Linux Tutorial

Linux Tutorial

This tutorial can be complete as long as you have access to a terminal that accepts Linux commands. If you
are working on Linux or Mac this is already installed on your computer and the application is name "terminal".
If you are on Windows some ways you can get access to it are

- Installing Cygwin and using the Cygwin Terminal (See the Working Locally Page on Canvas)
- Installing WSL

You can also go to the CSIF and complete the tutorial on a computer there.

Common Problems

Below is a list of common problems students have encountered while doing this tutorial and how to solve them

- If you leave a remote connection to the CSIF idle for too long, it will automatically disconnect you. You can fix this by closing the terminal you connected to the CSIF and relogging into it.
- Forgetting to connect to the school VPN before connecting to the CSIF.
  If this happens you will be denied access to the CSIF computer so make sure you are connected to the school VPN first.
- Not realizing that you will **always be placed into your home directory** any time you open a terminal/connect to the
  CSIF
  This means that if you are in the middle of the tutorial and say in the directory `/usr/home/first/second` and you
  turn your computer off for the night, the next time you reopen the tutorial you will be in `/usr/home` and so will
  need to use `cd` to get back to the directory you should be in to continue the tutorial.

If you want to reset the tutorial and do it from scratch

1. Type `cd`
2. Type `rm -rf first`
    - If the command complains about *first* not existing, ignore it.

The CSIF

The Computer Science Instructional Facility, CSIF, are computer labs in Kemper 067, 071, and 075 (all in the basement of Kemper) that are available to students to work on assignments for their CS classes.
They come preinstalled with all software that you will need to complete your assignments and serve as a convenient meeting point for partners to get together to work on their assignments in person.

The CSIF uses a networked file system, so it doesn't matter what computer you use. Any work you do on one computer will appear on all the others. Files are backed up regularly so you are never in danger of losing anything you do there.

Logging in Locally

All students enrolled in a CS course should have an account on the CSIF.
You use your normal UCD username and password to login into a CSIF computer. If you are enrolled in a CS
course but are still unable to log in to a computer contact support over email at support@cs.ucdavis.edu
or in person at 1114 Academic Surge.

To create a terminal window on a PC, click in the upper left corner of the screen.
A search box should appear in the upper middle portion of the screen. Type **terminal** in the box and select the
application called terminal. A terminal window should now be open. For the rest of
this tutorial will only be typing in one of these terminal windows. If you like you can drag the terminal application to the left side of the screen to add it to your favorites for easier access in the future

Once you have created a terminal window, you will see a “command line prompt” indicating that Unix is ready to accept your instructions. This prompt will usually have your username, computer name, and end with
a "$", but may be terminated by a "%". The prompt is generated by a program known as a *shell*. This is simply a program that accepts your commands and gets UNIX to execute them.

Logging in Remotely

If you are in the CSIF please **SKIP ahead** to Navigating the UNIX File System.

Connect to the School VPN

Before you can connect to the CSIF you need to connect to the school VPN. The instructions for how to install
and connect to the CSIF can be found
at https://servicehub.ucdavis.edu/servicehub?id=ucd_kb_article&sysparm_article=KB0006637.

Choose a Computer

Now that you are connected to the school VPN we need to figure out which computer you want to connect to. A full list of
all the
computers and their statuses can be found
at http://iceman.cs.ucdavis.edu/cgi-bin/nagios3/status.cgi?servicegroup=all&style=overview.
You'll only be able to access this link if you are **connected to the school VPN.**

You'll want to pick a computer that is whose status UP and preferably and whose services are OK. If a computer's
services are have warnings or criticals it will likely still work but if it doesn't, try a different PC.

The CSIF computers use a networked file system so any files you create on one computer are available on all computers. So if you find that your favorite computer is down for the day or is just running slow you can connect to a different computer and not lose any work.

