**Name: Aryan Penikal Roll no: 50**

**Linux Practical-1**

**File and Directory Related Commands**

|  |  |  |
| --- | --- | --- |
| **Name** | **Syntax and Options** | **Description** |
| touch | touch <option> <file or dir name>  -a : Change access time  -c : Avoids creating a new file  –help : Opens help menu  -v : Prints version | To create an empty file if not exists, or to modify a timestamp |
| cat | cat <option> <file>  cat > filename : To create/overwrite file  cat file : To read from file  cat >> file : To create/append to file  cat -n file : Print line number  cat file1 > file2 : To copy content of one file to another, can also append. | Display contents of file, append to file contents or rewrite file contents. Has several option related to above |
| ls | ls dir : Show content in a directory  ls -F : Gives full listing of files indicating type of file  ls -l : Long listing of files containing permission, time, size and more  ls -R : Recursive listing  ls -r : List in reverse order  ls -t : Last modification time  ls -a : Show hidden files as well | To list all the files and directories in the current or specified directory.  Can also list based on parameters |
| mkdir | mkdir [option] name  mkdir {dir1,dir2,dir3} : To create multiple directories  mkdir -p dir/dir/dir : To create a a directory structure | To create an empty directory |
| cp | cp [option] file1 file2  cp file1 file2 : Copy content of file1 into file2  cp file1 file2 file3 dir : Copy files into dir  cp -r dir1 dir2 : Copy contents of dir1 into dir2 recursively | To copy the file(s) or directory. Can also copy contents |
| mv | mv [option] source destination  mv file1 file2 : Rename file1 as file2 or overwrite file2 if it exists | To move files and directories in to another or renames them |
| rm | rm [option] file  rm file1 file2 : Remove multiple files  rm -r dir : To delete directory recursively | To remove files and directories (if used with -r option) |
| pwd | pwd  pwd -L : Print symbolic path  pwd -P : Print actual path | Prints current working directory |
| rmdir | rmdir directory  rmdir -version : print version information and exit | To remove empty directories |
| exit | exit : simply exits the command prompt  exit status : Exits with given status code | Close terminal window |

1. Login as “kc” user.
2. Create five empty files with the name a1, b2, c3, d4 and e5 in current working directory.

tycs123@tycs-123:~/Desktop$ touch a1 b2 c3 d4 e5

**tycs123@tycs-123:~/Desktop$ ls**

**a1 b2 c3 d4 e5**

1. Create three files with the name f6, g7, h8 using cat command with some meaningful contents with at least five lines each.

**tycs123@tycs-123:~/Desktop$ cat > f6**

**file f6**

**line1**

**line2**

**line3**

**line4**

**tycs123@tycs-123:~/Desktop$ cat > g8**

**file g8**

**line1**

**line2**

**line3**

**line4**

**tycs123@tycs-123:~/Desktop$ cat > h9**

**file h9**

**line1**

**line2**

**line3**

**line4**

Display the contents of files f6, g7 and h8.

cs@ubuntu:~$ cat f6

**tycs123@tycs-123:~/Desktop$ cat f6 g8 h9**

**file f6**

**line1**

**line2**

**line3**

**line4**

**file g8**

**line1**

**line2**

**line3**

**line4**

**file h9**

**line1**

**line2**

**line3**

**line4**

1. Display the contents of files f6, g7 and h8 along with line numbers.

**tycs123@tycs-123:~/Desktop$ cat -n f6 g8 h9**

**1 file f6**

**2 line1**

**3 line2**

**4 line3**

**5 line4**

**6 file g8**

**7 line1**

**8 line2**

**9 line3**

**10 line4**

**11 file h9**

**12 line1**

**13 line2**

**14 line3**

**15 line4**

1. Copy contents of f6 to a1, g7 to b2, h8 to c3.

**tycs123@tycs-123:~/Desktop$ cp f6 a1**

**tycs123@tycs-123:~/Desktop$ cat a1**

**file f6**

**line1**

**line2**

**line3**

**line4**

**tycs123@tycs-123:~/Desktop$ cp g8 b2**

**tycs123@tycs-123:~/Desktop$ cat b2**

**file g8**

**line1**

**line2**

**line3**

**line4**

**tycs123@tycs-123:~/Desktop$ cp h9 c3**

**tycs123@tycs-123:~/Desktop$ cat c3**

**file h9**

**line1**

**line2**

**line3**

**line4**

1. Add the contents of a1, b2 and c3 into “testfile”.

**tycs123@tycs-123:~/Desktop$ cat a1 b2 c3 > testfile**

**tycs123@tycs-123:~/Desktop$ cat testfile**

**file f6**

**line1**

**line2**

**line3**

**line4**

**file g8**

**line1**

**line2**

**line3**

**line4**

**file h9**

**line1**

**line2**

**line3**

**line4**

1. Create three empty directories with names dd1, dd2 and dd3 in current working directory.

**tycs123@tycs-123:~/Desktop$ mkdir dd1 dd2 dd3**

**tycs123@tycs-123:~/Desktop$ ls**

**a1 b2 c3 d4 dd1 dd2 dd3 e5 f6 g8 h9 testfile**

1. Copy the files a1 and b2 to the directories dd1.

**tycs123@tycs-123:~/Desktop$ cp -r a1 b2 dd1**

**tycs123@tycs-123:~/Desktop$ ls dd1**

**a1 b2 bin etc lib lib32 lib64 libx32 lost+found sbin sys**

1. Copy the files f6 and g7 to the directory dd2.

**tycs123@tycs-123:~/Desktop$ cp -r f6 g8 dd2**

**tycs123@tycs-123:~/Desktop$ ls dd2**

**f6 g8**

1. Display the contents of directory dd1 and dd2.

**tycs123@tycs-123:~/Desktop$ ls dd1**

**a1 b2 bin etc lib lib32 lib64 libx32 lost+found sbin sys**

**tycs123@tycs-123:~/Desktop$ ls dd2**

**f6 g8**

1. Copy directory dd1 into dd3.

**tycs123@tycs-123:~/Desktop$ cp -r dd1 dd3**

1. Display contents of directory dd3.

**tycs123@tycs-123:~/Desktop$ ls dd3**

**dd1**

1. Copy directory dd2 into dd4.

**tycs123@tycs-123:~/Desktop$ cp -r dd2 dd4**

1. Display contents of directory dd4.

**tycs123@tycs-123:~/Desktop$ ls dd4**

**f6 g8**

1. Remove the directory dd2 along with its contents.

**tycs123@tycs-123:~/Desktop$ rm -r dd2**

**tycs123@tycs-123:~/Desktop$ ls**

**a1 b2 c3 d4 dd1 dd3 dd4 e5 f6 g8 h9 testfile**

1. Rename the file a1 by newa1 and file b2 by newb2.

**tycs123@tycs-123:~/Desktop$ mv a1 newa1**

**tycs123@tycs-123:~/Desktop$ mv b2 newb2**

1. Rename files d4 and e5 in directory dd4.

**tycs123@tycs-123:~/Desktop$ mv d4 e5 dd4**

1. Display contents of directory dd4.

**tycs123@tycs-123:~/Desktop$ ls dd4**

**d4 e5 f6 g8**

1. Rename directory dd4 with dd5.

**tycs123@tycs-123:~/Desktop$ mv dd4 dd5**

1. Display contents of directory dd5.

**tycs123@tycs-123:~/Desktop$ ls dd5**

**d4 e5 f6 g8**

1. Rename directory dd3 into dd5.

**tycs123@tycs-123:~/Desktop$ mv dd3 dd5**

1. Display contents of directory dd5.

**tycs123@tycs-123:~/Desktop$ ls dd5**

**d4 dd3 e5 f6 g8**

1. Create a directory tree d1/d2/d3 in one command.

**tycs123@tycs-123:~/Desktop$ mkdir -p d1/d2/d3**

1. Change to directory tree d1/d2/d3 in one command.

tycs123@tycs-123:~/Desktop$ cd d1/d2/d3

1. Display the present working directory.

tycs123@tycs-123:~/Desktop/d1/d2/d3$

1. Create files x1, x2 and x3 with some contents in this directory.

tycs123@tycs-123:~/Desktop/d1/d2/d3$ cat > x1

file x1

tycs123@tycs-123:~/Desktop/d1/d2/d3$ cat > x2

file x2

tycs123@tycs-123:~/Desktop/d1/d2/d3$ cat > x3

file x3

1. Copy file x3 into directory d2.

tycs123@tycs-123:~/Desktop/d1/d2/d3$ cd ..

tycs123@tycs-123:~/Desktop/d1/d2$ cd ..

tycs123@tycs-123:~/Desktop/d1$ cd ..

tycs123@tycs-123:~/Desktop$ cp d1/d2/d3/x3 d1/d2

1. Display the contents of directory d2.

tycs123@tycs-123:~/Desktop$ ls d1/d2

d3 x3

1. Rename file x1 into directory d1.

tycs123@tycs-123:~/Desktop$ mv d1/d2/d3/x1 d1

1. Display the contents of directory d1.

tycs123@tycs-123:~/Desktop$ ls d1

d2 x1

1. Change from current working directory to home directory in one command.

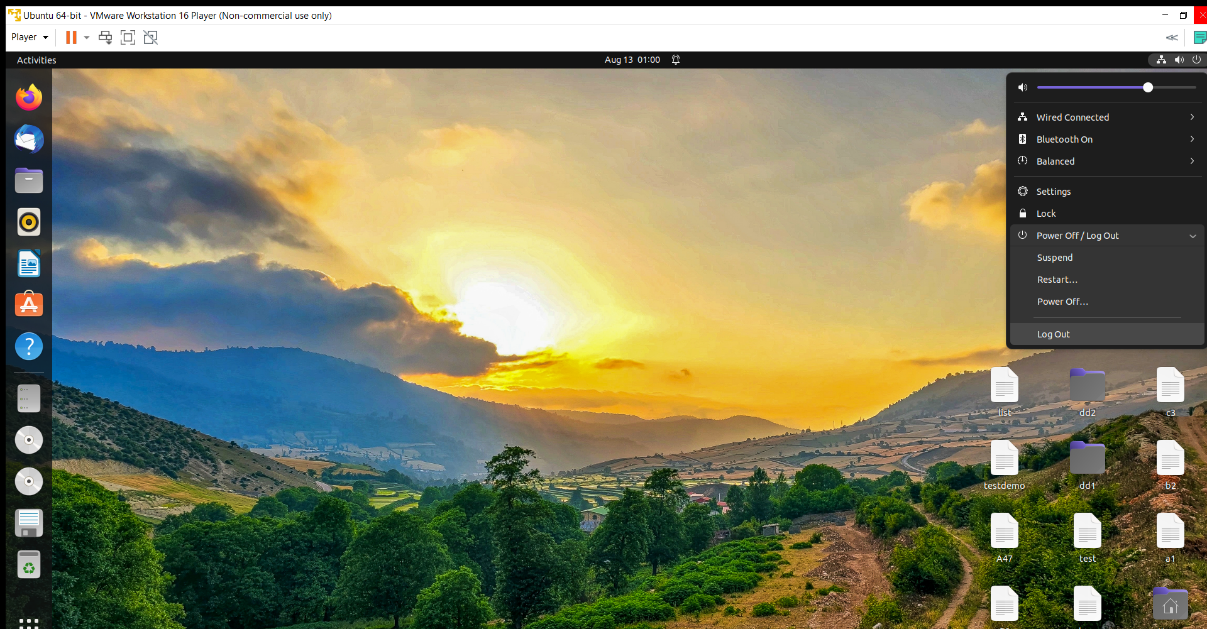
tycs123@tycs-123:~/Desktop/d1/d2/d3$ cd ../../../

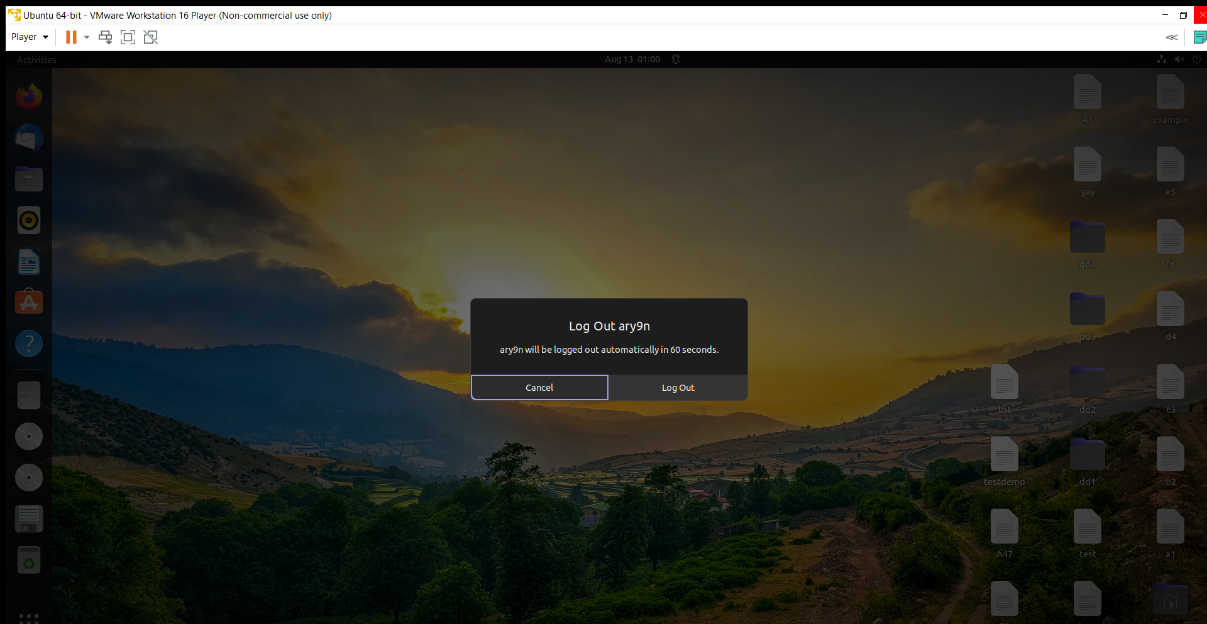
tycs123@tycs-123:~/Desktop$

1. Remove the directory tree d1/d2/d3 in one command. Is directory removed? If no, Why?

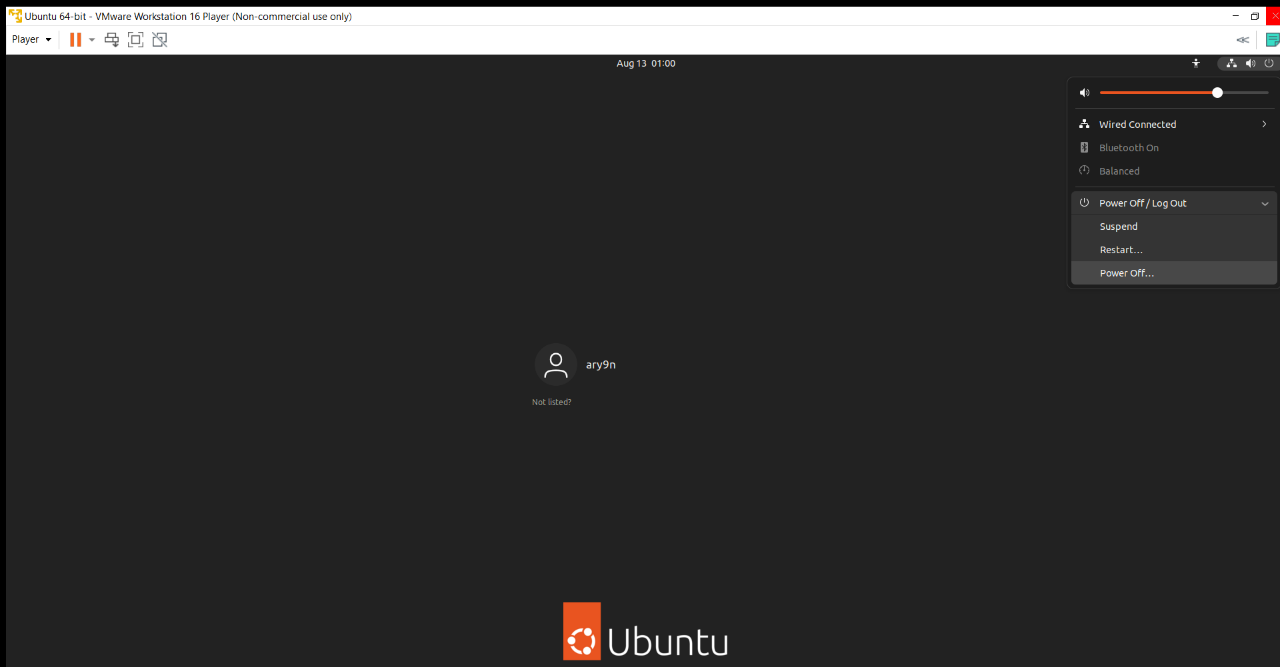
Cannot be removed as directory not empty.

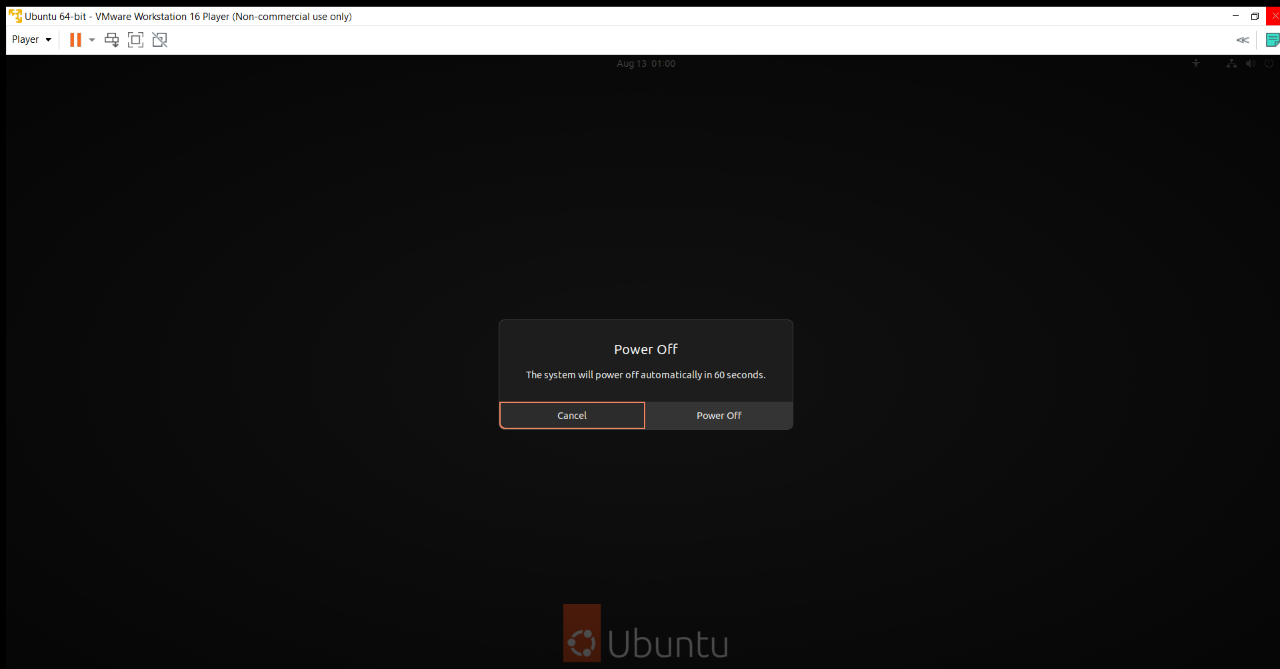
1. Remove your all files and directories.





1. Save your work and logout from current terminal.





**Name: Aryan Penikal Roll No: 50**

**Linux Practical-2**

1. **ls :** To display the contents of the directory. Options are as mentioned below
2. **more :** To view text files or extensive outputs in the command prompt, one screen at a time.

Allows user to scroll up and down the page. Often used with pipe. Controls:

1. Enter key - To scroll down line by line
2. Space bar : To go to next page
3. b key : To go back one page
4. **wc :** Used for counting word, lines and characters in given input. Can take multiple files as input. Options:
5. -l : To display no of lines
6. -w : To display no of words
7. -c : To display no of characters or bytes
8. **file :** To display the type of a file. Can display in brief mode and can display the file type of all files in

given directory.

1. **who :** Information about users currently logged in
2. **Regex in linux** : Contains multiple symbols some of which are
3. . : Wild card character. Matches any one character other than new line
4. ^ : Matches start of the string
5. $ : Matches end of string
6. \* : Matches zero or more characters
7. \ : Used for escape following a character
8. () : Used to match a set of regex
9. ? : Matches one 1 character
10. [] : Used to match any characters present in the brackets. Can also have a range of characters

|  |  |
| --- | --- |
| **Options (LS)** | **Description** |
| ls -a | List all files including hidden files |
| ls -l | Shows the files in long list format |
| ls -lh | Shows file sizes in human readable format |
| ls -lhS | Shows file sizes in human readable format but in descending order |
| ls -n | To print group ID and owner ID instead of names |
| ls -r | To print list in reverse order |
| ls -R | To recursively display the contents |
| ls –version | Check version of ls command |
| ls -t | Sorts files based on last modification time. |

**Write the commands for the following:-**

1. List all files and directories.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls**

**a1 b2 c3 d4 dd1 dd2 dd3 dd4 e5 f6 g7 h8 testfile**

1. List all files and directories in multicolumn format.

**total 52**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 d4**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 dd1**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd4**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 g7**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 h8**

**-rw-rw-r-- 1 ary9n ary9n 108 Aug 12 22:55 testfile**

1. List all files identifying directories and executable files.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -F**

**a1 b2 c3 d4 dd1/ dd2/ dd3/ dd4/ e5 f6 g7 h8 testfile**

1. List the contents of the directory along with all hidden files.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -la**

**total 60**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxr-x--- 14 ary9n ary9n 4096 Aug 12 23:14 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 d4**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 dd1**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd4**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 g7**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 h8**

**-rw-rw-r-- 1 ary9n ary9n 108 Aug 12 22:55 testfile**

1. List recursive list of all files and directories.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -R**

**.:**

**a1 b2 c3 d4 dd1 dd2 dd3 dd4 e5 f6 g7 h8 testfile**

**./dd1:**

**a1 b2**

**./dd2:**

**c3 d4 dd1**

**./dd2/dd1:**

**a1 b2**

**./dd3:**

**dd2 f6 g7**

**./dd3/dd2:**

**c3 d4 dd1**

**./dd3/dd2/dd1:**

**a1 b2**

**./dd4:**

**dd2 f6 g7**

**./dd4/dd2:**

**c3 d4 dd1**

**./dd4/dd2/dd1:**

**a1 b2**

1. Give the file listing in reverse order.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -r**

**testfile h8 g7 f6 e5 dd4 dd3 dd2 dd1 d4 c3 b2 a1**

1. Give long listing of all files and directories.

(List all files and directories with their attributes and file permissions.)

