COMP3511

Lab 01: Introduction to the Lab Environment

REN, Zhenghang (zrenak@cse.ust.hk)

Lab Tutorial

- Short introduction to Linux
- How to login your lab environment
- How to use your lab environment (with demo)

After this tutorial you should be able to:

- Login/logout lab server
- Interact with OS in lab server: change directory, list directory files, create files, edit files, save files, delete files……

Lab Environment

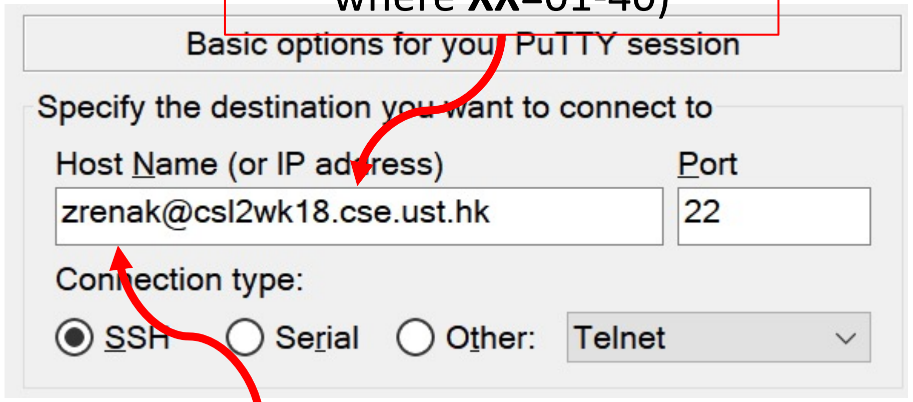
- Linux environment (contrast to Windows and MacOS you are familiar with)
- Accessed remotely via SSH in terminal

- Beware! Don’t store large files on your lab environment

Getting Started (in Windows)

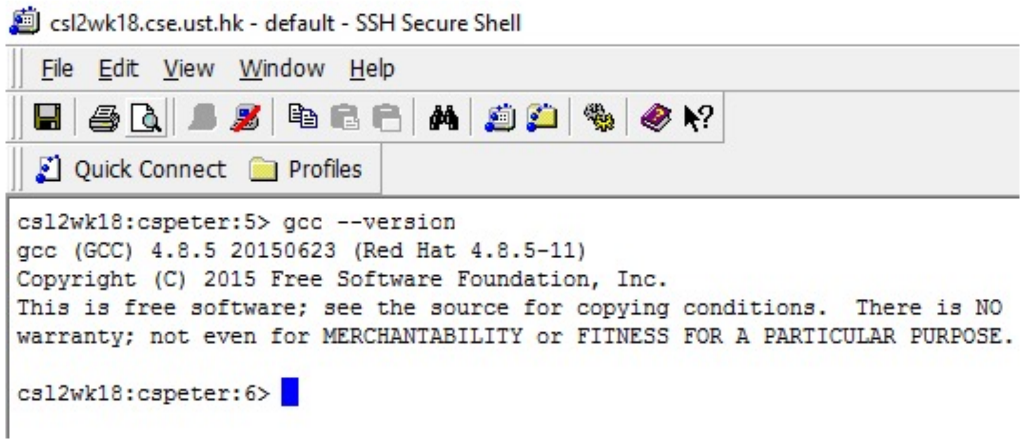
- Use `SSH (Secure SHell client)` or `Putty` (`https://www.putty.org`)
  - Host Name (address): `csl2wkXX.cse.ust.hk` (where `XX=01..40`)
  - ITSC username (e.g. `cspeter`)
  - Port Number: `22`
- Save config

Enter a machine name (csl2wk**XX**.cse.ust.hk, where **XX**=01-40)


Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
Connection type:
SSH
Serial
Other:
Telnet

Host Name (or IP address): zrenak@csl2wk18.cse.ust.hk
Port: 22
Connection type: [✓ SSH] Serial Other: Telnet

