

# COMP4651

## Lab0: Linux fundamentals

---

Wei Wang  
CSE@HKUST  
Fall 2021



THE DEPARTMENT OF  
**COMPUTER SCIENCE & ENGINEERING**  
計算機科學及工程學系

# Outline

---

- ▶ SSH connection (p3 - p13)
- ▶ Basic Linux commands (p14 - p46)
- ▶ Using text editor (p47 - p53)
- ▶ (Appendix for Windows Users) (p54 - p55)

# Why Linux?

---

- ▶ Almost all open-source big data systems, such as Hadoop and Spark, run in Linux/Unix OS
- ▶ We'll do labs on Amazon EC2, using Linux virtual machines

# Connecting to a Linux machine

---

- ▶ CSE Lab 2 (csl2) has 53 workstations running CentOS
  - ▶ reachable at [csl2wkXY.cse.ust.hk](http://csl2wkXY.cse.ust.hk), where XY is the workstation ID (01..53)
- ▶ Log onto workstation 01 from your laptop (Lab3 Windows machines also have an open-ssh client)
  - ▶ in the command line, type:  
  
`ssh your_CSD_Account_ID@csl2wk01.cse.ust.hk`
  - ▶ type your CSD password, and type “yes” if it asks you “The authenticity of the host can’t be established...”

# Secure SHell: SSH

---

- ▶ OpenSSH/Sun\_SSH
  - ▶ OpenSSH is included in a number of Linux distributions
  - ▶ also available in the Lab3 Windows machines
  - ▶ we will use it to connect to our VMs in the AWS cloud!
  - ▶ type `ssh -V` at the command line to check its version

# Secure SHell: SSH

---

- ▶ Basic usage
  - ▶ `ssh user@hostname`
- ▶ TCP connection at default port 22

```
[csl2wk01:weiwa:31> ssh
usage: ssh [-1246AaCfGKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
          [-D [bind_address:]port] [-E log_file] [-e escape_char]
          [-F configfile] [-I pkcs11] [-i identity_file]
          [-L [bind_address:]port:host:hostport] [-l login_name] [-m mac_spec]
          [-O ctl_cmd] [-o option] [-p port]
          [-Q cipher | cipher-auth | mac | kex | key]
          [-R [bind_address:]port:host:hostport] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] [user@]hostname [command]
```

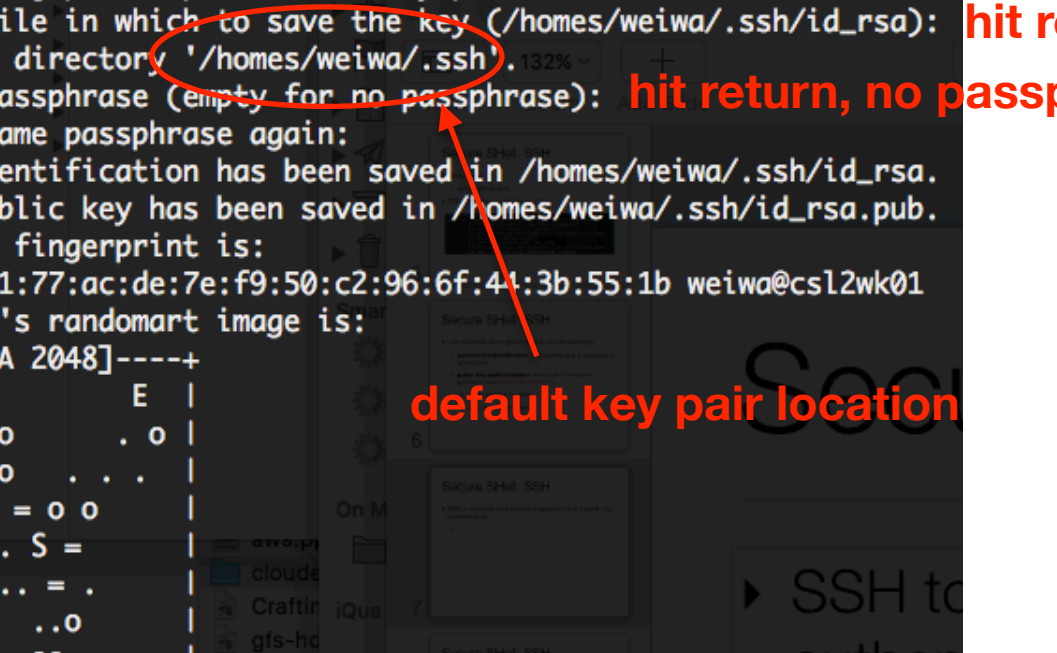
# Secure SHell: SSH

---

- ▶ User authentication: give access to an intended host
  - ▶ **password authentication:** uses username and password to authenticate
  - ▶ **public key authentication:** uses a pair of computer generated keys: **public and private key**
    - ▶ private key kept on client
    - ▶ public key stored on server

