

**Computer Science 328 -01**

## **Lab Assignment 3**

# **Due Date: Wednesday, February 7, 2024 (before midnight)**

## **Objectives:**

## **Learning file types, file system, ownership and permission**

## **Learning local and network file access**

## **Task 1:** **Learning file types, file system, ownership and permission**

## Please refer to the sites for the information of Linux file system:

<https://opensource.com/life/16/10/introduction-linux-filesystems>

<https://www.tecmint.com/find-linux-filesystem-type/>

Start Ubuntu VM2 and open a terminal => use ***df*** command to find out all the file system types being used in the system => then use ***lsblk*** command to list all system file types => Take a screen shot of the terminal to show both results

|  |
| --- |
| Insert your screen shot here: |

Question: Explain the differences of the results from above.

|  |
| --- |
| Your answer:  DF display the information about the amount of free disk space on the file systems  lsblk displays the information about hard drives and its partitions, shows the storage devices and  a tree like relations. |

Describe what these four typical directories are used for in a Linux system.

|  |
| --- |
| 1. /home   Usually used to store the personal data for each user under /home/username, it also used as the default starting location   1. /etc   Under this directory, many apps’ configurations files are stored, along with many system settings   1. /sbin   This is used for storing system binaries such as boot utilities, administration tools, device management   1. /bin   It is used to store essential binary executable and command files, usually stores system commands and root filesystem, minimal environment and independence form /user. |

* Next try these steps below to experiment with file permission and owernship:
* Open a terminal
* Create a new group named **netgroup** for the system as follow:  
   > **sudo groupadd netgroup**
* Create a new user named **jlow** (“Jake Low”) with a home directory.
* Change password for the new user **jlow** with a password **letmein**.
* Change the primary group of **jlow** to **netgroup** as follow:  
   > **sudo usermod -g netgroup jlow**

Append **sudo** group to jlow as secondary group:

> **sudo usermod -aG sudo jlow**

* Type **reboot** and then login as **jlow** user (Jake Low) with the password **letmein**.
* Change directory to **jlow**’s home directory (cd ~).
* Use **cat** command to create***firstfile*** with “Hello” as file contents.
* Use **cat** command to create***secondfile*** with “Goodbye” as file contents.
* Change file permissions for both files as follow:  
   > **chmod 757 firstfile**  
   > **chmod 757 secondfile**
* List the files to see the owner and primary group of the two files, and write them down below:

|  |
| --- |
| Owner of **firstfile** : jlow Ownner of **secondfile**:jlow  Primary group of **firstfile**:netgroup Primary group of **secondfile**:netgroup |

* Now **jlow** user tries the following command :  
   > **sudo chown root:netgroup secondfile**

\* Then **jlow** user tries the following commands:

> **cat secondfile >> firstfile**  
 > **cat firstfile >> secondfile**

|  |
| --- |
| Describe what happens and explain why it happens.    The first operation is execute succesfully but the second returns an permission denied  this is because secondfile has a root ownership which can override jlow’s permission but this does not work the other way around. jlow can not override root’s access permission |

- Understanding the structure of the password files in our Linux systems is very important from an administrator’s point of view. For security reason, Linux is using two separate files for handling passwords. The first file is */etc/passwd* and it is accessible by different groups of users. Each line of the file has the following structure:

username : passwd : uid : gid : fullname : homedir : shell  
  
For example:

*$ cat /etc/passwd*  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin

ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
 :

jdoe:x:500:500:john doe:/home/jdoe:/bin/bash  
ppan:x:501:501:peter pan:/home/ppan:/bin/bash  
  
The second file is */etc/shadow* and it contains the encrypted passwords and other password information. This file is only accessible by root for security reason.

*$ cat /etc/shadow*

root:a3mUhgkEFmmBM:11577:0:99999:7:::

bin:\*:11577:0:99999:7:::

daemon:\*:11577:0:99999:7:::

:

jdoe:ngHjii397KKld/9b:11577:0:99999:7:::

ppan:lrdGEty109mE:11577:0:99999:7:::

**Now do a search and find out the shadow password format from the Internet. Describe the meaning of each field for a shadow password entry below.**

|  |
| --- |
| 1. Username: account name 2. Password for the user: encrypted password is in hash format 3. Last password change : The date of the last password change, expressed as the number of days since Jan 1, 1970 (Unix time) 4. minimum password age: The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change her password again. An empty field and value 0 mean that there are no minimum password age. 5. maximum password age: The maximum number of days the password is valid, after that user is forced to change her password again. 6. password warning period: The number of days before password is to expire that user is warned that his/her password must be changed 7. password inactivity period: The number of days after password expires that account is disabled. 8. account expiry date : The date of expiration of the account, expressed as the number of days since Jan 1, 1970. |

**Task 2:**

* Open and run Kali VM, which has already been installed with Apache2 server and PHP engine.

