Lab 01: Linux Environment and C Programming – Compile/Run

CSCI3150 - Introduction to Operating Systems

WANG Zhiqi
zqwang@cse.cuhk.edu.hk

Linux Environment Installation

We provide you an image with the following configurations:

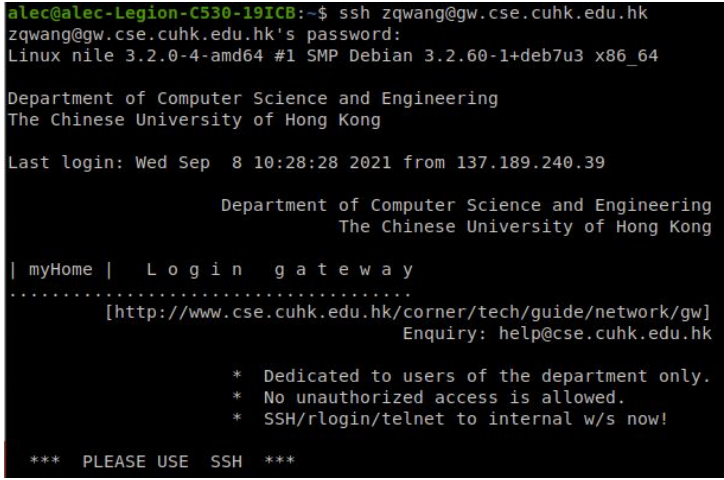
- OS: XUbuntu 18.04LTS (32 bit)
- CPU: 4
- Memory: 1GB
- Disk: 10 GB
- gcc: 7.4.0

- Please follow this link to install. Normally we will grade your assignments in this environment.
- Try to Google first when your see any error message.

Other Options to Access Linux

If you really cannot use VirtualBox in your computer:

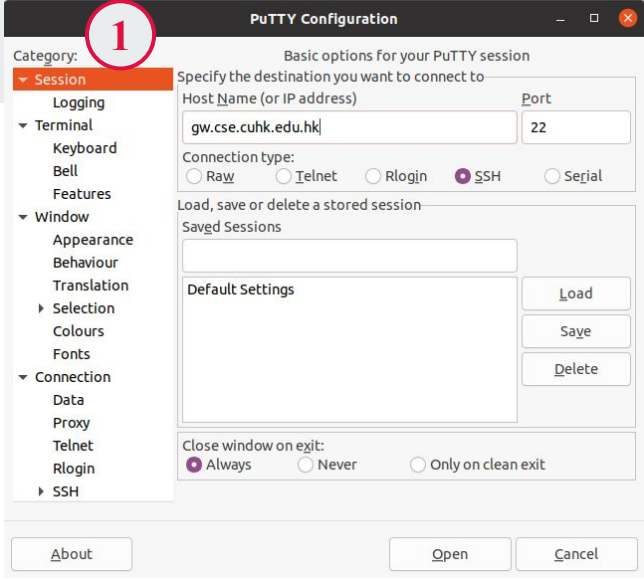
- **CSE Linux server**
  - Mac Users
    i. Open a terminal
    ii. Type "ssh YourUnixName@gw.cse.cuhk.edu.hk", and enter your unix password.
    iii. Type "ssh linux2" to connect to the Linux server.

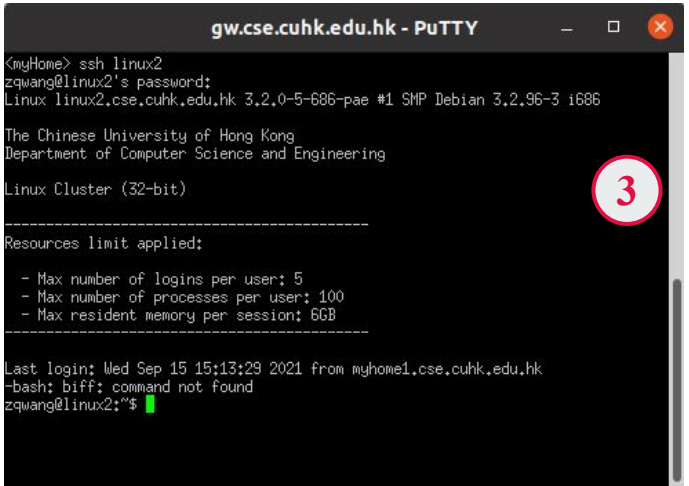


Other Options to Access Linux

If you really cannot use VirtualBox in your computer:

- **CSE Linux server**
  - Windows Users
    i. Install Putty
    ii. Follow the steps in the Figures.





