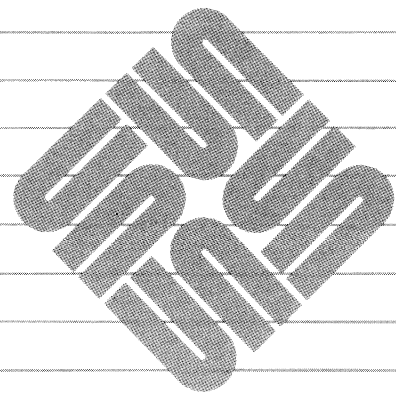




# Using the Network: Beginner's Guide



Sun Workstation® and Sun Microsystems® are registered trademarks of Sun Microsystems, Inc.

SunView™, SunOS™, Sun386i™, and the combination of Sun with a numeric suffix are trademarks of Sun Microsystems, Inc.

UNIX is a registered trademark of AT & T Bell Laboratories.

All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations.

Copyright © 1987, 1988 by Sun Microsystems, Inc.

This publication is protected by Federal Copyright Law, with all rights reserved. No part of this publication may be reproduced, stored in a retrieval system, translated, transcribed, or transmitted, in any form, or by any means manual, electric, electronic, electro-magnetic, mechanical, chemical, optical, or otherwise, without prior explicit written permission from Sun Microsystems.

---

# Contents

<b>Chapter 1 Overview</b> .....	<b>3</b>
1.1. What Is a Network? .....	3
NFS .....	3
The Yellow Pages .....	6
<b>Chapter 2 Login Access to Other Machines</b> .....	<b>11</b>
2.1. Trust Amongst Machines on a Network .....	11
/etc/hosts .....	11
/etc/hosts.equiv .....	12
2.2. Remote Login With <code>rlogin</code> .....	13
Logging In to Another Machine With <code>rlogin</code> .....	13
<code>rlogin</code> to a Machine Where You Don't Have a Home Directory .....	13
<code>rlogin</code> to a Machine As Someone Else .....	14
<code>rlogin</code> to a Non-Existent Machine .....	14
Aborting an <code>rlogin</code> Connection .....	14
Suspending an <code>rlogin</code> Connection .....	15
Verifying Your Identity with <code>who am i</code> .....	16
2.3. Remote Login With <code>telnet</code> .....	16
Logging In to a Machine Running Another Operating System With <code>telnet</code> .....	16
<code>telnet</code> to a Non-Existent Machine .....	17
Aborting a <code>telnet</code> Connection .....	17
Suspending a <code>telnet</code> Connection .....	18

<b>Chapter 3</b> Mounting Remote Filesystems .....	<b>23</b>
3.1. The <code>mount</code> Command .....	23
3.2. The <code>/etc/fstab</code> File .....	24
The <code>umount</code> Command .....	25
3.3. Options to <code>mount</code> and <code>umount</code> .....	26
3.4. The Automounter .....	27
3.5. Who's Mounting This System? <code>showmount</code> .....	29
<b>Chapter 4</b> Issuing Commands to Remote Machines .....	<b>33</b>
4.1. Status Information: <code>ping</code> , <code>rup</code> , and <code>perfmeter</code> .....	33
Dead or Alive: <code>ping</code> .....	33
Remote Uptime: <code>rup</code> .....	34
Performance Meter: <code>perfmeter</code> .....	34
Remote User Information With <code>rusers</code> .....	35
The <code>finger</code> Command .....	36
4.2. Remote File Copy With <code>rcp</code> and <code>ftp</code> .....	36
Copying from Another Machine to Your Machine Using <code>rcp</code> .....	37
Copying from Your Machine to Another Machine Using <code>rcp</code> .....	37
Copying Directories With <code>rcp</code> .....	38
Copying from Another Machine to Your Machine Using <code>ftp</code> .....	38
Copying from Your Machine to Another Machine Using <code>ftp</code> .....	39
4.3. Remote Command Execution With <code>rsh</code> .....	40
4.4. Network Error Notifications .....	41
<b>Chapter 5</b> Using Dialup Terminals .....	<b>45</b>
5.1. Equipment .....	45
5.2. Considerations .....	45
Phone Number .....	46
Baud Rate .....	46
5.3. The Process .....	46
5.4. Remote Login With <code>tip</code> .....	46
Establishing a Connection With <code>tip</code> .....	47
Problems Establishing a Connection With <code>tip</code> .....	47

<b>Appendix A</b> Command Summary .....	<b>51</b>
<b>Appendix B</b> Glossary .....	<b>57</b>
<b>Index</b> .....	<b>61</b>



---

## Tables

Table 4-1 Network Error Notifications: Problems and Solutions .....	42
---	----







---

## Figures

Figure 1-1 A Non-Network Environment .....	4
Figure 1-2 A Typical Network Set-Up .....	5
Figure 2-1 An <code>/etc/hosts</code> File .....	12
Figure 2-2 <code>/etc/hosts.equiv</code> File .....	12
Figure 2-3 <code>rlogin</code> : Logging In to Another Machine .....	13
Figure 2-4 <code>rlogin</code> : Logging In to a Machine Without a Home Directory .....	13
Figure 2-5 <code>rlogin</code> : Logging In to Another Machine With a Different Username .....	14
Figure 2-6 <code>rlogin</code> : Logging In to a Non-Existent Machine .....	14
Figure 2-7 <code>rlogin</code> : Aborting a Connection .....	15
Figure 2-8 Aborting to an Intermediate <code>rlogin</code> .....	15
Figure 2-9 <code>rlogin</code> : Suspending a Connection .....	15
Figure 2-10 Discovering Your True Identity .....	16
Figure 2-11 <code>telnet</code> : Logging In to a Machine Running Another Operating System .....	17
Figure 2-12 <code>telnet</code> : Logging In to a Non-Existent Machine .....	17
Figure 2-13 <code>telnet</code> : Aborting a Connection .....	18
Figure 2-14 <code>telnet</code> : Suspending a Connection .....	19
Figure 3-1 Mounting Remote Filesystems .....	24
Figure 3-2 Mounting to A Different Directory .....	24
Figure 3-3 Mounting to a Nonexistent Directory .....	24
Figure 3-4 A Sample <code>fstab</code> File .....	25

Figure 3-5 Unmounting a Filesystem .....	26
Figure 3-6 Unmounting Selected Filesystems .....	26
Figure 3-7 Starting the Automounter .....	28
Figure 3-8 Using the Automounter: I .....	28
Figure 3-9 Using the Automounter: II .....	28
Figure 3-10 <code>showmount -a</code> .....	29
Figure 4-1 Using <code>ping</code> .....	33
Figure 4-2 <code>rup</code> : Uptime and Load Average for a Specific Machine .....	34
Figure 4-3 <code>rup</code> : Uptime and Load Average for the Machines on Your Ethernet .....	34
Figure 4-4 <code>perfmeter</code> .....	34
Figure 4-5 <code>rusers</code> .....	35
Figure 4-6 <code>rusers</code> for a Specific Machine .....	35
Figure 4-7 <code>rusers</code> : Detailed Information .....	35
Figure 4-8 <code>finger</code> .....	36
Figure 4-9 <code>rcp</code> : Copying from Another Machine to Your Machine .....	37
Figure 4-10 Copying and Renaming with <code>rcp</code> .....	37
Figure 4-11 <code>rcp</code> : Copying from Your Machine to Another Machine .....	38
Figure 4-12 <code>ftp</code> : Copying from Another Machine to Your Machine .....	39
Figure 4-13 <code>ftp</code> : Copying from Your Machine to Another Machine .....	40
Figure 4-14 Executing a Command on Another Machine .....	40
Figure 5-1 <code>tip</code> : Establishing a Connection .....	47
Figure 5-2 <code>tip</code> : Call Fails .....	47
Figure 5-3 <code>tip</code> : Synchronization Problem .....	47

---

## Preface

This document explains how to use the network facilities available on the Sun Workstation. We assume that you have a fair amount of experience with the Sun Workstation, and the SunOS operating system.

We provide examples so you can learn how to log in to and execute commands on remote machines, as well as to gain dialup access to a machine. However, we don't present detailed explanations of the inner workings of the network programs. As in each of the *Beginner's Guides*, we refer to the other Sun documentation, drawing a road map for you to follow when you wish to learn more about a certain topic.

In addition to explaining remote logins, remote command execution, and dialup access, *Using the Network: Beginner's Guide* provides a command summary and glossary for easy access to the material.

### Prerequisite documents

We suggest you read the Beginner's Guides in the following order:

*Getting Started with SunOS: Beginner's Guide*  
*SunView 1 Beginner's Guide*  
*Mail and Messages: Beginner's Guide*  
*Using the Network: Beginner's Guide*  
*Setting Up Your SunOS Environment: Beginner's Guide*  
*Doing More with SunOS: Beginner's Guide*  
*Self-Help with Problems: Beginner's Guide*

### Companion Documents

You will also find the following manuals to be useful:

*SunOS Reference Manual*  
*System and Network Administration*

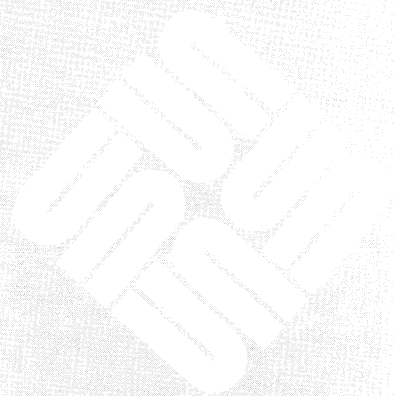




---

## Overview

Overview .....	3
1.1. What Is a Network? .....	3
NFS .....	3
The Yellow Pages .....	6





---

## Overview

Networks provide you with the opportunity to use other machines while logged in on your own machine. You can log in to other machines, or you can execute commands without logging in to other machines.

### 1.1. What Is a Network?

A *network* is a connection between machines, allowing them to transmit information to one another. Networks are often referred to as being *local area networks* (LANs), which range over a small area, generally less than a few thousand feet; *wide area networks* (WANs), which can span thousands of miles; or *campus area networks* (CANs), which are intermediate in size.

Networks may themselves be groups of networks; such a super-network is called an *internetwork*. For example, you may be part of a network of machines on your building's floor, and part of an internetwork connecting your local network with similar networks across the country. As the difference between a network and an internetwork is generally invisible to the user, we will use the term "network" to refer to both networks and internetworks.

