Basic Linux Commands (Part 1)

Due date

- End of Week 3 lab class

Evaluation

- 3% of final grade.

Submission

Hand in or email (in email subject line: CST8102-14F lab2 submission) completed lab before due date.

Materials

- Student laptop computer
- Ubuntu 14.04.1 installed in VMWare Workstation

Procedure

Exercise #1: Command ***pwd***

Read the man pages for pwd.

1) **`man pwd`**

   Read the manual pages of **pwd** command

   What is the purpose/output of **pwd** command?

   Print current working directory

2) Press **q** to quit the manual pages of **pwd**.

Exercise #2: command ***cd***

Type the following command and press **Enter**

1) **cd**
   - this brings you into your home directory

   Record the bash prompt: `user@localhost :____~_____$`

2) **pwd**

   Record the output of that command: _____/home/user_____

3) **cd ~**
   - this brings you into your home directory

   Record the bash prompt: `user@localhost :_____~_____$`

4) **pwd**

   Record the output of that command: _____/home/user_____

5) **cd /etc**

   Record the bash prompt: `user@localhost :___/etc_____$`

6) **pwd**

   Record the output of that command: _____/etc_____

7) **cd ..**
   - this brings you one level up, in this case **etc**'s parent directory, which is root directory

   Record the bash prompt: `user@localhost :____/_____$`

8) **pwd**

Record the output of that command: _____/_____

9) **cd home/*user*** *(use your actual username instead of "user)*

Record the bash prompt: user@localhost :_____~_____$
Note that we are using the relative path.
What would the command line look like if we were to use the absolute path?

*__cd /home/user*_____

10) **pwd**

Record the output of that command: ____/home/user_____

11) **cd /usr/local/bin/**

Record the bash prompt: user@localhost :__/usr/local/bin_____$

12) **pwd**

Record the output of that command: __/usr/local/bin_____

13) **cd ../../sbin**

Record the bash prompt: user@localhost :_/usr/sbin_____$

14) **pwd**

Record the output of that command: ____/usr/sbin_____

15) **cd /**

Record the bash prompt: user@localhost :_____/_____$

16) **pwd**

Record the output of that command: _____/_____

17) **cd bin**

Record the bash prompt: user@localhost :_____/bin_____$
Note that we are using the relative path. What would the command line look like if we were to use the absolute path?

____*cd /bin*_____

18) **pwd**

Record the output of that command: _____/bin_____

Exercise #3: command ***ls***

1) **ls /bin/ls**
2) **ls /home/*user***
3) **ls -a /home/*user***
4) **ls -al /home/*user***
5) **ls /ho**, then press the **[Tab]** key – the shell will fill in the rest of the file name for you.

   Press the 'up arrow' key twice. You will notice that previously typed in commands can be recalled by using the arrow keys.

Exercise #4: command ***more***

Follow the steps outlined below:

1) **cd /etc** - to go into the /etc directory (lots of files in here!)
2) **ls -la**
3) **ls -al | more** - to view the contents one screen at a time

   The piping capacity of Linux using the | symbol (**Shift-\)**

   Use the [spacebar] to jump to the next screen of information
   You can use **q** to abort the command

4) **cd /home**

Exercise #5: command ***mkdir***

1) user@localhost :/home$ **cd**

   - What is the purpose of the cd command without arguments?

   ____go to the user's home directory_____

2) user@localhost :~$ **mkdir cst8102 ; cd cst8102**

3) user@localhost :~/cst8102$ **mkdir labs tests**

4) user@localhost :~/cst8102$ **ls**

- What is the output of the above command?

___labs tests_____

5) user@localhost :~/cst8102$ **mkdir labs/lab1 tests/test1**
6) user@localhost :~/cst8102$ **ls labs tests**

- What is the output of the above command?

labs:
1ab1

tests:
test1

---

7) user@localhost :~/cst8102$ **mkdir lectures/lecture1**

- Record the error message:

  mkdir: cannot create directory ‘lectures/lecture1’: No such file or directory

- Explain why this command did not execute successfully:

  The parent directory lectures dose not exist.

8) user@localhost :~/cst8102$ **mkdir -p lectures/lecture1**

- Did the command execute successfully?

____Yes_____

Exercise #6: command ***rmdir***

1) user@localhost :~/cst8102$ **ls -l**

- What is the output of that command? (Give a description)

Long format list of all files and directories that are available in current directory

2) user@localhost :~/cst8102$ **rmdir tests**
3) user@localhost :~/cst8102$ **ls -l**

- Record the error message:

  rmdir: failed to remove ‘tests’: Directory not empty

```
4) user@localhost :~/cst8102$ cd  tests
5) user@localhost :~/cst8102/tests$ rmdir  test1

6) user@localhost :~/cst8102/tests$ cd ..
7) user@localhost :~/cst8102$ rmdir tests
```

- Does the command produce an error message?

  __No._____

```
8) user@localhost :~/cst8102$ rmdir  lectures/lecture1
9) user@localhost :~/cst8102$ rmdir  lectures
10)user@localhost :~/cst8102$ ls
```

- Is **lectures** removed?

____Yes_____

Review exercise

Enter the commands below in your home directory.

```
1.  mkdir ~/lab2
2.  cd lab2
3.  mkdir linux  ./windows  ./unix
4.  mkdir linux/ubuntu  linux/fedora  unix/bsd
5.  rmdir linux
6.  rmdir windows
7.  mkdir -p ~/lab2/linux/android/kitkat
8.  rmdir -p ~/lab2/unix/bsd
9.  cd adroid
10. cd linux/android/kitkat
11. cd ../../
12. pwd
```

Answer these questions based **only** on the above 12 commands:

1) How many directories have you successfully deleted?