SSHing on MAC

1. Open a terminal. You can find this under `Applications/Utilities/Terminal`
2. Type **ssh username@pc_computer_name.cs.ucdavis.edu** where username is your UC Davis username and computer name
   is the name of computer you selected in the previous step (pc13, pc16, pc3, etc).
   - As an example, if I wanted to connect to pc7 I would type `ssh mfbutner@pc7.cs.ucdavis.edu`
3. If this is your first time connecting to this computer ssh will ask you if you are sure you want to connect.
   It looks something like

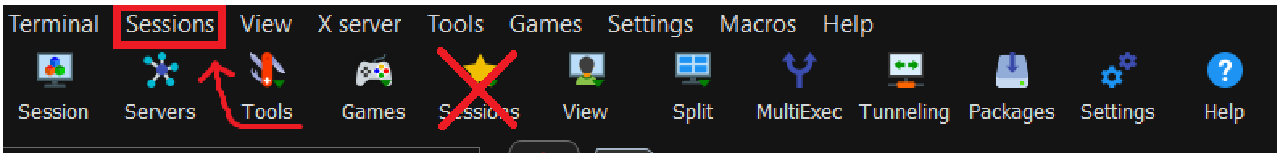
   ```
   The authenticity of host 'pc17 (169.237.5.14)' can't be established.
   RSA key fingerprint is f0:ad:72:d2:2e:84:ab:53:2e:06:60:7d:3e:38:17:3c.
   Are you sure you want to continue
   connecting (yes/no)?
   ```

   Type **yes** to confirm that you want to connect.
4. You will then be prompted to enter your password. Enter it now. When you type nothing will show up.
   This is normal and to help protect your password from being stolen by anyone looking at your screen

You should now be connected to the CSIF.
Skip down to Navigating the UNIX File System to continue the tutorial.

SSHing on Windows

You can follow the same steps as listed under SSHing on MAC but open a Cygwin or WSL terminal.
Alternatively, you can use the application MobaXTerm which
can make things a little easier.

1. Download MobaXTerm at https://mobaxterm.mobatek.net/download.html. Select the Home Edition
2. Run MobaXTerm
3. Click on the Session's drop down menu and select "New session".
   - The drop down menu, not the one with star
   - 

4. Click SSH
5. Under "Remote Host" type the *pc_name.cs.ucdavis.edu* where pc_name is the name of the computer
   you selected in the previous step(pc3, pc5, pc16, etc.).
6. Check "Specify username" and enter your UC Davis username in the box
7. Click Ok
8. You will then be prompted to enter your password. Enter it now. When you type nothing will show up.
   This is normal and to help protect your password from being stolen by anyone looking at your screen
   - You'll then be asked if you like to save your password.
     If you don't want to have to enter it every time say yes.

You should now be connected to the CSIF.

If you want to connect to the same computer again in the future, you can do so
through the sessions button or through the Quick Connect Box on the left.

Navigating the UNIX File System

Each directory in the file system has a *path name* that uniquely specifies the location of that directory within
the computer’s file structure. As you move within the file structure, you can use the command pwd to display the path name of current location. pwd stands for "print working directory". Type `pwd` now.

The shell should display */home/* followed by your username, e.g. */home/ssdavis*. This is your *home directory*.
This is the place where you "live" in the Unix file system. Each user has their own *home directory* that is completely under their control.

The cd (change directory) command allows you to move to any other public directory in the file system. The
syntax for cd is: cd *path_name*. Now change to the */bin* directory by typing `cd /bin`.
After executing the directory change, use `pwd` to see that you really are in the */bin* directory.

The ls (list) command allows you to see the contents of a directory. The syntax for ls is: ls [-al] [*filename*] ,
where the "[]" indicate optional values. Now simply type `ls`.

You will see a list of the names of the files in the */bin* directory. The */bin* directory holds many of the basic UNIX
utility programs.

Before you look into the options for ls, I want you back in your home directory. You could type *cd /home/*
followed by your username, e.g. cd */home/mfbutner*, but there is a shortcut. Simply type `cd`

After the shell has executed the change directory command, type `pwd`. You can see that simply typing `cd` without a path name will automatically return you to your home directory.

Now type `ls`.

If you have not created any files or directories in your account, then nothing will be displayed.

Many shell commands have options that may follow the command after a space and a dash. The options
change the behavior of the command. For ls, the options change what you see giving you specialized types of information which you wouldn't see using the basic command.

Now type: `ls –a`. The –a option stands for “all”.