Machines participating in a network communicate using a network *protocol*, or shared network language, to transmit the appropriate information to the right place. An *internetwork protocol* — sometimes referred to as a *gateway* or *relay* — links networks together.

#### NFS

If you are using a Sun Workstation, you are not necessarily on a network. You may, for example, bring your machine up *single-user*, or you may simply be on a machine which is not connected to any network.

Still, most Sun users are on a network of one sort or another. The *Network File System*, or NFS<sup>™</sup>, allows users on different machines to share files across a network.<sup>1</sup> NFS is both a communications protocol — a set of rules machines follow for talking with each other — and a collection of software utilizing that protocol. NFS makes use of services provided by a network to allow for transparent transfer and access of files between machines; that is, NFS allows you to treat some files on other machines as though they were on your own machine. NFS was developed by Sun Microsystems and has been adopted by a number of other companies.

---

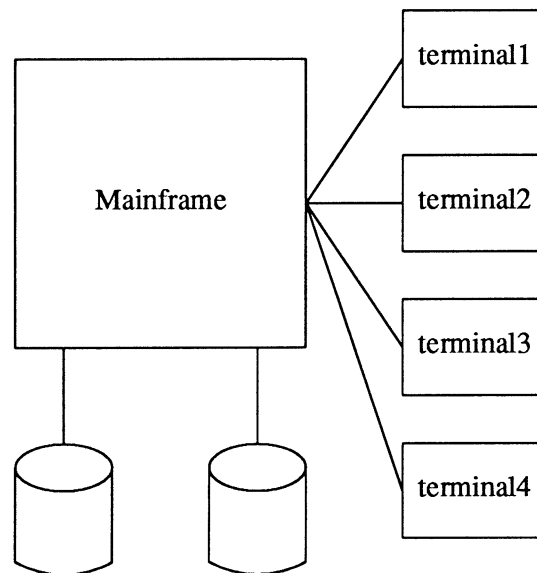
<sup>1</sup> NFS is a trademark of Sun Microsystems.

NFS is not limited to machines which run SunOs; many other systems, from personal computers to super-computers, can use it to share files across a network.

NFS doesn't exist on the network — a network is just a lifeless collection of wires, cables, phones, and other communications devices. Rather, each machine utilizing NFS runs its own set of the NFS daemons. (A *daemon* — sometimes known as a *server*<sup>2</sup> — is a program which runs on your machine, doing various specific housekeeping chores. A printer daemon might handle the queueing up and printing of files, while a mail daemon takes care of sending messages back and forth between users. Most daemons are invisible to the user.)

Traditionally, a multi-user system had a single processor (or set of processors) and disks serving a number of users:

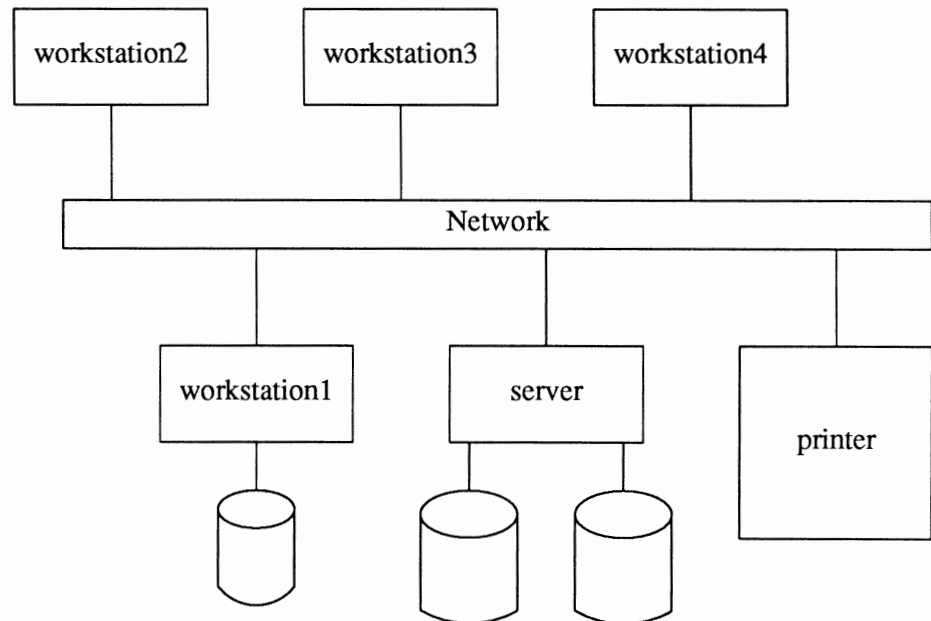
Figure 1-1 *A Non-Network Environment*



The problem with the set-up in Figure 1-1 is that all the users are competing for the same processor. With a network of workstations, however, each user has his or her own central processor, and can access files located on other machines. NFS allows files to be scattered among the various machines on the network, so to the user the whole network appears somewhat like one big computer:

<sup>2</sup> The word *server* has several definitions; see the glossary.



Figure 1-2 *A Typical Network Set-Up*

A *file system* (aka *filesystem*) is the files and subdirectories, with their corresponding files, contained in a directory. For example, everything in `/usr` and its subdirectories makes up the `/usr` file system.

Using NFS, you can get access to other machine's files by *mounting* file systems on those machines. Chapter 3 gives an introduction to mounting remote file systems; for a fuller discussion of NFS, see the *Network Programming* manual. On the Sun386i, mounting is done automatically, so you can skip this section.

## Types of Networks

Some common networks you may encounter include:

### Local Area Networks

As mentioned earlier, small networks are sometimes called Local Area Networks (LANs). In local area networks, machines are connected by cables. Two types of LAN networks common to Sun machines are based on Ethernet and (soon) FDDI technologies; you may, for example, hear references to "sending something over the Ethernet." Such networks allow real-time communication between machines, so you can log in directly to another machine and run programs as though it were in front of you. You can also copy files to and from remote machines on a LAN.

### ARPANET-Based Networks

In 1969 the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense sponsored the development of the a broadscale communication network. Today there are a number of networks which use ARPANET technology; among the best-known and most-accessible is the Defense Data Network, which consists of two networks, the ARPANET and the MILNET. Beyond this is the *Internet*, which is a number of networks connected to the DDN using the *Internet Protocols*.

Users on machines that attach directly to the network, by cables or microwave connections, can log in "real-time" to other machines on the

network.

**Note:** UUCP is an abbreviation for the UNIX to UNIX CoPy program that machines based on the UNIX system, including Suns, employ to “talk” with each other.

#### UUCP Network

The UUCP network is not permanent system of hardware connecting machines, but a program which allows machines to use telephones to transmit data, thus creating a sort of ad hoc network each time the program is invoked. (People disagree whether on whether it should be considered a network at all.) You can use UUCP to communicate with machines across the United States and throughout the world.

An ad hoc UUCP network is not a real-time network like a LAN network; in other words, you can’t log in to another machine using UUCP. However, you *can* send information, like mail messages, to other machines and receive answers back from them.

Because UUCP is not a real-time network, most of the information in this manual, such as logging in to remote machines, copying from other machines, and issuing commands to other machines, does not apply to it. For information about how to send and receive mail messages on the UUCP network, see *Mail and Messages: Beginner’s Guide*.

People using machines on a UUCP network sometimes can send mail messages over ARPANET-based networks by utilizing a *mail gateway*, a machine that transfers data from one network to another.<sup>3</sup>

#### The Yellow Pages

Because networks can get very large, doing system administration on them can get quite complicated: one needs to keep track of who belongs to a network, who has permissions to access which filesystems, and so on. For this reason, Sun provides the *Yellow Pages* (YP). The Yellow Pages is a network database lookup service. That is, it maintains certain files with information about the machines which belong to the network and their users, and it provides users all over the network with the ability to look up the information contained in those files. By default there are a number of databases in the Yellow Pages, although a System Administrator may delete or add databases as desired (except on the Sun386i). These default databases include `hosts`, which has machine names along with their addresses on the network; `passwd`, which, like your local `/etc/passwd` file, contains password and other information associated with usernames; `group`, which holds the various groups that users have divided themselves into; and `aliases`, which contains mail aliases that the network recognizes. Not all systems run the YP — ask your System Administrator if yours does.<sup>4</sup>

The Yellow Pages is a *distributed* database; that means that copies of its data files are kept in various places throughout the network, for faster access by users. (There is one central, master copy, and changes to it are propagated by YP down to the other “slave” copies.) The Yellow Pages lookup service revolves around two daemons: `ypserv`, which is the lookup function on the machine which has

---

<sup>3</sup> For information about how to send and receive mail messages on ARPANET-based networks, see *Mail and Messages: Beginner’s Guide*.

<sup>4</sup> All Sun386i systems run the Yellow Pages.

a copy of the Yellow Pages, and `ypbind`, which lives on your machine and connects it to `ypserv`. In order to use the Yellow Pages database, these two daemons must be running. To find out if your machine is using the Yellow Pages, use the `ps` and `grep` commands; you should see something like this:

```
venus% ps -aux | grep ypbind
root  27  0.0  1.1  66  34 ?  I   0:15 /usr/etc/ypbind
venus%
```

(You can use `ps -aux` to check for the presence of any daemon, including the NFS daemons.)

You may never need to access the YP files; generally only System Administrators ever tinker with them. However, you may occasionally want to read these files. Some commands to help you include the following:

`ypcat`

This displays the YP data file you name. You can pipe it to the `grep` command to find the entries you want.

`ypmatch`

`ypmatch` looks up an entry, or "key," in a the YP database file you specify.

`ypwhich`

This command tells you which YP server supplies Yellow Pages services for your machine.

`yppasswd`

You use `yppasswd` to change your network, as opposed to your machine's, password. (Often they're the same.)

For further information on these commands, type `man` followed by the command's name. For a fuller treatment of the Yellow Pages, type `man ypfiles` or consult the *Network Programming* manual .



