsfd* daemons present
to serve client requests for file access.

ps -ax | egrep ‘mountd|nfsd’	Displays all mountd and nsfd daemons.
#nfsd 8 &	Starts eight daemons to service requests.

The *mountd* daemon maintains information about client access
in /etc/rmtab on server.

more /etc/rmtab	Lists all mounts by clients.
showmount -a	Lists all mounts by clients.
showmount -d	Lists directories mounted by clients.

When a client requests access to a server through a mount request,
the *rpc.mountd* daemon on the server examines the request.

#mount -t nfs -v -o rw,hard,nosiud,intr *hostname*:*pathname mount_point*

Requests read-write file access which
guaranties writes, does not accept set userid,
and allows keyboard interrupts to release
the client when the server dies.

#mount -t nfs -v -o ro,soft,nocto *hostname*:*pathname mount_point*

Requests readonly file access which
does not hang the client when the server dies
and does not update file information.

Any directory can serve as a *mount point*.

mount	Lists filesystems mounted as a client.
umount *mount_point*	Releases a file system.

The /etc/fstab file maintains a list of regularly mounted local and remote file systems.

more /etc/fstab — Lists the regularly mounted file systems.

A representation of remote mounts in /etc/fstab is given below.

```
hostname:pathname    mount_point    nfs    rw,hard,nosiud,intr \
0    0
hostname:pathname    mount_point    nfs    ro,soft,nocto \
0    0
```

more /etc/mtab — Displays mounted filesystems.

showmount *hostname* — Displays recent remote mounts on a server.

The *biod* daemons on a client are not necessary,
but they improve perfromance of the Network File System (NFS).

ps -ax | grep biod — Displays the biod daemons on the system.

#biod 4 & — Starts four biod daemons on the client.

The client should also have a *portmap* daemon
which was started at bootime.

ps -ax | grep portmap — Displays the portmap daemon on the system.

The file service may not be operational.

rpcinfo -p *hostname* — Checks server availability.

rpcinfo -u *hostname* mount — Checks the availability of a mountd daemon on the server.

showmount -e *hostname* — Lists exports on a server.

showmount -d *hostname* — Lists directories mounted by clients on a server.

UNIX-to-UNIX Copy

The UNIX-to-UNIX Copy (uucp) commands provide unattended file transfer between systems.
These commands were developed
for use with direct or circuit-switched connections
which were slow and not always immediately available.
However, they work just as well with packet-switched connections
especially when you want to automate the transfer of many large files.

Any changes to the the configuration of the uucp system
should be made wih the *uucp administrative username*.

grep uucp /etc/password   Displays the uucp account information.

uucp:*password*:*userid*:*groupid*:uucp administator:\
/usr/lib/uucp:/bin/csh

Any connection and file transfers with another system
should be made using the *uucp operation username* for that system.

u_hostname:*password*:*userid*:*groupid*:uucp operations:\
/var/spool/uucppublic:/usr/lib/uucp/uucico

The local uucp operator uses the *uucico* command
to login to another system as a uucp operator on that system.
The remote uucp operator account runs the *uucico* command on login
instead of a shell.
These *uucico* commands copy in and copy out data and executable files
using reliable file transfer methods,
and execute a *uuxqt* command on each system to start processes
to handle the executable files.

ls /usr/lib/uucp	Displays the uucp operations commands.

uucico	Handles accessing and access from remote systems and file transfers to and from remote systems.
uusched	Schedules the uucico activities.
uuxqt	Handles file execution requests on the local system.
uucp	Creates file transfer requests.
uux	Creates remote execution requests.
uustat	Reports on uucico activities.

Uucp commands are run on a regular basis as crontab entries.

#su - uucp -c crontab < *crontab.file*

Starts a session for the uucp administrator.
and schedules activities for uucp.

Sun uses four crontab files.

uudaemon.poll	Schedules copies to or from remote systems and any subsequent remote executions.
uudaemon.hour	Starts up copies to or from remote systems and any sussequent local executions.
uudaemon.admin	Mails status reports to the uucp username.
uudaemon.cleanup	Removes old or failed work files and old logs.

Other systems use one or more shell scripts that execute the *crontab* command to set up the uucp crontab entry.

The *uucp*, *uux*, and *mail* commands set up work for the *uucico* and *uuxqt* commands in /var/spool/uucp.

ls -a /var/spool/uucp/*hostname*	Lists the control, data, executable, and temporary files for accessing a system.

A directory for the local system must exist for uucico to function.

#mv /var/spool/uucp/noname /var/spool/uucp/*hostname*

The unattended activities of uucico and uuxqt are recorded in files
in /var/spool/uucp as well.

ls /var/spool/uucp/.Log	Tracks operations for each remote system.

These files were previously combined in a LOGFILE file.

ls /var/spool/uucp/.Admin	Tracks operations for the local system.

These files were previously combined in a SYSLOG file.

ls /var/spool/uucp/.Status	Tracks errors.

These files were previously combined in an ERRLOG file.

uucp/uustat -q -p	Displays uucp system status.
uucp/uulog *hostname*	Displays uucp activity for a particular system.

Configuration files for uucp are found in /etc/uucp.

ls -a /etc/uucp	Displays the uucp configuration files.

