

## 1 Basic commands

This section describes a list of commonly used commands that are available on the EECS UNIX systems. Most commands are executed by simply typing the name of the command at the command prompt (*machine\_name%*) with the relevant arguments and hitting return. For the purposes of this document the [ ] notation refers to an argument that the user supplies depending on the situation. Most of the commands that are described in this section are substantially more complicated than the explanation given. See the relevant online manual pages for details (accessing them with the `man` command is covered later in this section).

## 2 Terminology

**directory** A collection of files (a “folder” in Windows or MacOS).

**current directory** The directory that you are currently working in. In many graphical operating systems, a user can “dig” through the directory structure by navigating through a series of windows. The current directory is equivalent to the topmost active window in a graphical operating system that is displaying the contents of a directory.

**home directory** The topmost directory associated with your login account. When you first log in to the system, the home directory is the first directory that is set as your current directory.

**path** A name of a file or directory. In addition to your files, each server stores hundreds of other users’ files, each with its own directory within the storage device. Since directories are usually nested in other directories, the UNIX operating system uses a path, or series of directories, when referring to files. For example, if relative to one’s current directory there exists a directory named “A” and within “A” there existed a directory named “B” then the path would be A/B. Using the `cd` command (as described below), B can be set as the current directory by simply typing

```
cd A/B
```

directory nested above – In UNIX, the directory nested above the current directory is referred to by “..”. Thus, using the `cd` command (as described below), the command

```
cd ..
```

would set the current directory to the directory that is nested above the current directory.

---

\*Adopted from an anonymous document from previous semesters

## 2.1 Commands

There are graphical user interfaces for UNIX, but the traditional (and most flexible) method to communicate with the system is through the command line.

### **pwd**

The **pwd** command displays the path of the current directory starting from the very root level of the hard drive to the current directory's location in its nested hierarchy. Type

```
pwd
```

A path should then appear on the screen.

### **ls**

The **ls** command lists the contents of the current directory or, optionally, the contents of the directory the user chooses to specify. This operation is similar to double-clicking on a Mac or Windows “folder” and viewing its contents.

- To view the contents of the current directory, type

```
ls
```

A listing of files in the current directory should appear.

- To view the contents of a directory that is not the current directory, type

```
ls [ path of directory ]
```

(Note: don't type the brackets, just the path.) A listing of files in the specified directory should appear.

- To get more details about files, add an operand (“switch”) **-l** immediately after the **ls**. Thus,

```
ls -l
```

will produce lines like this:

```
-rw-r--r--    1 cs61b-yu cs61b      1234 Aug 20 14:29 Foo.java
```

which tell you the name of a file, its size (1234 bytes), date and time of last modification, and access rights (you may read and write it, others may read only).

**cd**

The `cd` command changes the current directory to another directory specified or simply sets the current directory to the user's home directory.

- To change the current directory to another directory, type

```
cd [ path of directory ]
```

The current directory should now be changed.

- To change the current directory to your home directory, type

```
cd
```

The current directory should now be your home directory.

**mkdir**

The `mkdir` command creates a new directory. This is similar to selecting “New Folder” from the File menu on Macintosh and Windows systems.

To create a directory inside the current directory, type

```
mkdir [ name of new directory ]
```

A new directory with the name [*name of new directory*] should now exist within the current directory.

**cp**

The `cp` command is used to make a copy of a file. This is similar to highlighting a file on a Macintosh or Windows system, and selecting “Duplicate” from the File menu. To create a copy of a file, type

```
cp [ name of file ] [ name of copy ]
```

**rsync**

Although you can use `cp` with appropriate options to copy a whole directory, we suggest that you use the `rsync` command instead. This command has the added advantage that you can copy from one machine to another (if it has proper software, anyway). This is not an official Unix command, but most installations of Unix and MacOS X have it, and it is available for Windows.

- To copy an entire directory, type one of the commands

```
rsync -a [ path of existing directory ] [ path of new copy ]  
rsync -av [ path of existing directory ] [ path of new copy ]
```

The second form is “verbose,” printing the names of files it copies.

- To copy an entire directory, except for certain files, use `--exclude` options. For example, to copy a directory, except for the hidden `.svn` directories that Subversion creates, type

```
rsync -a --exclude .svn [ path of existing directory ] [ path of new copy ]
```

- To copy an entire directory to or from another machine, prefix the remote directory name with the user name and the host name of the remote computer. For example, if you are on your home UNIX systems and wish to copy a directory from your class account, you might type

```
rsync -av cs61b-yu@quasar.cs.berkeley.edu:[path of existing directory] \  
[ path of new copy ]
```

## **mv**

The `mv` command is used to either rename an existing file or folder to something different or to “move” a file from one directory to another (an operation similar to dragging a file from one directory to another on a Macintosh or Windows system).

- To rename an existing file, type

```
mv [ name of existing file ] [ new name for existing file ]
```

- To move an existing file to a new directory without otherwise renaming it, type

```
mv [ name of existing file ] [ path of new directory ]
```

## **rm**

The `rm` command is used to delete a file. This operation is similar to dragging an item to the Trash (or Recycle Bin) and Emptying it. Files that are “rm’d” cannot easily be recovered (though sometimes the system administrator may have a backup). Use this command with caution.

- To remove an existing file, type

```
rm [ name of file ]
```

- To remove an entire directory and all files in it, type

```
rm -r [ name of directory ]
```

Be careful with this; it can destroy a lot of data! The default shell on the instructional machines prompts you for each file. You can avoid this with

```
/bin/rm -r [ name of directory ]
```

## **rmdir**

The `rmdir` command is a separate command for removing empty directories only. As such, you may find it a bit safer than `rm -r`. To remove an existing directory, type

```
rmdir [ name of directory ]
```

### 3 UNIX online manual

Despite the seemingly user-hostile nature of UNIX, the UNIX system does provide lots of documentation online. As mentioned before, almost all of the commands listed in the previous section are more complicated than the explanation given suggests. There are two situations when you may want to use the online documentation:

- You want to accomplish “something” but you don’t know precisely what the name of the command is.
- You know what the name of a command is but would like to learn more about how to use it. To find more information about how to accomplish “something”, type

```
man -k [ single keyword ] | less
```

This will return a list of possible commands that the student could investigate further. The construct “|less” sends the output of the `man` command to the `less` command, which slows down the display so that you may scroll up and down the display a page at a time. To scroll up type `b`. To scroll down, type the space bar. To highlight a keyword or phrase type

```
/ [ keyword ]
```

- To find more information about a command, type

```
man [ command name ] | less
```

A lengthy description of the command should appear. Use the `less` command to scroll up and down the screen as well as to highlight keywords to improve readability. More general online help files are available in the `/usr/pub` directory on all the instructional UNIX systems.