3

List them using absolute path:

```
~/lab2/windows
~/lab2/unix
~/lab2/unix/bsd
```

2) How many directories in total have you created? (Including deleted directories)

9

List them by names:

```
lab2, linux, windows, unix, ubuntu, fedora, bsd,
android, kitkat
```

3) How many directories are left in the directory **lab2**?

1

List them using relative paths: (current directory is user's home directory)

```
Lab2/linux
```

4) How many error messages have you encountered?

3

Record the error message along with the command number (1-12):

5. rmdir: failed to remove 'linux': Directory not empty
8. rmdir: failed to remove directory '/home/user1/lab2': Directory not empty
9. bash: cd: adroid: No such file or directory

5) Record the output of the command **`pwd`** :

```
/home/user/lab2/linux
```

Basic Linux Commands (Part 2)

Due date

- End of Week 5 lab class

Evaluation

- 3% of final grade.

Submission

Hand in or email (in email subject line: CST8102-14F lab3 submission) completed lab before due date.

Materials

- Student laptop computer
- Ubuntu 14.04.1 installed in VMWare Workstation

Procedure

Command touch

The **touch** command updates different time stamps. As a side benefit it is used to create empty files.

Exercise #1: Creating empty files & updating the modification time

1) user@localhost :~$ **`touch clock`**
2) user@localhost :~$ **`ls -l clock`**
    - Record the time stamp:

      _____
3) user@localhost :~$ **`sleep 61`**

   Wait for one minute.

4) user@localhost :~$ **`touch clock`**
5) user@localhost :~$ **`ls -l clock`**
    - Record the time stamp:

      _____

Command: cp

The **cp** command makes a copy of an existing set of files or directories into another area of the system.
The syntax for the cp command is:

- **cp** ***[-r] source destination***

Exercise #2: Copying files to a directory

1) user@localhost :~$ **`mkdir lab3ex`**
2) user@localhost :~$ **`cd lab3ex`**
3) user@localhost :~/lab3ex$ **`touch f1 f2 f3`**
4) user@localhost :~/lab3ex$ **`ls`**
    - What is the output of that command?

      f1 f2 f3

5) user@localhost :~/lab3ex$ **`mkdir lab3`**

6) user@localhost :~/lab3ex$ **`ls`**

- What is the output of that command?

  f1 f2 f3 lab3

7) user@localhost :~/lab3ex$ **`cp f1 f2 f3 lab3`**

8) user@localhost :~/lab3ex$ **`ls lab3`**

- What is the output of that command?

  f1 f2 f3

9) user@localhost :~/lab3ex$ **`mkdir coffee`**

10) user@localhost :~/lab3ex$ **`cd coffee`**

11) user@localhost :~/lab3ex/coffee$ **`touch cream sugar`**

12) user@localhost :~/lab3ex/coffee$ **`cd ..`**

13) user@localhost :~/lab3ex$ **`cp coffee/cream coffee/sugar lab3`**

14) user@localhost :~/lab3ex$ **`ls lab3`**

- What is the output of the command?

  cream f1 f2 f3 sugar

Exercise #3: Copying directories to a directory (-r option)

1) user@localhost :~/lab3ex$ **`mkdir dir1 dir2 dir3`**

- Record the command that you use to verify that the directories have been created? _____**ls**_____

2) user@localhost :~/lab3ex$ **`cp dir1 dir2 dir3 lab3`**

- Record one of the messages displayed on the screen:

  cp: omitting directory 'dir1'

3) user@localhost :~/lab3ex$ **`ls lab3`**

- Have the directories been copied? _____**No**_____

4) user@localhost :~/lab3ex$ **`cp -r dir1 dir2 dir3 lab3`**

5) user@localhost :~/lab3ex$ **`ls lab3`**

- Have the directories been copied? _____**Yes**_____

6) user@localhost:~/lab3ex$ **`sudo apt-get install tree`**

- *(hint: the above command installs "tree" command which is not included in the default Ubuntu installation)*

7) user@localhost:~/lab3ex$ **`tree`**

Exercise #4: Copying directories to a directory , cont'd (-r & --parents option)

1) user@localhost :~/lab3ex$ **`mkdir -p parent/child`**

2) user@localhost :~/lab3ex$ **`cd parent ; touch f1 ; cd ..`**

3) user@localhost :~/lab3ex$ **`cp -r --parents parent/child lab3`**

4) user@localhost :~/lab3ex$ **`tree lab3`**

What is the output of the command?

```
lab3
├── cream
├── dir1
├── dir2
├── dir3
├── f1
├── f2
├── f3
├── parent
│   └── child
└── sugar
```

Command: mv

The **mv**, for Move File or Directory, command moves files and directories to a different directory. It is also used to rename files within the same directory.
The syntax for the mv command is:

- **mv** ***source destination***

Exercise #5: Renaming files

1) user@localhost :~/lab3ex$ **`cd lab3`**
2) user@localhost :~/lab3ex/lab3$ **`mv f1 m1`**
3) user@localhost :~/lab3ex/lab3$ **`ls`**
    - Has the file been renamed from **f1** to **m1**?