Enter your ITSC username

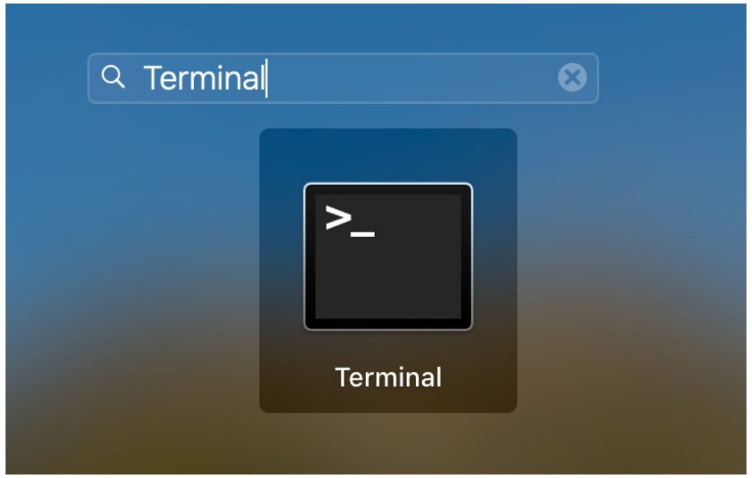


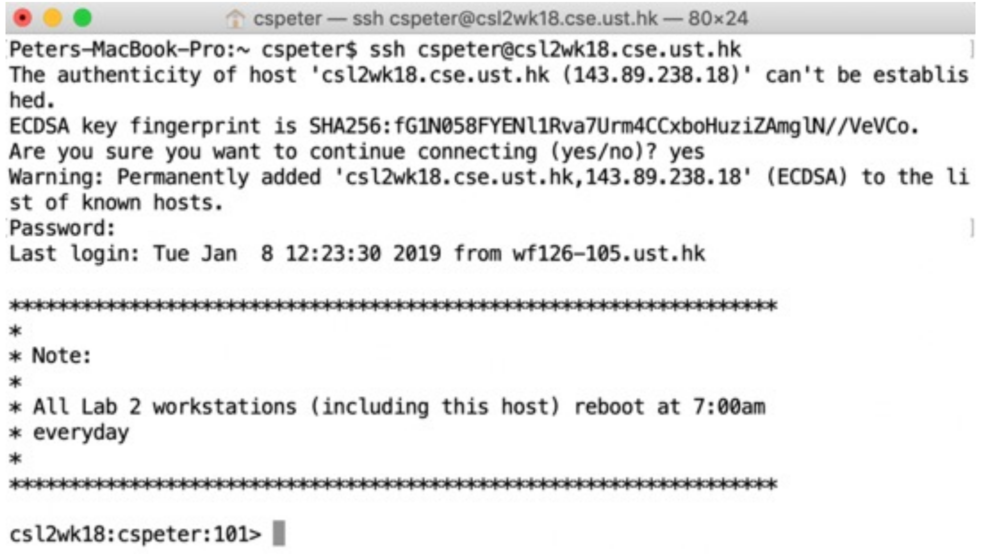
```
csl2wk18:cspeter:5> gcc --version
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-11)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

csl2wk18:cspeter:6>
```

Getting Started (in Mac/Linux)

- In Mac, open `Terminal` and then type in the `ssh` command
  - `ssh [Your ITSC username]@csl2wkXX.cse.ust.hk` (where `XX=01..40`)
- In Linux, there should be a similar terminal software (e.g. Konsole, GNOME terminal)





```
Peters-MacBook-Pro:~ cspeter$ ssh cspeter@csl2wk18.cse.ust.hk
The authenticity of host 'csl2wk18.cse.ust.hk (143.89.238.18)' can't be establis
hed.
ECDSA key fingerprint is SHA256:fG1N058FYENl1Rva7Urm4CCxboHuziZAmglN//VeVCo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'csl2wk18.cse.ust.hk,143.89.238.18' (ECDSA) to the li
st of known hosts.
Password:
Last login: Tue Jan  8 12:23:30 2019 from wf126-105.ust.hk

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

csl2wk18:cspeter:101> 
```

Try login

What is Linux?

