**cal**

This command will print a calendar for a specified month and/or year.

To show this month's calendar, enter:

cal

To show a twelve-month calendar for 2008, enter:

cal 2008

To show a calendar for just the month of June 1970, enter:

cal 6 1970

**cat**

This command outputs the contents of a text file. You can use it to read brief files or to concatenate files together.

To append file1 onto the end of file2, enter:

cat file1 >> file2

To view the contents of a file named myfile, enter:

cat myfile

Because cat displays text without pausing, its output may quickly scroll off your screen. Use the [less](https://kb.iu.edu/d/afsk#less) command (described below) or an editor for reading longer text files.

For more, see [In Unix, how do I combine several text files into a single file?](https://kb.iu.edu/d/afas)

**cd**

This command changes your current directory location. By default, your Unix login session begins in your home directory.

To switch to a subdirectory (of the current directory) named myfiles, enter:

cd myfiles

To switch to a directory named /home/dvader/empire\_docs, enter:

cd /home/dvader/empire\_docs

To move to the parent directory of the current directory, enter:

cd ..

To move to the root directory, enter:

cd /

To return to your home directory, enter:

cd

**chmod**

This command changes the permission information associated with a file. Every file (including directories, which Unix treats as files) on a Unix system is stored with records indicating who has permission to read, write, or execute the file, abbreviated as r, w, and x. These permissions are broken down for three categories of user: first, the owner of the file; second, a [group](https://kb.iu.edu/d/aeqw) with which both the user and the file may be associated; and third, all other users. These categories are abbreviated as u for owner (or user), g for group, and o for other.

To allow yourself to execute a file that you own named myfile, enter:

chmod u+x myfile

To allow anyone who has access to the directory in which myfile is stored to read or execute myfile, enter:

chmod o+rx myfile

You can view the permission settings of a file using the [ls](https://kb.iu.edu/d/afsk#ls) command, described below.

**Note:** Be careful with the chmod command. If you tamper with the directory permissions of your home directory, for example, you could lock yourself out or allow others unrestricted access to your account and its contents.

For more, see [Change permissions for a file in Unix](https://kb.iu.edu/d/abdb).

**cp**

This command copies a file, preserving the original and creating an identical copy. If you already have a file with the new name, cp will overwrite and destroy the duplicate. For this reason, it's safest to always add -i after the cp command, to force the system to ask for your approval before it destroys any files. The general syntax for cp is:

cp -i oldfile newfile

To copy a file named meeting1 in the directory /home/dvader/notes to your current directory, enter:

cp -i /home/dvader/notes/meeting1 .

The . (period) indicates the current directory as destination, and the -i ensures that if there is another file named meeting1 in the current directory, you will not overwrite it by accident.

To copy a file named oldfile in the current directory to the new name newfile in the mystuff subdirectory of your home directory, enter:

cp -i oldfile ~/mystuff/newfile

The ~ character (tilde) is interpreted as the path of your home directory.

**Note:** You must have permission to read a file in order to copy it.

**date**

The date command displays the current day, date, time, and year.

To see this information, enter:

date

**df**

This command reports file system disk usage (i.e., the amount of space taken up on mounted file systems). For each mounted file system, df reports the file system device, the number of [blocks](https://kb.iu.edu/d/afru) used, the number of blocks available, and the directory where the file system is mounted.

To find out how much disk space is used on each file system, enter the following command:

df

If the df command is not configured to show blocks in kilobytes by default, you can issue the following command:

df -k

**du**

This command reports disk usage (i.e., the amount of space taken up by a group of files). The du command descends all subdirectories from the directory in which you enter the command, reporting the size of their contents, and finally reporting a total size for all the files it finds.

To find out how much disk space your files take up, switch to your home directory with the [cd](https://kb.iu.edu/d/afsk#cd) command, and enter:

du

The numbers reported are the sizes of the files; on different systems, these sizes will be in units of either 512 byte blocks or kilobytes. To learn which is the case, use the [man](https://kb.iu.edu/d/afsk#man) command, described below. On most systems, du -k will give sizes in kilobytes.

