LINUX SHELL PROGRAMMING

Operating System Lab

Introduction

- Many people says that Linux is a command based operating system.
- So many of us thinks that Linux is not so user friendly OS.
- But it is not true. Linux is a GUI based OS with a Shell which is more powerful than its counter part in Windows OS.
- We will be familiar with some shell commands.

Identity

- Type ***uname*** *and Linux will tell his name to you*

```
$ uname
    Linux
```

- If you want to know your name type ***whoami***

```
$ whoami
    user1
$ who am i
    user1       tt3a     Jan 12 07:14
$ who
    user1       tty3a    Jun 10 09:15
    user2       tty3c    Jun 10 09:25
```

Manual

- For each command Linux contains manual. To view the manual : ***man*** *name*
  - ***man uname***

Creating Files

- There are two commands to create file.
  - *touch*
  - *cat*

```
$ touch sample
$ touch sample1 sample2 sample3
```

- The size of the file would be zero bytes since *touch* doesn't allow to store anything in a file.

- Useful to create several empty files quickly.

Creating Files (Cont.)

- Store lines in a file

```
$ cat > test
Hello world
I love my country
```

- After completion of writing press the keys ***ctrl+d***.

- In Unix the keys ***ctrl+d*** indicate the EOF or end of file character.

Creating Files (Cont.)

- To see the contents of the file test

```
$ cat test
Hello world
I love my country
```

- It can concatenate the contents of two files and store them in the third file.

```
$ cat sample1 sample2 > newsample
```

- This would create newsample which contains contents of sample1 followed by that of sample2.

```
$ cat sample1 sample2 >> newsample
```

File Operations

- To copy a file : **cp**
  - Copy source to destination or multiple sources to directory
  - **-i** [prompt before overwrite]
  - **-r** [copy directories recursively]
  - **-u** [copy only when the src file is newer than the dest file or when the dest file is missing]

```
$ cp  /usr/home/chapter1  /usr/newhome/ch1
```

File Operations (Cont.)

- To remove a file or directory : **rm**
  - **-f** [ignore nonexistent files, never prompt]
  - **-i** [prompt before any removal]
  - **-r** [remove the contents of directories recursively]
  - **-v** [explain what is being done]

```
$ rm -i  /usr/home/chapter1
$ rm -r  home
```

File Operations (Cont.)

- To move or rename a file : **mv**
  - rename src to dest or move src(s) to directory
  - **-i** [prompt before overwrite]
  - **-u** [move only when the src file is newer than the dest file or when the dest file is missing]
  - **-v** [explain what is being done]

```
$ mv  olddir  newdir
$ mv  file1  file2  newdir
```

Directory and File Listings

- To list information about directory or files : **ls**
- This command contains some options.
  - -a [do not hide entries starting with .]
  - -A [do not list implied . and ..]
  - -h [print sizes in human readable format]
  - -l [use a long listing format]
  - -S [sort by file size]

Directory and File Listings (Cont.)

- Any file name which begins with a '.' is treated as a hidden file.

```
$ ls –a
.
..
Home1
Dev0
Sample1
```

- . stands for the current directory.
- .. stands for the parent previous directory.

Directory and File Listings (Cont.)


```
$ ls p*
pakde
pommies

$ ls ?ain
gain
main
Pain

$ ls /mydir/*x
cc_fax
i_tax
myUnix
```

```
$ ls [aeiou]*
area
even
ion

$ ls [!aeiou]*
dell
sam

$ ls [a-m][c-z][4-9]??
az6ps
```

Directory and File Listings (Cont.)


```
$ ls  -l
total 22
-rwxr-x--x    1   user1   group   24 Jun 06 10:12  carribeans
-rwxr-x-wx    1   user1   group   23 Jun 06 00:05  kangaroos
-rwxr-xr-x    1   user1   group   12 Jun 06 12:54  kiwis
drwxr-xr-x    1   user1   group   10 Jun 06 11:09  mydir
-rwxr-xrwx    2   user1   group   22 Jun 06 14:04  pakde
-rwxrwxr-x    2   user1   group   16 Jun 06 22:25  pommies
-rwxr-xr-x    1   user1   group   04 Jun 06 23:16  springboks
-rwxr-xr-x    1   user1   group   04 Jun 06 10:17  zulus
```

Directory and File Listings (Cont.)

