In these exercises, you will practice using commands for navigating and managing files and directories.

**1.1. Get the location of the present working directory**

**pwd**

When working in a Linux terminal, you will always be working from a directory. By default, you will start in your home directory. To get the absolute path of your present working directory, enter the following:

1. pwd

This will print the name of the directory you are currently working in.

**1.2. List the files and directories in a directory**

**ls**

To list the files and directories in the current directory, enter the following:

1. ls

If your directory happens to be empty, ls will not return anything.

The following command will list the many binary and executable files which are present in your /bin (binaries) directory.

1. ls /bin

The /bin directory happens to be where Linux commmands such as ls and pwd are stored. For example, you can see that ls is present by entering the following:

1. ls /bin/ls

To list all files starting with b in the /bin directory, try entering the following:

1. ls /bin/b\*

***Tip:****The asterisk \* is a special character called a wildcard. It is used to represent any string of characters.*

To list all files ending in r in the /bin directory, enter the following:

1. ls /bin/\*r

To print a longer list of files with additional information, such as the last-modified date, enter the following:

1. ls -l

Here are some common options that you can try with the ls command:

| **Option** | **Description** |
| --- | --- |
| -a | list all files, including hidden files |
| -d | list directories only, do not include files |
| -h | with -l and -s, print sizes like 1K, 234M, 2G |
| -l | include attributes like permissions, owner, size, and last-modified date |
| -S | sort by file size, largest first |
| -t | sort by last-modified date, newest first |
| -r | reverse the sort order |

To get a long list of all files in /etc, including any hidden files, enter the following:

1. ls -la /etc

Here we combined the options -l and -a by using the shorter notation, -la.

**2.1. Create a directory**

**mkdir**

The mkdir command is used to create a new directory.

To create a directory named scripts in your current directory, run the following command:

1. mkdir scripts

Use the ls command to verify whether the scripts directory was created:

1. ls

You should see a directory named scripts listed.

**2.2. Change your current working directory**

**cd**

To change your present working directory to the scripts directory, run the following command:

1. cd scripts

Now use the pwd command to verify whether your current working directory has changed as expected:

1. pwd

You can enter cd without any directory name to move back to your home directory:

1. cd

Then, enter the pwd command to verify whether your current working directory has changed:

1. pwd

The syntax .. is a shortcut that refers to the parent directory of your current directory. Run the following command to move the directories up one level:

1. cd ..

**2.3. Create an empty file**

**touch**

First, return to your home directory by entering:

1. cd

Next, use the touch command to create an empty file named myfile.txt:

1. touch myfile.txt

Now use the ls command to verify the creation of myfile.txt:

1. ls

If the file already exists, the touch command updates the access timestamp, or last-modified date of the file. To see this, enter:

1. touch myfile.txt

And use the date command to verify the date change:

1. date -r myfile.txt

**3.1. Search for and locate files**

**find**

The find command is used to search for files in a directory. You can search for files based on different attributes, such as the file's name, type, owner, size, or timestamp.

The find command conducts a search of the entire directory tree starting from the given directory name.

For example, the following command finds all .txt files in the /etc directory and all of its subdirectories:

1. find /etc -name '\*.txt'

You can also search for .conf files using the below command:

1. find /etc -name '\*.conf'

***Note:****Along with listing all the .txt files, the terminal may return "Permission denied" errors.  
These errors are normal, as you have limited access permissions on the lab machine.*

**3.2. Remove files**

**rm**

The rm command is used to delete files, ideally with the -i option, which creates a prompt to ask for confirmation before every deletion.

To remove the file myfile.txt, enter the following command and press y to confirm deletion, or n to deny deletion:

1. rm -i myfile.txt

Use the ls command to verify removal:

1. ls

***Tip:****When you are only removing one file with the rm command, the -i option is redundant. But if you want to remove multiple files, for example by using a wildcard to find all filenames matching a pattern, it's best practice to confirm or deny each deletion by including the -i option.*

*Be careful when deleting files or directories! There is normally no way to restore a deleted file once it is deleted, as there is no trash folder. This is why you should always back up, or archive, your important files. You will learn more about archiving files soon.*

**3.3. Move and rename a file**

**mv**

You can use the mv command to move files from one directory to another and/or rename them.

Before doing so, let's first create a new file called users.txt:

1. touch users.txt

You should always use caution when moving a file. If the target file already exists, it will be overwritten, or replaced, by the source file.

Conveniently, however, when the source and target directories are the same, you can use mv to rename a file.

To illustrate this, use mv to rename users.txt to user-info.txt by entering the following command:

1. mv users.txt user-info.txt

Because the source and target directories are the same (your present working directory), the mv command will rename the file.

Now use the ls command to verify the name change:

1. ls

Now, you can move user-info.txt to the /tmp directory as follows:

1. mv user-info.txt /tmp

Use the ls command twice to verify the move:

1. ls
2. ls -l /tmp

**3.4. Copy files**

**cp**

You can use the cp command to copy user-info.txt, which is now in your /tmp directory, to your current working directory:

1. cp /tmp/user-info.txt user-info.txt

Use the ls command to verify that the copy was successful:

1. ls

At times, you may want to copy the contents of an existing file into a new one.

The following command copies the content of /etc/passwd to a file named users.txt within the current directory:

1. cp /etc/passwd users.txt

Again, use the ls command to verify if the copy was successful:

1. ls

**1. Display the contents of the /home directory.**

1. ls /home

**2. Ensure that you are in your home directory.**

1. cd
2. pwd

**3. Create a new directory called tmp and verify its creation.**

1. mkdir tmp
2. ls

**4. Create a new, empty file named display.sh in the tmp directory, and verify its creation.**

1. cd tmp
2. touch display.sh
3. ls -l

**5. Create a copy of display.sh, called report.sh, within the same directory.**

1. cp display.sh report.sh

**6. Move your copied file, report.sh, up one level in the directory tree to the parent directory. Verify your changes.**

1. mv report.sh ../
2. ls
3. ls ../

**7. Delete the file display.sh.**

1. rm -i display.sh

**8. List the files in /etc directory in the ascending order of their access time.**

1. ls -ltr /etc/

**9. Copy the file /var/log/bootstrap.log to your current directory.**

1. cp /var/log/bootstrap.log .