

Linux Tutorial

Ken-ichi Nomura

3rd Magics Materials Software Workshop



*Gaithersburg Marriott
Washingtonian Center*
November 11-13, 2018



Wireless Network Configuration

Network Name:

Marriott_CONFERENCE
(only for laptop)

Passcode: usc2018

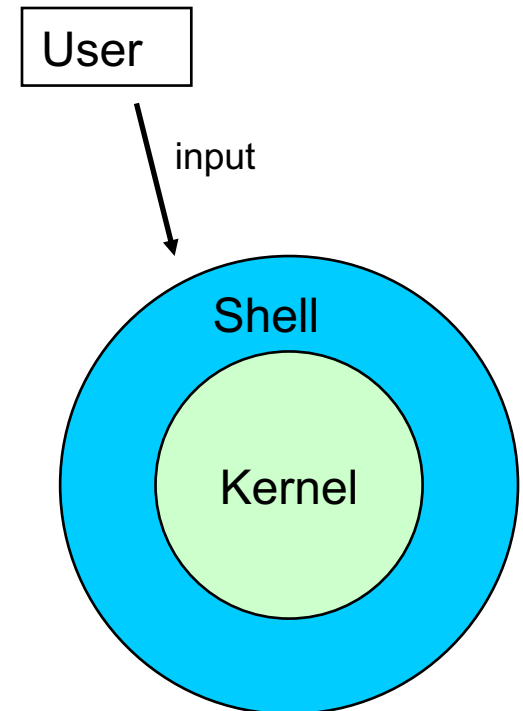
Outline

1. Introduction to Linux
2. Basic Commands
3. Relative & Absolute Path
4. Redirect, Append and Pipe
5. Text Editor
6. Some Tips
7. HPC Cluster and SLURM

Introduction to Linux

Kernel & Shell

- Like Windows, Linux is an operating system (OS).
- Linux consists of kernel & shell.
- Kernel is the main part of Linux system that controls hardware, CPU, memory, storage disks, network etc.
- Shell is a program to bridge between users and the kernel. Users use commands to tell what you want to do to the kernel.



Introduction to Linux

Text-based Command Line Interface (CUI)

- No mouse but keyboard. User needs to type series of commands to tell computer what you want to do.

Case Sensitivity

- Unix is case-sensitive. **MYFILE.doc**, **Myfile.doc** and **mYfiLe.Doc** are all different files.

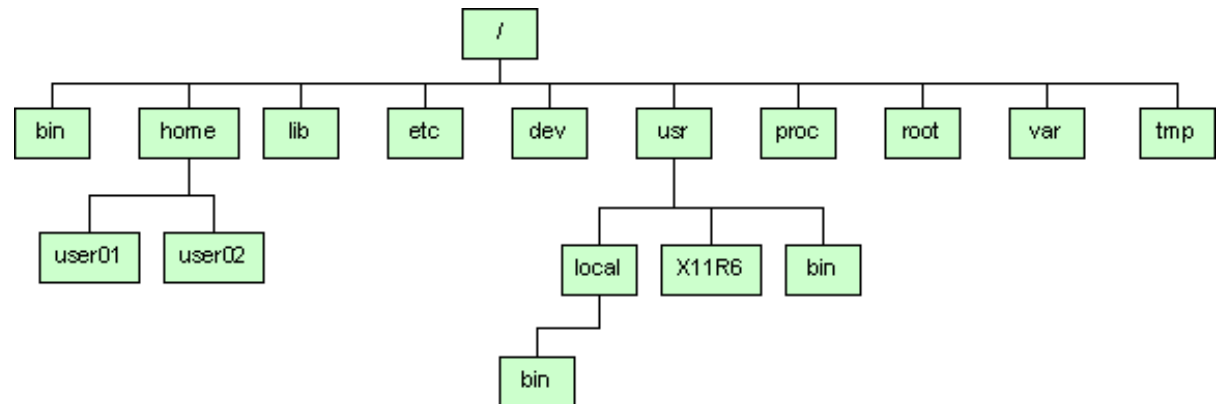
Normal and Super users

- One special user for administration who can do anything, called root or super user.
- Others are normal accounts

Introduction to Linux

Directory Structure

- Directory (like folder in Windows) are organized in a hierarchical tree structure.
- Top directory is “/” (slash or root).
- Users have the own directory (home directory).
- User and system directories.



Basic Commands

ls	List file and directory
cd	Change directory
cp	Copy file or directory
mv	Move file or directory
rm	Remove file or directory
pwd	Display current directory
mkdir	Create directory
rmdir	Remove directory
less	Display contents of text file, press 'q' to exit.
man	Display online manual
history	Display your command history

Basic Commands (Cont.)

