Intro to Unix And Linux

Essential Commands

Rotate your screen

```
xrandr --output HDMI-0 --rotate left
```

The **`file`** Command

- Gives you the type of contents in a file
- Type the following commands:
  - **`file /etc/passwd`**
  - **`file /bin/ls`**

The `cat` Command

- Displays the contents of a file
  - Useful for viewing small files
- Use the “-n” option to add line number
  - Useful for viewing progams
- Type the following commands
  - **`cat /etc/fstab`**
  - **`cat -n .bashrc`**
  - **`cat /usr/share/dict/linux.words`**

The **`more`** and **`less`** commands

- Useful for viewing large files
- Type the commands
  - **`more /etc/passwd`**
  - **`less /etc/passwd`**
- Commands are similar.
- **`less`** is newer and can scroll backwards.
- Press “h” for help to see options.
- Often used by “piping” the output of anther command to one of them:
  - **`ls -l /etc | more`**

The `head` and `tail` Commands

- **`head`** shows the first 10 lines of a file
- **`tail`** shows the last 10 lines of a file
- The number of lines can be changes with the –n option
- Type the commands
  - **`head /etc/passwd`**
  - **`tail /etc/passwd`**
  - **`head –n 5 /etc/passwd`**

The `wc` command

- wc stands for “word count” but it gives the
  - Number of lines
  - Number or words
  - Number of bytes

  In a file

  Type the command:

```
wc .bash_profile
```

The **`locate`** and **`find`** commands

- **`locate`** searches a database of files for ones whose names match the argument
- **`find`** searches a specific directory
- Type the commnds
  - **`locate passwd`**
  - **`locate words | head`**
  - **`find /etc -name passwd`**
- The **`find`** command has many options
  - Use **`man find`** to list them

The **`cmp`** and **`diff`** commands

- **`cmp`** compares files to see if they are different
- **`diff`** shows the differences
- Type the commands
  - **`ls > myfiles1`**
  - **`ls -l`**
  - **`ls > myfiles2`**
  - **`ls -l`**
  - **`cmp myfiles1 myfiles2`**
  - **`diff myfiles1 myfiles2`**
  - **`rm myfiles*`**

The **`diff`** command (continued)

- The diff command tells us the changes we need to make to get the files to match. For details see:

https://www.geeksforgeeks.org/diff-command-linux-examples/

https://www.computerhope.com/unix/udiff.htm

Shell Variables

- The **`set`** command by itself displays the names and values of all shell variables
- To see the first 10, type:
  - **`set | head`**

Aliases

- Suppose you frequently type the command:
  - **`ls -al | more`**
- To save some keystrokes, you could create an alias:
  - **`alias dir='ls -al |more'`**
- Then just type the command:
  - **`dir`**

- If you put the alias command in your **`.bashrc`** or **`.bash_profile`** file, your alias will be available every time you login

Edit the .bashrc file

- Use the nano text editor to edit the .bashrc file
- Type “cd” and press Enter to make sure you are in your home directory
- Type “nano .bashrc” to open the file in the editor
- Use your arrow keys, not your mouse, to move to the last line
- Type the alias from the last page:
  – **`alias dir='ls -al |more'`**

