

Practical 7 – Linux Administration - Files and Directories

Pre-Requisites

Objectives

- Standard Directories
- Listing Directory Contents
- Absolute and Relative Pathnames
- Hidden Files
- Creating and Switching Directories
- Copying Files
- Moving Files
- Removing Files and Directories
- File Permission
- Links

Exercise 1 - Standard Directories

1. Run the "**ls -l /**" to display the content of the root directory.

2. Standard sub-directories are:

S/N	Sub-directory	Description
1.	/bin	Contains user executable programs. For example, the ls program is located in /bin.
2.	/sbin	Contains system executable programs used by the root user and the system. For example, the clock program is located in /sbin.
3.	/lib	Contains shared library files used by /bin and /sbin.
4.	/dev	Contains special file system entries for devices attached to the system.
5.	/boot	Contains the Linux kernel and bootloader programs. The Linux kernel program is typically known as "vmlinuz".
6.	/etc	Contains system configuration files. Files contain user account information are located here.
7.	/proc	Contains special files pertaining to the state of the running Linux system. These files are virtual files.
8.	/mnt	Contains temporarily mounted file systems.
9.	/usr	Contains programs that can be run any users of the system.
10.	/var	Contains variable data files pertaining to the on-going system status such as log files of system activities.
11.	/home	Contains sub-directories of user accounts to store personal data files.
12.	/tmp	Contains temporary files.
13.	/root	This is the home directory of the root user.
14.	/opt	Contains software packages for installation. Packages are stored in sub-directories under the /opt.

3. Run the "**ls** /bin" to display the content of the "/bin" directory.

Did you notice any familiar commands in this directory?

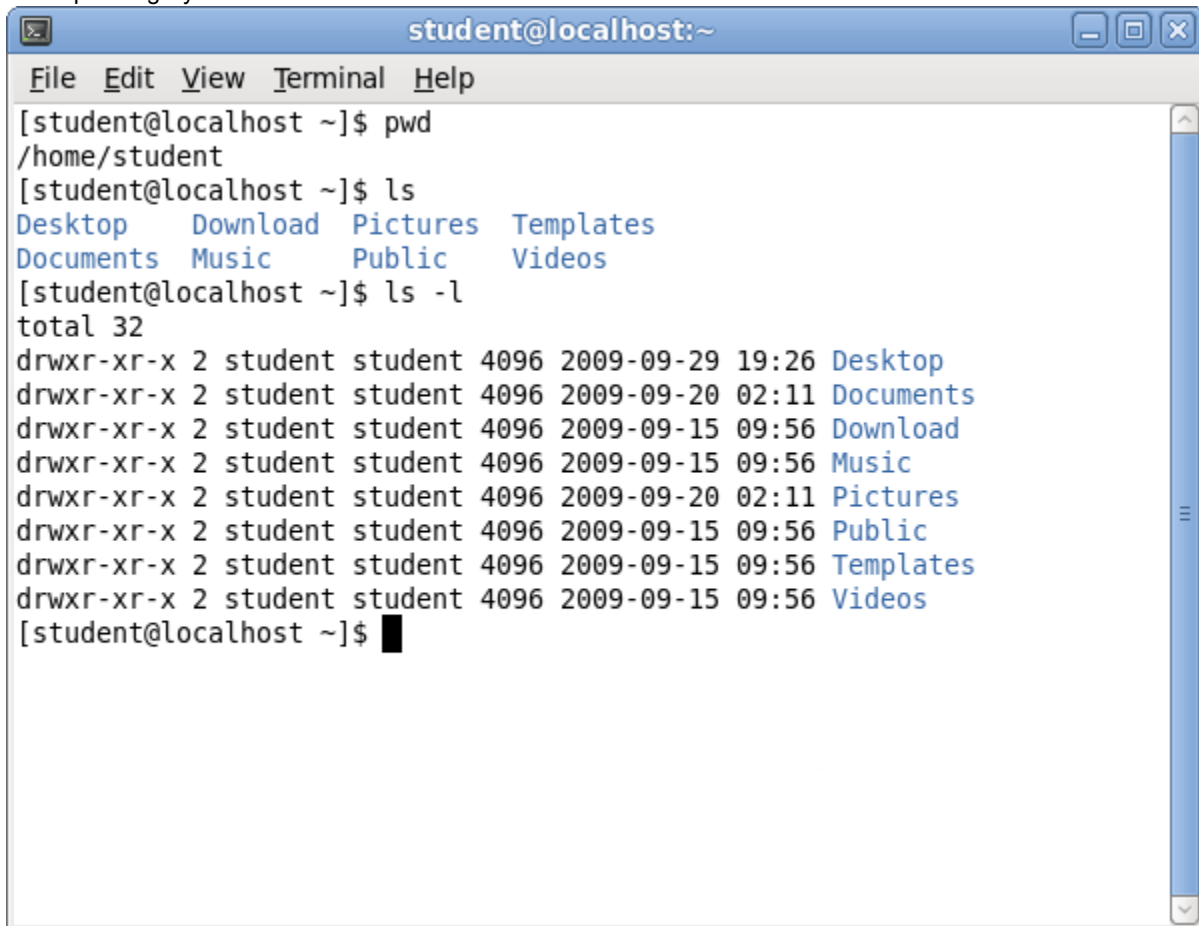
```

student@localhost:~
File Edit View Terminal Help
[student@localhost ~]$ ls /bin
alsaunmute      ed              mknod           sh
arch            egrep          mktemp         sleep
awk            env            more           sort
basename       ex            mount          stty
bash           false         mountpoint     su
cat            fgrep         mv            sync
chgrp          find          netstat       tar
chmod          fusermount    nice          taskset
chown          gawk          nisdomainname touch
cp            grep          ntfs-3g       tracepath
cpio           gtar          ntfs-3g.probe tracepath6
cut           gunzip        ntfsmount     traceroute
date          gzip          ping          traceroute6
dbus-cleanup-sockets hostname      ping6         true
dbus-daemon   ipcalc       plymouth      unlockmgr_server
dbus-monitor  iptables-xml ps            uname
dbus-send     kbd_mode    pwd           unicode_start
dbus-uuidgen  kill        red           unicode_stop
dd           link        rm            unlink
df           ln          rmdir        usleep
dmesg        loadkeys    rpm          vi
dnsdomainname login        rvi          view
doexec       ls          rview        view
domainname   mail        sed          ypdomainname
dumpkeys     mailx       setfont      zcat
echo         mkdir       setserial
[student@localhost ~]$

