

CS 197U: Lab #4

Arun Dunna
adunna@cs.umass.edu

Revision 1.0 - February 17, 2019

NOTE: Submit your question responses to Moodle as a single PDF file.

**If a question asks for command(s), give the complete command with programs, options, arguments, pipes, redirects, etc.
ex: “vim file.txt” instead of “vim”**

The PDF should be named in the format: *lastname_firstname_lab4.pdf*

Introduction

This Lab will cover system administration, more user and group management, cron jobs, and basic shell scripting.

You should experiment with installing software via Make when you find the opportunity. You will need to install software via Make at some point in the future, if you haven't already, to your Unix-like system.

Some commands will require sudo in this lab. If a command does not require sudo, and you include sudo in your answer, you will only receive partial credit as you should only use it when necessary.

1 Local Work

For this section, you should work from your own local system. That is, you will not SSH into the course server until the next section.

Question 1

Make a few files (at least 3) in your home directory - you can put content in them if you'd like, but that is optional.

Change the owner of each of the files to the root user.

Q: What full command did you use to change the owner?

Question 2

Now, change the owner of each of the files back to your own user, which if you're on the VM will be student.

Q: What full command did you use to change the owner?

Question 3

Make a new directory named *grpdir* and move each of the files into the directory.

Change the group for the new directory and all files inside to the root group using one command (no pipes/redirects or anything else).

Q: What full command did you use to change the group?

Question 4

Q: List out 5 of the system's users with their respective user IDs and shells.

Question 5

Create a new user named *testuser* and a new group named *testgroup*.

Q: What commands did you use to do this?

Question 6

Change the permissions of the *grpdir* directory to 470.

Change the group of the directory and all files inside it to *testgroup*. Then, try to write a file inside the directory.

Q: What command did you use to change the group? What happened when you tried to write a file to the directory? Why is this?

Question 7

Add your own user, which if you're on the VM will be *student*, to the *testgroup* group you just created.

Q: What command did you use to do this?

Question 8

Now try again to write a file to the *grpdir* directory.

Q: What happened this time? Why?

Question 9

Change the password for *testuser* to whatever you'd like, and in a separate command, add the user to the testgroup you created.

Q: What commands did you use?

Question 10

Edit your cron file to write the datetime to the file *~/dateoutput* on reboot.

Q: What command did you use to do this? What line did you add to the cron file?

Question 11

Edit your cron file to write the datetime to the file */dateoutput* every minute. Ensure you don't use *sudo* or similar.

Q: After a few minutes check for the file. What do you find? Why?

When you're done, remove the cron line you just added so that it won't write every minute.

2 Remote Work

Connect to the course server to do the work in this section. To do so, we want you to use the *ssh* command. The hostname of the server is **197u.adunna.me**.

All of this should be done in your home directory (i.e. if you're asked to create/move/modify/remove files, it should all be done in your home directory):

Question 12

Create the *~/lab4* directory and go into it.

Write a shell script to *~/lab4/utctime.sh* that writes the current datetime in UTC to *~/lab4/currutc.txt* whenever you run it. Test the script and ensure it works.

Q: What are the contents of your script file? What did you have to change about it before running it?

Question 13

Write a shell script to `~/lab4/username.sh` that prompts the user for their First and Last Names. It should then output the user's full name, along with the username of the current user. These should all be on separate lines in the output.

Test the script and ensure it works.

Example of output:

```
adunna@cs197u:~$ ./username.sh
Enter your first name: Arun
Enter your last name: Dunna
Your full name is Arun Dunna, and your username is adunna!
```

Q: What are the contents of your script file? What output do you get when you test it?

Question 14

Write a shell script to `~/lab4/add3var.sh` that uses 3 number variables that are preset (use any three numbers you like), and then outputs the addition of the 3 numbers.

Q: What are the contents of your script file? What output do you get when you test it?

Question 15

Write a shell script to `~/lab4/add3prompt.sh` that prompts the user for 3 numbers and then outputs the addition of the 3 numbers.

Q: What are the contents of your script file? What output do you get when you test it?

Question 16

We can use the `shuf` program to generate random numbers. For example, `shuf -i 1-10 -n 1` will generate one number in the range of 1 to 10.

Write a shell script to `~/lab4/add3rand.sh` that generates 3 random numbers, in the range of 1 to 1000, and then outputs the addition of the 3 numbers. It should also output each of the individual numbers that it generated.

Q: What are the contents of your script file? What output do you get when you test it a few times?