

**Faculty of** **Technology and Engineering**

# U & P U. Patel Department of Computer Engineering

Date: 14 / 08 / 2022

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Academic Year | : | 2022-23 | Semester | : | 5 |
| Course code | : | CE354 | Course name | : | Operating System |

PRACTICAL – 2

**PART A**

**AIM: Manage local users, groups and creation of multiple users from excel sheet**

|  |  |
| --- | --- |
| 1. | Run id command to view the current user and group information. |
| **Command** | **Id** |
| obs | This Command helps to print -id user-id and other group-id. |
| 2. | Display the current working directory |
| **command** | **pwd** |
| obs | It will print the current path directory |
| 3. | Print the value of HOME and PATH variable to determine the home directory and user’s executable path respectively. |
| **command** | **echo $HOME**  **echo $path** |
| obs | Echo $HOME will print the path of HOME variable  Echo $PATH will print the path of all routes variable |
| 4. | Run su and su – command .Observe the output for the same. What is the main difference between then?(su: switching user). |
| **command** | **su**  **su -** |
| obs | Su command without arguments keeps almost all environmenr variables belonging to the original user. |
| 5. | Run sudo su at the shell prompt to become the root user. |
| **command** | **sudo su** |
| obs | The command starats a login shell and you will find yourself in root’s home directory with root’s environment. |
| 6. | Run id command to view the current user and group information. |
| **command** | **id** |
| obs | The output shows the ID of current user UID and GID. |
| 7. | Dislay the current working directory |
| **command** | **pwd** |
| obs | It display the current working directory. |
| 8. | Print the value of HOME and PATH variable to determine the home directory and user’s executable’s path respectively. |
| **command** | **echo $HOME**  **echo $PATH** |
| obs | It will print the value of HOME and PATH variable to determine the home directory and user’s executable’s path. |
| 9. | Exit the current user’s shell to return to the student user’s shell. |
| **command** | **exit** |
| obs | It exit the current user’s shell to return to the student user’s shell. |
| 10. | Attempt to view the last five lines of /var/log/auth.log without using sudo |
| **command** | **tail -5/var/log/auth.log** |
| obs | Logs not viewed |
| 11. | Attempt to view the last five liens of /var/log/auth.log using sudo |
| **command** | **sudo tail -5/var/log/auth.log** |
| obs | It display Last five lines viewed. |
| 12. | Attempt to make a copy of /etc/rpc as /etc/rpcOLD without using sudo |
| **command** | **Cp/etc/rpc/etc/rpcOLD** |
| obs | Copy of the file not made |
| 13. | Attempt to make a copy of /etc/rpc as /etc/rpcOLD with sudo. |
| **command** | **sudo cp/etc/rpc/etc/rpcOLD** |
| obs | File with the above name created |
| 14. | Attempt to delete/etc/rpcOLD without using sudo |
| **command** | **rm /etc/rpcOLD** |
| obs | File not detected |
| 15. | Attempt to delete/etc/rpsc OLD with sudo |
| **command** | **Sudo rm/etc/rpcOLD** |
| obs | File detected when sudo command is used. |
| 16. | Check the UID for root user,administrator and local users. |
| **command** | **Id –u 20CE020**  **Id –u administrator**  **Id –u root** |
| obs | Id for local user is 1002 ,administrator is 1000 and root is 0 |
| 17. | Adduser user01. |
| **command** | **adduser user01** |
| obs | It add user named user01. |
| 18. | Create the group group01 with the GID of 10000. |
| **command** | **groupadd –g 10000 group01.** |
| obs | Group will be add of group id 10000. |
| 19. | Create the group group02. |
| **command** | **groupadd –g 10001 group02** |
| obs | Group will be add of group id 10001. |
| 20. | Examine /etc/group to verify the supplemental group memberships. |
| **command** | **Cat /etc/group** |
| 21. | Use the usermod –aG command to add a user to a supplementary group. Add user01 to the group created. |
| **command** | **Sudo usermod –aG sudo 20CE020** |
| 22. | Observe /etc/group and etc/passwd |
| **command** | **cat/etc/group**  **cat/etc/passwd** |

**PART-B  
Aim: Control access to files**

|  |  |
| --- | --- |
| 1. | Check the permission of files created. |
| **Command** | **ls –l file1.txt** |
| **obs** | **This command helps to find permission of files** |
| 2. | Check the permission of directory created. |
| **Command** | **ls – ld** |
| **obs** | **This command helps to find permission of folders** |
| 3. | Set read and write permissions for others with numeric mode to file1.txt |
| **command** | **chmod 006 file1.txt**  **ls – l file1.txt** |
| 4. | Remove write permission for user ,group and others to folder CE. |
| **command** | **chmod a-w ce**  **ls –ld ce** |
| 5. | Create a directory 5CE under CE.Observe the response. |
| **command** | **cd ce**  **mkdir 5ce**  **We are not allowed to create a directory as we have removed the write permission previously.** |
| 6. | Set read, write and execute permissions for user, group and others to 5CE.  //For granting the permission of write. |
| **command** | **chmod a+w ce**  **mkdir 5ce**  **chmod 777 5ce** |
| 7. | Set read and execute permission for group and no permission for other to file2.txt. |
| **command** | **vi file2.txt**  **chmod 050 file2.txt**  **ls –l file2.txt** |
| 8. | Change the ownership of file to user01. |
| **command** | **Sudo chown user01 file1.txt** |
| 9. | Change the group ownership of file to group01 |
| **command** | **Sudo chown :group01 file1.txt** |
| 10. | Change the ownership of both group and user at the same time. |
| **command** | **Vi file2.txt**  **Sudo chown user01:group01 file2.txt** |
| 11. | Set the special permission on directory.  a.The setid permission on an executable file means that commands run as the suer owning the file, not as the user that ran the command. One example is the passwd command:run ls –l .usr/bin/passwd  b. The special permission setgid on a directory means that files created in the directory inherit their group ownership from the directory, rather than inheriting it from the creating user. Run ls-ld/run/log/journal  c. The sticky bit for a direcyory sets a special restriction on deletion of files. Only the owner of the file (and root) can delete files within the directory. Run ls –ld/tm0p |
| **command** | **A). ls –l usr/bin/passwd**  **-rwse-xr-x 1 root 68208 May 28 2020/usr/bin/passwd**  **B). ls –ld /run/log/journal**  **Drwxr-sr-x+ 2 root system-journal 40**  **C). ls –ld /tmp**  **Drwxrwxrwx 21 root root 4096** |
| 12. | Set the setusid, setgid and sticky bot for different files and perform the operations accordingly. |
| Command | **setusid** |
| 13. | Display the current value of shell’s mask |
| **command** | **umask** |
| 14. | Check the permission of directories. |
| **command** | **mkdir mydict**  **ls –ld mydict 775** |
| 15. | Check the permission of files. |
| **command** | **Vi file3.txt**  **ls –l file3.txt 664** |
| 16. | Set the umask to 542. |
| **command** | **umask 542** |
| 17. | Check the permission of files and directories. |
| **command** | **Ls –l**  **Ls –ld** |
| **obs** | **Here for find permissions ls –l for files and ls –ld for directory.** |
| 18. | Try to open the file and directory created. |
| **command** | **Vi file2.txt** |
| 19. | Try to open the file as other user. |
|  | **File won’t be opened as we have change the umask to 542.** |
| 20. | Submit the exercise on MS Team. |