---

## Login Access to Other Machines

Login Access to Other Machines .....	11
2.1. Trust Amongst Machines on a Network .....	11
/etc/hosts .....	11
/etc/hosts.equiv .....	12
2.2. Remote Login With <code>rlogin</code> .....	13
Logging In to Another Machine With <code>rlogin</code> .....	13
<code>rlogin</code> to a Machine Where You Don't Have a Home Directory .....	13
<code>rlogin</code> to a Machine As Someone Else .....	14
<code>rlogin</code> to a Non-Existent Machine .....	14
Aborting an <code>rlogin</code> Connection .....	14
Suspending an <code>rlogin</code> Connection .....	15
Verifying Your Identity with <code>who am i</code> .....	16
2.3. Remote Login With <code>telnet</code> .....	16
Logging In to a Machine Running Another Operating System With <code>telnet</code> .....	16
<code>telnet</code> to a Non-Existent Machine .....	17
Aborting a <code>telnet</code> Connection .....	17
Suspending a <code>telnet</code> Connection .....	18



---

## Login Access to Other Machines

You can log in from your machine to another machine using one of these programs:

<code>rlogin</code>	from one machine running an operating system based on the UNIX system (like SunOS) to another such machine
<code>telnet</code>	from one machine to another machine (which may have a different operating system) on a network using the <code>telnet</code> Protocol.

Additionally, Chapter 5 discusses `tip`, a program for establishing connections over phone lines, allowing you to log in to other machines.

Some systems *trust* certain users and certain machines to log in without requiring a password and other security mechanisms.

### 2.1. Trust Amongst Machines on a Network

You can allow or restrict login access by other users to your machine. You, or your system administrator, can set up your machine so that users must supply a password to log in to your machine, or so that certain users from certain machines simply can't log in to your machine.

When another machine running the UNIX system<sup>5</sup> on your local network *trusts* your username on your machine, that machine doesn't require that you type your password or undergo other security checks to log in. The system files `/etc/passwd` and `/etc/hosts.equiv` control the users and the machines that may log in to your machine and to your account.

`/etc/hosts`

Before we describe the files which control access privileges, let's consider how one machine finds another over the network. The `/etc/hosts` file contains a list of machines — also known as *hosts*— and their corresponding addresses on the network. (The file can also contain any optional nicknames for machines.) Your own machine's address is stored here. Programs use `/etc/hosts` to locate other machines for communicating over the network.

However, if you're running the Yellow Pages, the YP file `hosts` may supplant, or supplement, your own `/etc/hosts` file; the YP `hosts` contains the

---

<sup>5</sup> SunOS is an enhanced version of UNIX.

network addresses of every machine on the network.<sup>6</sup> Therefore, you (or your System Administrator) need only maintain an up-to-date `/etc/hosts` file when the Yellow Pages are not available.

Here is a sample `/etc/hosts` file:

Figure 2-1 *An /etc/hosts File*

```
#
# Sun Host Database
#
# If the yellow pages is running, this file is only consulted when booting
#
127.0.0.1      localhost loghost
#
192.9.90.114   gaia
192.9.90.103   venus
```

`/etc/hosts.equiv`

The `/etc/hosts.equiv` file contains a list of machine names that your machine trusts. When a user on one of the machines in the list tries to log in or execute a command on your machine, your machine checks in your `/etc/passwd` file to see whether it should permit access. `/etc/passwd` regulates who can log in on a machine; it contains information about users, including their usernames, their passwords, their ID numbers, and so on.<sup>7</sup>

If that person's username is in `/etc/passwd`, then your machine checks `/etc/hosts.equiv`. If it contains the name of the machine that he or she is using, then your machine allows him or her to log in. If the username isn't there, then he or she is prompted for a password.

Here is a sample `/etc/hosts.equiv` file that permits any users on machines `gaia`, `pluto`, and `verlaine` (who also appear in your `/etc/passwd` file) to log in or execute commands on your machine from their machine.

Figure 2-2 */etc/hosts.equiv File*

Note: You must be `root` to modify this file. See *Doing More with SunOS: Beginner's Guide* for information on becoming `root`.

```
venus% cat /etc/hosts.equiv
gaia
pluto
verlaine
venus%
```

If your system supports the Yellow Pages, you may put a single plus sign (+) all by itself in your `/etc/hosts.equiv` file. This signals that anyone on the network served by the Yellow Pages (who is also named in `/etc/passwd`)

<sup>6</sup> Not every system runs the Yellow Pages; not all that do use the `hosts` scheme.

<sup>7</sup> For more information about the `/etc/passwd` file, see the chapter on processes and other users in *Doing More with SunOS: Beginner's Guide*; see the *SunOS Reference Manual*; or type `man 5 passwd`.



may log in to your machine without providing a password. As a matter of fact, `/etc/passwd` also uses the plus sign convention, so you can thus allow everyone on the network access to your machine. You can also selectively bar some people in the Yellow Pages database from logging in to your machine.

You can further restrict other people's access to your machine with the `.rhosts` file, located in your home directory. For more on `.rhosts` and `/etc/hosts.equiv`, see the *SunOS Reference Manual*, the *System and Network Administration* manual, or type **man hosts.equiv**.

## 2.2. Remote Login With `rlogin`

### Logging In to Another Machine With `rlogin`

`rlogin` logs you in to other UNIX machines on a network.

Type `rlogin` and the *machine name* of the other machine. Should a password prompt appear, type your password for that machine followed by **Return**. If your machine's name is in the other machine's `/etc/hosts.equiv` file, then the other machine trusts your machine name and won't require you to type your password.

Figure 2-3 `rlogin`: *Logging In to Another Machine*

```
venus% rlogin jupiter
Password: (Here you type your password.)
Last login: Mon Oct 20 00:30:52 from venus
SunOS Release 4.0 (SUN) #9: Sat Nov 16 12:51:59 PST 1985
jupiter% pwd
/home/medici
jupiter% logout
Connection closed.
venus%
```

### `rlogin` to a Machine Where You Don't Have a Home Directory

In Figure 2-3 user `medici` logged in to `jupiter` at the directory `/home/medici`, as indicated by the `pwd` command. When you log in to a machine where you don't have a home directory, `rlogin` displays a notification that you have no home directory on that machine, and logs you in to the root directory (`/`) of that machine.

Figure 2-4 `rlogin`: *Logging In to a Machine Without a Home Directory*

```
venus% rlogin neptune
Password:
No directory! Logging in with home=/
Last login: Mon Nov 25 16:58:57 from venus
SunOS Release 4.0 (SUN) #9: Sat Nov 16 12:51:59 PST 1985
neptune% pwd
/
neptune% logout
Connection closed.
venus%
```

### rlogin to a Machine As Someone Else

rlogin as we've described it above allows you to log in to another machine, but under your own username. From time to time you may want to log in as someone else, so that you can fully manipulate files on the remote machine. One example of this would be when you're off working on someone else's machine (and using their username) and you want to log in to your own machine as yourself. The `-l` option to rlogin allows you to do this. The format is as follows:

```
rlogin machine-name -l username
```

For example, here's how user `medici` on machine `venus` logs in on machine `aphrodite` as `cosimo`:

Figure 2-5 rlogin: Logging In to Another Machine With a Different Username

```
venus% rlogin aphrodite -l cosimo
Password:
Last login: Tue Nov 26 00:02:00 from venus
SunOS Release 4.0 (SUN) #9: Sat Nov 16 12:51:59 PST 1985
aphrodite% pwd
/home/cosimo
aphrodite% logout
Connection closed.
venus%
```

Note that when you log in as someone else, you go to that person's home directory.

### rlogin to a Non-Existent Machine

If you attempt to log in to a machine whose name isn't known to your machine, say the machine `andromeda`, rlogin searches unsuccessfully through the hosts database for that machine, then displays the following notification:

Figure 2-6 rlogin: Logging In to a Non-Existent Machine

```
venus% rlogin andromeda
andromeda: unknown host
venus%
```

If you see any error messages that you don't understand, try looking at Table 4-1 in Section 4.4, which contains solutions to common networking problems.

### Aborting an rlogin Connection

**Note:** Usually you abort an rlogin connection only when you can't terminate the connection using `logout` at the end of the work session.

To abort an rlogin connection, type a tilde character followed by a period character (`~.`) at the beginning of a line. The login connection to the other machine aborts, and you find yourself back at your original machine.

When you log in to a series of machines, accessing each machine through another machine, and you use `~.` to abort the connection to any of the machines in the series, you return to the machine where you started.

Figure 2-7 rlogin: *Aborting a Connection*

```

venus% rlogin comet
Last login: Thu Nov 21 05:04:03 from venus
SunOS Release 4.0 (SUN) #9: Sat Nov 16 12:51:59 PST 1985
comet% ~. (Sometimes ~ doesn't echo.)
Closed connection.
venus%

```

To disconnect to an intermediate rlogin, use two tildes (~ ~ .). For example:

Figure 2-8 *Aborting to an Intermediate rlogin*

```

venus% rlogin comet
comet% rlogin jupiter
jupiter% ~ ~. (Sometimes ~ ~ doesn't echo.)
comet%

```

### Suspending an rlogin Connection

When you want to *suspend* an rlogin connection, so that you can return to it later, type the tilde character (~) followed by **Ctrl-Z**. The rlogin connection becomes a *stopped process*, and you are put back into the machine you logged in from. To reactivate the connection, type fg, or % followed by the *job number* of the stopped process (default job number for % is the job you most recently stopped or put in the background).<sup>8</sup>

Figure 2-9 rlogin: *Suspending a Connection*

```

venus% rlogin animation
Last login: Thu Nov 21 07:07:07 from venus
SunOS Release 4.0 (SUN) #9: Sat Nov 16 13:21:24 PST 1985
animation% ~ (Sometimes ~Z doesn't echo on the screen.)

Stopped
venus% pwd
/home/medici
venus% %
rlogin titan (Type Return here to get the command prompt.)

animation% logout
Connection closed.
venus%

```

As is the case with aborting rlogin with ~ ~ ., using *two* tildes and a **Ctrl-Z** will suspend you to an intermediate rlogin.

<sup>8</sup> For more information on running jobs in the background, see the chapter on timesaving features in *Getting Started with SunOS: Beginner's Guide* and the section on job control in the chapter about the C shell in *Doing More with SunOS: Beginner's Guide*.

