UNIX Navigation Screenshot Safari

Basic command line and file navigation exercise

The following is by no means a full lesson in command-line navigation, but if you master these simple basics, I feel you will be able to get around and play in UNIX and Linux terminal environments with some confidence. Please, at first only type the commands listed here and nothing else. You will need to document your progress later. If you make mistakes that is ok, just retype and continue. Do not press enter repeatedly. This exercise should all fit on one terminal screen. After you've completed the assignment, practice until you have the commands memorized and down. Please be aware that I use the the term "folder" and "directory" interchangeably. The technical term in UNIX is "directory" but both terms mean the same thing.

Open terminal [if there is not a UNIX or Linux environment available there will be alternative instructions for replit]. At the CLI (**C**ommand **L**ine **I**nterface) prompt, type the following command and press enter. Please be aware that casing (capitalization) and spacing matter.

```
pwd
```

This command prints your working, or "current" directory. This is your home directory. Now type

```
ls
```

This lists the contents of the folder or directory. In this folder, there should be a directory listed named "Documents." To work in this directory, type the following command.

```
cd Documents
```

You notice your prompt will now say Documents right before the prompt symbol [$, %, depending on version]. The command cd stands for "**c**hange **d**irectory." To jump all the way back to your home folder (sometimes called "root") from wherever you are, type

```
cd ~
```

To jump back to the directory you were just in type

```
cd -
```

You are now back in Documents. Let's make a new directory called "unix."

```
mkdir unix
```

Now let's list our Documents directory to see that new directory.

```
ls
```

We can make a new directory named "gawk" in that directory without being currently in that directory by typing in the name of the directory it resides in separated by '/' like so

```
mkdir unix/gawk
```

we can jump down to "gawk" by changing directories the same way.

```
cd unix/gawk
```

Now, we can create a new blank text file in our directory [for replit the command will be gawk$ echo >”blank.txt” [enter] otherwise it will not show up in the ls listing]

```
touch blank.txt
```

List your directory to see your new file.

```
ls
```

Now let’s delete that file with the remove command.

```
rm blank.txt
```

List our directory again to see if it is still there.

```
ls
```

Now to go up one level in the file hierarchy, type

```
cd ..
```

You are back in your “unix” directory. Now let's take a screenshot from our command line. [This will vary from version to version and OS to OS if running UNIX virtually. the command on macOSX is “screencapture” The following is the GNU/Linux version and also works on the windows command line as an alternative to setting up a nix environment on replit.] This screenshot you will turn in for your assignment so after the command “scrot” name your file with your last name & first name Pascal cased, date, and followed with the word unix. All one word (LastFirstMMDDYYunix). Do not forget to add .png to the end of the file name. [Turn-in instructions here]

```
scrot GreenbergHank042522unix.png
```

You can open files from the command line too. Take a look at your screenshot. Here’s a shortcut. Press the up arrow a few times and watch your prompt. Did you see your command line history? Now arrow down until you see your last “scrot” command. Now arrow over so the cursor is just behind the ’t’ in “scrot” delete “scrot” and type

```
open GreenbergHank042522unix.png
```

Your image preview program should open and you can see your work. Now close that preview program and get back to your terminal. We do not need that directory named “gawk”. Let’s remove that with the remove directory command (NOTE: the following command only removes empty directories. Use rm -r ‘remove recursively’ option '-r' to remove a folder and all its contents, however, use this with extreme caution.)

```
rmdir gawk
```

Now for our next lesson, we need a directory in our “unix” folder named “awk.”

```
mkdir awk
```

Now one last command. If your terminal screen gets too crowded and you want to start with a fresh screen, type

```
clear
```

[instructions to add files to the "awk" directory for the mini-lesson/demo on the AWK programming language]