How to use Linux commands?

When you log on a Linux computer, you will see,
`[rajak@hpc-login3 ~]$`

This is called **command prompt** and it means that Linux is ready for your inputs.

A Linux command usually consists of three parts, **command name**, **options**, and **arguments**.

Example) `[rajak~]$ ls -l .bashrc`

Basic Commands (Cont.)

- Whitespace is necessary between command name, options and arguments.
- Options start with “-”

Example)

```
mkdir mydir
```

```
rm test.txt
```

```
cp . bashrc test.txt
```

```
ls -l .bashrc
```


Basic Commands (Exercise)

1. Type following commands and observe difference in their outputs.

`ls` `ls -la`

`ls -a` `ls -Fa`

2. Make a directory and change current position.

`mkdir testdir`

`pwd`

`cd testdir`

`pwd`

`cd`

`pwd`

`rmdir testdir`

Basic Commands (Exercise)

3. In your home directory,

```
ls .bash_profile
```

```
cp .bash_profile sample.txt
```

```
less sample.txt (press “q” key to exit)
```

```
mv sample.txt sample2.txt
```

```
rm sample2.txt
```

4. Check disk usage

```
df
```

```
df -h
```

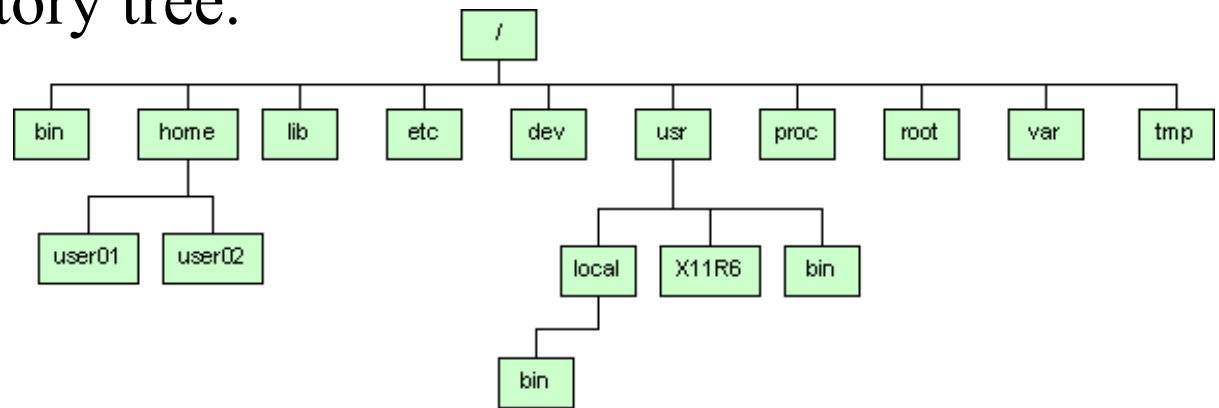
```
du
```

```
du -h
```

```
du -sh
```


Relative & Absolute Path

- **path** is a location in the directory tree.
- To specify a path, you can use two different expressions, **relative** or **absolute** path.
- With a **relative path**, a location is specified relative to your current location.
- With an **absolute path**, a location is specified from the top of the directory tree.



Relative & Absolute Path (Cont.)

- In relative path,
 - . current directory
 - .. parent directory
- Example)
cd example1
pwd
cd /auto/hpc-23/rajak/example2
- Absolute path always starts with “/”
- Example)
cd /Users/rajak
/Users/rajak/file.txt

Relative & Absolute Path (Cont.)

Relative path example

In home directory, type following command to observe how output of `pwd` command changes.

```
pwd
cd .
pwd
cd ..
pwd
cd ..
pwd
cd
```

Absolute path example

From your home directory, try to move two directories up using absolute path.

Example)

```
pwd
cd /Users/rajak
pwd
cd /Users
pwd
```


Redirect, Append and Pipe

Redirect and append

- Output from command is usually 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 other command.

Redirect, Append and Pipe

`head` shows first several lines.

`tail` shows last several lines.

`grep` shows lines matching a given pattern.