```

Exercise 2 - Listing Directory Contents

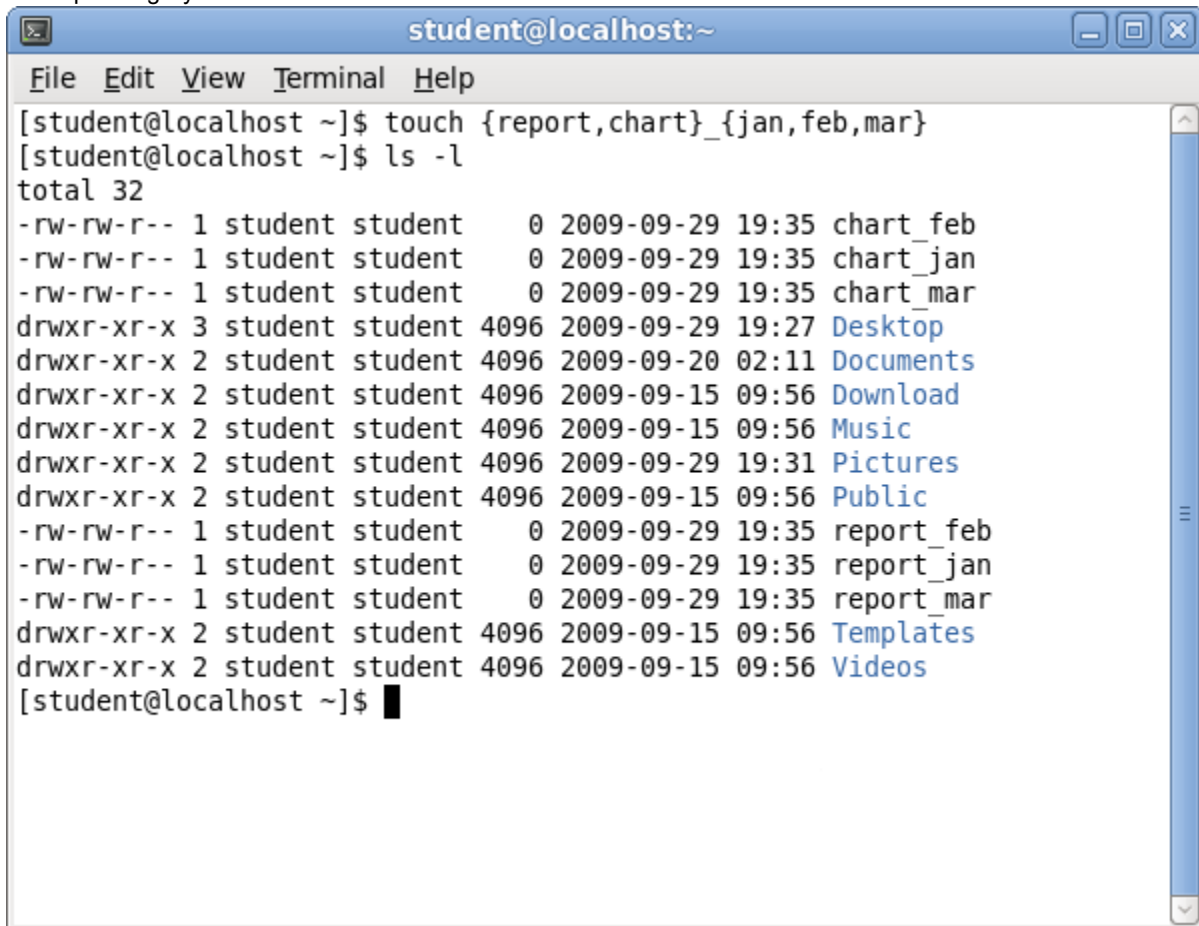
1. Run the **pwd** command to display the current directory.
 Run the **ls** command to display the content of the current directory.
 Run the "**ls -l**" command to display the content of the current directory in long format.

A terminal window titled 'student@localhost:~' with a menu bar (File, Edit, View, Terminal, Help). The terminal shows the following commands and output:

```
[student@localhost ~]$ pwd
/home/student
[student@localhost ~]$ ls
Desktop  Download  Pictures  Templates
Documents Music      Public    Videos
[student@localhost ~]$ ls -l
total 32
drwxr-xr-x 2 student student 4096 2009-09-29 19:26 Desktop
drwxr-xr-x 2 student student 4096 2009-09-20 02:11 Documents
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Download
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Music
drwxr-xr-x 2 student student 4096 2009-09-20 02:11 Pictures
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Public
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Templates
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Videos
[student@localhost ~]$
```

Run the **touch** command as shown to create some files.

Run the "**ls -l**" command to display the files created with this command.

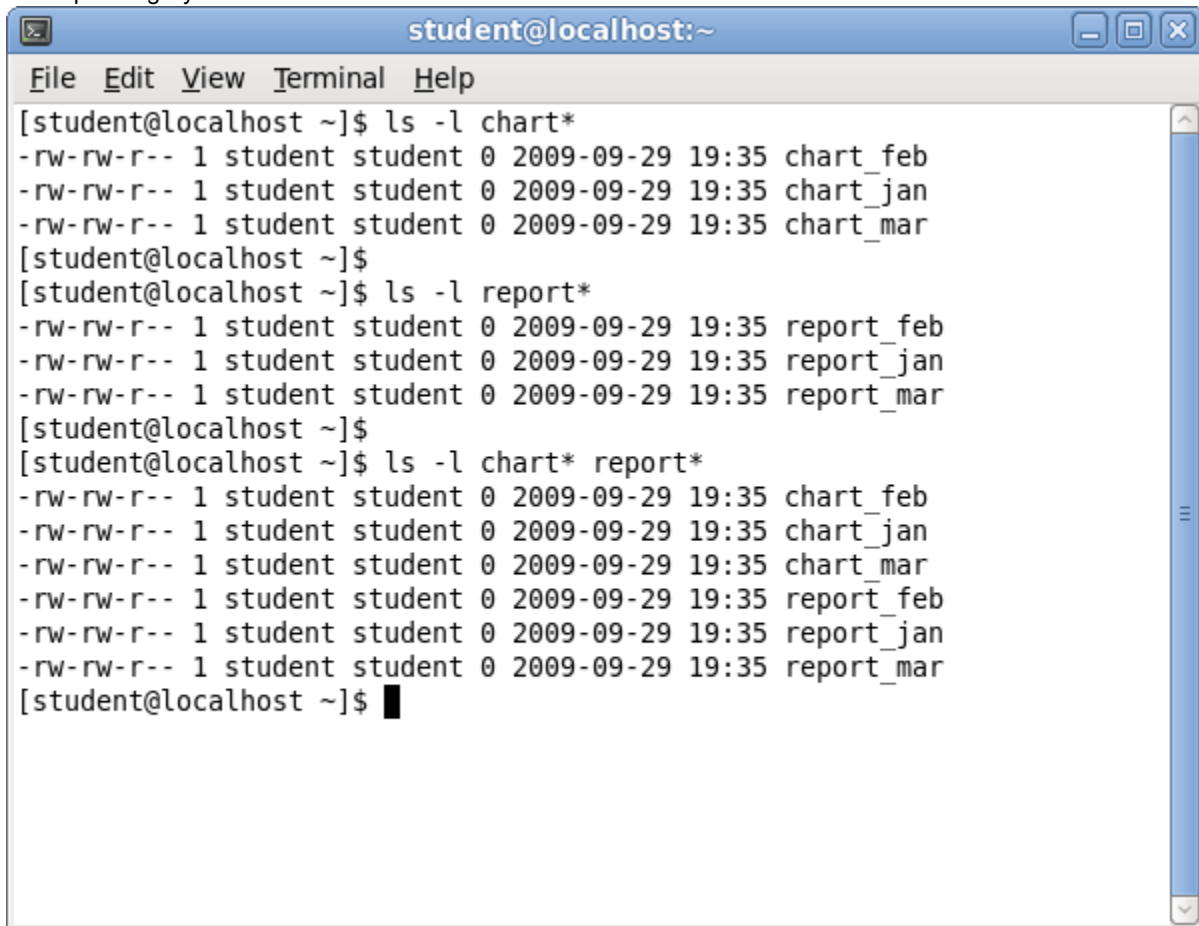
A terminal window titled 'student@localhost:~' with a menu bar (File, Edit, View, Terminal, Help). The terminal shows the execution of 'touch {report,chart}_{jan,feb,mar}' and 'ls -l'. The output of 'ls -l' lists files with permissions, owner, group, size, date, time, and name. The files listed are chart_feb, chart_jan, chart_mar, Desktop, Documents, Download, Music, Pictures, Public, report_feb, report_jan, report_mar, Templates, and Videos. The prompt '[student@localhost ~]\$' is followed by a cursor.

```
student@localhost:~  
File Edit View Terminal Help  
[student@localhost ~]$ touch {report,chart}_{jan,feb,mar}  
[student@localhost ~]$ ls -l  
total 32  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_mar  
drwxr-xr-x 3 student student 4096 2009-09-29 19:27 Desktop  
drwxr-xr-x 2 student student 4096 2009-09-20 02:11 Documents  
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Download  
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Music  
drwxr-xr-x 2 student student 4096 2009-09-29 19:31 Pictures  
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Public  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_mar  
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Templates  
drwxr-xr-x 2 student student 4096 2009-09-15 09:56 Videos  
[student@localhost ~]$
```

Run the "**ls -l chart***" command to display all files with filename starting with "chart".

Run the "**ls -l report***" command to display all files with filename starting with "report".