**Verifying Your Identity with**`who am i`

It can sometimes get difficult to remember just who and where you are. (Not just in logging in to other machines. In life, too.) SunOS probably can't help you much with the latter, but in the case where you have logged in to other machines, and perhaps logged in as someone else, and perhaps done so serially from remote identity to remote identity, we offer the following command:

```
who am i
```

`who am i` displays the machine you're logged into, along with your current identity. (It also gives you the name of your terminal or workstation and the time.) Best of all, you can play pretty fast and loose with the syntax of this command:

Figure 2-10 *Discovering Your True Identity*

```
venus% rlogin tripod -l billyboy
Password:
Last login: Tue Nov 26 00:02:00 from portnoy
SunOS 4.2 Release 4.0 (SUN) #9: Tue Nov 17 21:10:51 PST 1987
tripod% who eats trash
tripod!billyboy      tty3      Nov 20 20:07
tripod% logout
Connection closed.
venus% who killed sis
venus!medici        tty3      Nov 20 20:08
venus%
```

For further information about `rlogin`, see its Man Page, online or in the *SunOS Reference Manual*.

**2.3. Remote Login With**`telnet`

Because you can log in from one UNIX system machine to another such machine with `rlogin`, you need to use `telnet` only when you want to log in to a machine running another operating system.<sup>9</sup>

Therefore, `telnet` is most useful when your machine is on a network with machines that run other operating systems.

**Logging In to a Machine  
Running Another Operating  
System With `telnet`**

For example, to log in to machine `tops20`, running the TOPS20<sup>10</sup> operating system, type `telnet`, followed by its *machine name*. After `telnet` notifies you of the connection with the other machine and identifies your *escape character*, try logging in to the machine as you ordinarily would.

<sup>9</sup> You may have to use `telnet` to log in to machines running varieties of the UNIX operating system that aren't compatible with the current version of the Berkeley UNIX operating system (BSD).

<sup>10</sup> TOPS20 is a trademark of Digital Equipment Corporation.

Figure 2-11 telnet: *Logging In to a Machine Running Another Operating System*

```

venus% telnet tops20
Trying...
Connected to tops20.
Escape character is '^]'.

Yoyodyne Corp., TOPS-20 Monitor 6.1 (6762)-4
@LOG MEDICI

. . .

@LOGOUT
Connection closed by foreign host.
venus%

```

telnet to a Non-Existent Machine

If you attempt to log in to a machine that isn't a part of your network, say the machine `andromeda`, `telnet` searches unsuccessfully through the hosts database for that machine, then displays a notification to that effect and a prompt. Exit from `telnet` by typing `quit`, or the abbreviation `q`.

Figure 2-12 telnet: *Logging In to a Non-Existent Machine*

```

venus% telnet andromeda
andromeda: unknown host
telnet> q
venus%

```

If you see any notifications that you don't understand, look at the `telnet` Man Page, online or in the *SunOS Reference Manual*.

### Aborting a telnet Connection

**Note:** Just as with `rlogin`, one aborts a `telnet` connection only when one can't terminate the connection using `logout` at the end of the work session.

When you want to abort a `telnet` connection, type the `telnet escape character` (usually `[Ctrl-]`), press the `[Ctrl]` key and the close-bracket key simultaneously), followed by `quit` to the `telnet>` prompt. The login connection to the other machine aborts, and you find yourself back at your original machine.<sup>11</sup>

<sup>11</sup> When you log in to a series of machines, accessing each machine through another machine, and you abort the connection to any of the machines in the series, you return to the machine where you started.

Figure 2-13 telnet: *Aborting a Connection*

```

venus% telnet tops20
Trying...
Connected to tops20.
Escape character is '^]'.

Yoyodyne Corp., TOPS-20 Monitor 6.1 (6762)-4
@LOG MEDICI

. . .
@ (Type [Ctrl-] to get telnet> prompt.)
telnet> quit
venus%

```

### Suspending a telnet Connection

When you want to *suspend* a telnet connection, so that you can return to it later, type the standard escape character (usually [Ctrl-]) followed by `z` to the telnet> prompt. The telnet program becomes a *background process*. To reactivate the connection, type `fg`, or `%` followed by the job number of the background process (default job number for `%` is the job you most recently put in the background).<sup>12</sup>

<sup>12</sup> The `z` command only works with the C shell (`csh`). For more information on running jobs in the background, see the chapter on timesaving features in *Getting Started with SunOS: Beginner's Guide*, and the section on job control in the chapter about the C shell in *Doing More with SunOS: Beginner's Guide*.

Figure 2-14 telnet: *Suspending a Connection*

```
venus% telnet tops20
Trying...
Connected to tops20.
Escape character is '^]'.

Yoyodyne Corp., TOPS-20 Monitor 6.1 (6762)-4
@LOG MEDICI

. . .
@          (Type Ctrl-L to get telnet> prompt.)
telnet> z

Stopped
venus% % (Type Return twice to get command prompt of other system.)
telnet tops20

@logout
Connection closed by foreign host.
venus%
```

For further information about `telnet`, see its Man Page, online or in the *SunOS Reference Manual*





---

## Mounting Remote Filesystems

Mounting Remote Filesystems .....	23
3.1. The mount Command .....	23
3.2. The /etc/fstab File .....	24
The umount Command .....	25
3.3. Options to mount and umount .....	26
3.4. The Automounter .....	27
3.5. Who's Mounting This System? showmount .....	29



---

## Mounting Remote Filesystems

You may very often find that you have a need to access programs which are located on another machine. Many times it is not practical, because of space limitations, to copy the files you need over to your machine. For this reason, NFS (described in Chapter 1) offers you the ability to *mount* other filesystems, allowing you to access files on other machines as though they were on your own.

For example, if your company had a lot of software which everyone used, such as spreadsheets, demos, and text editors, a lot of valuable storage space would be taken up if each user had to have his or her own copies of all the programs. People could use `rlogin` to access other machines, but this would be unsatisfactory if they had to do so every time they wanted to run a program. What your company could do, though, would be to put the relevant files on one host (also known as a *server*) and have users mount the filesystem containing those files.

Sun386i users can overlook this material, because, on Sun386i systems, the automounter (described in Section 3.4) automatically mounts required filesystems for the user.

### 3.1. The `mount` Command

Suppose your company's demos are located in `/usr/demos` on a server called `slugfest`. You create (or have your System Administrator create) a directory, `/usr/demos`, on your own system. This `/usr/demos` is called a *mount point*; you then use the `mount` command to create a sort of invisible connection between `/usr/demos` on `slugfest` and your `/usr/demos` mount point. As long as you maintain the mounting, every time you refer to `/usr/demos`, you access the one on `slugfest`. You access `slugfest`'s `/usr/demos` as though it were on your own machine — you can list its contents with `ls`, `cd` in and out of it, view its files with `more`, run its programs, and so on. Mounting allows you to do all of this without having to `rlogin` in to `slugfest`.

Use the `mount` command to mount other filesystems. The general format of the command is

```
mount machine-name:filesystem directory
```

where *machine-name* is the name of the remote machine which contains the filesystem you want to access; *filesystem* is that remote filesystem; and *directory* is a directory on your system. (It is, in fact, the mount point). Here's how you would mount `/usr/demos` and view its contents:

Note: You must be `root` to use `mount`. See *Doing More with SunOS: Beginner's Guide* for information on becoming `root`.

Figure 3-1 *Mounting Remote Filesystems*

```

venus# mount slugfest:/usr/demos /usr/demos
venus# ls /usr/demos
Readme.goban      canvas_demo      globeframes
SRC               cursor_demo      jumpdemo
bouncedemo       framedemo        spheresdemo
venus#

```

You do not have to mount a remote filesystem to a filesystem of the same path-name. In other words, you don't have to mount `/usr/demos` on `slugfest` to a directory called `/usr/demos` on your machine. You can mount it to any directory you want. In fact, since most users aren't allowed to create directories in `/usr`, you could, for example, create a subdirectory `dem` in your home directory and mount `/usr/demos` there (assume `/home/art/medici` is your home directory):

Figure 3-2 *Mounting to A Different Directory*

Use `su` to become `root`.

```

venus% mkdir /home/art/medici/dem
venus% su
Password:      (Type root's password here.)
venus# mount slugfest:/usr/demos /home/art/medici/dem
venus#        (Type Ctrl-D to stop being root)
venus% ls /home/art/medici/dem
Readme.goban      canvas_demo      globeframes
SRC               cursor_demo      jumpdemo
bouncedemo       framedemo        spheresdemo
venus%

```

The important thing is that you make sure that there's a mount point on your system to mount the remote filesystem to. Suppose you try to mount `/usr/demos` to a directory, `/usr/showme`, without creating `/usr/showme` first. You'll get an error like this:

Figure 3-3 *Mounting to a Nonexistent Directory*

```

venus# mount slugfest:/usr/demos /usr/showme
mount: slugfest:/usr/demos on /usr/showme: no such
file or directory
mount: giving up on /usr/showme
venus#

```

### 3.2. The `/etc/fstab` File

Note: you must be `root` to modify your `/etc/fstab` file.

Rather than having to use `mount` for every filesystem you want to mount, you can make use of the file `fstab` which is located in the directory `/etc`.<sup>13</sup>

<sup>13</sup> *fstab* stands for "file system table."

`fstab` is a table of filesystems you want to mount, with any filesystem options you want. Every time your system is booted, it checks `fstab` to see what filesystems to mount. This can save you considerable time over using the `mount` command directly for each filesystem you want mounted.

The general format of an entry in an `fstab` file is

```
machine:filesystem directory type option freq pass
```

where

*machine:filesystem*

is the remote filesystem you want to mount and the machine on which it is located

*directory*

is the directory on your system you want to mount it to

*type*

is the filesystem type, in this case `nfs`

*options*

are the filesystem options, such as those described in Section 3.3

*freq*

is the number of days between dumps

*pass*

partition check pass number

Here's an example of an `/etc/fstab` file:

Figure 3-4 A Sample `fstab` File

```
dirt:/export/root/venus /          nfs rw          0 0
dirt:/usr                /usr          nfs ro          0 0
dirt:/home/dirt          /home/dirt    nfs rw          0 0
moon:/usr/spool/news     /usr/spool/news nfs ro,soft,bg,intr 0 0
anchor:/usr/games        /usr/games    nfs ro,soft,bg,intr 0 0
boodle:/usr/boodle       /usr/boodle   nfs ro,soft,bg,intr 0 0
boodle:/usr/man          /usr/man      nfs ro,soft,bg,intr 0 0
boodle:/usr/tools        /usr/doctools nfs rw,hard,bg,intr 0 0
capulet:/usr/demo        /usr/demo     nfs ro,soft,bg,intr 0 0
olympia:/usr/doc         /usr/doc      nfs ro,hard,bg,intr 0 0
olympia:/usr/olympia     /usr/olympia  nfs rw,hard,bg,intr 0 0
buoy:/usr/buoy           /usr/buoy     nfs rw,hard     0 0
yow:/usr/src             /usr/src      nfs,ro,soft,bg,intr 0 0
```

## The `umount` Command

`umount` *unmounts* file systems which you have previously mounted. `umount` works similarly to `mount`; however, you only need to specify the mount point on your system when using it. You do not need to mention the remote machine it's mounted from. So, for example, to unmount `/usr/demos`, which we mounted in Figure 3-1, you would type as follows:

Figure 3-5 *Unmounting a Filesystem*

```
venus# umount /usr/demos
venus#
```

As with `mount`, you must be `root` to use `umount`.