- An UNIX-like operating system
- Open-source, easy to customize
- A popular choice of programmers
    - The shell, although difficult to learn at the beginning, has proven to be productive and convenient for programmers
    - Closer environment to the servers where applications are hosted on
- A popular choice for servers
    - Even Microsoft Azure is based on Linux

Why learn Linux?

- To manage a server for your application
- To use powerful tools like Kali Linux for cyber security studies
- To use cloud computing services
- To use containers (Docker, Kubernetes)

- For this course: Linux offers simple and uniform lab environment

The Terminal

- The piece of hardware that allows you to interact with the computer
  - The monitor
  - The keyboard
- A computer was connected to multiple terminals for sharing
  - Computers were expensive
- Now terminal refers to the Text UI program
  - A virtual terminal
  - Emulating text I/O of early terminals

Shell

- The piece of software that provides text interaction with the computer
  - Linux uses bash as the default shell
  - There are other shells, e.g. sh, csh, zsh, ……
  - csh is the default shell in Lab 2
  - You can customize the shell by editing `~/.cshrc_user`
- Users tell the computer what to do with commands
  - Shell commands are programs that do specific tasks
  - Commands may or may not give text feedback
  - Some commands accept arguments

Shell vs Terminal

- The **terminal** is the GUI window that you see on the screen. It takes commands and shows output. (input and output module)
- The **shell** is the software that interprets and executes the various commands that we type in the **terminal**.

Interaction with Linux OS

- Shell Commands

- Directory
    - List/Change directory
    - Create, Rename, Move, Remove
- File
    - Check file contents
    - Create, Rename, Edit, Move, Remove
- Getting Help

Shell Commands

- Example shell commands
  - `ls` lists files under a directory
  - `cd` changes the working directory to somewhere else
  - `pwd` shows current directory
- Clear the output: `clear`
- Example command arguments
  - Arguments can be a character or a word
  - Character arguments can stack together
  - `ls -a -l` or in short `ls -al`
    - 'a' shows hidden files, 'l' list detailed information of files
  - Each command is different, consult the manual or help
    - `man ls`
    - `ls --help`

Essential Shell Commands

`ls`	`cd`	`cat`
`rm`	`mkdir`	`rmdir`
`pwd`	`echo`	`whoami`
`less`	`more`	`man`
`info`	`touch`	`exit`

- Use the key <Tab> to auto-complete commands
- Use arrow keys (up and down) to find previously used commands
- Use argument `--help` on any command to show their usage
- Use `man (a command)` to show detailed command description
    - E.g. `man less`

Directory - Path

- An **absolute path** specifies the path of the file starting from the root (/) directory
- Relative path using dot(.) and dotdot (..)
  - . indicates the current directory
  - .. indicates the directory in the upper level
- Example:
  - `../test.txt`
  - It means the file text.txt located in the upper level

Related commands

- `pwd`
  - Print the absolution path of the current path
- `ls`
  - List out the content in the current working directory
  - Examples:
    - `ls –l`
      - List out the detailed information about the current directory
    - `ls –lh`
      - List out the detailed information, with a human readable format
    - `ls /home`
      - List out the content of the home directory

Related commands

- `cd`
  - Change directory
  - Examples:
    - `cd ..`
      - Change the current directory to the upper level
    - `cd .`
      - Change the current directory to the current directory
      - Nothing will happen
    - `cd /etc/init.d/`
      - Change the current directory to `/etc/init.d/`
    - `cd ~`
      - Change the current directory back to your **home directory**

