Typographical conventions

In what follows, we shall use the following typographical conventions:

* Characters written like this are commands to be typed into the computer as they stand.
* Characters written *like this* indicate non-specific file or directory names.
* Words inserted within square brackets (e.g. [Ctrl]) indicate keys to be pressed.

So, for example,

% ls *anydirectory*[Enter]

means "at the Linux prompt % (or $ on our system), type ls followed by the name of some directory, then press the key marked Enter"

Don't forget to press the [Enter] key: commands are not sent to the computer until this is done.

UNIX/Linux is case-sensitive

Note: UNIX is case-sensitve, so LS is not the same as ls.  
  
The same applies to filenames, so myfile.txt, MyFile.txt and MYFILE.TXT are three seperate files. Beware if copying files to a PC, since DOS and Windows do not make this distinction.

Getting Started

This course uses an online IDE and a set of command line tools which make submitting your assignments easier. You will even be able to automatically check your work before you submit. These tools were created by Harvard University and require a GitHub account.

Let's get you set up for checking and submitting your work.

1. Create a GitHub account here: [https://github.com/ (Links to an external site.)](https://github.com/)
2. Register your account here: [https://submit.cs50.io/invites/6ff235d3e38e4d1ab3157f569fc03c43 (Links to an external site.)](https://submit.cs50.io/invites/6ff235d3e38e4d1ab3157f569fc03c43)
3. Go to the online IDE here: [https://ide.cs50.io/ (Links to an external site.)](https://ide.cs50.io/)

You are now set up to check and submit assignments through the online IDE. *Spend some time clicking around the online IDE to get familiar with it. Try opening a new terminal, creating and opening files, etc. Also, take a look at some of the options such as IDE themes available*.

Checking Your Work

Let's see how you are doing so far. We first need to set up a secure connection with the check server. Carefully follow the steps here: [https://cs50.readthedocs.io/github/#ssh (Links to an external site.)](https://cs50.readthedocs.io/github/#ssh)

Next, in a terminal tab in the online IDE, try the following command.

% check50 uwrf-csis/csis248/main/lab1

You should see

You seem to be missing these required files:  
typescript  
You are currently in: ~, did you perhaps intend another directory?

Showing Your Work

In order to grade this lab and future labs, you'll need to create a record showing that you did the lab. We can use a command line utility called "script" to do this, we'll use the **script** command. Let's see how this command works.

% script

After typing this command, you should see something like

Script started, file is typescript  
ubuntu@ide-8210bda92ffc4712915fcb2ff712fbfd:~$

The second line ending with '$' is the command prompt that the script command prints. Any command you type into this prompt is recorded in a text file called "typescript".

To stop recording your commands, type "exit" into the prompt.

% exit

Now type

% check50 uwrf-csis/csis248/main/lab1

You should be asked to enter your GitHub username and password. Do this. You should see

Connecting.......  
Authenticating.....  
Verifying.......  
Preparing................  
Uploading................  
Waiting for results....................................  
Results for uwrf-csis/csis248/main/lab1 generated by check50 v3.1.2  
:) typescript file exists  
:( command "ls" is present  
    Make sure that you have tried all commands in the lab. To start the script command so that it appends to you typescript file, use 'script -a typescript'  
:( command "ls -a" is present  
    Make sure that you have tried all commands in the lab. To start the script command so that it appends to you typescript file, use 'script -a typescript'  
:( command "mkdir" is present  
    Make sure that you have tried all commands in the lab. To start the script command so that it appends to you typescript file, use 'script -a typescript'  
:( command "cd" is present  
    Make sure that you have tried all commands in the lab. To start the script command so that it appends to you typescript file, use 'script -a typescript'  
:( command "cd ~" is present  
    Make sure that you have tried all commands in the lab. To start the script command so that it appends to you typescript file, use 'script -a typescript'  
:( command "cd .." is present  
    Make sure that you have tried all commands in the lab. To start the script command so that it appends to you typescript file, use 'script -a typescript'  
:( command "pwd" is present  
    Make sure that you have tried all commands in the lab. To start the script command so that it appends to you typescript file, use 'script -a typescript'  
To see the results in your browser go to https://submit.cs50.io/check50/42c2d684f30e88b024fe0c9f328c129204aa1d2b

The output shows that we have a typescript file present, but we have not typed in any of the commands for this lab. Note there are 8 checks, and only 1 passes. Complete the lab and all checks should pass.