Run the "**ls -l chart* report***" command to display all files with filename starting with "chart" and "report".



```
student@localhost:~  
File Edit View Terminal Help  
[student@localhost ~]$ ls -l chart*  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_mar  
[student@localhost ~]$  
[student@localhost ~]$ ls -l report*  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_mar  
[student@localhost ~]$  
[student@localhost ~]$ ls -l chart* report*  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 chart_mar  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 19:35 report_mar  
[student@localhost ~]$
```

Exercise 3 - Absolute and Relative Pathnames

1. Absolute path starts from the root and the path name begins with a slash.

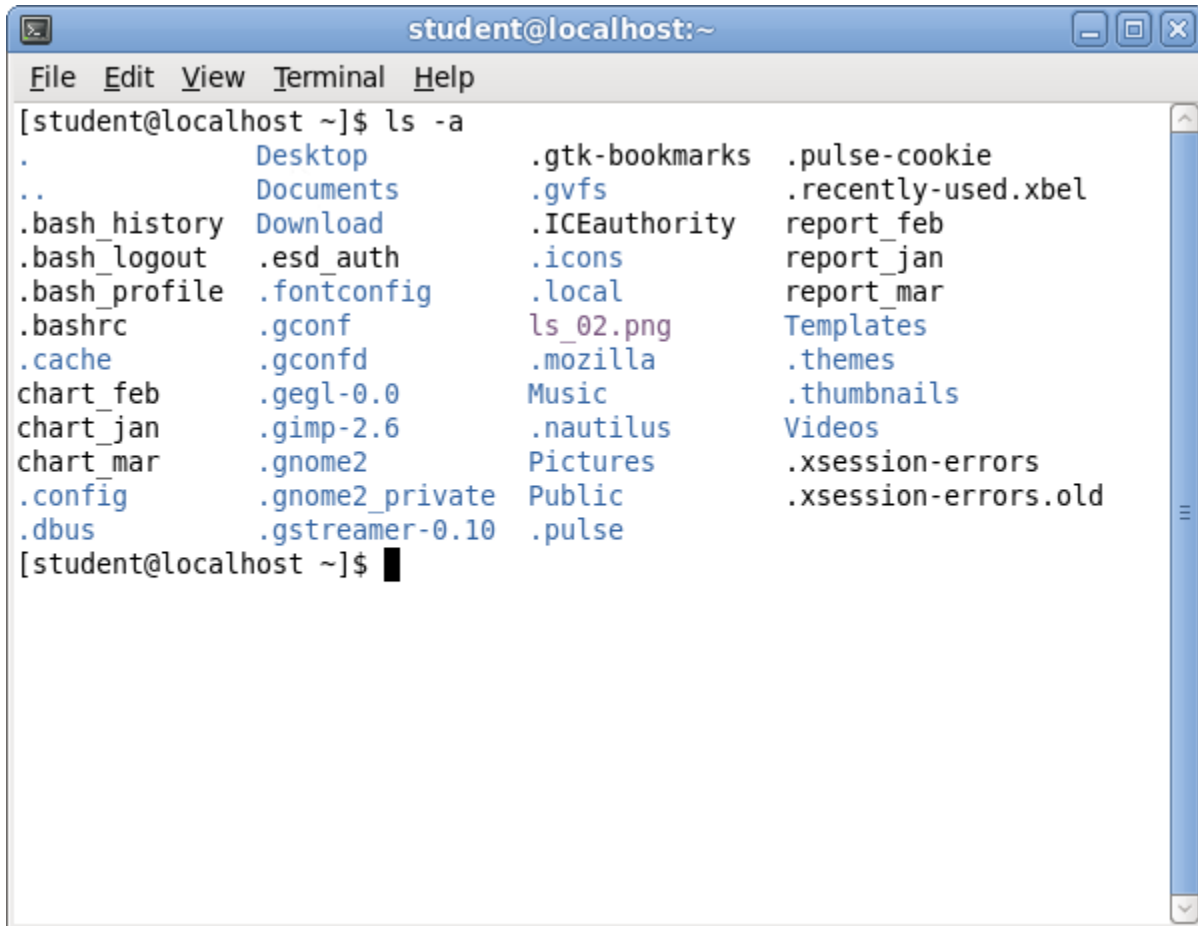
Relative paths reference the current directory and the path name does not begin with a slash.

Note

```
cd /etc  
cd ~/tmp  
cd ../backup  
cd Desktop
```

Exercise 4 - Hidden Files

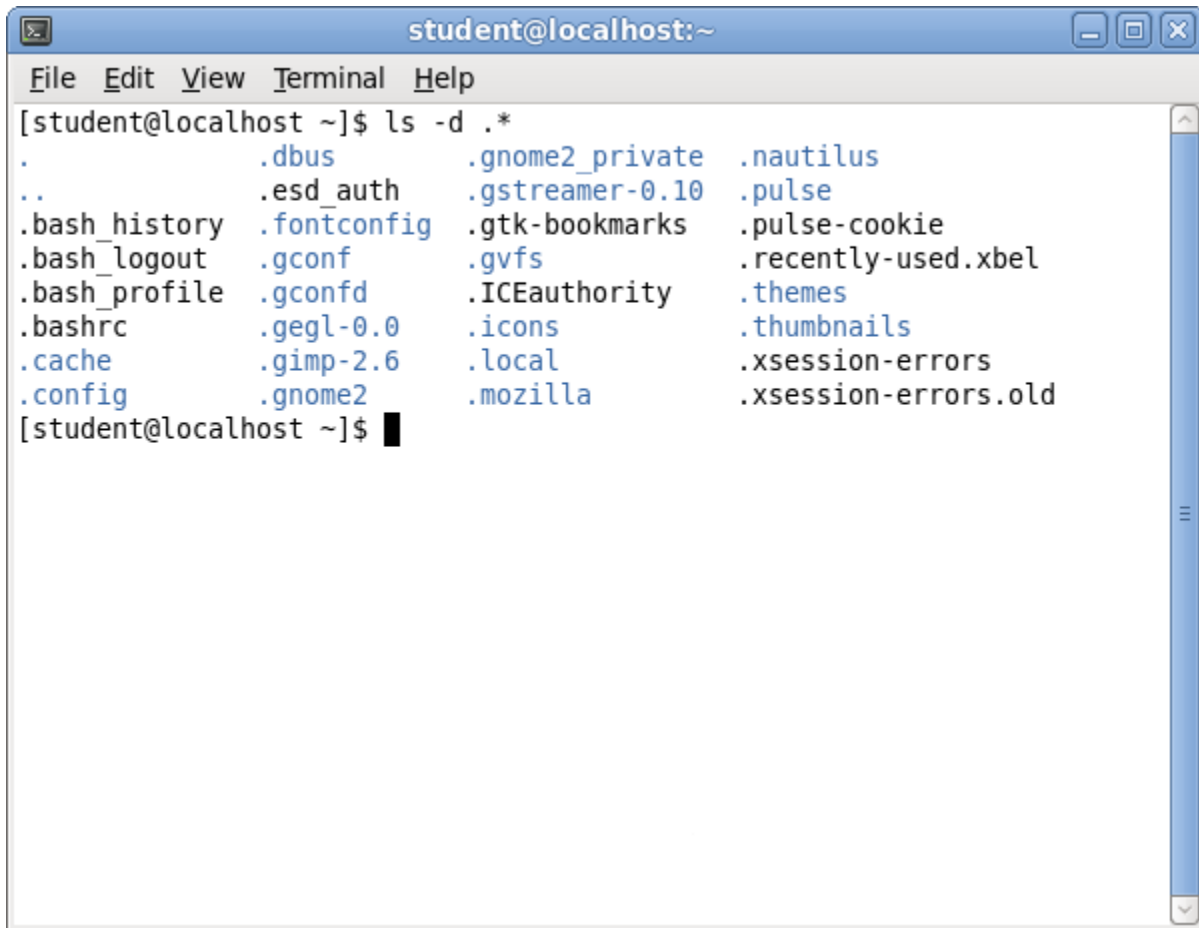
1. The filename of hidden files begin with a dot.



