LINUX LAB CYCLE

I. WORKING WITH DIRECTORIES

1. From the current directory list the contents of the /etc.
2. From the current directory list the contents of the /bin and /sbin.
3. List all the files in your home directory including hidden files.
4. List the files /boot in human-readable form.
5. Create dir1/dir2/dir3 in one command in your home directory.
6. Remove dir1/dir2/dir3 in a single command.
7. How will you change over to the parent of, parent of current directory without using any directory names?
8. How will you print the listing of the parent directory from the current directory?

II. WORKING WITH FILES

1. Display the type of file of /bin/cat, /etc/passwd & /usr/bin/passwd
2. Create a directory mywork.
3. Create two files file1.txt and file2.txt
4. Change the date on file1.txt to yesterdays date.
5. Copy file1.txt to copy.file1.txt
6. Rename copy.file1.txt to myfile
7. Create a directory test in the home folder and copy a file from /etc
8. Use one command to remove test and all files in it.
9. Create a directory etcbackup and copy all the files *.conf from /etc

III. FILE CONTENT

1. Create a file called ‘text’ and store your name, age, gender, and place in it.
2. Display the contents of the file ‘text’.
3. Create a file ‘matter’ and type any two sentences in it.
4. Combine contents of the file ‘text’ and ‘matter’ into another file ‘txtmat’.
5. Display the first 12 lines of /etc/services
6. Display the last line of /etc/passwd
7. Use cat to create a file cout.txt that looks like
   One
   Two
   Three
   Four
8. Use cat to create a backup *(catcount)* of the file count.txt
9. Display catcount in the reverse order.
10. Display the readable characters from the /usr/bin/passwd
11. Use ls to find the biggest file in /etc.
12. Open two terminal windows and make sure you are in the same directory for both and follow the commands given below
    Type echo this is the first line tailing.txt
    Then issue the command tall-f tailing.txt in the second terminal
    Go back to the second terminal and type echo this is the second line tailing.txt

Type echo this is the last line>> tailing.txt
Verify tail -f shows all the lines in second window-press ctrl+c to terminate tail command

13. Use cat to create a file named tailing.txt that contains the contents of tailing.txt followed by contents of/etc/passwd

IV. FILTERS

1. Put a sorted list of all bash users in bashusers.txt.
2. Put a sorted list of all logged on users in onlineusers.txt
3. Make a list of all filenames in /etc that contain the string net
4. Make a sorted list of files in /etc that contain the case insensitive string net
5. Look at the output of /sbin/iconfig. Write a line that displays only ip address and the subnet mask
6. Write a line that removes all non-letters from a stream
7. Write a line that receives a text file and outputs all words on a separate line

V. SHELL SCRIPT

Write a shell script to

1. To add two user-given values.
2. To find the largest of three numbers
3. To perform arithmetic operations using 'case… esac'
4. To print the Fibonacci series using the for-do-done loop
5. To check whether the given no. is prime or not using the while-do-done loop.
6. To check whether the given number is Armstrong or not using the until-do-done loop.
7. To display welcome message according to time (GOOD MORNING, GOOD AFTERNOON, etc).
8. To get two strings from the user and check whether the two strings are equal or not, length is 0 or not and then concatenate the two strings.
9. To get a subdirectory name from the user and list the contents inside the directory. Also, display how many entries of the subdirectory start with the file name "ab".
10. To get two file names from the user and check whether they are the same or not. If both the files are same delete the second one.
11. To generate the following pattern by accepting the limit.

```
1
1  2
1   2   3
1   2   3    4
1   2   3
1   2
1
```

Note: here given example is for the limit 4

12. To check whether the entered year is a leap year or not.

13. To search for a pattern in a given file.
14. To list all files for which you have read, write, and execute permissions.
15. To perform file operations using menu menu-driven program. (use Do you want to continue for repeating)
    1. Display
    2. Edit
    3. Copy
    4. Move
    5. Rename
    6. Delete

VI. PARAMETER HANDLING

16. To find the perimeter of the circle using parameter handling
17. To get two file names as command line arguments and perform the comparison of these two files.
18. To check if the given word is a palindrome. (input word as argument)
19. To add n user given values (input as arguments)
20. To take a filename as an argument, list the files if it's a directory, and display the contents if it is a file.

VII. USERS AND GROUPS

1. Create a user called bourneuser, and give him the bourne shell (/bin/sh) as his default shell.
2. Create groups players, sports, athletics, tennis, and cricket.
3. Create the users Serena Williams, Venus Williams, Sachin Tendulkar. All of them are under the group players. Verify that the users and their home directory are properly created.
4. Add Serena and Venus to the group tennis.
5. Try the commands who, whoami, w, id, echo $USER, echo $UID.
6. In one command, make Sachin a member of sports and cricket.
7. Rename the athletics group to athlete.
8. Use the id command to verify that Sachin is a member of cricket.
9. Write the command to add a user Anju Bobby George with the following options - create home directory, make the user belong to the two groups - players (primary group) and athlete, and assign tcsh as the primary command shell.

VIII. FILES & PERMISSIONS

1. As a normal user, create a directory ~/permissions. Create a file myfile.txt owned by yourself in there.
2. Copy a file owned by root from /etc/hosts to the permissions directory. Who owns this file now?
3. As root, create a file file1.txt in the users ~/permissions directory.

4. As a normal user, look who owns the file created by root.
5. Change the ownership of all files in the ~/permissions to yourself.
6. Make sure you have all rights and others can only read.
7. Display umask in octal and in symbolic form.
8. Set umask to 077, but use the symbolic format to set it. Verify that it works.
9. Create a directory that belongs to a group, where every member of that group can read and write to files, and create files. Make sure that people can delete only their own files.

IX. BASIC NETWORKING COMMANDS

1. Use the write utility to communicate with another user to inform him/her to submit the assignment on Monday.
2. Using the mail facility send mail to your friend with the subject backup. Informing him to backup all files from his home directory. Send a copy to admin as well.
3. Use message command to enable and disable messaging.
4. Demonstrate the use of wall command.
5. Demonstrate the use of talk command.

X. SCHEDULING

1. Let the user back up a directory /home onto /tmp/data/backup every day at 5pm.
2. Using proper scheduling mechanisms schedule the following tasks.
   i. Execute a shell script that backups up all the user files in their home directory to /backup/users every day at 5pm except Sunday.
   ii. Schedule a job that executes a shell script that deletes all files in the home directory at midnight of 31st December 2024.

XI. FILE COMPRESSION

1. Create a directory 'system' and copy the contents of /etc directory to this directory and apply all compressing tools and decompress the same. (tar, cpio, dump, rsync and restore).

XII. BACKING UP LINUX

1. Using dump utility perform a level 0 backup /home/(username) directory onto /tmp/(username), restore the file onto /tmp/restore/(username)
2. Using tar perform a full backup of your home directory.
3. Perform an incremental backup using tar utility about the above full backup.
4. Create a backup of all files having .java/.c/.sh extension in your system using cpio utility and extract the archive to /tmp/cpio/(username).

XIII. SERVERS

1. Using ftp login to server 172.16.12.204 and for authentication give your username and password. Upload and download files from the server.
    - Use mget to download multiple files.
    - Use mput to upload multiple files.
2. Using anonymous user access logon to ftp server 172.16.12.100 to upload and download files.
3. Using ftp get kernel-devel-2.6.18-8.el5.i686.rpm and install it on your client machine. [location /var/ftp/pub/server]