

## BIOC8145 Assignment #1 Answers

### 1. UNIX tutorial exercises (25 pts):

- a. Exercise 1A: mkdir backups
- b. Exercise 1B: Explore UNIX file system using ls, pwd, cd
- c. Exercise 2A: cp science.txt science.bak
- d. Exercise 2B: mkdir tempstuff  
rmdir tempstuff
- e. Exercise 3A: cat > list2  
orange plum mango grapefruit ^D  
cat list2
- f. Exercise 3B: cat list1 list2 | grep p | sort > sorted\_plist.txt  
enscript -p sorted\_plist.ps sorted\_plist.txt ...or...  
cat list1 list2 | grep p | sort | enscript -p sorted\_plist.ps
- g. Exercise 5A: ls -l  
drwxr-xr-x 3 sb3de staff 96 Mar 25 10:58 backups  
-rw-r--r--@ 1 sb3de staff 7767 Mar 25 10:43 science.txt  
chmod g+w science.txt (give "write" permissions to "group")  
chmod g+w backups (give "write" permission to "group")  
chmod o-x backups (remove "execute" permission to "other")  
ls -l  
drwxrwxr-- 3 sb3de staff 96 Mar 25 10:58 backups  
-rw-rw-r--@ 1 sb3de staff 7767 Mar 25 10:43 science.txt

### 2. Answer the following additional questions (60 pts):

- a. What are the three parts that make up UNIX? The kernel, the shell and the programs.
- b. Briefly describe the functionality of these three parts. The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the filestore and communications in response to system calls. The shell acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI). It interprets the commands the user types in and arranges for them to be carried out. The commands are themselves programs: when they terminate, the shell gives the user another prompt (% on our systems). UNIX contains a set of commands or programs that allow a wide array of filesystem management operations.
- c. How is the UNIX file-system/set of directories structured? The file-system is arranged in a hierarchical structure, like an inverted tree. The top of the hierarchy is traditionally called root.
- d. From the unixstuff/backups directory, list three different commands that return you to the home directly. cd cd ~/ cd ../..
- e. Does the output from commands "grep -ivc and science.txt" and "grep -ic and science.txt" add up to the output of "wc -l science.txt"? Yes. Why or why not? Every sentence in science.txt either contains "and" (whose count grep -ic outputs) or does not contain "and" (whose count grep -ivc outputs).
- f. In Tutorial 2 Summary, they mistakenly replaced a command that they introduced in the body of the tutorial (i.e., they introduced a command in the body of the tutorial that's missing in the summary and included a command in the summary that's missing in the body of tutorial). What command in the body of the tutorial was replaced? less What command was added in the summary? more Do they do the same thing? They are very similar. Is there any difference in what they do (if so, briefly explain)? Yes, but for our purposes they are very similar. What's interesting about this mistake in this tutorial is that only someone who "thinks in UNIX" would

make a mistake like this. BTW, “more” has been updated. It used to be that you couldn’t scroll up!

- g. Briefly define “standard input”, “standard output” and “standard error”. Standard output is the output of UNIX commands that are written to the terminal of the screen. Standard input is read from the keyboard. Standard error is where processes/programs write their error messages (by default) to the terminal of the screen.
- h. Use pipes and grep to count how many lines from the command “apropos copy” contain the word “copy”. `apropos copy | grep -w copy | wc -l` gives me 107 lines. `apropos copy | grep copy | wc -l` gives me 168 lines. Both address the question.
- i. In your unixstuff directory, create a new file, “biglist2” by executing the command “paste list1 list2 > biglist2”. Use the “cat” command to view the contents of biglist2. What’s in the file? “biglist2” contains two “fields” or columns if you will that are tab separated. Now, execute the commands “cut -f1 biglist2” and “cut -f2 biglist2”. What output does each of these commands produce? The first command (cut -f1) outputs (or cuts out) the first field or column from “biglist2”. The second command (cut -f2) outputs (or cuts out) the second field or column from “biglist2”.
- j. Execute the following set of commands and briefly explain the output: “cat list1 list1”; This outputs the contents of list1 twice (one at a time). “cat list1 list1 | sort”; This outputs a sorted list with repeated words appearing next to each other. “cat list1 list1 | sort | uniq”. This outputs a sorted list with only one copy of a repeated word. What is the output if you skip “sort” in the last command (i.e., “cat list1 list1 | uniq”)? This produces the same output as cat list1 list1 so the pipe to uniq does nothing here.
- k. Type “echo \$PATH”. Briefly describe the output in terms of the functionality of this environment variable. The output is a set of directories separated by a colon “:” in which the shell looks for the command you typed where the command is in one of these directories.
- l. Assume you had the program “units” in your home directory “~/units174/bin”, and you executed the command “export PATH=\$PATH:\$HOME/units174/bin”. What would this enable you to do? This would allow you to type and execute the command “units” from any directory. Importantly, it would not have to be in your ~/units174/bin directory nor would you have to give a full path followed by the command as in ~/units174/bin/units.

3. Answer the following questions (15 pts):

- a. What is the default starting directory of a job? They let you know in two places on the site: (1) “In our installation of SLURM, the default starting directory is the directory from which the batch job was submitted”; (2) By default SLURM changes to the directory from which the job was submitted”. Why is this important to know? Any command or program that you specify that is not in your search path will not get executed unless you submit the job from the same directory that contains that command or program. In the example SLURM job command file to run a serial R batch job (near the bottom of the page; assume it’s named myRprog.slurm), which line of code would this effect? The last line. While Rscript would run because you put R in your search path (in principle; see b. below) by executing “module load R”, the system would not find your program “myRprog.R” unless you submitted the job from the same directory that contains “myRprog.R”. What script/program (shown in the command file) should you have in the directory from which you executed the command “sbatch myRprog.slurm”? myRprog.R.
- b. Will running sbatch myRprog.slurm work? No! Hint: execute the command “module load R”. What happens when you do this? You get an error: “Lmod has detected the following error...”. Second hint: go to the R and RStudio on Rivanna page: <https://www.rc.virginia.edu/userinfo/rivanna/software/r/> and look at the instructions “Loading the R module”. Execute the command “module load gcc R”. This command works! This properly loads R into your search path. Then type “R”. You can type “q()” to exit. What

happened when you typed “R”? You’re in the R environment! You can execute R commands. It’s not as nice as working in RStudio though.

- c. Write a script that would submit a single-core job to the cluster with the following options: 1 task, wallclock time no longer than 1 hr, output results to “my\_analysis.out”, standard partition request, charge account bioc8145 and execute the (non-existent) R program myRprogram.R. Assume you named this file/script R\_job.slurm. How would you submit it and run your job? Don’t actually execute this command. We’ll do that shortly.

```
#!/bin/bash
```

```
#SBATCH -n 1
```

```
#SBATCH -t 01:00:00
```

```
#SBATCH -o my_analysis.out
```

```
#SBATCH -p standard
```

```
#SBATCH -A bioc8145
```

```
module load gcc R
```

```
Rscript myRprogram.R
```

```
sbatch R_job.slurm (This command would submit the job).
```