      Yes

Exercise #6: Moving files

1) user@localhost :~/lab3ex/lab3$ **`touch red green blue`**
2) user@localhost :~/lab3ex/lab3$ **`mkdir colors`**
3) user@localhost :~/lab3ex/lab3$ **`mv red green blue`**

4) Record the error message:

   mv: target 'blue' is not a directory

5) user@localhost :~/lab3ex/lab3$ **`mv red green blue colors`**

6) user@localhost :~/lab3ex/lab3$ **`ls`**
    - Are the files red, green and blue still in the current directory?

      No

7) user@localhost :~/lab3ex/lab3$ **`ls colors`**
    - Have the files been moved?

_____Yes_____

Exercise #7: Moving directories

1) user@localhost :~/lab3ex/lab3$ **`mkdir toddlers children sandbox`**
2) user@localhost :~/lab3ex/lab3$ **`mv toddlers children sandbox`**
    - Are the toddlers and children in the **sandbox**?

      _____Yes_____

3) user@localhost :~/lab3ex/lab3$ **`cd ..`**

Command: rm

The **rm** or Remove Files or Directory command allows you to delete the content of any directory. It is both a dangerous and useful command because of its flexibility. Unlike DOS and Windows, a file that is deleted in Linux is gone.
The syntax for the rm command is:

- **rm** ***file_list***

Exercise #8: Deleting files

1) user@localhost :~/lab3ex$ **`cd lab3/sandbox`**
2) user@localhost :~/lab3ex/lab3/sandbox$ **`touch child1 child2 child3`**
3) user@localhost :~/lab3ex/lab3/sandbox$ **`ls`**
    - Are child1,child2 and child3 in the sandbox? **_____Yes_____**
4) user@localhost :~/lab3ex/lab3/sandbox$ **`rm child1 child2 child3`**
5) user@localhost :~/lab3ex/lab3/sandbox$ **`ls`**
    - Are child1,child2 and child3 still in the sandbox? **____No_____**

6) user@localhost :~/lab3ex/lab3/sandbox$ **cd ..**

7) user@localhost :~/lab3ex/lab3$ **rmdir sandbox**

- Record the error message

  rmdir: failed to remove 'sandbox': Directory not empty

8) user@localhost :~/lab3ex/lab3$ **cd ..**

Exercise #9: Deleting directories

1) user@localhost :~/lab3ex$ **rmdir lab3**

- Record the error message:

  rmdir: failed to remove 'lab3': Directory not empty

2) user@localhost :~/lab3ex$ **rm -r lab3**

- Has the directory been deleted?

  yes

Command: cat

Cat is an utility to view, create, or append to small files.

Exercise #10: Viewing files with cat

1) user@localhost :~/lab3ex$ **cd ; cat /etc/issue**

2) user@localhost :~$ **cat /etc/fstab**

3) user@localhost :~$ **cat /etc/issue /etc/fstab**

4) user@localhost :~$ **cat .bashrc | more**

Exercise #11: Clear screen with command clear

- user@localhost :~$ **`clear`**

Output redirection

Exercise #12: Redirect output to a file

- user@localhost:~$ **`ls –al ~/ > lsout`**
- user@localhost:~$ **`cat lsout`**
- user@localhost:~$ **`ls –al / > lsout`**
- user@localhost:~$ **`cat lsout`**

  Is "**lsout**" overwritten? _____**Yes**_____

- user@localhost:~/ $ **`ls –a /etc >> lsout`**
- user@localhost:~/lab6$ **`cat lsout | more`**

  Is "**lsout**" overwritten? _____**No**_____

Exercise #13: Not to overwrite a file

- user@localhost:~$ **`set –C`**

- user@localhost:~$ **`ls /home > lsout`**

  Record the message: **bash: lsout: cannot overwrite existing file**

Review exercise

Assume that the commands listed below are executed in the **user's home directory**.

1. **`cd ; mkdir –p ~/lab3rv/linux`**

2. `cd lab3rv/linux`
3. `touch ubuntu fedora suse`
4. `cp ubuntu fedora ../`
5. `cp suse ../mint`
6. `mv fedora ~/lab3rv/redhat`
7. `mv ubuntu arch`
8. `mkdir –p ~/lab3rv/unix ; cd ~/lab3rv/unix`
9. `touch bsd solaris`
10. `cp ~/lab3rv/unix/solaris ~/lab3rv`
11. `cd ..`
12. `rm –r ~/lab3rv/unix`

Answer the following questions after executing the 12 commands above:

1. How many **directories** are created during the review exercise? **3**

   List them using absolute paths:

   ~/lab3rv
   ~/lab3rv/linux
   ~/lab3rv/unix

2. How many **regular files** remain in the directory ~/**lab3rv**? **5**
   (Do not include files in sub-directories).
   List them using absolute paths:
   ~/lab3rv/fedora
   ~/lab3rv/mint
   ~/lab3rv/redhat
   ~/lab3rv/solaris
   ~/lab3rv/ubuntu

3. How many **regular files** are left in the directory **~/lab3/linux**? **2**

List them using relative paths (Assume the current directory is the user's home directory):

`lab3rv/linux/arch`

`lab3rv/linux/suse`

4. What is the **current** directory at the end of the review exercise?

   `~/lab3rv`

5. How many directories are deleted successfully? **1**

   List them using absolute paths:

   `~/lab3rv/unix`

Linux File Permissions

Due date

- End of Week 6 lab class

Evaluation

- 3% of final grade.

Submission

Hand in or email (in email subject line: CST8102-14F lab4 submission) completed lab before due date.