A terminal window titled "student@localhost:~" with a menu bar (File, Edit, View, Terminal, Help). The command "[student@localhost ~]\$ ls -a" has been executed, displaying a list of files and directories in four columns. The output includes standard system directories like Desktop, Documents, and Downloads, as well as various hidden files and configuration files such as .bash_history, .cache, .config, .dbus, .fontconfig, .gconf, .gstreamer-0.10, .gtk-bookmarks, .gvfs, .ICEauthority, .icons, .local, .mozilla, .nautilus, .pulse, .pulse-cookie, .recently-used.xbel, .themes, .thumbnails, .xsession-errors, and .xsession-errors.old. The prompt "[student@localhost ~]\$ " is visible at the bottom.

```
[student@localhost ~]$ ls -a
.          Desktop      .gtk-bookmarks  .pulse-cookie
..         Documents    .gvfs           .recently-used.xbel
.bash_history Download        .ICEauthority   report_feb
.bash_logout .esd_auth      .icons          report_jan
.bash_profile .fontconfig    .local          report_mar
.bashrc      .gconf         ls_02.png       Templates
.cache       .gconfd        .mozilla        .themes
chart_feb    .gegl-0.0      Music           .thumbnails
chart_jan    .gimp-2.6      .nautilus       Videos
chart_mar    .gnome2        Pictures        .xsession-errors
.config      .gnome2_private Public           .xsession-errors.old
.dbus        .gstreamer-0.10 .pulse
```

Run the "**ls .***" command to list all hidden files.



A terminal window titled "student@localhost:~" with a menu bar (File, Edit, View, Terminal, Help). The command "[student@localhost ~]\$ ls -d .*" has been executed, displaying a list of hidden files and directories in four columns. The files are: ., .., .bash_history, .bash_logout, .bash_profile, .bashrc, .cache, .config, .dbus, .esd_auth, .fontconfig, .gconf, .gconfd, .gegl-0.0, .gimp-2.6, .gnome, .gnome2_private, .gstreamer-0.10, .gtk-bookmarks, .gvfs, .ICEauthority, .icons, .local, .mozilla, .nautilus, .pulse, .pulse-cookie, .recently-used.xbel, .themes, .thumbnails, .xsession-errors, and .xsession-errors.old. The prompt "[student@localhost ~]\$" is followed by a cursor.

```
[student@localhost ~]$ ls -d .*
.          .dbus      .gnome2_private .nautilus
..         .esd_auth  .gstreamer-0.10 .pulse
.bash_history .fontconfig .gtk-bookmarks  .pulse-cookie
.bash_logout .gconf     .gvfs          .recently-used.xbel
.bash_profile .gconfd    .ICEauthority  .themes
.bashrc      .gegl-0.0 .icons         .thumbnails
.cache       .gimp-2.6 .local         .xsession-errors
.config      .gnome     .mozilla       .xsession-errors.old
[student@localhost ~]$
```


Exercise 5 - Creating and Switching Directories

1. Run the **mkdir** make directory command as shown in the example to create some directories.

Question 1

How do you know you have created directories and not files?

Answer

Run "ls -l" command to display the file information in long format. A letter "d" in the file type field means the file is a directory.

```

student@localhost:~
File Edit View Terminal Help
[student@localhost ~]$ mkdir hr sales
[student@localhost ~]$ ls
chart_feb Documents hr Pictures report_mar
chart_jan Download ls_02.png Public sales
chart_mar hidden_01.png ls_03.png report_feb Templates
Desktop hidden_02.png Music report_jan Videos
[student@localhost ~]$ mkdir hr/report
[student@localhost ~]$ mkdir hr/chart
[student@localhost ~]$ cd sales
[student@localhost sales]$ mkdir report chart
[student@localhost sales]$ cd ..
[student@localhost ~]$ mkdir backup
[student@localhost ~]$ ls
backup Documents ls_02.png report_feb Videos
chart_feb Download ls_03.png report_jan
chart_jan hidden_01.png Music report_mar
chart_mar hidden_02.png Pictures sales
Desktop hr Public Templates
[student@localhost ~]$

```

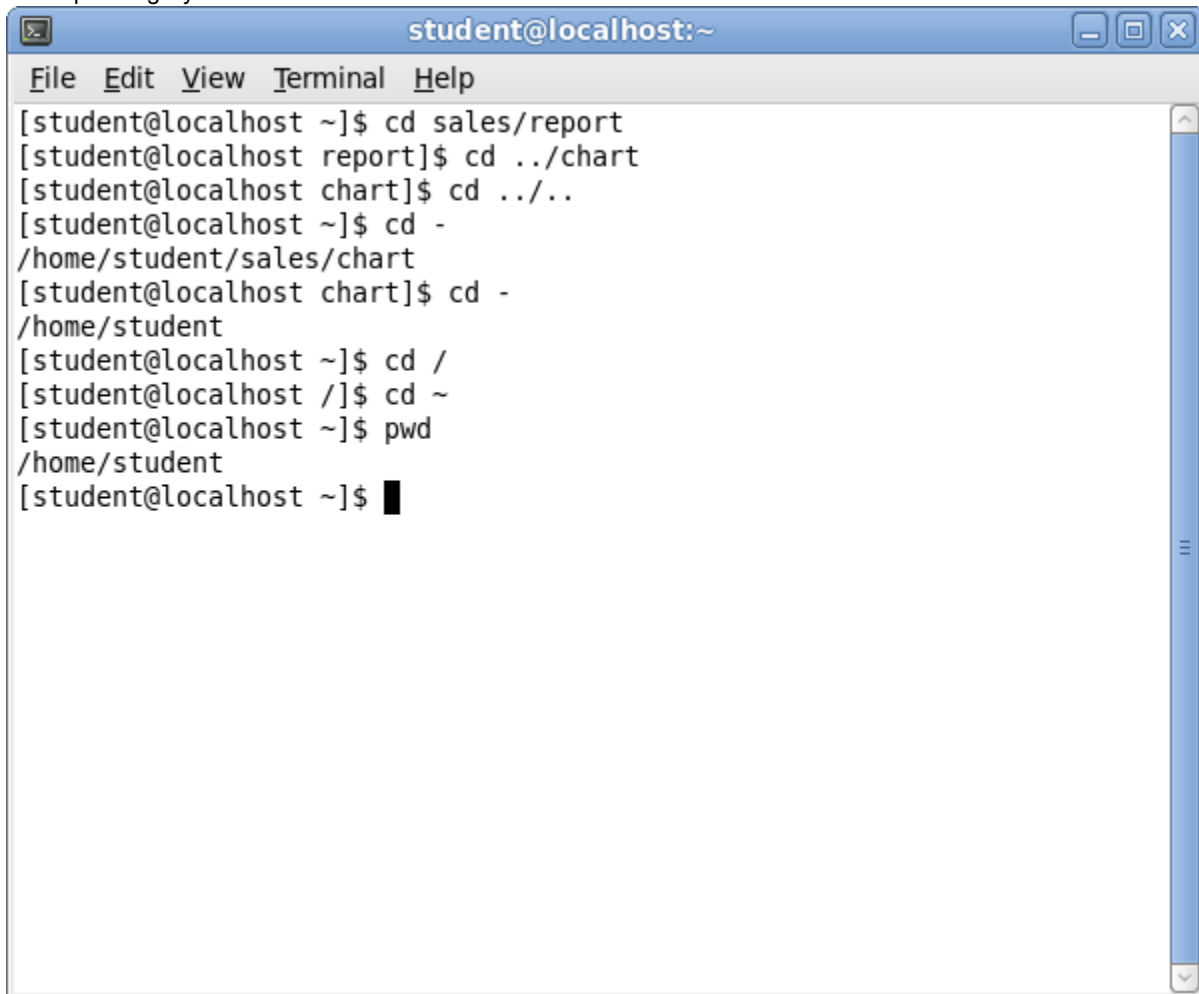
Run the **cd** change directory command as shown in the example to switch between the directories.

Note

The command "**cd ~**" will change directory to your home directory.

The command "**cd /**" will change directory to the root directory.