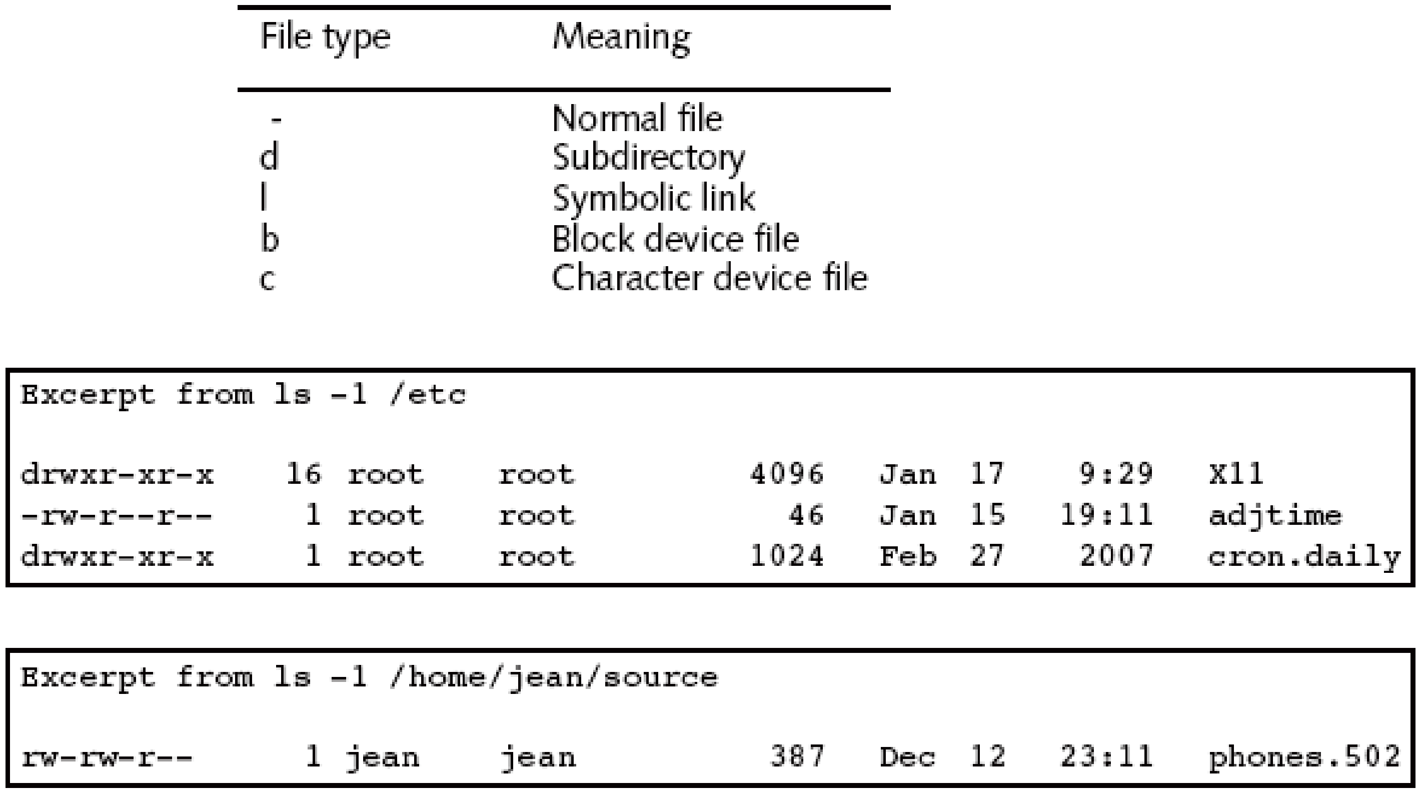
History

- Commands that you execute in a shell are saved in memory so you can execute them again.
- You can scroll back through them with the up arrow
- To see all these commands, execute the **`history`** command (with the tail command to limit output)
  - **`history |tail -5`**
- Each command is assigned a number.
- You can re-execute a command by using this number with an ! character preceding it:
  - **`!12`**

Configuring File Permissions for Security

- Users can set **permissions** for files/directories they own so as to establish security
  - System administrators also set permissions to protect system and shared files
- Permissions manage who can read, write, or execute files
- Original file owner of a file is the account that created it
  - File ownership can be transferred to another account

Configuring File Permissions for Security (continued)



File type	Meaning
-	Normal file
d	Subdirectory
l	Symbolic link
b	Block device file
c	Character device file

```
Excerpt from ls -l /etc

drwxr-xr-x   16  root    root         4096   Jan  17    9:29   X11
-rw-r--r--    1  root    root           46   Jan  15   19:11   adjtime
drwxr-xr-x    1  root    root         1024   Feb  27    2007   cron.daily
```

```
Excerpt from ls -l /home/jean/source

rw-rw-r--     1 jean    jean           387   Dec  12   23:11   phones.502
```

Figure 2-6 File types described in directory listings

Configuring File Permissions for Security (continued)

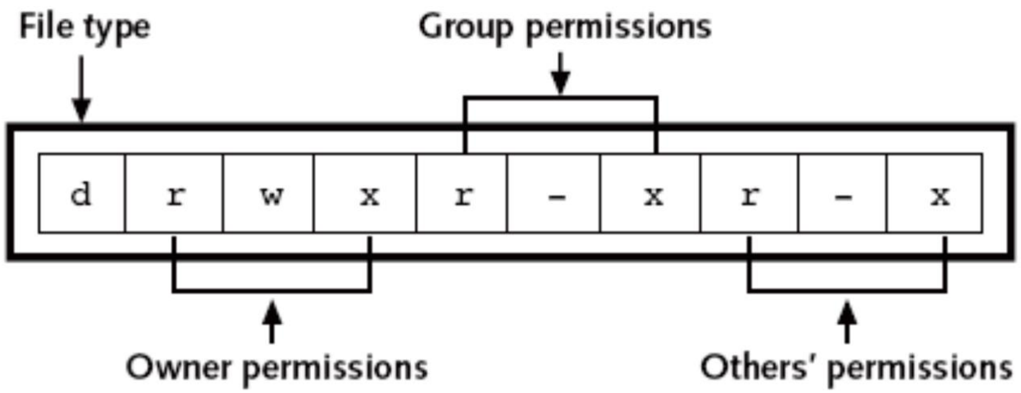


Figure 2-7 Example of the file type and the file permissions for a file

Syntax **chmod** [-option] *mode filename*

Dissection

- The argument can include the mode (permissions) and must include the file name. You can also use a wildcard to set the permissions on multiple files.
- Permissions are applied to owner (u), group (g), and others (o). The permissions are read (r), write (w), and execute (x). Use a plus sign (+) before the permissions to allow them or a hyphen (-) to disallow permissions. Octal permissions are assigned by a numeric value for each owner, group, and others.