File Type	Meaning
-	Ordinary file
d	Directory file
c	Character special file
b	Block special file
l	Symbolic link
s	Semaphore
p	Named pipe
m	Shared memory file

Directory and File Listings (Cont.)

- To create Link

```
$ ln    poem   mypoem
```

- Avoids unnecessary duplication of the same file contents in different directories.

- By default any new file that we create has one link whereas any new directory we create has two links.

Directory and File Permissions

- Each file or directory has 3 security groups.
  - Owner
  - Group
  - All Others
- Each security group has 3 flags that control the access status : read, write, execute
- They are listed as 'rwx' or a "-" if the access is turned off.
  - rwxrwxrwx[read, write and executable for owner, group and all others]
  - rw-r--r--[read and write by owner, read only for group and all others]

Directory and File Permissions (Cont.)

- To change the permissions type chmod
  - u, g, o or all [whose permission you are changing]
- + or - [type of change: add or subtract permission]
- combination of r, w or x [which permission you are changing: read, write or execute]
- To change Permission
  - chmod [who] [+/-/=] [permissions] file

Directory and File Listings (Cont.)

- file or directory [name of file or directory to change]
  - chmod go+rw file1 file2 add read and write access for group and others for files 'file1' and 'file2'
  - chmod a+rwx file1 add read, write and execute for everyone for 'file1'.
  - chmod 555 file1

```
$ chmod go-x myfile
$ chmod go+r,go-w myfile
$ chmod go=r,u=rw myfile
$ chmod +w myfile
$ chmod 744 myfile
```

A Bit of Mathematics

- Calculator is invoked at shell prompt by typing ***bc***.
- The input to the calculator is taken line by line.
- By typing ***bc*** at prompt the calculator mode starts and the $ the prompt disappears.
- Typing quit ends tryst with bc.

```
$ bc
10/2*2
10
2.5*2.5+2
8.25
quit
```

A Bit of Mathematics (Cont.)

- Working with floats

```
$ bc
scale = 1
22/7
3.1
2.25+1
3.35
quit
```

- After Setting the scale variable if the answer of an expression turns out more than what scale can provide then the value in scale is ignored and the correct answer is displayed.

A Bit of Mathematics (Cont.)

- Working with different base

```
$ bc
obase=16
ibase=2
11010011
89275
1000+100
C
quit
```

- By setting the variable ibase to 2 and obase to 16 all input that is supplied is taken as binary whereas all output is displayed in hexadecimal.

A Bit of Mathematics (Cont.)

- Working with functions

```
$ bc
Sqrt(196)
14
$ bc -l
s(3.14)
0
quit
```

- Trigonometric functions expect their arguments in radians and not in degrees.

A Bit of Mathematics (Cont.)

- Working with variables

```
$ bc
a=4
b=5
c=b-a
c
1
quit
```

```
$ bc
for(i=1 ; i<=5 ; i++) i
1
2
3
4
5
quit
```

```
$ bc
4+2
6
 .+1
7
quit
```

A Bit of Mathematics (Cont.)

- Another math related command in Unix: ***factor***
- If a positive number less than 2^46 is types in then it factorise the number and print its prime factors. Each one is printed the proper number of times.

```
$ factor 15
15: 3 5
$ factor
15
15: 3 5
28
28: 2 2 7
to quit: ctrl+d or q
```

A Bit of Mathematics (Cont.)

- Another math related command in Unix: ***factor***
- If a positive number less than 2^46 is types in then it factorize the number and print its prime factors. Each one is printed the proper number of times.

```
$ factor 15
15: 3 5
$ factor
15
15: 3 5
28
28: 2 2 7
to quit: ctrl+d or q
```

Redirection

- **Input redirection:**
  - < -get input from file instead of the keyboard
- **Output redirection:**
  - > -send output to file instead of the terminal window
- **Append output:**
  - >> -command is used to append to a file if it already exists

Redirection (Cont.)

- Example:

```
$ ls  –l  >  note.txt
```

Calender

- To view calendar in the shell: ***cal***

```
$ cal
$ cal  2012
$ cal  January  2101
$ cal  2  1997
$ cal  feb  2001
$ cal  sep  1752
```

Thanks