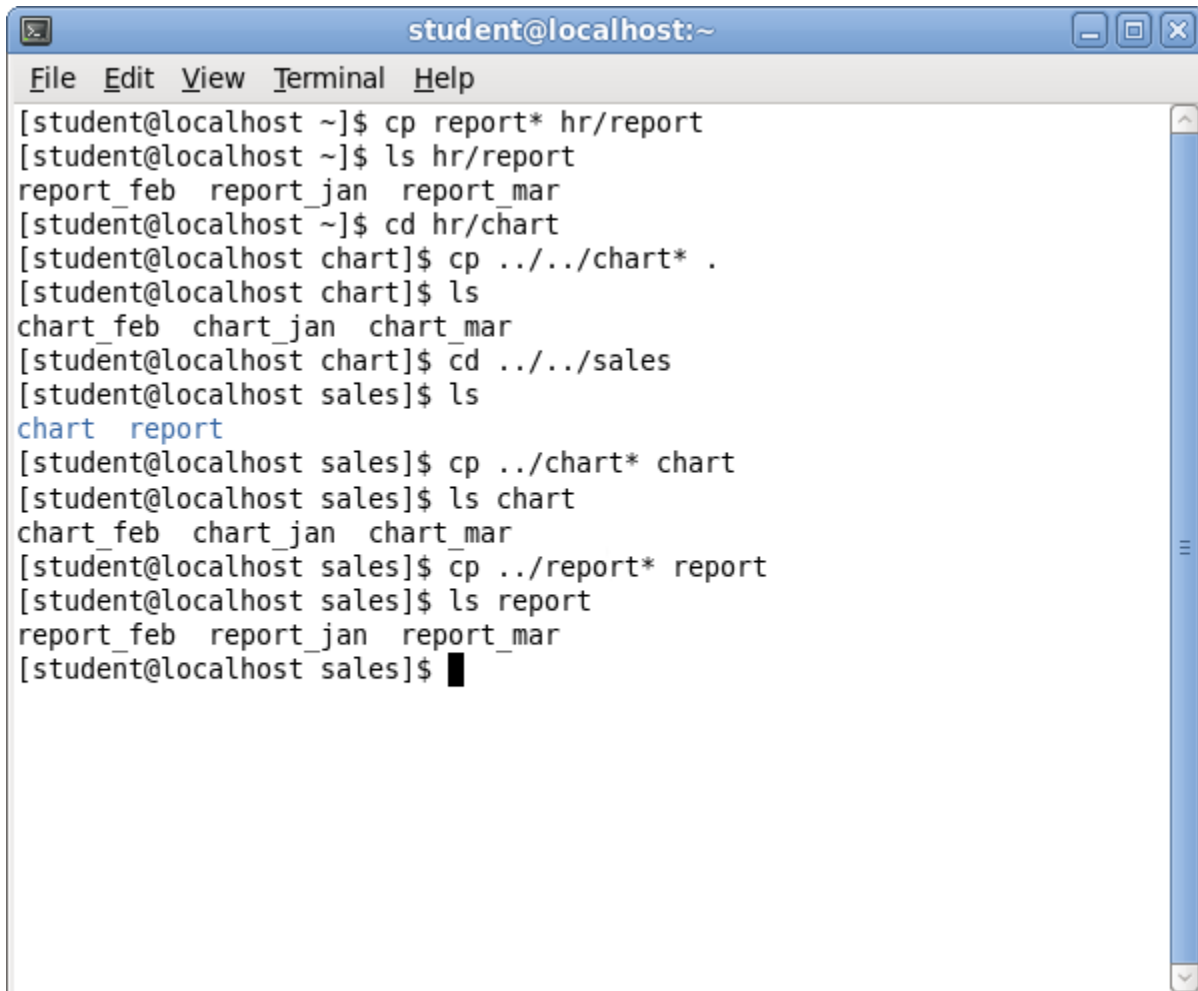
The command "**cd -**" will change directory to the previous working directory. This command is very useful when you need to switch between 2 directories frequently.

A terminal window titled 'student@localhost:~' with a menu bar (File, Edit, View, Terminal, Help) and a scrollbar on the right. The terminal shows a sequence of directory navigation commands and their outputs.

```
student@localhost:~  
File Edit View Terminal Help  
[student@localhost ~]$ cd sales/report  
[student@localhost report]$ cd ../chart  
[student@localhost chart]$ cd ../../  
[student@localhost ~]$ cd -  
/home/student/sales/chart  
[student@localhost chart]$ cd -  
/home/student  
[student@localhost ~]$ cd /  
[student@localhost /]$ cd ~  
[student@localhost ~]$ pwd  
/home/student  
[student@localhost ~]$ █
```

Exercise 6 - Copying Files

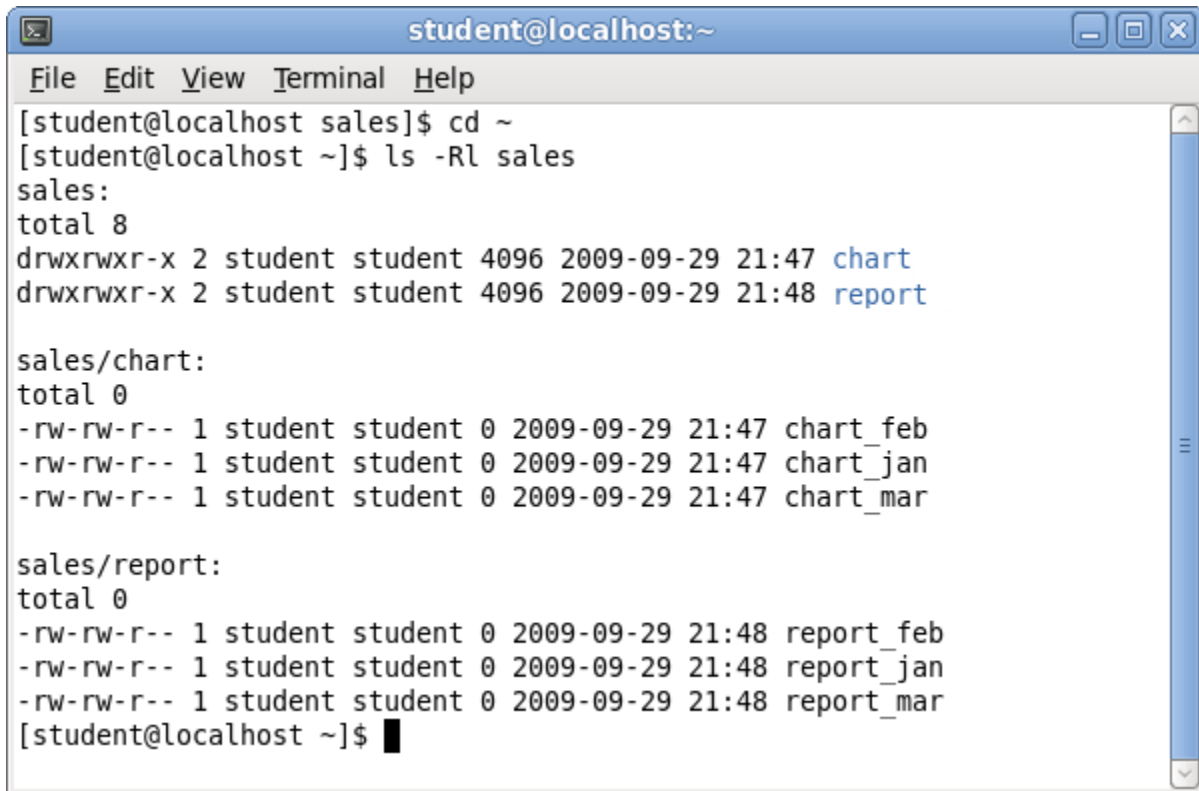
1. Run the **cp** copy command as shown in the example to copy files from your home directory to the directories you have created.



```
student@localhost:~  
File Edit View Terminal Help  
[student@localhost ~]$ cp report* hr/report  
[student@localhost ~]$ ls hr/report  
report_feb  report_jan  report_mar  
[student@localhost ~]$ cd hr/chart  
[student@localhost chart]$ cp ../../chart* .  
[student@localhost chart]$ ls  
chart_feb  chart_jan  chart_mar  
[student@localhost chart]$ cd ../../sales  
[student@localhost sales]$ ls  
chart  report  
[student@localhost sales]$ cp ../chart* chart  
[student@localhost sales]$ ls chart  
chart_feb  chart_jan  chart_mar  
[student@localhost sales]$ cp ../report* report  
[student@localhost sales]$ ls report  
report_feb  report_jan  report_mar  
[student@localhost sales]$
```

Verify that you have copied the files correctly.

