BÁO CÁO LINUX ADMINSTRATION

VŨ NGỌC CƯỜNG

MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG BIỂU

TỔNG QUAN

A Linux Distribution is an Operating System made from a software collection that is based upon the Linux kerneand, often, a package management system.

The Linux OS comprises:

- the Linux Kernel
- the GNU shelutilities
- the graphicadesktop environment and more.

CHƯƠNG 1. LINUX TERMINAL

A TerminaEmulator and is a cruciapart of any Linux system because it basically allows you to access the system through a shell.

A shelis a program that takes commands from the user and gives them to the operating system's kerneto execute. It's also called the command interpreter. The shelgets started when the user logs in or starts the terminal.

Linux is a case-sensitive operating system.

1.1 Getting help - man page

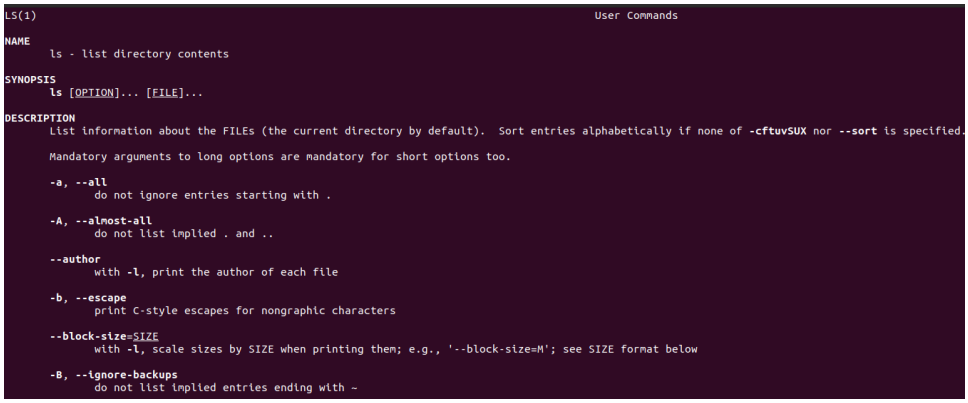
man command in Linux is used to display the user manuaof any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

Every manuais divided into the following sections:

- Executable programs or shelcommands
- System calls (functions provided by the kernel)
- Library calls (functions within program libraries
- Games
- Speciafiles (usually found in /dev)
- File formats and conventions eg /etc/passwd
- Miscellaneous (including macro packages and conventions), e.g. groff(7)
- System administration commands (usually only for root)
- Kerneroutines [Non standard]

MAN Pages

man command => Ex: man ls



```
LS(1)                                                                    User Commands

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default).  Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~
```

Hình 1-1 Ví dụ về man page ls

The man page is displayed with the less command

Bảng 1-1 Man page shortcut

h	getting help
q	quit
enter	show next line
space	show next screen
/string	search forward for a string
?string	search backwards for a string
n / N	next/previous appearance

Getting help for shelbuilt-in commands

help command => Ex: help cd

command --help => Ex: rm --help

Searching for a command, feature or keyword in alman Pages

man -k uname

man -k "copy files"

apropos passwd

1.2 Type

checking if a command is shelbuilt-in or executable file

type rm => rm is /usr/bin/rm

type cd => cd is a shelbuiltin

```
cuong@cuong-Vostro-3578:~$ type rm
rm is /usr/bin/rm
cuong@cuong-Vostro-3578:~$ type cd
cd is a shell builtin
cuong@cuong-Vostro-3578:~$
```

Hình 1-2 Ví dụ command type

1.3 Keyboard Shortcut

TAB	autocompletes the command or the filename if its unique
TAB TAB (press twice)	displays alcommands or filenames that start with those letters
CTR+ L	clearing the terminal
CTR+ D	closing the shel(exit)
CTR+ U	cutting (removing) the current line
CTR+ A	moving the cursor to the start of the line
Ctr+ E	moving the cursor to the end of the line
CTR+ C	stopping the current command
CTR+ Z	sleeping a the running program
CTR+ ALT + T	opening a terminal

1.4 Linux root access

On Linux there are 2 main categories of users:

1. non-privileged users - have no speciarights on the system.
2. The root user (superuser or the administrator).

- Root privileges are the powers that the root account has on the system. The root account is the most privileged on the system and has absolute power over it.
- Root exists on any Linux system is there's only one.
- It's not recommended to use root for ordinary tasks. When root permissions are needed you simply become root only to perform that particular administrative task.

Running a command as root (only users that belong to sudo group [Ubuntu] or whee[CentOS])

sudo command

becoming root temporarily in the terminal

sudo su => enter the user's password

setting the root password

sudo passwd root

changing a user's password

passwd username

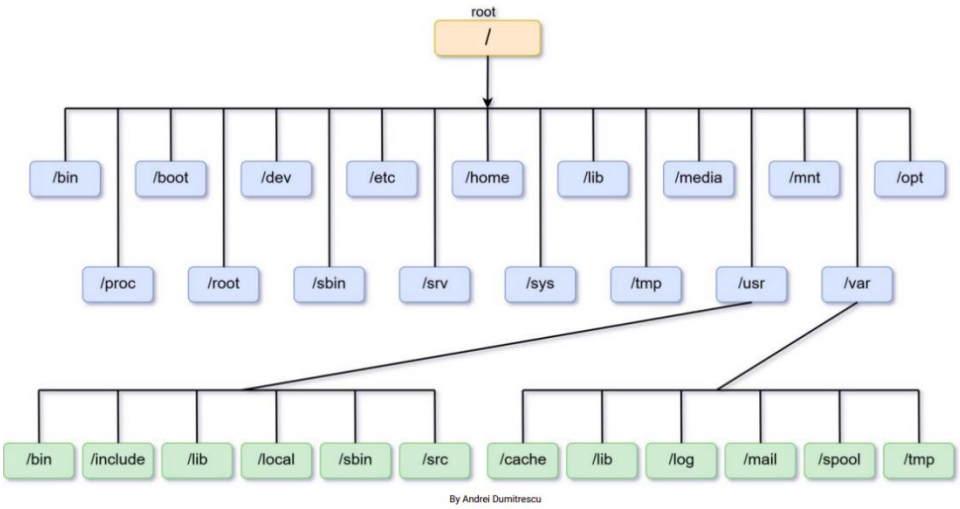
becoming root temporarily in the terminal

su => enter the root password

CHƯƠNG 2. LINUX FILE SYSTEM

- A file system controls how data is stored and retrieved.
- Each group of data is called a file and the structure and the logic rules used to manage files and their names are called file systems.
- A file system is a logicacollection of files on a partition or disk.
- On a Linux system, everything is considered to be a file.