Materials

- Student laptop computer
- Ubuntu 14.04.1 installed in VMWare Workstation

Procedure

Exercise #1: Testing permissions

While logged in as a regular user, use the following command to create a directory named **top** in user's home directory:

- **`mkdir -p /home/`*`user`*`/lab4/top`**

Following the instructions below to complete **Table #1.**

a) Change the permission of the **top** directory using the **chmod** command. The exact command is given in the second column of the table.

b) Execute the commands listed in the first row (starting with the third column) for that permission level. For each command line document whether the command line executes successfully or not: Use **PD** for Permission Denied, **OK** for success, **NF** for "No such file or directory"

The commands are:

- `ls -l top`
- `mkdir top/sub`
- `rmdir top/sub`
- `cd top`
- **cd .. (execute this ONLY if your current directory is top!)**

c) Follow the above procedure for each row of the table (row 1 to 8).

Note: Before you run each **chmod** command in the table below, make sure your current directory is **~/lab4**.

Table #1: ***Testing directory permissions***

Row #	Command line to modify permissions	ls -l top	mkdir top/sub	rmdir top/sub	cd top
1	chmod u+r-w+x top	OK	PD	NF	OK
2	chmod u-r+wx top	PD	OK	OK	OK
3	chmod u+rw-x top	OK	PD	PD	PD
4	chmod u-rw+x top	PD	PD	NF	OK
5	chmod u-r+w-x top	PD	PD	PD	PD
6	chmod u+r-wx top	OK	PD	PD	PD
7	chmod u-rwx top	PD	PD	PD	PD
8	chmod u+rwx top	OK	OK	OK	OK

Default permissions

Exercise #2: Viewing a user's default permissions

Login as a regular **user**.

1) Type **umask** and record the output of the command: _____**0002**_____

- Based on the **umask**, what are the default permissions for directories and files in octal mode, based on your **umask**:
  directory: __775_____ file: ____664_____

2) Verify it by creating a new file with the **touch** command.

   - Record the default permissions set on the file in symbolic mode:

     _____rw-rw-r--_____

   - What is the default permissions set on the file in octal mode:

     _____664_____

3) Verify it by creating a new directory with the **mkdir** command.

   - Record the default permissions set on the directory in symbolic mode:

     ____rwxrwxr-x_____

   - What is the default permissions set on the directory in octal mode:

     ____775_____

Exercise #3: Changing default permissions

1) Set the umask to 044, record the command you use __umask 044_____

2) Type **umask** and record the output of the command: ___044_____

   - Based on the umask, what are the default permissions for directories and files in octal mode, based on your umask:
     directory: __733_____ file: __622_____

3) Verify it by creating a new file.

   - Record the default permissions set on the file in symbolic mode:

     ___rw--w--w-_____

   - What is the default permissions set on the file in octal mode:

     ___622_____

4) Verify it by creating a new directory.

- Record the default permissions set on the directory in symbolic mode:

  ___**rwx-wx-wx**_____

- What is the default permissions set on the directory in octal mode:

  ___**733**_____

Ownership

Exercise #4: Creating new users

Create the two user accounts with the following commands:

1) **su – root**
2) **useradd -d /home/user1 user1 -m**
3) **useradd -d /home/user2 user2 -m**
4) **passwd user1**
   - type in a password when prompted. If you do not type the username after the passwd command, you are changing the root password!
5) **passwd user2**

Exercise #5: Creating shared directory

1) **mkdir /shared**

- Who is the owner of the **/shared** directory? ___**root**_____
- What is the group name of the **/shared** directory? ___**root**_____

2) Give full access permissions to **/shared** for everybody

- Record the command you use: _____**chmod 777 /shared**_____

Exercise #6: Making changes from user1

1) **su – user1**

2) Has the prompt changed to *"$"?*____***yes***_____

3) **`cd /shared`**

4) **`cat > plan`**

Hint: Input "this is a test" at the blinking cursor. Press **ctrl+d** *when you are done.*

5) Who is the owner of that file? _____**user1**_____

6) What is the group name of that file? ___**user1**_____

7) **`chmod o-rwx /shared/plan`**

- Make sure that others have no access permissions. Verify with **`ls -l`** that you achieved the desired result.

Exercise #7: Making changes from user2

1) Login as **user2** and try to modify the file using the following commands:

```
su – user2

cat >> /shared/plan
```

2) Record the message: _____**Permissin dennied**_____

Why? _____ **Others have no access permissions to the file**_____

Exercise #8: Changing file ownership

1) Login as root and change the ownership of **plan** to **user2** using the following commands:

```
su - root
chown user2.user2 /shared/plan
```

2) Verify that **user2** is the owner of plan with command: **`ls –l /shared/plan`**

3) Login as **user2** and try to modify the **/shared/plan**. Can you do it? __**Yes**_____

4) Login as **user1** and try to modify the **/shared/plan**. Can you do it? __**No**____

5) While you are logged in as **user2,** try to delete the file. Can you do it (eventually)? ___**Yes**_____

Exercise #9: Minimum Permissions

Circle the minimum permissions required to successfully complete the actions listed below. (*hint: use lecture note #3 as reference*)

To copy a file the user requires

- for the source directory: R W X
- for the target directory: R W X
- for the file: R W X

- To move a file the user requires
  - for the source directory: R W X
  - for the target directory: R W X
  - for the file: R W X

- To delete a file the user requires
  - for the directory: R W X
  - for the file: R W X

The Linux File System (Part 1)

Due date

- End of Week 7 lab class

Evaluation

- 3% of final grade.

Hand in or email (in email subject line: CST8102-14F lab5 submission) completed lab before due date.

