Displaying Host and User Information [6]

This section describes the TCP/IP utilities you can use to display host and user information about the network. Table 3 summarizes the utilities.

Table 3. Functions of TCP/IP utilities for information display

Utility	Function
`finger`(1B)	Lists information about one user or all users on the host.
`hostname`(1)	Prints the name of the local host system.
`ping`(8)	Checks connections to other TCP/IP hosts in the network.

Using the `finger` utility
6.1

When no option or operand is specified after the `finger` command, the `finger` utility displays the following information about each user currently logged in to the local host:

* Login name

* Full name

* Write status (given as an asterisk (*) before the terminal name if write permission is denied)

* Terminal name

* Idle time (represented in minutes if it is a single integer, hours and minutes if a colon (:) is present, or days and hours if a `d` is present in the date field)

* Login time

When you specify the appropriate option or operand after the `finger` command, the `finger` utility displays a longer list. The list can include all of the information in the preceding list as well as the following items:

- User's home directory and login shell.

- Any plan that the user has placed in file `.plan` in his or her home directory, and the project on which he or she is working from the file `.project`, also in the home directory; for `.project` file, text must begin on line 1 of the file for `finger` to read it.

  Note: To access the listed information in a UNICOS MLS system, your security label must dominate the security label of the information. See the *UNICOS Multilevel Security (MLS) Feature User's Guide*, publication SG–2111, for more information.

  A `nobody` account must exist on the system also. However, the label range of the account may not allow `finger` to be used across the network. See your system administrator to find out if you can use `finger`.

You can specify one or more user names on the `finger` command, which causes it to print information about each user specified. The `finger` command accepts account (login) names, as well as first and last names.

Invoke the `finger` command with the following format:

```
finger [–b] [–f] [–h] [–i] [–l] [–m] [–p] [–q] [–s] [–w] [name . . .]
```

–b	Gives a briefer long-form list of users.
–f	Suppresses the heading in the short and quick output format requested by the –s or –q option.
–h	Suppresses printing of the `.project` file.
–i	Gives the same quick list as the –q option, but includes idle time.
–l	Gives the long-form list; this is the default mode.
–m	Matches arguments only for login name.

-p   Suppresses printing of the `.plan` file.

-q   Gives a quick list that includes only login name, terminal name, and login time.

-s   Gives a short-form list of users.

-w   Suppresses printing of the full name in the short-form list format.

name   Specifies the login name or the full name of a validated user on the local host. You can specify any number of users; separate names with blanks.

Example 1:

In this example, the `finger` command is entered without arguments or options. It lists the login name, full name, terminal name, write status, idle time, login time, and office location for each current user on the local host.

```
$ finger

Login     Name        TTY         Idle        When          Office
mpd       MelDio      tty00       41          Thu 17:24     14
karen     K.Jone      ttyp2       2:05        Thu 16:24     2
mpd       MelDio      *ttyq1      4:35        Thu 13:03     14
```

Example 2:

In this example, the `finger` command is followed by a login name (`mpd`). In response, `finger` lists the login name, full name, terminal name, shell, write status, idle time, login time, and plan for user `mpd` on the local host. By default, the listing is displayed in long format.

```
$ finger mpd

Login name: mpd            In real life: Mel Dio
Directory: /u1/mpd       Shell: /bin/sh
On since Dec 19 17:24:42 on ttyp000 from
mywork
2 hours 16 minutes Idle time
Plan:
To make a plan by Tuesday
```

Using the hostname utility

6.2

The `hostname` utility prints the name of the current host system. Systems personnel can use the `hostname` command to set the name of the host system.

