Austin Dollar
3/11/21
CSCI 344
Lab 7

Creating a multi-level File System in Python

The code (copied and pasted below due to being too large for a screenshot)

```
#!/usr/bin/python3.7
# filename : directory.py

from datetime import datetime

###### declare root list#####
root = []


##########declare classes###############
class Item:
    def __init__(self, name, parentdir, permissions):
        self.name = name
        self.parentdir = parentdir
        self.permissions = permissions


class Directory(Item):
    def __init__(self, name, parentdir, permissions):
        self.UpdDate = datetime.now()
        self.dir = []
        Item.__init__(self, name, parentdir, permissions)


class File(Item):
    def __init__(self, name, parentdir, permissions, size):
      self.size = size
      Item.__init__(self, name, parentdir, permissions)


######################function calls for each case#######################
```

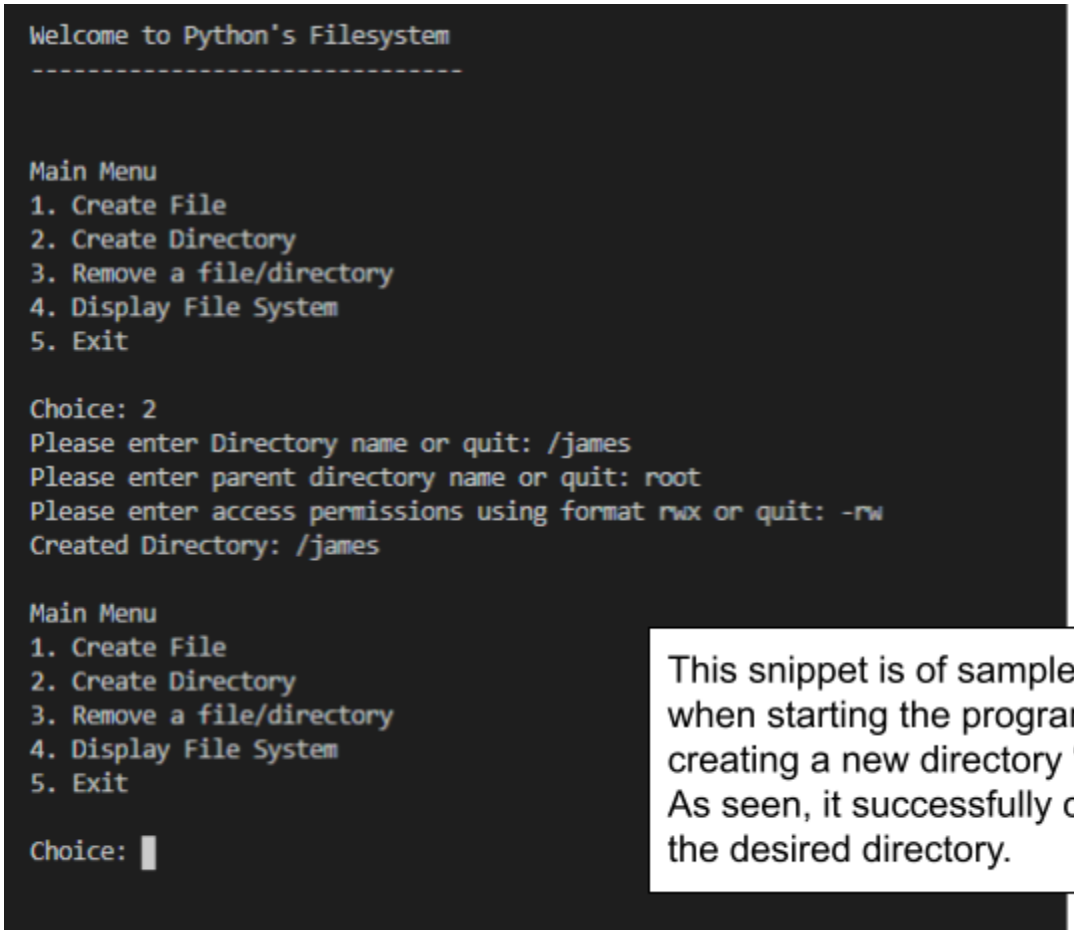
```
def command1():
    fname = input("    Please enter File name or quit: ")
    if fname == "quit":
        return
    pdname = input("    Please enter Directory name or quit: ")
    if pdname == "quit":
        return
    if any(obj.name == pdname for obj in root) == False:
        print("    ERROR: Parent Directory does not exist.")
        return
    faccess = input("    Please enter access permissions using format rwx or quit: ")
    if faccess == "quit":
        return
    size = input("    Please enter size(1-Small, 2-Medium, 3-Large: ")
    for obj in root:
        if obj.name == pdname:
            obj.dir.append(File(fname, pdname, faccess, size))
            print("    Created File: " + obj.name + "/" + fname)
    print(" ")
    random = input("    ")
    return


def command2():
    dname = input("    Please enter Directory name or quit: ")
    if dname == "quit":
        return
    pname = input("    Please enter parent directory name or quit: ")
    if pname == "quit":
        return
    if (any(obj.name == pname for obj in root) == False) and (pname != "root"):
        print("    ERROR: Parent Directory does not exist.")
        return
    access = input("    Please enter access permissions using format rwx or quit: ")
    if access == "quit":
        return
    root.append(Directory(dname, pname, access))
    print("    Created Directory: " + dname)



def command3():
```

```
    ford = input("    File or Directory?: ")
    if ford == "file":
        frname = input("    Please enter file name or quit: ")
        if frname == "quit":
           return
        pname = input("    Please enter parent directory name or quit: ")
        if pname == "quit":
           return
        if (any(obj.name == pname for obj in root) == False) and (pname != "root"):
           print("    ERROR: Parent Directory does not exist.")
           return
        for obj in root:
           if obj.name == pname:
               for rm in obj.dir:
                   if rm.name == frname:
                       obj.dir.remove(rm)
    elif ford == "directory":
        pname = input("    Please enter parent directory name or quit: ")
        if pname == "quit":
           return
        if (any(obj.name == pname for obj in root) == False) and (pname != "root"):
           print("    ERROR: Parent Directory does not exist.")
           return
        for obj in root:
           if obj.name == pname:
               root.remove(obj)
    print( "     " + pname + "/" + frname + " deleted")




def command4():
    for obj in root:
        print("     " + obj.name)
        for fill in obj.dir:
            print("         " + fill.name)
```

```
#######################main driver of code#######################
exit = 0
startup = """
    Welcome to Python's Filesystem
    ------------------------------
    """
print(startup)

while exit != -1:

    ######main menu#######
    MainMenu = """
    Main Menu
    1. Create File
    2. Create Directory
    3. Remove a file/directory
    4. Display File System
    5. Exit
    """
    print(MainMenu)


    ###setup vars###
    command = input("    Choice: ")




#####handle case 1: create file######
    if command == "1":
        command1()


    #####handle case 2: create directory#####
    elif command == "2":
        command2()



#####handle case 3: remove file/directory######
```

```
    elif command == "3":
        command3()


    ######handle case 4: Display file system#########
    elif command == "4":
        command4()


    #####handle case 5: exit#####
    elif command == "5":
        exitString = """
    Goodbye!
    ------------------------------
        """
        print(exitString)
        exit = -1


    #### error message if invalid command####
    else:
        print("    ERROR! Try again")
```