Materials

- Student laptop computer
- Ubuntu 14.04.1 installed in VMWare Workstation

Before you get started...

IMPORTANT – Create a full clone of your Ubuntu virtual machine in case you really mess things up. You should already have the clone that was created in lab1. In case you don't have it in place, create it again following the instructions in lab1.

Procedure

Exercise #1: Viewing existing partitions

To manage partitions, use the **/sbin/fdisk** command.

The syntax of the **fdisk** command is: **`fdisk device_name`**

- Create a directory called lab and make it your working directory
- Type **`fdisk /dev/sda`**
- Within the **fdisk** utility type **m** for a list of menu options at the "Command (**m** for help):" prompt
- Record the (one-character) fdisk command to:
    - display/list all partitions: **_p**___
    - create a new partition: **_n**___
    - delete a partition: **_d**___
    - list partition types: **_l**___
    - change a partition's system identification: **_t**___
    - save changes made to the partition table: **_w**___
    - exit **fdisk** without saving: **_q**__

- Select the option that lists the partition types and record the **system id** of the following types:
    - "Linux": **_83**____
    - "Linux swap": **__82**___

Exercise #2: Creating a partition

Add a new virtual hard drive in your Ubuntu virtual machin as demonstrated in the lecture period. Make it **2 GB** in size. DO NOT USE YOUR EXISTING PARTITION (**`/dev/sda`**)!!!

1) Create an primary partition:

    - The size of the primary partition is **400MB** and is of type “Linux”

2) Create another primary partition:
   - The size of the primary partition is **300 MB** and is of type of "Linux"

3) Create an extended partition to host three logical drives as the following: (keep in mind that you must make it large enough to encapsulate the logical drive described below. HINT: There will be problems if you try to make it exactly **1000MB.** You will need to experiment and perform some math):

   - The size of the first logical drive is **200MB** and is of type "Linux".
   - The size of the second logical drive is 3**00MB** and is of type "Linux".
   - The size of the third logical drive is 4**00MB** and is of type "Linux Swap".

4) Record the information on the partitions which have been created on the new 1.0GB hard drive:

List all the primary partitions on the new 2GB drive	**/dev/sdb1** **/dev/sdb2**
Name the extended partition if one exists on the new 2 GB drive	**/dev/sdb3**
List all logical drives if they exist on the new 2GB drive	**/dev/sdb5** **/dev/sdb6** **/dev/sdb7**
Can you create additional primary partitions on the new 2GB drive?	Y, N and Why? **One more primary partition can be created if there is free space available**
Can you create additional logical drives on the new 2GB drive?	Y, N and why? **More logical drives can be created if there is free space left in the extended partition**

- Save the changes.
- Reboot by typing reboot.

Exercise #3: Deleting a partition

- Delete the logical drive of **200MB**.
- Save the changes.
- Reboot with command **reboot**
- What do you notice in terms of the partition numbering?

___ **The previous /dev/sda7 is now dev/sda6**_____

_____

Exercise #4: Basic commands review

1) Log in as the default user, what is your default prompt?

   Answer varies

2) What does **whoami** return?

   Answer varies

3) What dose the command **uname** return?

   Linux

4) To display Linux kernel version, you should type:

   `uname -r`

5) Type **hostname** at command line, record the output

   Answer varies

6) Type **pwd**. What does it display?

   Answer varies

7) What does **`wc -l /etc/passwd`** display?

   Answer varies

8) What is the purpose of **wc** command? (using **man** for help)

   displays a count of lines, words, and characters in a file

9) Type the following commands and record the output**:**

   `touch lab51 lab52`

   `ls –i lab51 lab52`

Answer varies

10) Type **`man ls`** and see what the **`-i`** switch does. Explain what it does.

Dispay file index node number

11) What is the command to take you to your home directory? Be specific. Show two methods.

```
cd
cd ~
```

12) How many top-level subdirectories are located under the root(/) of the file system?

Answer varies

13) What command do you use to remove a empty directory?

```
rmdir
```

14) What command and options do you use to delete a complete directory structure including files stored in it?

```
rm -r
```

15) What is the command used to shutdown Linux immediately at command line? (Assume you log in as root)

```
shutdown -h now
```

The Linux File System （Part2）

Due date

- End of Week 10 lab class

Evaluation

- 3% of final grade.

Submission

Hand in or email (in email subject line: CST8102-14F lab6 submission) completed lab before due date.

Materials

- Student laptop computer
- Ubuntu 14.04.1 installed in VMWare Workstation

Procedure

Root privilege is needed for executing most of the commands in this lab.

Exercise #1: Creating a linux filesystem

Use the **mkfs** command to format your newly created two primary partitions in **Lab5**.

Record the command you use:

```
mkfs -t ext4 /dev/sdb1
mkfs -t ext4 /dev/sdb2
```

Note: If no filesystem type is specified with mkfs, mkfs will default to ext2.

Exercise #2: Creating a swap filesystem

When working with swap space we use two commands. One command is used to create a swap filesystem, which is used by the virtual memory system to temporarily store data. The command is **mkswap** and the syntax is:

- **mkswap *device_name***

  Record the command you use to create a swap filesystem on the swap partition you created in **Lab5**:

  mkswap /dev/sdb6

The second command is used to activate the swap space, so that the virtual memory system can use the swap space. The command is **swapon** and the syntax is:

- **swapon *device_name***

  Record the command you use to activate the swap partition:

  swapon /dev/sdb6

To verify the swap partitions that are currently active, use the following command:

- **swapon -s**

  Note: No argument is required.

