

# Using UNIX and the Andrew System

This document contains the following sections:

- [What is Andrew?](#)
- [What is UNIX?](#)
- [Using Other Computers from Andrew](#)

---

For information related to this topic refer to:

- [Andrew Files and Directories](#)  
(<http://www.cmu.edu/computing/doc/unix/afs-files.html>)
- [Andrew Linux Support Statement](#)  
(<http://www.cmu.edu/computing/doc/os/linux.html>)
- [Andrew Solaris Support Statement](#)  
(<http://www.cmu.edu/computing/doc/os/solaris.html>)
- [Increasing and Managing Your AFS Quota](#)  
(<http://www.cmu.edu/computing/doc/accounts/quota/index.html>)
- [Logging into and out of AFS](#)  
(<http://www.cmu.edu/computing/doc/unix/afs-login.html>)
- [Project Volumes](#)  
(<http://www.cmu.edu/computing/doc/accounts/project-volume/index.html>)
- [Setting Directory Protections and Using PTS Groups in Andrew](#)  
(<http://www.cmu.edu/computing/doc/unix/pts-groups/index.html>)
- [Your Andrew Unix Account](#)  
(<http://www.cmu.edu/computing/doc/unix/andrew-acct/index.html>)

*Last Updated: 8/24/07*

# What is Unix?

**This document contains the following sections:**

- [UNIX Introduction and the File System](#)
- [Your Home Directory and Pathnames](#)
- [Commonly Used UNIX Commands](#)
- [Protect Your Files](#)
- [Your Andrew Quota](#)

*Last Updated: 8/24/07*

## What is UNIX?

UNIX is an operating system that allows you to store, create, and manipulate large amounts of information on a computer. UNIX is a multi-processing system, which means you can have more than one process running at any given time on one machine.

On campus, the term "UNIX" is also used to refer to the Andrew System UNIX servers. For more information on the UNIX servers, see the following help files, *unix servers and system policy*.

## The UNIX File System

UNIX is structured in terms of files and directories.

A **file** is a collection of data; it can be a paper, a graph, a computer program, a drawing, or anything else you might want to store on a computer.

A **directory** is a special type of file that can contain other files and directories. A directory within a directory is called a subdirectory. The terms directory and subdirectory are often used interchangeably when referring to the UNIX file system (a subdirectory can have files and subdirectories of its own, and so on).

The entire system of directories and files is arranged in a tree-like structure (often called the hierarchical tree), beginning with the root (the directory from which all other directories stem), and extending into branches and leaves. The root of the UNIX tree is at the top instead of the bottom, and all directories grow down from there.

The directory that you are working in is called your current or working directory.

*Last Updated: 8/24/07*

## Your Home Directory and Pathnames

Andrew is based on the UNIX operating system. You have a personal directory in the Andrew system, known as your home directory (your user ID is the name of your home directory). Your home directory is a part of the file system, and it contains subdirectories and files (some created by system administrators, plus any you create yourself).

Whenever you perform such actions as copying a file, renaming a file, removing a file, changing to another directory, or listing your files, you are using UNIX commands.

Whenever you log into Andrew you automatically enter your home directory. This directory was created by the Andrew system administrators and contains a number of default (or automatically present) subdirectories and files. They are:

- OldFiles
- private
- public

### Pathnames

A path name (also called simply path) is like the address of a file or directory. No two files or directories have the same path. A pathname is a chain of directory names, starting at the root and proceeding down the tree until it stops at a particular file or directory. Each directory name is separated by a slash (/).

#### Example 1

An example pathname would be:

```
/afs/andrew.cmu.edu/usr11/juse
```

Working backwards, we see that this pathname refers to the home directory of a user whose ID is "juser." Joe's home directory is a subdirectory of "usr11," which is one of 24 user directories on Andrew to which each user is randomly assigned.

The user directories are all subdirectories of andrew.cmu.edu, which is a directory of all Andrew accounts at Carnegie Mellon University (the .edu indicates to outside users that CMU is an educational institution).

"AFS" is the name of the distributed file system used at Carnegie Mellon, which includes other universities and sites as well.

The slash (/) at the very beginning of the pathname stands for root, the top level directory.

