Assignment Arrangement & VM Setting

Assignment Schedule of 2023-24 Term2

Homework	Release Date	Due Date	Programming Assessment
Assignment 1	2024/02/01	2024/02/28	10%
Assignment 2	2024/02/29	2024/03/20	10%
Assignment 3	2024/03/21	2024/04/10	15%
Assignment 4	2024/04/11	2024/05/01	15%

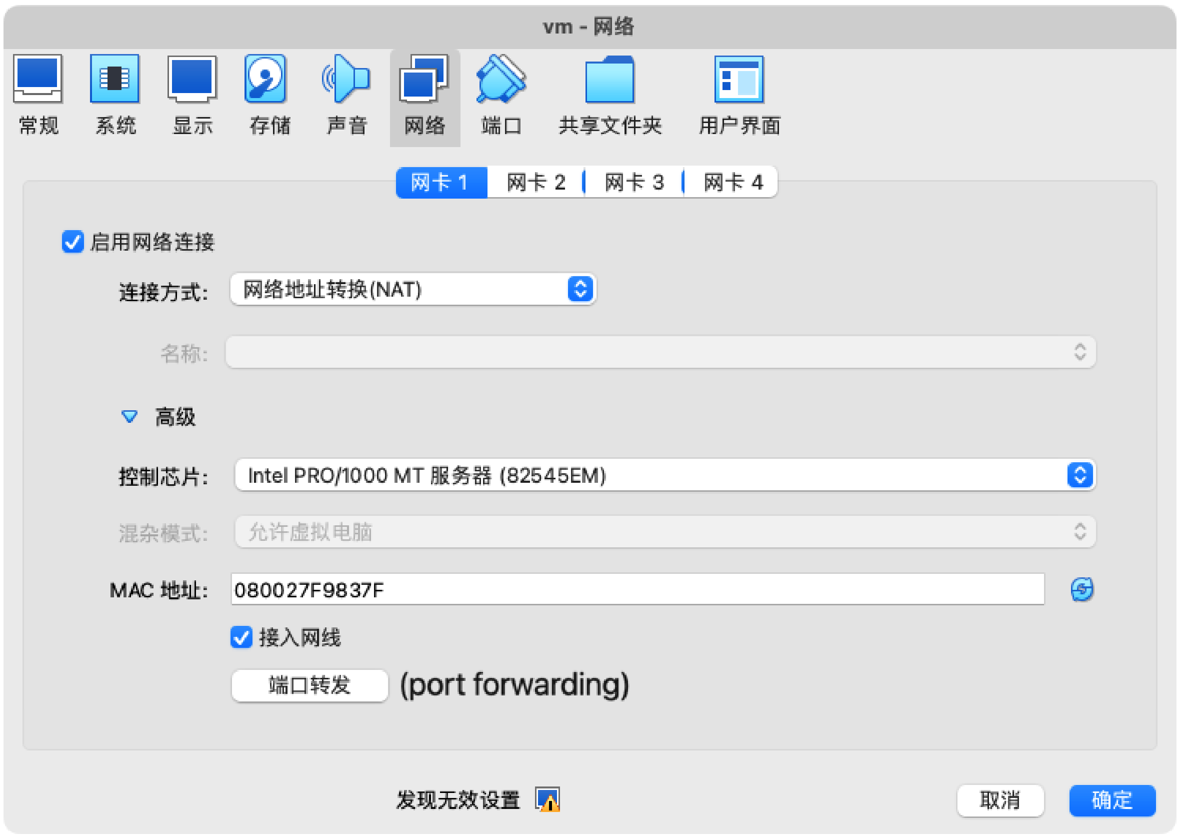
Virtual Machine Login and Usage Instruction

We provide the **CSC3150_a3_xv6.ova** file for x64 chip users (can be imported into VirtualBox or VMware) and the **CSC3150_a3_xv6.qcow2** file for Macbook m1/2 users (can be imported into UTM).

For UTM users (Macbook M1/M2), follow this quick tutorial to import the .qcow2 file. The default network setting (Shared Network) is good.