Redirect, Append and Pipe (Exercise)

1. In your home directory, copy `.bash_profile` to `sample.txt` for this exercise.

```
ls .bash_profile
```

```
cp .bash_profile sample.txt
```

```
less sample.txt
```

2. Redirect example

```
head -n 3 sample.txt
```

```
head -n 3 sample.txt > redirect.txt
```


Redirect, Append and Pipe (Exercise)

3. Append example

```
tail -n 3 sample.txt
```

```
tail -n 3 sample.txt >> redirect.txt
```

```
less redirect.txt
```

4. Pipe example

```
less redirect.txt
```

```
grep PATH redirect.txt
```

```
tail redirect.txt | grep PATH
```

```
rm sample.txt
```

```
rm redirect.txt
```


Text Editor

- Files that store input parameters are in **text** format.
- You **must know how to edit text files** on Linux.
- Two popular text editors, **vi and Emacs**, are widely used, though they are **too complicated** for Linux beginners.
- **nano** is an easy and simple to use, **good for new users**.

Text Editor (Cont.)

Arrow-keys	Move cursor
Enter	Change line
Delete	Delete a character
CTRL+x	Save data and exit nano

Open a new file to edit with nano.

```
> nano hello.txt
```

You will see screen like on the right



```
GNU nano 2.3.1      File: hello.txt
[ New File ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

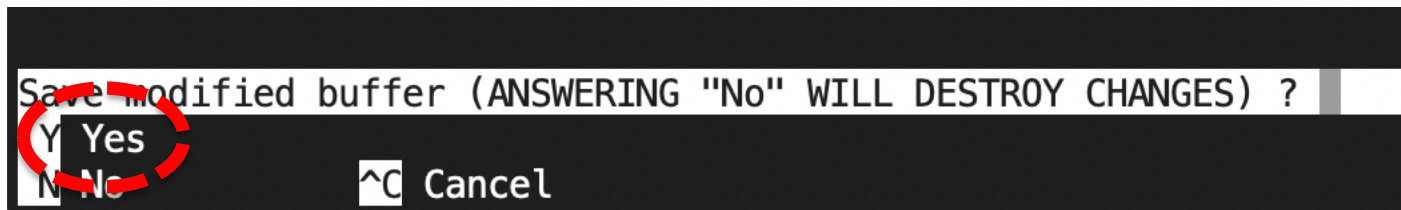

Text Editor (Cont.)

Step 1: Type “Hello!”



```
GNU nano 2.3.1
Hello!
```

Step 2: Save & Close the file by pressing **CTRL** and **x** keys together. Press **y** to save your edit.



```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes
N No      ^C Cancel
```

Step 3: nano will ask filename to be written. Press **Enter** key.



```
File Name to Write: hello.txt
^G Get Help      M-D DOS Format    M-A Append      M-B Ba
^C Cancel        M-M Mac Format    M-P Prepend
```


Text Editor (Cont.)

CTRL+w	Search
CTRL+d	Delete a character
CTRL+k	Remove a line
CTRL+u	Paste text in buffer
CTRL+a	Move to the beginning of line
CTRL+e	Move to the end of line
CTRL+v	Move forward one page
CTRL+y	Move backward one page

Text Editor (Cont.)

Write a “hello world” program, compile and runt it.

1. Open hello.f90 with nano

```
> nano hello.f90
```

2. Type in following Fortran code, save your edit and close the file.

```
Program Hello  
    write(*,*) "Hello World"  
stop  
end
```

3. Compile the code.

```
> gfortran hello.f90
```

4. Run it.

```
> ./a.out
```


Compiling & Running Programs

You need to **compile your source codes** (text file) into an **executable** (binary file) so that computer can understand how to execute it.

`gcc/icc` : C compilers

