Assignments:

1. List the contents of a directory and their attributes
   * ls
2. Long list the content with file, directory ownership, permissions, sizes, etc…
   * ls -l
3. Display the size of the file in human readable format
   * ls -lh
4. Show all files and folders including hidden one
   * ls -a
5. list directories recursively
   * ls -R
6. Sort the files by size with largest at the top
   * ls -S
7. Sort the files by last time modified displaying the newest first.
   * ls -r
8. Display the location of a program/command, where it is installed.
   * which
9. Which command is used to switch directory from one to another
   * cd
10. List all the environment variables set for the current shell environment
    * printenv
11. Did you notice something in the output of "env" command?
    * Evn[option]..[-][name-value]…[command..[args]…]
    * Set each name to value in the environment and run command.
12. Which command is used to print the text or any variables value in the Console/Terminal?
    * printf
13. Print the value of the env variable "PATH" on the console
    * echo "$PATH"
    * printf "%s\n", $PATH
14. Is linux a case-sensitive operating system?
    * Yes
15. Is, "Ls" same as "ls"?
    * No
16. Display your currently logged in user
    * w
17. how do you change the currrently logged in user to another user?
    * su - user2name
18. Which command is used to leave a shell environment that you are currently logged in to?
    * exit
19. How do you reboot the system?
    * sudo reboot
20. How do you shutdown the system?
    * shutdown
21. Display all the major running processes in the system
    * ps
22. Understand the output of command used in above question of displaying processes, explain the meaning of each column and what data it displays?
    * ps command dispays the status information about processes. The ps command displays process Id (PID), TeleTYpewriter (TTY), processor time used (TIME), and name of the command (COMM)
23. Display the name of the system kernel
    * -o
24. display the kernel release number
    * -r
25. display teh machine type of the current kernel
    * uname -r
26. Display the name of the operating system that the kernel is running on
    * uname
27. Display all info that uname command can show.
    * The command uname display information about the system.
    * Uname -a/uname -s/uname -n/uname -r/ uname -p/uname -m/ uname -
28. Display the name of directory that you are currently pointing to
    * pwd
29. change the current directory to another directory that you have in your system.
    * CWD / cd
30. Go up one directory
    * cd..
31. Return to last directory
    * cd-
32. change the current directory to home(logged in user's) directory
    * cd~ OR cd
33. How to check all the command used from the prompt (Command History)
    * Command [number] history 10
    * $ history
34. In which file the history of commands are stored in?
    * ~/.bash\_history
35. How many lines of history does the system keep and from where you can change it?
36. How can you modify bash's history behaviour
    * To customize bash command history, we will have to make changes in the ~/.bashrc file. To edit the ~/.bashrc file
    * $ nano ~/.bashrc
37. Display all the commands entered so far, now, try to run a particular command from the history list without typing that command.

NOTE:- TAB key is your friend when it comes to command completion and having long file and directory names autocompleted at the bash prompt for you. JUST BE LAZY AND USE TAB KEY FOR AUTO COMPLETION ;-)

* + Use Get-Command to see all the command entered so far

1. What are the different types of shell and where are they used and how do we use them?
   * Different types of Shell are

1. Login Shell

2. Non Login Shell

* + - * Bourne Shell.(sh)
      * C Shell.(csh)
      * TC shell.(tcsh)
      * Korn Shell.(ksh)
      * Bourne Again Shell.(bash)

Shell Scripting is basically used to avoid repetitive work.

1. What is the difference between login shell and non-login shell?
2. How do we start login shell and non-login shell?
   * The login shell is started after successfully login the /bin/login file, by reading the /etc/password file. And Non login shell is started by a program without a login. In case the program just passes the shell executable.
3. What happens when you start a login shell (which files are read and used and Why)?
   * When the bash is invoked as a login shell it first reads and execute the command from /etc/profile, if that exists. After reading the file it looks for ~/.bash profile, ~/.bash login, and ~/.profile and reads and execute the command from the first one that is readable.
4. What happens when you start a non-login shell (Which files are read and used and Why)?
   * When non login shell is started , bash reads and executes the command from ~/.bashhrc, if that file exists. The –rcfile file option will force bash to read and excute commamds from file instead of ~/.bashrc .
5. What are Shell Configuration Files, why do we need it?
   * Shell configuration files are executed automatically when you log in and out of a shell. They initialize and configure a shell upon login and perform cleanup operation upon logout.
6. Explain the Order of file usage from the system/user's home directory when user logs in to the System.
7. What are Shell Variables, list major shell variables and what do they represent?
   * A variable is a location in memory that is used to hold a value. This location is assigned a name to make is descriptive. The value could be any type of data such as a name, number text or filename/ directory. So a variable is nothing more a pointer to a particular data. A shell allows user to create, assign or delete a variable. However these variable are temporary and are deleted after shell session is closed.
8. How we see all our env variables?
   * To see all our env variables us command “ get-childitem env:\* “.
9. How we see all env variables in alphabical order?
   * To check env variables in alphabetical order use “ get-childitem env:\* “ command.
10. What Format does the env var and its values are stored?
11. How do you create your own variable?
    * To create own variable use ~/.bashrc file.
12. How do you start a new bash shell?
13. Difference between Local/Shell variables to Global Variable
    * Local shell variable are declared in function and global shell variables are declared outside the function.
    * If local variable is not initialized then garbage value is stored and if global variable is not defined then a zero is stored.
    * Local variable are created when function start execution and deleted when function is terminated whereas the global variable is created before the program's global execution starts and lost when the program terminates.
14. Making a variable accessible from other the shell in the system.
15. Show the real life use case of exporting a variable
    * Export -p