# SSH without a password

- ▶ Log onto a remote host through a public key authentication
- ▶ Generate a public-private key pair: `ssh-keygen`



A terminal window showing the execution of the `ssh-keygen` command. The output is as follows:

```

csl2wk01:weiwa:32> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/homes/weiwa/.ssh/id_rsa):
Created directory '/homes/weiwa/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /homes/weiwa/.ssh/id_rsa.
Your public key has been saved in /homes/weiwa/.ssh/id_rsa.pub.
The key fingerprint is:
b0:81:81:77:ac:de:7e:f9:50:c2:96:6f:44:3b:55:1b weiwa@csl2wk01
The key's randomart image is:
+---[ RSA 2048]-----+
...                E
. .00      .  o
..00      .  .
. = o o
. . S =
Tags      . . = .
. . .o
. .00
. .
+-----+

```

Annotations on the image:

- A red circle highlights the default path `/homes/weiwa/.ssh/` in the prompt "Enter file in which to save the key".
- A red arrow points from the text "default key pair location" to the circled path.
- Red text "hit return" is placed next to the first prompt.
- Red text "hit return, no passphrase" is placed next to the second and third prompts.




# SSH without a password

---

- ▶ Let's take a look into “~/.ssh”:
  - ▶ `ls ~/.ssh`
  - ▶ private key: `id_rsa`; public key: `id_rsa.pub`
- ▶ Now add the public key to the authorized keys
  - ▶ `touch ~/.ssh/authorized_keys`
  - ▶ `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
  - ▶ Prevent the file from being accessed by others:
    - ▶ `chmod 600 authorized_keys`

# SSH without a password

---

- ▶ Log out of the workstation and go back to the laptop: `logout`
- ▶ On your laptop (**Windows users refer to the Appendix**)
  - ▶ Get the private key by Secure Copy (`scp`) and rename it as “csl2wk01.rsa” (or whatever file name you prefer):
    - ▶ `scp you@csl2wk01.cse.ust.hk:~/.ssh/id_rsa ~/.ssh/csl2wk01.rsa`
  - ▶ Prevent others to accessing the private key:
    - ▶ `chmod 600 ~/.ssh/csl2wk01.rsa`
  - ▶ Log onto the workstation through public key authentication:
    - ▶ `ssh(-i)~/.ssh/csl2wk01.rsa you@csl2wk01.cse.ust.hk`  
 *specify an identity file*

# SSH without a password

---

- ▶ Anyone with the private key **csl2wk01.rsa** can SSH to the workstation under your identity! Remove the private key **on the workstation (csl2wkxx)**
  - ▶ `rm ~/.ssh/id_rsa`
- ▶ One more thing to do **on your laptop**
  - ▶ `echo -e "Host csl2wk01 \nHostname csl2wk01.cse.ust.hk \nUser your_login_name \nIdentityFile ~/.ssh/csl2wk01.rsa" >> ~/.ssh/config`
  - ▶ `ssh csl2wk01`
    - ▶ (If error, check the “config” file by “`cat ~/.ssh/config`”, you can also revise it by “`vim ~/.ssh/config`”, seeing “Using text editor” part for reference)

# SSH known-host mechanism

- ▶ ssh is warning you that it doesn't really know the remote host yet when you ssh to a host you never talked to before. Type "yes", and add it to the known hosts
  - ▶ take a look into ~/.ssh/known\_hosts

```
peng@peng-QTC6:~/Download$ ssh -i temp3.pem ubuntu@54.199.159.63
The authenticity of host '54.199.159.63 (54.199.159.63)' can't be established.
ECDSA key fingerprint is b1:20:b2:a4:9c:24:80:ef:32:68:ba:03:bc:92:ac:8a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.199.159.63' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

A detailed explanation: <http://security.stackexchange.com/questions/20706/what-is-the-difference-between-authorized-keys-and-known-hosts-file-for-ssh>

# Terminating an SSH session

---

- ▶ Several ways depending on your Linux system
  - ▶ type `exit`
  - ▶ Or type `logout`
  - ▶ Or hit "`Ctrl + d`" on the keyboard

The end of SSH Connection Session.  
Any questions?

# Basic Linux commands

- ▶ Login to the Linux system again and play with it.
  - ▶ `ssh csl2wk01`