#### Example 2

The full name of the file notes.1 in user Joe's history subdirectory, would be:

```
/afs/andrew.cmu.edu/usr11/juser/history/notes.
```

If you wish to access a file in a subdirectory of your working directory you can simply type the subdirectory and file name. Because of the way the Andrew system and user accounts are set up, Andrew can still recognize this as the full pathname.

#### Example

The path name to a file called notes.1 in your history subdirectory, from your home directory would be:

```
history/notes.1
```

*Last Updated: 8/24/07*

## Commonly Used UNIX Commands

There are a few things you need to remember when entering UNIX commands:

- UNIX commands are typed at a prompt. Very often the prompt is a percent sign **%**. Sometimes it is the name of the machine followed by the **%**. The cursor may be a blinking white square on a black background, a blinking black square on a white background, or a blinking color square on a color background.
- UNIX commands are case sensitive and are usually lower case. This means that **ls** and **LS** are completely different commands. At the end of this section is a chart of some commonly used UNIX commands.
- Spaces are very important. There is always a space between the command and the file or directory it acts upon. In the following example of a command line, **cd** is the command and **/afs/andrew/course/76/271-Summer/dheagy** is the directory.  
Example: **cd /afs/andrew/course/76/271-Summer/juser**
- In order for a UNIX command to be executed, you must press **Enter** at the end of a command line. If all goes well, the prompt and cursor appear on the next line waiting for the next command. It often looks like you haven't done anything, but unless there is an error message, your command has been executed. If there is a problem, you will generate an error message, such as "Command not found." If you generate this type of error, you probably typed something incorrectly. Try again.
- If you type something and you want to get out of it, press **Ctrl + C**.

Command	Action
<b>cat &lt;file&gt;</b>	Print contents of file in the command window
<b>cd &lt;directory&gt;</b>	Change directories
<b>cp &lt;file&gt; &lt;file2&gt;</b>	Copy the contents of file into file2
<b>history</b>	List history of all commands issued at system prompt
<b>ls</b>	List the files and subdirectories in a directory
<b>ls -F</b>	List the difference between files and directories (directories will have a /)
<b>ls -l</b>	List files with status information (list the long way)
<b>ls -lt</b>	List file information in long format, sorted by time with newest files (or newly changed files appearing first)

<b>ls -a</b>	List all the files in a directory, including dot files
<b>mkdir &lt;directory&gt;</b>	Make a directory
<b>mv &lt;file&gt; &lt;file2&gt;</b>	Move file to file 2
<b>pwd</b>	Print the pathname of the current directory
<b>rm &lt;file&gt;</b>	Remove, or delete, files
<b>rmdir &lt;directory&gt;</b>	Remove directory
<b>fs la &lt;directory&gt;</b>	List access rights for <directory> (for more information on access rights, read <a href="http://www.cmu.edu/computing/doc/andrwprotect.pdf">File and Directory Protections (http://www.cmu.edu/computing/doc/andrwprotect.pdf)</a> )

## Moving Around the File System (cd)

To move to your home directory:

1. At the system prompt, type **cd** and press Return.  
After a moment, the system prompt returns, indicating that you have moved to your home directory (if you were already in your home directory, the prompt will still return as normal).  
No matter where you are in the file system, you can use the **cd** (change directory) command to get you back to your home directory immediately. Even if you have not changed directories since you logged in, do the following action as a check:

To move to a subdirectory of your own:

1. Type **cd <path>** and press return.  
**Note:** When you are changing directories down from your current working directory, it is not necessary to type the full pathname.

### Example

To change from your home directory to your public directory, in your home directory, at the system prompt, you would type:

```
cd public
```

To move to someone else's home directory:

1. Type **cd <path>** and press Return.  
In the above command, <path> is the full path to the other users directory.

### Example

To move to Joe User's home directory, you would use the **cd** command and specify Joe's home directory:

```
cd /afs/andrew.cmu.edu/usr11/juser
```

For more information on **cd** see the on-line help for *cd* .

## The Tilde (~)

The **tilde (~)** can be used to abbreviate the pathname to any user's home directory, including your own. It is helpful when you don't know another user's full pathname, or if you just want to save time typing.

