

# OC - Fall 2018 (IIIT Sri City)

## Practice Assignment 3

---

1. List the files in the directory `/bin` that end in `.sh`. What command did you use?
2. Write the command to list the files in the current directory that begin with upper case letters?
3. Copy all files in the current directory whose names end in `.c` or `.h` into the subdirectory `Programs`. What command did you use?
4. Write UNIX command to get all files with `r`, `g` or `i` on the 3rd place.
5. Copy all files in the current directory whose names contain the character strings `notes` or `misc` into the subdirectory `Misc`. What commands did you use? How to do this using one line of command?
6. Copy all files which begin with `copy.me` into the `UStoreIt` subdirectory. Move all files which begin with `move.me` into the `UStoreIt` subdirectory. What commands did you use?
7. Delete all files which contain the sequence `del`. What command did you use?
8. What will be the output of the following command sequences:  
a) ``echo x=4`; echo $x`                      b) `eval `echo x=4`; echo $x`
9. Create a variable `Age` and assign some values. Open a new subshell by typing `bash`. Try to run the command `echo $Age`. Are you able to access the value of `Age` assigned by parent shell? If not, how to access it?
10. What will be the effect over current directory of parent shell if sub-shell is changing the current directory? Test with example.
11. Write the command needed for the below mentioned actions:

Action	Command Needed
A. Change to your home directory	
B. Make a directory named <code>public_html</code>	
C. Allow group and others to be able to read and execute on your home directory	
D. Allow group and others to be able to read and execute on the <code>public_html</code> directory	

Action	Command Needed
E. Verify the permissions on your home directory and on public_html	
F. Use touch to create an empty file named index.html in the public_html directory	
G. Allow group and others to be able to read all files in the public_html directory	
H. Verify the permissions of the file(s) in public_html (your home page files)	

## 12. Regular Expression examples using "grep"

Create a directory called **reg\_exp**, then create a file **file1** to the new directory as follows:

```
cat > file1
```

```
line 1: Rather than teach your gadgets what to do,
line 2: Intel researchers say that in the not-too-distant future
line 3: they will learn about you on their own.
line 4: That means where you are, how you're feeling, and what you want.
line 5: It's actually not as creepy as it sounds.
line 6: Intel Chief Technology Officer and Director of Intel Labs Justin Rattner
line 7: took the stage Wednesday at the annual Intel Developer Forum
line 8: here to talk about the future of "context-aware computing,"
line 9: what Intel is doing about it, and how gadgets
line 10: can make life easier for their owners,
line 11: but in a way that the owners can control.
line 12: Context-aware computing is Intel's term for devices
line 13: that anticipate what people need or want and guide them accordingly.
line 14: The context is gathered through a combination of
line 15: "hard sensors"--cameras that detect movement and
line 16: GPS-based location information--and "soft sensors"—
line 17: such as calendar information or pieces of data you input into a device.
```

### Note:

**Use single or double quotes (remember rules regarding quoting special characters) when specifying regular expressions with the grep utility.**

Before issuing the following commands, **cat file1** to see output. Try to guess what these separate commands will do ahead of time, then run the commands to check your guess.

If any command does not work, try to use egrep (or) grep -E.

```
grep '^T' file1
grep '^t' file1
grep '^G' file1
grep -i '^d' file1
```

```

grep '?$' file1
grep '$' file1
grep '#' file1
grep '\<the\>' file1
grep '.not' file1
grep '[?!]' file1
grep '$[?!]' file1
grep '[?!]$\n' file1
grep '^[^@!#%.A-Za-z]' file1
grep -i 'in a' file1
grep 'e{2}' file1

