A look at Text Editors

- ***Command***
- vi myfile
- wq! - for saving the file in vi editor
- q! - for just saving the file without update.
- ctrl + u = will give us undo to the file
- :w! - saves the file.
- nano myfile = you will be in the insert mode automatically and then we can type ctrl+k = to delete all the lines. Then type your own data and then press ctrl + X for saving the data.

Basic file manipulation command

- ***Come to user directory called aamir.***
- pwd
- mkdir mydirectory
- ls
- ls -al => will give us whole directory working structure ==> -a : do not ignore files starting with . ==> -l : printthe author of each file.
- man ls ==> will give us the details of that file
- using the tab command we can go and proceed with directory we are looking for.
- for changing the name of a file we can type **mv <oldfile> <newfile> .**
- ls
- tail <newfile> ==> We will see the content same as the oldfile has it.
- head <newfile> ==> same command to look the contents of file.
- cat <newfile> ==> we will look the contents of file.
- rm <file> ==> will remove the file and there is no way again retrieve it.
- cd ==> will bring us to the user directory.
- cd ~/ ==> this will bring us to the user directory too.
- mv <oldirectory> <newdirectory> ==> will change the name of existing directory to new directory.
- if directory is empty and you want to renmove it than just type **rmdir <directory name>.**
- if directory is not empty and there is a file inside it , so use command like **rm -r <directory name>** . -r ==> is used for recursive.
- In the directory if you want to copy one file to another file we use **cp <oldfile> <newfile>.**
- For directory to copy we have to use the command like **cp -r <old directory> <nwedirectory> . Please use -r for directory.**
- To forcibly remove the directory from working ==> **rm -rf <directory name>.**

Command Line keyboard shortcuts

- ctrl + a ==> will take you to the begining of the line if you need to type in the comand line.
- ctrl + e ==> will take you to the end of the line.
- ctrl + k ==> will delete from the line where your cursor is prompting.
- ctrl + l ==> will take you from the middle to the first of the command line.
- ctrl + u ==> will take from the cursor to delete before the line.
- ctrl + w ==> will delete the particular word where it starts from.
- ctrl + t == > it switch places of word from where it cursor prompt.
- ctrl + f ==> move forward
- ctrl + b == > move back
- ctrl + h == > will remove the one word back
- ctrl + d ==> 1 character delete at a time.
- ctr l + p ==> will take us to the previous command.

Aliases

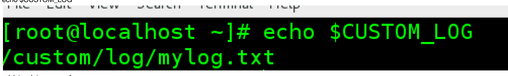
- cd /var/log - this is used in moreover production environment.
- ls
- there is a messages folder inside it.
- tail messages - This will show us the 10 default messages.
- tail -n 20 messages - This will show us a 20 line of messages.
- head messages - same as tail messages command
- head -n 20 messages - same as 20 line of messages.
- clear
- tail -f messages - This is widely used by author in his production environment and this command give us the appended messages in the loop.
- echo "hello world" >> messages
- now again type tail -f messages == > you will see the hello world in a production environment messages.
- alias showlog="tail -f /var/log/messages"
- When you come out of the shell command , your alias does not work.

Bash Shell Configuration File and Shell Variables

- alias showlog="tail -f messages"
- Now when you switch to the other user it will go it means show log will not work
- export MYNAME="Aamir"
- echo $MYNAME - it will display the Aamir
- put this in Vi .bashrc file and then switch user , you will see your name if you command echo $MYNAME
- # env
- # echo $LOGNAME

Setting & Unsetting Environment Variables

- mkdir -p /custom/log
- export CUSTOM_LOG = "/custom/log/mylog.txt"
- echo "My log Message" > $CUSTOM_LOG
- cat my log.txt
- To remove the environment variables we need to use command like **unset CUSTOM_LOG**
- echo $CUSTOM_LOG - No file will be seen
- We can set vi .bashrc file , our export CUSTOM_LOG and save it.
- then change the user .
- echo $CUSTOM_LOG



- cd /etc - let see now for every user.
- ls | grep profile
- cd profile.d
- ls
- vi custom.sh ==> export CUSTOM_LOG = "/custom/log/mylog.txt
- ls ==>. This will set for every user , when anyone log in it.