### Example

Instead of typing **cd /afs/andrew.cmu.edu/juser** as shown in the previous section, you could use the tilde to take the place of the path to Joe's directory (everything up until the user ID "juser"). The command would look like:

```
cd ~juser
```

**Note:** The tilde can be used with any UNIX command, however, you should never use the tilde in command files such as `.login` or in your preferences file (see the `.login` or preferences help files for more information). In special files such as those, the tilde may not be recognized and can prevent Andrew and UNIX from working properly for you.

## The Dot and Dot Dot

The **dot(.)** stands for your current working directory. You can use it in commands like **cp** or **mv** . For more information, see copying files and moving files.

The **dot dot (..)** stands for the directory above your current working directory. If you are in your sample directory, then the dot dot represents your home directory. It can be used with commands like **cp** , **cd** or **mv**.

## Knowing Where You are in the File System (pwd)

As you move around in the file system, UNIX does not tell you that you have moved to any specific place. You can ask UNIX to tell you where you are by using the **pwd** (print working directory) command. To use this command, at the system prompt:

1. Type **pwd** and press Return:

The pathname of your current working directory is printed out on your screen. You will see something like the following at the system prompt:

```
/afs/andrew.cmu.edu/usr<user#>/<youruserID
```

## Seeing the Contents of a Directory (ls)

To see the names of the files and subdirectories stored in a directory, use the **ls** (list) command. The **ls** command has a number of options, known as switches, which can be added to the command to make it perform a specific action.

To list your file and directory names:

1. Type **ls** at the system prompt and press Return:

A list of the files in your home directory appears. This list does not include dot files (files that begin with a dot). See the section below for more information about what dot files are.

To list your files with status information:

1. Type **ls -l** at the system prompt and press Return:  
A list of the files in your home directory appears. For each file you should see its name, its owner, the date it was last changed, and its size in kilobytes. Those files that are directories have a **d** before their names; those that are plain files have an **-rw-**. This list does not include dot files (files that begin with a dot). See the section below for more information about what dot files are.

To list your file names and see the difference between file and directories:

1. Type **ls -F** at the system prompt and press Return:  
A listing of all the files and directories appears. Directories will have a "/".

To list all your files, including dot files:

1. Type **ls -a** at the system prompt and press Return:  
A listing of all the files in your home directory appears, including dot files.

## Dot files

**Dot files** are used primarily to control system functions. They are called **dot files** because their names begin with a dot (.). Although you can modify some of them if you like, it is not necessary for you to edit or change them in any way. In fact, unless you are following the directions in a help document, or know a great deal about UNIX and how it functions with Andrew, you should not add or delete anything from a dot file.

There is also a file called simply "." and one called "..". These represent your current directory and its parent directory, respectively.

## Create a New Directory (mkdir)

There may be times when you want to create a new directory in which to store information. By placing files with common subjects into their own directories, you can organize your Andrew files.

To create a new directory:

1. Type **mkdir <directoryname>** at the system prompt and press Return.  
After a pause, the system prompt returns, indicating that a directory named sample has been created as a subdirectory of your home directory.
2. Use the **ls** command to verify that the new directory was created.

## Copy Files (cp)

You may want to make a copy of a file, either in the same directory, to another directory, or from someone else's account into yours. You copy a file using the **cp** command. Whenever you copy a file into the same directory, you must give it a new name. When copying to or from another directory, you can change the name if you like, but it is not necessary.

To copy a file in the same directory:

1. Type **cp <file> <file.copy>** at the system prompt and press **Return**.  
After a moment, the system prompt returns, indicating that the file has been copied.

To copy a file into another directory.

1. Type **cp <file> <directory>** and press **Return**.  
After a moment, the system prompt returns, indicating that the file has been copied into the directory specified.  
**Note:** In this case you did not rename the file. This is not necessary because the files, are in separate directories and therefore have different pathnames by which Andrew can distinguish them.

To copy a file from another user's account:

1. Type **cp <path>/<file> <path>/<file>** and press **Return**.  
Where <path>/<file> is the full path and file name of the file you want to copy and <path>/<file> is the path and name of where you would like the file to be copied to.  
After a moment, the system prompt returns, indicating that the files has been copied.

### Example

If you wanted to copy a file called notes from your friend Joe's account into your sample directory and name that file notes.joe, you would type:

```
cp ~juser/notes sample/notes.joe
```

For more information on copying files see the on-line help for *cp*.

