Basic Unix/Linux Commands

Jemmy Hu
SHARCNET
University of Waterloo

Linux Features

- multitasking: **several programs running at once**
- multiusers: **several users at the same machine at once**
- multiplatforms: **it runs on many different *CPU***

1) Getting started: try a few simply commands

hostname	Print system name
whoami	Print the current user id (username)
pwd	Print Working Directory
date	Display the date & time

```
[jemmyhu@wha781 ~]$ hostname
Wha781
[jemmyhu@wha781 ~]$ whoami
jemmyhu
[jemmyhu@wha781 ~]$ pwd
/home/jemmyhu
[jemmyhu@wha781 ~]$ date
Tue Jan 16 11:08:30 EST 2007
[jemmyhu@wha781 ~]$
```

2) Password Changing: passwd

NOTE: you are supposed to do this on SHARCNET webportal, not on the cluster

```
[jemmyhu@wha781 ~]$ passwd
Changing password for user jemmyhu.
Changing password for jemmyhu
(current) UNIX password:
(You will write your actual password after colon.)
New password:
(You will write your new password after colon.)
Re-type new password:
(You will repeat your new password after colon.)
Password changed
```

If you write your old password and a new password correctly, the new one becomes valid.

NOTE: The following Demos were done on a linux cluster named watsci1

3) list contents of the working directory: ls

ls displays the names of all subdirectories and files in the working (current) directory.

```
[jemmyhu@watsci1 ~]$ ls
g03uw-teach-C.02-1.x86_64.rpm  gaussian  gaussview  README  test  test1
[jemmyhu@watsci1 ~]$
```

If the command **ls** is written with parameter ***-l*** then the command lists contents of the working directory with details.

```
[jemmyhu@watsci1 ~]$ ls -l
total 156204
-rw-rw-r--  1 jemmyhu jemmyhu 159772215 Dec 21 15:01 g03uw-teach-C.02-1.x86_64.rpm
drwxrwxr-x  4 jemmyhu jemmyhu      4096 Jan  3 11:07 gaussian
drwxrwxr-x  3 jemmyhu jemmyhu      4096 Jan  9 12:08 gaussview
-rw-rw-r--  1 jemmyhu jemmyhu      3226 Jan 10 20:45 README
drwxr-x---  5 jemmyhu jemmyhu      4096 Jan 10 14:48 test
-rw-rw-r--  1 jemmyhu jemmyhu         0 Jan 10 14:48 test1
[jemmyhu@watsci1 ~]$
```

The first column is empty for data and programs or in first column is written character *d* if the item is directory etc. The next three columns are permissions for the user, the columns 5, 6, 7 for the user's group (in this example the group is called user), and the last three for the rest of the word. The next information in the row is the size of the file (in bytes), the date of the last update, and the name of the file.

4) Make Directory: mkdir

mkdir my_dir

makes new directory my_dir (the path is given relative) as a subdirectory of the current directory.

5) Remove Directory: rmdir

rmdir my_dir

removes directory my_dir if it is empty. If you want to remove not empty directory

rm -r your_file

6) Change working directory: cd

cd The system is returned to the user home directory

```
[jemmyhu@watsci1 ~]$ cd
[jemmyhu@watsci1 ~]$ pwd
/home/jemmyhu
[jemmyhu@watsci1 ~]$
```

cd my_dir

changes the directory my_dir which path is given relative to the working directory. If the working directory is, for example, /home/smith then the working directory will become /home/smith/try_it