```
peng@peng-QTC6:~/linux_tutorial$
```

The host

Current directory  
("path")

The "Prompt"

# Basic Commands: man

- Help

Whenever you need help with a command, type "man" and the command name.

- Basic Syntax

man [option(s)] keyword(s)

hit "j" ("k") to scroll down (up) by one line;

hit "f" ("b") to scroll down (up) by one screen;

hit "q" to exit

Of course, you can always use Google...

# Introduction to Command: man

```
MAN(1)                                Manual pager utils                                MAN(1)

NAME
    man - an interface to the on-line reference manuals

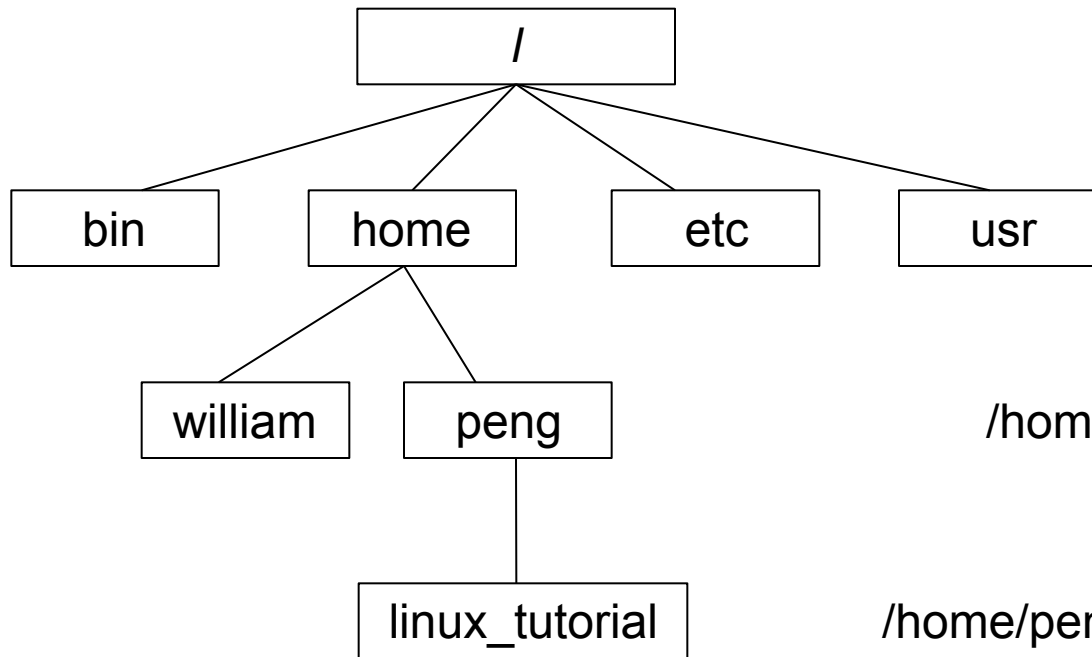
SYNOPSIS
    man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
    locale] [-m system[,...]] [-M path] [-S list] [-e extension] [-i|-I]
    [--regex|--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P
    pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
    cation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
    [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
    man -f [whatis options] page ...
    man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
    locale] [-P pager] [-r prompt] [-7] [-E encoding] [-p string] [-t]
    [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
    man -w|-W [-C file] [-d] [-D] page ...
    man -c [-C file] [-d] [-D] page ...
    man [-hV]

DESCRIPTION
    Manual page man(1) line 1 (press h for help or q to quit)
```



# Linux File System

a **hierarchy** of directories and files



/home/peng

/home/peng/linux\_tutorial

The "path"

# Linux File System

- Directory Commands
  - pwd: print name of the current/working directory
  - cd: change to a specific directory
  - ls: list directory contents
  - mkdir: create a directory
  - rmdir: delete a directory
- File Commands
- File Permissions (reference only)

# Linux File System

- Command: pwd