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -R -la**

**.:**

**total 60**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxr-x--- 14 ary9n ary9n 4096 Aug 12 23:14 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 d4**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 dd1**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd4**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 g7**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 h8**

**-rw-rw-r-- 1 ary9n ary9n 108 Aug 12 22:55 testfile**

**./dd1:**

**total 16**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 .**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 b2**

**./dd2:**

**total 20**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 .**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 d4**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 dd1**

**./dd2/dd1:**

**total 16**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 .**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 b2**

**./dd3:**

**total 20**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 ..**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 g7**

**./dd3/dd2:**

**total 20**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 d4**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:58 dd1**

**./dd3/dd2/dd1:**

**total 16**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 b2**

**./dd4:**

**total 20**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 ..**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 g7**

**./dd4/dd2:**

**total 20**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 d4**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:58 dd1**

**./dd4/dd2/dd1:**

**total 16**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:58 b2**

1. List all filenames sorted by last modification time.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -sort -t**

**total 52**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:52 a1**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:52 b2**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:52 c3**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:53 d4**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:53 f6**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:54 e5**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:54 g7**

**4 -rw-rw-r-- 1 ary9n 36 Aug 12 22:54 h8**

**4 -rw-rw-r-- 1 ary9n 108 Aug 12 22:55 testfile**

**4 drwxrwxr-x 2 ary9n 4096 Aug 12 22:57 dd1**

**4 drwxrwxr-x 3 ary9n 4096 Aug 12 22:57 dd2**

**4 drwxrwxr-x 3 ary9n 4096 Aug 12 22:58 dd3**

**4 drwxrwxr-x 3 ary9n 4096 Aug 12 22:58 dd4**

1. Give listing sorted by last modification time.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -la -t**

**total 60**

**drwxr-x--- 14 ary9n ary9n 4096 Aug 12 23:18 ..**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 .**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd4**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 dd1**

**-rw-rw-r-- 1 ary9n ary9n 108 Aug 12 22:55 testfile**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 h8**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 g7**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 d4**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 a1**

1. Displays all files sorted according to the extension.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -sort -x**

**total 52**

**4 a1 4 b2 4 c3 4 d4 4 f6 4 e5 4 g7 4 h8 4 testfile 4 dd1 4 dd2**

**4 dd3 4 dd4**

1. Displays all files sorted according to the extension in reverse order.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -xr**

**dd4 dd3 dd2 dd1 testfile h8 g7 e5 f6 d4 c3 b2 a1**

1. Displays all files sorted according to the last modification time in reverse

order.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -ltr**

**total 52**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 d4**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 g7**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 h8**

**-rw-rw-r-- 1 ary9n ary9n 108 Aug 12 22:55 testfile**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 dd1**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd4**

**ary9n@ary9n-virtual-machine:~/Desktop$**

1. Display all files and directories including hidden files and executable files.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -Xa**

**a1 b2 c3 d4 dd1 dd2 dd3 dd4 e5 f6 g7 h8 testfile . ..**

1. Display detailed listing including hidden files sorted in reverse order.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -lar**

**total 60**

**-rw-rw-r-- 1 ary9n ary9n 108 Aug 12 22:55 testfile**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 h8**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 g7**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 e5**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd4**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 ary9n ary9n 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 2 ary9n ary9n 4096 Aug 12 22:57 dd1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 d4**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 a1**

**drwxr-x--- 14 ary9n ary9n 4096 Aug 12 23:18 ..**

**drwxr-xr-x 6 ary9n ary9n 4096 Aug 12 22:58 .**

1. List the contents of the directory “dd1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls dd1**

**a1 b2**

1. Give listing of “dd1” with their attributes and file permissions.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -l dd1**

**total 8**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:57 b2**

1. List all filenames with one screen at a time.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -lah | more**

**total 60K**

**drwxr-xr-x 6 ary9n ary9n 4.0K Aug 12 22:58 .**

**drwxr-x--- 14 ary9n ary9n 4.0K Aug 12 23:18 ..**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 a1**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 d4**

**drwxrwxr-x 2 ary9n ary9n 4.0K Aug 12 22:57 dd1**

**drwxrwxr-x 3 ary9n ary9n 4.0K Aug 12 22:57 dd2**

**drwxrwxr-x 3 ary9n ary9n 4.0K Aug 12 22:58 dd3**

**drwxrwxr-x 3 ary9n ary9n 4.0K Aug 12 22:58 dd4**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 g7**

**-rw-rw-r-- 1 ary9n ary9n 36 Aug 12 22:54 h8**

**-rw-rw-r-- 1 ary9n ary9n 108 Aug 12 22:55 testfile**

1. List all filenames with 2 characters

**ary9n@ary9n-virtual-machine:~/Desktop$ ls ??**

**a1 b2 c3 d4 e5 f6 g7 h8**

1. List all filenames with 3 characters.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls ???**

**dd1:**

**a1 b2**

**dd2:**

**c3 d4 dd1**

**dd3:**

**dd2 f6 g7**

**dd4:**

**dd2 f6 g7**

1. List all filenames starting with a lowercase letter ‘a’.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls a\***

**a1**

1. List all filenames starting with an uppercase letter ‘A’.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls A\***

**ls: cannot access 'A\*': No such file or directory**

1. List all filenames starting with vowel.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls [aeiouAEIOU]\***

**a1 e5**

1. List all filenames with the last character as ‘a’ or ‘b’ or ‘c’ or ‘d’.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls \*[abcd]**

**ls: cannot access '\*[abcd]': No such file or directory**

1. List all filenames with exactly three characters in which the second character is a vowel.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls ?[AEIOUaeiou]?**

**yay**

1. List all filenames with three characters in which first character is an uppercase letter and second character is a digit.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls [A-Z]?[0-9]**

**A47**

1. List all filenames starting with an uppercase letter vowel and ends with a digit.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls [AEIOU]\*[0-9]**

**A1 A47**

1. List all filenames in which last two characters is a digit in ranger 1 to 9.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls \*[1-9][1-9]**

**A47**

1. List all filenames with four characters in which first character is ‘a’ and third character is ‘b’.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls a?b?**

**ls: cannot access 'a?b?': No such file or directory**

1. List all filenames whose first character is any thing other than a small case letter vowel.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls [!aeiou]\***

**A1 A47 b2 B2 c3 d4 f6 g7 h8 testfile yay**

**dd1:**

**a1 b2**

**dd2:**

**c3 d4 dd1**

**dd3:**

**dd2 f6 g7**

**dd4:**

**dd2 f6 g7**

1. List all files and directories with five characters whose first character is in the range a to m, second character is in the range c to z and third character is a digit in the range 4 to 9.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls [a-m][c-z][4-9]??**

**ls: cannot access '[a-m][c-z][4-9]??': No such file or directory**

1. Find the type of all files.

**ary9n@ary9n-virtual-machine:~/Desktop$ file \***

**a1: ASCII text**

**A1: empty**

**A47: empty**

**b2: ASCII text**

**B2: empty**

**c3: ASCII text**

**d4: ASCII text**

**dd1: directory**

**dd2: directory**

**dd3: directory**

**dd4: directory**

**e5: ASCII text**

**f6: ASCII text**

**g7: ASCII text**

**h8: ASCII text**

**testfile: ASCII text**

**yay: empty**

1. Display and count the number of lines, words and characters of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc demo**

**5 10 36 demo**

1. Display and count the number of lines of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -l demo**

**5 demo**

1. Display and count the number of words of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -w demo**

**10 demo**

1. Display and count the number of characters of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -c demo**

**36 demo**

1. Display and count the number of lines, words and characters of file “demo” and “example” and “test”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc demo example test**

**5 10 36 demo**

**5 10 36 example**

**15 30 108 test**

**25 50 180 total**

1. Count the number of all files and directories.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -n**

**total 52**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:52 a1**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:42 A1**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:45 A47**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:42 B2**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:53 d4**

**drwxrwxr-x 2 1000 1000 4096 Aug 12 22:57 dd1**

**drwxrwxr-x 3 1000 1000 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 3 1000 1000 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 1000 1000 4096 Aug 12 22:58 dd4**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:54 demo**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:54 example**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 1000 1000 108 Aug 12 22:55 test**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:41 yay**

1. Count the number of all users currently logged in to the system.

**ary9n@ary9n-virtual-machine:~/Desktop$ who -q**

**ary9n**

**# users=1**

1. Store the number of users currently logged in to the system in file “testdemo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ who -q > testdemo**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat testdemo**

**ary9n**

**# users=1**

1. Display and store the listing of all filenames in file “list”.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls > list**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat list**

**a1**

**A1**

**A47**

**b2**

**B2**

**c3**

**d4**

**dd1**

**dd2**

**dd3**

**dd4**

**demo**

**e5**

**example**

**f6**

**list**

**test**

**testdemo**

**yay**

**50 - Aryan Penikal**

**Linux Practical-3**

**General Purpose Utility and More Bash Shell Commands**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Syntax and Options** |
| cal | Shows current month calendar with current date highlighted. Can also show several varied outputs | cal : Simply shows current month and date  cal Year : Shows calendar of mentioned year  cal -y : Current year calendar  cal month year : Shows mentioned month and year calendar  cal -j : Shows calendar of current month in Julian calendar format |
| date | To display system date and time | date : current date and time  date -u : GMT format  %D : mm/dd/yy format                               %d : Display day of month  %a : Displays abbreviated weekday           %A : Full weekday  %h : Abbreviated month name                   %B : Full month name  %m : Month of the year                              %y : Last 2 digits of year  %Y : Four digit year                                   %T : 24 hour format time  %H : Hour                            %M: Minute                     %S : Seconds |
| echo | To display string passed | echo string : Displays mentioned string with options  echo \* : Display all files and folders  echo -n : Omit trailing new line  echo -e : Enables interpretation of backslash escapes. Escape characters:  \b : Backspace  \c : Suppress trailing new line  \n : New line  \t : Horizontal tab space  \v : Vertical tab space |
| bc | A command line calculator. Can be used to create and assign values to variables. Also accepts many types of operators | Used with echo command using a pipe as : echo “12+5” | bc  bc : Starts command line calculator |
| who | Information about users logged in | who [options]  who -H: display headers          who -u : List of logged in users  who -a : All details of current logged in users  whoami : System’s username is displayed  id : User identification information |
| w | View users currently logged in | w [options] [username]  Shows system time, Up time, no of users, and more information |

1. Display the calendar of the current month of the current year.

ary9n@ary9n-virtual-machine:~/Desktop$ cal

August 2022

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31

1. Display the calendar of the year 2009.

ary9n@ary9n-virtual-machine:~/Desktop$ cal 2009

2009

January February March

Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa

1 2 3 1 2 3 4 5 6 7 1 2 3 4 5 6 7

4 5 6 7 8 9 10 8 9 10 11 12 13 14 8 9 10 11 12 13 14

11 12 13 14 15 16 17 15 16 17 18 19 20 21 15 16 17 18 19 20 21

18 19 20 21 22 23 24 22 23 24 25 26 27 28 22 23 24 25 26 27 28

25 26 27 28 29 30 31 29 30 31

April May June

Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa

1 2 3 4 1 2 1 2 3 4 5 6

5 6 7 8 9 10 11 3 4 5 6 7 8 9 7 8 9 10 11 12 13

12 13 14 15 16 17 18 10 11 12 13 14 15 16 14 15 16 17 18 19 20

19 20 21 22 23 24 25 17 18 19 20 21 22 23 21 22 23 24 25 26 27

26 27 28 29 30 24 25 26 27 28 29 30 28 29 30

31

July August September

Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa

1 2 3 4 1 1 2 3 4 5

5 6 7 8 9 10 11 2 3 4 5 6 7 8 6 7 8 9 10 11 12

12 13 14 15 16 17 18 9 10 11 12 13 14 15 13 14 15 16 17 18 19

19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26

26 27 28 29 30 31 23 24 25 26 27 28 29 27 28 29 30

30 31

October November December

Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa

1 2 3 1 2 3 4 5 6 7 1 2 3 4 5

4 5 6 7 8 9 10 8 9 10 11 12 13 14 6 7 8 9 10 11 12

11 12 13 14 15 16 17 15 16 17 18 19 20 21 13 14 15 16 17 18 19

18 19 20 21 22 23 24 22 23 24 25 26 27 28 20 21 22 23 24 25 26

25 26 27 28 29 30 31 29 30 27 28 29 30 31

1. Display the calendar of the month July of the year 1998.

ary9n@ary9n-virtual-machine:~/Desktop$ cal 7 1998

July 1998

Su Mo Tu We Th Fr Sa

1 2 3 4

5 6 7 8 9 10 11

12 13 14 15 16 17 18

19 20 21 22 23 24 25

26 27 28 29 30 31

1. Display today’s date.

ary9n@ary9n-virtual-machine:~/Desktop$ date

Saturday 13 August 2022 12:05:32 AM IST

1. Display date as mm/dd/yy.

ary9n@ary9n-virtual-machine:~/Desktop$ date "+%D"

08/13/22

1. Display current hour, minutes and seconds separately.

ary9n@ary9n-virtual-machine:~/Desktop$ date "+%H"

00

ary9n@ary9n-virtual-machine:~/Desktop$ date "+%M"

09

ary9n@ary9n-virtual-machine:~/Desktop$ date "+%S"

16

1. Display time in hours, minutes and seconds as HH:MM:SS.

ary9n@ary9n-virtual-machine:~/Desktop$ date "+%T"

00:10:29

1. Display abbreviated month name.

ary9n@ary9n-virtual-machine:~/Desktop$ date +"%b"

Aug

1. Display day of the week.

ary9n@ary9n-virtual-machine:~/Desktop$ date "+%w"

6

1. Display full weekday.

ary9n@ary9n-virtual-machine:~/Desktop$ date +"%A"

Saturday

1. Display abbreviated weekday.

ary9n@ary9n-virtual-machine:~/Desktop$ date +"%a"

Sat

1. Display time in AM/PM notation.

ary9n@ary9n-virtual-machine:~/Desktop$ date "+%r"

12:14:37 AM IST

1. Display last two digits of the year.

ary9n@ary9n-virtual-machine:~/Desktop$ date +'%y'

22

1. Display full year.

ary9n@ary9n-virtual-machine:~/Desktop$ date +'%Y'

2022

1. Display the date and time on two different lines.

ary9n@ary9n-virtual-machine:~/Desktop$ date +"%D%n%T"

08/13/22

00:15:11

1. Calculate the following using bc command:-
   1. 6 + 7
2. 12 \* 12
3. 144 / 3

iv) scale = 2 25 / 7

1. (8 + 2) \* 5

1. (2 + 3) + 8 - 4 \* (6 – 3)
2. 2 ^ 6
3. 12\*12; 12 / 6
4. ibase=2

11001010

1. obase=2

14

ary9n@ary9n-virtual-machine:~/Desktop$ bc

bc 1.07.1

Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.

This is free software with ABSOLUTELY NO WARRANTY.

For details type `warranty'.

6+7

13

12\*12

144

144/3

48

scale=2

25/7

3.57

(8+2)\*5

50

(2+3)+8-4\*(6-3)

1

2^6

64

12\*12;12/6

144

2.00

ibase=2

11001010

202

obase=2

14

11

1. Display all the users currently logged in the system.

ary9n@ary9n-virtual-machine:~/Desktop$ who

ary9n tty2 2022-08-12 22:51 (tty2)

1. Display the login name of the current terminal.

ary9n@ary9n-virtual-machine:~/Desktop$ who

ary9n tty2 2022-08-12 22:51 (tty2)

1. Display all the users currently logged in the system with heading.

ary9n@ary9n-virtual-machine:~/Desktop$ w

00:40:22 up 1:49, 1 user, load average: 0.04, 0.01, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

ary9n tty2 tty2 22:51 1:49m 0.01s 0.01s /usr/libexec/gn

1. Display the terminal file of the current terminal.

ary9n@ary9n-virtual-machine:~/Desktop$ who -Hu

NAME LINE TIME IDLE PID COMMENT

ary9n tty2 2022-08-12 22:51 01:49 1653 (tty2)

1. Display the message on to the terminal as “Hello World”.

ary9n@ary9n-virtual-machine:~/Desktop$ echo 'hello world'

hello world

1. Display the message on to the terminal as “Hello World”.

ary9n@ary9n-virtual-machine:~/Desktop$ echo 'hello world'

hello world

1. What will be the output of the following:-
   * 1. x = 10

echo The value of x is $x

ary9n@ary9n-virtual-machine:~/Desktop$ x=10

ary9n@ary9n-virtual-machine:~/Desktop$ echo The value of x is $x

The value of x is 10

* + 1. str=”Linux Practical 1”

echo The string is $str

ary9n@ary9n-virtual-machine:~/Desktop$ str='Linux Prac'

ary9n@ary9n-virtual-machine:~/Desktop$ echo The string is $str

The string is Linux Prac

* + 1. d=`date`

echo “Today’s date is $d”

ary9n@ary9n-virtual-machine:~/Desktop$ d=`date`

ary9n@ary9n-virtual-machine:~/Desktop$ echo Todays date is $d

Todays date is Saturday 13 August 2022 12:49:07 AM IST

* + 1. echo “The number of users currently logged in are `who` ”

ary9n@ary9n-virtual-machine:~/Desktop$ echo "The number of users currently logged in are `who`"

The number of users currently logged in are ary9n tty2 2022-08-12 22:51 (tty2)

1. Find the type of all files.

**ary9n@ary9n-virtual-machine:~/Desktop$ file \***

**a1: ASCII text**

**A1: empty**

**A47: empty**

**b2: ASCII text**

**B2: empty**

**c3: ASCII text**

**d4: ASCII text**

**dd1: directory**

**dd2: directory**

**dd3: directory**

**dd4: directory**

**e5: ASCII text**

**f6: ASCII text**

**g7: ASCII text**

**h8: ASCII text**

**testfile: ASCII text**

**yay: empty**

1. Display and count the number of lines, words and characters of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc demo**

**5 10 36 demo**

1. Display and count the number of lines of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -l demo**

**5 demo**

1. Display and count the number of words of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -w demo**

**10 demo**

1. Display and count the number of characters of file “demo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -c demo**

**36 demo**

1. Display and count the number of lines, words and characters of file “demo” and “example” and “test”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc demo example test**

**5 10 36 demo**

**5 10 36 example**

**15 30 108 test**

**25 50 180 total**

1. Count the number of all files and directories.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls -n**

**total 52**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:52 a1**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:42 A1**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:45 A47**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:52 b2**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:42 B2**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:52 c3**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:53 d4**

**drwxrwxr-x 2 1000 1000 4096 Aug 12 22:57 dd1**

**drwxrwxr-x 3 1000 1000 4096 Aug 12 22:57 dd2**

**drwxrwxr-x 3 1000 1000 4096 Aug 12 22:58 dd3**

**drwxrwxr-x 3 1000 1000 4096 Aug 12 22:58 dd4**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:54 demo**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:54 e5**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:54 example**

**-rw-rw-r-- 1 1000 1000 36 Aug 12 22:53 f6**

**-rw-rw-r-- 1 1000 1000 108 Aug 12 22:55 test**

**-rw-rw-r-- 1 1000 1000 0 Aug 12 23:41 yay**

1. Count the number of all users currently logged in to the system.

**ary9n@ary9n-virtual-machine:~/Desktop$ who -q**

**ary9n**

**# users=1**

1. Store the number of users currently logged in to the system in file “testdemo”.

**ary9n@ary9n-virtual-machine:~/Desktop$ who -q > testdemo**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat testdemo**

**ary9n**

**# users=1**

1. Display and store the listing of all filenames in file “list”.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls > list**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat list**

**a1**

**A1**

**A47**

**b2**

**B2**

**c3**

**d4**

**dd1**

**dd2**

**dd3**

**dd4**

**demo**

**e5**

**example**

**f6**

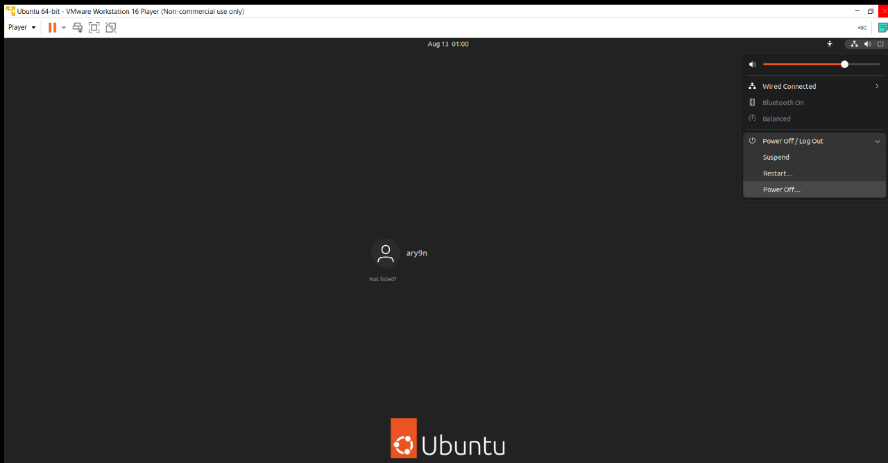
**list**

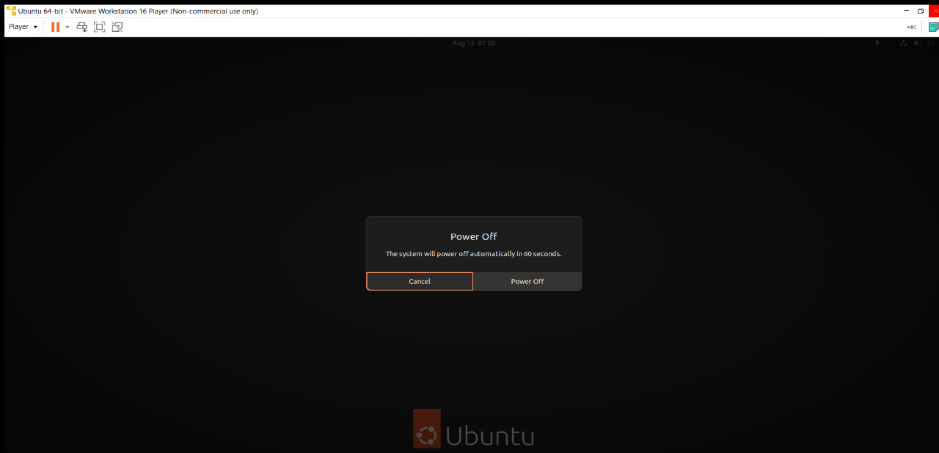
**test**

**testdemo**

**yay**

1. Display and store the list of all users currently logged in into the system in file users.
2. Save your work and logout from current terminal.