**find**

The find command lists all of the files within a directory and its subdirectories that match a set of conditions. This command is most commonly used to find all of the files that have a certain name.

To find all of the files named myfile.txt in your current directory and all of its subdirectories, enter:

find . -name myfile.txt -print

To look in your current directory and its subdirectories for all of the files that end in the extension .txt, enter:

find . -name "\*.txt" -print

In these examples, the . (period) represents your current directory. It can be replaced by the full pathname of another directory to search. For instance, to search for files named myfile.txt in the directory /home/user/myusername and its subdirectories, enter:

find /home/user/myusername/ -name myfile.txt -print

On some systems, omitting the final / (slash) after the directory name can cause find to fail to return any results.

As a shortcut for searching in your home directory, enter:

find "$HOME/" -name myfile.txt -print

For more, see [Use the Unix find command to search for files](https://kb.iu.edu/d/admm).

**jobs**

This command reports any programs that you suspended and still have running or waiting in the background (if you had pressed Ctrl-z to suspend an editing session, for example). For a list of suspended jobs, enter:

jobs

Each job will be listed with a number; to resume a job, enter % (percent sign) followed by the number of the job. To restart job number two, for example, enter:

%2

This command is only available in the csh, bash, tcsh, and ksh shells.

**kill**

Use this command as a last resort to destroy any jobs or programs that you suspended and are unable to restart. Use the jobs command to see a list of suspended jobs. To kill suspended job number three, for example, enter:

kill %3

Now check the jobs command again. If the job has not been cancelled, harsher measures may be necessary. Enter:

kill -9 %3

**less and more**

Both less and more display the contents of a file one screen at a time, waiting for you to press the Spacebar between screens. This lets you read text without it scrolling quickly off your screen. The less utility is generally more flexible and powerful than more, but more is available on all Unix systems while less may not be.

To read the contents of a file named textfile in the current directory, enter:

less textfile

The less utility is often used for reading the output of other commands. For example, to read the output of the ls command one screen at a time, enter:

ls -la | less

In both examples, you could substitute more for less with similar results. To exit either less or more, press q. To exit less after viewing the file, press q.

**Note:** Do not use less or more with executables (binary files), such as output files produced by compilers. Doing so will display garbage and may lock up your terminal.

**lpr and lp**

These commands print a file on a printer connected to the computer network. The lpr command is used on BSD systems, and the lp command is used in System V. Both commands may be used on the [UITS](https://kb.iu.edu/d/ahaw) systems.

To print a file named myfile on a printer named lp1 with lpr, enter:

lpr -Plp1 myfile

To print the same file to the same printer with lp, enter:

lp -dlp1 myfile

**Note:** Do not print to a printer whose name or location is unfamiliar to you.

**ls**

This command will list the files stored in a directory. To see a brief, multi-column list of the files in the current directory, enter:

ls

To also see "dot" files (configuration files that begin with a period, such as .login), enter:

ls -a

To see the file permissions, owners, and sizes of all files, enter:

ls -la

If the listing is long and scrolls off your screen before you can read it, combine ls with the [less](https://kb.iu.edu/d/afsk#less) utility, for example:

ls -la | less

For more, see [In Unix, how do I list the files in a directory?](https://kb.iu.edu/d/abds)

**man**

This command displays the manual page for a particular command. If you are unsure how to use a command or want to find out all its options, you might want to try using man to view the manual page.

For example, to learn more about the ls command, enter:

man ls

To learn more about man, enter:

man man

If you are not sure of the exact command name, you can use man with the -k option to help you find the command you need. To see one line summaries of each reference page that contains the keyword you specify, enter:

man -k keyword

Replace keyword in the above example with the keyword which you want to reference. Also see [In Unix, what is the man command, and how do I use it to read manual pages?](https://kb.iu.edu/d/afjm)

**mkdir**

This command will make a new subdirectory.

