

---

# CSCE 215 – Unix/Linux Fundamentals

## Fall 2015 – Assignment 1

### 10 Points

---

**Assigned** August 24, 2015  
**Due:** August 30, 2015 (5:00 pm)

**Read Chapters 1 and 2, of “Your UNIX”.** If you do not have the book, do not fret, the internet is abound with many Unix tutorials. If you are unable to type `google.com:unix tutorials` into the address bar of a web browser, I recommend the following resource [www.ee.surrey.ac.uk/Teaching/Unix/](http://www.ee.surrey.ac.uk/Teaching/Unix/). A link to this resource has been provided on the course website.

To log on remotely, you must ssh to one of our linux/unix workstations. A list is available on our department website ([www.cse.sc.edu](http://www.cse.sc.edu)).

1. “Login” using your College of Engineering and Computing userid & password, then
2. Under “Computer Resources” select “Computer Labs” and finally select “Linux Workstations”.

You can get ssh.com’s Win32 ssh client from the following address  
<https://www.cse.sc.edu/~pokeefe/unix-linux/resources/SSHSecureShellClient-3.2.9.exe>

**NOTE:** Use port 222 when connecting via ssh to one of our linux workstations. The default port (22) has been disabled on our machines.

## CAREFULLY READ THE DIRECTIONS.

This assignment is not designed to trick you or require an inordinate amount of learning on your part, it is supposed to introduce you to the linux environment and ensure that you can submit your assignments to our departmental dropbox.

### Part 1 – Can you create a .txt file?

Compose a text file using VI (**text** implies a .txt extension not a **MS Word** .doc or .docx extension) that lists the following information: (Use the vi text editor available via command line)

1. Your name.
2. Your university email address ([@email.sc.edu](mailto:@email.sc.edu)). You should setup a forwarding rule to your regular email if you do not intend to use this email address outside of class.
3. Other contact information.
4. Enrollment info: What department are you in? Are you enrolled or auditing? How long have you been at USC-Columbia? Are you transferring in from another institution?
5. Programming experience (know languages, years experience).
6. Unix experience (if any).
7. Name of the text editor you are using to edit this file.
8. Name the file [your userid].info.txt (Example: [baymax.info.txt](#)).

---

## Part 2 - Can you Unix?

In this part of the assignment you will execute some unix commands.

```
script [your userid].part2.txt
whoami

printf "$USER\n"
echo $USER
whoami

echo $HOSTNAME

echo $(date +%m-%d-%Y)
echo $(date +%r)
exit
```

A short explanation of the above commands

**script** – this command essentially dumps all of the terminal output to a filename

**whoami** – prints the effective userid

**echo** – display message on screen, writes each given string to standard output

**\$USER** – a built in environment variable for the current user

**printf** – prints to the specified file, standard output by default

**\$HOSTNAME** – a built in environment variable that stores the system's hostname

**\$(date +%m-%d-%Y)** – the current month, day, and year

**\$(date +%r)** – the current time

**exit** – stops the script command and writes the output to the specified file

## Part 3 - Scripting?

I have provided a script, the contents of which appear on the next page, to introduce you to scripting. A *script* is a piece of code that is written to execute a repeatable task that has some exploitable structure.

For example, I know that I will be required to do a lab everyday. I also know that Mr. O'Keefe wants our userid to be at the top of all deliverables. What's more, my code needs to be able to run on one of the departmental unix machines, so why not make his job easier and my grade safer by providing the machine name in my output.

---

```
#!/bin/sh
#
# Test script - written by Nick Stiffler
# Entry level script to be used by csce215 students.

if [ $# -eq 1 ]; then
    STUDENTNAME=$USER
    ASSIGNMENT=$1
else
    echo "usage: 'basename $0' [assignment]"
    exit 1
fi

if [ ! -d "$ASSIGNMENT" ]; then
    # Control will enter here if $ASSIGNMENT doesn't exist.
    mkdir $ASSIGNMENT
fi

# Move into the directory
cd $ASSIGNMENT

# Variable for the filename
FILENAME=$USER\_ $ASSIGNMENT.txt

# Variable for the computer name
HOST=$(hostname)
# Question: Why can't we just use $HOSTNAME like we did in Part 2?
# Answer: Because we are in a shell script and not a bash script.

# Variables for the date and time
DATE=$(date +"%m-%d-%Y")
TIME=$(date +"%r")

if [ -f "$FILENAME" ]; then
    # Control will enter here if the $FILENAME exists.
    sed -i "2s/.*/$HOST/g" "$FILENAME"
    sed -i "3s/.*/$DATE - $TIME/g" "$FILENAME"
else
    # Control will enter here if the $FILENAME does not exist.
    printf "%s\n" "$USER" > $FILENAME
    printf "%s\n" "$HOST" >> $FILENAME
    printf "%s" "$DATE" >> $FILENAME
    printf " - %s\n" "$TIME" >> $FILENAME
fi
```

---

I think the easiest way to understand the code in the provided script is to trace what happens when the following command is issued.

```
./create_assignment assignment1
```

If you get an error saying

```
bash: ./create_assignment: no such file or directory
```

Then you have a permission error. To make the code executable (which is what we want because we want to run the code) we need to issue the following command:

```
chmod +x create_assignment
```

- The first “if-else” block tests to make sure that there is one command line argument after the script name. If the user does not supply the correct number of arguments, then an error message is printed.
- The code then looks for a directory that matches the user supplied input. If there is no directory that corresponds to this name, then a new directory is created. We then move into that directory.
- We then create a variable to hold the filename, **If I were to run this example** that would translate to  
`p_assignment1.txt`
- We then create variables for the hostname and date/time.
- The next “if-else” block does the bulk of the work. If the file does not exist, we will enter the **else** part of the code. The code does the following:
  - The first printf statement substitutes the literal of \$USER into where %s appears. The “\n” indicates that we want a newline at the end. The “>” will create/overwrite the specified filename.
  - The second printf statement substitutes the literal of \$HOST into where %s appears. Notice that this time we are using “> >” which means we would like to create/append to a specified file.
  - The third printf statement substitutes the literal of \$DATE into where %s appears. Notice that we do not have a “\n” this time, so subsequent prints will be on the same line.
  - The last printf statement substitutes the literal of \$TIME into where %s appears.

If the file already exists, the sed commands will update the file in the following ways:

- The first sed command replaces anything that exists on the second line with the \$HOST.
- The second sed command replaces anything that exists on the third line with the current date and time.

**You are not expected to know everything in this script (YET).** I thought that this would be an excellent example to demonstrate some of the useful unix tools at your disposal.

## Submission

You should now have three .txt files, one for each part of the assignment. **ASSUMING** that all three files are in the same directory, you are one command away from submitting this assignment. (hint: ls -l)

Run the following command:

```
cat [userid].info.txt [userid].part2.txt >> [userid]_assignment1.txt
```

[userid]\_assignment1.txt should now contain all three parts of the assignment.

Once you have verified that this is the case, submit this file by logging in to the CSE Dropbox website ([www.dropbox.cse.sc.edu](http://www.dropbox.cse.sc.edu)). Use your CEC/“Windows” userid:password. Click on the link for this course-section. And then click on the link for Assignment 1.