### 3.3. Options to `mount` and `umount`

There are various ways to use `mount` and `umount`. A few of the more common ways are given here. Please note that you can combine a number of these options; for the exact syntax, consult the *SunOS Reference Manual* or type `man mount`.

(with no options)

`mount` with no arguments displays what you currently have mounted. You do *not* have to be `root` to use the `mount` command this way.

`-p` (“Print.”) `mount` displays a list of mounted filesystems in the format used in `fstab`. You don’t need to be `root` to use `mount` this way, either.

`-v` (“Verbose.”) Displays a message indicating each filesystem in `/etc/fstab` being mounted as it occurs. This option is also available with `umount`.

`-a` (“All.”) `mount` will attempt to mount all the filesystems in `/etc/fstab`.

*filesystem*

or

*directory*

If you use `mount` with either the name of a filesystem to be mounted **or** the name of one of your directories to mount to (but not both), `mount` searches `fstab` for the corresponding entry and mounts the filesystem as indicated therein. Also true for `umount`.

`-h host`

(This is a `umount` option.) `umount` unmounts all the filesystems which are remote-mounted from the server *host*. In Figure 3-4 there are a number of filesystems mounted from the server *boodle*; to unmount them, you would give a command like this:

Figure 3-6 *Unmounting Selected Filesystems*

```
venus# umount -v -h boodle
/usr/man: Unmounted
/usr/boodle: Unmounted
/usr/tools: Unmounted
venus#
```

#### Filesystem Options

`-o options`

There are a number of *filesystem options* to `mount`. These affect the access

you have to the remote filesystem, and the manner that `mount` goes about trying to mount the system. These are some of the more common filesystem options:

`hard or soft`

If you soft-mount a remote filesystem, `mount` will try up to a certain number of times, and then give up, returning you an error message; when you hard-mount it, `mount` will continue trying the mount until it is successful.

There are advantages to both hard- and soft-mounting. Hard-mounting ensures that `mount` will not give up on a server which is only temporarily inaccessible, but will eventually connect you. It is a good idea to always hard-mount filesystems which you write to; that way you lessen the chance of data being lost during a write operation. On the other hand, if you are hard-mounted to a server which crashes, your system may hang up while waiting to connect with the remote server.

`bg or fg`

You can specify whether you want `mount` to retry failed mounting attempts in the *background* or the *foreground*. When a mount occurs in the foreground, the shell is tied up until the mount succeeds; if you run the mount in the background, the shell and you can do other things during the mount attempt.

`rw or ro`

You can mount the other filesystem as *read-write* or *read-only*. It is a good idea to mount other systems as `ro` if you are only going to access their files and programs, not write or change them.

`intr`

If you hard-mount a filesystem, `intr` allows you to interrupt the mount from the keyboard (with `Ctrl-C`).

For information on other filesystem options, see the *SunOS Reference Manual*.

Remember, you can combine various options when using the `mount` command — see the *SunOS Reference Manual*.

### 3.4. The Automounter

Just as the `mount` command lessens the need to rlogin in to another machine every time you want to access its files, the *automounter* lessens the need for hand-mounting remote systems. The automounter mounts remote file systems “on the fly”; that is, it creates a temporary mount point in `/tmp_mnt` for you, mounts the remote system, and unmounts it automatically. It does so without your having to change your `fstab` file. In fact, you do not need to become `root` to dynamically mount and unmount remote filesystems.

The mounting takes place invisibly when you refer to the remote filesystem is a unique way. Thus, you can `cd` to a remote directory, `ls` its contents, `more` a file on another machine, and so on, all without having to mount anything first.

The automounter finds the remote machine and filesystem by using either a file on your machine or a Yellow Pages map. The syntax for using the automounter can vary from site to site, or machine to machine. Here is a general example of how it is used:<sup>14</sup>

Suppose your site uses the Yellow Pages `hosts` and `passwd` databases. These contain the addresses of machines on the network, and the list of users and their home directories, respectively. First, you would start the automounter daemon with a command line similar to this:

Figure 3-7 Starting the Automounter

You must be `root` to *start* the automounter. Many people have the `automount` daemon start up when they boot their system, by modifying their `/etc/rc.local` file. Consult your System Administrator.

```
venus# automount -m /net -hosts /homes -passwd
venus#
```

Now a reference to `/net/machine/filesystem` will cause the automounter to automatically mount `filesystem` from the machine `machine`. For example, in Figure 3-8, the command `ls /net/jupiter/var/spool` causes the automounter to mount `/var/spool` from the machine `jupiter`, and `ls` lists its contents:

Figure 3-8 Using the Automounter: I

```
venus% ls /net/jupiter/var/spool
cron          lw           ppd          uucppublic
ipd           mail        rwho         vpd
lpd           mqueue      secretmail
lpd.lock      news        uucp
venus%
```

Likewise, a reference to `/homes/username` causes the automounter to mount the home directory of the person with the username `username`. Using this syntax, `cat` displays the contents of the file `rhyme.file` in `bcleaver`'s home directory:

Figure 3-9 Using the Automounter: II

```
venus% cat /homes/bcleaver/rhyme.file
A man, he lived by the sewer
And by the sewer he died
And at the coroner's inquest
They called it sewer-side.
venus%
```

<sup>14</sup> Because each site running NFS is unique, consult your System Administrator on how to use the automounter.



If you go a specified period (usually five minutes) without accessing the remote filesystem, `automount` unmounts it for you.

For more information on using the automounter, type `man automount`.

### 3.5. Who's Mounting This System?

`showmount`

The `showmount` command gives the opposite information from typing `mount`. It gives information about who has mounted filesystems from a particular server.

For example, let's assume that user `michael` on the machine `yazoo` has mounted `/usr/games` from the server `slugfest`; `reed` on `zippy` has mounted `/usr/demos` from the same server, and `roger`, on `yow`, has mounted `/usr/tools`.

The `-a` option displays all the remote mounts in the form `hostname:directory`.

Figure 3-10 `showmount -a`

```
venus% showmount -a slugfest
yazoo:/usr/games
yow:/usr/tools
zippy:/usr/demos
venus%
```



---

## Issuing Commands to Remote Machines

Issuing Commands to Remote Machines .....	33
4.1. Status Information: <code>ping</code> , <code>rup</code> , and <code>perfmeter</code> .....	33
Dead or Alive: <code>ping</code> .....	33
Remote Uptime: <code>rup</code> .....	34
Performance Meter: <code>perfmeter</code> .....	34
Remote User Information With <code>rusers</code> .....	35
The <code>finger</code> Command .....	36
4.2. Remote File Copy With <code>rcp</code> and <code>ftp</code> .....	36
Copying from Another Machine to Your Machine Using <code>rcp</code> .....	37
Copying from Your Machine to Another Machine Using <code>rcp</code> .....	37
Copying Directories With <code>rcp</code> .....	38
Copying from Another Machine to Your Machine Using <code>ftp</code> .....	38
Copying from Your Machine to Another Machine Using <code>ftp</code> .....	39
4.3. Remote Command Execution With <code>rsh</code> .....	40
4.4. Network Error Notifications .....	41



---

## Issuing Commands to Remote Machines

Your Sun Workstation supports many programs that execute commands on other machines while you are logged in on your own machine. You can use these programs to:

- Obtain status information (`rup`, `ping`, and `perfmeter`)
- List user information (`rusers` and `finger`)
- Copy files or directories (`rcp` and `ftp`)
- Execute commands (`rsh`)

In addition to describing these functions, this chapter provides examples of error notifications common to network use, and solutions for, or explanations of the problems they represent.

### 4.1. Status Information:

`ping`, `rup`, and  
`perfmeter`

**Dead or Alive:** `ping`

`ping`, `rup`, and `perfmeter` provide *status information* for other machines.

`ping` checks to see if another machine is up and running or not. `ping` is useful because it's a quick way to determine the status of another machine before you try to access it, or if you're having trouble accessing it. The `ping` command takes the form

```
ping host timeout
```

where *host* is the machine you're testing. *timeout* is an optional amount of time you want `ping` to try for. If you don't include *timeout*, `ping` will try for 20 seconds, or until it gets an answer.

Figure 4-1 *Using ping*

```
venus% ping pong
pong is alive
venus%
```

**Remote Uptime:** `rup`

To find out the length of time a system has been “up and running,” and to view its load average, type `rup` followed by the *machine name* of the desired machine.

Figure 4-2 `rup: Uptime and Load Average for a Specific Machine`

**Note:** `rup` stands for **remote uptime**, a version of the `uptime` command that runs `uptime` on other machines.

```
venus% rup pluto
      pluto  up 1 day,  5:47,   load average: 3.64, 3.53, 2.80
venus%
```

To get the same set of information for all of the machines that are “up and running” and on your local network, type `rup` without any arguments:

Figure 4-3 `rup: Uptime and Load Average for the Machines on Your Ethernet`

```
venus% rup
mercury  up 1 day, 23:31,   load average: 0.04, 0.00, 0.00
venus    up 8 days, 21:27,   load average: 0.26, 0.13, 0.02
earth    up 3 days, 20:44,   load average: 1.07, 0.90, 0.54
mars     up          36 mins, load average: 0.01, 0.00, 0.00
jupiter  up          20:30,   load average: 0.52, 0.20, 0.05
saturn   up 8 days, 18:29,   load average: 1.27, 1.12, 1.00
neptune  up 2 days,  2:38,   load average: 2.27, 1.73, 1.18
uranus   up 1 day,  3:19,   load average: 0.64, 0.90, 1.04
pluto    up 1 day,  5:47,   load average: 3.64, 3.53, 2.80
venus%
```

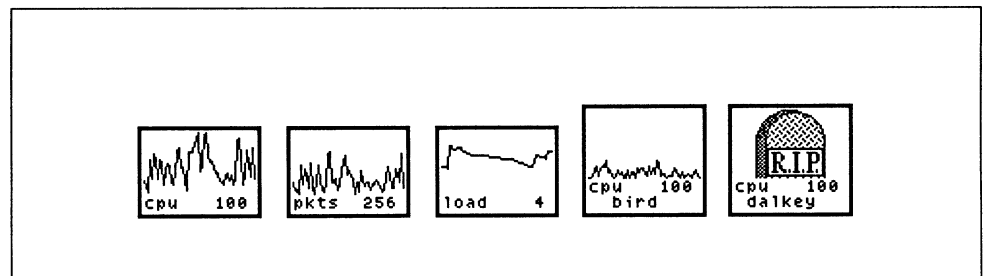
To find out more, see the `rup` Man Page, online or in the *SunOS Reference Manual*.

