

Fall 2017

Reading Supplement, 1 September 2017
Tips on Using Linux and the Command Line Interface

Directories, Folders, CLIs, GUIs

A “folder” is the same thing as a “directory”—a place to save files.

You can navigate between folders in one of two ways—by typing commands through the terminal window (this is sometimes called a “CLI,” or “command line interface”) or by the more familiar method of clicking and double-clicking on images of folders and icons representing files (this is an example of a “GUI,” or “graphical user interface”).

In this class we concentrate primarily on the command-line interface, since this material is valuable to people who go on to become “power users”—professional developers, systems administrators, etc., who want to work as efficiently as possible. You are free to use the GUI for your own programming, but instructions will almost always be given in CLI form.

Home Directory vs Desktop

You have a “home directory” that has the same name as your account, e.g., Jane Smith has a home directory named “smithj.” Within this directory is a subdirectory named “Desktop.” When you see a file icon or folder on your screen, that is actually saved in your **Desktop** subdirectory.

When you right-click on the screen and choose “Open in Terminal,” commands typed in your terminal window will be relative to the **Desktop** directory since that’s where you clicked to open the window. *The desktop is not your home directory*—it is one level below it.

It is okay to store things on the **Desktop**. Just be aware that certain commands described in Table 1, such as the command “cd,” will automatically return you to your *home directory* and *not* to the **Desktop**.

Using the Terminal Window and Keyboard

If you would like the terminal window application to appear in your list of applications on the left (the application “launcher”), do the following:

- Right-click on the screen and select the “Open in Terminal” command. The icon for the



terminal application should appear on the left—it looks like this: ***You are not in your home directory—you are in the Desktop folder!*** To get to the home folder, type the command “cd”.

- Right-click on the terminal icon on the left and select “Lock to Launcher”. This will permanently add the terminal window to the launcher.

Remember—the next time you launch the Terminal application from this new icon, it will open in your home directory, not the desktop.

Prompts

Terminal window commands are typed at the *prompt*. The prompt is usually some combination of your username, your computer (e.g., “aldenv100”) and the current folder or directory. For instance, in the text below, the prompt “smithj@aldenv52:~\$” means that user smithj, working at machine aldenv52, is in the home directory. The command “cd cs390” changes to directory cs390, and the new prompt indicates this. Typing the command “cd lab1” changes to the lab1 subdirectory, and the prompt now shows this as well:

```
smithj@aldenv52:~$ cd cs390           [in home directory]
smithj@aldenv52:~/cs390$ cd lab1     [now in cs390 subdirectory]
... etc. ...
```

If you ever see a prompt like “smithj@aldenv100:~/Desktop” then you know that you are in the Desktop subdirectory of your home directory.

Arrow Keys

The “arrow keys” at the lower right of your keyboard enable you to move back through the “history” of commands you have typed. For instance, it is often the case, when developing a program, that you will have to repeatedly type certain commands. There is no need to keep retyping these commands! Using the up-arrow key (“↑”) will let you rapidly move back to an earlier command so all you need to do is press the “Enter” or “Return” key. You can also use the left-arrow and right-arrow keys (“←” and “→”) to move back and forth in a typed command to correct typing errors or to use a modified version of an earlier command.

Special Keys

The “Ctrl” key (bottom row, below the “shift” key) is used in conjunction with other keys for special commands. For instance, the key combination Ctrl-C will “kill” a process that is running in the terminal window. (Why would you ever want to do that? We will see that if you write a program that goes into an “infinite loop,” you must terminate it with a Ctrl-C command.)

The key combination Ctrl-Z will *suspend* a process that is running in the terminal window; it can later be *resumed* in a number of ways. This is occasionally useful as we’ll see later.

Linux Commands

Table 1 shows some common Linux commands. You’ll note that many of them are just two letters (“rm” for “remove,” “cp” for “copy,” etc.).

Table 1: Some Common Linux Commands

Command	Meaning	Example
<code>ls</code>	List files in current directory	smithj@aldenv113:~/cs390/lab1\$ <code>ls</code> Lab1.r Lab1.txt
<code>mkdir name</code>	Make a new directory called <i>name</i> in the current directory	smithj@aldenv113:~/cs390\$ <code>mkdir lab2</code>
<code>cd name</code>	Change to directory <i>name</i> in the current directory	smithj@aldenv113:~/cs390\$ <code>cd lab2</code> smithj@aldenv52 ~/cs390/lab2\$
<code>cd</code>	Change to home directory	smithj@aldenv113:~/cs390\$ <code>cd</code> smithj@aldenv113:~\$
<code>cd ..</code>	Change to directory one level up	smithj@aldenv113:~/cs390/lab1\$ <code>cd ..</code> smithj@aldenv113:~/cs390\$
<code>cp name1 name2</code>	Copy file <i>name1</i> to <i>name2</i>	smithj@aldenv113:~\$ <code>ls</code> Fun.r Fun.txt smithj@aldenv113:~\$ <code>cp Fun.r Lab2.r</code> smithj@aldenv113:~\$ <code>ls</code> Fun.r Fun.txt Lab2.r
<code>mv name1 name2</code>	Rename file <i>name1</i> as <i>name2</i>	smithj@aldenv113:~\$ <code>ls</code> Fun.r Fun.txt smithj@aldenv113:~\$ <code>mv Fun.r Lab2.r</code> smithj@aldenv113:~\$ <code>ls</code> Fun.r Lab2.r
<code>rm name</code>	Remove file <i>name</i>	smithj@aldenv113:~\$ <code>ls</code> Fun.r Lab2.r smithj@aldenv113:~\$ <code>rm Fun.r</code> smithj@aldenv113:~\$ <code>ls</code> Lab2.r
<code>pwd</code>	Shows which directory you are in	smithj@aldenv113:~\$ <code>pwd</code> /home/j/smithj/Desktop
<code>exit</code>	Close the terminal	