You can run the **ls** with the option "-Rl" to list all files including files in the sub-directories.

A terminal window titled "student@localhost:~" with a menu bar (File, Edit, View, Terminal, Help). The terminal shows the following commands and output:

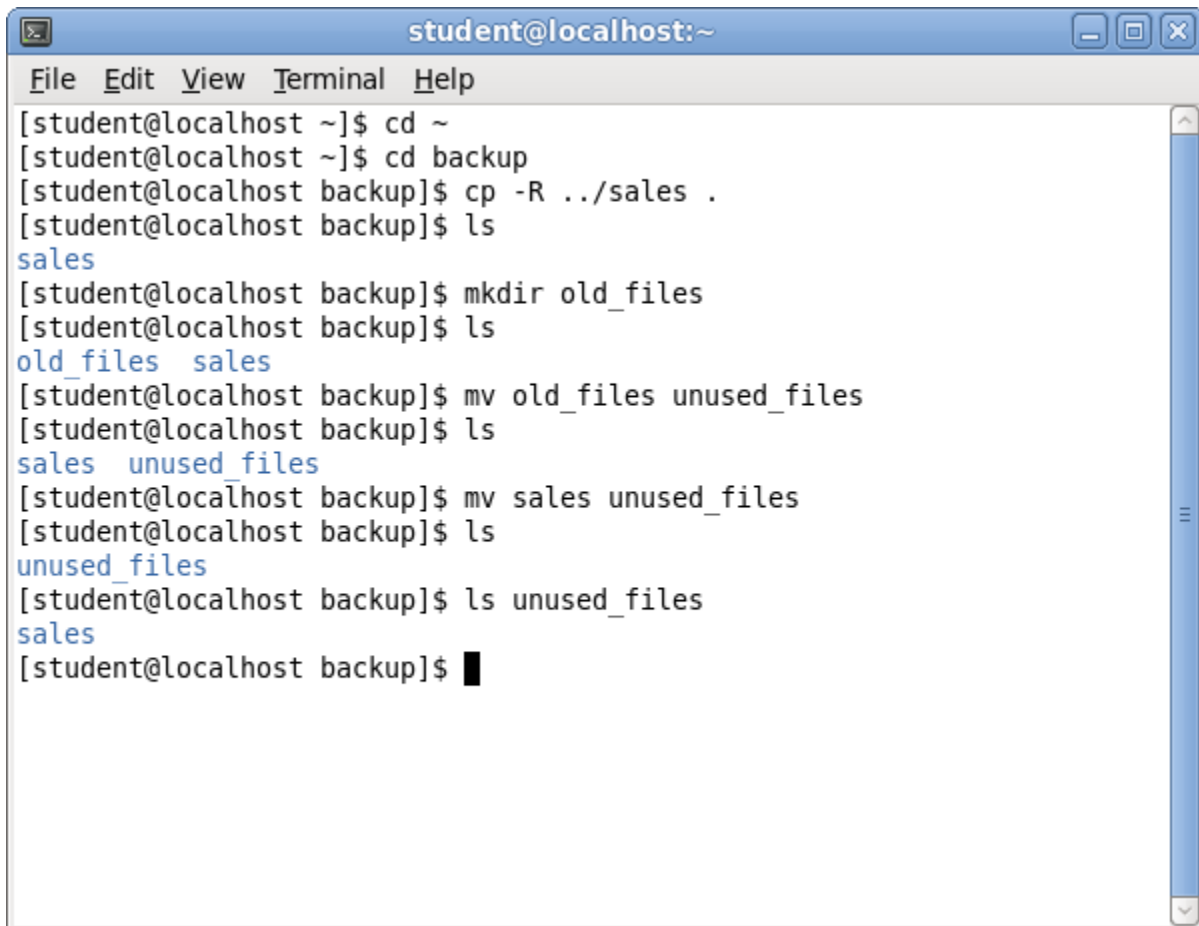
```
[student@localhost sales]$ cd ~  
[student@localhost ~]$ ls -Rl sales  
sales:  
total 8  
drwxrwxr-x 2 student student 4096 2009-09-29 21:47 chart  
drwxrwxr-x 2 student student 4096 2009-09-29 21:48 report  
  
sales/chart:  
total 0  
-rw-rw-r-- 1 student student 0 2009-09-29 21:47 chart_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 21:47 chart_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 21:47 chart_mar  
  
sales/report:  
total 0  
-rw-rw-r-- 1 student student 0 2009-09-29 21:48 report_feb  
-rw-rw-r-- 1 student student 0 2009-09-29 21:48 report_jan  
-rw-rw-r-- 1 student student 0 2009-09-29 21:48 report_mar  
[student@localhost ~]$
```

Exercise 7 - Moving Files

1. Run the **mv** move command as shown in the example to move files from one directory to another directory.

Note

The **mv** move command can be used to rename file.

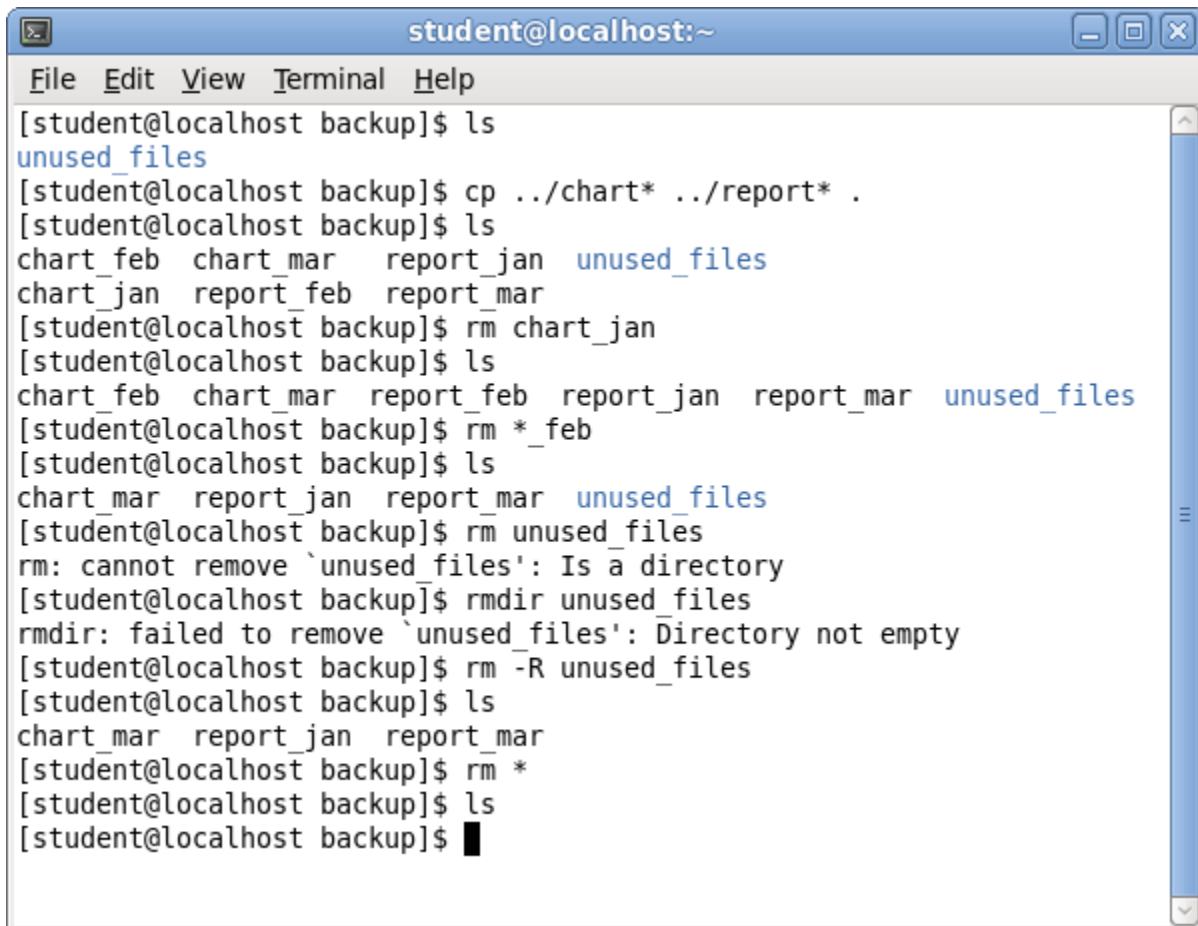
A terminal window titled 'student@localhost:~' with a menu bar (File, Edit, View, Terminal, Help). The terminal shows a sequence of commands: 'cd ~', 'cd backup', 'cp -R ../sales .', 'ls' (output: 'sales'), 'mkdir old_files', 'ls' (output: 'old_files sales'), 'mv old_files unused_files', 'ls' (output: 'sales unused_files'), 'mv sales unused_files', 'ls' (output: 'unused_files'), 'ls unused_files' (output: 'sales'), and finally a prompt '[student@localhost backup]\$' with a cursor.

```
student@localhost:~  
File Edit View Terminal Help  
[student@localhost ~]$ cd ~  
[student@localhost ~]$ cd backup  
[student@localhost backup]$ cp -R ../sales .  
[student@localhost backup]$ ls  
sales  
[student@localhost backup]$ mkdir old_files  
[student@localhost backup]$ ls  
old_files sales  
[student@localhost backup]$ mv old_files unused_files  
[student@localhost backup]$ ls  
sales unused_files  
[student@localhost backup]$ mv sales unused_files  
[student@localhost backup]$ ls  
unused_files  
[student@localhost backup]$ ls unused_files  
sales  
[student@localhost backup]$
```