**Name: Aryan Penikal Roll No: 50**

**Linux Practical-4**

**File and Simple Filter Commands**

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Syntax and Options** |
| sort | To sort a file’s content | sort [options] [file] : Sort file mentioned  sort -r : reverse order sorting  sort -n : Sort numerically  sort -k : sort given column  sort -c : Check if file is sorted  sort -u : remove duplicates  sort -t : Specifies the delimiter |
| cmp | Compare 2 files byte by byte | cmp [options] [file] [file]  cmp -b : Display the differing bytes  cmp -l : Byte position and value for all differing bytes  cmp -n : Limit no of bytes |
| comm | Compare sorted files line by line. Shows common lines and unique lines | comm [option] file1 file2 : If no options, shows 3 column output  -1 : Suppress 1st column     -2 : suppress 2nd column  -3 : Suppress 3rd column  - -check-order : check if input is sorted |
| diff | Shows differences in files. Shows which lines need to be changed to make them identical | diff [options] file1 file2  Output has a: add, c : change, d: delete values  -i : Case insensitive comparison  -c: Context mode                   -u : Unified mode |
| head | Shows first 10 lines of file | head [option] file  -n : Display specific lines only |
| tail | Shows last 10 lines of file | tail [option] file  -n : Display specific lines only |
| uniq | Used to filter out repeated lines in a file. | uniq [option] [input]  -c : Count no of times line was repeated  -d : Only print repeated lines  -i : ignore case |
| tr | To translate or delete characters. | tr [option] set1 set2 : transform characters from set1 to set2.  -d : Delete characters from 1st set  -t : Truncates set 1  Set1 and set2 can be regex pattern too. |
| cut | To cut specific columns from the file | cut option [file]  -b : Extract specific bytes  -d : Specified delimiter  -f : Specific column to cut |
| paste | To join files horizontally specified by tab | paste option [files]  -d : Delimiter  -s : Serial or sequential merging |

1. Create the following file with name “west” with the contents:-

California

Washington

Oregon

Nevada

Utah

**ary9n@ary9n-virtual-machine:~/Desktop$ cat > west**

**California**

**Washington**

**Oregon**

**Nevada**

**Utah**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat west**

**California**

**Washington**

**Oregon**

**Nevada**

**Utah**

1. Create the following file with name “coast” with the contents:-

Florida

Washington

Maine

Oregon

California

Georgia

**ary9n@ary9n-virtual-machine:~/Desktop$ cat > coast**

**Florida**

**Washington**

**Maine**

**Oregon**

**California**

**Georgia**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat coast**

**Florida**

**Washington**

**Maine**

**Oregon**

**California**

**Georgia**

Write the commands for the following:-

1. Sort the above two files and store the output in “sor\_west” and “sor\_coast”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort west > sor\_west**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat sor\_west**

**California**

**Nevada**

**Oregon**

**Utah**

**Washington**

**ary9n@ary9n-virtual-machine:~/Desktop$ sort coast > sor\_coast**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat sor\_coast**

**California**

**Florida**

**Georgia**

**Maine**

**Oregon**

**Washington**

**ary9n@ary9n-**

1. Compare two files “sor\_west” and “sor\_coast” and check whether they differ.

**ary9n@ary9n-virtual-machine:~/Desktop$ cmp sor\_west sor\_coast**

**sor\_west sor\_coast differ: byte 12, line 2**

1. Compare two files “sor\_west” and “sor\_coast” byte by byte and display its octal value.

**ary9n@ary9n-virtual-machine:~/Desktop$ cmp -lb sor\_west sor\_coast**

**12 116 N 106 F**

**13 145 e 154 l**

**14 166 v 157 o**

**15 141 a 162 r**

**16 144 d 151 i**

**17 141 a 144 d**

**18 12 ^J 141 a**

**19 117 O 12 ^J**

**20 162 r 107 G**

**22 147 g 157 o**

**23 157 o 162 r**

**24 156 n 147 g**

**25 12 ^J 151 i**

**26 125 U 141 a**

**27 164 t 12 ^J**

**28 141 a 115 M**

**29 150 h 141 a**

**30 12 ^J 151 i**

**31 127 W 156 n**

**32 141 a 145 e**

**33 163 s 12 ^J**

**34 150 h 117 O**

**35 151 i 162 r**

**36 156 n 145 e**

**38 164 t 157 o**

**39 157 o 156 n**

**40 156 n 12 ^J**

**41 12 ^J 127 W**

**cmp: EOF on sor\_west after byte 41**

1. Display the lines unique to file “sor\_west”, lines unique to file “sor\_coast” and lines common to both.

**ary9n@ary9n-virtual-machine:~/Desktop$ comm sor\_west sor\_coast**

**California**

**Florida**

**Georgia**

**Maine**

**Nevada**

**Oregon**

**Utah**

**Washington**

**ary9n@ary9n-virtual-machine:~/D**

1. Display the lines which are unique to file “sor\_west”.

**ary9n@ary9n-virtual-machine:~/Desktop$ comm -23 sor\_west sor\_coast**

**Nevada**

**Utah**

1. Display the lines which are unique to file “sor\_coast”.

**ary9n@ary9n-virtual-machine:~/Desktop$ comm -13 sor\_west sor\_coast**

**Florida**

**Georgia**

**Maine**

1. Display the lines which are common to both files.

**ary9n@ary9n-virtual-machine:~/Desktop$ comm -12 sor\_west sor\_coast**

**California**

**Oregon**

**Washington**

1. Display lines unique to file “sor\_west“ and lines unique to file “sort\_coast”.

**ary9n@ary9n-virtual-machine:~/Desktop$ comm -3 sor\_west sor\_coast**

**Florida**

**Georgia**

**Maine**

**Nevada**

**Utah**

1. Display lines unique to file “sor\_west“ and lines common to both files.

**ary9n@ary9n-virtual-machine:~/Desktop$ comm -2 sor\_west sor\_coast**

**California**

**Nevada**

**Oregon**

**Utah**

**Washington**

1. Display lines unique to file “sor\_coast“ and lines common to both files.

**ary9n@ary9n-virtual-machine:~/Desktop$ comm -1 sor\_west sor\_coast**

**California**

**Florida**

**Georgia**

**Maine**

**Oregon**

**Washington**

1. Display the differences of lines between two files “sor\_west” and “sor\_coast” in context form.

**ary9n@ary9n-virtual-machine:~/Desktop$ diff sor\_west sor\_coast**

**2c2,4**

**< Nevada**

**---**

**> Florida**

**> Georgia**

**> Maine**

**4d5**

**< Utah**

1. Create the file with the name “testu1” with the following contents:-

DCNII

ADVJAVAII

LINUXOS

SEII

WEBNETII

CPROG

DM

CG

COREJAVA

DBMSI

DCNII

LINUXOS

CPROG

COD

MICROPROC

WEBNETII

SEII

CG

**ary9n@ary9n-virtual-machine:~/Desktop$ cat testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

1. Create the file with the name “testu2” with the following contents:-

01:accounts:6213:a

01:accounts:6213:a

02:admin:6403:b

03:marketing:6521:c

03:marketing:6521:c

04:personnel:7630:d

05:production:8589:e

05:production:8589:e

06:sales:9876:f

**ary9n@ary9n-virtual-machine:~/Desktop$ cat testu2**

**01:accounts:6213:a**

**01:accounts:6213:a**

**02:admin:6403:b**

**03:marketing:6521:c**

**03:marketing:6521:c**

**04:personnel:7630:d**

**05:production:8589:e**

**05:production:8589:e**

**06:sales:9876:f**

1. Create the file with the name “testu3” with the following contents:-

50

20

10

0

-1

5

2

100

10

**ary9n@ary9n-virtual-machine:~/Desktop$ cat testu3**

**50**

**20**

**10**

**0**

**-1**

**5**

**2**

**100**

**10**

1. Create the file with the name “testu4” with the following contents:-

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:d

02:admin:6521:b

03:marketing:6521:c

**ary9n@ary9n-virtual-machine:~/Desktop$ cat testu4**

**01:accounts:6213:a**

**06:sales:6213:f**

**05:production:5489:e**

**04:personnel:7630:d**

**02:admin:6521:b**

**03:marketing:6521:c**

Write the commands for the following using simple filter commands:-

1. Display first ten lines of file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ head testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

1. Display first seven lines of file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ head -7 testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

1. Display last ten lines of file “testu1”.

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

1. Display last five lines of file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ tail -5 testu1**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

1. Display from line number three onwards of file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ tail -n +3 testu1**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

1. Sort the file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort testu1**

**ADVJAVAII**

**CG**

**CG**

**COD**

**COREJAVA**

**CPROG**

**CPROG**

**DBMSI**

**DCNII**

**DCNII**

**DM**

**LINUXOS**

**LINUXOS**

**MICROPROC**

**SEII**

**SEII**

**WEBNETII**

**WEBNETII**

1. Sort the file “testu3” in ascending order.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort testu3**

**0**

**-1**

**10**

**10**

**100**

**2**

**20**

**5**

**50**

1. Sort the file “testu3” in descending order.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -r testu3**

**50**

**5**

**20**

**2**

**100**

**10**

**10**

**-1**

**0**

1. Remove repeated lines and sort the file “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ uniq testu2 | sort**

**01:accounts:6213:a**

**02:admin:6403:b**

**03:marketing:6521:c**

**04:personnel:7630:d**

**05:production:8589:e**

**06:sales:9876:f**

1. Sort on 4th and 5th column of second field of file “testu4”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -t":" -k 2.4,2.5 testu4**

**05:production:5489:e**

**06:sales:6213:f**

**02:admin:6521:b**

**03:marketing:6521:c**

**01:accounts:6213:a**

**04:personnel:7630:d**

1. Sort on second field of file “testu4”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -t":" -k 2 testu4**

**01:accounts:6213:a**

**02:admin:6521:b**

**03:marketing:6521:c**

**04:personnel:7630:d**

**05:production:5489:e**

**06:sales:6213:f**

1. Sort on third field of file “testu4”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -t":" -k 3 testu4**

**05:production:5489:e**

**01:accounts:6213:a**

**06:sales:6213:f**

**02:admin:6521:b**

**03:marketing:6521:c**

**04:personnel:7630:d**

1. Check whether file “testu2” is already sorted. Observe the output?

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -c testu2**

1. Check whether file “testu4” is already sorted. Observe the output?

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -c testu4**

**sort: testu4:3: disorder: 05:production:5489:e**

1. Check whether second field of “testu2” is already sorted. Observe the output?

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -c -t":" -k 2 testu2**

1. Check whether second field of “testu4” is already sorted. Observe the output?

**ary9n@ary9n-virtual-machine:~/Desktop$ sort -c -t":" -k 2 testu4**

**sort: testu4:3: disorder: 05:production:5489:e**

1. Store the sorted output of file “testu1” in “sort\_testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort testu1 > sor\_testu1**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat sor\_testu1**

**ADVJAVAII**

**CG**

**CG**

**COD**

**COREJAVA**

**CPROG**

**CPROG**

**DBMSI**

**DCNII**

**DCNII**

**DM**

**LINUXOS**

**LINUXOS**

**MICROPROC**

**SEII**

**SEII**

**WEBNETII**

**WEBNETII**

1. Store the sorted output of file “testu3” in “sort\_testu3”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sort testu3 > sor\_testu3**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat sor\_testu3**

**0**

**-1**

**10**

**10**

**100**

**2**

**20**

**5**

**50**

1. Merge the sorted output of file “sort\_testu1” and “sort\_testu3”.

**ary9n@ary9n-virtual-machine:~/Desktop$ paste sor\_testu1 sor\_testu3**

**ADVJAVAII 0**

**CG -1**

**CG 10**

**COD 10**

**COREJAVA 100**

**CPROG 2**

**CPROG 20**

**DBMSI 5**

**DCNII 50**

**DCNII**

**DM**

**LINUXOS**

**LINUXOS**

**MICROPROC**

**SEII**

**SEII**

**WEBNETII**

**WEBNETII**

1. Display unique lines only in file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ uniq sor\_testu1**

**ADVJAVAII**

**CG**

**COD**

**COREJAVA**

**CPROG**

**DBMSI**

**DCNII**

**DM**

**LINUXOS**

**MICROPROC**

**SEII**

**WEBNETII**

1. Display unique lines only in file “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ uniq testu2**

**01:accounts:6213:a**

**02:admin:6403:b**

**03:marketing:6521:c**

**04:personnel:7630:d**

**05:production:8589:e**

**06:sales:9876:f**

1. Display only non-repeated lines in file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ uniq -u testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

1. Display only non-repeated lines in file “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ uniq -u testu2**

**02:admin:6403:b**

**04:personnel:7630:d**

**06:sales:9876:f**

1. Display one copy of duplicate lines in files “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ uniq -d testu1**

1. Display one copy of duplicate lines in files “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ uniq -d testu2**

**01:accounts:6213:a**

**03:marketing:6521:c**

**05:production:8589:e**

1. Count the frequency of occurrence of all lines in file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -l testu1**

**18 testu1**

1. Count the frequency of occurrence of all lines in files “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ wc -l testu2**

**9 testu2**

1. Translate a character ‘a’ with ‘A’ in files “testu1” and “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ tr 'a' 'A' < testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

**ary9n@ary9n-virtual-machine:~/Desktop$ tr 'a' 'A' < testu2**

**01:Accounts:6213:A**

**01:Accounts:6213:A**

**02:Admin:6403:b**

**03:mArketing:6521:c**

**03:mArketing:6521:c**

**04:personnel:7630:d**

**05:production:8589:e**

**05:production:8589:e**

**06:sAles:9876:f**

1. Translate characters ‘x’, ’y’, ‘z’ with ‘X’,’Y’,’Z’ respectively in files “testu1” and “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ tr "xyz" "XYZ" < testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

**ary9n@ary9n-virtual-machine:~/Desktop$ tr "xyz" "XYZ" < testu2**

**01:accounts:6213:a**

**01:accounts:6213:a**

**02:admin:6403:b**

**03:marketing:6521:c**

**03:marketing:6521:c**

**04:personnel:7630:d**

**05:production:8589:e**

**05:production:8589:e**

**06:sales:9876:f**

1. Translate all lowercase letters ‘a’ to ‘z’ with ‘A’ to ‘Z’ in files “testu1” and “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ tr '[a-z]' '[A-Z]' < testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

**ary9n@ary9n-virtual-machine:~/Desktop$ tr '[a-z]' '[A-Z]' < testu2**

**01:ACCOUNTS:6213:A**

**01:ACCOUNTS:6213:A**

**02:ADMIN:6403:B**

**03:MARKETING:6521:C**

**03:MARKETING:6521:C**

**04:PERSONNEL:7630:D**

**05:PRODUCTION:8589:E**

**05:PRODUCTION:8589:E**

**06:SALES:9876:F**

1. Translate all occurrences of “:” with “|” in “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ tr ':' '|' < testu2**

**01|accounts|6213|a**

**01|accounts|6213|a**

**02|admin|6403|b**

**03|marketing|6521|c**

**03|marketing|6521|c**

**04|personnel|7630|d**

**05|production|8589|e**

**05|production|8589|e**

**06|sales|9876|f**

1. Delete all occurrences of character ‘b’ from the file “testu1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ tr -d 'b' < testu1**

**DCNII**

**ADVJAVAII**

**LINUXOS**

**SEII**

**WEBNETII**

**CPROG**

**DM**

**CG**

**COREJAVA**

**DBMSI**

**DCNII**

**LINUXOS**

**CPROG**

**COD**

**MICROPROC**

**WEBNETII**

**SEII**

**CG**

1. Display only last field of file “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep -o '[^:]\*$' testu2**

**a**

**a**

**b**

**c**

**c**

**d**

**e**

**e**

**f**

1. Display second and third field of file “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ cut -d':' -f 2,3 testu2**

**accounts:6213**

**accounts:6213**

**admin:6403**

**marketing:6521**

**marketing:6521**

**personnel:7630**

**production:8589**

**production:8589**

**sales:9876**

1. Display the characters from 1 to 4 and 12 to 14 of file “testu2”.

**ary9n@ary9n-virtual-machine:~/Desktop$ cut -c 1-4,12-14 testu2**

**01:a:62**

**01:a:62**

**02:a03:**

**03:mg:6**

**03:mg:6**

**04:pl:7**

**05:pon:**

**05:pon:**

**06:s76:**

1. Save the second and fourth field of file “testu2” in file “cut\_testu21”

**ary9n@ary9n-virtual-machine:~/Desktop$ cut -d':' -f 2,4 testu2 > cut\_testu21**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat cut\_testu21**

**accounts:a**

**accounts:a**

**admin:b**

**marketing:c**

**marketing:c**

**personnel:d**

**production:e**

**production:e**

**sales:f**

1. Save the first field of file “testu2” in file “cut\_testu22”

**ary9n@ary9n-virtual-machine:~/Desktop$ cut -d':' -f 1 testu2 > cut\_testu22**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat cut\_testu22**

**01**

**01**

**02**

**03**

**03**

**04**

**05**

**05**

**06**

1. Paste “cut\_testu22” and “cut\_testu21” vertically.

**ary9n@ary9n-virtual-machine:~/Desktop$ paste cut\_testu22 cut\_testu21**

**01 accounts:a**

**01 accounts:a**

**02 admin:b**

**03 marketing:c**

**03 marketing:c**

**04 personnel:d**

**05 production:e**

**05 production:e**

**06 sales:f**

1. Paste “cut\_testu22” and “cut\_testu21” vertically, fields should be separated by pipe.

**ary9n@ary9n-virtual-machine:~/Desktop$ paste -d'|' cut\_testu22 cut\_testu21**

**01|accounts:a**

**01|accounts:a**

**02|admin:b**

**03|marketing:c**

**03|marketing:c**

**04|personnel:d**

**05|production:e**

**05|production:e**

**06|sales:f**

1. Paste “cut\_testu22” and “cut\_testu21” sequentially.

**ary9n@ary9n-virtual-machine:~/Desktop$ paste cut\_testu22 cut\_testu21 | sort**

**01 accounts:a**

**01 accounts:a**

**02 admin:b**

**03 marketing:c**

**03 marketing:c**

**04 personnel:d**

**05 production:e**

**05 production:e**

**06 sales:f**

1. Save the contents of “cut\_test21” and “cut\_test22” in “cut\_test23”.

**ary9n@ary9n-virtual-machine:~/Desktop$ paste cut\_testu22 cut\_testu21 > cut\_testu23**

**ary9n@ary9n-virtual-machine:~/Desktop$ cat cut\_testu23**

**01 accounts:a**

**01 accounts:a**

**02 admin:b**

**03 marketing:c**

**03 marketing:c**

**04 personnel:d**

**05 production:e**

**05 production:e**

**06 sales:f**

**50 - Aryan Penikal**

**Linux Practical-5**

**Searching data in Files**

**Commands:**

1. **grep :** Searches a file for given pattern and given options. Options as mentioned below. It can also work with regular expressions.
2. **egrep** : Belongs to family of grep command. Works the same as grep -E.Faster than grep command. Most of the options are same as than in grep command
3. **fgrep** : To search for mixed character strings in a file. Used when multiple regular expressions are to be evaluated. Also used when multiple files are to be searched.Options are similar to ones in grep command since it belongs to same family.

|  |  |
| --- | --- |
| **Options for grep** | **Description** |
| -c | No of lines that match the pattern |
| -i | Ignores case for matching-n |
| -n | Displays matched lines and line numbers |
| -v | Prints all lines that do not match a pattern |
| -e | Used when multiple patterns are to be matched |
| -f | Takes patterns from a file |
| -E | Takes pattern as an extended regular expression |
| -o | Print only matched parts of a line |

**Commands used in this practical are as follows:-**

**( grep, egrep and its options, character class of grep and egrep)**

Create a file **with the name Student** with the following fields separated by a **blank space** having the below mentioned values:

**Field Roll No First Name Last Name Date of Birth Marks (out of 600)**

**Values Numeric Character Character dd-mm-yy Numeric**

**Insert at least 5 records in this file.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ cat > student

52 Aryan Penikal 18-12-2002 589

12 Sagun Chatla 19-07-2002 550

1 Tom Holland 30-05-200 475

33 Arnold Schwarzenegger 03-01-2000 299

24 John Cena 08-10-1999 400

**Write and execute the commands for the following using grep with character class:**

1. **Search for the First Name as “Tom”.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep -i Tom student

1 Tom Holland 30-05-2000 475

1. **Search for the four letter First Name only.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ cut -d" " -f 2 student|grep ^....$ > pat2

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep -f pat2 student

24 John Cena 08-10-1999 400

1. **Search for the Roll No starting with “1”.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep ^1 student

12 Sagun Chatla 19-07-2002 550

1 Tom Holland 30-05-2000 475

1. **Search for the marks with last digit as “0”.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ cut -d" " -f 5 student|grep [\*0] >> pat3\

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep -f pat3 student

12 Sagun Chatla 19-07-2002 550

24 John Cena 08-10-1999 400

1. **Search for the name with first letter in First Name is ‘A’ and last character is ‘r’ in Last Name.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ cut -d" " -f 2,3 student|grep ^[A].\*[r]$ >> pat4

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep -f pat4 student

33 Arnold Schwarzenegger 03-01-2000 299

1. **Search for the Roll No that are not stating with “2”.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep ^[^2] student

52 Aryan Penikal 18-12-2002 589

12 Sagun Chatla 19-07-2002 550

1 Tom Holland 30-05-2000 475

33 Arnold Schwarzenegger 03-01-2000 299

1. **Search for the date of birth in which year ends with “0”.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ cut -d" " -f 4 student|grep .\*[0]$ > pat5

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep -f pat5 student

1 Tom Holland 30-05-2000 475

33 Arnold Schwarzenegger 03-01-2000 299

1. **Search for the First Name which starts with an uppercase letter vowel**.