**Performance Meter:**

`perfmeter`

**Note:** You can only use `perfmeter` when running *SunView*, Sun’s window system.

`perfmeter` gives a visual readout of the status of a machine — yours or someone else’s. It can tell you how busy someone is; for example, if a machine seems to be inactive, its owner may be away from his or her desk. Here’s how a user might run `perfmeter`:

Figure 4-4 `perfmeter`

In Figure 4-4, the user has five versions of `perfmeter` running. The three on the left give the user information about his or her own machine: the one labeled `cpu` shows how much of the machine’s central processor is being used; `pkts`

indicates how much information is going over the network; and *load* shows the average number of runnable processes in the last minute. The *perfmeter* display second from right show the central processor activity for the remote machine *bird*, while the little tombstone figure on the far right shows that machine *dalkey* is not running. *perfmeter* displays other useful information; to find out more about it, see the *SunView 1 Beginner's Guide*, and the *perfmeter* Man Page, online or in the *SunOS Reference Manual*.

### Remote User Information With `rusers`

`rusers` tells you who's logged on to other machines on your network. The command `rusers` by itself shows each machine on the network, followed by the users on that machine:

Figure 4-5 `rusers`

```
venus% rusers
homeplate      billyb
absalom        john
babylon        manchild
garden         russell      cousy      havilcek
texas          houston
venus%
```

**Note:** `rusers` stands for remote users, a version of the `users` command that runs `users` on other machines.

Note in Figure 4-5 that machine *garden* has three different users on it.

If you wanted information on just one machine, you would type `rusers` followed by the machine's name:

Figure 4-6 `rusers` for a Specific Machine

```
venus% rusers garden
garden      russell      cousy      havilcek
venus%
```

For more detailed information, there's the `-l` option, which gives the username, the machine and terminal names, the time the user logged on, how long the users's been idle (if more than one minute), and the name of a machine that the user rlogged in from, if any:

Figure 4-7 `rusers: Detailed Information`

```
venus% rusers -l garden
russell  garden:ttyd8      Nov 18 09:00      7:24
cousy    garden:console     Nov 18 16:13
havilcek garden:ttyp0       Nov 18 11:43      21 (quirk)
venus%
```

The `-l` option can also be used when no machine name is given, as in Figure 4-5. For more information, see the `rusers` Man Page, online or in the *SunOS Reference Manual*.

## The `finger` Command

`rusers` is a good way of getting some information about who's logged in on other machines. The `finger` command, though, *really* gives you the lowdown on other users. In fact, unlike commands such as `ping` or `rusers`, `finger` doesn't give you information about other machines. It tells you only about other users. In fact, `finger` is so user-oriented that it accepts people's real names, as well as their usernames, as arguments.

Here's what `finger` tells you:

- the user's login name
- his or her real name
- his or her home directory and login shell
- the last time he or she logged in
- the last time he or she received mail, and the last time he or she read it
- the name of his or her terminal(s), and how long it's been idle

Here is a slightly simplified example of a two typical `finger` requests. Your output may vary somewhat.

Figure 4-8 `finger`

```
venus% finger moby@sea
[sea]
Login name: moby                In real life: Ishmael Wong
Directory: /home/shipwreck/moby  Shell: /bin/csh
On since Nov 14 06:33:41 on console  4 days 14 hours Idle Time
New mail received Wed Nov 18 20:34:02 1987;
unread since Wed Nov 18 16:20:24 1987
venus% finger Henry Stamper
Login name: hank                In real life: Henry Stamper, Jr
Directory: /home/oregon/hank    Shell: /bin/csh
Last login Wed Oct 21 16:16 on tty0 from cairo
No unread mail
```

`finger` is useful for making sure that the user you're looking for is still alive and kicking (computer-wise, anyway). `finger` has various options and further capacities; to learn about them, see the *SunOS Reference Manual*. or type `man finger`.

## 4.2. Remote File Copy With `rnp` and `ftp`

Note: If you have a filesystem on another machine *mounted*, you do not have to use remote copy commands to put and get files in that filesystem. See Chapter 3.

To copy files from another machine onto your machine, or copy files from your machine onto another machine, use one of the *remote file copy* commands. Each program has a slightly different specialty for file copying:

- `rnp` Most common, but only between UNIX system-based machines ; permits copying of entire directories
- `ftp` uses a standard, generalized protocol



### Copying from Another Machine to Your Machine Using `rcp`

To remote copy a file from another machine onto your machine with `rcp`, use the following syntax:

```
rcp  machinename:file directory
```

where *machinename* is the name of the machine you want to copy from; *file* is the file you want to copy; and *directory* is where you want to put the file on your system.

For example, to copy a file called `/home/charon/new.toy` from the machine called `pluto` to the directory called `/home/medici/toys` on your machine, `venus`:

Figure 4-9 `rcp`: Copying from Another Machine to Your Machine

```
venus% rcp pluto:/home/charon/new.toy /home/medici/toys
venus%
```

You can use normal SunOS shorthand for directories (such as `~` for your home directory and `.` for the current directory).<sup>15</sup> When you want to call the file by a different name on your own machine, specify a destination *filename* at the end of the destination directory on your machine. For example, here's how you could copy the file `new.toy` from machine `pluto` to your home directory and rename it `my.toy`:

Figure 4-10 Copying and Renaming with `rcp`

```
venus% rcp pluto:/home/charon/new.toy ~/my.toy
venus%
```

If `rcp` displays an error notification, see the section on error notifications (Section 4.4).

### Copying from Your Machine to Another Machine Using `rcp`

To `rcp` a file from your machine onto another machine, you reverse the syntax above:

```
rcp  file machinename:directory
```

where *file* is the file on your machine you want to copy; *machinename* is the machine you want to copy to; and *directory* is the place you want to send the file to.

For example, to copy a file called `/home/medici/old.toy` from your machine to the directory called `/home/charon/trash` on the machine

<sup>15</sup> For a description of these naming conventions, see the *Doing More with SunOS: Beginner's Guide*.

```
pluto :
```

Figure 4-11 `rcp`: Copying from Your Machine to Another Machine

```
venus% rcp /home/medici/old.toy pluto:/home/charon/trash
venus%
```

When you want to call the file by a different name on the other machine, specify a destination file at the end of the destination directory on that machine, as in Figure 4-10.

If `rcp` displays an error notification, see the section on error notifications (Section 4.4).

### Copying Directories With

`rcp`

Warning: copying directories with `rcp` doesn't preserve permission and ownership settings.

To copy a directory and its contents from another machine to your machine, or *vice versa*, use `rcp` with the `-r` option. Then, follow the steps for copying files, only replace the filenames with the appropriate directory names.

To copy a directory and its contents from another machine to your machine, type:

```
rcp -r machine-name:directory local-directory
```

(Where *machine-name* is the name of the remote machine; *directory* is the directory on that machine that you want to copy; and *local-directory* is the directory on your machine you want to copy to.)

To copy a directory and its contents from your machine to another machine, type:

```
rcp -r local-directory machine-name:directory
```

(Where *local-directory* is the directory on your machine you want to copy and *directory* is the place on the other machine you want to copy to.)

Like `rlogin`, `rcp` uses the other machine's `/etc/hosts.equiv` and `/etc/passwd` files to determine if you have unchallenged access.

### Copying from Another Machine to Your Machine Using `ftp`

To copy a file from a machine that isn't necessarily running a version of UNIX operating system, use `ftp`, the file transfer program. Here are the steps for copying a file from another machine to your machine using `ftp`:

- Type `ftp` to start the program.
- When you see the `ftp>` prompt, type `open` followed by the *machine name* of the other machine.
- Wait for `ftp` to establish a connection.
- Type your username *on the other machine* to the `Name` prompt.
- Type your password *on the other machine* to the `Password` prompt.

- Wait for `ftp` to log you in to the other machine.
- When you see the `ftp>` prompt, type `get`, followed by the *filename* (use absolute pathname) of the file on the other machine, and the destination *filename* (absolute pathname) where you want the file to appear.
- When `ftp` notifies you that the transfer is complete, type `quit` to close the network connection.

Figure 4-12 `ftp`: Copying from Another Machine to Your Machine

```

venus% ftp
ftp> open pluto
Connected to pluto.
220 titan FTP server (Version 4.125 Sat Nov 2 13:01:21 PST 1985) ready
Name (pluto:medici): medici
Password (pluto:medici): (Type password followed by Return.)
331 Password required for medici.
230 User medici logged in.

ftp> get /home/medici/new.toy /home/medici/toys
200 PORT command okay.
150 Opening data connection for /home/medici/new.toy . . . (176 bytes)
226 Transfer complete.
183 bytes received in 0.01 seconds (9.4 Kbytes/s)

ftp> quit
venus%

```

### Copying from Your Machine to Another Machine Using `ftp`

Here are the steps for copying a file from your machine to another machine using `ftp`:

- Type `ftp` to start the program.
- When you see the `ftp>` prompt, type `open` followed by the *machine name* of the other machine.
- Wait for `ftp` to establish a connection.
- Type your username *on the other machine* to the `Name` prompt.
- Type your password *on the other machine* to the `Password` prompt.
- Wait for `ftp` to log you in to the other machine.
- When you see the `ftp>` prompt, type `send`, followed by the *filename* (use absolute pathname) of the file on your machine, and the destination *filename* (absolute pathname) where you want the file to appear on the other machine.
- When `ftp` notifies you that the transfer is complete, type `quit` to close the network connection.