- Record the output of the above command

Filename	Type	Size	Used	Priority
/dev/sda5	partition	1046524	0	-1
/dev/sdb6	partition	409596	0	-2

Exercise #3: Mount & unmount a Linux filesystem

Create a log file:

Redirect the output of **`fdisk -l`** to the log file named ~/**fslab6** *(if you login as root, ~/fslab6 means /root/fslab6)* using the following command:

- **`fdisk –l > ~/fslab6`**

Append an empty line into the log file using

- **`echo "" >>  ~/fslab6`**

A newly created filesystem is not recorded in the **/etc/fstab** file. Therefore, we need to mount the filesystem manually. In this exercise we mount the Linux partition that we created in lab5.

The syntax of the mount command is: **`mount –t`** ***`type device mount-point`***

To mount the newly created Linux partition:

- **`mkdir /mnt/new`**
  - Create a mount point
- **`mount -t ext4 /dev/sdb1 /mnt/new`**
  - /dev/sdb1 is the newly created primary partition in lab5 part I
- **`ls /mnt/new`**
  - List the directory contents. Since this a new partition you will see only one directory that is created by mkfs: lost+found.

Add to log file:

- **`mount >> ~/fslab6`**
  - Append the output of the mount command to **~/fslab6**
- **`echo "" >>  ~/fslab6`**
  - Append an empty line.
- **`umount /mnt/new`**
  - Unmount the filesystem located on the partition

Exercise #4: Mount & unmount a CDROM disk

Put a CD-ROM or DVD disk in the CDROM drive and follow the steps below (Use the Ubuntu installation ISO image if you don't have a physical disk):

1) **`mkdir /mnt/cdrom`**

   - Create mount point for mounting cdrom

2) **`mount -t iso9660 /dev/cdrom /mnt/cdrom`**

   - Mount a CDROM filesystem and makes the files on it available in the **/mnt/cdrom** directory

3) **`ls -l /mnt/cdrom`**

   - List the contents of the **/mnt/cdrom** directory

Add to log file:

- Append the output of the **mount** command to **~/fslab6**
- Append an empty line

4) **`umount /mnt/cdrom`**

   - Unmount the CDROM filesystem

Exercise #5: Mount a partition automatically during startup

Use **vim** to insert a new line at the end of the **/etc/fstab** file to automatically mount the first primary partition (/dev/sdb1) you created in lab5 at the mount point **/mnt/blank** when the system boots up. Be very careful when editing this file.

Add to log file:

- **`cat /etc/fstab >> ~/fslab6`**
  - Append the new **/etc/fstab** file to **~/fslab6**

Print a copy of the log file, *~/fslab6*, and hand it in along with your lab6 document before the due date.

```
Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000319b5

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    39845887    19921920   83  Linux
/dev/sda2        39847934    41940991     1046529    5  Extended
/dev/sda5        39847936    41940991     1046528   82  Linux swap /
Solaris

Disk /dev/sdb: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders, total 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x355dc252

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048      821247      409600   83  Linux
/dev/sdb2          821248     1435647      307200   83  Linux
/dev/sdb3         1435648     3483647     1024000    5  Extended
/dev/sdb5         1437696     1847295      204800   83  Linux
/dev/sdb6         2465792     3284991      409600   83  Linux


/dev/sda1 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/cgroup type tmpfs (rw)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs
(rw,noexec,nosuid,nodev,size=104857600,mode=0755)
none on /sys/fs/pstore type pstore (rw)
systemd on /sys/fs/cgroup/systemd type cgroup
(rw,noexec,nosuid,nodev,none,name=systemd)
vmware-vmblock on /run/vmblock-fuse type fuse.vmware-vmblock
(rw,nosuid,nodev,default_permissions,allow_other)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse
(rw,nosuid,nodev,user=user1)
/dev/sdb1 on /mnt/new type ext4 (rw)

/dev/sda1 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/cgroup type tmpfs (rw)
none on /sys/fs/fuse/connections type fusectl (rw)
```

```
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs
(rw,noexec,nosuid,nodev,size=104857600,mode=0755)
none on /sys/fs/pstore type pstore (rw)
systemd on /sys/fs/cgroup/systemd type cgroup
(rw,noexec,nosuid,nodev,none,name=systemd)
vmware-vmblock on /run/vmblock-fuse type fuse.vmware-vmblock
(rw,nosuid,nodev,default_permissions,allow_other)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse
(rw,nosuid,nodev,user=user1)
/dev/sr0 on /media/user1/Ubuntu 14.04.1 LTS i386 type iso9660
(ro,nosuid,nodev,uid=1000,gid=1000,iocharset=utf8,mode=0400,dmode=0500,
uhelper=udisks2)
/dev/sr0 on /mnt/cdrom type iso9660 (ro)

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name
devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=ea658fc6-01d7-4cf3-bb62-66e84560982e /               ext4
errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=3198a9c1-ac2d-403d-958b-27a573fef720 none            swap    sw
0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto,exec,utf8 0
0
/dev/sdb1       /mnt/blank      ext4    rw                  0         0
```

CST8102-14F Lab7 **Name:_____** **Section#: _____**

Linux User and Group Management

Due date

End of Week 11 lab class

Evaluation

3% of final grade.

Submission

Hand in or email (in email subject line: CST8102-14F lab7 submission) completed lab before due date.