Configuring File Permissions for Security (continued)

- The system administrator assigns group ids when he or she adds a new user account
  - A **group id (GID)** gives a group of users equal access to files that they all share
- Using *chmod* to change permissions of a file:

```
chmod ugo+rwx myfile
chmod go-wx account_info
```

  - Or, use the octal permission format

```
chmod 711 data
chmod 644 data
```

Configuring File Permissions for Security (continued)

- Try these commands

```
ls -l /etc > etc.files
ls -l
chmod  640 etc.file
ls -l
chmod 775 etc.file
ls -l
rm etc.file
```

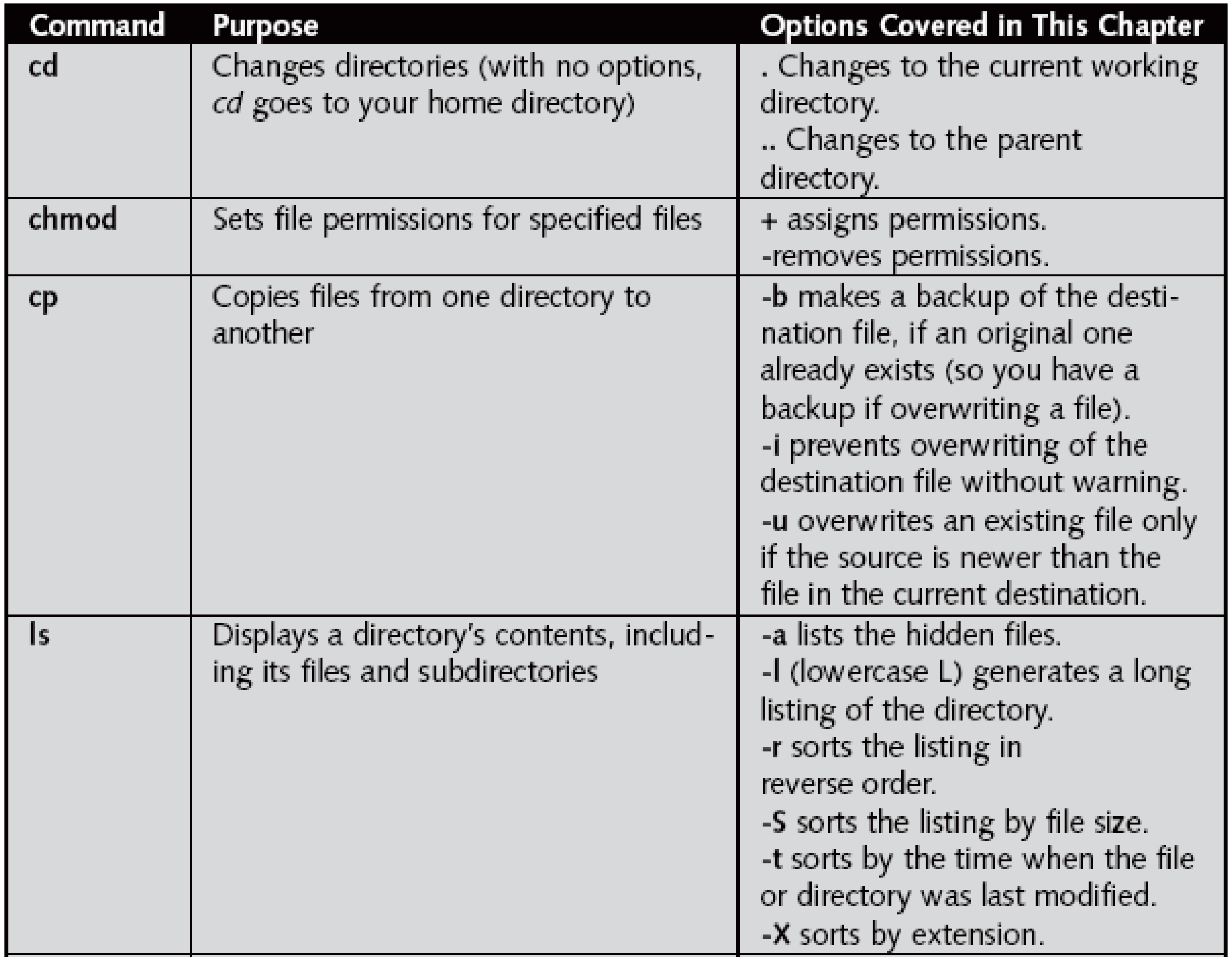
Configuring File Permissions for Security (continued)

