cse15l-lab-reports

Lab Report 1 - Remote Access and FileSystem

For each of the commands `cd`, `ls`, and `cat`, and using the workspace you created in this lab we are doing the following:

- Share an example of using the command with no arguments.
- Share an exmaple of using the command with a path to a directory as an argument.
- Share an example of using the command with a path to a file as an argument.

cd

Small overview, the `cd` command stands for "change directory", this will switch whatever respective directory we are orignally in to the one we are calling.

1) An example of using the `cd` command without any arguements would look be as followed

```
[user@sahara ~]$ cd
[user@sahara ~]$
```

In this, we see that when we run the `cd` command in the working directory `home`, the following output does not produce anything. It is simply the action of us changing the directory. `cd` without arguemnts will just return us to the home directory

2) An example of using the command with a path a directory as as arguement would show the following

```
[user@sahara ~]$ cd lecture1
[user@sahara ~/lecture1]$ cd Hello.class
```

We see again, there is no output. However we can notice, when we use the command with a a path to the directory to the terminal, in this case lecture 1, the new line to write a new command displays `user@sahara ~/lecture1]$` meaning that we are now working out of the directory of lecture1

3) Finally, lets naviagate to a file.

```
[user@sahara ~/lecture1]$ cd Hello.class
bash: cd: Hello.class: Not a directory
[user@sahara ~/lecture1]$
```

It looks like there is an error. Why is that? This is because `cd` means change directory. When we pass an arguement with a file, a file is not a directory. Therefore if we run the command `cd` and try to access a file that is not in the current directory or the file itself, an error will occur becasue there is no directory that matches that name.

ls

A small overvirew of `ls`, shorthand for "list, as in listing the files and folders of the given path our console is currently working with

1) An example of using `ls` as itself and no entitity to folllow is:

```
[user@sahara ~]$ ls
lecture1
```

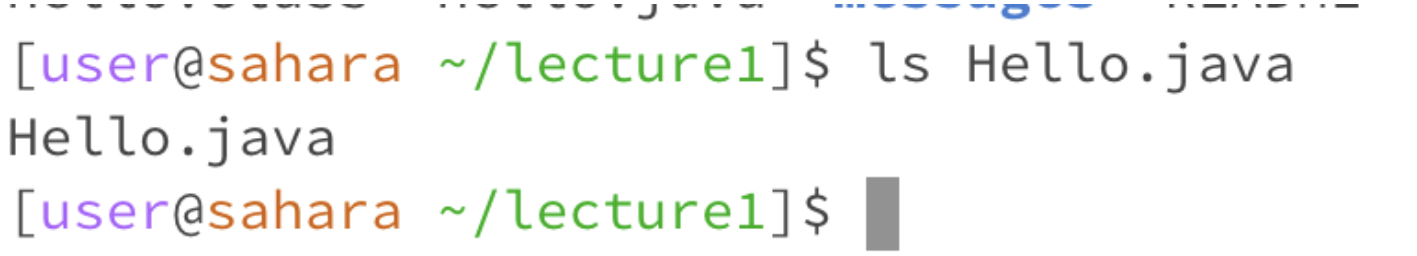
Here, we see the output is lecture1, this is becasue we are curerntly in the home directory. if we change our path to lecture1, and then run the empty `ls` command, we would see the contents within leture 1.

2) Next, lets access a directory with this command:

```
[user@sahara ~]$ ls lecture1
Hello.class  Hello.java  messages  README
```

Since we have accessed the directory, we can show the name of the files and folders within that directory.

3) Finally, we will use the `ls` command to a file:



using the `ls lecture1/messages` command, we can see the contents of this file. If we were to run another ls command to a file such as `Hello.class` , there is only "one file" within the "file" so we would get the output `Hello.class` as it is a single entitiy.

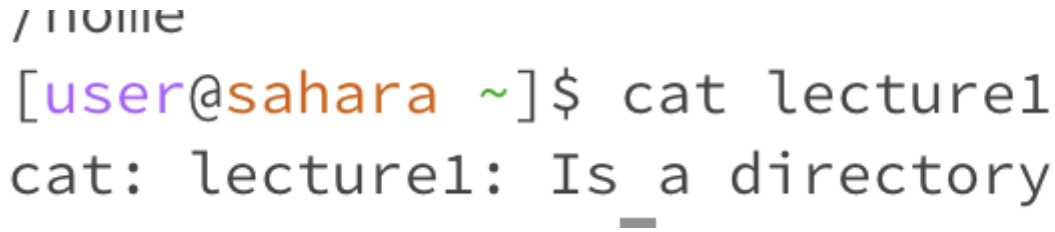
cd

Lets now utilize the cat command. cat is shorthand for "concatenate"- which is uswed to print the contents that are in the files of the respective path.

1) As we have continually done, lets run cat without any arguemnt:

Confusing right? What do we do now? We see this visual because when we run a `cat` command the console is waiting for us to input something until we personally terminate the program

2) Next, lets run cat with a path to a directory



We see the output `cat: lecture: Is a directory` This is because the purpose of `cat` is to print the contents of the files. When we call `cat` to something other than a file, there ovbiosly is nothing to print therefore we incur this error message.

3) Finally, lets print what the command is wanting us to do, aka a file

```
[user@sahara ~]$ cat lecture1/Hello.java
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;

public class Hello {
  public static void main(String[] args) throws IOException {
    String content = Files.readString(Path.of(args[0]), StandardCharsets.UTF_8);

    System.out.println(content);
  }
```

This is all the contents of the Hello.java file! `Hello.java` is a file in the directory `lecture1` . So the path `cd lecture1/Hello.java` will access the directory of lecture 1 and the file of Hello.java and print its contents!