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ cut -d" " -f 2 student|grep ^[AEIOU] > pat6

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep -f pat6 student

52 Aryan Penikal 18-12-2002 589

33 Arnold Schwarzenegger 03-01-2000 299

**Create the file with the name “test\_u4” with the following contents:-**

01**:**accounts**:**6213**:**a

06**:**sales**:**6213**:**f

05**:**production**:**5489**:**e

04**:**personnel**:**7630**:**a

02**:**admin**:**6521**:**b

01**:**marketing**:**6521**:**c

07**:**sales**:**6135**:**a

01**:**accounts**:**6003**:**h

10**:**marketing**:**6215**:**j

11**:**production**:**3480**:**i

14**:**Personnel**:**7306**:**k

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ gedit test\_u4

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ cat test\_u4

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

**Write and execute the commands for the following using grep:-**

1. **Display the records containing “accounts”.**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep accounts test\_u4

01:accounts:6213:a

01:accounts:6003:h

1. **Display all records containing “personnel”.**

ary9n@ary9n-virtual-machine:~$ grep personnel test\_u4

04:personnel:7630:a

1. **Display the record of “Admin”. Observe the output?**

ary9n@ary9n-virtual-machine:~/Desktop/prac5$ grep Admin test\_u4

ary9n@ary9n-virtual-machine:~/Desktop/prac5$

1. **Count the occurrence of “production”.**

ary9n@ary9n-virtual-machine:~$ grep -c production test\_u4

2

1. **Display all the records of “marketing” along with line numbers.**

ary9n@ary9n-virtual-machine:~$ grep -n marketing test\_u4

7:01:marketing:6521:c

12:10:marketing:6215:j

1. **Display all the records except “sales”.**

ary9n@ary9n-virtual-machine:~$ grep -v sales test\_u4

01:accounts:6213:a

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

1. **Display the records of “accounts” and “admin”.**

ary9n@ary9n-virtual-machine:~$ grep -e accounts -e admin test\_u4

01:accounts:6213:a

02:admin:6521:b

01:accounts:6003:h

1. **Display the filenames containing “marketing”.**

ary9n@ary9n-virtual-machine:~$ grep -l marketing \*

test\_u4

1. **Store the patterns “admin”, “production” and “sales” in new file and display the records containing these patterns.**

ary9n@ary9n-virtual-machine:~$ cat > pat1

admin

production

sales

ary9n@ary9n-virtual-machine:~$ grep -f pat1 test\_u4

06:sales:6213:f

05:production:5489:e

02:admin:6521:b

07:sales:6135:a

11:production:3480:i

1. **Display all the records not containing “marketing” along with line numbers.**

ary9n@ary9n-virtual-machine:~$ grep -nv marketing test\_u4

1:01:accounts:6213:a

2:06:sales:6213:f

3:05:production:5489:e

4:

5:04:personnel:7630:a

6:02:admin:6521:b

8:

9:07:sales:6135:a

10:01:accounts:6003:h

11:

13:11:production:3480:i

14:14:Personnel:7306:k

15:

1. **Count all the records not containing “production”.**

ary9n@ary9n-virtual-machine:~$ grep -cv production test\_u4

13

1. **Display all the records of “admin” along with line numbers.**

ary9n@ary9n-virtual-machine:~$ grep -n admin test\_u4

6:02:admin:6521:b

1. **Display all the record except “admin” along with line numbers.**

ary9n@ary9n-virtual-machine:~$ grep -nv admin test\_u4

1:01:accounts:6213:a

2:06:sales:6213:f

3:05:production:5489:e

4:

5:04:personnel:7630:a

7:01:marketing:6521:c

8:

9:07:sales:6135:a

10:01:accounts:6003:h

11:

12:10:marketing:6215:j

13:11:production:3480:i

14:14:Personnel:7306:k

15:

1. **Display all records starting with “1”.**

ary9n@ary9n-virtual-machine:~$ grep ^1 test\_u4

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

1. **Display all records not starting with “1”.**

ary9n@ary9n-virtual-machine:~$ grep ^[^1] test\_u4

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

1. **Display all records starting with “1” along with line numbers.**

ary9n@ary9n-virtual-machine:~$ grep -n ^1 test\_u4

12:10:marketing:6215:j

13:11:production:3480:i

14:14:Personnel:7306:k

1. **Display all records not starting with “1” along with line numbers.**

ary9n@ary9n-virtual-machine:~$ grep -n ^[^1] test\_u4

1:01:accounts:6213:a

2:06:sales:6213:f

3:05:production:5489:e

5:04:personnel:7630:a

6:02:admin:6521:b

7:01:marketing:6521:c

9:07:sales:6135:a

10:01:accounts:6003:h

1. **Display all records end with “a”.**

ary9n@ary9n-virtual-machine:~$ grep a$ test\_u4

01:accounts:6213:a

04:personnel:7630:a

07:sales:6135:a

1. **Display all records not end with “a”.**

ary9n@ary9n-virtual-machine:~$ grep [^a]$ test\_u4

06:sales:6213:f

05:production:5489:e

02:admin:6521:b

01:marketing:6521:c

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

1. **Display all records containing nothing along with line numbers.**

ary9n@ary9n-virtual-machine:~$ grep -n ^$ test\_u4

4:

8:

11:

15:

1. **Display all the lines with first character as “6” followed by any five characters at the end.**

ary9n@ary9n-virtual-machine:~$ grep 6.....$ test\_u4

01:accounts:6213:a

06:sales:6213:f

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

10:marketing:6215:j

1. **Display all the records starting with “0” followed by any three characters**

ary9n@ary9n-virtual-machine:~$ grep ^0... test\_u4

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

1. **Display all the records with first character not as “6” followed by any five characters at the end.**

ary9n@ary9n-virtual-machine:~$ grep [^6]..... test\_u4

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

1. **Display all the records not starting with “0” followed by any three characters.**

ary9n@ary9n-virtual-machine:~$ grep ^[^0]... test\_u4

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

**Write and execute the commands for the following using egrep and fgrep :**

1. **Display all the records with the names “sengupta”, “dasgupta” and “ramgupta” using egrep command.**

ary9n@ary9n-virtual-machine:~$ grep -E '(sengupta|dasgupta|ramgupta)' demo1

1 Aryan sengupta 52

2 Sagun dasgupta 12

3 Cristiano ramgupta 7

4 Kaka Sengupta 22

1. **Display all the records of “ramgupta” and “dasgupta” using egrep command.**

ary9n@ary9n-virtual-machine:~$ egrep '(das|ram)gupta' demo1

2 Sagun dasgupta 12

3 Cristiano ramgupta 7

1. **Display all the records of “sagun”, “cristiano” and “kaka” using fgrep command**.

ary9n@ary9n-virtual-machine:~$ fgrep 'sagun

> cristiano

> kaka' demo1

2 Sagun dasgupta 12

3 Cristiano ramgupta 7

4 Kaka Sengupta 22

**Name: Aryan Penikal Roll No: 50**

**Linux Practical-6**

**SED editor**

**Theory:**

SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.

* SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).
* SED command in unix supports regular expression which allows it perform complex pattern matching.

**Syntax:**

* sed OPTIONS... [SCRIPT] [INPUTFILE...]

1. **Create the file with the name “sedfile” with the following contents:-**

**Carrots:veg:1.39:1:n**

**Milk:Dairy:1.89:2:n**

**Magazine:Sundry:3.50:1:y**

**Cheese:Dairy:4.39:1:n**

**Sandwich:Deli:3.89:2:y**

**Onions:Veg:0.89:6:n**

**Chicken:Meat:4.89:2:n**

**Newspaper:Sundry:1.00:1:y**

**Fish:Meat:3.79:3:n**

**Floorwax:Hshld:4.65:1:y**

**Melon:Fruit:1.98:3:n**

**Celery:Veg:1.79:1:n**

**Dairy:Dairy:Dairy**

**Veg:Veg:Veg**

**End:of:Data in Sed Editor**

**Write and execute the commands for the following using sed:**

1. **Quits after printing first six lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '6q' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

1. **Display line numbers from four to seven.**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '4,7p'

4 Cheese:Dairy:4.39:1:n

5 Sandwich:Deli:3.89:2:y

6 Onions:Veg:0.89:6:n

7

1. **Display only last line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '$p' sedfile

End:of:Data in Sed Editor

**or**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '$p'

18 End:of:Data in Sed Editor

1. **Display only line number 8.**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '8p'

8 Chicken:Meat:4.89:2:n

1. **Display lines containing “Onions”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Onions/p' sedfile

Onions:Veg:0.89:6:n

1. **Display all lines containing “Dairy”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Dairy/p' sedfile

Milk:Dairy:1.89:2:n

Cheese:Dairy:4.39:1:n

Dairy:Dairy:Dairy

1. **Delete line 2.**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '2!p'

1 Carrots:veg:1.39:1:n

3 Magazine:Sundry:3.50:1:y

4 Cheese:Dairy:4.39:1:n

5 Sandwich:Deli:3.89:2:y

6 Onions:Veg:0.89:6:n

7

8 Chicken:Meat:4.89:2:n

9 Newspaper:Sundry:1.00:1:y

10 Fish:Meat:3.79:3:n

11

12 Floorwax:Hshld:4.65:1:y

13 Melon:Fruit:1.98:3:n

14

15 Celery:Veg:1.79:1:n

16 Dairy:Dairy:Dairy

17 Veg:Veg:Veg

18 End:of:Data in Sed Editor

1. **Delete lines 7 to 10.**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '7,10!p'

1 Carrots:veg:1.39:1:n

2 Milk:Dairy:1.89:2:n

3 Magazine:Sundry:3.50:1:y

4 Cheese:Dairy:4.39:1:n

5 Sandwich:Deli:3.89:2:y

6 Onions:Veg:0.89:6:n

11

12 Floorwax:Hshld:4.65:1:y

13 Melon:Fruit:1.98:3:n

14

15 Celery:Veg:1.79:1:n

16 Dairy:Dairy:Dairy

17 Veg:Veg:Veg

18 End:of:Data in Sed Editor

1. **Delete lines containing “Meat”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Meat/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Newspaper:Sundry:1.00:1:y

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Display lines start at 3 through the first line matching the string “Meat”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '3,/Meat/p' sedfile

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

1. **Delete lines start at 4 through the first line matching the string “Fruit”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '4,/Fruit/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Display lines number at interval of 2 i.e. starting at 1, every other line is output.**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '1~2p'

1 Carrots:veg:1.39:1:n

3 Magazine:Sundry:3.50:1:y

5 Sandwich:Deli:3.89:2:y

7

9 Newspaper:Sundry:1.00:1:y

11

13 Melon:Fruit:1.98:3:n

15 Celery:Veg:1.79:1:n

17 Veg:Veg:Veg

1. **Display lines starting at 2 and outputs every third line from there.**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '2~3p'

2 Milk:Dairy:1.89:2:n

5 Sandwich:Deli:3.89:2:y

8 Chicken:Meat:4.89:2:n

11

14

17 Veg:Veg:Veg

1. **Display lines starting with “S”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/^S/p' sedfile

Sandwich:Deli:3.89:2:y

1. **Display lines ends with “n”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/n$/p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Cheese:Dairy:4.39:1:n

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Fish:Meat:3.79:3:n

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

1. **Display lines that are not starting with “C”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/^C/!p' sedfile

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Display lines that are not ending with “y”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/y$/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Cheese:Dairy:4.39:1:n

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Fish:Meat:3.79:3:n

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Display all lines that are not starting with “v” or “V” in second field.**

ary9n@ary9n-virtual-machine:~/Desktop$ cut -d":" -f 2 sedfile | sed -n '/[v,V]/!p'

Dairy

Sundry

Dairy

Deli

Meat

Sundry

Meat

Hshld

Fruit

Dairy

of

1. **Display all lines that are not ending with “y” in second field.**

ary9n@ary9n-virtual-machine:~/Desktop$ cut -d":" -f 2 sedfile | sed -n '/[y$]/!p'

veg

Deli

Veg

Meat

Meat

Hshld

Fruit

Veg

Veg

of

1. **Delete lines containing nothing.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '/^$/d' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Delete all lines containing “2” in fourth field.**

ary9n@ary9n-virtual-machine:~/Desktop$ cut -d":" -f4 sedfile | sed '/2/d'

1

1

1

6

1

3

1

3

1

1. **Delete all lines containing “y” in fifth field.**

ary9n@ary9n-virtual-machine:~/Desktop$ cut -d":" -f5 sedfile | sed '/y/d'

n

n

n

n

n

n

n

n

1. **Delete all lines that do not start with C.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '/^C/d' sedfile

Carrots:veg:1.39:1:n

Cheese:Dairy:4.39:1:n

Chicken:Meat:4.89:2:n

Celery:Veg:1.79:1:n

1. **Display all lines that are not empty.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/^$/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Delete any other line that does not contain the string “Fish”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '/Fish/d' sedfile

Fish:Meat:3.79:3:n

1. **Display lines between the strings “Milk” and “Newspaper”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Milk/,/Newspaper/p' sedfile

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

1. **Display lines between the string “Sundry” and line number 11.**

ary9n@ary9n-virtual-machine:~/Desktop$ cat -n sedfile | sed -n '/Sundry/,11p'

3 Magazine:Sundry:3.50:1:y

4 Cheese:Dairy:4.39:1:n

5 Sandwich:Deli:3.89:2:y

6 Onions:Veg:0.89:6:n

7

8 Chicken:Meat:4.89:2:n

9 Newspaper:Sundry:1.00:1:y

10 Fish:Meat:3.79:3:n

11

1. **Replace only first occurrence of “Dairy” with “DAIRY” in all lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed 's/Dairy/DAIRY/' sedfile

Carrots:veg:1.39:1:n

Milk:DAIRY:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:DAIRY:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

DAIRY:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

**or**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n 's/Dairy/DAIRY/p' sedfile

Milk:DAIRY:1.89:2:n

Cheese:DAIRY:4.39:1:n

DAIRY:Dairy:Dairy

1. **Replace only first occurrence of “3” with “4” in all lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed 's/3/4/' sedfile

Carrots:veg:1.49:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:4.50:1:y

Cheese:Dairy:4.49:1:n

Sandwich:Deli:4.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:4.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:4:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

**or**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n 's/3/4/p' sedfile

Carrots:veg:1.49:1:n

Magazine:Sundry:4.50:1:y

Cheese:Dairy:4.49:1:n

Sandwich:Deli:4.89:2:y

Fish:Meat:4.79:3:n

Melon:Fruit:1.98:4:n

1. **Replace blank lines with “BLANK LINE” in all lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed 's/^$/BLANK LINE/' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

BLANK LINE

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

BLANK LINE

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

BLANK LINE

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Replace all occurrences of “Dairy” with “DAIRY” in all lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n 's/Dairy/DAIRY/gp' sedfile

Milk:DAIRY:1.89:2:n

Cheese:DAIRY:4.39:1:n

DAIRY:DAIRY:DAIRY

1. **Replace all occurrences of “3” with “4” in all lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n 's/3/4/gp' sedfile

Carrots:veg:1.49:1:n

Magazine:Sundry:4.50:1:y

Cheese:Dairy:4.49:1:n

Sandwich:Deli:4.89:2:y

Fish:Meat:4.79:4:n

Melon:Fruit:1.98:4:n

1. **Replace only first occurrence of “Veg” with “VEG” in lines 5 to 17.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '5,17s/Veg/VEG/p' sedfile

Onions:VEG:0.89:6:n

Celery:VEG:1.79:1:n

VEG:Veg:Veg

1. **Replace all occurrences of “Veg” with “VEG” in lines 5 to 17.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '5,17s/Veg/VEG/gp' sedfile

Onions:VEG:0.89:6:n

Celery:VEG:1.79:1:n

VEG:VEG:VEG

1. **Replace all occurrences of “9” with “1” in lines 3 to 8 and print only modified lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '3,8s/9/1/gp' sedfile

Cheese:Dairy:4.31:1:n

Sandwich:Deli:3.81:2:y

Onions:Veg:0.81:6:n

Chicken:Meat:4.81:2:n

1. **Replace only first occurrence of “e” with”EEE” only in line 4 and print only this line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '4s/e/EEE/p' sedfile

ChEEEese:Dairy:4.39:1:n

1. **Replace all occurrences of “:” with”|” only in last line and print only this line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '$s/:/|/gp' sedfile

End|of|Data in Sed Editor

1. **Replace all occurrences of “y” with “yes” on lines between “Deli” and “Fruit” and print only modified lines.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Deli/,/Fruit/s/y/yes/gp' sedfile

Sandwich:Deli:3.89:2:yes

Newspaper:Sundryes:1.00:1:yes

Floorwax:Hshld:4.65:1:yes

1. **Replace all occurrences of “n” with “NO” in lines containing “Meat”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Meat/s/n/NO/gp' sedfile

ChickeNO:Meat:4.89:2:NO

Fish:Meat:3.79:3:NO

1. **Replace all occurrences of “y” with “YES” except lines between “Cheese” and “Newspaper”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Cheese/,/Newspaper/!s/y/YES/gp' sedfile

Milk:DairYES:1.89:2:n

Magazine:SundrYES:3.50:1:YES

Floorwax:Hshld:4.65:1:YES

CelerYES:Veg:1.79:1:n

DairYES:DairYES:DairYES

1. **Replace all occurrences of “:” with “~” in all other lines except lines containing “Veg”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Veg/!s/:/~/gp' sedfile

Carrots~veg~1.39~1~n

Milk~Dairy~1.89~2~n

Magazine~Sundry~3.50~1~y

Cheese~Dairy~4.39~1~n

Sandwich~Deli~3.89~2~y

Chicken~Meat~4.89~2~n

Newspaper~Sundry~1.00~1~y

Fish~Meat~3.79~3~n

Floorwax~Hshld~4.65~1~y

Melon~Fruit~1.98~3~n

Dairy~Dairy~Dairy

End~of~Data in Sed Editor

1. **Print the line number along with the line that contains the string “Floorwax”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '/Floorwax/=' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

12

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Print only line number for a line that contains the string “Floorwax”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Floorwax/=' sedfile

12

1. **Inserts line “Linux Practical” before first line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '1 i Linux Practical' sedfile

Linux Practical

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Insert line “\*\*\*\*\*\*\*\*\*\*\*\*” before every line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed 'i \*\*\*\*\*\*\*\*\*\*\*\* ' sedfile

\*\*\*\*\*\*\*\*\*\*\*\*

Carrots:veg:1.39:1:n

\*\*\*\*\*\*\*\*\*\*\*\*

Milk:Dairy:1.89:2:n

\*\*\*\*\*\*\*\*\*\*\*\*

Magazine:Sundry:3.50:1:y

\*\*\*\*\*\*\*\*\*\*\*\*

Cheese:Dairy:4.39:1:n

\*\*\*\*\*\*\*\*\*\*\*\*

Sandwich:Deli:3.89:2:y

\*\*\*\*\*\*\*\*\*\*\*\*

Onions:Veg:0.89:6:n

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

Chicken:Meat:4.89:2:n

\*\*\*\*\*\*\*\*\*\*\*\*

Newspaper:Sundry:1.00:1:y

\*\*\*\*\*\*\*\*\*\*\*\*

Fish:Meat:3.79:3:n

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

Floorwax:Hshld:4.65:1:y

\*\*\*\*\*\*\*\*\*\*\*\*

Melon:Fruit:1.98:3:n

\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*

Celery:Veg:1.79:1:n

\*\*\*\*\*\*\*\*\*\*\*\*

Dairy:Dairy:Dairy

\*\*\*\*\*\*\*\*\*\*\*\*

Veg:Veg:Veg

\*\*\*\*\*\*\*\*\*\*\*\*

End:of:Data in Sed Editor

1. **Appends line “Sed Editor” after fourth line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '4 a Sed Editor' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sed Editor

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Append after each line this line “-----------------------“.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed 'a -----------------------' sedfile

Carrots:veg:1.39:1:n

-----------------------

Milk:Dairy:1.89:2:n

-----------------------

Magazine:Sundry:3.50:1:y

-----------------------

Cheese:Dairy:4.39:1:n

-----------------------

Sandwich:Deli:3.89:2:y

-----------------------

Onions:Veg:0.89:6:n

-----------------------

-----------------------

Chicken:Meat:4.89:2:n

-----------------------

Newspaper:Sundry:1.00:1:y

-----------------------

Fish:Meat:3.79:3:n

-----------------------

-----------------------

Floorwax:Hshld:4.65:1:y

-----------------------

Melon:Fruit:1.98:3:n

-----------------------

-----------------------

Celery:Veg:1.79:1:n

-----------------------

Dairy:Dairy:Dairy

-----------------------

Veg:Veg:Veg

-----------------------

End:of:Data in Sed Editor

-----------------------

1. **Inserts three lines “Sed Editor...”, “This is Linux Practical…” and “This is seventh practical…” before fifth line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '5 i Sed Editor...\

> This is Linux Practical…\

> This is seventh practical…' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sed Editor...

