Understanding Unix/Linux for Developers

Outline

1. Overview of Unix System
2. Basic Commands
3. Relative & Absolute Path
4. Redirect, Append and Pipe
5. Permission
6. Process Management
7. Install Software
8. Text Editor
9. Foreground and Background Jobs

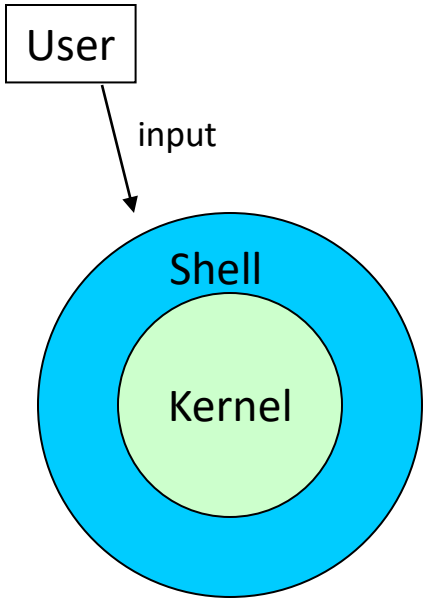
Overview of Unix System

Kernel & Shell

- Unix/Linux is operating system (OS).
- Unix system is described as kernel & shell.

- Kernel is a main program of Unix system. it controls hard wares, CPU, memory, hard disk, network card etc.

- Shell is an interface between user and kernel. Shell interprets your input as commands and pass them to kernel.



Unix Overview (cont.)

Multi-user & Multi-process

- Many people can use one machine at the same time.

File & Process

- Data, directory, process, hard disk etc (almost everything) are expressed as a file.
- Process is an running program identified by a unique id (PID).

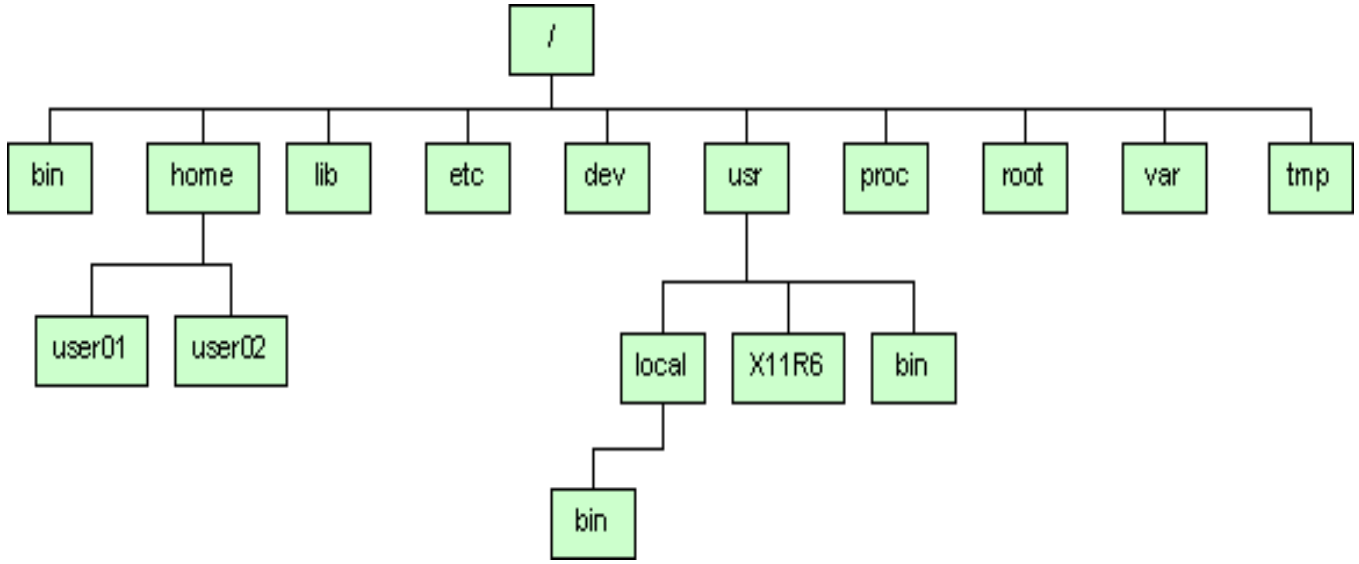
Unix Overview (cont.)

Directory Structure

- Files are put in a directory.
- All directories are in a hierarchical structure (tree structure).
- User can put and remove any directories on the tree.
- Top directory is “/”, which is called slash or root.
- Users have the own directory. (home directory)

Unix Overview (cont.)

Directory Structure



Unix Overview (cont.)