To create a subdirectory named mystuff in the current directory, enter:

mkdir mystuff

To create a subdirectory named morestuff in the existing directory named /tmp, enter:

mkdir /tmp/morestuff

**Note:** To make a subdirectory in a particular directory, you must have permission to write to that directory.

**mv**

This command will move a file. You can use mv not only to change the directory location of a file, but also to rename files. Unlike the cp command, mv will not preserve the original file.

**Note:** As with the cp command, you should always use -i to make sure you do not overwrite an existing file.

To rename a file named oldname in the current directory to the new name newname, enter:

mv -i oldname newname

To move a file named hw1 from a subdirectory named newhw to another subdirectory named oldhw (both subdirectories of the current directory), enter:

mv -i newhw/hw1 oldhw

If, in this last operation, you also wanted to give the file a new name, such as firsthw, you would enter:

mv -i newhw/hw1 oldhw/firsthw

**ps**

The ps command displays information about programs (i.e., processes) that are currently running. Entered without arguments, it lists basic information about interactive processes you own. However, it also has many options for determining what processes to display, as well as the amount of information about each. Like lp and lpr, the options available differ between BSD and System V implementations. For example, to view detailed information about all running processes, in a BSD system, you would use ps with the following arguments:

ps -alxww

To display similar information in System V, use the arguments:

ps -elf

For more information about ps refer to the ps [man](https://kb.iu.edu/d/afsk#man) page on your system. Also see [In Unix, what do the output fields of the ps command mean?](https://kb.iu.edu/d/afnv)

**pwd**

This command reports the current directory path. Enter the command by itself:

pwd

For more, see [In Unix, how do I determine my current working directory?](https://kb.iu.edu/d/abnf)

**rm**

This command will remove (destroy) a file. You should enter this command with the -i option, so that you'll be asked to confirm each file deletion. To remove a file named junk, enter:

rm -i junk

**Note:** Using rm will remove a file permanently, so be sure you really want to delete a file before you use rm.

To remove a non-empty subdirectory, rm accepts the -r option. On most systems this will prompt you to confirm the removal of each file. This behavior can be prevented by adding the -f option. To remove an entire subdirectory named oldstuff and all of its contents, enter:

rm -rf oldstuff

**Note:** Using this command will cause rm to descend into each subdirectory within the specified subdirectory and remove all files without prompting you. Use this command with caution, as it is very easy to accidently delete important files. As a precaution, use the ls command to list the files within the subdirectory you wish to remove. To browse through a subdirectory named oldstuff, enter:

ls -R oldstuff | less

**rmdir**

This command will remove a subdirectory. To remove a subdirectory named oldstuff, enter:

rmdir oldstuff

**Note:** The directory you specify for removal must be empty. To clean it out, switch to the directory and use the ls and rm commands to inspect and delete files.

**set**

This command displays or changes various settings and options associated with your Unix session.

To see the status of all settings, enter the command without options:

set

If the output scrolls off your screen, combine set with [less](https://kb.iu.edu/d/afsk#less):

set | less

The syntax used for changing settings is different for the various kinds of Unix shells; see the [man](https://kb.iu.edu/d/afsk#man) entries for set and the references listed at the end of this document for more information.

**vi**

This command starts the vi text editor. To edit a file named myfile in the current directory, enter:

vi myfile

The vi editor works fairly differently from other text editors. If you have not used it before, you should probably look at a tutorial, such as [Use the vi text editor](https://kb.iu.edu/d/adxz).

The very least you need to know to start using vi is that in order to enter text, you need to switch the program from command mode to insert mode by pressing i. To navigate around the document with the cursor keys, you must switch back to command mode by pressing Esc. To execute any of the following commands, you must switch from command mode to ex mode by pressing : (the colon key): Enter w to save; wq to save and quit; q! to quit without saving.

**w and who**

The w and who commands are similar programs that list all users logged into the computer. If you use w, you also get a list of what they are doing. If you use who, you also get the IP numbers or computer names of the terminals they are using.