1.0 Start Recording Your Commands

First, let's run **script** again to record the commands that are typed in to the prompt. Run the following

% script -a

The "**-a**" at the end of the command above is an option which tells script to append new commands to the current typescript file. ***If you do not run script with the -a option, the typescript file will be overwritten! This means that you will have to redo the lab, so make sure to use the -a option with the script command. It doesn't hurt to simply always use -a.***

1.1 Listing files and directories

When you first open a terminal, your current working directory is often your home directory. Your home directory has the same name as your user-name, and it is where your personal files and subdirectories are saved.

To find out what is in your current working directory, type

% ls

The ls command lists the contents of your current working directory.

There may be no files visible in your home directory, in which case, the Linux prompt will be returned.

ls does not, in fact, cause all the files in your home directory to be listed, but only those ones whose name does not begin with a dot (.) Files beginning with a dot (.) are known as hidden files and usually contain important program configuration information. They are hidden because you should not change them unless you are very familiar with Linux!!!

To list all files in your home directory including those whose names begin with a dot, type

% ls -a

ls is an example of a command which can take options: -a is an example of an option. The options change the behavior of the command. There are online manual pages that tell you which options a particular command can take, and how each option modifies the behavior of the command. (See later in this tutorial)

1.1.1 **Important Note**

The **script** command records ALL keystrokes! So, if you hit "extra" keys when you typed in the above command (**ls -a**), those keys are recorded. This can cause checks to fail when they shouldn't. Here is a video explaining this and how to enter commands and ensure that the commands are checked correctly:

1.2 Making Directories

We will now make a subdirectory in your home directory to hold the files you will be creating and using in the course of this tutorial. To make a subdirectory called unixstuff in your current working directory type

% mkdir unixstuff

To see the directory you have just created, type

% ls

1.3 Changing to a different directory

The command cd directory means change the current working directory to 'directory'. The current working directory may be thought of as the directory you are in, i.e. your current position in the file-system tree.

To change to the directory you have just made, type

% cd unixstuff

Type ls to see the contents (which should be empty)

Exercise 1a

Make another directory inside the **unixstuff**directory called **backups**

1.4 The directories . and ..

Still in the **unixstuff**directory, type

% ls -a

As you can see, in the **unixstuff**directory (and in all other directories), there are two special directories called ( **.**) and ( **..**)

In UNIX, ( **.**) means the current directory, so typing

% cd .

NOTE: there is a space between cd and the dot

means stay where you are (the **unixstuff**directory).

This may not seem very useful at first, but using ( **.**) as the name of the current directory will save a lot of typing, as we shall see later in the tutorial.

( **..**) means the parent of the current directory, so typing

% cd ..

will take you one directory up the hierarchy (back to your home directory). Try it now.

Note: typing cd with no argument always returns you to your home directory. This is very useful if you are lost in the file system.

1.5 Pathnames

Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory, type cd to get back to your home-directory and then type

% pwd

The full pathname will look something like this -

**/home/username**

which means that 'username' (your home directory) is in the directory home. / is the "root" directory. To go to the root directory, you can use **cd /**

Exercise 1b

Use the commands ls, pwd and cd to explore the file system.

Find the files named "passwd", "stdout",  and "echo"?

(Remember, if you get lost, type cd by itself to return to your home-directory)

1.6 More about home directories and pathnames

First type cd to get back to your home-directory, then type

% ls unixstuff

to list the contents of your unixstuff directory.

Now type

% ls backups

You will get a message like this -

backups: No such file or directory

The reason is, **backups**is not in your current working directory. To use a command on a file (or directory) not in the current working directory (the directory you are currently in), you must either cd to the correct directory, or specify its full pathname. To list the contents of your backups directory, you must type

% ls unixstuff/backups

~ (your home directory)

Home directories can also be referred to by the tilde **~**character. It can be used to specify paths starting at your home directory. So typing

% ls ~/unixstuff

will list the contents of your unixstuff directory, no matter where you currently are in the file system.

What do you think

% ls ~

would list?

What do you think

% ls ~/..

would list?

What do you think

% ls ~/.

would list?

1.7 Submit Your Work

Stop the script command by typing

% exit

Then type

% check50 uwrf-csis/csis248/main/lab1

Make sure all checks pass. If they do not, then you probably skipped a step in the lab. The checks show which commands are missing. Rerun the **script -a** command and finish the lab and verify that all checks pass. Don't forget to use the **exit** command to stop the script command.

If all checks pass, you should see

:) typescript file exists  
:) command "ls" is present  
:) command "ls -a" is present  
:) command "mkdir" is present  
:) command "cd" is present  
:) command "cd ~" is present  
:) command "cd .." is present  
:) command "pwd" is present

Finally, submit your work by typing the following command and answering the question that follows.