Important Directories

- /bin This contains files that are essential for correct operation of the system. These are available for use by all users.
- /home This is where user home directories are stored.
- /var This directory is used to store files which change frequently, and must be available to be written to.
- /etc Various system configuration files are stored here.

Unix Overview (cont.)

Important Directories

- /dev This contains various devices as files, e.g. hard disk, CD-ROM drive, etc.
- /sbin Binaries which are only expected to be used by the super user.
- /tmp Temporary files.

Unix Overview (cont.)

Normal user and Super user

- In Unix system, there is one special user for administrator, which can do anything.
- This special user is called root or superuser.

Case Sensitivity

- Unix is case-sensitive.
- MYFILE.doc, Myfile.doc, mYfiLe.Doc are different.

Online Manual

- Unix has well-written online manuals.

Basic Commands

How to run commands

- When you log on Unix machine, you will see,

[someone]$

- One command consists of three parts, i.e. command name, options, arguments.

Example)

[someone~]$ command-name optionA optionB argument1 argument2

Basic Commands

How to run commands

- Between command name, options and arguments, space is necessary.

- Opitions always start with "-"

- Example:

  cd ..

  ls –l .bashrc

  mv fileA fileB

Basic Commands

Commands

- ls show files in current position
- cd change directory
- cp copy file or directory
- mv move file or directory
- rm remove file or directory
- pwd show current position
- mkdir create directory
- rmdir remove directory
- less, more, cat display file contents
- man display online manual

Basic Commands

Commands

• su	switch user
• passwd	change password
• useradd	create new user account
• userdel	delete user account
• mount	mount file system
• umount	unmount file system
• df	show disk space usage
• shutdown	reboot or turn off machine

Basic Commands

1. Type following command in your directory.

   ls

   ls –a

   ls –la

   ls -Fa

2. Make a directory

   mkdir linux

   pwd

   cd linux

   pwd

   cd

   pwd

   rmdir linux

3. In your home directory,

   ls .bash_profile

   cp .bash_profile sample.txt

   less sample.txt (note: to quit less, press "q")

   rm sample.txt

4. check disk space usage

   df

   df -h

Relative & Absolute Path

- Path means a position in the directory tree.
- To express a path, you can use relative path or absolute path.
- In relative path expression, the path is not defined uniquely, depends on your current path.
- In absolute path expression, the path is defined uniquely, does not depend on your current path.

Absolute Path

- Address from the root

  /home/linux/

  ~/linux

  ~: ~: Alt+N

- Similar to:

  Lausanne University/Lausanne/Canton de Vaud/ Switzerland/Europe/Earth/Solar System/

Relative Path

- Relative to your current location

  . : your current location

  .. : one directory above your current location

  pwd: gives you your current location

- Example

  ls ./linux : lists the content of the dir linux
  ls ../../ : lists everything that is two dir higer

- Similar to:

  Go Left/turn right/take the TSOL/go

Relative & Absolute Path

- Relative Path

  pwd

  cd .

  pwd

  cd ..

  pwd

  cd ..

  pwd

  cd

- Ablsoute Path

  cd

  mkdir mydir

  pwd

  cd /Users/invite

  pwd

  cd /Users

  pwd

  cd /

  pwd

  cd /Users/invite

  cd ~/mydir

Redirect, Append and Pipe

Redirect and append

- Output of command is displayed on screen.
- Using “>”, you can redirect the output from screen to a file.
- Using “>>” you can append the output to the bottom of the file.

Pipe

- Some commands require input from a file or other commands.
- Using “|”, you can use output from other command as input to the command.
- On MacOSX, The Pipe sign: (**Shift+Alt+N: franc, Alt+7)**

Redirect, Append and Pipe

Commands

- head show first several lines and omit other lines.
- tail show last several lines and omit other lines.
- **grep XXX File** show lines matching pattern XXX in File

Redirect, Append and Pipe

- In home directory, type

  ls -1 > sample.txt

  less sample.txt
- Use redirect.

  head -3 sample.txt

  head -3 sample.txt > redirect.txt
- Use append.

  tail -3 sample.txt

  tail -3 sample.txt >> redirect.txt

  less redirect.txt

- Use pipe.

  less redirect.txt
  grep Desk redirect.txt
  grep –n Desk redirect.txt
  man grep
  tail redirect.txt | grep Desk
  rm sample.txt
  rm redirect.txt

Sorting

Commands

- sort — Sorts using the first field of each line.

- -n — Sorts considering the numeric value of the strings
- -k3 — Sorts using the third field of each line
- -rnk3 — Sorts in reverse order, using the numeric value of the third field

Redirect, Append and Pipe

- Identify the largest file in a directory:

```
ls –la /bin/ | sort –nk5 | tail -1
```

Permission

- All of files and directories have owner and permission.
- There are three types of permission, readable, writeable and executable.
- Permissions are given to three kinds of group. owner, group member and others.