On Linux file and directory names are case-sensitive.



Hình 2-1 Linux file system

2.1 The Filesystem Hierarchy Standard

/bin contains binaries or user executable files which are available to alusers.

/sbin contains applications that only the superuser (hence the initias) wilneed.

/boot contains files required for starting your system.

/home is where you wilfind your users' home directories. Under this directory there is another directory for each user, if that particular user has a home directory. root has its home directory separated from the rest of the users' home directories and is /root

/dev contains device files.

/etc contains most, if not alsystem-wide configuration files.

/lib contains shared library files used by different applications.

/media is used for externastorage wilbe automatically mounted.

/mnt is like /media but it’s not very often used these days.

/tmp contains temporary files, usually saved there by applications that are running. Non-privileged users may also store files here temporarily.

/proc is a virtuadirectory. It contains information about your computer hardware, such as information about your CPU, RAM memory or Kernel. The files and directories are generated when your computer starts, or on the fly, as your system is running and things change.

/sys contains information about devices, drivers, and some kernefeatures.

/srv contains data for servers.

/run is a temporary file system which runs in RAM.

/usr contains many other subdirectories binaries files, shared libraries and so on. On some distributions like CentOS many commands are saved in /usr/bin and /usr/sbin instead of /bin and /sbin.

/var typically contains variable-length files such as logs which are files that register events that happen on the system.

2.2 Linux Paths

Bảng 2-1 Linux path command

Symbol	Function
.	the current working directory
..	the parent directory
~	the user's home directory
cd	changing the current directory to user's home directory
cd ~	changing the current directory to user's home directory
cd -	changing the current directory to the last directory
cd /path_to_dir	changing the current directory to path_to_dir

pwd	printing the current working directory
sudo apt instaltree	installing tree
tree directory/	Ex: tree .
tree -d	prints only directories
tree -f	prints absolute paths

2.3 Ls commnads

Bång 2-2 listing the current directory with ls commands

~	user's home directory
.	current directory
..	parent directory
ls ~ /var /	listing more directories
ls -l	long listing
-a	listing all files and directories including hidden ones
ls -1 /etc	listing on a single column
ls -ld /etc	displaying information about the directory, not about its contents
ls -h /etc	displaying the size in human readable format
ls -Sh /var/log	displaying sorting by size

Note: ls does not display the size of a directory and alits contents. Use du instead

- du -sh ~
- -X => displaying sorting by extension
- ls -lX /etc

- --hide => hiding some files
- ls --hide=*.log /var/log
- -R => displaying a directory recursively
- ls -lR ~
- -i => displaying the inode number
- ls -li /etc

2.4 File timestamps

Every file on Linux has three timestamps:

- The access timestamp or atime is the last time the file was read (ls -lu)
- The modified timestamp or mtime is the last time the contents of the file was modified (ls -l, ls -lt)
- The changed timestamp ctime is the last time when some metadata related to the file was changed (ls -lc)

2.5 Viewing Files

Viewing files (cat, less, more, head, tail, watch)

Displaying the contents of a file: **cat filename**

Bảng 2-3 Command cat

cat filename1 filename2	displaying more files
can -n filename	displaying more files
cat filename1 filename2 > filename3	concatenating 2 files
less filename	viewing a file using less

Bảng 2-4 Command less

h	getting help
q	quit
enter	show next line
space	show next screen
/string	search forward for a string

?string	search backwards for a string
n / N	next/previous appearance

Bảng 2-5 Command tail

tail filename	showing the last 10 lines of a file
tail -n 15 filename	showing the last 15 lines of a file
tail -n +5 filename	showing the last lines of a file starting with line no. 5
tail -f filename	showing the last 10 lines of the file in real-time
head filename	showing the first 10 lines of a file
head -n 15 filename	showing the first 15 lines of a file
watch -n 3 ls -l	running repeatedly a command with refresh of 3 seconds

2.5.1 Working with files and directories

Bảng 2-6 Creating a new file or directory

touch filename	creating a new file or updating the timestamps if the file already exists
mkdir dir1	creating a new directory
mkdir -p mydir1/mydir2/mydir3	creating a directory and its parents as well

Bảng 2-7 The cp command

cp file1 file2	copying file1 to file2 in the current directory

cp file1 dir1/file2	copying file1 to dir1 as another name (file2)
cp -i file1 file2	copying a file prompting the user if it overwrites the destination
cp -p file1 file2	preserving the file permissions, group and ownership when copying
cp -v file1 file2	being verbose
cp -v file1 file2	being verbose
cp -r dir1 dir2/	recursively copying dir1 to dir2 in the current directory
cp -r file1 file2 dir1 dir2 destination_directory/	copy more source files and directories to a destination directory

Bảng 2-8 The mv command

mv file1 file2	renaming file1 to file2
mv file1 dir1/	moving file1 to dir1
mv -i file1 dir1/	moving a file prompting the user if it overwrites the destination file
mv -n file1 dir1/	preventing a existing file from being overwritten
mv -u file1 dir1/	moving only if the source file is newer than the destination file or when the destination file is missing
mv file1 dir1/file2	moving file1 to dir1 as file2
mv file1 file2 dir1/ dir2/ destination_directory/	moving more source files and directories to a destination directory