Exercise 8 - Removing Files and Directories

1. Run the **rm** remove command and **rmdir** remove directory command as shown in the example to remove files and directories.

The option "R" can be used to remove files and directories recursively.



```

student@localhost:~
File Edit View Terminal Help
[student@localhost backup]$ ls
unused_files
[student@localhost backup]$ cp ../chart* ../report* .
[student@localhost backup]$ ls
chart_feb chart_mar report_jan unused_files
chart_jan report_feb report_mar
[student@localhost backup]$ rm chart_jan
[student@localhost backup]$ ls
chart_feb chart_mar report_feb report_jan report_mar unused_files
[student@localhost backup]$ rm *_feb
[student@localhost backup]$ ls
chart_mar report_jan report_mar unused_files
[student@localhost backup]$ rm unused_files
rm: cannot remove `unused_files': Is a directory
[student@localhost backup]$ rmdir unused_files
rmdir: failed to remove `unused_files': Directory not empty
[student@localhost backup]$ rm -R unused_files
[student@localhost backup]$ ls
chart_mar report_jan report_mar
[student@localhost backup]$ rm *
[student@localhost backup]$ ls
[student@localhost backup]$
  
```

Exercise 9 - File Permission

1. Linux defines 3 different types of permissions to a file.

1.	Read
2.	Write
3.	Execute

For each file, permissions (read, write and execute) are defined for 3 different types of users.

1.	User	Owner of the file.
2.	Group	Represents all users belonging to a the group
3.	Others	Represents all users that are not a member of the group and not the owner of the file.

Run the **ls** command as shown to display the file permissions.

```

student@localhost:~
File Edit View Terminal Help
[student@localhost ~]$ ls -ld /bin
drwxr-xr-x 2 root root 4096 2009-09-15 11:01 /bin
[student@localhost ~]$ ls -al /bin/ls
-rwxr-xr-x 1 root root 115932 2009-03-02 21:57 /bin/ls
[student@localhost ~]$

```

	user	group	other
d	r w x	r w x	r w x

Character	File Type
d	Directory
-	File
l	Link

Character	Numeric	Binary	Permission
		r w x	
r	4	1 0 0	Read
w	2	0 1 0	Write
x	1	0 0 1	Execute
-	0	0 0 0	No permission

2. Verify the following permissions:

```

644 = rw-r--r--
755 = rwxr-xrwx
000 = -----
711 = rwx--x--x
700 = rwx-----
777 = rwxrwxrwx
111 = --x--x--x
600 = rw-----
731 = rwx-wx--x

```

3. Permissions can be set using the numeric representation or the symbolic representation.
4. Run the **chmod** command as shown to set the file permission using numeric representation.

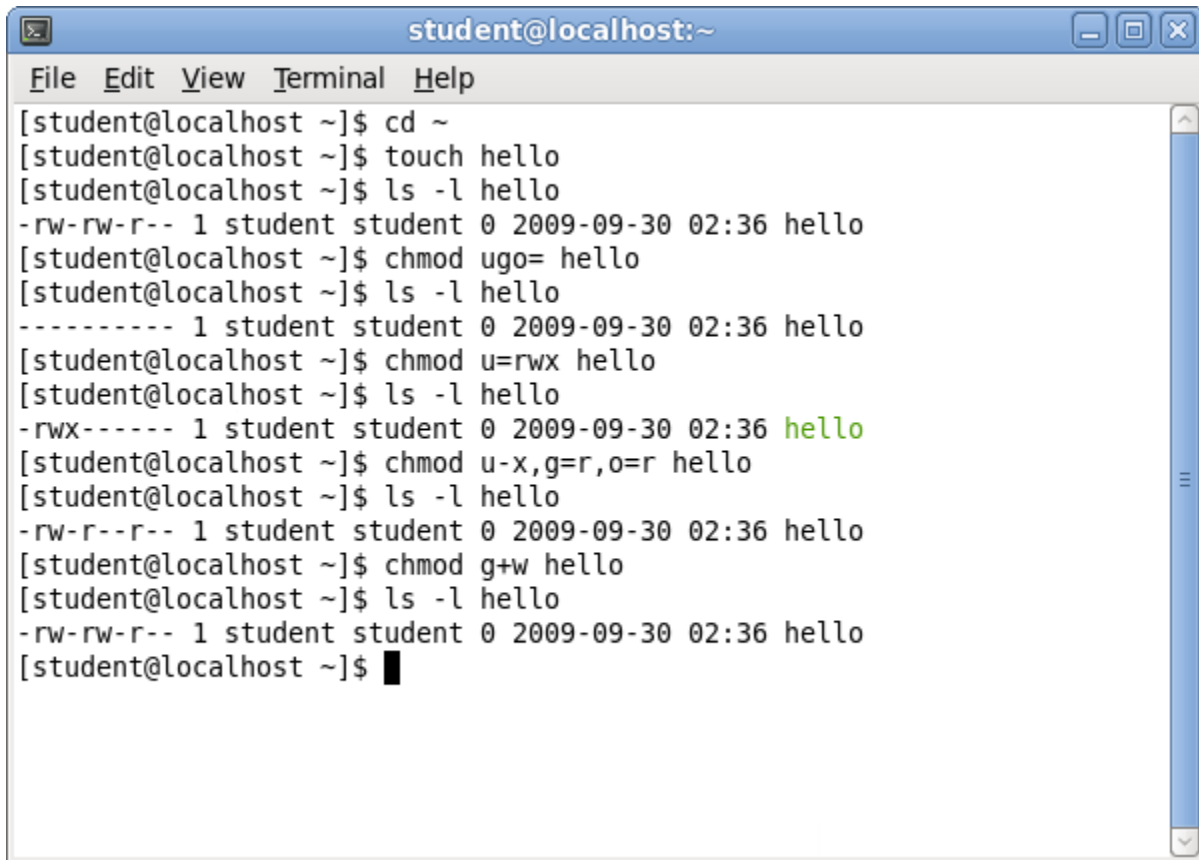
```

student@localhost:~
File Edit View Terminal Help
[student@localhost backup]$ cd ~
[student@localhost ~]$ touch hello
[student@localhost ~]$ ls -l hello
-rw-rw-r-- 1 student student 0 2009-09-30 02:24 hello
[student@localhost ~]$ chmod 000 hello
[student@localhost ~]$ ls -l hello
----- 1 student student 0 2009-09-30 02:24 hello
[student@localhost ~]$ chmod 700 hello
[student@localhost ~]$ ls -l hello
-rwx----- 1 student student 0 2009-09-30 02:24 hello
[student@localhost ~]$ chmod 644 hello
[student@localhost ~]$ ls -l hello
-rw-r--r-- 1 student student 0 2009-09-30 02:24 hello
[student@localhost ~]$ chmod 664 hello
[student@localhost ~]$ ls -l hello
-rw-rw-r-- 1 student student 0 2009-09-30 02:24 hello
[student@localhost ~]$