This is Linux Practical…

This is seventh practical…

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Appends three lines “Sed Editor…”, “This is Linux Practical…” and “This is seventh practical…” after seventh line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '7 a Sed Editor…\

> This is Linux Practical…\

> This is seventh practical…' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Sed Editor…

This is Linux Practical…

This is seventh practical…

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Insert line “Linux OS” before last line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '$ i Linux OS' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

Linux OS

End:of:Data in Sed Editor

1. **Append line “Linux OS” after last line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '$ a Linux OS' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

Linux OS

1. **Translate characters ‘D’,’O’,’V’ with ‘d’,’o’,’v’ respectively.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed 'y/DOV/dov/' sedfile

Carrots:veg:1.39:1:n

Milk:dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:dairy:4.39:1:n

Sandwich:deli:3.89:2:y

onions:veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:veg:1.79:1:n

dairy:dairy:dairy

veg:veg:veg

End:of:data in Sed Editor

1. **Print both the text and nonprintable ASCII characters.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n 'l' sedfile

Carrots:veg:1.39:1:n$

Milk:Dairy:1.89:2:n$

Magazine:Sundry:3.50:1:y$

Cheese:Dairy:4.39:1:n$

Sandwich:Deli:3.89:2:y$

Onions:Veg:0.89:6:n$

$

Chicken:Meat:4.89:2:n$

Newspaper:Sundry:1.00:1:y$

Fish:Meat:3.79:3:n$

$

Floorwax:Hshld:4.65:1:y$

Melon:Fruit:1.98:3:n$

$

Celery:Veg:1.79:1:n$

Dairy:Dairy:Dairy$

Veg:Veg:Veg$

End:of:Data\tin \tSed Editor$

1. **Write line 12 into new file “new1”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '12 w new1' sedfile

ary9n@ary9n-virtual-machine:~/Desktop$ cat new1

Floorwax:Hshld:4.65:1:y

1. **Write line 2 to 4 into new file “new2”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '2,4 w new2' sedfile

ary9n@ary9n-virtual-machine:~/Desktop$ cat new2

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

1. **Write lines containing text “Celery” into new file “new3”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Celery/ w new3' sedfile

ary9n@ary9n-virtual-machine:~/Desktop$ cat new3

Celery:Veg:1.79:1:n

1. **Write lines containing text between “Onions” and “Melon” into new file “new4”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Onions/,/Melon/ w new4' sedfile

ary9n@ary9n-virtual-machine:~/Desktop$ cat new4

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

1. **Write lines containing text between lines 2 to “Onions” into new file “new5”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '2,/Onions/ w new5' sedfile

ary9n@ary9n-virtual-machine:~/Desktop$ cat new5

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

1. **Write lines containing text between lines “Chicken” to 15 into new file “new6”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed -n '/Chicken/,15 w new6' sedfile

ary9n@ary9n-virtual-machine:~/Desktop$ cat new6

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

1. **Read the contents of file “data1 “after line 6.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '6 r data1' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Hello  
This is file data1

End of data 1

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1. **Read the contents of file “data2 “after last line.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '$ r data2' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

Hello  
This is file data2

End of data2

1. **Read the contents of file “names “after pattern “End”.**

ary9n@ary9n-virtual-machine:~/Desktop$ sed '/End/ r data2' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

Hello  
This is file names

End of names

**Name: Aryan Penikal Roll.no.:50**

**Linux Practical – 7**

**AWK editor**

**Theory:**

Awk is a scripting language used for manipulating data and generating reports. The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line. Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then perform the associated actions.

Awk is abbreviated from the names of the developers – Aho, Weinberger, and Kernighan.

**What can we do with awk?**

**1. AWK Operations:**   
(a) Scans a file line by line   
(b) Splits each input line into fields   
(c) Compares input line/fields to pattern   
(d) Performs action(s) on matched lines

**2. Useful For:**   
(a) Transform data files   
(b) Produce formatted reports

**3. Programming Constructs:**   
(a) Format output lines   
(b) Arithmetic and string operations   
(c) Conditionals and loops

**A)Create a file empdata, which contains the following fields:-**

**Fieldname Datatype Value**

**1.** Employee name character

**2.** Employee code numeric starts with letter ‘E’

**3.** Department code character MKT, HRD, PUR

**4.** Grade character A-C

**5.** Designation character manager, director, gm, executive

**6.** Years of experience numeric

**7.** Date of birth dd-mm-yy

**8.** Region character Pune, Mumbai etc…

**9.** Basic pay numeric

**Insert at least five records in above file; character fields in each record may not be same in the same case. ‘~’ is used as a field separator. Give commands for the following:-**

**cat empdata**

Messi~E1~HRD~A~gm~5~10-10-1987~Pune~15000

Ronaldo~E2~MKT~B~manager~6~1-2-1988~Mumbai~10000

Kaka~E3~PUR~C~director~7~22-1-1988~Pune~10000

Bale~E4~MKT~A~executive~6~10-9-1986~Mumbai~130000

Modric~E5~HRD~B~gm~9~3-2-1984~Pune~13000

Cruyff~E6~HRD~B~manager~10~30-11-1987~Pune~20000

Ibrahimovic~E7~HRD~C~gm~12~5-5-1985~Delhi~17000

Gullit~E8~PUR~A~executive~17~22-12-1977~Mumbai~120000

Maldini~E9~MKT~C~manager~4~12-12-1997~Mumbai~1800

Carlos~E10~HRD~A~executive~15~19-3-1973~Mumbai~180000

Neuer~E11~HRD~A~executive~9~30-11-1987~Pune~30000

**Display all employees who are manager in office.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$5 =="manager" {print $0}' empdata

Cruyff~E6~HRD~B~manager~10~30-11-1987~Pune~20000

Maldini~E9~MKT~C~manager~4~12-12-1997~Mumbai~1800

1. **List the details of an employee “Ronaldo” in Mumbai office.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$1 =="Ronaldo" && $8 =="Mumbai" {print $0}' empdata

Carlos~E10~HRD~A~executive~15~19-3-1973~Mumbai~180000

1. **Display all employees who are not in the department MKT. display the output sorted on department code.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$3 !="MKT" {print $0}' empdata | sort -t"~" -k 3

Carlos~E10~HRD~A~executive~15~19-3-1973~Mumbai~180000

Messi~E1~HRD~A~executive~5~13-12-2000~Mumbai~150000

Cruyff~E6~HRD~B~manager~10~30-11-1987~Pune~20000

Ibrahimovic~E7~HRD~C~gm~12~5-5-1985~Delhi~17000

Gullit~E8~PUR~A~executive~17~22-12-1977~Mumbai~120000

Bale~E4~PUR~B~gm~10~23-4-1990~Delhi~9000

Kaka~E3~PUR~C~director~7~22-1-1988~Pune~10000

1. **Display all employees whose years of experience are more than 5.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$6 > 5 {print $1}' empdata

Kaka

Bale

Modric

Cruyff

Ibrahimovic

Gullit

Ronaldo

1. **List only employee name, department code and basic pay of employees who are executive.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$5 == "executive" {print $1,$3,$9}' empdata

Messi HRD 150000

Modric MKT 200000

Gullit PUR 120000

Ronaldo HRD 180000

1. **Display all employees having grade ‘A’.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$4 == "A" {print $1}' empdata

Messi

Modric

Gullit

Ronaldo

1. **Count total number of employees whose department code is HRD.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$3 == "HRD"' empdata | wc -l

4

1. **Display employee’s names with salary above 10,000.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$9 > 10000 {print $1}' empdata

Messi

Ronaldo

Modric

Cruyff

Ibrahimovic

Gullit

Ronaldo

1. **Display only designation and basic pay of employees having number of experience between 3 and 5.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$6 >= 3 && $6 <= 5 {print $5,$9}' empdata

executive 150000

manager 1800

1. **Find the number of employees in Pune office.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$8 == "Pune"' empdata | wc -l

3

1. **Display employees who get basic pay less than 15000, also calculate and display average basic pay.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$9 < 15000 {print $0} {total +=$9;count++} END {print "Average salary="total/count}' empdata

Kaka~E3~PUR~C~director~7~22-1-1988~Pune~10000

Bale~E4~PUR~B~gm~10~23-4-1990~Delhi~9000

Maldini~E9~MKT~C~manager~4~12-12-1997~Mumbai~1800

Average salary=72280

1. **Locate the employees with same date of birth in Pune office.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$8=="Pune" && n=x[$7]{print n"\n"$0;} {x[$7]=$0}' empdata

Cruyff~E6~HRD~B~manager~10~30-11-1987~Pune~20000

Neuer~E11~HRD~A~executive~9~30-11-1987~Pune~30000

1. **Locate all for Bale, Ronaldo and Kaka as employee name.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$1=="Bale"||$1=="Ronaldo"||$1=="Kaka" {print $0}' empdata

**or**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$1~/Bale|Ronaldo|Kaka/ {print $0}' empdata

Ronaldo~E2~MKT~B~gm~5~10-10-1987~Pune~15000

Kaka~E3~PUR~C~director~7~22-1-1988~Pune~10000

Bale~E4~PUR~B~gm~10~23-4-1990~Delhi~9000

Carlos~E10~HRD~A~executive~15~19-3-1973~Mumbai~1800

1. **Locate all except for Bale, Ronaldo and Kaka as employee name.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$1!="Bale" && $1!="Ronaldo" && $1!="Kaka" {print $0}' empdata

**or**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '$1!~/Bale|Ronaldo|Kaka/ {print $0}' empdata

Messi~E1~HRD~A~executive~5~13-12-2000~Mumbai~150000

Modric~E5~MKT~A~executive~16~3-8-1980~Mumbai~200000

Cruyff~E6~HRD~B~manager~10~30-11-1987~Pune~20000

Ibrahimovic~E7~HRD~C~gm~12~5-5-1985~Delhi~17000

Gullit~E8~PUR~A~executive~17~22-12-1977~Mumbai~1200000

Maldini~E9~MKT~C~manager~4~12-12-1997~Mumbai~1800

Neuer~E11~HRD~A~executive~9~30-11-1987~Pune~30000

1. **Find the employees who have designation as director and find the 40% of basic pay as da and 15% of basic pay as hra.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" 'BEGIN{printf "Name\tECode\tDCode\tGrade\tDesig\t\tEXP\tDOB\t\tRegion\tPay\t\tDA\t\tHRA\t\n"}$5=="director" {da=0.4\*$9;hra=0.15\*$9}{printf $1"\t"$2"\t"$3"\t"$4"\t"$5"\t"$6"\t"$7"\t"$8"\t"$9"\t\t"da"\t\t"hra"\n"}' empdata

Name ECode DCode Grade Desig EXP DOB Region Pay DA HRA

Messi E1 HRD A executive 5 13-12-2000 Mumbai 150000

Ronaldo E2 MKT B gm 5 10-10-1987 Pune 15000

Kaka E3 PUR C director 7 22-1-1988 Pune 10000 4000 1500

Bale E4 PUR B gm 10 23-4-1990 Delhi 9000 4000 1500

Modric E5 MKT A executive 16 3-8-1980 Mumbai 200000 4000 1500

Cruyff E6 HRD B manager 10 30-11-1987 Pune 20000 4000 1500

Ibrahimovic E7 HRD C gm 12 5-5-1985 Delhi 17000 4000 1500

Gullit E8 PUR A executive 17 22-12-1977 Mumbai 1200000 4000 1500

Maldini E9 MKT C manager 4 12-12-1997 Mumbai 1800 4000 1500

Ronaldo E10 HRD A executive 15 19-3-1973 Mumbai 180000 4000 1500

Neuer E11 HRD A executive 9 30-11-1987 Pune 30000 4000 1500

1. **Store employee name and date of birth in a file’ nbdata’**.

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F"~" '{print $1,$7}' empdata > nbdata

ary9n@ary9n-virtual-machine:~/Desktop$ cat nbdata

Messi 13-12-2000

Ronaldo 10-10-1987

Kaka 22-1-1988

Bale 23-4-1990

Modric 3-8-1980

Cruyff 30-11-1987

Ibrahimovic 5-5-1985

Gullit 22-12-1977

Maldini 12-12-1997

Ronaldo 19-3-1973

Neuer 30-11-1987

**B)**

**Create a file student with following fields:-**

**Fieldname Datatype Value**

Student code numeric

Student name character

Batch code character B11-B15

No. of modules numeric 1-5

Average marks numeric

**Insert at least five records in above file; ‘:’ is used as a field separator. Give commands for the following:-**

1. **Display the details of student in order of their names ignoring case.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '{print $0}' student | sort -t":" -k 2 -f

S12:Abel:B11:4:50

S5:Abel:B14:2:45

S9:alLy:B13:4:77

S1:An:B11:5:70

S4:anika:B13:1:30

S7:Hana:B13:5:90

S2:Joe:B12:2:40

S10:Liya:B11:1:50

S11:Naina:B14:3:65

S3:Naina:B15:4:60

S6:Sandra:B12:3:66

S8:Yukta:B15:3:57

1. **Display the details of students whose number of modules is greater than 3.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '$4>3 {print $0}' student

S1:An:B11:5:70

S3:Naina:B15:4:60

S7:Hana:B13:5:90

S9:alLy:B13:4:77

S12:Abel:B11:4:50

1. **Store the list of rank holders in file ‘merit’ along with student code and student name, and marks & display its contents.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '{print $1,$2,$5}' student | sort -t" " -k 3 -nr > merit

ary9n@ary9n-virtual-machine:~/Desktop$ cat merit

S7 Hana 90

S9 alLy 77

S1 An 70

S6 Sandra 66

S11 Naina 65

S3 Naina 60

S8 Yukta 57

S12 Abel 50

S10 Liya 50

S5 Abel 45

S2 Joe 40

S4 anika 30

1. **Count the number of students in batch B13.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '$3=="B13"{count++}END{printf "The number of students in batch 13 is %d\n",count}' student

The number of students in batch 13 is 3

1. **Display the names of students with same names.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'n=x[$2] {print n"\n"$2}{x[$2]=$2}' student

Naina

Naina

Abel

Abel

1. **Display the students belonging to batch codes B12 or B15.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '$3~/B12|B13/ {printf "Student: %s\tBatch:%s\n",$2,$3}' student

Student: Joe Batch:B12

Student: anika Batch:B13

Student: Sandra Batch:B12

Student: Hana Batch:B13

Student: alLy Batch:B13

1. **Display all the names not starting with ‘a’ or ‘A’.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Names not starting with 'a' or 'A':\n"} $2~/^[^aA]/ {printf "%s\n",$2}' student

Names not starting with a or A:

Joe

Naina

Sandra

Hana

Yukta

Liya

Naina

1. **Display all the names starting with ‘a’ or ‘A’.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Names starting with 'a' or 'A':\n"} $2~/^[aA]/ {printf "%s\n",$2}' student

Names starting with a or A:

An

anika

Abel

alLy

Abel

1. **Display and count the number of students having marks in the range 40 to 60. Also display the total and average marks.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf"Students who scored marks between 40 and 50:\n"}$5>=40 && $5<=50 {print $0;count++;total+=$5;avg=total/count}END{printf "Count: %d\nTotal marks: %d\nAverage Marks: %d\n",count,total,avg}' student

Students who scored marks between 40 and 50:

S2:Joe:B12:2:40

S5:Abel:B14:2:45

S10:Liya:B11:1:50

S12:Abel:B11:4:50

Count: 4

Total marks: 185

Average Marks: 46

1. **Display the student’s records from line number 2 to 4.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'FNR>=2 && FNR<=4 {print $0}' student

S2:Joe:B12:2:40

S3:Naina:B15:4:60

S4:anika:B13:1:30

1. **Display the student’s records that are having number of fields 5.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'NF==5 {print $0}' student

S1:An:B11:5:70

S2:Joe:B12:2:40

S3:Naina:B15:4:60

S4:anika:B13:1:30

S5:Abel:B14:2:45

S6:Sandra:B12:3:66

S7:Hana:B13:5:90

S8:Yukta:B15:3:57

S9:alLy:B13:4:77

S10:Liya:B11:1:50

S11:Naina:B14:3:65

S12:Abel:B11:4:50

1. **Display the student’s records that are having number of fields less than or equal to 4.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'NF<=4 {print $0}' student

S13:Jane:4

S14:Rene:B12:1

1. **Display the student code, student name and marks that are having number of fields greater than 5.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'NF>5 {print $1,$2,$5}' student

S15 Jenna 65

1. **Display the student’s name having the length greater than 3.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '$2~/..../ {print $2}' student

**or**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'length($2)>3 {print $2}' student

Naina

anika

Abel

Sandra

Hana

Yukta

alLy

Liya

Naina

Abel

Jane

Rene

Jenna

1. **Display the student’s records having the length of student name less than or equal to 3.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'length($2)<=3 {print $0}' student

S1:An:B11:5:70

S2:Joe:B12:2:40

1. **Display the student’s records having the length less than or equal to 15.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'length($0)<=15 {print $0}' student

S1:An:B11:5:70

S2:Joe:B12:2:40

S13:Jane:4

S14:Rene:B12:1

1. **Display the student’s records having the length greater than or equal to 15.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'length($0)>=15 {print $0}' student

S2:Joe:B12:2:40

S3:Naina:B15:4:60

S4:anika:B13:1:30

S5:Abel:B14:2:45

S6:Sandra:B12:3:66

S7:Hana:B13:5:90

S8:Yukta:B15:3:57

S9:alLy:B13:4:77

S10:Liya:B11:1:50

S11:Naina:B14:3:65

S12:Abel:B11:4:50

S15:Jenna:B14:1:65:Marbles

1. **Display the student’s records having the length is in the range 5 to 15.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'length($0)>=5 && length($0)<=15 {print $0}' student

S1:An:B11:5:70

S2:Joe:B12:2:40

S13:Jane:4

S14:Rene:B12:1

1. **Display the line number and index having character ‘b’ in student name.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '$2~/b/ {print NR" "$0" Index:" index($2,"b")}' student

5 S5:Abel:B14:2:45 Index:2

12 S12:Abel:B11:4:50 Index:2

1. **Display the line number, student name and index having character ‘b’ in student name.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" '$2~/b/ {print NR" "$2" Index:" index($2,"b")}' student

5 Abel Index:2

12 Abel Index:

1. **Display the index, student name and marks having character ‘b’ in student name.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Line Number:\t Name:\t Marks:\t\n"}$2~/b/ {print NR" \t\t "$2,"\t",$5}' student

Line Number: Name: Marks:

5 Abel 45

12 Abel 50

1. **Display the line number, index, student name and marks having character ‘b’ in student name.**

ary9n@ary9n-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Line Number:\t Name:\t Marks:\t Index of 'b' in Name:\n"}$2~/b/ {print NR" \t\t "$2,"\t",$5,"\t",index($2,"b")}' student

Line Number: Name: Marks: Index of b in Name:

5 Abel 45 2

12 Abel 50 2

**Name:** Aryan Penikal  **Roll No:** 50

**Practical 9 Group and User Management**

**Theory:**

In Linux, users and groups can be created by any administrator using some simple commands.

User account information is stored in the /etc/passwd file. This information includes the account name, home directory location, and default shell, among other values.

Each field is separated by a : character, and not all fields must be populated, but you must delineate them.

Here's an example of the /etc/passwd fields:

username:password:UID:GID:comment:home:shell

Earlier in Linux, password hashes were stored in the /etc/passwd file. This file was world-readable, allowing inquisitive users to pull password hashes for other accounts from the file and run them through password-cracking utilities. Eventually, the password hashes were moved to a file readable only by root: /etc/shadow. Today, the password field in the /etc/passwd file is marked with an x.

Here's an example of /etc/shadow fields:

username:password:last password change:min:max:warning:inactive:expired

Similar to the /etc/passwd file above, the /etc/group file contains group account information. This information can be essential for troubleshooting, security audits, and ensuring users can access the resources they need.

The fields in the /etc/group file are:

groupname:password:GID:group members

There are various options to manage groups and users in linux. Some options that will be looked into are:

1. **groupadd :** To create new groups of users in linux. One can verify the creation by checking the /etc/group file’s content

**Options:**

|  |  |
| --- | --- |
| -g | Add Group id |
| -o | Used when duplicate group id is to be entered |
| -help | Display details of the command |

1. **groupmod :** To change the details of a particular group in linux. Again the verification of the /etc/group file can be done to check the changes have been made or not.

**Options:**

|  |  |
| --- | --- |
| -g | Add Group id |
| -o | Used when duplicate group id is to be entered |
| -n | Change the name of the group |
| -help | Display details of the command |

**3. gpasswd :** To add users to groups. Can also be used to set the password for the group when no options and only group name is provided as inputs.

**Options:**

|  |  |
| --- | --- |
| -a | Add a single user to the group at a time |
| -M | Add multiple users to the group at a time |
| -d | Remove user from group |
| -A | Make a user the group administrator |

1. **groupdel :** To delete groups
2. **user add :** To add users. Has tons of options with regards to creating users.

**Options:**

|  |  |
| --- | --- |
| -u | Add specific user id |
| -m -d | Used when a different home directory is provided |
| -g | User with given group id |
| -e | Set the account expiry date |
| -G | To add a new user to multiple groups |
| -M | Creating user without home directory |
| -s | Creating user with given login shell |
| -c | Creating user with custom comments |
| -f | Specifies no of days after which password expires |

1. **usermod :** To change details of the user. Same options will be used as with useradd.