Figure 4-13 *ftp: Copying from Your Machine to Another Machine*

```

venus% ftp
ftp> open pluto
Connected to pluto.
220 titan FTP server (Version 4.125 Sat Nov 2 13:01:21 PST 1985) ready.
Name (pluto:medici): medici
Password (pluto:medici): (Type password followed by Return.)
331 Password required for medici.
230 User medici logged in.

ftp> send /home/medici/old.toy /home/medici/trash
200 PORT command okay.
150 Opening data connection for /home/medici/old.toy . . . (176 bytes)
226 Transfer complete.
183 bytes received in 0.01 seconds (9.4 Kbytes/s)

ftp> quit
venus%

```

quit to close the network connection.

#### Further Information

For more information on `rcp` or `ftp`, see their Man Pages, online or in the *SunOS Reference Manual*.

### 4.3. Remote Command Execution With `rsh`

**Note:** `rsh` stands for remote shell, or an interpreter capable of executing commands on another machine.

`rsh` allows you to execute a single command on another machine without having to formally log in. It can save time when you know you only want to do one thing on the remote machine.

To execute a command on another machine, type `rsh` followed by the machine's name and the command. For example, suppose you want to see the contents of the directory `/home/fresno/crops` on the machine `fresno`:

Figure 4-14 *Executing a Command on Another Machine*

```

venus% rsh fresno ls /home/fresno/crops
corn          kiwi          oranges
grapes        olives        peaches
venus%

```

When you execute a command on another machines using `rsh`, `rsh` doesn't log in; it talks to a daemon that executes the command on the other machine. However, if you have a `.cshrc` file in your home directory *on the other machine*, `rsh` reads it. So `rsh` uses any pertinent aliases that you have defined on the other machine when executing the command.

Like `rlogin` and `rcp`, `rsh` uses the other machine's `/etc/hosts.equiv` and `/etc/passwd` files to determine whether you have unchallenged access privileges.

For more information, see the `rsh` Man Page, online or in the *SunOS Reference Manual*.

#### **4.4. Network Error Notifications**

This section provides a table of network error notifications and possible solutions for, or explanations of the problems associated with each error notification.

Table 4-1 *Network Error Notifications: Problems and Solutions*

<i>Error Notification</i>	<i>Problem</i>
Address already in use.	Other machine is down or not functional
Connection refused.	The other machine is up, but its daemons aren't ready to complete a connection.
Connection timed out.	One machine or the other is off, hung (stuck), or down
File not found.	File doesn't exist on other machine.
Host name for your address unknown.	Other machine needs your machine name in its /etc/hosts file
Login incorrect.	You mistyped a password, or your username isn't in the other machine's /etc/passwd
Network is unreachable.	<i>Routing problem</i> — a gateway machine or other network connection is broken
...No such file or directory	File or directory on other machine doesn't exist, or you don't have read permission.
... not found.	Directory on other machine doesn't exist
<i>Nothing at all</i>	Directory on other machine is empty
Permission denied.	Machine name isn't in /etc/hosts.equiv file; password required; may need write permission for directory on your machine
RPC_PMAP_FAILURE	Daemon not running properly
RPC_TIMED_OUT	Couldn't connect to other machine, it may be down
RPC_UNKNOWNHOST	Machine name doesn't exist on network
unknown host	Other machine name not in /etc/hosts file on your machine.

---

## Using Dialup Terminals

Using Dialup Terminals .....	45
5.1. Equipment .....	45
5.2. Considerations .....	45
Phone Number .....	46
Baud Rate .....	46
5.3. The Process .....	46
5.4. Remote Login With <code>tip</code> .....	46
Establishing a Connection With <code>tip</code> .....	47
Problems Establishing a Connection With <code>tip</code> .....	47





---

## Using Dialup Terminals

**Note:** A *terminal* may be a process running on a machine, running without any association to the piece of hardware usually considered a terminal.

This chapter describes how to establish a dialup connection from a terminal, personal computer, or workstation to another machine. Because of the great variety of equipment involved, it is not possible to describe the process in complete detail, so general guidelines appear here. It also gives an introduction to the SunOS program `tip`, for establishing telephone connections between machines.

When you wish to establish a dialup connection from a machine that supports `tip` or `cu` to another machine, see the section on `tip` (Section 5.4) or the Man Pages, online or in the *SunOS Reference Manual*.

### 5.1. Equipment

You need two pieces of equipment to initiate a dialup connection:

- a terminal, workstation, or personal computer
- a modem (and telephone lines!)

A hardware terminal is a keyboard and screen device that has no real CPU, or central processing unit, so for the most part it doesn't support general computation or programs. However, to establish a dialup connection to a machine, the computation ability available on a workstation or personal computer isn't necessary.

The modem is a device that permits you to transmit computer signals from your terminal, workstation, or personal computer through a phone line to another such device that translates the signals so they are again accessible by a user at the destination machine.

### 5.2. Considerations

Many considerations may affect your attempt to connect to a machine by dialup access, such as:

- Interpretation of `(Return)` and `(Line Feed)`
- Xon/Xoff protocol
- Parity (odd, even, or no parity)
- 7-bit transmission versus 8-bit transmission

Besides pointing out that the destination machine must have a modem to interpret the signals transmitted by your modem, this manual attempts to explain only a few details about two considerations that are especially important: the *phone number* and the *baud rate*.

**Phone Number**

To dial up the other machine, you have to know its phone number — ask someone you know or contact that machine's system administrator. Of course, some machines don't support dialup connections.

**Baud Rate**

The *baud rate* is the rate at which your terminal, workstation, or personal computer exchanges data (usually bits per second) with the machine you're accessing. The two most common baud rates for dialup connections are 300 baud and 1200 baud. Both sides of the connection must transmit and receive at the same baud rate or they'll get "confused."<sup>16</sup>

Some machines set or adjust baud rates automatically. But if gobbledegook appears on your screen when you establish a connection to the other machine, it probably means that you need to adjust your baud rates for sending and receiving data. See the terminal, workstation, or personal computer manual to learn how to do so.

**5.3. The Process**

To dial up from your terminal, workstation, or personal computer to a machine, follow these steps:

- Power on your terminal, workstation, or personal computer and your modem
- Dial up the other machine by typing the appropriate commands from your keyboard to your modem (read the instruction manual for your modem!)
- Connect to the other machine (sometimes you have to type `Return` once or twice to establish the connection)
- Log in to the other machine
- Complete your work session
- Log out from the other machine
- Disconnect from the other machine (sometimes you have to type an *escape character* to regain the attention of the modem)
- Power off your terminal, workstation, or personal computer and the modem (unless you use them continuously).

**5.4. Remote Login With**

<sup>tip</sup>  
**Note:** `cu`, an interface to `tip` based on what was originally a UNIX Version 7 command, is available. See its Man Page if you want to know more about it.

You can use `tip` to log in to machines that aren't on your local network, but are accessible by dialing up. `tip`, the terminal interface processor, lets you dial up other machines and establish a modem connection.

---

<sup>16</sup> A bit of trivia: the word *baud* originates from the name Baudot, the person who invented the Baudot Code.

### Establishing a Connection With `tip`

To establish a `tip` connection, type `tip` followed by the phone number of the machine to which you want to connect.

Figure 5-1 `tip`: *Establishing a Connection*

```
venus% tip 555-0000
dialing...connected (Type Return twice.)

Yoyodyne Corp., TOPS-20 Monitor 6.1 (6762)-4
@LOG MEDICI

. . .

@LOGOUT
Connection closed by foreign host.
venus%
```

### Problems Establishing a Connection With `tip`

`tip` may fail to make a connection for one of several reasons. When the phone line is busy or doesn't answer, `tip` prints the message `call failed`. Check the phone number and try again. If the problem persists, try looking at the `tip` Man Page, online or in the *SunOS Reference Manual*.

Figure 5-2 `tip`: *Call Fails*

```
venus% tip 555-9999
dialing...call failed
[EOT]
venus%
```

Sometimes the system at the other end of the phone line may not respond properly. Perhaps it is down temporarily or has some kind of maintenance problem. Contact a user on the other system, or its system administrator, to correct the problem.

When the local machine is loaded with many processes, `tip` may not be able to synchronize with a modem to connect with the other machine. Try again, and if that doesn't work, wait until the system is less loaded.

Figure 5-3 `tip`: *Synchronization Problem*

```
venus% tip 555-0101
can't synchronize with modem
call failed
[EOT]
venus%
```



# A

---

## Command Summary

Command Summary .....	51
-----------------------	----



---

## Command Summary

This is a summary of all commands mentioned in this manual. Each command appears in alphabetical order by name, and includes a syntax diagram, and a brief paragraph describing its function.

- `finger`    `finger name`  
 Gives information about the user named by *name*. *name* can be a person's username or real name.
- `ftp`        `ftp`  
 File transfer program that includes commands:
- `get`        `get filename local-filename`  
                   Transfer file *filename* from the machine to which a connection is open to this machine and call it *local-filename*.
- `open`       `open machine-name`  
                   Establish a connection to *machine-name*.
- `quit`       `quit`  
                   Quit ftp.
- `send`       `send local-filename filename`  
                   Transfer file *local-filename* from this machine to become file *filename* on the machine to which a connection is open.
- `mount`       `mount`  
 Display all mounted filesystems.
- `mount machine-name:filesystem directory`  
                   Mount the specified filesystem *filesystem* on machine *machine-name* to directory *directory*.
- `mount options`
- `-a`        Mount all filesystems in `/etc/fstab`.
- `-p`        Print all mounted filesystems in a format useful to `/etc/fstab`.

- v            Display mounts as they occur.
- o*option*    Mount the filesystem according to *option*.  
Valid options include *bg/fg* (background vs. foreground); *ro/rw* (read-only vs. read-write); *hard/soft* (retry on error or not).
- mount [*filesystem* | *directory*]  
Mount the remote filesystem *filesystem* to *directory* as indicated in */etc/fstab*.
- perfmeter  
perfmeter [*machine-name*]  
Provides status information window for this machine or the machine specified by *machine-name*. Runs under SunView.
- ping        ping *machine-name*  
Tells whether another machine is running or not.
- rcp        rcp *local-filename other-machine-name : destination-filename*  
Copy file from *local-filename* on your machine to the *destination-filename* on machine *other-machine-name* (use absolute pathnames for files).
- rcp *other-machine-name : filename local-filename*  
Copy file *filename* from machine *other-machine-name* to *local-filename* (use absolute pathnames for files).
- rcp -r *local-directory-name other-machine-name : destination-directory-name*  
With -r option, copy from *local-directory-name* on your machine to the *destination-directory-name* on machine *other-machine-name*.
- rcp -r *other-machine-name : directory-name local-directory-name*  
With -r option, copy directory *directory-name* from machine *other-machine-name* to *local-directory-name*.
- rlogin     rlogin *machine-name* [-l *username*]  
Log in from your machine to the machine called *machine-name* using the current username; when you specify the -l *username* option, rlogin attempts to log in with the *username* you specify. "~ ." aborts an rlogin connection. ~ followed by **Ctrl-Z** suspends an rlogin connection.
- rsh        rsh *machine-name command*  
Execute command *command* on *machine-name*.
- rup        rup [*machine-name*]  
Display system uptime and load average information for all machines on local Ethernet, or for machine specified.