Bảng 2-9 The rm command

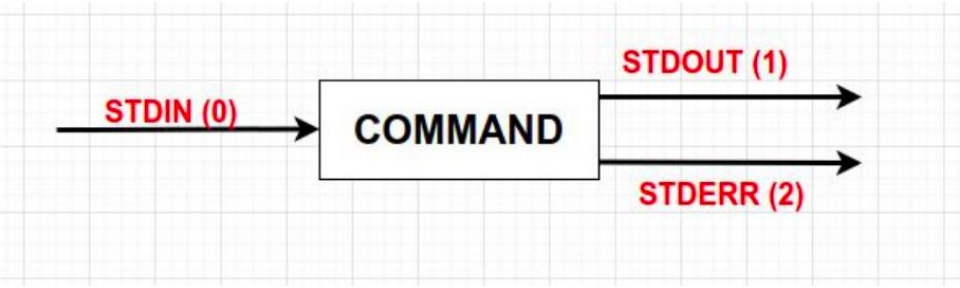
rm file1	removing a file
rm -v file1	being verbose when removing a file
rm -r dir1/	removing a directory
rm -rf dir1/	removing a directory without prompting
rm -ri fil1 dir1/	removing a file and a directory prompting the user for confirmation
shred -vu -n 100 file1	secure removal of a file (verbose with 100 rounds of overwriting)

2.5.2 Pipe

Pipe is a form of redirection that redirects standard output to some other destination for further processing

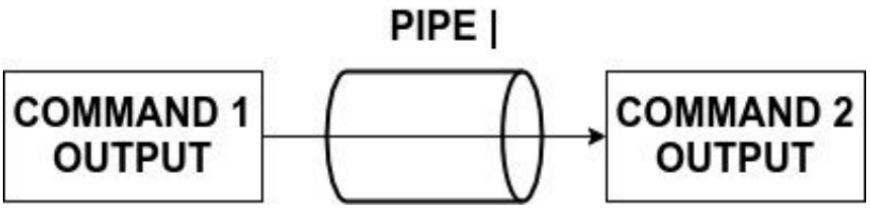
Every Linux command or program we run has three data streams connected to it:

1. STDIN (0) - Standard Input
2. STDOUT (1) - Standard Output
3. STDERR (2) - Standard Error



Hình 2-2 Linux commands

Using the pipe symbol (|) we can connect two or more commands at a time. command1 | command2 | command3 ….



Hình 2-3 Describe how pipe work

Piping Examples:

ls -lSh /etc/ | head # see the first 10 files by size

ps -ef | grep sshd # checking if sshd is running

ps aux --sort=-%mem | head -n 3 # showing the first 3 process by memory consumption

2.5.3 Command Redirection

output redirection

ps aux > running_processes.txt

who -H > loggedin_users.txt

appending to a file

id >> loggedin_users.txt

output and error redirection

tail -n 10 /var/log/*.log > output.txt 2> errors.txt

2.5.4 Finding Files and Dirrectories: Locate

Bảng 2-10 Finding Files and Directories: Locate

sudo updatedb	updating the locate db
locate -S	displaying statistics
locate filename	finding file by name
locate -i filename	

locate -b '\filename'	
locate -b filename	finding using the basename
locate -r 'regex'	finding using regular expressions
locate -e filename	checking that the file exists
which command	showing command path
which -a command	

showing command path

- which command
- which -a command

2.5.5 FIND

find PATH OPTIONS

Example: find ~ -type f -size +1M # => finding all files in ~ bigger than 1 MB

Options: -type f, d, l, s, p

- -name filename
- -iname filename # => case-insensitive
- -size n, +n, -n
- -perm permissions
- -links n, +n, -n
- -atime n, -mtime n, ctime n
- -user owner
- -group group_owner

Searching for text patterns (grep)

grep [OPTIONS] pattern file

Options:

Bảng 2-11 Grep options

-n	print line number
-i	case insensitive
-v	inverse the match
-w	search for whole words
-a	search in binary files
-R	search in directory recursively
-c	display only the no. of matches
-C n	display a context (n lines before and after the match)
strings binary_file	printing ASCII chars from a binary file

Note: Different between locate and find

Find: searches in real system. It slower but always up to date and more option (size, modify time)

Locate: Use a previous built database (updatedb)

- faster, but use older database

- Hard link and istruct node
- The inode structures

Each file on the disk has a data structure called index node or inode associated with it.

- This structure stores metadata information about the file such as the type, file's permission, file's owner and group owner, timestamp information, file size and so on.
- It actually contains all file information except the file contents and the name.

- Each inode is uniquely identified by an integer number called inode number (ls -i).

To see number of hard link: l

Two file point to one same data

Add link: ln a.xt

2.5.6 Hardlink and Softlink (Symbolic link)

Softlink: a special sort of file that point to a different file - like a shortcut the connection is a logical one, not a duplication can point at entire directory o link to files on remote computer

Hardlink: equivalent to a file stored in hard drive and it reference or point to spot on a hard drive

Is a mirror copy of original file

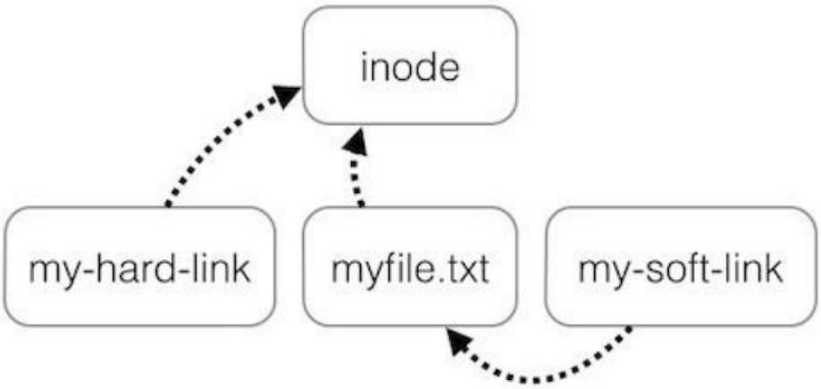
Delete ogirinal file don't affect a hard link but softlink have