```

Symbol	Description
u	User (owner of the file)
g	Group
o	Others

Operator	Description
+	Add a permission
-	Remove a permission
=	Assign a permission



```
student@localhost:~  
File Edit View Terminal Help  
[student@localhost ~]$ cd ~  
[student@localhost ~]$ touch hello  
[student@localhost ~]$ ls -l hello  
-rw-rw-r-- 1 student student 0 2009-09-30 02:36 hello  
[student@localhost ~]$ chmod ugo= hello  
[student@localhost ~]$ ls -l hello  
----- 1 student student 0 2009-09-30 02:36 hello  
[student@localhost ~]$ chmod u=rwx hello  
[student@localhost ~]$ ls -l hello  
-rwx----- 1 student student 0 2009-09-30 02:36 hello  
[student@localhost ~]$ chmod u-x,g=r,o=r hello  
[student@localhost ~]$ ls -l hello  
-rw-r--r-- 1 student student 0 2009-09-30 02:36 hello  
[student@localhost ~]$ chmod g+w hello  
[student@localhost ~]$ ls -l hello  
-rw-rw-r-- 1 student student 0 2009-09-30 02:36 hello  
[student@localhost ~]$
```

Exercise 10 - Links

1.

Note

A link is a file that point to another file.
There are 2 types of links - hard link and symbolic links.

Run the **echo** command as shown to create a file named "apple.txt" containing the line "apple".

Run the "**ls -l** apple.txt" command to list the created file.

Run the "**ln** apple.txt orange.txt" command to create a hard link "orange.txt" that points to "apple.txt".

Run the "**ls -l** *.txt" command to list all files with filename ending with ".txt".

Run the "**cat** apple.txt" command to display the content of the file.

Run the "**cat** orange.txt" command to display the content of the file.

```

student@localhost:~
File Edit View Terminal Help
[student@localhost ~]$ cd ~
[student@localhost ~]$ echo "apple" > apple.txt
[student@localhost ~]$ ls -l apple.txt
-rw-rw-r-- 1 student student 6 2009-09-30 02:57 apple.txt
[student@localhost ~]$ ln apple.txt orange.txt
[student@localhost ~]$ ls -l *.txt
-rw-rw-r-- 2 student student 6 2009-09-30 02:57 apple.txt
-rw-rw-r-- 2 student student 6 2009-09-30 02:57 orange.txt
[student@localhost ~]$ cat apple.txt
apple
[student@localhost ~]$ cat orange.txt
apple
[student@localhost ~]$

```

Question 2

Can you tell the difference between "apple.txt" and "orange.txt"?

Why both the files have the same content?

If you change the content of "apple.txt" will the content of "orange.txt" be changed?

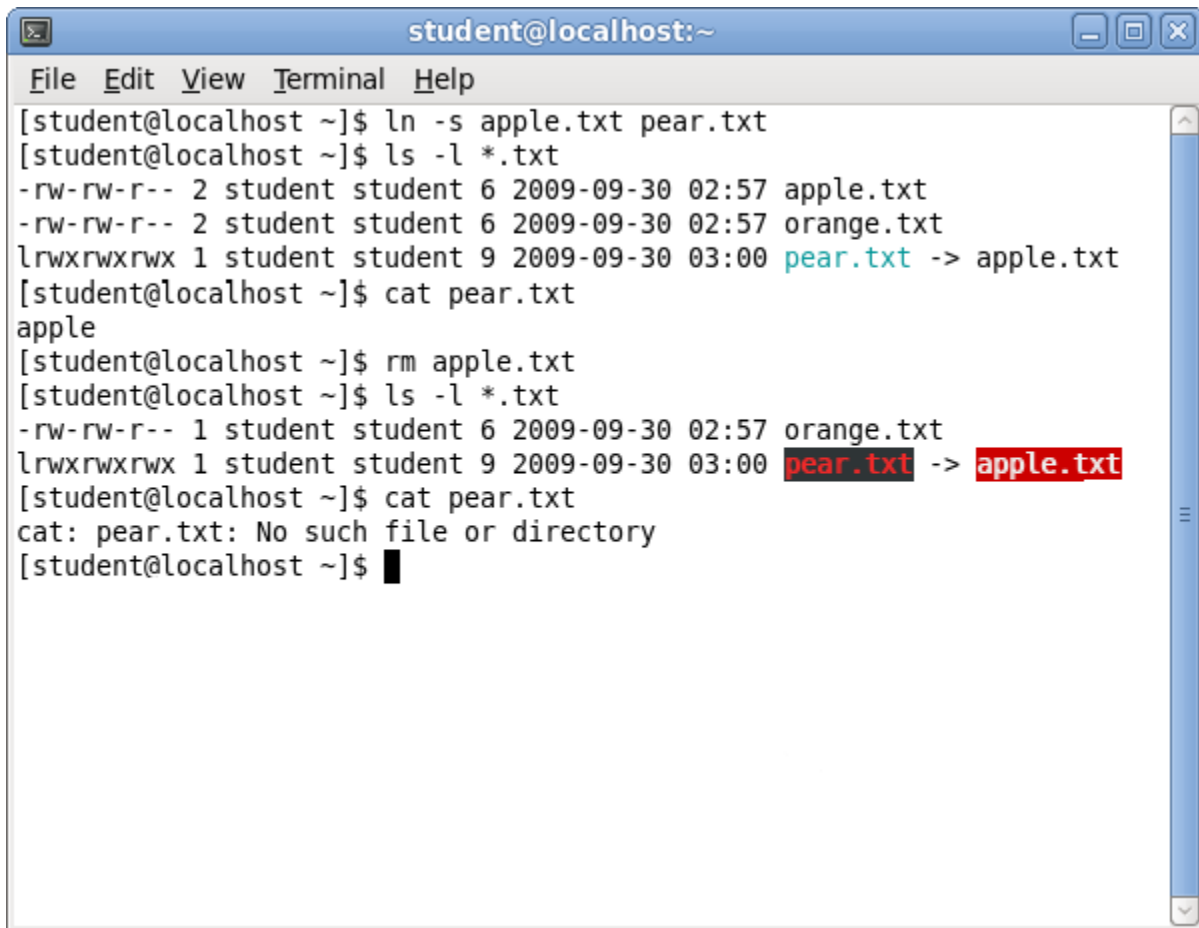
There is a number in second column of the **ls** output. What is the meaning of that number?

Answer

Physically there is only one file. Both "apple.txt" and "orange.txt" refer to the same file. Any changes to the content of the file are viewable by from both "apple.txt" and "orange.txt". The number is the link count.

Run the "**ln -s apple.txt pear.txt**" command to create a symbolic link "pear.txt". The "-s" option is used to create symbolic links.

Follow the example to run the rest of the commands.

A terminal window titled "student@localhost:~" with a menu bar (File, Edit, View, Terminal, Help). The terminal shows the following commands and output:

```
[student@localhost ~]$ ln -s apple.txt pear.txt
[student@localhost ~]$ ls -l *.txt
-rw-rw-r-- 2 student student 6 2009-09-30 02:57 apple.txt
-rw-rw-r-- 2 student student 6 2009-09-30 02:57 orange.txt
lrwxrwxrwx 1 student student 9 2009-09-30 03:00 pear.txt -> apple.txt
[student@localhost ~]$ cat pear.txt
apple
[student@localhost ~]$ rm apple.txt
[student@localhost ~]$ ls -l *.txt
-rw-rw-r-- 1 student student 6 2009-09-30 02:57 orange.txt
lrwxrwxrwx 1 student student 9 2009-09-30 03:00 pear.txt -> apple.txt
[student@localhost ~]$ cat pear.txt
cat: pear.txt: No such file or directory
[student@localhost ~]$
```

Question 3

What differences between hard links and symbolic links can you discover from the **ls** command output?

Answer

Unlike hard links, the link count for symbolic links is one.

The name of the file in which the symbolic links is pointing to is displayed.

Symbolic links can point to a file that does not exist.