## Move Files (mv)

Another way of getting files from one place to another is to move them using the **mv** command. The difference between **mv** and **cp** is that **cp** puts a copy of the file in a new location without disturbing the original copy, but **mv** deletes the file from its old location after saving it in the new location. You can use **mv** anyplace that you would use **cp**.

To move a file a to new file in the same directory (i.e rename a file):

1. Type **mv <file> <file2>** at the system prompt and press **Return**.

### Example

If you want to move a file named notes.joe to a file named notes.working, type:

```
mv notes.joe notes.working
```

**Note:** Using **mv** in this case is simply renaming the file.

To move a file to a new file in a different directory:

1. Type **mv <file> <path>/<file>** and press **Return**.

### Example

If you want to move a file called notes from your home directory into your public directory, you would use the following command while in your home directory, at the system prompt:

```
mv notes public/notes
```

For more information on moving files see the on-line help for *mv* .

## Move Directories

Moving a directory is exactly the same as moving a file. You would use the **mv** command to move or rename a directory.

To move a directory to a new directory in the same directory (i.e rename a directory):

1. Type **mv <directory> <directory2>** at the system prompt and press **Return**.

To move a directory to a new directory in a different directory:

1. Type **mv <directory> <path>/<directory>** and press **Return**.

For more information on moving directories see the on-line help for *mv*.

## Remove Files and Directories

You may want to get rid of a file, or an entire directory. Before you delete anything, make sure you really want to get rid of it permanently, because it is difficult to retrieve deleted files (though it is possible, see the help document [Your Andrew Account \(http://www.cmu.edu/computing/doc/unix/andrew-acct/index.html\)](http://www.cmu.edu/computing/doc/unix/andrew-acct/index.html)).

### Removing a file

To permanently remove a file, you need to use the **rm** command. To remove a file:

1. Type **rm <file>** at the system prompt and press return.

#### Example

Suppose you are finished with the notes.working file and you want to remove it. To do so, you would type:

```
rm notes.working
```

For more information on removing files see the help document on *rm* .

### Removing a Directory

To remove a directory you would use the **rmdir** command. To remove a directory:

1. Use the **cd** command to move up one level from the directory you want to remove.
2. Type **rmdir <directory name>** at the system prompt and press Return.  
This will permanently remove the specified directory.

If you receive an error message telling you the specified directory is not empty, this means you must not have removed all the files from it. To remove a directory that contains files, use the **rm -r** command; see the **rm** help file for more information.

**Example,**

If you have a directory called `sample`, which is a subdirectory of your home directory, you would first change to your home directory by typing:

```
cd
```

and then you would type:

```
rmdir sample
```

and press Return to remove the directory, `sample`.

For more information on removing directories see the help document on *rmdir*.

*Last Updated: 8/24/07*

## Protect Your Files

In UNIX, and on Andrew, it is possible to set protections or access rights to your directories. This gives you the option of limiting the people who can see a particular directory or file. Read the *protection setting* help file for more information on how to do this.

You should realize, however, that despite the ability to set protections, no computer system with multiple users can be completely secure. Never keep anything on-line that is of a highly personal or confidential matter.

*Last Updated: 8/24/07*

## Your Andrew Quota

Your file system quota is the amount of storage space that you have been given on the Andrew System. To see how much space you have been allotted and how much of it you have used up so far:

1. Type **fs lq** at the system prompt and press Return.  
This will list the volume name, the amount of quota you have, the amount used, the percent used, and the percent of the partition used.

If you would like more information on your quota, and what to do if you begin receiving quota error messages, see the help document [Your Andrew Account \(http://www.cmu.edu/computing/doc/unix/andrew-acct/index.html\)](http://www.cmu.edu/computing/doc/unix/andrew-acct/index.html).

*Last Updated: 8/24/07*

## What is Andrew

This document gives a brief introduction to using the **Andrew System** and **UNIX**. The Andrew System, or "Andrew," is a set of computer tools that let you write and edit documents, store files, write your own programs, and other activities.

Andrew consists of two main parts: a user interface and a file system.