The output



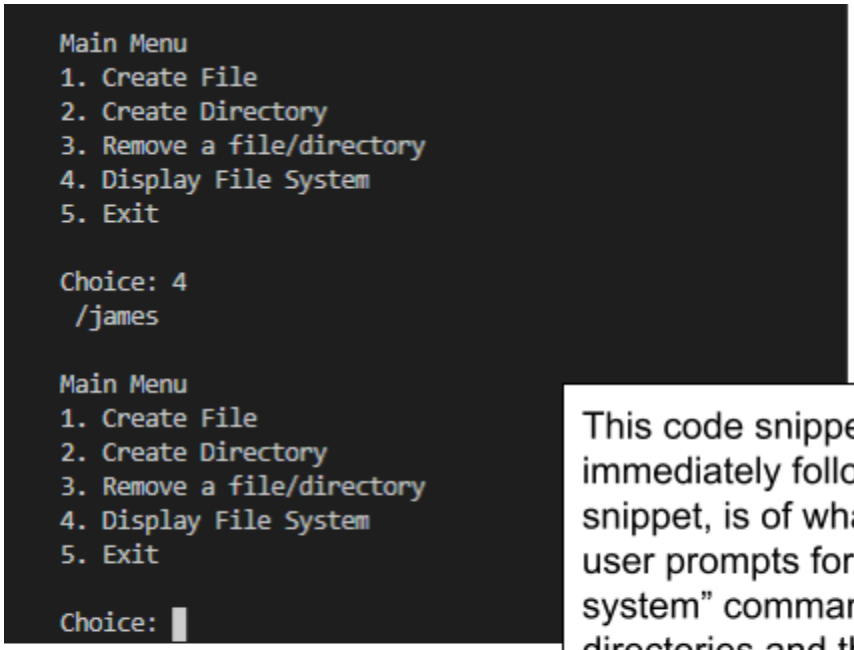
This snippet is of sample output when starting the program, then creating a new directory “james”. As seen, it successfully created the desired directory.

```
Main Menu
1. Create File
2. Create Directory
3. Remove a file/directory
4. Display File System
5. Exit

Choice: 1
Please enter File name or quit: File1
Please enter Directory name or quit: /james
Please enter access permissions using format rwx or quit: rwx
Please enter size(1-Small, 2-Medium, 3-Large: 3
Created File: /james/File1

Hello World!

Main Menu
1. Create File
2. Create Directory
3. Remove a file/directory
4. Display File System
5. Exit

Choice:
```