Customizing the Bash Prompt

- **set | grep PS - PS is a default in the command.**
- **This is not used much it is just for customizing and play with the terminal command and how the Bash is seen real time.**
- **export PS1='\t' ->> this will change prompt to in time.**

Command Line History

- whoami
- ls -a
- cat .bash_history
- history | grep clear
- Ctrl + r ==> and then press w , will give us the output which command we have use of w.
- history -c ==> will clear the history.
- man history

Create a simple Bash Script to Automate commands

- It was the introduction to bash scripting , so we can automate the process , to not to type command and command .

Finding files in Linux

Which and Whereis

—> Which and Whereis are tools that help you locate man pages and the binary location for a specific program. These are useful tools if you need the full path to a program.

Command

- **which showlog**
- **which cp**
- **whereis cp —> These commands give us the binary file for the particular command.**

Finding Files with Locate

—> Locate is an extremely useful tool. It allows us to search a pre-populated database and serach for files and their locations by name. However , it does require the database to be updated by a cron job or manually by using the updatedb command. This lesson will talk about these concept.

- which locate
- Cl
- locate pwd
- cd
- touch hello
- ls -al :- This command will not show the hello file in it we have to update the database.
- locate hello
- pwd
- **updatedb - will update the hello file in root directory**
- **locate hello - You will able to see your file in root directory.**
- locate kernel - You will able to see the kernel file associated with it.
- locate user - You will see user file associated with it.
- locate /home - You will see the file which are locate in /home.
- locate kernel | grep /usr - The file which are associated with kernel of /usr will come out.

The Powerful Find Command

—> We are going to talk about the find command. it is an extremely powerful tool for searching files and applying "action" against the return result set. This command is a must know for anybody using Linux.

Note: please work in the root directory and go in the cd /etc

- find . -name ‘cron*’ —> Will give us the name of a file inside the cron
- find . -type f -name ‘cron*’ —> Will give us the file from the cron
- find . -type d -name ‘cron*’ —> Will give us the directory from the cron.
- Now we will work on the command file called **cd /home.**
- **ls**
- **cd aamir**
- **create a 3 files by using touch command - touch file1 , touch file2 and touch file3.**
- **chmod 777 file1**
- **chmod 555 file2**
- **cd .. —> come out of aamir directory**
- **find /home -perm 777**
- **find /home -perm 555**
- **Now if you want to change the file permission of 777 to 555 use the command like find /home -perm 777 -exec chmod 555 {} \;**
- **find /home -perm 555 - You will see 2 files ; file 1 and file2.**
- **find / -mtime +1 ==> Will give me modified file of 1 day.**
- **find . -size 512c ==> find inside in the home directory of bytes 512.**
- **find / -size 1MB ==> find inside of each file which are of 1 Megabyte.**
- **Come to the user directory now for aamir.**
- **cd aamir**
- **ls**
- **find . -name 'file*'**
- **find . -name 'file*' -exec rm {} \; ==> will remove the file**

Note : Lab : Done .

Use Streams, Pipes,Redirects,Grep and Cut!

Wc,split,cat and diff commands

- *We are going to use these commands to introduce streams and redirects in Linux. By themselves , these commands themselves are powerful Linux tools to help you manipulate files.*
- cat file1 file2
- cat file*
- cat file* | wc -l ==> total number of line in all the file will be displayed.
- split -l 2 file1 —> will split the file1 in two different file.
- split -l 1 file1 —> will split the file1 in 4 different file.
- diff <file1> <file2> —> will display the difference between the two files.

Streams(stdin,stdout,stderr) and Redirects

- In these lesson we are going to learn how to append standard output and standard error to designated location. We will learn the significance of /dev/null and how we can use it. We will also learn how to use redirects to append information to files or overwrite existing files.
- echo "my new entry" > <filename> ==> will create the line in new file and with the single > command it will override the before anything written.
- echo "my new world" >> <filename> ==> will append the echo written
- set -o noclobber —> These command will prevent you to overwrite.
- set +o noclobber —> These command will again make you to overwrite.

Pipes

- In these lesson we will learn about Pipes, a powerful tool in Linux which you will use on a daily basis. Pipes allow us to take the output of one command and pipe it into the input of another. It is an extremely easy and powerful concept and this lesson will help teach that concept concept with examples.