Hardlink refer to data itself

Softlink point to the path to the data

Symlink can cross the file system

Hardlink is locally to the file system or the parition



Hình 2-4 Inode data structure

Add a soft link: ls –s a.txt b.txt

```
cuong@cuong-Vostro-3578:~/Documents$ ls -l
total 4188
-rw-rw-r-- 1 cuong cuong  531286 Thg 12  5 01:34 978-3-319-26555-1_10.pdf
-rw-rw-r-- 1 cuong cuong   30547 Thg 11 30 00:05 buy.png
drwxrwxr-x 2 cuong cuong    4096 Thg 12 22 21:19 DeckList
-rw-rw-r-- 1 cuong cuong 2008954 Thg 12 17 14:57 Emil_Laftchiev-Dissertation_11_20_2014.PDF
-rw-rw-r-- 1 cuong cuong    1730 Thg 5   4  2021 employee.csv
drwxrwxr-x 2 cuong cuong    4096 Thg 12 19 20:23 First
-rw-rw-r-- 1 cuong cuong   35617 Thg 12 19 12:24 kurikara.jpg
-rw-rw-r-- 1 cuong cuong 1387682 Thg 12 14 15:11 notes_017_wavelets_intro.pdf
drwxrwxr-x 2 cuong cuong    4096 Thg 12 19 20:51 Second
-rw-rw-r-- 1 cuong cuong  265432 Thg 12 21 00:24 sigmod_apca_2001.pdf
-rw-rw-r-- 1 cuong cuong     381 Thg 12 19 21:47 test.txt
cuong@cuong-Vostro-3578:~/Documents$
```

Hình 2-5 Output of ls -l

CHƯƠNG 3. USER ACCOUNT MANAGEMENT

3.1 Group

There are two types of groups that a user can belong to:

1. The primary group: the id is stored in /etc/passwd and the group name in /etc/group
2. Secondary groups: stored in /etc/group

```
cuong@cuong-Vostro-3578:~$ tail -n 5 /etc/group
wireshark:x:133:cuong
vboxusers:x:134:
mlocate:x:135:
rdma:x:136:
docker:x:137:
```

Hình 3-1 See user group

3.2 Account Management

IMPORTANT FILES

Bảng 3-1 Important files

/etc/passwd	Users and info: username:x:uid:gid:comment:home_directory:login_shell
/etc/shadow	users' passwords
/etc/group	groups

3.3 creating a user account

useradd [OPTIONS] username

Bảng 3-2 Useradd options

-m	create home directory
-d directory	specify another home directory
-c "comment"	
-s shell	

-G	specify the secondary groups (must exist)
-g	specify the primary group (must exist)

Exemple:

useradd -m -d /home/john -c "C Developer" -s /bin/bash -G sudo,adm,mail john

3.4 Changing a user account

usermod [OPTIONS] username

Uses the same options as useradd

Example:

usermod -aG developers,managers john # => adding the user to two secondary groups

Deleting a user account: -r removes user's home directory as well

userdel -r username

groupadd group_name	creating a group
groupdel group_name	deleting a group
cat /etc/groups	displaying all groups
groups	displaying the groups a user belongs to
usermod -aG sudo cuong	creating admin users (add the user to sudo group in Ubuntu and wheel group in CentOS)

Monitoring Users

who -H	displays logged in users
id	displays the current user and its groups

whoami	displays EUID
w	listing who's logged in and what's their current process
uptime	
last	printing information about the logins and logouts of the users
last -u username	

File Permission

- File permissions (file modes) specify who can access, change or execute a file on a Linux System.
- It ensures that only authorized users and processes can access files and directories.
- Each file or directory has an owner and a group. By default, the owner is the user who creates the file and the group is the primary group of that user.
- The ownership of a file or a directory can be changed only by root using the chown and chgrp commands.

For each file the permissions are assigned to three different categories of users:

1. The file owner.
2. The group owner.
3. Others (anyone else or the whole world)

The Octal Notation

The number that represents the permission in base-8 can be a either a 3 or a 4-digit number with digits from 0 to 7. The leading zero (0) can be omitted.

- 0755 = 755 and 0644 = 644.
- When a 3 digit number is used, the first digit represents the permissions of the file's owner, the second one the file's group, and the last one the permissions of the others class.
- r, w, and x have their own fixed number value:
  - r (read) = 4
  - w (write) = 2
  - x (execute) = 1
  - - (no permissions) = 0
  - The permissions number of a specific user class is represented by the sum of the values of the permissions for that group.

3.5 Changing File Permissions (chmod)

- chmod is the command used to change the permissions of a file or a directory using either the symbolic or the numeric notation.
- Only the root, or the file's owner, can change the file's permissions.

chmod [who][OPERATION][permissions] filename

who signifies the user category whose permissions will be changed.

- u: the user that owns the file.
- g: the group that the file belongs to.
- o: the other users.

The OPERATION flags define whether the permissions are to be removed, added, or set:

- -: a hyphen means remove the specified permissions.
- + : the plus sign means Add the specified permissions.
- = : equals means change the current permissions to the specified permissions.

The permissions are specified using the letters r, w and x.

Changing File Ownership (chown, chgrp)

In Linux, all files are associated with an owner and a group owner.

- The chown and chgrp commands are used to change the files owner and group.
- Only root can change the file owner.
- Normal users can change the group of the file only if they own the file and only to a group of which they are a member of. root can change the group ownership of all files.

Example:

ls -l hello.c

-rw-rw-r-- 1 root root 78 Thg 11 15 09:52 hello.c

chown cuongvn hello.c

ls -l hello.c

-rw-r--r-- 1 cuongvn root 78 Thg 11 15 09:52 hello.c

chgrp adm hello.c

ls -l hello.c

-rw-r--r-- 1 cuongvn adm 78 Thg 11 15 09:52 hello.c

3.5.1 Special Permissions - SUID (Set User ID)

- Besides r, w and x for the owner, group and others there are 3 extra special permissions for each file or directory: SUID or Set User ID, SGID or Set Group ID and Sticky Bit.
- These special permissions are for a file or directory overall, not just for a user category.
- When an executable file with SUID is executed then the resulting process will have the permissions of the owner of the command, not the permissions of the user who executes the command.