`rusers` `rusers [machine-name]`  
Display usernames of users logged in on all machines of the local Ethernet, or for *machine-name* if specified.

`showmount`  
`showmount machine-name`  
Displays which other machines have mounted filesystems on the machine *machine-name*.  
  
With the `-a` option, gives the filesystems on *machine-name* which have been mounted on other machines.

`telnet` `telnet [machine-name]`  
File transfer program that includes commands:

`z` `z`  
Suspend telnet program.

`quit` `quit`  
Quit telnet.  
`Ctrl-]` obtains telnet prompt.

`tip` `tip phone-number`  
Dial up *phone-number* and attempt to start a login connection.

`umount filesystem`  
Unmount filesystem *filesystem*.  
  
options include:

`-h host` Unmount all filesystems associated with the remote machine *host*.  
`-a` Unmount all filesystems in `/etc/fstab`.  
`-v` Report on each unmount as it occurs.

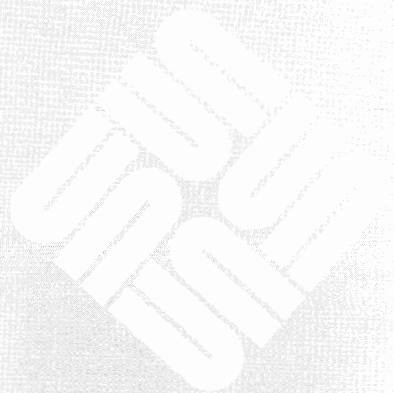


# B

---

## Glossary

Glossary .....	57
----------------	----





# B

---

## Glossary

This glossary lists network facility terms in common use, especially in this manual. For commands, see the command summary, Appendix A .

### **absolute pathname**

For a file or directory, the list of directories from the root directory through the tree structure to the desired filename or directory name, each name in the series separated by a forward slash character (/).

### **background process**

A command that you have directed the system to work on while you continue to type commands to the command interpreter.

### **daemon**

A process that runs in the background, handling commands delivered for remote command execution.

### **dialup connection**

A connection between machines through a phone line.

### **distributed file system**

A file system that exists on more than one machine, permitting easy user access to files on other machines.

### **establish a connection**

Initiate link for transfer of data between two machines.

### **gateway**

A device that transfers data from one network to another — a link between networks.

### **job number**

A number that the system assigns to each process running on that machine.

### **local area network**

A network of machines in close proximity, connected by cables.

### **modem**

A device that enables a machine or terminal to establish a connection and transfer data through telephone lines.

### **mount point**

A directory on your system to which you mount a filesystem which exists on

a remote machine.

**mounting a file system**

Establishing a transparent link, using the Network File System, to a directory on another machine so that the directory appears to be on the local machine, and can be accessed as such.

**network**

A connection between machines, allowing them to transmit information to one another. There are two general kinds of networks: local networks and remote networks.

**Network File System (NFS)**

A set of programs that permits users to directly access files on other machines, and the protocol which allows that access.

**network path**

A series of machine names used to direct mail from one user to another.

**protocol**

A “language” that networks use to communicate with each other — IP (internet protocol), TCP, and UDP are examples.

**read permission**

An attribute of a file that permits you to read and copy the file.

**remote network**

A network other than the local, cable-connected network; it may contain a local network, however.

**remote command execution**

Typed instructions for a command to occur on another machine, and report back any output to the initial machine.

**remote shell**

A command interpreter that you initiate on one machine, but that executes on another machine specified on the command line.

**routing**

The process of determining a pathway for data to get from one machine in a network to another machine through a gateway.

**server**

1) A dedicated machine which provides files and network facilities to client machines; 2) Any machine which is providing files for a client machine (such a machine can be both a server and a client); 3) a daemon which actually handles the providing of files.

**status information**

Information about the system, such as the system load averages, and how long since the last system downtime.

**suspend**

Stop a program so that one can type commands to the interpreter and later return to execution of the program.

**terminal**

A process running on a machine that originates with the physical device called a terminal, or as a software representation of such a physical device, like a window.

**trust**

When a machine permits other machines and usernames access to accounts on that system without requiring a password or other security mechanisms.

**user information**

Information about users on a machine, such as usernames, time since logging in, and “terminal” number.

**window and mouse program**

A program that makes use of SunView, the window system and associated software, so that you can use the mouse to locate and select items within a window application on your screen.

**window-based tool**

See **window and mouse program**.

**write permission**

Access to a file that permits you to change the file.

**Yellow Pages**

A network database which includes information on users and machines on the network; also a service for looking things up in that database.

|

|

|



---

# Index

## A

Address already in use, 42  
aliases, 40  
ARPANET, 6  
automount, 27

## B

background process, 15, 18  
baud rate, 46  
Baudot, 46

## C

C shell, 18  
campus area network, 3  
CAN, 3  
central processing unit, 45  
command  
  cu, 45, 46  
  finger, 36  
  ftp, 38  
  mount, 23  
  perfmeter, 34  
  ping, 33  
  rcp, 37  
  rlogin, 13  
  rsh, 40  
  rup, 34  
  rusers, 35  
  showmount, 29  
  telnet, 16  
  tip, 45, 46  
  umount, 26  
  uptime, 34  
  users, 35  
  w, 35  
  who, 35  
command execution on another machine, 33, 40  
Connection refused, 42  
Connection timed out, 42  
copying directories over network, 38  
copying files over network, 36, 37, 38  
creating a mount point, 23  
  .cshrc file, 40  
cu command, 45, 46

## D

daemon, definition of, 4, 40  
Defense Data Network, 6  
dialup connection, 45, 46  
  modem, 45  
  personal computer, 45  
  terminal, 45  
  workstation, 45  
directory copy, 38

## E

error notification  
  Address already in use., 42  
  Connection refused, 42  
  Connection timed out, 42  
  Host name for your address unknown, 42  
  Login incorrect, 42  
  Network is unreachable, 42  
  Permission denied, 42  
  RPC\_TIMED\_OUT, 42  
  unknown host, 42  
error notifications, 41  
escape character in telnet, 17  
/etc/hosts file, 12  
/etc/hosts.equiv file, 12  
executing commands on another machine, 33, 40

## F

file  
  .cshrc, 40  
  /etc/hosts, 12  
  /etc/hosts.equiv, 12  
file copy over network, 36, 37, 38  
file system, 5  
filesystem, 5  
filesystem options to mount, 27  
finger command, 36  
ftp command, 38, 51

## G

gateway, 5

## H

hard mounting, 27  
host, 12  
Host name for your address unknown, 42

## I

Internet Protocols, 5  
 internetwork, 3  
 internetwork protocol, 5

## J

job number, 15, 18

## L

LAN, 3  
 load average, 34  
 local area network, 3  
 local network, 5  
 login access to other machines, 11  
 Login incorrect, 42

## M

MILNET, 6  
 modem, 45  
 modem connection, 46  
 mount command, 23  
 mount point, 23  
   creating, 23  
 mounted directories, 36  
 mounting remote filesystems, 5, 23  
   filesystem options, 27  
   hard, 27  
   soft, 27

## N

network, 5  
   ARPANET, 6  
   common problems, 41  
   copying directory, 38  
   Defense Data Network, 6  
   definition, 5  
   dialup connection, 46  
   error notifications, 41  
   internetwork protocol, 5  
   local, 5  
   login over, 11  
   MILNET, 6  
   modem connection, 46  
   protocol, 5  
   remote, 5, 6  
   remote login, 11, 13, 16, 46  
   UUCP, 6  
 Network File System, 3, 36  
 Network is unreachable, 42  
 NFS, 3

## P

perfmeter command, 34, 52  
 perfmeter icon, 34  
 Permission denied, 42  
 personal computer, dialup connection to, 45  
 ping command, 33, 52  
 protocol, 5  
   TCP, 36

## R

r`cp` command, 37, 52  
   -r option, 38, 52  
 remote aliases, 40  
 remote command execution, 33, 40  
 remote daemon, 40  
 remote file copy, 36, 37, 38  
 remote load average, 34  
 remote login, 11, 13, 16, 46  
   aborting a connection, 14, 18  
   call failed, 47  
   different operating system, 16  
   different username, 14  
   no connection established, 47  
   no home directory, 13  
   non-existent machine, 14, 17  
   problem, 13, 14, 17, 47  
   suspending a connection, 15, 18  
   synchronization with modem, 47  
   trust, 11  
 remote network, 5, 6  
 remote status information, 33  
 remote system uptime, 34  
 remote user information, 35  
   .rhosts file, 13  
 rlogin command, 13, 52  
   -l option, 14, 52  
 RPC\_TIMED\_OUT, 42  
 rsh command, 40, 52  
 rup command, 34, 52  
 rusers command, 35, 53

## S

server, definition of, 58, 4, 23  
 showmount command, 29  
 soft mounting, 27  
 status information, 33  
 Sun386i, 5, 6  
 system information, 34  
 system uptime, 34

## T

TCP protocol, 36  
 telnet command, 16, 53  
   escape character, 17  
 terminal, dialup connection to, 45  
 tip command, 45, 46, 53  
 trust between machines, 11

## U

umount command, 26  
 unknown host, 42  
 unmounting remote filesystems, 26  
 uptime command, 34  
 user information, 35  
 users command, 35  
 UUCP, 6

## W

w command, 35

WAN, 3

who command, 35

wide area network, 3

workstation, dialup connection to, 45

## Y

Yellow Pages, 6, 12

YP, *see* Yellow Pages

ypbind, 7

ypserv, 7

---

Notes

---

Notes

---

Notes

---

Notes

---

Notes