```
peng@peng-QTC6:~/linux_tutorial$ pwd  
/home/peng/linux_tutorial  
peng@peng-QTC6:~/linux_tutorial$ █
```

# Linux File System

- Directory Commands
  - pwd: print the complete path of current/working directory
  - cd: change to a specific directory

# Linux File System

- Command: `cd ~`, or simply `cd`

```
peng@peng-QTC6:~/linux_tutorial$ pwd
/home/peng/linux_tutorial
peng@peng-QTC6:~/linux_tutorial$ cd ~
peng@peng-QTC6:~$ pwd
/home/peng
peng@peng-QTC6:~$ █
```

“~” is the location of your home directory

# Linux File System

- Command: `cd ..` , goes back to the parent directory

```
peng@peng-QTC6:~/linux_tutorial$ pwd
/home/peng/linux_tutorial
peng@peng-QTC6:~/linux_tutorial$ cd ~
peng@peng-QTC6:~$ pwd
/home/peng
peng@peng-QTC6:~$ cd .. ←
peng@peng-QTC6:/home$ pwd
/home
peng@peng-QTC6:/home$
```

“~” is the location of your home directory

# Linux File System

- Command: cd

```
peng@peng-QTC6:~/linux_tutorial$ pwd
/home/peng/linux_tutorial
peng@peng-QTC6:~/linux_tutorial$ cd ~
peng@peng-QTC6:~$ pwd
/home/peng
peng@peng-QTC6:~$ cd ..
peng@peng-QTC6:/home$ pwd
/home
peng@peng-QTC6:/home$ cd /home/peng/linux_tutorial/
peng@peng-QTC6:~/linux_tutorial$ pwd
/home/peng/linux_tutorial
peng@peng-QTC6:~/linux_tutorial$ █
```

cd a particular directory

# Linux File System

- Directory Commands
  - pwd: print name of current/working directory
  - cd: change to a specific directory
  - ls: list directory contents



# Linux File System

- Command: ls
- Options:
  - -l long list (displays lots of info)
  - -a show hidden files (system files, etc.)
  - -t sort by modification time
  - -S sort by size
  - -h list file sizes in human readable format

# Linux File System

- Command: ls -ltr

```
peng@peng-QTC6:/home$ ls -ltr
total 960
-rw-r--r--  1 root root 978770 Nov 10 22:31 lecture18.pptx
drwxr-xr-x 36 peng peng  4096 Jan 12 13:23 peng
peng@peng-QTC6:/home$
```

List files by time with long listing

# Linux File System

- Directory Commands
  - pwd: print name of current/working directory
  - cd: change to a specific directory
  - ls: list directory contents
  - **mkdir: create a directory**

# Linux File System

- Command: mkdir. Creates a new folder in the current directory

```
peng@peng-QTC6:~/linux_tutorial$ ls
copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$ mkdir new_directory
peng@peng-QTC6:~/linux_tutorial$ ls
copy.txt  download.txt  new_directory
peng@peng-QTC6:~/linux_tutorial$
```

# Linux File System

- Directory Commands
  - pwd: print name of current/working directory
  - cd: change to a specific directory
  - ls: list directory contents
  - mkdir: create a directory
  - rmdir : delete an empty directory
  - rm -r: delete a directory even though it's not empty (use with caution)
- File Commands
- File Permissions

# Linux File System

- Command: rmdir

```
peng@peng-QTC6:~/linux_tutorial$ ls
copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$ mkdir new_directory
peng@peng-QTC6:~/linux_tutorial$ ls
copy.txt  download.txt  new_directory
peng@peng-QTC6:~/linux_tutorial$ rmdir new_directory
peng@peng-QTC6:~/linux_tutorial$ ls
copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$
```

# Linux File System

- Directory Commands
- File Commands
  - cp: copy files and directories
  - mv: move (rename) files
  - rm: remove a file

# Linux File System

- Command: cp

```
peng@peng-QTC6:~/linux_tutorial$ ls
copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$ cp copy.txt copy1.txt
peng@peng-QTC6:~/linux_tutorial$ ls
copy1.txt copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$
```



# Linux File System

- Directory Commands
- File Commands
  - cp: copy files and directories
  - mv: move (rename) files

# Linux File System

- Command: mv