Grep,egrep and fgrep

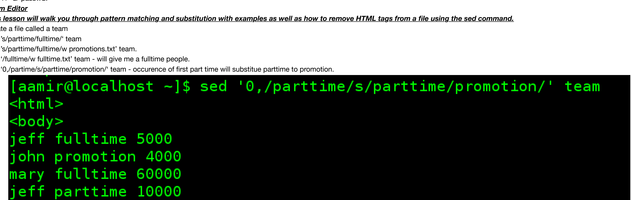
- In these course we will take a look at grep and some of the aditional options with grep. We will get our feet wet with regular expressions and learn when we should use grep,egrep or fgrep.
- Command:
  - grep hello testf - where testf is a folder , and it will grep the hello from the test file.
  - grep ^hello testf - it will display the command which conatin first word hello.
  - grep -c ^hello testf - it will give us a count that how many file will have a word hello in it.
  - grep -c hello$ testf - it will give us a count that which line contain a word at end hello.
  - grep [h] testf - will grep the h word.
  - grep [hjklop] testf - will grep the pattern associated with it.
  - grep [a-g] testf - will grep the word between a and g.
  - grep [ab] testf - will grep the word in between ab.
  - grep [1-9] testf - will grep the word in between and 1 to 9.
  - grep -f grepinput testf - if we give the word in grepinput and if you want to find in the testf.
  - grep -ir cron /etc - This will give us the only the filename which contain cron word in it.
- egrep: extendedgrep expression.
  - egrep 'hello.*world' testf
  - egrep -i 'hello.*world' testf —> will give us a hello world with capital letter too.
  - egrep -i '^hello|world' testf —> This will search for the hello or world associated with it.
  - egrep -v '^hello|world' testf —> will give us the word which does not contain hello or world in the file.
  - egrep -vi '^hello|world' testf
  - egrep 'hello|world' testf | grep -vi Jeff.
- fgrep: it is used for exact match.
  - fgrep hello$ testf.

Cut Command

- ***It is very important tool when it comes to manipulating data from files.***
- ***We are working in /etc directory and then command cat passwd.***
- cut -f1 -d: passwd.
- cut -f1 -d/ passwd.

sed Stream Editor

- ***This lesson will walk you through pattern matching and substitution with examples as well as how to remove HTML tags from a file using the sed command.***
- create a file called a team
- sed 's/parttime/fulltime/' team
- sed 's/parttime/fulltime/w promotions.txt' team.
- sed '/fulltime/w fulltime.txt' team - will give me a fulltime people.
- sed '0,/partime/s/parttime/promotion/' team - occurence of first part time will substitue parttime to promotion.



- Sed 's/<[^>]*>//' team - will remove the html file from my file.

tee command

- ls | tee mynewfile —> it will take whatever data in ls to mynewfile.

Advance Commands and Examples

Using Test/ Tests On the Linux Command Line

- In these lesson we will learn how to use test for testing conditions on file types and comparing values. It is a very powerful tool worth learning.
- test -f file1 && echo "true" — > it will display us a true value because file1 is a file.
- test -w file1 && echo "true" —> It will display a true because file1 is a writable file.
- test 3 -gt 2 && echo "true" || echo "False" —> It will dispaly a true.
- test 1 -gt 2 && echo "true" || echo "False" —> It will display a false.
- [5 -eq 5]; echo$? ==> will display 0.
- [5 -eq 6]; echo$?==> will display 1.
- ["hello" = "hello"]; echo$? —> will display 0.
- ["hello world" = "hello"]; echo$? —> will display 1.

Lab Check for Text on WebSite from the Linux Command Line Using curl and Test

- ***We are going to build a command that uses curl and a condition to check text on a website.If the text matches then it will return true if not it will return false.***

- silent or quite mode don't show progress meter or error messages —> -s : command is used for that.

- **curl -s google.com**
- **curl -s google.com | egrep -ci "301 moved"**
- **curl -s google.com | egrep -ci "301 moved" > /dev/null**
- **curl -s google.com | egrep -ci "301 moved" > /dev/null && echo "file has moved".**

The above command will give us to check whether the 301 has moved or not.

Writing a For Loop on the command line

- ***This will help you learn how to write a for loop on the command line to automate those tasks.***
- ***Command:***
  - for name in 'ls'; do echo $name; done; —> will display everything for in ls.
  - for file in 'ls /etc'; do echo $file;done; —> will display everything for in /etc.