Uucico communicates with remote systems through direct connections, through modem connections, and through network (TCP) connections.

The /etc/uucp/Dialers file (444) describes the procedures necessary to initialize and control various modems.

more /etc/uucp/Dialers	Displays known modem control information.

A representation for a typical /etc/uucp/Dialers entry is given below.

```
#comment
dialers_entry  WwPp  modem_script
```

The *WwPp* field lists the substitutions for the *wait_for_tone* character and the *pause* character of each modem.

The /etc/uucp/Devices file (444) describes the ports
through which uucico can communicate.
Earlier versions of uucp used an L-devices file.

more /etc/uucp/Devices   Lists devices used to communicate via uucp.

A representation for typical /etc/uucp/Devices entries is given below.

```
#comment
devices_entry     tty_device     -     speed     dialer_entry
ACU               tty_1          -     1200      hayes
DIRECT            tty_1          -     1200      Direct
ACU               tty_1          -     9600      hayes
Direct            tty_1          -     9600      Direct
hostname          tty_2          -     9600      Direct
Direct            tty_2          -     9600      Direct
TCP               -              -     Any       TCP
```

The *devices_entry* is identified as ACU for modems (autocall units),
Direct or *hostname* for direct (null modem), or TCP for networks.
Each callout line should have two entries:
the second entry is used by the *cu* command.

```
#cu -ltty_device  -sspeed  modem_script
```

The *uucp* commands implicitly understand Direct and TCP *device_entries*.

The /etc/uucp/Systems file describes how to get to
various remote systems using the /etc/uucp/Devices file.
Earlier versions of uucp used an L.sys file.

more /etc/uucp/Systems Lists hosts that communicate via uucp.

A representation of typical /etc/uucp/Systems entries is displayed below.

```
#comment
hostname  schedule     devices_entry  speed  phone     login_script
hostname  Any          hostname       9600   -         in: Unm rd: pw
hostname  Never        ACU            1200   5551212   in: Unm rd: pw
hostname  Wk1700-0800  ACU            1200   Dc_entry  in: Unm rd: pw
hostname  Sa,Su        ACU            2400   Dc_entry  in: Unm rd: pw
hostname  Any          TCP            -      hostname  in: Unm rd: pw
```

The schedule field can be of the following tokens

Any Never Wk Su Mo ...

possibly followed by two 24-hour times separated by a dash
for example, Wk1700-0800 to indicate 5 PM through 8 AM weekdays.
Multiple entries are separated by commas.
Any part of the entries in the phone field can be replaced with
entries from /etc/uucp/Dialcodes.
The *phone* field is sent to the modem as part of the *modem_scripts*
in /etc/uucp/Dialers.
The *login_script* is a sequence of words exchanged by the systems on login
separated by spaces.

ogin: *u_hostname* ssword: *password*

Since the /etc/uucp/Systems file contains information to access
many other systems, it is a major security risk for these systems
when the file is readable by anyoine but the uucp administrator (400).

uuname Lists hosts that can be accessed with uucp.

uuname -l Lists the local hostname.

Uucico executes the /etc/uucp/remote.unknown file
for hosts that are not described in the /etc/uucp/Systems file
when those hosts attempt to access the local system.

Security for the uucp access is provided by
the /etc/uucp/Permissions file (400).

more /etc/uucp/Permissions	Lists permissions for call in access (LOGNAME=) and callout access (MACHINE=).

Default permissions allow access only to the /var/spool/uucppublic directory.

The /etc/uucp/Poll file describes the hours that various hosts
are called by *uusched*.

hostname Tab *hour hour* ...

The /etc/inetd.conf and /etc/services must have uucp entries
in order for uucp to work over the network.

```
uucp  stream  tcp  nowait  root  /user/etc/in.uucpd  in.uucpd

uucp  540/tcp  uucpd
```

The *uucp* programs can be given exclusive rights to a device.

#chown uucp.*group tty_device*	Makes uucp the individual owner of the communications port.
#chmod 600 *tty_device*	Allows uucp exclusive read-write access to the communications port.

The working directories for uucp must have set permissions.

#chmod 711 /var/spool/uucp/*hostname*

Limits group owner and outsiders
extend permission.

uucheck -v Verifies all file permissions.

The uucp configuration can be tested with the *cu* command.

#cu -d -l *devices_entry dialer_entry* Attempts to call out on a device.

~. Exits cu.

#cu *hostname* Displays *login:* prompt of remote system
if successful.

Login as the uucp operator on the remote system to get a *Shere=* message.
Use ~. to disconnect.

The uucico or uutry commands can also be used to test the connection.

#/usr/lib/uucp/uucico -r1 -x4 -s*hostname*

#/usr/lib/uucp/uutry*hostname*

Suspend uucico and kill it when you are finished; it can't be interrupted.
Remove any *hostname* files in the /var/spool/uucp/.Status directory
if retries are prevented.
These commands allow the identification of information
for the *login_script* placed in the /etc/uucp/Systems file.

Lock files in /var/spool/locks can prevent access to a device.
They contain the process ID of the controlling process.
The uusched deamon should periodically clean locks.

/usr/lib/uucp/uucleanup Clears /var/spool/uucp.

The public directory /var/spool/uucppublic must be cleared manually.