cd .. Move to superior directory

```
[jemmyhu@watsci1 ~]$ mkdir my_dir
[jemmyhu@watsci1 ~]$ ls
g03uw-teach-C.02-1.x86_64.rpm  gaussian  gaussview  my_dir  README  test  test1
```

```
[jemmyhu@watsci1 ~]$ cd my_dir/
[jemmyhu@watsci1 my_dir]$ ls
[jemmyhu@watsci1 my_dir]$ pwd
/home/jemmyhu/my_dir
[jemmyhu@watsci1 my_dir]$ touch test
[jemmyhu@watsci1 my_dir]$ ls
test
[jemmyhu@watsci1 my_dir]$ cd ..
[jemmyhu@watsci1 ~]$ pwd
/home/jemmyhu
[jemmyhu@watsci1 ~]$ ls
g03uw-teach-C.02-1.x86_64.rpm  gaussian  gaussview  my_dir  README  test  test1
[jemmyhu@watsci1 ~]$ rmdir my_dir
rmdir: `my_dir': Directory not empty
[jemmyhu@watsci1 ~]$ rm -r my_dir
[jemmyhu@watsci1 ~]$ ls
g03uw-teach-C.02-1.x86_64.rpm  gaussian  gaussview  README  test  test1
[jemmyhu@watsci1 ~]$
```

7) Create an empty file: touch

```
[jemmyhu@watsci1 ~]$ touch test2
 [jemmyhu@watsci1 ~]$ ls
g03uw-teach-C.02-1.x86_64.rpm  gaussian  gaussview  my_dir  README  test  test1  test2
[jemmyhu@watsci1 ~]$ ls -l test2
-rw-rw-r--  1 jemmyhu jemmyhu 0 Jan 16 12:22 test2
[jemmyhu@watsci1 ~]$
```

8) Display file: more, less

more filename (less filename)

User can controle the output:

- press space...the next screen is displayed
- press enter...the next row is displayed
- press q.......the command is finished

9) Copy file: cp

cp file_1 file_2

copies file_1 to file_2. The both files must be in the same working directory. If they are in various directories, the path must be given.

10) Rename and/or Move file: mv

mv file_1 file_2

moves file_1 to file_2. The both files must be in the same working directory. If they are in various directories, the path must be given. The file_1 is removed from the disk.

11) Remove file: rm

rm file_a

removes the file_a from the system at all. If you use wildcard. For example

rm h*c

you will remove all files beginning with *h* and ending with *c* which are in working directory.

If you write

```
rm *
```

you will erase all files from your working directory. If you write

```
rm -i *
```

it will be done also but the system will ask for permission before removing each file.
The command

```
rm -r your_file
```

causes removing of the directory your_file even if this directory is not empty.

12) Find file: find

```
find /usr -name lpr
```

finds a file lpr in subdirectores in directory usr.

```
[jemmyhu@watsci1 ~]$ find /usr -name lpr
/usr/share/terminfo/l/lpr
find: /usr/share/ssl/CA: Permission denied
/usr/bin/lpr
[jemmyhu@watsci1 ~]$
```

13) Access Permission of file: chmod

Example

```
[jemmyhu@watsci1 ~]$ ls -l test1
-rw-rw-r--  1 jemmyhu jemmyhu 30 Jan 16 11:54 test1
```

The letters written from the 2nd to the 10th column are for file permission. The first three columns are permissions for the user, the columns 5, 6, 7 for the user's group (in this example the group is called user), and the last three for the rest of the word. The next table clarifies the meaning of the letters written from the 2nd to the 10th column:

Character	Meaning
r	Permission for reading
w	Permission for writing
x	File is executable

If we want the file to be read and write exclusively by you only, we can write the command:

```
[jemmyhu@watsci1 ~]$ chmod 600 test1
[jemmyhu@watsci1 ~]$ ls -l test1
-rw-------  1 jemmyhu jemmyhu 30 Jan 16 11:54 test1
```

To understand this you must know that number 1 "allows" and number 0 "suppresses" access permission and you need to know relation between octal and binary numbers:

Octal scale	Binary scale
0	000
1	001
2	010

3	011
4	100
5	101
6	110
7	111

14) Some Postfixes of Files

Postfix	Meaning
.c	Source program in C language
.f	Source program in Fortran 77
.f90	Source program in Fortran 90
.p	Source program in Pascal
.pbm	bi-level, black and white image (2 bits per pixel)
.pgm	grayscale (8 bits per pixel)
.ppm	color (24 bits per pixel)
.jpg	compressed by JPEG

15) Determine file Type: file

```
[jemmyhu@watsci1 ~]$ file test1
test1: ASCII text
```

16) Input and Output Redirection

The command

```
p < my_in_file > my_out_file
```

causes input to the executable program p from my_in_file and output from the program p to the my_out_file. It is written from the beginning of this file.