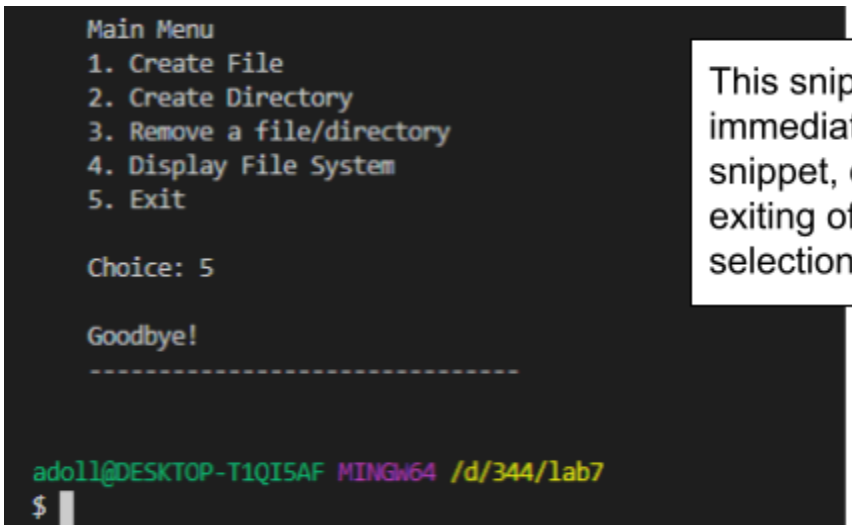
This snippet is immediately following the previous creation of a directory. It allows the user to enter all the required specifications of a file, then creates said file in the given directory and write text to it. If the given directory does not exist, it tells the user so, and prompts the user to redo their file creation.

```
Hello World!

Main Menu
1. Create File
2. Create Directory
3. Remove a file/directory
4. Display File System
5. Exit

Choice: 3
File or Directory?: file
Please enter file name or quit: File1
Please enter parent directory name or quit: /james
 /james/File1 deleted

Main Menu
1. Create File
2. Create Directory
3. Remove a file/directory
4. Display File System
5. Exit

Choice:
```

This snippet, immediately following the previous creation of a file, shows how to delete a file. The system prompts the user if they are deleting a file or an entire directory. Then, prompts for their parent directory, amd located the file, then deletes it.



This code snippet, captured immediately following the previous snippet, is of what occurs when the user prompts for the “display file system” command. It displays all directories and their contents. Here, we deleted the only file we added to the only directory we created, so it simply outputs that directory, “james”



This snippet, taken immediately after the previous snippet, displays successful exiting of the program upon selection of the exit call.

```
Main Menu
1. Create File
2. Create Directory
3. Remove a file/directory
4. Display File System
5. Exit

Choice: 6
ERROR! Try again

Main Menu
1. Create File
2. Create Directory
3. Remove a file/directory
4. Display File System
5. Exit
```

This snippet is a sample of how an incorrect choice is handled. It displays “error” then reroutes the user back to the main menu to see the values for correct input

```
Choice: 4
 /Dir1
     aFile
     bFile
     cFile
     dFile
 /Dir2
     aFile
     bFile
     eFile
 /Dir3
     eFile
     gFile
     zFile

Main Menu
1. Create File
2. Create Directory
3. Remove a file/directory
4. Display File System
5. Exit
```

This snippet is provided as per the specifications of the lab report, with at least three directories of three files each having been created, then the “Display File System” is called, which displays all of the created directories and files.

Conclusion

Overall, this assignment went extremely well, however, it took time. The hardest part for me was getting used to python syntax, as I have not used Python previously. To tackle this, it required some Google searches to figure out how to do some of the things I wanted and needed to do for this implementation, such as accessing a list inside of an object that is inside of a list. Things like that was when I ran into issues, as even though they are similar, when tackling Python rather than C(++), you kind of have to think in a different way, as it is simply syntactically different. One final difficulty I had was in regards to the output. Due to the way you print lines with python, I could not get everything flush on the left of the screen, so I had to indent all outputs by a specific amount. This was difficult to do, but resulted in an actual easier to read final product. Overall, this lab assignment really helped me improve upon my overall programming knowledge, and gave me a great crash course into python.