Can be used with -L (Lock user account), or U (Unlock user account)

1. **adduser :** To create a user wherein all the data will be asked after the issuance of the command.
2. **chage** : To verify the details of the user. chage -l : Will display the long details of the specified user.
3. **passwd :** To set password for the user
4. **userdel -r :** To delete all users in a group

### Create a new group account with name “tycsg1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupadd tycsg1**

### Display the details of group “tycsg1”.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg1 /etc/group**

**tycsg1:x:1001:**

1. Create new group accounts with names “tycsg2”, “tycsg3”, “tycsg4”, “tycsg5” and “tycsg6”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupadd tycsg2**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupadd tycsg3**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupadd tycsg4**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupadd tycsg5**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupadd tycsg6**

1. Display the details of all above created groups.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ^tycs /etc/group**

**tycsg1:x:1001:**

**tycsg2:x:1002:**

**tycsg3:x:1003:**

**tycsg4:x:1004:**

**tycsg5:x:1005:**

**tycsg6:x:1006:**

### Create a new group “tycsg8” with specific unique groupid, and then display its details.

### ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg8 /etc/group

### tycsg8:x:1009:

### Create a new group “tycsg7” with duplicate groupid, and then display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupadd -o -g 1009 tycsg7**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ^tycs /etc/group**

**tycsg1:x:1001:**

**tycsg2:x:1002:**

**tycsg3:x:1003:**

**tycsg4:x:1004:**

**tycsg5:x:1005:**

**tycsg6:x:1006:**

**tycsg8:x:1009:**

**tycsg7:x:100**

### Change the name of group "tycsg1" to "newtycsg1", and then display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupmod -n newtycsg1 tycsg1**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ^newtycs /etc/group**

**newtycsg1:x:1001:**

1. Change groupid of group “newtycsg1”, and then display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupmod -g 1010 newtycsg1**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ^newtycs /etc/group**

**newtycsg1:x:1010:**

1. Use same groupid for other group “tycsg2” using –o option.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupmod -o -g 1010 tycsg2**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg2 /etc/group**

**tycsg2:x:1010:**

### Display help message for above commands and its option

### ary9n@ary9n-virtual-machine:~/Desktop$ groupadd --help

### Usage: groupadd [options] GROUP

### Options:

### -f, --force exit successfully if the group already exists,

### and cancel -g if the GID is already used

### -g, --gid GID use GID for the new group

### -h, --help display this help message and exit

### -K, --key KEY=VALUE override /etc/login.defs defaults

### -o, --non-unique allow to create groups with duplicate

### (non-unique) GID

### -p, --password PASSWORD use this encrypted password for the new group

### -r, --system create a system account

### -R, --root CHROOT\_DIR directory to chroot into

### -P, --prefix PREFIX\_DIR directory prefix

### --extrausers Use the extra users database

### ary9n@ary9n-virtual-machine:~/Desktop$ groupmod --help

### Usage: groupmod [options] GROUP

### Options:

### -g, --gid GID change the group ID to GID

### -h, --help display this help message and exit

### -n, --new-name NEW\_GROUP change the name to NEW\_GROUP

### -o, --non-unique allow to use a duplicate (non-unique) GID

### -p, --password PASSWORD change the password to this (encrypted)

### PASSWORD

### -R, --root CHROOT\_DIR directory to chroot into

### -P, --prefix PREFIX\_DIR prefix directory where are located the /etc/\* files

1. Add user “ty1” in group “tycsg3”, and then display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo gpasswd -a ty1 tycsg3**

**Adding user ty1 to group tycsg3**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg3 /etc/group**

**tycsg3:x:1003:ty1**

**ary9n@ary9n-virtual-machine:~/Desktop$**

### Add multiple users “ty2”, “ty3”, “ty4”, “ty5”, and “ty6” to the group “tycsg3”, and then display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo gpasswd -M ty2,ty3,ty4,ty5,ty6 tycsg3**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg3 /etc/group**

**tycsg3:x:1003:ty2,ty3,ty4,ty5,ty6**

### Remove user “ty2” from the group “tycsg3”, and then display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo gpasswd -d ty2 tycsg3**

**Removing user ty2 from group tycsg3**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg3 /etc/group**

**tycsg3:x:1003:ty3,ty4,ty5,ty6**

### Remove multiple users “ty3” and “ty4” from the group “tycsg3”, and then display its details.

### ary9n@ary9n-virtual-machine:~/Desktop$ sudo gpasswd -d ty3 tycsg3

### Removing user ty3 from group tycsg3

### ary9n@ary9n-virtual-machine:~/Desktop$ sudo gpasswd -d ty4 tycsg3

### Removing user ty4 from group tycsg3

### ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg3 /etc/group

### tycsg3:x:1003:ty5,ty6

### Make user which is a group member “ty5” of “tycsg4” as a group administrator.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo gpasswd -A ty5 tycsg3**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep tycsg3 /etc/group**

**tycsg3:x:1003:ty5,ty6**

### Set password for group “tycsg5”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo grep tycsg5 /etc/gshadow**

**tycsg5:$6$cubT//L2$lodbywbTZABuYfEtG3gm/BvgbMXCV4lk8QE5CIYK246D61y.tncGP1/QznGBRoIkPJLdFGScduddW.qYuaY.a0::**

### Display the details of group accounts and their passwords.

### ary9n@ary9n-virtual-machine:~/Desktop$ sudo cat /etc/gshadow

### root:\*::

### daemon:\*::

### bin:\*::

### sys:\*::

### adm:\*::syslog,ary9n

### tty:\*::

### disk:\*::

### lp:\*::

### mail:\*::

### news:\*::

### uucp:\*::

### man:\*::

### proxy:\*::

### kmem:\*::

### dialout:\*::

### fax:\*::

### voice:\*::

### cdrom:\*::ary9n

### floppy:\*::

### tape:\*::

### sudo:\*::ary9n

### audio:\*::pulse

### dip:\*::ary9n

### www-data:\*::

### backup:\*::

### operator:\*::

### list:\*::

### irc:\*::

### src:\*::

### gnats:\*::

### shadow:\*::

### utmp:\*::

### video:\*::

### sasl:\*::

### plugdev:\*::ary9n

### staff:\*::

### games:\*::

### users:\*::

### nogroup:\*::

### systemd-journal:!::

### systemd-network:!::

### systemd-resolve:!::

### crontab:!::

### messagebus:!::

### systemd-timesync:!::

### input:!::

### sgx:!::

### kvm:!::

### render:!::

### syslog:!::

### tss:!::

### bluetooth:!::

### ssl-cert:!::

### uuidd:!::

### systemd-oom:!::

### tcpdump:!::

### \_ssh:!::

### avahi-autoipd:!::

### netdev:!::

### avahi:!::

### lpadmin:!::ary9n

### rtkit:!::

### whoopsie:!::

### sssd:!::

### nm-openvpn:!::

### scanner:!::saned

### saned:!::

### colord:!::

### geoclue:!::

### pulse:!::

### pulse-access:!::

### gdm:!::

### lxd:!::ary9n

### ary9n:!::

### sambashare:!::ary9n

### tycsg2:!::

### tycsg3:!:ty5:ty5,ty6

### tycsg4:!:ty5:

### tycsg5:$6$cubT//L2$lodbywbTZABuYfEtG3gm/BvgbMXCV4lk8QE5CIYK246D61y.tncGP1/QznGBRoIkPJLdFGScduddW.qYuaY.a0::

### tycsg6:!::

### tycsg8:!::

### tycsg7:!::

### newtycsg1:!::

### ty1:!::

### ty2:!::

### ty3:!::

### ty4:!::

### ty5:!::

### ty6:!::

### Display help message for above command and its options.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo gpasswd --help**

**Usage: gpasswd [option] GROUP**

**Options:**

**-a, --add USER add USER to GROUP**

**-d, --delete USER remove USER from GROUP**

**-h, --help display this help message and exit**

**-Q, --root CHROOT\_DIR directory to chroot into**

**-r, --remove-password remove the GROUP's password**

**-R, --restrict restrict access to GROUP to its members**

**-M, --members USER,... set the list of members of GROUP**

**--extrausers use the extra users database**

**-A, --administrators ADMIN,...**

**set the list of administrators for GROUP**

**Except for the -A and -M options, the options cannot be combined.**

1. Delete the groups “newtycsg1” and “tycsg2”

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupdel newtycsg1**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo groupdel tycsg2**

### Display help message for above command and its options.

**ary9n@ary9n-virtual-machine:~/Desktop$ groupdel --help**

**Usage: groupdel [options] GROUP**

**Options:**

**-h, --help display this help message and exit**

**-R, --root CHROOT\_DIR directory to chroot into**

**-P, --prefix PREFIX\_DIR prefix directory where are located the /etc/\* files**

**-f, --force delete group even if it is the primary group of a user**

**--extrausers Use the extra users database**

### Create a new user in “ty1a”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd ty1a**

### Display the details of user “ty1a”.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty1a /etc/passwd**

**ty1a:x:1007:1007::/home/ty1a:/bin/sh**

### Create a new user “ty4a” with different home directory.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -m -d /home/ty2aa ty4a**

### Display the details of user “ty4a”.

**ary9n@ary9n-virtual-machine:~/Desktop$ ls /home**

**ary9n ty2a ty2aa**

### Create a new user “ty3b” with specific user-id.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -u 1040 ty3b**

### Display the details of user “ty3b”.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty3b /etc/passwd**

**ty3b:x:1040:1040::/home/ty3b:/bin/sh**

### Create a new user “ty4b” with specific group-id.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -g 1040 ty4b**

### Display the details of user “ty4b”.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty4b /etc/passwd**

**ty4b:x:1041:1040::/home/ty4b:/bin/sh**

### Create a new user “ty5a” to multiple groups “tycsg3”, “tycsg4” and “tycsg5”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -G tycsg3,tycsg4,tycsg5 ty5a**

1. Verify that the multiple groups assigned to the user “ty5a”. **ary9n@ary9n-virtual-machine:~/Desktop$ grep ty5a /etc/group**

**tycsg3:x:1003:ty5,ty6,ty5a**

**tycsg4:x:1004:ty5a**

**tycsg5:x:1005:ty5a**

**ty5a:x:1042:**

### Create a new user “ty6a” without home directory.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -M ty6a**

### Display the details of user “ty6a”.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty6a /etc/passwd**

**ty6a:x:1043:1043::/home/ty6a:/bin/sh**

### Create a new user “ty7a” with account expiry date.

### ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -e 2022-12-18 ty7a

### Verify the details of user “ty7a”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo chage -l ty7a**

**Last password change : Sep 25, 2022**

**Password expires : never**

**Password inactive : never**

**Account expires : Dec 18, 2022**

**Minimum number of days between password change : 0**

**Maximum number of days between password change : 99999**

**Number of days of warning before password expires : 7**

### Create a new user “ty8a” with password expiry date by defining the number of days after a password expires.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -e 2022-12-18 -f 30 ty8a**

### Verify the details of user “ty8a”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo chage -l ty8a**

**Last password change : Sep 25, 2022**

**Password expires : never**

**Password inactive : never**

**Account expires : Dec 18, 2022**

**Minimum number of days between password change : 0**

**Maximum number of days between password change : 99999**

**Number of days of warning before password expires : 7**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo grep ty8a /etc/shadow**

**ty8a:!:19260:0:99999:7:30:19344:**

1. Create a new user “ty9a”user with custom comments. Display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -c "Hello" ty9a**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo grep ty9a /etc/passwd**

**ty9a:x:1046:1046:Hello:/home/ty9a:/bin/sh**

1. Create a new user “ty10a” with login shell. Display its details. **ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -s /bin/sh ty10a**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty10a /etc/passwd**

**ty10a:x:1047:1047::/home/ty10a:/bin/sh**

### Create a new user “ty11a” with specific home directory, default shell and comment. Display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -m -d /home/ty11aa -s /bin/sh -c "Hello World" ty11a**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty11a /etc/passwd**

**ty11a:x:1048:1048:Hello World:/home/ty11aa:/bin/sh**

### Create a new user “ty12” with home directory, custom shell, comment and UID/GID. Display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo useradd -m -d /home/ty12aa -s /bin/bash -c "Hello World" -u 1050 -g 1048 ty12a**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty12a /etc/passwd**

**ty12a:x:1050:1048:Hello World:/home/ty12aa:/bin/bash**

1. Create a new user “ty13” with changed shell.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo adduser ty13a**

**Adding user `ty13a' ...**

**Adding new group `ty13a' (1010) ...**

**Adding new user `ty13a' (1010) with group `ty13a' ...**

**Creating home directory `/home/ty13a' ...**

**Copying files from `/etc/skel' ...**

**New password:**

**BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic**

**Retype new password:**

**Sorry, passwords do not match.**

**New password:**

**Retype new password:**

**passwd: password updated successfully**

**Changing the user information for ty13a**

**Enter the new value, or press ENTER for the default**

**Full Name []: Aryan**

**Room Number []: 510**

**Work Phone []: 1234567890**

**Home Phone []: 1234567789**

**Other []: 00000000**

**Is the information correct? [Y/n] ary9n@ary9n-virtual-machine:~/Desktop$ grep ty13a /etc/passwd**

**ty13a:x:1010:1010:Aryan,510,1234567890,1234567789,00000000:/home/ty13a:/bin/bash**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo usermod -s /bin/sh ty13a**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty13a /etc/passwd**

**ty13a:x:1010:1010:Aryan,510,1234567890,1234567789,00000000:/home/ty13a:/bin/sh**

1. Set password for all above created user accounts. **ary9n@ary9n-virtual-machine:~/Desktop$ sudo passwd ty12a**

**[sudo] password for ary9n:**

**New password:**

**Retype new password:**

**passwd: password updated successfully**

1. Display help for above command.

**ary9n@ary9n-virtual-machine:~/Desktop$ useradd -h**

**Usage: useradd [options] LOGIN**

**useradd -D**

**useradd -D [options]**

**Options:**

**--badnames do not check for bad names**

**-b, --base-dir BASE\_DIR base directory for the home directory of the**

**new account**

**--btrfs-subvolume-home use BTRFS subvolume for home directory**

**-c, --comment COMMENT GECOS field of the new account**

**-d, --home-dir HOME\_DIR home directory of the new account**

**-D, --defaults print or change default useradd configuration**

**-e, --expiredate EXPIRE\_DATE expiration date of the new account**

**-f, --inactive INACTIVE password inactivity period of the new account**

**-g, --gid GROUP name or ID of the primary group of the new**

**account**

**-G, --groups GROUPS list of supplementary groups of the new**

**account**

**-h, --help display this help message and exit**

**-k, --skel SKEL\_DIR use this alternative skeleton directory**

**-K, --key KEY=VALUE override /etc/login.defs defaults**

**-l, --no-log-init do not add the user to the lastlog and**

**faillog databases**

**-m, --create-home create the user's home directory**

**-M, --no-create-home do not create the user's home directory**

**-N, --no-user-group do not create a group with the same name as**

**the user**

**-o, --non-unique allow to create users with duplicate**

**(non-unique) UID**

**-p, --password PASSWORD encrypted password of the new account**

**-r, --system create a system account**

**-R, --root CHROOT\_DIR directory to chroot into**

**-P, --prefix PREFIX\_DIR prefix directory where are located the /etc/\* files**

**-s, --shell SHELL login shell of the new account**

**-u, --uid UID user ID of the new account**

**-U, --user-group create a group with the same name as the user**

**-Z, --selinux-user SEUSER use a specific SEUSER for the SELinux user mapping**

**--extrausers Use the extra users database**

**ary9n@ary9n-virtual-machine:~/Desktop$ useradd -h**

**Usage: useradd [options] LOGIN**

**useradd -D**

**useradd -D [options]**

**Options:**

**--badnames do not check for bad names**

**-b, --base-dir BASE\_DIR base directory for the home directory of the**

**new account**

**--btrfs-subvolume-home use BTRFS subvolume for home directory**

**-c, --comment COMMENT GECOS field of the new account**

**-d, --home-dir HOME\_DIR home directory of the new account**

**-D, --defaults print or change default useradd configuration**

**-e, --expiredate EXPIRE\_DATE expiration date of the new account**

**-f, --inactive INACTIVE password inactivity period of the new account**

**-g, --gid GROUP name or ID of the primary group of the new**

**account**

**-G, --groups GROUPS list of supplementary groups of the new**

**account**

**-h, --help display this help message and exit**

**-k, --skel SKEL\_DIR use this alternative skeleton directory**

**-K, --key KEY=VALUE override /etc/login.defs defaults**

**-l, --no-log-init do not add the user to the lastlog and**

**faillog databases**

**-m, --create-home create the user's home directory**

**-M, --no-create-home do not create the user's home directory**

**-N, --no-user-group do not create a group with the same name as**

**the user**

**-o, --non-unique allow to create users with duplicate**

**(non-unique) UID**

**-p, --password PASSWORD encrypted password of the new account**

**-r, --system create a system account**

**-R, --root CHROOT\_DIR directory to chroot into**

**-P, --prefix PREFIX\_DIR prefix directory where are located the /etc/\* files**

**-s, --shell SHELL login shell of the new account**

**-u, --uid UID user ID of the new account**

**-U, --user-group create a group with the same name as the user**

**-Z, --selinux-user SEUSER use a specific SEUSER for the SELinux user mapping**

**--extrausers Use the extra users database**

1. Change information of user account “ty9a”. Display its details. **ary9n@ary9n-virtual-machine:~/Desktop$ grep ty12a /etc/passwd**

**ty12a:x:1050:1048:Hello World:/home/ty12aa:/bin/bash**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo usermod -c 'updated' ty12a**

**[sudo] password for ary9n:**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty12a /etc/passwd**

**ty12a:x:1050:1048:updated:/home/ty12aa:/bin/bash**

### Change home directory of user account “ty10” to “ty100” in /home. Display its details.

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty10a /etc/passwd**

**ty10a:x:1047:1047::/home/ty10a:/bin/sh**

**ary9n@ary9n-virtual-machine:~/Desktop$ mkdir dir1**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo uermod -d /home/dir1 ty10a**

**sudo: uermod: command not found**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo usermod -d /home/dir1 ty10a**

**ary9n@ary9n-virtual-machine:~/Desktop$ grep ty10a /etc/passwd**

**ty10a:x:1047:1047::/home/dir1:/bin/sh**

### Set expiry date on user account “ty11”.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo chage -l ty11a**

**Last password change : Sep 25, 2022**

**Password expires : never**

**Password inactive : never**

**Account expires : never**

**Minimum number of days between password change : 0**

**Maximum number of days between password change : 99999**

**Number of days of warning before password expires : 7**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo usermod -e 2022-12-18 ty11a**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo chage -l ty11a**

**Last password change : Sep 25, 2022**

**Password expires : never**

**Password inactive : never**

**Account expires : Dec 18, 2022**

**Minimum number of days between password change : 0**

**Maximum number of days between password change : 99999**

**Number of days of warning before password expires : 7**

### Verify the details.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo chage -l ty11a**

**Last password change : Sep 25, 2022**

**Password expires : never**

**Password inactive : never**

**Account expires : Dec 18, 2022**

**Minimum number of days between password change : 0**

**Maximum number of days between password change : 99999**

**Number of days of warning before password expires : 7**

1. Change primary group of user “ty12a” to group “ty10a”. **ary9n@ary9n-virtual-machine:~$ sudousermod -g ty10a ty12a**

**ary9n@ary9n-virtual-machine:~$ id ty12a**

**uid=1050(ty12a) gid=1047(ty10a) groups=1047(ty10a)**

1. Add a new group “tycsg8” to an existing user “ty13a”. **ary9n@ary9n -virtual-machine:~$ sudousermod -G tycsg8 ty13a**

**ary9n@ary9n-virtual-machine:~$ id ty13a**

**uid=1007(ty13a) gid=1007(ty13a) groups=1007(ty13a),1027(tycsg8)**

### Add supplementary group “tycsg7” to user “ty10a” and primary group “ty10a” remains the same.

**Ary9n@ary9n-virtual-machine:~$ sudousermod -a -G tycsg7 ty12a Ary9n@ary9n-virtual-machine:~$ id ty12a**

**uid=1050(ty12a) gid=1047(ty10a) groups=1047(ty10a),1027(tycsg8) Ary9n@ary9n-virtual-machine:~$ grep tycsg7 /etc/group tycsg7:x:1027:ty12a**

1. Change login name of user “ty5” to “ty55”. Verify the details. **Ary9n@ary9n-virtual-machine:~$ sudousermod -l ty55a ty5a Ary9n@ary9n-virtual-machine:~$ id ty5a**

**id: ‘ty5a’: no such user**

**Ary9n@ary9n-virtual-machine:~$ id ty55a uid=1042(ty55a) gid=1042(ty5a)**

**groups=1042(ty5a),1023(tycsg3),1024(tycsg4),1025(tycsg5)**

1. Lock the user account “ty12a”. Display the details. **Ary9n@ary9n-virtual-machine:~$ sudousermod -L ty12a Ary9n@ary9n-virtual-machine:~$ sudo grep ty12a /etc/shadow**

**ty12a:!$6$iHjAlA2A$yBZrAHUQiT1BSeodwAu7G8bImQq5un2obEZYI/Vq0Sdwk OdedtohOsaPX0HHofTiytDABAu4xK7KbNMqA1w..1:18564:0:99999:7:::**

1. Unlock the user account “ty12a”. Display the details. **Ary9n@ary9n-virtual-machine:~$ sudousermod -U ty12a Ary9n@ary9n-virtual-machine:~$ sudo grep ty12a /etc/shadow**

**ty12a:$6$iHjAlA2A$yBZrAHUQiT1BSeodwAu7G8bImQq5un2obEZYI/Vq0SdwkO dedtohOsaPX0HHofTiytDABAu4xK7KbNMqA1w..1:18564:0:99999:7:::**

### Change user’s default home directory to a new location of user account “ty2a”.

**Ary9n@ary9n-virtual-machine:~$ sudousermod -d /var/ty2a -m ty2a Ary9n@ary9n-virtual-machine:~$ ls /var**

**backups crash local log metrics run spool ty2a cache lib lock mail opt snap tmp www Ary9n@ary9n-virtual-machine:~$ grep ty2a /etc/passwd ty2a:x:1031:1036::/var/ty2a:**

1. Change shell to /bin/zsh of user account “ty10a”.

**Ary9n@ary9n-virtual-machine:~$ sudousermod -s /bin/zsh ty10a**

**Ary9n@ary9n-virtual-machine:~$ grep ty10a /etc/passwd ty10a:x:1047:1047::/home/dir1:/bin/zsh**

1. Display the id of user account “ty11a”.

**Ary9n@ary9n-virtual-machine:~$ id ty11a uid=1048(ty11a) gid=1048(ty11a) groups=1048(ty11a)**

1. Change user id of user account “ty11a”. **Ary9n@ary9n-virtual-machine:~$ id ty11a uid=1048(ty11a) gid=1048(ty11a) groups=1048(ty11a)**

**Ary9n@ary9n-virtual-machine:~$ sudousermod -u 1084 ty11a Ary9n@ary9n-virtual-machine:~$ id ty11a**

**uid=1084(ty11a) gid=1048(ty11a) groups=1048(ty11a)**

### Modify account details with multiple options of user account “ty12a”.Verify the details.

**Ary9n@ary9n-virtual-machine:~$ sudousermod -d /var/dir1 -e 2020-10-30 -s /bin/sh - c "Ty12 user" -u 1051 -g 1037 -a -G tycsg7 ty12a**

**Ary9n@ary9n-virtual-machine:~$ grep ty12a /etc/passwd ty12a:x:1051:1037:Ty12 user:/var/dir1:/bin/sh Ary9n@ary9n-virtual-machine:~$ id ty12a**

**uid=1051(ty12a) gid=1037(ty3a) groups=1037(ty3a),1027(tycsg8)**

### Display the original user id and group id of user account “ty13a”.

**Ary9n@ary9n-virtual-machine:~$ id ty13a**

**uid=1007(ty13a) gid=1007(ty13a) groups=1007(ty13a),1027(tycsg8)**

### Change UID and GID of the user “ty13a”.

**Ary9n@ary9n-virtual-machine:~$ sudousermod -u 1052 -g 1037 ty13a**

### Now, display the user id and group id of user account “ty13a”.

**Ary9n@ary9n-virtual-machine:~$ id ty13a**

**uid=1052(ty13a) gid=1037(ty3a) groups=1037(ty3a),1027(tycsg8)**

### Delete all above created user along with its related files and group accounts. STEPS:

* 1. Delete all users created. Check /etc/passwd to verify if all users are

deleted.