The command

```
p < my_in_file >> my_out_file
```

has the same meaning but the output is appended to the contents of my_out_file

17) View process: ps

ps –u yourid

```
[jemmyhu@watsci1 ~]$ ps -u jemmyhu
  PID TTY          TIME CMD
25591 ?        00:00:00 sshd
25592 pts/0    00:00:00 bash
25982 pts/0    00:00:00 ps
[jemmyhu@watsci1 ~]$
```

18) Starting and Stopping Processes in the background (do not try these now)

```
$ p&
[1] 13456
```

the process *p* starts and it is running on the background. The number which appears in the screen is *PID* (Process Identification Number). You can do on the foreground other activities but you can stop the process *p* at any time by a command

```
$ kill -9 13456
```

which stops the process *p* before this has been finished normally.

19) Checking disk space: du

du command lists the size, in kilobytes, of all directories at or below the current point in the file system. Examples:

```
[jemmyhu@watsci1 gaussview]$ pwd
/home/jemmyhu/gaussview
[jemmyhu@watsci1 gaussview]$ du
10908   ./TAR/gv/lib
6280    ./TAR/gv/help/pix
84      ./TAR/gv/help/icons
2320    ./TAR/gv/help/refs
10632   ./TAR/gv/help
60      ./TAR/gv/bin
824     ./TAR/gv/data/fonts
84      ./TAR/gv/data/elements/fragments
84      ./TAR/gv/data/elements/bitmaps
184     ./TAR/gv/data/elements
408     ./TAR/gv/data/biofrags/fragments
420     ./TAR/gv/data/biofrags
148     ./TAR/gv/data/rings/fragments
148     ./TAR/gv/data/rings/bitmaps
304     ./TAR/gv/data/rings
88      ./TAR/gv/data/rgroups/fragments
88      ./TAR/gv/data/rgroups/bitmaps
184     ./TAR/gv/data/rgroups
4092    ./TAR/gv/data
31548   ./TAR/gv
135028  ./TAR
135040  .
[jemmyhu@watsci1 gaussview]$ du ../test
88      ../test/serial
720     ../test/mpi
8       ../test/openmp
820     ../test
[jemmyhu@watsci1 gaussview]$
```

20) Measure Program running time: time

```
[jemmyhu@watsci1 jeffclass]$ time ./pi-serial
PI = 3.141593

real    0m0.002s
user    0m0.001s
```

```
sys     0m0.002s
[jemmyhu@watsci1 jeffclass]$
```

21) Manual and Command man

The important command of *Unix/Linux* is command man.Command man accesses information from the on-line version of *Unix/Linux*. You can find the description of man by typing
man man.
for example:

```
    man ls
LS(1)                         User Commands                         LS(1)

NAME
       ls - list directory contents

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

DESCRIPTION …..
```

Other important commands include

diff	Display the differences between two files
df	Display free disk space
echo	Display message on screen
env	Environment variables
grep	Search file(s) for lines that match a given pattern
gzip	Compress or decompress named file(s)
tar	Tape ARchiver
top	List processes running on the system
wc	Print byte, word, and line counts
which	Locate a program file in the user's path

For a list of Linux commands, please go to http://www.ss64.com/bash/

References:

Quick Linux Tutorial, Jiri Vogel, http://www.fsid.cvut.cz/cz/U201/LINUX.HTML