Setting SUID:

- Absolute Mode: chmod 4XXX file
- Relative Mode: chmod u+s file

ls -l /usr/bin/passwd

-rwsr-xr-x 1 root root 68208 apr 16 15:36 /usr/bin/passwd

3.5.2 Special Permissions - SGID (Set Group ID)

- SGID is set mainly to directories.
- If you set SGID on directories, all files or directories created inside that directory will be owned by the same group owner of the directory where SGID was configured.
- This is useful in creating shared directories, which are directories that are writable at the group level.

Setting SGID:

- Absolute Mode: chmod 2XXX directory
- Relative Mode: chmod g+s directory

ls -ld /programming/

drwxrwxr-x 10 cuongvn cuongvn 4096 Thg 12 1 10:40 DSA/

3.6 File permissions commands

LEGEND

- u = User
- g = Group
- = Others/World
- a = all
- r = Read
- w = write
- x = execute
- - = no access

displaying the permissions (ls and stat)

$ ls -l /etc/passwd

-rw-r--r-- 1 root root 2862 Thg 12 15 14:00 /etc/passwd

$ stat /etc/shadow

```
cuongvn@cuongvn:~$ stat /etc/shadow
  File: /etc/shadow
  Size: 1457            Blocks: 8          IO Block: 4096   regular file
Device: 803h/2051d      Inode: 1313345     Links: 1
Access: (0640/-rw-r-----)  Uid: (    0/    root)   Gid: (   42/  shadow)
Access: 2023-01-09 15:32:11.736000238 +0700
Modify: 2022-12-15 14:00:46.133277970 +0700
Change: 2022-12-15 14:00:46.133277970 +0700
 Birth: 2022-12-15 14:00:46.133277970 +0700
```

Hình 3-2 Output of stat /etc/shadow

Changing the permissions using the relative (symbolic) mode

- chmod u+r filename
- chmod u+r,g-wx,o-rwx filename
- chmod ug+rwx,o-wx filename
- chmod ugo+x filename
- chmod a+r,a-wx filename

Changing the permissions using the absolute (octal) mode

Bảng 3-3 Permissions example

Permission	Example
rwx rwx rwx	chmod 777 filename
rwx rwx r-x	chmod 775 filename
rwx r-x r-x	chmod 755 filename
rwx r-x ---	chmod 750 filename
rw- rw- r--	chmod 664 filename
rw- r-- r-- chmod	chmod 644 filename
rw- r-- ---	chmod 640 filename

Setting the permissions as of a reference file

chmod --reference=file1 file2

Changing permissions recursively

chmod -R u+rw,o-rwx filename

SUID (Set User ID)

Displaying the SUID permission

ls -l /usr/bin/umount

-rwsr-xr-x 1 root root 35192 Thg 2 21 2022 /usr/bin/umount

stat /usr/bin/umount

```
cuongvn@cuongvn:~$ stat /usr/bin/umount
  File: /usr/bin/umount
  Size: 35192           Blocks: 72         IO Block: 4096   regular file
Device: 803h/2051d      Inode: 656608      Links: 1
Access: (4755/-rwsr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2023-01-09 15:32:20.476000331 +0700
Modify: 2022-02-21 08:49:57.000000000 +0700
Change: 2022-11-15 09:12:01.368826599 +0700
 Birth: 2022-11-15 09:12:01.368826599 +0700
```

Hình 3-3 Output of stat /usr/bin/umount

3.6.1 Setting SUID

chmod u+s executable_file

chmod 4XXX executable_file # => Ex: chmod 4755 script.sh

3.6.2 SGID (Set Group ID)

Displaying the SGID permission

ls -ld Cuong/

drwxrwxr-x 5 cuongvn cuongvn 4096 Thg 1 3 15:45 Cuong/

stat Cuong/

```
cuongvn@cuongvn:~$ stat Cuong/
  File: Cuong/
  Size: 4096            Blocks: 8          IO Block: 4096   directory
Device: 803h/2051d      Inode: 1572909     Links: 5
Access: (0775/drwxrwxr-x)  Uid: ( 1000/ cuongvn)   Gid: ( 1000/ cuongvn)
Access: 2023-01-10 08:55:18.672434669 +0700
Modify: 2023-01-03 15:45:01.867775587 +0700
Change: 2023-01-03 15:45:01.867775587 +0700
 Birth: 2022-11-15 09:41:47.819498515 +0700
```

Hình 3-4 Output of stat dir/

3.6.3 Setting SGID

chmod 2750 projects/

chmod g+s projects/

3.6.4 The Sticky Bit

Displaying the sticky bit permission

ls -ld /tmp/

drwxrwxrwt 22 root root 4096 Thg 1 10 08:55 /tmp/

stat /tmp/

```
cuongvn@cuongvn:~$ stat /tmp/
  File: /tmp/
  Size: 4096            Blocks: 8          IO Block: 4096   directory
Device: 803h/2051d      Inode: 393217      Links: 22
Access: (1777/drwxrwxrwt)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2023-01-10 08:47:38.639134717 +0700
Modify: 2023-01-10 08:55:23.574817714 +0700
Change: 2023-01-10 08:55:23.574817714 +0700
 Birth: 2022-11-15 09:11:55.052017889 +0700
```

Hình 3-5 Output of stat /tmp

Setting the sticky bit

```
cuongvn@cuongvn:~$ mkdir temp
cuongvn@cuongvn:~$ chmod 1777 temp/
cuongvn@cuongvn:~$ chmod o+t temp/
cuongvn@cuongvn:~$ ls -ld temp/
drwxrwxrwt 2 cuongvn cuongvn 4096 Thg 1  10 09:21 temp/
```

Hình 3-6 Setting the sticky bit

3.7 UMASK

Displaying the UMASK

umask

Setting a new umask value

umask new_value # => Ex: umask 0022

Changing File Ownership (root only)

Bảng 3-4 Changing file ownership

chown new_owner file/directory	changing the owner
chgrp new_group file/directory	changing the group owner

chown new_owner:new_group file/directory	changing both the owner and the group owner
chown -R new-owner file/directory	changing recursively the owner or the group owner
lsattr filename	displaying the file attributes
chattr +-attribute filename	changing the file attributes

Note:

Each file and directory has an owner and a group.