* 1. Delete all groups created. Check /etc/group to verify if all groups are deleted.
  2. Delete the directories created in /home.

**Command for deleting users:**

**Ary9n@ary9n-virtual-machine:~$ sudouserdel -r tycs2020**

### Command for deleting groups:

**Ary9n@ary9n-virtual-machine:~$ sudogroupdel tycsg3**

### Command for deleting directories in /home:

**Ary9n@ary9n-virtual-machine:~$ sudo rm -r /home/ty11aa**

**Name – Aryan Penikal Roll No: 50**

**PRACTICAL – 10** - **Install Apache Web server**

**THEORY:**

Apache is a widely popular and open source Hypertext Transfer Protocol (HTTP) server software. It is released under the Apache license. Apache offers some of the following benefits and advantages:

* It is stable.
* It is used, backed, and supported by several major sites and organizations.
* The entire program and related components are open source.
* It works on a large number of platforms (all popular variants of Linux/UNIX, some of the not-sopopular variants of UNIX, and even Microsoft Windows).
* It is extremely flexible.
* It has proved to be secure.

HTTP traffic makes up a significant portion of the world’s Internet traffic, and Apache is a server implementation of HTTP. Browsers such as Firefox, Chrome, Opera, Curl, wget, Edge, Safari, and Internet Explorer are client implementations of HTTP.

The default port for HTTP requests is port 80, but you can also configure a web server to use a different (arbitrarily chosen) port that is not in use by another service. This is one of the mechanisms for running multiple web servers or sites on the same host, with each server or site on a different port. Some sites also use this arrangement for multiple configurations of their web servers to support various types of client requests. When a site runs a web server on a nonstandard port, you can see that port number in the site’s URL. For example, the web address www.example.com with the default port number (80) implicitly and explicitly displayed would read http://www.example.com and http://www.example.com:80, respectively. But serving the same site on a nonstandard port such as port 8080 will require the port number to be explicitly stated, like <http://www.example.com:8080>.

Running a web server on a Linux/UNIX platform follows the traditional Linux/UNIX permissions and ownership model. In terms of permissions, that means each process has an owner and that owner has limited rights on the system. Whenever a program (process) is started, it inherits the permissions of its parent process. For example, if you’re logged in as root, the shell in which you’re doing all your work has all the same rights as the root user. In addition, any process you start from this shell will inherit all the permissions of that root. Processes may give up rights, but they cannot gain rights.

To carry out initial network-related functions, the Apache HTTP server must start with root permissions. Specifically, it needs to bind itself to port 80 so that it can listen for requests and accept connections. Once it does this, Apache can give up its rights and run as a non-root user (unprivileged user), as specified in its configuration files. Different Linux distributions may have varying defaults for this user, but it is usually one of the following: nobody, www, apache, wwwrun, www-data, or daemon. Remember that when running as an unprivileged user, Apache can read only the files that the user has permissions to read. Security is especially important for sites that use executable scripts such as Common Gateway Interface (CGI) or PHP scripts. By limiting the permissions of the web server, you decrease the likelihood that someone can send a malicious and executable request to the server.

**STEPS**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo apt update**

[sudo] password for ary9n:

Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]

Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease

Get:3 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [135 kB]

Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]

Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [348 kB]

Get:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]

Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [325 kB]

Get:8 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [80.0 kB]

Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [13.0 kB]

Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [304 kB]

Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [607 kB]

Get:12 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [47.2 kB]

Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [209 kB]

Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [284 kB]

Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [12.1 kB]

Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [142 kB]

Get:17 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [93.2 kB]

Get:18 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [8,760 B]

Get:19 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [340 kB]

Get:20 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [52.9 kB]

Get:21 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [278 kB]

Get:22 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [422 kB]

Get:23 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [247 kB]

Get:24 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]

Get:25 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [6,756 B]

Get:26 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [5,204 B]

Get:27 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12.5 kB]

Fetched 4,298 kB in 11s (407 kB/s)

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

187 packages can be upgraded. Run 'apt list --upgradable' to see them.

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo apt install apache2**

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following packages were automatically installed and are no longer required:

libgssrpc4 libkadm5clnt-mit12 libkadm5srv-mit12 libkdb5-10

Use 'sudo apt autoremove' to remove them.

The following additional packages will be installed:

apache2-bin apache2-data apache2-utils libapr1 libaprutil1

libaprutil1-dbd-sqlite3 libaprutil1-ldap

Suggested packages:

apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser

The following NEW packages will be installed:

apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1

libaprutil1-dbd-sqlite3 libaprutil1-ldap

0 upgraded, 8 newly installed, 0 to remove and 187 not upgraded.

Need to get 1,918 kB of archives.

After this operation, 7,702 kB of additional disk space will be used.

Do you want to continue? [Y/n] y

Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libapr1 amd64 1.7.0-8build1 [107 kB]

Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4 [92.4 kB]

Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4 [11.3 kB]

Get:4 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4 [9,162 B]

Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.1 [1,347 kB]

Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.1 [165 kB]

Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.1 [89.1 kB]

Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.1 [97.8 kB]

Fetched 1,918 kB in 5s (402 kB/s)

Selecting previously unselected package libapr1:amd64.

(Reading database ... 203178 files and directories currently installed.)

Preparing to unpack .../0-libapr1\_1.7.0-8build1\_amd64.deb ...

Unpacking libapr1:amd64 (1.7.0-8build1) ...

Selecting previously unselected package libaprutil1:amd64.

Preparing to unpack .../1-libaprutil1\_1.6.1-5ubuntu4\_amd64.deb ...

Unpacking libaprutil1:amd64 (1.6.1-5ubuntu4) ...

Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.

Preparing to unpack .../2-libaprutil1-dbd-sqlite3\_1.6.1-5ubuntu4\_amd64.deb ...

Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.1-5ubuntu4) ...

Selecting previously unselected package libaprutil1-ldap:amd64.

Preparing to unpack .../3-libaprutil1-ldap\_1.6.1-5ubuntu4\_amd64.deb ...

Unpacking libaprutil1-ldap:amd64 (1.6.1-5ubuntu4) ...

Selecting previously unselected package apache2-bin.

Preparing to unpack .../4-apache2-bin\_2.4.52-1ubuntu4.1\_amd64.deb ...

Unpacking apache2-bin (2.4.52-1ubuntu4.1) ...

Selecting previously unselected package apache2-data.

Preparing to unpack .../5-apache2-data\_2.4.52-1ubuntu4.1\_all.deb ...

Unpacking apache2-data (2.4.52-1ubuntu4.1) ...

Selecting previously unselected package apache2-utils.

Preparing to unpack .../6-apache2-utils\_2.4.52-1ubuntu4.1\_amd64.deb ...

Unpacking apache2-utils (2.4.52-1ubuntu4.1) ...

Selecting previously unselected package apache2.

Preparing to unpack .../7-apache2\_2.4.52-1ubuntu4.1\_amd64.deb ...

Unpacking apache2 (2.4.52-1ubuntu4.1) ...

Setting up libapr1:amd64 (1.7.0-8build1) ...

Setting up apache2-data (2.4.52-1ubuntu4.1) ...

Setting up libaprutil1:amd64 (1.6.1-5ubuntu4) ...

Setting up libaprutil1-ldap:amd64 (1.6.1-5ubuntu4) ...

Setting up libaprutil1-dbd-sqlite3:amd64 (1.6.1-5ubuntu4) ...

Setting up apache2-utils (2.4.52-1ubuntu4.1) ...

Setting up apache2-bin (2.4.52-1ubuntu4.1) ...

Setting up apache2 (2.4.52-1ubuntu4.1) ...

Enabling module mpm\_event.

Enabling module authz\_core.

Enabling module authz\_host.

Enabling module authn\_core.

Enabling module auth\_basic.

Enabling module access\_compat.

Enabling module authn\_file.

Enabling module authz\_user.

Enabling module alias.

Enabling module dir.

Enabling module autoindex.

Enabling module env.

Enabling module mime.

Enabling module negotiation.

Enabling module setenvif.

Enabling module filter.

Enabling module deflate.

Enabling module status.

Enabling module reqtimeout.

Enabling conf charset.

Enabling conf localized-error-pages.

Enabling conf other-vhosts-access-log.

Enabling conf security.

Enabling conf serve-cgi-bin.

Enabling site 000-default.

Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /l

ib/systemd/system/apache2.service.

Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.

service → /lib/systemd/system/apache-htcacheclean.service.

Processing triggers for ufw (0.36.1-4build1) ...

Rules updated for profile 'OpenSSH'

Firewall reloaded

Processing triggers for man-db (2.10.2-1) ...

Processing triggers for libc-bin (2.35-0ubuntu3) ...

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw app list**

Available applications:

Apache

Apache Full

Apache Secure

CUPS

OpenSSH

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw allow 'Apache Full'**

Rule added

Rule added (v6)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo systemctl status apache2**

● apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>

Active: active (running) since Wed 2022-09-28 19:45:15 IST; 2min 19s ago

Docs: https://httpd.apache.org/docs/2.4/

Main PID: 4057 (apache2)

Tasks: 55 (limit: 2247)

Memory: 5.0M

CPU: 54ms

CGroup: /system.slice/apache2.service

├─4057 /usr/sbin/apache2 -k start

├─4058 /usr/sbin/apache2 -k start

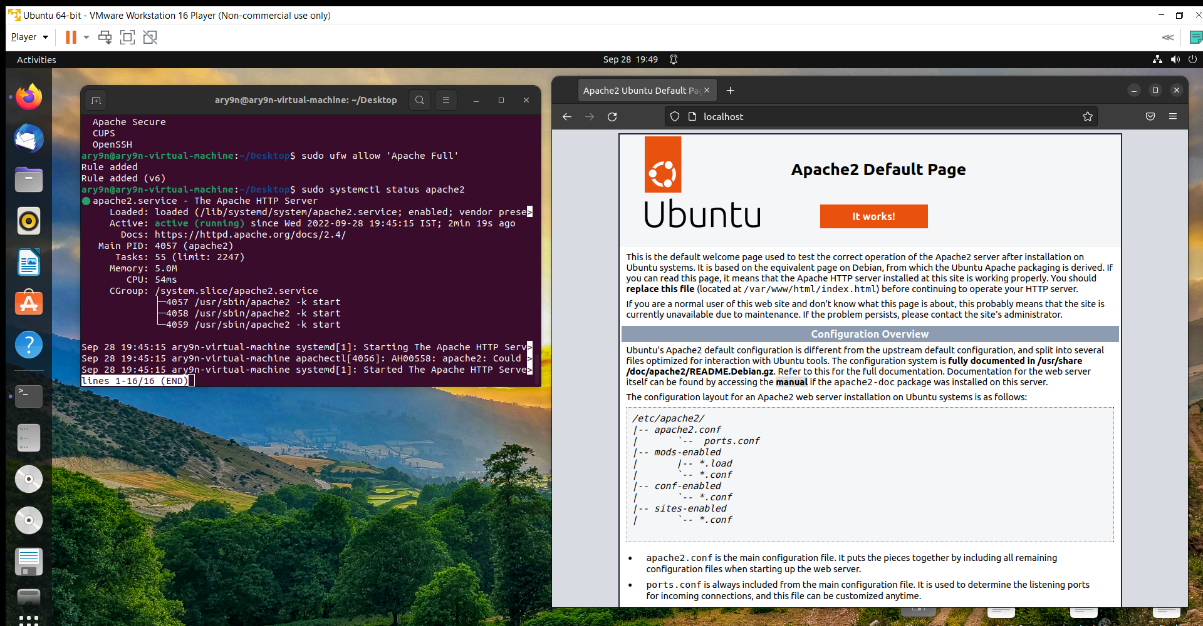
└─4059 /usr/sbin/apache2 -k start

Sep 28 19:45:15 ary9n-virtual-machine systemd[1]: Starting The Apache HTTP Serv>

Sep 28 19:45:15 ary9n-virtual-machine apachectl[4056]: AH00558: apache2: Could >

Sep 28 19:45:15 ary9n-virtual-machine systemd[1]: Started The Apache HTTP Serve>

lines 1-16/16 (END)



**ary9n@ary9n-virtual-machine:/var/www/html$ sudo gedit demo.html**

<!DOCTYPE HTML>

<html>

<head>

<title>Demo File</title>

<head>

<body>

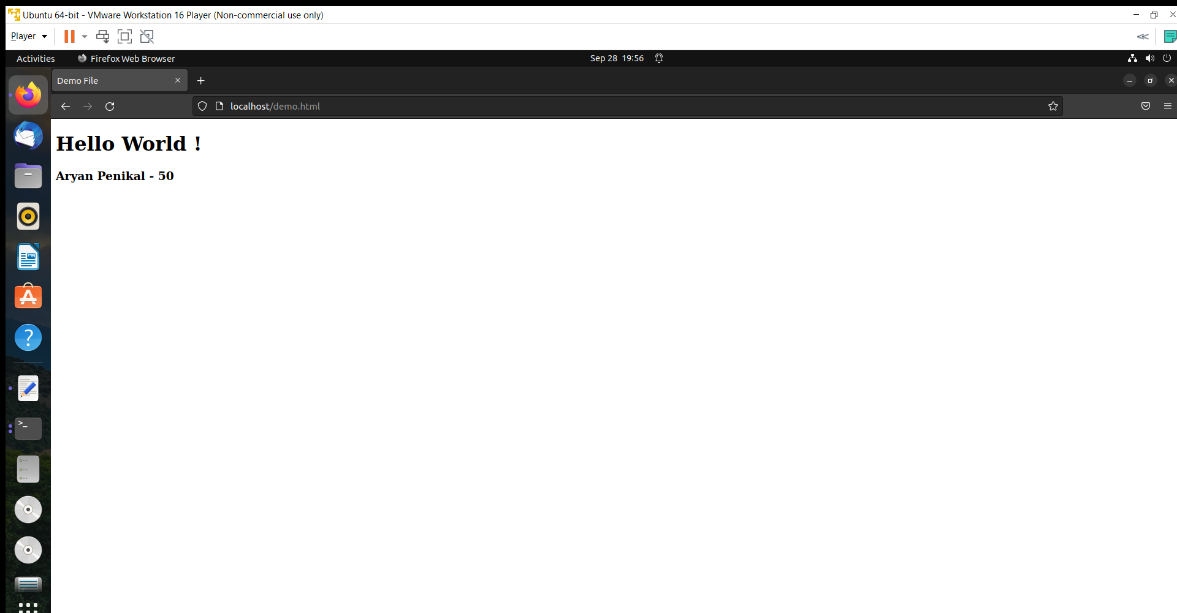
<h1>Hello World !</h1>

<h3>Aryan Penikal - 50 </h3>

</body>

</html>

**Output:**



**Name: Aryan Penikal Roll No: 50**

**Practical 11 – Install MySQL Server**

**Theory:**

MySQL is an open-source relational database management system (RDBMS). A relational

database organizes data into one or more data tables in which data types may be related to

each other; these relations help structure the data. SQL is a language programmers use to

create, modify and extract data from the relational database, as well as control user access to

the database. In addition to relational databases and SQL, an RDBMS like MySQL works

with an operating system to implement a relational database in a computer’s storage system,

manages users, allows for network access and facilitates testing database integrity and

creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public

License, and is also available under a variety of proprietary licenses. MySQL was owned and

sponsored by the Swedish company MySQL AB, which was bought by Sun

Microsystems (now Oracle Corporation).

**Steps:**

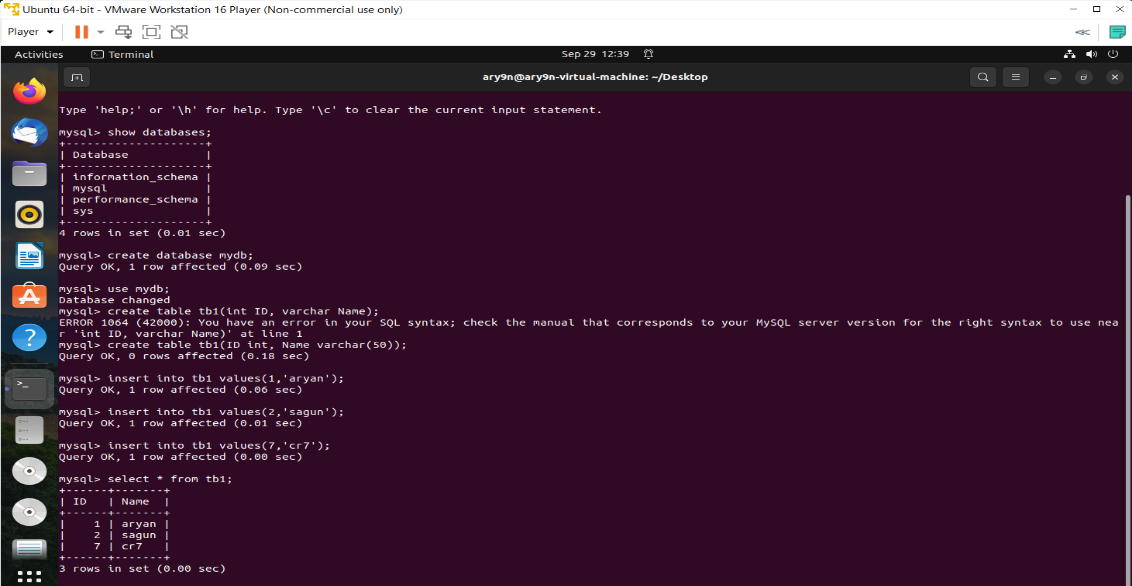
ary9n@ary9n-virtual-machine:~/Desktop$ sudo apt install mysql-server

ary9n@ary9n-virtual-machine:~/Desktop$ sudo mysql\_secure\_installation

ary9n@ary9n-virtual-machine:~/Desktop$ mysqladmin -u root -p version

ary9n@ary9n-virtual-machine:~/Desktop$ mysql -u root –p

**Output:**

****

**Name: Aryan Penikal Roll No: 52**

**Practical 12 – Install Python**

**Theory:**

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* web development (server-side),
* software development,
* mathematics,
* system scripting.

What can Python do?

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-oriented way or a functional way.

Good to know

* The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
* In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

* Python was designed for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**Steps:**

1. ary9n@ary9n-virtual-machine:~$ sudo apt update
2. ary9n@ary9n-virtual-machine:~$ sudo apt install software-properties-common
3. ary9n@ary9n-virtual-machine:~$ sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev wget
4. ary9n@ary9n-virtual-machine:~$ cd /usr/src
5. ary9n@ary9n-virtual-machine:/usr/src$ sudo wget https://www.python.org/ftp/python/3.8.5/Python-3.8.5.tgz
6. ary9n@ary9n-virtual-machine:/usr/src$ sudo tar -xzf Python-3.8.5.tgz
7. ary9n@ary9n-virtual-machine:/usr/src$ ls
8. ary9n@ary9n-virtual-machine:/usr/src$ cd Python-3.8.5
9. ary9n@ary9n-virtual-machine:/usr/src/Python-3.8.5$ sudo ./configure --enable-optimizations
10. ary9n@ary9n-virtual-machine:/usr/src/Python-3.8.5$ sudo make altinstall
11. ary9n@ary9n-virtual-machine:/usr/src/Python-3.8.5$ python3.8 –version
12. ary9n@ary9n-virtual-machine:/usr/src/Python-3.8.5$ cd ~
13. ary9n@ary9n-virtual-machine:/usr/src/Python-3.8.5$ gedit hello.py
14. ary9n@ary9n-virtual-machine:/usr/src/Python-3.8.5$ python3.8 hello.py
15. Hello world

**Name: Aryan Penikal Roll No: 50**

**Practical 13 – Install PhpMyAdmin**

**Theory:**

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

**Features**

* Intuitive web interface
* Support for most MySQL features:
  + browse and drop databases, tables, views, fields and indexes
  + create, copy, drop, rename and alter databases, tables, fields and indexes
  + maintenance server, databases and tables, with proposals on server configuration
  + execute, edit and bookmark any SQL-statement, even batch-queries
  + manage MySQL user accounts and privileges
  + manage stored procedures and triggers
* Import data from CSV and SQL
* Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LATEX and others
* Administering multiple servers
* Creating graphics of your database layout in various formats
* Creating complex queries using Query-by-example (QBE)
* Searching globally in a database or a subset of it
* Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
* And much more...