```
peng@peng-QTC6:~/linux_tutorial$ ls
copy1.txt  copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$ mkdir new_directory
peng@peng-QTC6:~/linux_tutorial$ mv copy1.txt ./new_directory/
peng@peng-QTC6:~/linux_tutorial$ cd new_directory/
peng@peng-QTC6:~/linux_tutorial/new_directory$ ls
copy1.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$
```

move file to a different directory

# Linux File System

- Command: mv

```
peng@peng-QTC6:~/linux_tutorial$ ls
copy1.txt  copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$ mkdir new_directory
peng@peng-QTC6:~/linux_tutorial$ mv copy1.txt ./new_directory/
peng@peng-QTC6:~/linux_tutorial$ cd new_directory/
peng@peng-QTC6:~/linux_tutorial/new_directory$ ls
copy1.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$ mv copy1.txt copy2.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$ ls
copy2.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$ █
```

↑  
rename a file

# Linux File System

- Directory Commands
- File Commands
  - cp: copy files and directories
  - mv: move (rename) files
  - rm: remove a file

# Linux File System

- Command: rm

```
peng@peng-QTC6:~/linux_tutorial$ ls
copy1.txt  copy.txt  download.txt
peng@peng-QTC6:~/linux_tutorial$ mkdir new_directory
peng@peng-QTC6:~/linux_tutorial$ mv copy1.txt ./new_directory/
peng@peng-QTC6:~/linux_tutorial$ cd new_directory/
peng@peng-QTC6:~/linux_tutorial/new_directory$ ls
copy1.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$ mv copy1.txt copy2.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$ ls
copy2.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$ rm copy2.txt
peng@peng-QTC6:~/linux_tutorial/new_directory$ ls
peng@peng-QTC6:~/linux_tutorial/new_directory$
```

# Linux File System

- Directory Commands
- File Commands
- **File Permissions** (for reference)
  - permission level
  - chmod

# Linux File System

- File Permissions
  - Each file in Unix/Linux has an associated permission level
  - This allows the user to prevent others from reading/writing/executing their files or directories
  - Use “ls -l *filename*” to find the permission level of that file

# Linux File System

- Directory Commands
- File Commands
- File Permissions
  - permission level
  - Command; chmod



# Linux File System

- Permission Level
  - “r” means “read” permission
  - “w” means “write” permission
  - “x” means “execute” permission

# Linux File System

```
peng@peng-QTC6:/home$ ls -l
total 960
-rw-r--r--  1 root root 978770 Nov 10 22:31 lecture18.pptx
drwxr-xr-x 36 peng peng  4096 Jan 12 13:32 peng
peng@peng-QTC6:/home$
```

d

character 1: type of file  
"d" means directory  
"-" means ordinary

# Linux File System

```
peng@peng-QTC6:/home$ ls -l
total 960
-rw-r--r--  1 root root 978770 Nov 10 22:31 lecture18.pptx
drwxr-xr-x 36 peng peng  4096 Jan 12 13:32 peng
peng@peng-QTC6:/home$
```

rwx

Characters 2-4 show  
owner permissions

# Linux File System

```
peng@peng-QTC6:/home$ ls -l
total 960
-rw-r--r--  1 root root 978770 Nov 10 22:31 lecture18.pptx
drwxr-xr-x 36 peng peng  4096 Jan 12 13:32 peng
peng@peng-QTC6:/home$
```

r-x

Characters 5-7 show  
group permissions

# Linux File System

```
peng@peng-QTC6:/home$ ls -l
total 960
-rw-r--r--  1 root root 978770 Nov 10 22:31 lecture18.pptx
drwxr-xr-x 36 peng peng  4096 Jan 12 13:32 peng
peng@peng-QTC6:/home$
```

r-x

Characters 8-10 show  
permissions for all other  
users

# Linux File System

- Command: chmod
  - change file permissions with “chmod”
  - Syntax: chmod [**user/group/others/all**]+[permission] [file(s)]
  - Numerical Permission:  
4 read (r), 2 write (w), 1 execute (x), and 0 non (-)

# Text Editor: VIM

- Basic usage

- `vim <filename>`

- it will create a new file if no such file exists

- Basic operations in command mode
- Enter the Insert mode
- Return to command mode

# Text Editor: VIM

- Basic usage  
*vim* <filename>