Materials

- Student laptop computer
- Ubuntu 14.04.1 installed in VMWare Workstation

Procedure

1. Log in to root account with the command: **`su - root`**
   a. Create the following account using **useradd** command:

   username: **user1001**
   home directory: **/home/user1001home** *(create the user's home directory if not exist)*
   login shell: **/bin/bash**

   Record the command you use
   `useradd user1001 -d /home/user1001home -m -s /bin/bash`

b. Define password for this user account and record the command you use

```
passwd user1001
```

c. Verify that the information of the new user account is recorded in the **/etc/passwd** file by typing commands:
   - `cat /etc/passwd | grep user1001`
   - `cat /etc/group | grep user1001`

d. What is the initial group of **user1001?** **user1001**

2. Delete the user account, **user1001**. Make sure the user's home directory is removed, as well as all files stored in it.

a. Record the command you use:

```
userdel -r user1001
```

b. Record the commands you use to verify the user account and the user's home directory are deleted.

```
cat /etc/passwd | grep user1001

ls /home
```

c. Record the output of the command: `grep user1001 /etc/group`

none

3. Use the **useradd** command to create the following user account:

username: **user1002**
home directory: **/home/user1002** *(create the user's home directory if not exist)*
initial group: **users**
login shell: **/bin/bash**
expiration date: **2014-12-31**

a. Record the command you use

```
useradd -m -g users -s /bin/bash -e 2014-12-31 user1002
```

b. Define password for this user account using **passwd** command

c. Record the commands you use to verify the user account is created:

```
grep user1002 /etc/passwd
ls /home
```

d. Log in to **user1002** with the command: **`su - user1002`**,
and copy **/etc/fstab** to **user1002**'s home directory with the following command:
`cp /etc/fstab ~/`

e. List all files in **user1002**'s home directory with the command:
`ls /home/user1002`

4. Switch to **root** user account with the command **`su - root`**, and change the home directory of **user1002** to **/home/user102home,** (Please note that the contents of the current home directory should be moved to the new home directory, which should be created if it does not already exist.)

   a. Record the command you use:

   ```
   usermod -d /home/user102home -m user1002
   ```

   b. Log in to **user1002** with the command**: `su - user1002`**

   c. Record the commands you use to verify the user's home directory has changed to **/home/user102home**

   ```
   cat /etc/passwd | grep user1002
   ls /home
   ```

5. Log in to **user1002**, and change the login shell of **user1002** to **/bin/sh**

   a. Record the command you use:

   ```
   chsh -s /bin/sh
   ```

   b. What is the command you use to verify the login shell has been changed to **/bin/sh**?

   ```
   grep user1002 /etc/passwd
   ```

   c. Switch to **root,** and change the login shell back to **/bin/bash** for **user1002 ,** record the command used:

```
usermod -s /bin/bash user1002
```

6. a. Change the primary (initial) group of **user1002** to **root**, and record the command you use:

```
usermod -g root user1002
```

b. Create a new group named **cst8102**, and record the command you use:

```
groupadd cst8102
```

c. Let **user1002** join the group **cst8102** as a supplementary group, and record the command you use:

```
usermod -G cst8102 user1002
```

d. What is the output of the command: **`groups user1002`**?

user1002: root cst8102

7. a. Log in to **user1002** with the command **`su - user1002`**, and change **user1002**'s group ID (initial group) to **cst8102** for the duration of current session, and record the command you use:

```
newgrp cst8102
```

b. What is the output of the command: **`groups`**?

cst8102 root

c. Log in to root and create a new group named **test**

d. Create a password for the group **test**, and record the command you use:

```
gpasswd test
```

e. Log in to **user1002** with the command **`su - user1002`**, and change **user1002**'s group ID (initial group) to **test** for the duration of current session, and record the command you use:

```
newgrp test
```

f. What is the output of the command: **`groups`**?

```
test root cst8102
```

8. Create the following user accounts using **useradd** commands:

    - Mick Jagger - login name: **mickj**
    - Jones Brian - login name: **jonesb**
    - Keith Richards - login name: **keithr**
    - Charlie Watts - login name: **charliew**

    a. Create a group named **stones**, record the command used

    ```
    groupadd stones
    ```

    b. Designate **Mick Jagger** as the group administrator, and record the command you use:

    ```
    gpasswd -A mickj stones
    ```

    c. Login as **mickj**, and add all the above user accounts to the group **stones,** record the commands you use:

    ```
    gpasswd -a jonesb stones
    gpasswd -a keithr stones
    gpasswd -a charliew stones
    ```

    d. What is the output of the command **`cat /etc/group | grep stones`**?