You should see a bit more than did you last time. Pay attention to the files that start with a ".".
Files that start with a "." are called hidden files and are not normally displayed as most users don't need to
interact with them.

There are two unusual files listed, ".", and "..".
These have special meanings in path names. "." stands for the current directory.

Now type `cd .` (with just one period after cd).

Now type `pwd`. You haven’t moved within the directory structure as you changed into the current directory
(the one you were already in)!

"..”" stands for the parent directory of the current directory. Now type `cd ..`
(with two periods after cd). Now type `pwd`. You should see “/home”.

Now return to your home directory by simply typing `cd`

Now type `ls –al`

The –l option stands for long. The shell will display a line of information for each
file. Here is an excerpt from my instructor's directory:

Permissions	Links	Owner	Group	Size	Date	Name
drwxr-xr-x	27	ssdavis	users	4096	Jan 4 21:29	.

Permissions	Links	Owner	Group	Size	Date	Name
drwxr-xr-x	4185	root	users	147456	Jan 7 07:56	..
-rw-r--r--	1	ssdavis	users	1648	Oct 5 1993	.DECkeymaprc
-rw-------	1	ssdavis	users	0	Apr 3 2000	.ICEauthority
-rw-r--r--	1	ssdavis	users	698	Oct 5 1993	.SGIkeymaprc
-rwxr--r--	1	ssdavis	users	2481	Oct 5 1993	.X11Startup
-rw-r--r--	1	ssdavis	users	2285	Feb 26 1997	.addressbook.lu
-rwx------	1	ssdavis	users	2546	Oct 13 10:30	.cshrc
-rw-------	1	ssdavis	users	1869	Sep 25 1996	.cshrc.COPY4U
-rwx------	1	ssdavis	bin	1864	Mar 26 1998	.cshrc.OLD

Your output won't have the headers. I added them so you would know what each field is.

Here is a brief introduction to the information displayed in the long format.

File Permissions

This is the first field in the output.

drwxr-xr-x

- The d at the very left says the file is a *directory*.
- The next letters rwx say that the owner (myself) has read, write and execute permissions for the file. For a
  directory, execute permission means I can cd to that directory. Read permission means I can use the command
  ls to see the contents of the directory. Write permission means I can create additional files in that directory.
  The next two groups of three show that other users (group and world) are only allowed read and execute
  permissions.

-rwxr--r--

- The dash at the very left indicates an ordinary file, as opposed to a directory. Here the read, write and execute
  permissions mean the ability to look at the file (using commands like more, see below), to edit the file (using a
  text editor, see below), or execute the file (if it is an executable program -- otherwise this permission doesn't
  mean much).

-rw-r--r--

- This set of permissions shows an ordinary file which can only be read and edited by the owner, and only read
  by everyone else.

The Remaining Fields

- Links: number of shortcuts to this file from other directories.
- Owner: who owns this file
- Group: what user group this file belongs to
- Size: the size of the file in bytes
- Date: when the file was last modified
- Name: the name of the file

Notice there were no pathnames after any of the ls commands. That means that you saw the contents of the
current working directory. If you do put a pathname afterwards, then you will see the contents of the directory identified by that pathname. Now type `ls /`

In this example, the pathname you gave ls was the *root* directory (indicated by a sole backslash), which is
the top directory in the Unix file system. All the files you see listed there are subdirectories of the root. Notice the subdirectory *home*. That is the subdirectory in which the home directories of everyone having an
account are located.

Creating and Deleting Directories

To create a directory use the *mkdir* command. Its syntax is *mkdir {directory_name}*+. The "{}+" indicate that there must be at least one directory name given

Now type `mkdir first`

Now type `ls -l` You should have a directory named first now in your file list.

Now type `cd first` to move into the *first* directory.

Type `pwd` to see where you are

Now type `mkdir second third`

Now type `ls -a`. The display should be:

```
. .. second third
```

The syntax for the remove directory command is rmdir *{directory_name}*+.

Now type `rmdir second` to remove the directory named second.

Type `ls` to check your work.

You should see only *third* listed.

Copying Files

Use `pwd` to make sure that you are still in */home/user_name/first*.