- **Sticky bit:** t (used in place of x)
  - Enables file to be executed, but only the file's owner or root have permission to delete or rename it

- **Set user id (SUID) bit:** s (used in place of x)
  - Gives current user temporary permissions to execute program-related files as though they are the owner

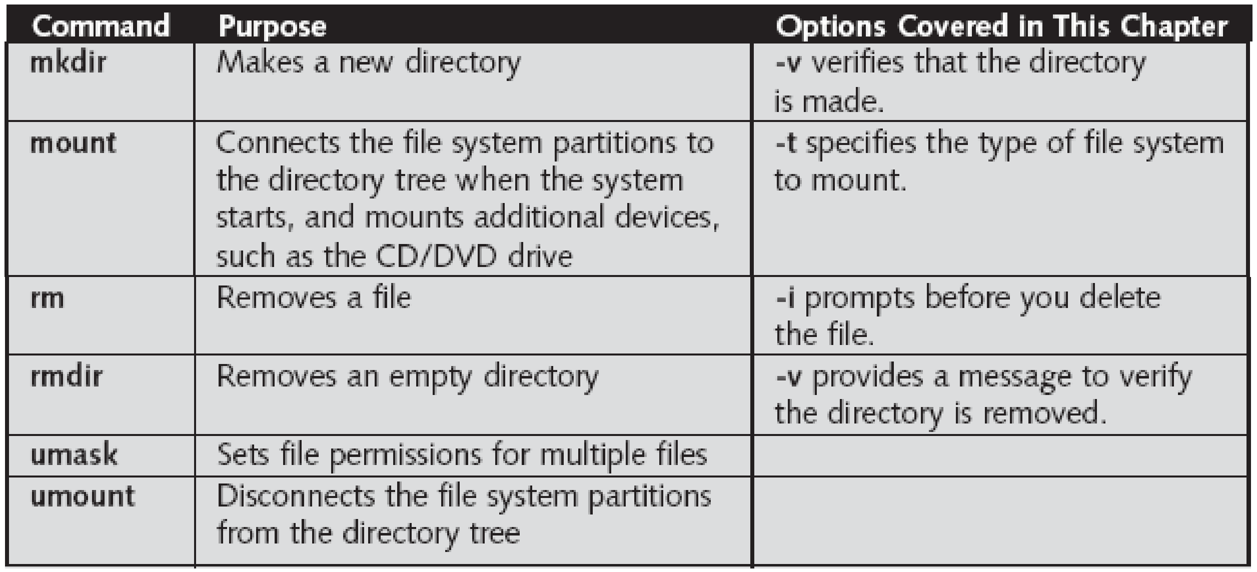
- **Set group ID (SGID) bit:** s (used in place of x)
  - Similar to SUID, but applies to groups

Command Summary



Command	Purpose	Options Covered in This Chapter
cd	Changes directories (with no options, *cd* goes to your home directory)	. Changes to the current working directory. .. Changes to the parent directory.
chmod	Sets file permissions for specified files	+ assigns permissions. -removes permissions.
cp	Copies files from one directory to another	**-b** makes a backup of the destination file, if an original one already exists (so you have a backup if overwriting a file). **-i** prevents overwriting of the destination file without warning. **-u** overwrites an existing file only if the source is newer than the file in the current destination.
ls	Displays a directory's contents, including its files and subdirectories	**-a** lists the hidden files. **-l** (lowercase L) generates a long listing of the directory. **-r** sorts the listing in reverse order. **-S** sorts the listing by file size. **-t** sorts by the time when the file or directory was last modified. **-X** sorts by extension.

Command Summary (continued)



Command	Purpose	Options Covered in This Chapter
mkdir	Makes a new directory	**-v** verifies that the directory is made.
mount	Connects the file system partitions to the directory tree when the system starts, and mounts additional devices, such as the CD/DVD drive	**-t** specifies the type of file system to mount.
rm	Removes a file	**-i** prompts before you delete the file.
rmdir	Removes an empty directory	**-v** provides a message to verify the directory is removed.
umask	Sets file permissions for multiple files	
umount	Disconnects the file system partitions from the directory tree	

Rotate Screen at Login

- To rotate you right screen automatically at login:
  - Create a file called “setscreen.desktop” by typing
    - `nano setscreen.desktop`
  - Type the following lines in the file and save it

```
[Desktop Entry]
Name=Set Screen Rotation
Exec=/bin/bash -c "xrandr --output HDMI-0 --rotate left"
Type=Application
```

  - Copy the file to the autostart directory by typing
    - `sudo cp setscreen.desktop /etc/xdg/autostart`