**Steps:**

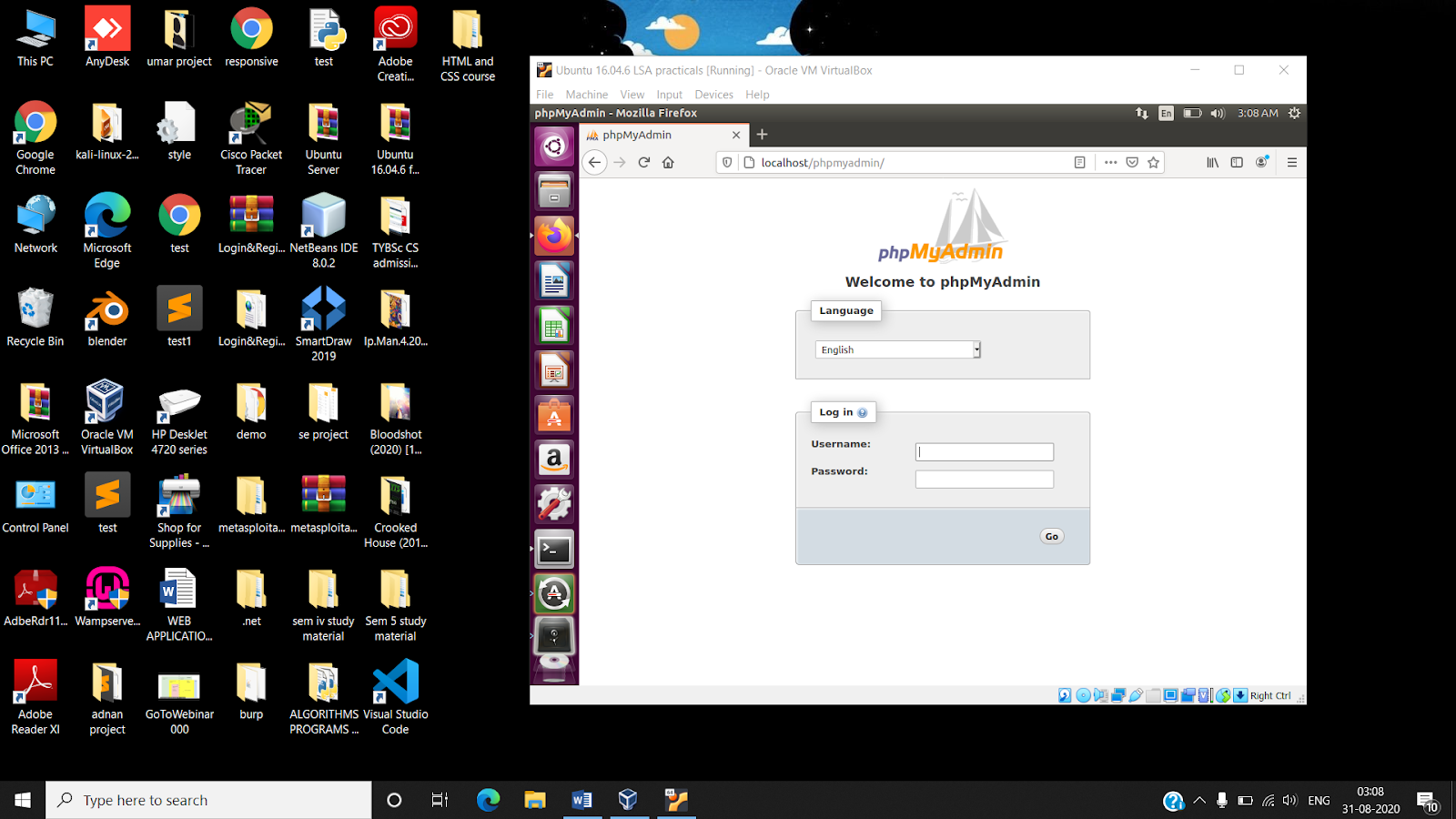
1) ary9n@ary9n-virtual-machine~$ sudo apt update

2) ary9n@ary9n-virtual-machine~$ sudo apt install phpmyadmin

3) ary9n@ary9n-virtual-machine~$ sudo systemctl restart apache2

4) ary9n@ary9n-virtual-machine~$ sudo systemctl restart mysql.service

5) ary9n@ary9n-virtual-machine~$ sudo a2enconf phpMyAdmin





**Name: Aryan Penikal Roll No: 50**

**Practical 14 – Installing SSH**

**Theory:**

SSH is a suite of network communication tools that are collectively based on an open

protocol/standard that is guided by the Internet Engineering Task Force (IETF). It allows

users to connect to a remote server just as they would using Telnet, rlogin, FTP, and so on,

except that the session is 100- percent encrypted. Someone using a packet sniffer merely sees

encrypted traffic going by. Should they capture the encrypted traffic, decrypting it could take

a long time.

Secure Shell relies on a technology called public-key cryptography. It works similarly to a

safe deposit box at the bank: You need two keys to open the box or at least multiple layers of

security/checks have to be crossed. In the case of public-key cryptography, you need two

mathematical keys: a public one and a private one. Your public key can be published on a

public web page, printed on a T-shirt, or posted on a billboard in the busiest part of town.

Anyone who asks for it can have a copy. Any data encrypted with the public key can be

decrypted with the private key. On the other hand, your private key must be protected to the

best of your ability. It is this piece of information that makes the data you want to encrypt

truly secure. Any data encrypted with the private key can be decrypted with the public key.

Every public key/private key combination is unique.

To make things even more interesting, SSH regularly changes its session key. (This is a

randomly generated, symmetric key for encrypting the communication between the SSH

client and server. It is shared by the two parties in a secure manner during SSH connection

setup.) In this way, the data stream gets encrypted differently every few minutes. Thus, even

if someone happened to figure out the key for a transmission, that miracle would be valid for

only a few minutes until the keys changed again.

The OpenSSH project was spearheaded by the OpenBSD project. OpenBSD is a version of

the Berkeley Software Distribution (BSD) operating system (another UNIX variant) that

strives for the best security of any operating system available. The core of the OpenSSH

package is considered part of the OpenBSD project and is thus simple and specific to the

OpenBSD operating system.

Here is a quick rundown of some SSH clients and other useful SSH resources:

**PuTTY:** This is probably one of the oldest and most popular SSH implementations for the

Win32 (Microsoft Windows) platforms. It is extremely lightweight and can either be used as

a stand-alone, self-contained executable or be installed like other Windows programs. The

website also hosts other tools such as pscp, which is a Windows command-line version of

Secure Copy (SCP).

OpenSSH for Apple OS X: OS X is actually a UNIX-based and UNIX-compliant operating

system. One of its main core components—the kernel—is based on the BSD kernel. So you

shouldn’t be too surprised that OpenSSH is part of the OS X system. When you open the

terminal application, you can simply issue the ssh command. OS X systems also ship with an

OpenSSH SSH server.

**OpenSSH**: is only as secure as the weakest connection between the user and the server. This

means that, for example, if a user uses Telnet to connect from host A to host B and then uses

ssh to connect to host C, the entire connection can be monitored from the link between host A

and host B. The fact that the link between host B and host C is encrypted becomes irrelevant.

**Steps:**

1) ary9n@ary9n-virtual-machine~$ sudo apt update

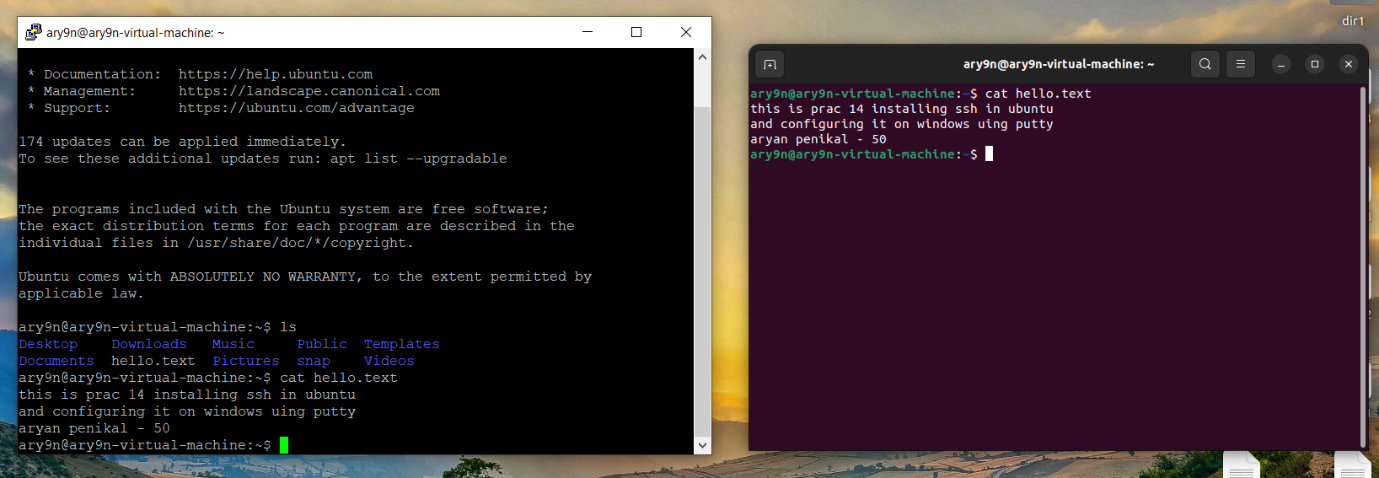
2) ary9n@ary9n-virtual-machine~$ sudo apt install openssh-server

3) ary9n@ary9n-virtual-machine~$ sudo systemctl status ssh

4) ary9n@ary9n-virtual-machine~$ ifconfig **or** ary9n@ary9n-virtual-machine~$ hostname –I

5) Go to Windows install and open PuTTy

**Output:**



**Name: Aryan Penikal Roll No: 50**

**Practical 15 – Install FTP**

**Theory:**

The File Transfer Protocol is a standard network protocol used for the transfer of computer

files between a client and server on a computer network. FTP is built on a client-server model

architecture using separate control and data connections between the client and the server.

The original design of FTP does not defend itself very well in the hostile Internet

environment that we have today, which necessitates the use of firewalls. Inasmuch as FTP

facilitates the exchange of files between an FTP client and an FTP server, its design has some

built-in nuances that are worthy of further mention. One of its nuances stems from the fact

that it uses two ports: a control port (port 21) and a data port (port 20). The control port

serves as a communication channel between the client and the server for the exchange of

commands and replies, and the data port is used purely for the exchange of data, which can

be a file, part of a file, or a directory listing.

FTP can operate in two modes: active FTP mode and passive FTP mode.

**Active FTP**

Active FTP was traditionally used in the original FTP specifications. In this mode, the client

connects from an ephemeral port (number greater than 1024) to the FTP server’s command

port (port 21). When the client is ready to transfer data, the server opens a connection from its

data port (port 20) to the IP address and ephemeral port combination provided by the client.

The key here is that the client does not make the actual data connection to the server but

instead informs the server of its own port by issuing the PORT command; the server then

connects back to the specified port. The server can be regarded as the active party (or the

agitator) in this FTP mode.

From the perspective of an FTP client that is behind a firewall, the active FTP mode poses a

slight problem: the firewall on the client side might frown upon (or disallow) connections

originating or initiated from the Internet from a privileged service port (such as data port 20)

to non-privileged service ports on the clients it is supposed to protect.

**Passive FTP**

The FTP client issues the PASV command to indicate that it wants to access data in the

passive mode, and the server then responds with an IP address and an ephemeral port number

on itself to which the client can connect to transfer the data. The PASV command issued by

the client tells the server to “listen” on a data port that is not its normal data port (that is, port

20) and to wait for a connection rather than initiate one. The key difference in comparison

with active FTP mode is that in passive FTP mode it is the client that initiates the connection

to the port and IP address provided by the server. And in this regard, the server can be

considered the passive party in the data communication.

From the perspective of an FTP server that is behind a firewall, passive FTP mode is a little

problematic, because a firewall’s natural instinct would be to disallow connections that

originate from the Internet that are destined for ephemeral ports of the systems that it is

supposed to protect. A typical symptom of this behavior occurs when a client appears to be

able to connect to the server without a problem, but the connection seems to hang whenever

an attempt to transfer data occurs.

**Steps:**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo apt update**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo apt install vsftpd**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo cp /etc/vsftpd.conf /etc/vsftpd1.conf**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw status**

Status: active

To Action From

-- ------ ----

21/tcp ALLOW Anywhere

20/tcp ALLOW Anywhere

40000:50000/tcp ALLOW Anywhere

OpenSSH ALLOW Anywhere

Apache Full ALLOW Anywhere

21/tcp (v6) ALLOW Anywhere (v6)

20/tcp (v6) ALLOW Anywhere (v6)

40000:50000/tcp (v6) ALLOW Anywhere (v6)

OpenSSH (v6) ALLOW Anywhere (v6)

Apache Full (v6) ALLOW Anywhere (v6)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw enable**

Firewall is active and enabled on system startup

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw allow 21/tcp**

Skipping adding existing rule

Skipping adding existing rule (v6)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw allow 20/tcp**

Skipping adding existing rule

Skipping adding existing rule (v6)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw allow 40000:50000/tcp**

Skipping adding existing rule

Skipping adding existing rule (v6)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw allow OpenSSH**

Skipping adding existing rule

Skipping adding existing rule (v6)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo systemctl status vsftpd**

● vsftpd.service - vsftpd FTP server

Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset>

Active: active (running) since Tue 2022-10-18 19:56:19 IST; 13min ago

Main PID: 1009 (vsftpd)

Tasks: 1 (limit: 2247)

Memory: 788.0K

CPU: 17ms

CGroup: /system.slice/vsftpd.service

└─1009 /usr/sbin/vsftpd /etc/vsftpd.conf

Oct 18 19:56:19 ary9n-virtual-machine systemd[1]: Starting vsftpd FTP server...

Oct 18 19:56:19 ary9n-virtual-machine systemd[1]: Started vsftpd FTP server.

...skipping...

● vsftpd.service - vsftpd FTP server

Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset>

Active: active (running) since Tue 2022-10-18 19:56:19 IST; 13min ago

Main PID: 1009 (vsftpd)

Tasks: 1 (limit: 2247)

Memory: 788.0K

CPU: 17ms

CGroup: /system.slice/vsftpd.service

└─1009 /usr/sbin/vsftpd /etc/vsftpd.conf

Oct 18 19:56:19 ary9n-virtual-machine systemd[1]: Starting vsftpd FTP server...

Oct 18 19:56:19 ary9n-virtual-machine systemd[1]: Started vsftpd FTP server.

~

~

~

~

~

~

~

~

~

~

~

lines 1-12/12 (END)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo systemctl restart vsftpd**

**ary9n@ary9n-virtual-machine:~/Desktop$ ftp 192.168.13.128**

Connected to 192.168.13.128.

220 (vsFTPd 3.0.5)

Name (192.168.13.128:ary9n): ary9n

331 Please specify the password.

Password:

230 Login successful.

Remote system type is UNIX.

**Name: Aryan Penikal Roll No: KCTBCS050**

**Practical 16 – Samba Installation**

**Theory:**

Samba is a powerful suite of applications that helps UNIX-based systems (such as Linux) interoperate with Windows-based and other operating systems. It is an open source implementation of the Server Message Block/Common Internet File System (SMB/CIFS) protocol suite.

Samba transparently provides file and print sharing services to Windows clients as well as other networked clients running other operating systems. It does this through the use of the native Microsoft networking protocols SMB/CIFS. From a system administrator’s point of view, this means you can deploy a Linux/UNIX-based server and use it to provide file sharing, authentication, print, and other services to other non-native Linux clients such as Microsoft Windows systems. Using Samba means that Windows systems can use their native tongue to talk to the Linux server—which means fewer hassles for you and seamless integration for your users.

The Linux/UNIX login/password mechanism is radically different from the Windows Active Directory model, which uses domain controllers (DCs). Thus, it’s important for the system administrator to maintain consistency in the logins and passwords across both platforms. Users may need to work in heterogeneous environments and may need access to the different platforms for various reasons. It is thus useful to make working in such environments as seamless as possible without having to worry about users’ needing to reauthenticate separately on the different platforms, worry about cached passwords that don’t match between servers, and other issues. Relative to Samba, several options are available for handling username and password issues in heterogeneous environments, including the following:

* **Linux pluggable authentication modules (PAMs):** Allow you to authenticate users against a DC. This means you still have two user lists—one local and one on the DC—but your users need to keep track of their passwords only on the Windows system.
* **Samba as a DC:** Allows you to keep all your logins and passwords on the Linux system, while all your Windows boxes authenticate with Samba. When Samba is used with a Lightweight Directory Access Protocol (LDAP) back-end for this, you will have a scalable and extensible solution.
* **Custom script:** Allows you to use your own custom script. For sites with a well-established system for maintaining logins and passwords, it isn’t unreasonable to come up with a custom script. This can be done using a scripting language with good cross-platform support. Such scripts can be coaxed to allow changes to the Security Access Manager (SAM) to update the DC’s password list.

Windows-based systems use encrypted passwords when communicating with the DC and any server requiring authentication (including Linux and Samba). The encryption algorithm used by Windows is different from UNIX’s, however, and, therefore, is not compatible. Here are your choices for handling this conflict:

* Edit the Registry on Windows clients to disable the use of encrypted passwords. The Registry entries that need to be changed are listed in the docs directory in the Samba package. As of version 3 of Samba, this option is no longer necessary.
* Configure Samba to use Windows-style encrypted passwords.

The first solution has the benefit of not pushing you to a more complex password scheme. On the other hand, you may have to apply the Registry fix on all your clients. The second option, of course, has the opposite effect: for a little more complexity on the server side, you don’t have to modify any of your clients.

The smbd daemon handles the actual sharing of file systems and printer services for clients. It is also responsible for user authentication and resource-locking issues. It starts by binding to port 139 or 445 and then listens for requests. Every time a client authenticates itself, smbd makes a copy of itself; the original goes back to listening to its primary port for new requests, and the copy handles the connection for the client. This new copy also changes its effective user ID from root to the authenticated user.

The nmbd daemon is responsible for handling NetBIOS name service requests. nmbd can also be used as a drop-in replacement for a Windows Internet Name Server (WINS). It begins by binding itself to port 137; unlike smbd, however, nmbd does not create a new instance of itself to handle every query. In addition to name service requests, nmbd handles requests from master browsers, domain browsers, and WINS servers—and as such, it participates in the browsing protocols that make up the popular Windows My Network Places of systems. The services provided by the smbd and nmbd daemons complement each other.

Finally, the service provided by winbindd can be used to query native Windows servers for user and group information, which can then be used on purely Linux/UNIX platforms. It does this by using Microsoft Remote Procedure Call (RPC) calls, PAM, and the name service switch (NSS) capabilities found in modern C libraries. Its use can be extended through the use of a PAM module (pam\_winbind) to provide authentication services. This service is controlled separately from the main smb service and can run independently.

**Steps:**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo apt install samba**

[sudo] password for ary9n:

**ary9n@ary9n-virtual-machine:~/Desktop$ ls /etc/samba**

gdbcommands smb.conf tls

**ary9n@ary9n-virtual-machine:~/Desktop$ mkdir smbdir1**

**ary9n@ary9n-virtual-machine:~/Desktop$ cd smbdir1/**

**ary9n@ary9n-virtual-machine:~/Desktop/smbdir1$ cat > myfile1.txt**

hello

**ary9n@ary9n-virtual-machine:~/Desktop/smbdir1$ cd ..**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo adduser sambauser1**

Adding user `sambauser1' ...

Adding new group `sambauser1' (1016) ...

Adding new user `sambauser1' (1011) with group `sambauser1' ...

Creating home directory `/home/sambauser1' ...

Copying files from `/etc/skel' ...

New password:

Retype new password:

passwd: password updated successfully

Changing the user information for sambauser1

Enter the new value, or press ENTER for the default

Full Name []: Aryan Penikal

Room Number []: 50

Work Phone []: 1234567890

Home Phone []: 1234567890

Other []: 0000000000

Is the information correct? [Y/n] y

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo systemctl status smbd**

● smbd.service - Samba SMB Daemon

Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enab>

Active: active (running) since Tue 2022-10-18 20:24:40 IST; 2min 31s ago

Docs: man:smbd(8)

man:samba(7)

man:smb.conf(5)

Process: 18463 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (cod>

Main PID: 18492 (smbd)

Status: "smbd: ready to serve connections..."

Tasks: 4 (limit: 2247)

Memory: 15.1M

CPU: 133ms

CGroup: /system.slice/smbd.service

├─18492 /usr/sbin/smbd --foreground --no-process-group

├─18499 /usr/sbin/smbd --foreground --no-process-group

├─18500 /usr/sbin/smbd --foreground --no-process-group

└─18501 /usr/lib/x86\_64-linux-gnu/samba/samba-bgqd --ready-signal-fd=4>

Oct 18 20:24:40 ary9n-virtual-machine systemd[1]: Starting Samba SMB Daemon...

Oct 18 20:24:40 ary9n-virtual-machine update-apparmor-samba-profile[18477]: grep: />

Oct 18 20:24:40 ary9n-virtual-machine update-apparmor-samba-profile[18486]: diff: />

Oct 18 20:24:40 ary9n-virtual-machine systemd[1]: Started Samba SMB Daemon.

lines 1-22/22 (END)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo ufw allow samba**

Rule added

Rule added (v6)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo systemctl restart smbd**

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo systemctl status smbd**

● smbd.service - Samba SMB Daemon

Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enab>

Active: active (running) since Tue 2022-10-18 20:30:12 IST; 7s ago

Docs: man:smbd(8)

man:samba(7)

man:smb.conf(5)

Process: 19216 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (cod>

Main PID: 19226 (smbd)

Status: "smbd: ready to serve connections..."

Tasks: 4 (limit: 2247)

Memory: 9.0M

CPU: 104ms

CGroup: /system.slice/smbd.service

├─19226 /usr/sbin/smbd --foreground --no-process-group

├─19228 /usr/sbin/smbd --foreground --no-process-group

├─19229 /usr/sbin/smbd --foreground --no-process-group

└─19230 /usr/lib/x86\_64-linux-gnu/samba/samba-bgqd --ready-signal-fd=4>

Oct 18 20:30:12 ary9n-virtual-machine systemd[1]: Starting Samba SMB Daemon...

Oct 18 20:30:12 ary9n-virtual-machine systemd[1]: Started Samba SMB Daemon.

lines 1-20/20 (END)

**ary9n@ary9n-virtual-machine:~/Desktop$ sudo smbpasswd -a sambauser1**

New SMB password:

Retype new SMB password:

Added user sambauser1.

**Follow these steps:**

1. In Windows OS, click on Network icon on desktop

2. You will get ARY9N icon in window, click on that, it will display smbdir1 folder

3. Click on the folder you created

4. Enter the new username and password, which you created in Ubuntu for samba server. (i.e sambauser1 and its password)

5. Now you can share folder, files from Windows to Linux and the other way around

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A picture containing text, monitor, computer, electronics

Description automatically generated

A screenshot of a computer screen

Description automatically generated