If you aren't you can type `cd ~/first`. You can use "~" as shorthand for your home directory.

There are two syntaxes for the copy file command.

1. *cp source_file_name destination_file_name*
    - Copy source_file_name to the destination with the new name
2. *cp {source_file_name}+ destination_directory*
    - Copy one or more files into destination_directory

But before we can copy a file we need to create one first. Type `ls /home > users.txt`. This will create
a file named *users.txt* that has the usernames of everyone on the CSIF.

Now to copy the file type `cp users.txt myfile`

Use `ls` to make sure that you were successful. You should see *users.txt*, *myfile*, and *third* listed.

Now copy this file down into the *third* directory by typing `cp myfile third`

You could check your work by cding into the *third* directory and then using ls, but there is a simpler way. Just
type `ls third` to display the contents of the *third* directory. You should see only *myfile* listed.

Moving and Renaming Files

The mv command has three syntaxes

1. *mv old_file_name new_file_name*, to change the name of a file
2. *mv {filename}+ directory_name*, to move file(s) into a directory
3. *mv old_directory_name new_directory_name* to change the name of a directory.

You will try each one in order.

Now type `mv myfile wombat` to rename myfile to wombat.

Use `ls` to check your work. You should see only *third*, *users.txt*, and *wombat* listed.

Now type `mv third/myfile .` (note the period indicates the current directory) to move myfile to the current directory.

Again use `ls` to check your work. You should now have *myfile*, *users.txt*, *third*, and *wombat* in the current directory.

Now type `mv third fourth`, to rename the *third* directory to *fourth*.

Use `ls` to check your work. You should now see *fourth*, *myfile*, *users.txt*, and *wombat* listed.

Removing Files

The syntax of the command to remove files is *rm {file_name}*+.

Now type `rm myfile`

Use `ls` check your work. Your current directory, *first*,
should now contain only one directory, *fourth*, and two files, *wombat* and *users.txt*.

Shell Command Line Editing

Let’s say you wish to copy wombat into the fourth directory, but you **MAKE A MISTAKE** type `cp woombat forth/snake`, and, **BEFORE YOU HIT THE RETURN KEY**, you notice your errors. You can go back to make the
appropriate changes by using the left-arrow key to go to the 'o' in forth and the typing `u`; then use the left-arrow key
to an 'o' in woombat and hit the Delete key, which will remove the extra 'o';
and then hit the return key to process the command. Try it now.

Type `ls fourth` to check your work. You should see only *snake* listed.

On the other hand, if you have already hit the return key when you notice your typing error, you can use the up-arrow to go back to my previous command. Use the up-arrow to go back through several previous
commands, and the down-arrow to go forward. Note that you can also use this to repeat (without change) a previous command.

The shell also allows you to do file name completion, again a great saver of typing and
time. Let’s say you wish to move wombat into the fourth directory. The command would be "mv wombat fourth".

But rather than typing the whole thing, you can type `mv w` then hit the Tab key. (This is super useful and if
you ever see me typing really "fast" on the command line in examples in class it's because I'm using this.)

Since *wombat* is the only file in the current directory that starts with 'w', the shell would complete the file name (wombat) for you, you would then just type `f` and again hit the Tab key.

Since *fourth* is the only file that begins with an 'f' the shell will complete the file name for your again, giving you "mv wombat fourth/" on the command line!

Hit the then Enter key, and again type `ls fourth`, and you should see only *snake* and *wombat*.

The shell will complete a file name as long as there is only one file with a name that matches what you have
typed so far. This means that you may have to type several letters at the beginning of a file name if there are
other files with identical beginnings of their names. If there are multiple matches you can press Tab twice
to have the shell show you all the files that match what you have typed so far.

The Terminate Key

On the left side of the keyboard is a key labeled Ctrl. Use this key in combination with others to send control
signals to Unix. In UNIX literature, the Ctrl key is specified by using the ^. For example, ^C means "hold down the Ctrl key, then type c". Notice I don't mean capital C, even though by convention that is how it's written. ^C usually causes instant termination of any running program.