```
> gcc main.c
```

`gfortran/ifort` : Fortran compiler

```
> ifort main.f90
```

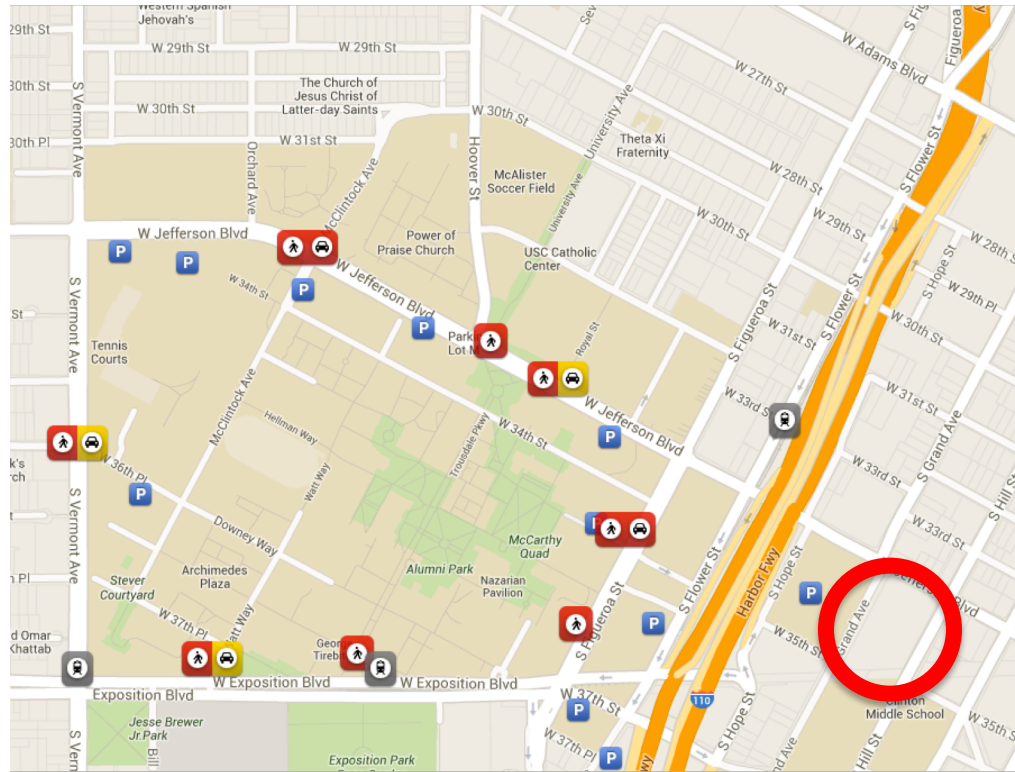

Compiling & Running Programs

If the compilation is successful done, you will see a file called **a.out**, which is default user-compiled executable filename.

To run the program, type

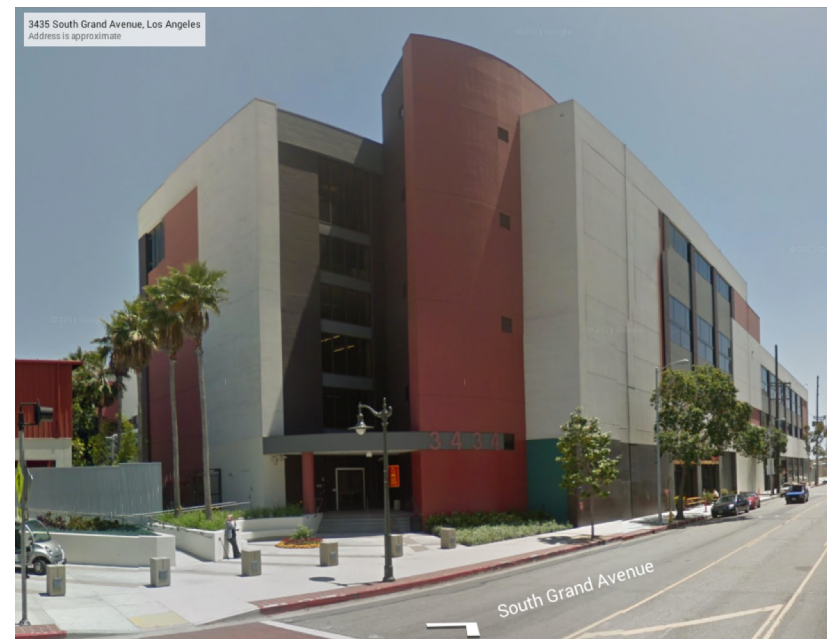
```
> ./a.out  
hello!
```


USC High-Performance Computing (HPC) Center



University Park Campus

Carol Little (CAL) building



Simple Linux Utility for Resource Management (SLURM)

sbatch - submit SLURM job to HPC cluster

--ntasks=1 Number of tasks you have
--time=60 how long (in minutes) you want
 to run your job
--output=md.out Output file
--job-name=T0.3 job name

Job Monitor/Cancel : **squeue & scancel**

squeue - show status of SLURM batch jobs

- a** all jobs are displayed
- u *username*** display status of specific user's job
- l** display job information in detail

> sbatch myjob.sh

Submitted batch job 1829927

> squeue -u knomura

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
1829927	quick	myjob.sh	knomura	R	0:03	1 hpc1411	

Job Monitor/Cancel : **squeue & scancel**

scancel - delete running job

scancel *jobid* delete a job

> **scancel 1829927**

> **squeue -u knomura**

JOBID	PARTITION	NAME	USER	ST	TIME	NODES
NODELIST(REASON)						