Other Options to Access Linux

If you still cannot connect to CSE Linux server, other options may be:

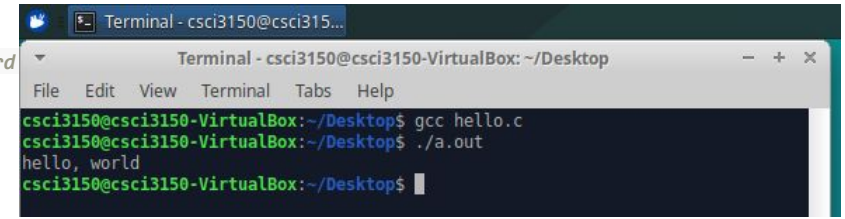
- Install a Linux distribution on your computer (Ubuntu, Debian, CentOS, Arch, etc.)
- Use Docker to access Linux (e.g. There are many materials on the Internet to access Ubuntu with Docker)
- ...

C Programing Review - Compile/Run

```c
/* header files go up here */
/* note that C comments are enclosed within a slash and a star, and may wrap
over lines */
// if you use gcc, two slashes will work too (and may be preferred)
#include <stdio.h>
/* main returns an integer */
int main(int argc, char *argv[])
{
        /* printf is our output function; by default, writes to standard
*/
        /* printf returns an integer, but we ignore that */
        printf("hello, world\n");
        /* return 0 to indicate all went well */
        return(0);
}
```

Compile the program:

prompt> gcc hello.c
prompt> ./a.out



Useful flags in gcc

```
gcc -o hw hello.c          # -o: to specify the executable name
gcc -Wall hello.c          # -Wall: gives much better warnings
gcc -g hello.c             # -g: to enable debugging with gdb
gcc -O hello.c             # -O: to turn on optimization
gcc –o hw -g -Wall hello.c # Combine these flags
```

Makefile tutorial

```
// hellomake.c

#include “hellomake.h”

int main() {
    // call a function in
another file
    myPrintHelloMake();
    return(0);
}
```

```
// hellofunc.c

#include <stdio.h>
#include <hellomake.h>

void myPrintHelloMake(void)
{
    printf("Hello
makefiles!\n");
    return;
}
```

```
// hellomake.h

void
myPrintHelloMake(void);
```

To compile them:

```
gcc -o hellomake hellomake.c hellofunc.c -I .
```

Makefile (first approach)

target: dependency1 dependency2 ...

<tab> command

```
hellomake: hellomake.c hellofunc.c
	gcc -o hellomake hellomake.c hellofunc.c -I .
```

Suppose we name it Makefile1, then we compile with it:
`make -f Makefile1`

```
csci3150@csci3150-VirtualBox:~/Desktop$ make -f Makefile1
gcc -o hellomake hellomake.c hellofunc.c -I .
csci3150@csci3150-VirtualBox:~/Desktop$ ./hellomake
Hello makefiles!
```

Makefile (second approach)

```
CC=gcc
CFLAGS=-I .

hellomake: hellomake.o hellofunc.o
	$(CC) -o hellomake hellomake.o hellofunc.o

clean:
	rm hellomake
```

Suppose we name it Makefile2, then we compile with it:
`make -f Makefile2`

```
csci3150@csci3150-VirtualBox:~/Desktop$ make -f Makefile2
gcc -I .   -c -o hellomake.o hellomake.c
gcc -I .   -c -o hellofunc.o hellofunc.c
gcc -o hellomake hellomake.o hellofunc.o
csci3150@csci3150-VirtualBox:~/Desktop$ ./hellomake
Hello makefiles!
```

Exercise (Deadline: 2021-09-23 23:59:59)

In the folder exercise, you can find a file ***main.c*** and two sub-folders: (1) foo; (2) bar. The ***main*** function in ***main.c*** will call the functions defined in foo/foo.c and bar/bar.c.

Under the folder exercise, there is a Makefile that will compile the main.c together with foo/foo.c and bar/bar.c and generate an executable file ***lab1***. You need to fill your code in Makefile.

Please only submit the ***Makefile*** to blackboard after you finish.