For example, the ping utility checks the quality of the network connection between your computer
and a remote computer. It will continue to check the quality until you terminate it by typing ^C.
The syntax of ping is *ping computer_name*.

Now type `ping pc3`. (If pc3 is down select another comptuer that is up)

Your screen will start filling with lines indicating the quality of the
connection to pc3. Terminate ping by typing `^C`.

ping should stop, and you should see the command prompt again.

You can use `^C` in most situations when a command or program is causing the computer to perform in an
unsatisfactory manner.

Displaying Files

The syntax of the command to display a file is: *cat {file_name}+*.

Now type `cat wombat`

This will display the entire contents of the file on the screen. For short files this might be fine, but usually
you will wish to see the file a little at a time.

You can use *more {file_name}+* to scroll through a file.

Now type `more wombat`.

You will see the first screen-worth of the file displayed. To scroll forward one line, press the Enter key. To
scroll forward an entire screen, press the Space bar. To quit in the middle of displaying a file, type the letter
`q`. There is no way to scroll backwards with more.

Sometimes you will just wish to see the first few lines of a file. You can then use the head command. The
syntax is *head [-n] {file_name}+*, where "n" is the number of lines to be displayed, e.g. head -3 snake. If you
leave out n, then head defaults to the first 10 lines.

Now type `head -4 snake`.

You should see the first 4 lines of the file displayed.

Logging Out

If you did not remotely connect to the CSIF please skip this step.

It is time for you to end your ssh session and return to your original computer.
Now type `logout`

You should receive the message “Connection to pc23 closed.”, where pc23 was the name of the remote
computer you connected to.

You should now see that you are back on your own computer. You can double-check this
by running the `pwd` and `ls` commands.

File Transfer

You can use the sftp (Secure File Transfer Protocol) command to retrieve or place files in remote locations,
provided you have permission to access the location where the files are.

The syntax of sftp is: *sftp computer_name*. You are now going to go through the process
of getting your wombat file from another computer.

Before doing this step, make sure you are **NOT** connected to the CSIF through ssh.
If you are you will end up transferring file from the CSIF to the CSIF instead of
from the CSIF to your computer. Also make sure you are in directory that you can easily find through your file browser. If you are not, use *ls*, *cd*, and *pwd* to navigate to a directory you can.

Now type `sftp user_name@computer_name.cs.ucdavis.edu`, where computer_name is some CSIF computer and user_name is
your UCD username name.

You can navigate the remote computer(the CSIF computer) in sftp using the cd, pwd, ls, etc.
If you place an l (the letter L) in front of these commands (lcd, lpwd, lls, etc.) and they will work on your local computer
(the one that is physically in front of you).

Now type `pwd`, and you will see that you are in your own home directory on the CSIF.

Now type `cd first/fourth` to move the remote sftp session into the *fourth* directory.

Now type `lpwd`. This will show you the directory we will be downloading the files from the CSIF to.

Now type `ls` to see the files available to you on the remote computer. You should see *snake* and *wombat* listed.

The "get" command retrieves files from the remote computer. Now type `get snake` to download `snake` to you computer.

Type `get wombat` to download wombat to your computer.

If you had wanted to move multiple files you could use *mget* which has the syntax of *mget {file_name}*+.
So we could have simplified the previous two commands into *mget snake wombat*.

If you had wanted to transfer a file from the local computer to the remote computer you would use *put* or *mput*.

Now that you are done transferring files type `bye` to end the sftp session.

To check that the files are actually on your computer type `ls`. You should also be able to see them if you open your file browser to this directory.

GUI File Transfer

There are GUI applications that can help you move files between computers without you having to type commands. If you are on Windows and using MobaXterm you should see the files on the remote computer on the left side of the screen. If you right-click one you should see the "Download" option appear. There is also an up-arrow in that area that can be used to upload files to the remote machine.

If you are on Mac, many students in the past have found Cyber Duck helpful for moving files around.

Learning More About Commands

We only cover the most basic of commands in this tutorial along with a few of their options. If you want to learn more about what a command is capable of doing or what comands are available Google is your friend.

Submitting Your Assignment

Find the Linux Tutorial Assignment on Prairie Learn and submit the *snake* and *wombat* files you created in this tutorial.