Example:

```
ls -l .bash_profile
-rw-r--r--  1 cnotred  cnotred  191 Jan  4 13:11 .bash_profile
```

- r:readable, w:writable, x: executable

Permission

Command

- chmod — change file mode, add or remove permission
- chown — change owner of the file

Example)

chmod a+w filename

add writable permission to all users

chmod o-x filename

remove executable permission from others

chmod a+x

Gives permission to the usser to execute a file

- u: user (owner), g: group, o: others a: all

Permission

- Check permission

  ls –l .bash_profile

  cp .bash_profile sample.txt

  ls –l sample.txt

- Remove readable permission from all.

  chmod a-r sample.txt

  ls –l sample.txt

  less sample.txt

- Add readable & writable premissions to file owner.

  chmod u+rw sample.txt

  ls –l sample.txt

  less sample.txt

  rm sample.txt

Process Management

- Process is a unit of running program.
- Each process has some information, like process ID, owner, priority, etc.

Example) Output of “top” command

```
  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
12035 nomura    15   0  1080 1080   840 R     0.3  0.2   0:00 top
    1 root      15   0   472  436   420 S     0.0  0.0   0:04 init
    2 root      15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root      15   0     0    0     0 SW    0.0  0.0   0:00 kapmd
    4 root      34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    5 root      15   0     0    0     0 SW    0.0  0.0   0:59 kswapd
    6 root      15   0     0    0     0 SW    0.0  0.0   0:00 bdflush
```

Process Management

Commands

- kill — Stop a program. The program is specified by process ID.
- killall — Stop a program. The program is specified by command name.
- ps — Show process status
- top — Show system usage statistics

Process Management

- Check your process.

  ps

  ps –u

- Check process of all users.

  top (To quit top, press "q")

  ps –e

  ps –ef

- Find your process.

  ps –ef | grep cnotred

Install Software

- Unix system has a "*de facto* standard" way to install a software.

  configure, make & make install

- Typical software installation procedure as following.
  1. Download source code. Usually, it's archived with tar command and compressed with gzip command.
  2. configure command creates Makefile automatically which is used to compile the source.
  3. Program compilation is written in Makefile.

Install Software

Commands

- gzip compress a file
- gunzip uncompress a file
- tar archive or expand files
- configure create Makefile
- make compile & install software

Install Software

Example: parallel programming library installation

```
gunzip software.tar.gz
tar –xvf  software.tar
cd software
./install OR make all OR …
```

Text Editor

pico

- Programs & configuration files are text file.
- There are two popular text editors, vi and Emacs.
- Although they are very powerful and useful, it is also true that they are complicated for beginners and difficult to learn.
- pico is an easy and simple alternative.

Text Editor

Commands

- Arrow-keys Move cursor
- CTRL+a Move to the beginning of the current line.
- CTRL+e Move to the end of the current line.
- CTRL+v Move forward one page.
- CTRL+y Move backward one page.
- CTRL+w Search for text.
- CTRL+d Delete the current character.
- CTRL+k Remove (cut) current line or selected text.
- CTRL+u Paste (uncut) last cut text at the cursor position.
- CTRL+o Save (output) the file.
- CTRL+x Exit Pico, saving the file.

- Autre: xemacs, emacs

Text Editor

- Create the file Hello
  pico hello.pl

- Write hello.pl as follows.

```
#!/usr/bin/perl
print “Hello World\n”;
```

- Make il executable
  chmod u+x hello.pl

- Run it!
  ./hello.pl

Foreground and Background

- Running job has two modes, “foreground” and “background”
- If program is running as “background”,

  the program keeps running even after your session was closed
- If program is running as “foreground”,

  Ctrl-C stop program

  Ctrl-Z let program background

Foreground and Background

- To run programs in background mode, use “&”

  [nomura@ssc-1]$ *command* &

- To get background job back into foreground mode, use “fg” command.

  [nomura@ssc-1]$ fg

Remote Login & File Transfer

- rshd, telnetd, ftpd, sshd are server program and provide similar services, remote login & file transfer.
- The major difference is security level.

rshd < telnetd + ftpd < sshd

Commands

Client

- rsh & rcp
- telnet & ftp
- ssh & scp

Server

- rshd
- telnetd & ftpd
- sshd

Remote Login & File Transfer

Remote login & file transfer system are based on server and client model. client program on your machine ask sever program certain service remote machine.

For example, telnet server provides remote login service. ftp server provides file transfer service.

Sample client programs;

WS FTP	FTP client
Internet Exploror	HTTP client
Eudora	POP, SMTP client