To invoke the `hostname` utility, enter the following command:

```
hostname
```

Example:

```
$ hostname
crayhost
```

Using the ping utility

6.3

The `ping`(8) utility provides a simple way to find out whether a host connected to the TCP/IP network is functioning. It also gives you some idea of the performance you can expect when communicating with the specified host. The `ping` utility sends out an `ICMP ECHO_REQUEST` packet (called a *ping*) to a host or gateway; it then waits for an `ICMP ECHO_RESPONSE` packet. Because every IP host is required to respond to ICMP packets, the absence of a response indicates that the host is not up. If a response is received, the amount of time it takes to get the response indicates the speed with which the host is responding.

The command format is as follows:

/etc/ping [-drv *host* [*packetsize* [*count*]]]

-d   Sets the socket-level debugging option
               (SO_DEBUG).

-r   Sets the SO_DONTROUTE option to bypass the
               routing tables.

-v   Turns on the verbose mode for packet
               descriptions; tells you whether you receive a
               packet other than the one the system expected.
               By default, verbose mode is off.

host   The official host name or Internet address of a
               remote host whose status you want to check.

packetsize   Specifies the number of bytes in the packet.
               Values can range from 0 to 4096; the default
               length is 64 bytes.

count   Specifies that the ping utility should send the
               echo packet *count* times. If *count* is 0 or not
               specified, the default is one packet per second
               until responses are no longer sent or until you
               stop ping with a SIGINT signal.

The ping utility sends one datagram per second and displays
one line of output on your screen for every ECHO_RESPONSE
returned. If no response is received, no lines are displayed. The
ping utility also computes round-trip times and packet-loss
statistics. These are displayed when all responses are received;
when the program reaches the maximum number specified in
count; or if the program is terminated with a SIGINT signal.

If you have not specified the maximum number of datagrams in
the *count* operand, or if the host to which you are sending a ping
goes down, the ping utility can loop forever unless you end the
process by telling the shell to send a SIGINT signal. Do this by
entering the user-definable character that sends the signal
(usually CONTROL-c).

An example of ping, using the default settings, follows. The output shows that the remote host, Internet address 128.1.0.1, is responding to the local host. The statistics summary shows the number of packets transmitted, the number of packets received, the percentage of packets lost, and a summary of the response time.

```
$ /etc/ping 128.1.0.1
PING 128.1.0.1: 56 data bytes
64 bytes from 128.1.0.1: icmp_seq=0. time=7. ms
64 bytes from 128.1.0.1: icmp_seq=1. time=12. ms
64 bytes from 128.1.0.1: icmp seq=2. time=16. ms
64 bytes from 128.1.0.1: icmp_seq=3. time=6. ms
64 bytes from 128.1.0.1: icmp_seq=4. time=7. ms
----128.1.0.1 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 6/9/16
```

The following example shows output with a packet size of 4096 bytes and a count of 10, with 1 packet dropped.

```
$ /etc/ping  128.1.0.1  4096  10
PING 128.1.0.1: 4069 data bytes
4096 bytes from 128.1.0.1: icmp seq=0. time=41. ms
4096 bytes from 128.1.0.1: icmp seq=1. time=22. ms
4096 bytes from 128.1.0.1: icmp seq=3. time=32. ms
4096 bytes from 128.1.0.1: icmp seq=4. time=31. ms
4096 bytes from 128.1.0.1: icmp seq=5. time=32. ms
4096 bytes from 128.1.0.1: icmp seq=6. time=30. ms
4096 bytes from 128.1.0.1: icmp seq=7. time=27. ms
4096 bytes from 128.1.0.1: icmp seq=8. time=34. ms
4096 bytes from 128.1.0.1: icmp seq=9. time=27. ms

----128.1.0.1 PING Statistics----
10 packets transmitted, 9 packets received, 10% packet loss
round-trip (ms)  min/avg/max = 22/31/41
```

The last example shows the `ping` output when a remote host is not responding.

```
$ /etc/ping 192.0.0.1 4096 10
PING 192.0.0.1: 4069 data bytes

----192.0.0.1 PING Statistics----
10 packets transmitted, 0 packets received, 100% packet loss
```