Default: Owner is the user who creates the file and group is the primary group of that user.

CHƯƠNG 4. LINUX PROCESS

A running instance of a program is called a process and it runs in its own memory space. Each time you execute a command, a new process starts.

- A process is an active entity as opposed to a program, which is considered to be a passive entity.
- A new process is created only when running an executable file (not when running Shell builtin commands).

4.1 Type of Linux Commands:

- Executable file on the disk
- Shell builtin commands

4.2 Process properties

- PID (Process ID) - a unique positive integer number
- User
- Group
- Priority / Nice

4.3 Type of Processes

- Parent
- Child
- Daemon
- Zombie (defunct)
- Orphan

All the process in OS are created when another process execute folk() system call

➔ First process – pid =1 , init

Proceess used folk system call – parent process creative process is its child

4.3.1 Deamon

- Background process
- Contains names that finish withd ‘d’, Ex: sshd

- Linux begins deamons at starting time

4.3.2 Zombie

- OS maintians a table – associates every process to the data necessary for it funtioning
- When a process terminates its execution the OS, release most of the resources and information related to that process, a terminated process whose data has not been collected is called zombie
- Remove quikly from memory and don't use any of the system resources

4.3.3 Orphan

- Whose parent process has finished or terminated thought it ramins running itself

4.3.4 Thread vs Process

- Multiple threads can exist within the same process and they shared resources such as memory
- Other process not share resources

Example:

Text editor: process

Autosae: thread

4.3.5 Forgeground and background process

Forgeground:

- Started by user and not by system services
- While they are runnning, user cannot start another process in same terminal
- Default: All process run in forgeground
- Input: Keyboard -> output: terminal

Backgorund:

- Can start other process in same terminal
- Use: &

4.4 Commands - ps, pstree, pgrep

4.4.1 Process Viewing (ps, pstree, pgrep)

checking if a command is shell built-in or executable file

type rm # => rm is /usr/bin/rm

type cd # => cd is a shell built-in

4.4.2 Displaying all processes started in the current terminal

ps

Bảng 4-1 Display all process started in current terminal

ps -ef	displaying all processes running in the system
ps aux	
ps aux | less	
ps aux --sort=%mem | less	sorting by memory and piping to less
ps -ef --forest	ASCII art process tree
ps -f -u username	displaying all processes of a specific user
pgrep -l sshd	checking if a process called sshd is running
ps -ef | grep sshd	
pstree	displaying a hierarchical tree structure of all running processes
pstree -c	prevent merging identical branches

4.4.3 Commnads – top : Dynamic Real-Time View of Processes(top)

Starting top

top

top shortcuts while it's running

Bảng 4-2 Top shortcut while it's running

h	getting help
space	manual refresh
d	setting the refresh delay in seconds
q	quitting top
u	display processes of a user
m	changing the display for the memory
1	individual statistics for each CPU
x/y	highlighting the running process and the sorting column
b	toggle between bold and text highlighting
<	move the sorting column to the left
>	move the sorting column to the right
F	entering the Field Management screen
W	saving top settings

Bảng 4-3 Atribute when running top

PID	Unique process ID
USER	Username of process owner
PR	Priority
NI	Nice value
VIRY	Amount of virtual memory used by a process
RES	Amount of physical memory used by a process
SHR	Amount of memory shared with other process

S	State	D: uninteruotable sleep
		R: Running
		S: Sleeping
		T: Traced or stopped
		Z: Zombie
%CPU	%CPU used by the process	
%MEM	%RAM used by the process	
TIME+	Total CPU time consumed by the process	
Command	Command usd to activate the process	

Running top in batch mode (3 refreshes, 1 second delay)

top -d 1 -n 3 -b > top_processes.txt

Interactive process viewer (top alternative)

sudo apt update && sudo apt install htop # => Installing htop

htop

4.4.4 Killing processes (kill, pkill, killall)

Bång 4-4 Killing processes shortcut

kill -l	listing all signals
kill pid	sending a signal (default SIGTERM - 15) to a process by pid
kill -SIGNAL pid1 pid2 pid3 ...	sending a signal to more processes
kill -1 pid	sending a specific signal (SIGHUP - 1) to a process by pid
kill -HUP pid	

kill -SIGHUP pid	
pkill process_name	sending a signal (default SIGTERM - 15) to process by process name
killall process_name	
kill $(pidof process_name)	
command & # => Ex: sleep 100 &	running a process in the background
jobs	Showing running jobs
Ctrl + Z	Stopping (pausing) the running process
fg %job_id	resuming and bringing to the foreground a process by job_d
bg %job_id	resuming in the background a process by job_d
nohup command & # => Ex: nohup wget http://site.com &	starting a process immune to SIGHUP

CHƯƠNG 5. NETWORKING ON LINUX

5.1 Getting info about the network interfaces (ifconfig, ip, route)

Bảng 5-1 Getting info about the network interface (ifconfig, ip, route)

ifconfig	displaying information about enabled interfaces
ifconfig -a	displaying information about all interfaces (enabled and disabled)
ip address show	
ifconfig enp0s3	displaying info about a specific interface
ip addr show dev enp0s3	
ip -4 address	showing only IPv4 info
ip -6 address	showing only IPv6 info
ip link show	displaying L2 info (including the MAC address)
ip link show dev enp0s3	
route	displaying the default gateway
route -n # numerical addresses	
ip route show	
systemd-resolve --status	displaying the DNS servers