Now we can simply start the Apache2 server in a terminal like this:  
 *$ service apache2 start*

* Next try to find the path of the document root of Apache2 server using “find” command. Issue a ***find***command to search for a directory with a directory name called **html** with the search starting from the root directory (/) of the Kali file system, and immediately pipe the search results to the ***head*** command for displaying the first 10 lines of the search results. Take a screen shot of the terminal to show the first 10 lines of the search results and insert your screen shot below:

|  |
| --- |
|  |

* Now you can create and access local files and directories:  
  = > starting in user’s home directory (cd ~), use cat command to create a short file named **phpinfo.php** with the following three lines of text:

<?php

phpinfo();

?>

= > use mv command to move **phpinfo.php** file into Apache2’s document root (/var/www/html) directory => take a screen shot of the terminal to show moving the php file into Apache2’s document root directory successfully

|  |
| --- |
| Insert your screen shot here: |

= > open a web browser and enter <http://localhost/phpinfo.php> into the URL box => take a screen shot of the browser showing the PHP information currently used in the VM

|  |
| --- |
| Insert your screen shot here: |

* Next please refer to “wget command.pdf” and “curl command.pdf” to download file(s) from a web site = > create a directory named **mydir** inside the Apache2’s document root (/var/www/htm/) => create a couple text files (**hello.txt** and **newyear.txt**) inside **mydir**, each with a short message like “Hello everyone!” and “Happy New Year!” respectively => use cd command to go to Downloads foler => use ls –l to show the existing files => use wget command to download **hello.txt** => use ls –l to show the file has been download into Downloads folder => take a screen shot of the whole terminal to show the file listing before and after the download.

|  |
| --- |
| Insert your screen shot here: |

= > then use curl command to download **newyear.txt** and save it under a different name (**happy2024.txt**) inside the Downloads folder => use ls –l to show the file with the new name => take a screen shot of the whole terminal to show the download occurred and file listing.

|  |
| --- |
| Insert your screen shot here: |

= > Don’t poweroff your current VM, leave it as a server VM => start the other two VMs as well instead.   
  
- Next use SSH network connection for copying files from a remote machine => install SSH server on Kali VM, and both Ubuntu VM1 and VM2 the same way in a terminal:  
   
 $ sudo apt install openssh-server

$ sudo systemctl enable ssh

$ sudo systemctl start ssh

Please refer to scp commands from the following web site:  
 <https://phoenixnap.com/kb/linux-scp-command>

= > choose one of the other two VMs as your client VM => use scp command to secure copy the file “happy2024.txt” inside Downloads folder from the server VM into your current client VM => take a screen shot of the terminal to show the secure copy completed successfully.

|  |
| --- |
| Insert your screen shot here: |

= > use cd command to go to Documents folder of your current client VM => create a folder named **reports** inside Documents folder => use touch command to create two empty files (report1.txt and report2.txt) => use mv command to move the two files into **reports** folder => use ***scp*** command to copy the **reports** folder from your current client VM into the server VM’s Documents folder => take a screen shot of the terminal to show that the secure copy completed successfully.

|  |
| --- |
| Insert your screen shot here: |

* Next use SSH connection to work on files and directories on a remote VM (server) from your local VM (client):  
  => use a command like this to make the remote connection:  
   $ ssh username\_at\_server@ server\_ip

= > once connected, go to Documents folder => use ***ls*** to show file listing => use ***rm***command to remove the **reports** folder => use ***ls*** command to show the file listing again => take a screen shot of the terminal to show all these results

|  |
| --- |
| Insert your screen shot here: |

**Submitting your work:**

Export this Word document with all your answers and screen shots as PDF format, and then submit your PDF file via Lab 3 assignment tab on Moodle by *Wednesday, February 7, 2024 (midnight).*