### The Andrew User Interface

The Andrew User Interface is the part of Andrew that you see on your screen when you login to an Andrew workstation. It is Andrew's windows, cursors, and menus, as well as many of the programs you can use, such as EZ (an editor), called application programs. Most of these programs were developed using a programming toolkit, a set of tools that make it easier to write Andrew programs.

### CMU's Distributed File System (AFS)

AFS is where your files and directories are stored, along with the files and directories that make Andrew operate.

*Last Updated: 8/24/07*

## Using Other Computers From Andrew

### Running Jobs on Other Andrew Workstations

**rem** allows you to run a job on an Andrew workstation other than the one you are logged into. It uses the **butler** program to locate an available workstation and then sends the command you give it to that workstation to be executed.

To run a command using **rem** , at the system prompt

1. Type **rem -system command** and press Return.  
In the above -system is the system type you would like to run the command on, and command is the command you want to run.

See the help document *Running Programs and Commands* for more information on **rem** and **Butler**.

### Talking to Other Andrew Workstations

**talk** is a visual communication program which copies lines from your terminal to that of another user. It is also available by typing **ucbtalk**.

To start talk:

1. Type **talk userID@hostname** at the system prompt and press Return.
  - If invoked inside an h19 or xterm window, talk will run in that environment. If invoked in a typescript window, then either an h19 or an xterm window will be created (depending on the window manager currently being used) so that **talk** can be run in an environment that provides proper terminal emulation.
  - If you wish to talk to someone on your own machine, then just the person's login name, i.e., userID. If you wish to talk to a user on another host, then you need to use the full host name. If you want to talk to a user who is logged in more than once, the **ttyname** argument may be used to indicate the appropriate terminal name.

When first called, talk sends the message:

```
Message from New Talk_Daemon@their_machine...
talk: connection requested by your_name@your_machine. talk:
respond with: talk your_name@your_machine
```

to the user you wish to talk to.

To reply to a talk request,

1. Type **talk your\_name@your\_machine** at the system prompt and press Return.  
It doesn't matter from which machine the recipient replies, as long as their login-name is the same. Once communication is established, the two parties may type simultaneously, with their output appearing in separate windows.  
See the help on *talk* for more information.

## Running Jobs on Other Systems

To create a secure communication with another host, you'll need to use a secure TELNET client.

To start ssh:

1. Type **ssh <host>** at the system prompt and press Return.  
Where <host> is the name of the system you want to telnet to. The ssh window appears and waits for you to enter a command.  
See the *SSH Secure Shell* (<http://www.cmu.edu/computing/doc/software/ssh/index.html>) or *Terminal* (<http://www.cmu.edu/computing/doc/software/terminal/index.html>) documentation for more information.

## Transferring Files Between Andrew and Other Networks

To transfer files between Andrew and other networks, you'll need to use a secure FTP client. Some of the network sites you may want to reach via FTP allow for anonymous ftp, which means you can login as anonymous. Other sites, however, may require that you have an account (with its associated user ID and password) on that system.

To start a secure file transfer:

1. Type **sftp hostname** at the system prompt and press **Return**:  
where hostname is the name of the computer to or from which you want to transfer files. SFTP will prompt you (in the Typescript window) for your username and password; supply the appropriate information at the prompts.

To transfer files from other computers to Andrew, i.e. use the **get** command. To use the get command:

1. Type **get <filename>** at the system prompt and press Return.

To transfer files from Andrew to other computers use the put command. To use put:

1. Type **put <filename>** at the system prompt and press Return.  
FTP will display messages indicating the status of the transfer. See the help on *ftp* for more information.

## Accessing the Library Information System

To access LIS II from an Andrew workstation and run the Motif user interface:

1. Type **library** at the system prompt and press Return.
  - On a DECstation or Sparcstation that is running X windows, the Motif version of LIS II will start and the Search window will be displayed.
  - On an IBM RT, the VT100 version of LIS II will start.

To access LIS II from an Andrew workstation and run the VT100 user interface:

1. Type **library -v** at the system prompt and press Return:  
The VT100 version of LIS II will start and the Search window will be displayed. This command will work with DECstation, Sparcstation, or IBM RT system types.

Follow the instructions that appear on the screen to use the Library Information System; see also the help on *library* for more details.

*Last Updated: 8/24/07*