Directory management

- Creating directories
  - `mkdir <arg1> <arg2> … <argn>`
- Renaming and moving directories
  - `mv <source> <destination>`
- Copying directories
  - `cp -r <source> <destination>`

Directory management

- Creating directories
  - `mkdir <arg1> <arg2> … <argn>`
- Renaming and moving directories
  - `mv <source> <destination>`
- Copying directories
  - `cp -r <source> <destination>`
- Remove directories
  - `rmdir <dir1> <dir2> … <dirN>`
  - All directories must be empty
  - If not empty: `rm –r <dir1> <dir2> … <dirN>`

File management

- Create an empty file or update its timestamp
  - `touch <file>`
  - It will be useful if you would like to change the last modified date of a file for some reasons
- Remove files
  - `rm <file1> <file2> … <fileN>`
- Remove ALL files recursively
  - `rm * -rf`
  - Don't do this if you are root user and in the root directory

View a text file

- There are 3 commands to view a text file
  - `cat <filename>`
  - `more <filename>`
  - `less <filename>`
- There are 2 commonly used command-line editors (nano and vim)
  - `nano <filename>`
  - `vim <filename>`

Command line editors (nano/vi)

- nano
  - Commands:
    - Arrow keys: Navigate the editor
    - Ctrl+X: exit nano
    - Ctrl+O: write output
    - Ctrl+K: (multiple times), each time it cut one line
    - Ctrl+U: Paste the copied lines from Ctrl+K
- vim
  - Commands:
    - ESC+i: Enter insert mode
    - ESC+dd: delete a line of text
    - ESC+4y: copy 4 lines
    - ESC+p: paste lines
    - ESC+wq!: exit and save
    - ESC+q!: exit but not save

Tutorials of using Nano/Vi

- There are many good online tutorials for nano/vi:
  - Nano: https://www.tecmint.com/learn-nano-text-editor-in-linux/
  - Vi: https://www.tutorialspoint.com/unix/unix-vi-editor.htm

Text Editing in Terminal with vim

- vim is a powerful tool for text editing in Unix
- To edit or create a file, use `vim <filename>`

- Now create a new file: `vim main.c`
  - When you first enter vim, you are in command mode, where every key serves as a command, not an input
  - vim have different modes: command mode, input mode, visual mode
  - <Esc> will lead vim back to command mode
- Press <i> to enter input mode, write the following Hello World for C

Text Editing in Terminal with VI

```
#include <stdio.h>
int main(){
    printf("Hello, World!\n");
    return 0;
}
```

- After you finished, press <Esc> to go back to command mode
- Press :w to save your file
- Press :q to quit VI
  - Alternatively, you can use :wq to finish both action
- Back in the Terminal,
  - Use `gcc main.c` to compile
  - Use `./a.out` to run the Hello World
  - In this example, `./` refers to the current working directory

Editing within vim

- When in command mode, these commands will enter input mode
  - <i> inserts at the current position
  - <a> inserts one character after current position
  - <o> inserts at a newline after current line
- Replace mode:
  - <r> replace the current character with the next you enter, will not change mode
  - <insert> enters replace mode

Editing within vim

- When in command mode,
  - <yy> copies one line
  - <dd> cuts one line
  - <10yy> copies 10 lines, <10dd> cuts 10 lines

Linux File System

- Hierarchical structure from the root directory /
- Your user folder (home) has alias of `~`
  - `cd` or `cd ~` go back to home directory

- Your shell records a working directory
  - The directory you are at right now
  - Your working directory is always home when you launch the shell
  - You can show your current working directory with `pwd`

Try something!

- Login your account.
- In your home folder:
  - Create directory named “intro”
  - Change to “intro” directory as working directory
  - Create a file named “helloworld” using vim or nano
  - Write the file with contents: “hello world”
  - Save the file and exit editor
  - View the contents of file using cat, more, or less
  - Remove the intro directory
- Logout

Thanks!