Assignment 1

The purpose of this first assignment is to introduce you gently to some simple UNIX commands and to motivate you to login to your account. It should not take long to do this. Everything that you need to know is contained in Lessons 1, 2, and 3.

In the directory named cs132 (whose absolute path is /data/biocs/b/student.accounts/cs132), there is a directory named projects, and within the projects directory there is a directory named project1. The project1 directory has been configured in such a way that each student in the class has the ability to create files within it and delete their own files, but not those created by other students.

Your objective in this assignment is to complete each of the following tasks.

1. Within the project1 subdirectory, create a directory whose name is your own user name. For example, I would create one named sweiss.

2. There is a command in Unix that displays a calendar. It is named cal. The cal command can display the current month, or three months, or the whole year. Read the man page for it, and figure out how to display a full year. Then use the cal command with the appropriate option and use I/O redirection to create a file in the directory you created in step 1 named 2014_calendar that contains the calendar for 2014.

3. The cs132 directory also contains a subdirectory named data. Within that is a subdirectory named misc_files. And sitting there is a file named directors. Read the man page for a command named head. Create a second file in the directory you created in step 1 that contains the first 10 lines of that directors file (using the I/O redirection operator.) Name that file my_directors.

4. Type the date command and put its output into a file named time_I_finished.

That is all you have to do in this assignment. It should not be hard. The point is to get you thinking and working. There should be three files inside your directory. Make sure you name your files exactly as I specified above, with your correct user name, in order to get full credit for your work.

Refresher About I/O Redirection

Normally, when you type a command in UNIX, it will display its output on the screen. To be precise, it displays its output in your terminal window. For example, if I type the command who, it will display the list of currently logged-in users in my terminal window. UNIX, like DOS, lets you instead send the output of a command to a file, which is called redirecting output. To send the output of a command into a file you use the greater-than symbol “>” followed by the file’s name. For example, if I type the command

```
echo "A mathematician is a device for turning coffee into theorems." > erdos_quote
```

and there does not yet exist a file named erdos_quote, then that file will be created in the directory in which I am working, and it will contain the sentence

```
A mathematician is a device for turning coffee into theorems.
```
The double-quote marks surrounding the sentence are not placed in the file but they must enclose the text that I want to send to the file if that text has any spaces in it.

In general, if I type any command followed by “> filename” where filename is the name of a new file, filename will be created and will contain the output of that command. So

```bash
ls -l $HOME > whats_in_home
```

will create a file named `whats_in_home` containing a long listing of the contents of my home directory.

**Other Tips**

If you make a mistake and need to delete a file or a directory, this is what you need to know:

To delete a file named `foo`, type

```bash
rm foo
```

This command might ask you if you are sure you want to delete it. Answer ‘y’ for yes or ‘n’ for no.

To delete a non-empty directory named `dir`, type

```bash
rmdir dir
```

If `dir` is not empty, it will not delete it. You must empty it first.