5.2 Setting the network interfaces (ifconfig, ip, route)

Bảng 5-2 Setting the network interfaces (ifconfig, ip, route)

ifconfig enp0s3 down	disabling an interface
ifconfig enp0s3 down	
ifconfig enp0s3 up	activating an interface

ip link set enp0s3 up	
ifconfig -a	checking its status
ip link show dev enp0s3	
ifconfig enp0s3 192.168.0.222/24 up	setting an ip address on an interface
ip address del 192.168.0.111/24 dev enp0s3	
ip address add 192.168.0.112/24 dev enp0s3	
ifconfig enp0s3:1 10.0.0.1/24	setting a secondary ip address on sub-interface

5.3 Deleting and setting a new default gateway

route del default gw 192.168.0.1

route add default gw 192.168.0.2

ip route del default

ip route add default via 192.168.0.1

5.4 Changing the MAC address

ifconfig enp0s3 down

ifconfig enp0s3 hw ether 08:00:27:51:05:a1

ifconfig enp0s3 up

changing the MAC address

ip link set dev enp0s3 address 08:00:27:51:05:a3

5.5 Network Static configuration using Netplan (Ubuntu)

1. Stop and disable the Network Manager

sudo systemctl stop NetworkManager

```
sudo systemctl disable NetworkManager
sudo systemctl status NetworkManager
sudo systemctl is-enabled NetworkManager
```

2. Create a YAML file in /etc/netplan

```
network:
    version: 2
    renderer: networkd
    ethernets:
    enp0s3:
        dhcp4: false
        addresses:
            - 192.168.0.20/24
        gateway4: "192.168.0.1"
        nameservers:
            addresses:
                - "8.8.8.8"
                - "8.8.4.4"
```

3. Apply the new config

```
sudo netplan apply
```

4. Check the configuration

```
ifconfig
route -a
```

SSH (Secure Shell)

- The SSH protocol is used for:
  - Secure Remote Management of Servers, Routers, other Networking Devices
  - Network File Copy: rsync, scp, sftp, winscp

- Tunneling, SSH Port Forwarding

sshd is the SSH server (daemon) and ssh or putty is the client Installation:

- Ubuntu: sudo apt update && sudo apt install openssh-server openssh-client
- CentOS: sudo dnf install openssh-server openssh-clients
- Checking its status: sudo systemctl status ssh

- Service Stop, Restart, Start: sudo systemctl [start|restart|stop] ssh
- Enable, Disable auto booting: sudo systemctl [enable|disable] ssh Server config file: /etc/ssh/sshd_config Client Config file: /etc/ssh/ssh_config

5.6 OpenSSH

1. Installing OpenSSH (client and server)

Ubuntu

sudo apt update && sudo apt install openssh-server openssh-client

CentOS

sudo dnf install openssh-server openssh-clients

Connecting to the server

ssh -p 22 username@server_ip # => Ex: ssh -p 2267 john@192.168.0.100

ssh -p 22 -l username server_ip

ssh -v -p 22 username@server_ip # => verbose

2. Controlling the SSHd daemon

Checking its status

sudo systemctl status ssh # => Ubuntu

sudo systemctl status sshd # => CentOS

Stopping the daemon

sudo systemctl stop ssh # => Ubuntu

sudo systemctl stop sshd # => CentOS

Restarting the daemon

sudo systemctl restart ssh # => Ubuntu

sudo systemctl restart sshd # => CentOS

Enabling at boot time

sudo systemctl enable ssh # => Ubuntu

sudo systemctl enable sshd # => CentOS

sudo systemctl is-enabled ssh # => Ubuntu

sudo systemctl is-enabled sshd # => CentOS

3. Securing the SSHd daemon

Change the configuration file (/etc/ssh/sshd_config) and then restart the server

man sshd_config

a) Change the port

Port 2278

b) Disable direct root login

PermitRootLogin no

c) Limit Users' SSH access

AllowUsers stud u1 u2 john

d) Filter SSH access at the firewall level (iptables)

e) Activate Public Key Authentication and Disable Password Authentication

f) Use only SSH Protocol version 2

g) Other configurations:

ClientAliveInterval 300

ClientAliveCountMax 0

MaxAuthTries 2

MaxStartUps 3

LoginGraceTime 20

5.7 Copying files using SCP and RSYNC

5.7.1 SCP

copying a local file to a remote destination

scp a.txt john@80.0.0.1:~

scp -P 2288 a.txt john@80.0.0.1:~ # using a custom port

copying a local file from a remote destination to the current directory

scp -P 2290 john@80.0.0.1:~/a.txt .

copying a local directory to a remote destination (-r)

scp -P 2290 -r projects/ john@80.0.0.1:~

5.7.2 RSYNC

synchronizing a directory

sudo rsync -av /etc/ ~/etc-backup/

mirroring (deleting from destination the files that were deleting from source)

sudo rsync -av --delete /etc/ ~/etc-backup/

excluding files

rsync -av --exclude-from='~/exclude.txt' source_directory/ destination_directory/

exclude.txt:

*.avi

music/

abc.mkv

rsync -av --exclude='*.mkv' --exclude='movie1.avi' source_directory/ destination_directory/

synchronizing a directory over the network using SSH

sudo rsync -av -e ssh /etc/ student@192.168.0.108:~/etc-backup/

using a custom port

sudo rsync -av -e 'ssh -p 2267' /etc/ student@192.168.0.108:~/etc-backup/

5.8 WGET

installing wget

apt install wget # => Ubuntu

dnf install wget # => CentOS

download a file in the current directory

wget https://cdimage.kali.org/kali-2020.2/kali-linux-2020.2-installer-amd64.iso

wget -c …	resuming the download
wget -P dir/	Download to specific direction
wget --limit-rate=100k	Limiting the rate
wget -i urls.txt	Download more files
wget -b	starting the download in the background
tail -f wget-log	checking its status

Getting an offline copy of a website

wget --mirror --convert-links --adjust-extension --page-requisites --no-parent http://example.org

wget -mkEpnp http://example.org

5.9 NETSTAT and SS

Displaying all open ports and connections

sudo netstat -tupan

sudo ss -tupan

netstat -tupan | grep :80 # => checking if port 80 is open

tupan:

- t: tcp port
- u: udp port
- p: pid
- a: all port
- n: numerical addr

5.10 LSOF

Bång 5-3 Lsof shortcut

lsof	listing all files that are open
lsof -u username	listing all files opened by the processes of a specific user
lsof -c sshd	listing all files opened by a specific process
lsof -iTCP -sTCP:LISTEN	listing all files that have opened TCP ports
lsof -iTCP -sTCP:LISTEN -nP	

5.11 Scanning hosts and networks using nmap

Scanning Networks is your own responsibility

Bảng 5-4 Scanning networks in the own respinsibility

nmap -sS 192.168.0.1	Syn Scan - Half Open Scanning (root only)
nmap -sT 192.168.0.1	Connect Scan
nmap -p- 192.168.0.1	Scanning all ports (0-65535)
nmap -p 20,22-100,443,1000-2000 192.168.0.	Specifying the ports to scan
nmap -p 22,80 -sV 192.168.0.1	Scan Version
nmap -sP 192.168.0.0/24	Ping scanning (entire Network)
nmap -Pn 192.168.0.0/24	Treat all hosts as online -- skip host discovery
nmap -sS 192.168.0.0/24 --exclude 192.168.0.10	Excluding an IP
nmap -oN output.txt 192.168.0.1	Saving the scanning report to a file
nmap -O 192.168.0.1	OS Detection
nmap -A 192.168.0.1	Enable OS detection, version detection, script scanning, and traceroute
nmap -p 80 -iL hosts.txt	reading the targets from a file (ip/name/network separated by a new line or a whitespace)
nmap -n -iL hosts.txt -p 80 -oN output.txt	exporting to out output file and disabling reverse DNS

CHƯƠNG 6. BASH SHELL SCRIPTING

6.1 Bash Aliases

Bảng 6-1 Bah aliases

alias	listing all Aliases
alias copy="cp -i"	creating an alias: alias_name="command"
to make the aliases you define persistent, add them to ~/.bashrc	
unalias copy	removing an alias: unalias alias_name

6.1.1 Useful Aliases

alias c="clear"

alias cl="clear;ls;pwd"

alias root="sudo su"

alias ports="netstat -tupan"

alias sshconfig="sudo vim /etc/ssh/sshd_config"

alias my_server="ssh -p 3245-l user100 80.0.0.1"

alias update=”sudo apt update && sudo apt dist-upgrade -y && sudo apt clean”

alias lt="ls -hSF --size -1"

alias ping='ping -c 5'

6.1.2 Interactive File Manipulation

alias cp="cp -i"

alias mv="mv -i"

alias rm="rm -i"

6.1.3 Important alias

This may look a bit confusing, but essentially, it makes all of the other aliases you define function correctly when used with sudo

alias sudo='sudo ' # use single quotes, not double quotes.

6.1.4 Shebang - là kí tự: #!

- Được đặt ở dòng đầu tiên của mỗi script
- Trong Linux Kernel system, khi script được chạy, đầu tiên program loader sẽ dựa vào Shebang để xác định script được chạy bởi trình biên dịch nào
- Nếu không có shebang, default chạy bằng shell

Ex: #! /urs/bin/python3

#: Bash comment

6.2 Running script

- By ./script_name
- By source script_name --> same as . script_name
- By shebang

If dont have execute permission

- permission denied
- still running

Script will execute in a new shell

script will execute in current shell

Note: If alias name is the same as command

Ex: rm for rm -i

⇨ Run rm not alias

6.3 Bash Variables

Bảng 6-2 Bash variables

variable_name=value	defining a variable
variable_name	# referencing the value of a variable (getting the variable value)
declare -r temperature=100	defining a read-only variable (constant)
unset version	removing (unsetting) a variable
env printenv	listing all environment variables
printenv PATH env | grep -i path	searching for an environment variable
export PATH=$PATH:~/scripts # in ~/.bashrc	changing the PATH
read MY_VAR echo $MY_VAR	getting user input
read -p "Enter the IP address: " ip ping -c 1 $ip read -s -p "Enter password:" pswd echo $pswd	displaying a message

6.3.1 Special variables and positional arguments

./script.sh filename1 dir1

Bảng 6-3 Special variables and posional arguments

$0	the name of the script itself (script.sh)
$1	the first positional argument (filename1)
$2	the second positional argument (dir1)
${10}	the tenth argument of the script
$#	the number of the positional arguments
"$*"	string representation of all positional argument
$?	the most recent foreground command exit status

6.4 Coding - If...Elif...Else Statements

```
if [ some_condition_is_true ]
then
        //execute this code
elif [ some_other_condition_is_true ]
then
        //execute_this_code
else
        //execute_this_code
Fi
```

6.5 Testing conditions

TESTING CONDITIONS => man test

For numbers (integers)

Bảng 6-4 Testing condition for number

-eq	equal to
-ne	not equal to
-lt	less than
-le	less than or equal to
-gt	greater than
-ge	greater than or equal to

For files:

Bảng 6-5 Testing condition for files

-s	file exists and is not empty
-f	file exists and is not a directory
-d	directory exists
-x	file is executable by the user
-w	file is writable by the user
-r	file is readable by the user

For Strings

Bảng 6-6 Testing condition for string

=	the equality operator for strings if using single square brackets [ ]
==	the equality operator for strings if using double square brackets [[ ]]
!=	the inequality operator for strings
-n $str	str is nonzero length
-z $str	str is zero length
&&	the logical and operator
||	the logical or operator

CHƯƠNG 7. TÀI LIỆU THAM KHẢO

[1] "Linux Admin Tutorial." https://www.tutorialspoint.com/linux_admin/index.htm (accessed Jan. 10, 2023).

[2] "Linux System Administration Basics - Part 1," Dec. 22, 2019. https://viblo.asia/p/linux-system-administration-basics-part-1-RnB5px1b5PG (accessed Jan. 10, 2023).

[3] "Linux Administration: The Complete Linux Bootcamp," *Udemy*. https://www.udemy.com/course/master-linux-administration/ (accessed Jan. 10, 2023).