



Tutorial on Linux Basics

KARUNYA LINUX CLUB

www.karunya.edu/linuxclub



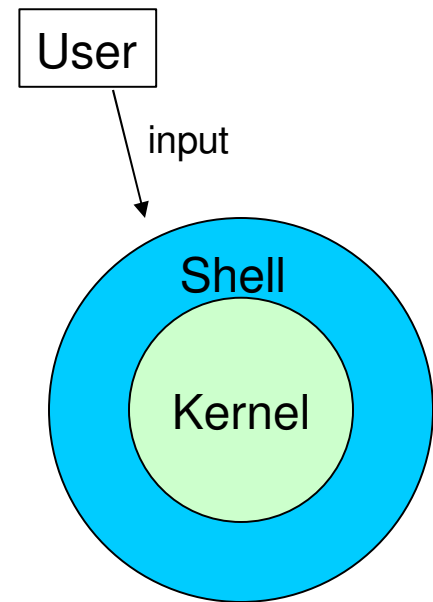
Outline

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

Overview of Linux System

Kernel & Shell

- Linux is operating system (OS).
- Linux system is described as kernel & shell.
- Kernel is a main program of Linux 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.





Linux 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).



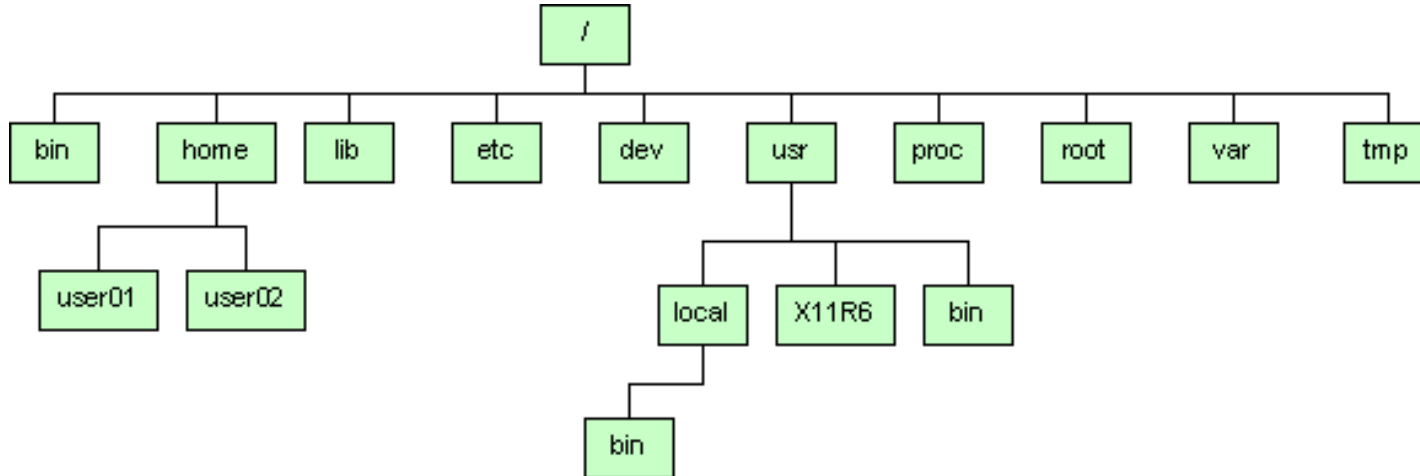
Linux Overview (cont.)

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)

Linux Overview (cont.)

Directory Structure



Linux 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.
- /mnt Provides a location for mounting devices, such as remote filesystems and removable media
- /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.

Linux Overview (cont.)

Important Directories

- /dev This contains various devices as files, e.g. hard disk, CD-ROM drive, etc.
- /root This is the root (administrator) user's home directory
- /sbin Binaries which are only expected to be used by the super user.
- /tmp Temporary files.
- /boot Has the bootable Linux kernel and boot loader configuration files (GRUB)
- /usr Contains user documentation, games, graphical files, libraries (lib), etc..



Linux Overview (cont.)

Normal user and Super user

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

Case Sensitivity

- Linux like UNIX is case-sensitive.
- MYFILE.doc, Myfile.doc, mYfiLe.Doc are different.

Online Manual

- Linux has well-written online manuals.



Basic Commands

How to run commands

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

```
[cswug@hyperion001 cswug]$
```

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

Example)

```
[cswug~]$ command-name optionA optionB argument1 argument2
```

Basic Commands

How to run commands

- Between command name, options and arguments, space is necessary.
- Options 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
- cat display file contents
- less display file contents pagewise
- 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

Practice 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. Try to change your password,

```
passwd username
(Type current password once, then
type new password twice. You don't have to
change password here. Just a practice)
```

5. 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.

Relative & Absolute Path

- Characters used in relative path
 - . current directory
 - .. parent directory

- Example)

```
cd ..  
./a.out
```

- Absolute path starts with “/”

- Example)

```
cd /home/user01  
/home/root/a.out
```


Relative & Absolute Path

- Use relative path.

In home directory, type

```
pwd
```

```
cd .
```

```
pwd
```

```
cd ..
```

```
pwd
```

```
cd ..
```

```
pwd
```

```
cd
```

- Use absolute path.

In home directory, type

```
pwd
```

```
cd /home/cswug
```

```
pwd
```

```
cd /home
```

```
pwd
```

```
cd /
```

```
pwd
```

```
cd /home/cswug
```



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.



Redirect, Append and Pipe

Commands

- head show first several lines and omit other lines.
- tail show last several lines and omit other lines.
- grep show lines matching a pattern

Redirect, Append and Pipe

- In home directory, type

```
ls .bash_profile
cp .bash_profile 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 PATH redirect.txt
tail redirect.txt | grep PATH
rm sample.txt
rm redirect.txt
```

Permission

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

Example)

```
[cswug@hyperion001 cswug]$ ls -l .bash_profile  
-rw-r--r--  1 cswug  cswug    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

- `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 permissions 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 informations, 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 *Sends specified signal to specified process. This process 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 cswug`

Install Software

- Linux 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

Executing commands line by line:

```
tar xvfz mpich.tar.gz  
cd mpich-1.2.5.2  
./configure --prefix=/usr/local -rsh=ssh  
make  
make install
```



Text Editor

- Programs & configuration files are text file.
- There are two popular text editors, vi and Emacs.
- Very powerful and useful.

Text Editor

Write “hello world” program in C (assuming GCC is installed on your Linux system).

- Type `vi hello.c` [*where vi is editor name and hello.c is filename*]
- Press `I` for inserting and type `hello.c` as follows.

```
#include <stdio.h>
int main(void){
    printf("Hello World\n");
}
```

After finishing typing save the file by pressing `ESC` button and then typing colon followed by `wq` as follows:

`:wq` [*w to save and q to quit*]

- Compile
`gcc hello.c`
- Run
`./a.out`

Text Editor

- We open the vi editor for editing/writing a file using vi command as :
vi filename *[where filename is the name of the file used]*
- Press I for inserting (ie-for typing in the editor), if any.
- After typing, press ESC followed by (either of the following)
 - :q *[to quit simply]*
 - :wq *[to save and quit]*
 - :q! *[to force quit without saving]*



Tutorial of Linux

E
N
D