-p : to view all the exported variable on current shell.

-f : it must be used if the name refers to function.

Name[=value] : you can assign value before exporting the syntax

$ export name[=value]

1. Convert the above script file into a command, The file should run with "myappp" instead of "myapp.sh"
   * Open terminal in system. Goto directory where we want to create a script.
   * Create a file nemes as myappp
   * Copy paste the above script using editor.
   * Make a script executable using command chmod +x myappp
   * Run the script using ./myappp
2. What is Globbing? Explain in depth with examples?
   * Bash does not support native regular expressions like some other standard programming languages. The Bash shell feature that is used for matching or expanding specific types of patterns is called globbing. Globbing is mainly used to match filenames or searching for content in a file. Globbing uses wildcard characters to create the pattern.
   * Example: Suppose, you want to search those text filenames whose names are 4 characters long and extension is .txt. You can apply globbing pattern by using ‘?’ four times to do this task.
   * Find out the list of all files and folder of the current directory.
   * $ ls -l
   * $ ls -l ????.txt
3. List all entries with extension ".sh"
   * Use ls will give you all the list of files and to find the type of the file just use ‘file’ command.
4. List all entries with numbers in it.
5. List all entries that starts with a character and ends with a number
6. List all entries that name length more than 5 characters
7. What is Quoting? and Why do we need it?
   * Quoting is used to remove the special meaning of certain characters or words to the shell. Quoting can be used to disable special treatment for special characters, to prevent reserved words from being recognized as such, and to prevent parameter expansion.
   * In Linux Shell, many special characters have their own special meanings. Sometimes they are used to perform an action while other times they are just used as a character, so the quoting mechanism performs this task it makes us use them in whatever way we want to.
8. Write few(minimum 3) unique examples that shows, how a particular problem is solved using Quoting.
9. How do you find a particular files/directories based on a particular search criteria?

HINT:- look for commands -> locate, find and whereis

* + To find a particular files in linux use ‘locate’ command which serves to find a files on file system.
  + Also you can use ‘find’ command to find different files, directories and perform subsequent operation on that.
  + And we can use ‘whereis’ command is used in linux to find the location of a file of source/binary file of a command and manuals section for a specified file.

1. Write major difference between locate, find and whereis?
   * Locate command is used to find a file on a file system. It searches through prebuild local database of the file on filesystem.
   * Where as the find command is able to find recursively search for any given path for various files.
   * And the whereis command is used find a file through a given path.
2. How Globbing is different from locate, find and whereis?
   * Globbing is basically used to find a filename using a special character called wildcards with a single filename.
3. Explain the Linux File System.
   * A linux file system is a structured collection of files on a disk drive or a partition. A partition is a segment in a memory and contains some specific data. In our machine, there can be various partition of memory. Generally every partition contain a file memory.
4. Explain absolute and Relative Paths
   * A relative path is defines as a path related to present working directly(pwd). It starts with current directory and never starts with a/.
   * A absolute path is defined as specifying location of a file or directory from a root directory.
5. What are the different ways of creating a File in linux System? Write an example of each and the difference between them.
   * We can use touch command to create a file in linus ( $ touch filename).
   * Also we can create a file a file in linux using cat command ($ cat > filename).
   * Echo command is also used to create a file in linux ($ echo “This is the text” > filename).
   * Redirectional operator when used along with linux command directs a output toward particular destination file ( $ grep This file.txt > new\_file1.txt)
   * Vim command is used to edit a file in linux ($ vim filename)
6. In how many ways we can delete the files from linux system? write an example of each and teh difference between them.
   * To remove files from directory use rm command ($ rm filename)
   * Unlike command is also used to delete a file from linux ( $ unlike filename)
7. Archiving files using linux command, write a command to archive set of files from linux commands.
   * Tar command is used in linux to archive files ( tar [options] [archive-file] [file or directory to be archeived]).
8. Extract the archived files from the above step.
   * To extract a file from archive use command -xvf ( $ tar xvf file.tar).