```

13. Create a big file "big.file". Check and note the size of the file "big.file" (hint use "ls -s"). Reduce it with **gzip** and check the size of "bigfile.gz". Notice the size of zipped version? Restore the file with **gunzip**.
14. Use tar to zip multiple files into one and then unzip it.
15. Create a pipeline one step at a time, you will add one command to the pipeline in each step, and notice how the output changes.
  - a) Begin with a single command, "ps", with the options for a full listing of every process: **ps -fe**
  - b) Use command line editing to add more, to display the output one screenful at a time. You are instructing the shell to take the standard output of ps and feed it into the standard input of more: **ps -fe | more**
  - c) Add grep root to filter for the string root. The standard output of ps goes into the standard input of grep, The standard output of grep goes into the standard input of more: **ps -fe | grep "\broot\b" | more**
  - d) Add sort so the output gets sorted: **ps -fe | grep "\broot\b" | sort | more**
16. Create a directory called "ITWS1" by issuing the command **mkdir ITWS1**. Change to your ITWS1 directory by issuing the command **cd ITWS1**. Create a directory called **UNIX**, and change to that directory.
  - a) Issue a UNIX command (not an editor!) to create a file called **stuff** that contains the following lines (include the mistakes!!)
 

```

Tha cow is mad
I tell you!
I think the cow is looking at me
Terrible news!
I think the cow is looking at me
          
```
  - b) Issue a UNIX command to verify the contents of the file **stuff**
  - c) Issue a UNIX command to determine the file type of **stuff**
  - d) Try to issue a series of UNIX commands to take the contents of **stuff** and modify the content (to be saved in the file **stuff.new** ) as shown below:
 

```

I tell you!
I think the cow is looking at me
Terrible news!
          
```

**Here is the challenge:**

**1 - you are not permitted to use an editor!**

**2 - You have to issue those commands on just one line!**

- e) Issue a UNIX command to verify the contents of the file **stuff.new**
- f) Issue a UNIX command to change the name of the file **stuff.new** to **stuff**, but make sure that the system makes the user confirm overwriting the existing file and select "**yes**".
- g) Issue a UNIX command to take the contents of the modified file **stuff** and place into a file called **cows** lines that only contain the pattern "cow"
- h) Issue a UNIX command to take the contents of the modified file **stuff** and place into a file called **not\_cows** all lines that do not contain the pattern "cow". (Hint: use the online manual to check for an appropriate option).
- i) Issue a UNIX command to compare the file **cows** with the file **not\_cows**
- j) Issue a UNIX command to compare the file **stuff** with the file **cows**. According to the results, what has to be done to the file **stuff** to make it identical to **cows**?

17. Create a directory called "file\_practice":

**mkdir -p ~/ITWS1/file\_practice**

Use the cd and touch commands to create several empty files:

**cd ~/ITWS1/file\_practice**

**touch a1 a11 a12 a23 a123 a1234 a543 a321 bab2 cab5 12abc12 1a1 abc1  
abcdef1 1 12 123**

Try to guess the output from each of the following commands before issuing them on the computer. After you have written the answers on paper, issue the command to check your work!

**ls**

**ls \***

**ls \*1\***

**ls ?**

**ls ??**

**file ??**

**ls ???**

**ls a[1]**

**ls a[123456789]**

**file a[123456789]**

**ls a[123456789]\***

**ls [a][1][2]\***

**file [a][1][2]\***

**ls [a-z]\*3**

**file [a-z]\*3**

**ls a\* [1]**

**ls a?\*[1]**

**ls \*[!1]** (Note: when issuing this command, notice that filenames do not end in 1)

18. open vi

\$ vi

issue the following command

:r! ls -la /usr/bin --> this is to generate relatively large text

Write the commands to do the following tasks in same sequence:

- a. jump to the first line
- b. scroll down one page
- c. delete 5<sup>th</sup> character from line 60
- d. add new line after line 60 and input text "previous changed"
- e. save the text to file "tmp"
- f. add "WARNING: " at the beginning of the newly added line
- g. move word by word to "changed"
- h. replace "changed" with "modified" in the newly added line
- i. find "kill" and delete 5 lines after the line containing it
- j. go to the last line
- k. delete characters 5 to 20 of the last line
- l. find "bash"
- m. replace all "sh" with "SH"
- n. copy all lines from between 50 and 60 into the paste buffer
- o. paste the content of paste buffer to the end of the file
- p. delete the last 10 lines of the file
- q. save and quit
- r. open vi with "tmp"
- s. change all "baSH" in "tmp" to "bash"
- t. save file