    ```
    stones:x:1007:jonesb,keithr,charliew
    ```

```
#!/bin/bash
# This program will ask user to input grades for Assignments, Test1, Test2, and
# Final Exam.  It will then calculate an overall letter grade for the user.

read -p "Out of 40%, input the Assignments grade: " assign
read -p "Out of 15%, input the Test 1 grade: " test1
read -p "Out of 15%, input the Test 2 grade: " test2
read -p "Out of 30%, input the Final Exam grade: " final
((total = assign+test1+test2+final))
case $total in
[9][0-9]|[1][0][0]) echo " Your grade is A+ ";;
[8][5-9]) echo " Your grade is A";;
[8][0-4]) echo " Your grade is A-";;
[7][7-9]) echo " Your grade is B+";;
[7][3-6]) echo " Your grade is B";;
[7][0-2]) echo " Your grade is B-";;
[6][7-9]) echo " Your grade is C+";;
[6][3-6]) echo " Your grade is C";;
[6][0-2]) echo " Your grade is C-";;
[5][7-9]) echo " Your grade is D+";;
[5][3-6]) echo " Your grade is D";;
[5][0-2]) echo " Your grade is D-";;
[0]|[0-4][0-9]) echo " Your grade is F";;
esac
```

```
# This program will display a menu to the user with account management options.
# The menu will have 7 options for the user to choose from.  The program will
# use a loop for the main menu and check the data using if statements.  The
# program will terminate when the user presses Q.

menu=z
while [ $menu != q ] && [ $menu != Q ]
do
clear
echo "Account Management Menu"
echo ""
echo "A)   Create a user account"
echo "B)   Delete a user account"
echo "C)   Change supplementary group for a user account"
echo "D)   Change initial group for a user account"
echo "E)   Change default login shell for a user account"
echo "F)   Change account expiration date for a user account"
echo "Q)   Quit"
echo ""
read -p "What would you like to do? " menu

if [ $menu = a ] || [ $menu = A ]
then
        echo "You chose to create a user account"
        read -p "Enter the new username: " uname
        read -p "Enter the user's home directory: " homedir
        read -p "Enter the user's default login shell" lshell
        useradd -d $homedir -s $lshell -m $uname
        sleep 3

elif [ $menu = b ] || [ $menu = B ]
then
        echo "You chose to delete a user account"
        read -p "Enter the username for the account you wish to delete: " uname
        userdel -r $uname
        sleep 3

elif [ $menu = c ] || [ $menu = C ]
then
        echo "You chose to change the supplementary group for a user account"
         read -p "Enter the username: " uname
        read -p "Enter the group name: " gname
        usermod -G $gname $uname
        sleep 3

elif  [ $menu = d ] || [ $menu = D ]
then
        echo "You chose to change the initial group for a user account"
        read -p "Enter the username: " uname
        read -p "Enter the initial group name: " gname
        usermod -g $gname $uname
        sleep 3
```

```
elif  [ $menu = e ] || [ $menu = E ]
then
        echo "You chose to change the default login shell for a user account"
        read -p "Enter the username: " uname
        read -p "Enter the shell name: " sname
        chsh -s $sname $uname
        sleep 3
elif  [ $menu = f ] || [ $menu = F ]
then
        echo "You chose to change the expiration date for a user account"
        read -p "Enter the username: " uname
        read -p "Enter the expiration date (YYYY-MM-DD): " edate
        usermod -e $edate $uname
        sleep 3

elif [ $menu = q ] || [ $menu = Q ]
then
        echo "You chose to quit, goodbye."
        sleep 3

else
        echo "That is not a valid selection, please try again."
        sleep 3
fi
done
```

```
# This program will accept 3 paramaters and perform a simple calculation based
# on those paramaters otherwise the program will enter a menu to ask the user
# for input on doing a simple calculation or to exit.



function add(){
  echo "mycalc $1 + $2"
  (($sum = $1 + $2))
  echo "The sum of $1 plus $2 is $sum"
  sleep 3
  clear
  exit
}

function sub(){
  echo "mycalc $1 - $2"
  (($sub = $1 - $2))
  echo "The subtraction of $1 minus $2 is $sub"
  sleep 3
  clear
  exit
}

choice=a
if [ $# = 3 ] && [ $2 = '+' ] || [ $2 = '-' ]
  then
  if [ $2 = "+" ] && [ $1 -le 0 ] || [ $1 -ge 0 ] && [ $3 -le 0 ] || [ $3 -ge 0 ]
    then
    add $1 $3
  elif [ $2 = "-" ] && [ $1 -le 0 ] || [ $1 -ge 0 ] && [ $3 -le 0 ] || [ $3 -ge 0 ]
    then
    sub $1 $3
  else
    echo "Entering main menu."
    sleep 3
    clear
  fi
else
  while [ $choice != 'X' ] && [ $choice != 'x' ]
    do
    echo "Please make a choice from the following menu"
    echo  "C) Calculation "
    echo  "X) Exit"
    read -p "Option" choice
    if [ $choice = 'C' ] || [ $choice = 'c' ]
      then
      choice1=a
      while [ $choice1 != 'X' ] && [ $choice1 != 'x' ]
        do
        read -p "Please enter an integer number or X to exit:" choice1
        if [ $choice1 = 'X' ] || [ $choice1 = 'x' ]
```

```
        then
        echo "Exiting the program."
        sleep 3
        clear
        exit
      elif [ $choice1 -le 0 ] || [ $choice1 -ge 0 ]
        then
         action=a
         while [ $action != 'X' ] && [ $action != 'x' ]
           do
           echo "Please select an action "
           echo "+) Addition"
           echo "-) Subtraction"
           echo "X) Exit"
           read action
           if [ $action = 'X' ] || [ $action = 'x' ]
             then
             echo "Exiting the program."
             sleep 3
             clear
             exit
             elif [ $action = '+' ] || [ $action = 'a' ] || [ $action = 'A' ]
             then
             read -p "Please enter another number " num2
             add $1 $num2
           elif [ $action = '-' ] || [ $action = s ] || [ $action = 'S' ]
             then
             read -p "Please enter another number " num2
             sub $1 $num2
           else
             echo "That command was not understood, please try again."
             sleep 3
             clear
           fi
         done
      else
        echo "Input not understood."
        sleep 3
        clear
      fi
    done
  elif [ choice = 'X' ] || [ choice = 'x' ]
    then
    echo "Exiting the program."
    sleep 3
    clear
    exit
  else
    echo "Input not understood."
    sleep 3
    clear
  fi
  done
fi
```