```
peng@peng-QTC6:~$ vim gen_host_list.py
peng@peng-QTC6:~$ █
```



# Text Editor: VIM

```
#!/usr/bin/env python
import sys
import math

def switch_count(depth, fanout):
    num_switches = 0
    for i in range(0, depth):
        num_switches += int(math.pow(fanout, i))
    return num_switches

# This method only works for binary trees (i.e., fanout = 2)
def find_switch(host):
    switch = depth + num_hosts
    for d in range(depth-1, 0, -1):
        n = int(math.pow(fanout, d))
        if (host > n):
            switch += switch_count(d, fanout)
            host += -n
    return switch

def find_port(host):
    port = (host % fanout)

-- INSERT --
```

1,1

Top

# Text Editor: VIM

- Basic usage
- **Basic operations in command mode**
  - dw: delete to end of current word
  - dd: delete (and copy) current line
  - r: replace current character
  - p: puts the previously deleted text
  - .: repeat the last command
- Enter the insert mode
- Return to command mode

# Text Editor: VIM

- Basic usage
- Basic operations in command mode
- Enter the editing mode
  - A: append any typed characters to the end of a line
  - i: enter inset mode and inset characters as you type
  - R: enter replace mode and replace characters with the ones being typed
- Return to command mode

# Text Editor: VIM

- Basic usage
- Basic operations in command mode
- Enter the editing mode
- **Return to command mode**
  - <press ESC if you are in the editing mode>
  - :q – leave vim
  - :q! – leave without saving
  - :wq – save and leave
  - :w – save without leave
  - :w <filename> – save in a new name.

# Other text editors

---

- ▶ Besides Vim, you can also use Emacs, Nano, etc., there is no constraint on which to use.

# Windows users with cmd.exe

---

- ▶ **On your own PC laptop.**

- ▶ Get the private key:

```
scp you@csl2wk01.cse.ust.hk:.ssh/id_rsa .ssh/csl2wk01.rsa
```

- ▶ Check whether you can successfully login (please logout afterwards)

```
ssh -i .ssh/csl2wk01.rsa you@csl2wk01.cse.ust.hk
```

- ▶ Configure (type four commands in the following order)

```
echo Host csl2wk01 >> .ssh/config
```

```
echo Hostname csl2wk01.cse.ust.hk >> .ssh/config
```

```
echo User your_login_name >> .ssh/config
```

```
echo IdentityFile %HOMEPATH%/.ssh/csl2wk01.rsa >> .ssh/config
```

- ▶ Now you can log in with short command: `ssh csl2wk01`

# Windows users with cmd.exe

```

Select C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.254]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\qwengaa>ssh qwengaa@csl2wk01.cse.ust.hk
The authenticity of host 'csl2wk01.cse.ust.hk (143.89.238.1)' can't be established.
ECDSA key fingerprint is SHA256:50EfIdmuPYQafzNbUBzEui3wBQmv2wUfehgpI6Cg5HQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'csl2wk01.cse.ust.hk,143.89.238.1' (ECDSA) to the list of known hosts.
Password:
Last login: Mon Sep 10 11:18:39 2018 from wf123-231.ust.hk

*****
*
* Note:
*
* All Lab 2 workstations (including this host) reboot at 7:00am
* everyday
*
*****

csl2wk01:qwengaa:101> ls .ssh
authorized_keys  id_rsa  id_rsa.pub
csl2wk01:qwengaa:102> logout
Connection to csl2wk01.cse.ust.hk closed.

C:\Users\qwengaa>scp qwengaa@csl2wk01.cse.ust.hk:~/.ssh/id_rsa .ssh/csl2wk01.rsa
Password:
id_rsa                                     100% 1675      1.6KB/s   00:00

C:\Users\qwengaa>echo Host csl2wk01 >> .ssh/config

C:\Users\qwengaa>echo Hostname csl2wk01.cse.ust.hk >> .ssh/config

C:\Users\qwengaa>echo User qwengaa >> .ssh/config

C:\Users\qwengaa>echo IdentityFile %HOMEPATH%/.ssh/csl2wk01.rsa >> .ssh/config

C:\Users\qwengaa>ssh csl2wk01
Last login: Mon Sep 10 12:14:20 2018 from 143.89.76.79
```

END



# Credit

---

- ▶ Some Slides adapted from Dr. Hong Xu's slides for CS 4296/5296 in CityU