For VirtualBox users

(Normally, all versions would run well. We checked the 7.0 version), network configuration can be referred to as follows **(*updated on 11/02/2023*)**: Only one network, i.e., **Network Address Translation (NAT).** Set port forwarding configuration as **host port:2200, guest port:22.**



☑ 启用网络连接

连接方式: 网络地址转换(NAT)

名称:

▽ 高级

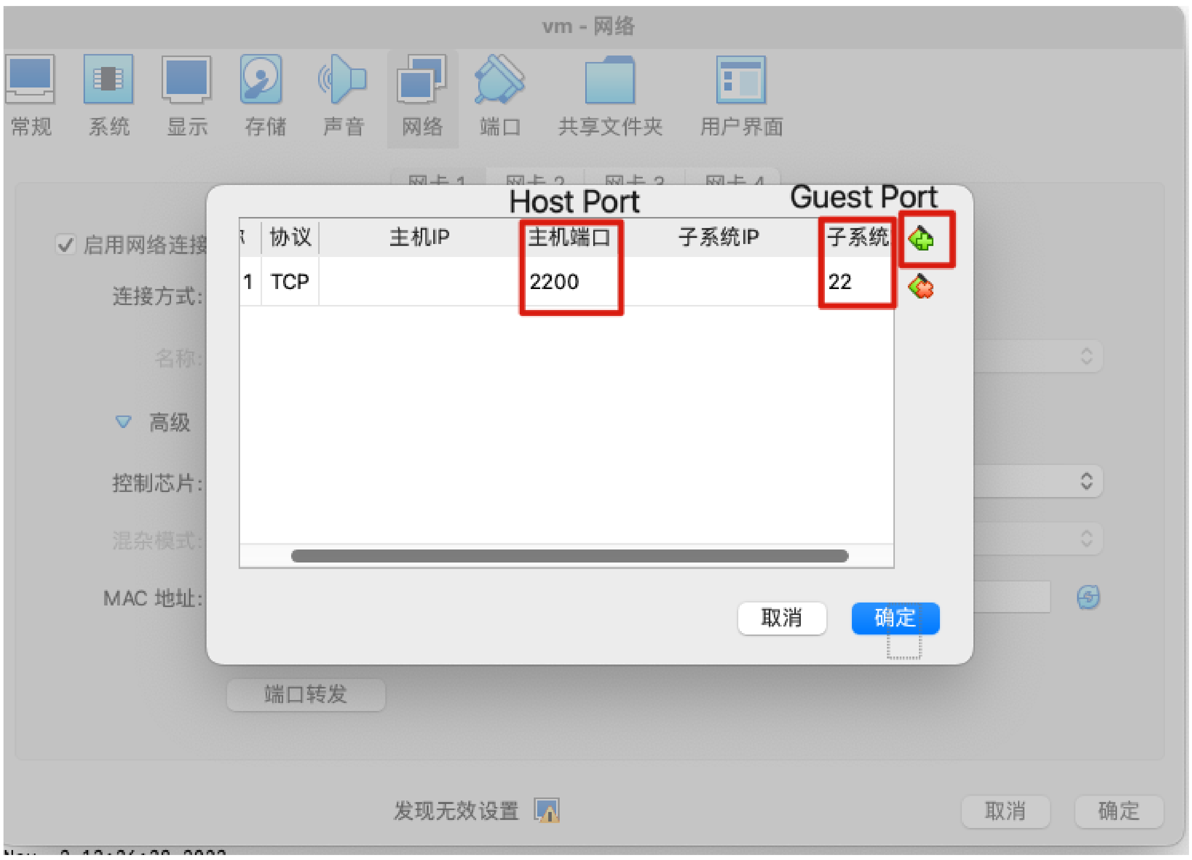
控制芯片: Intel PRO/1000 MT 服务器 (82545EM)

混杂模式: 允许虚拟电脑

MAC 地址: 080027F9837F

☑ 接入网线

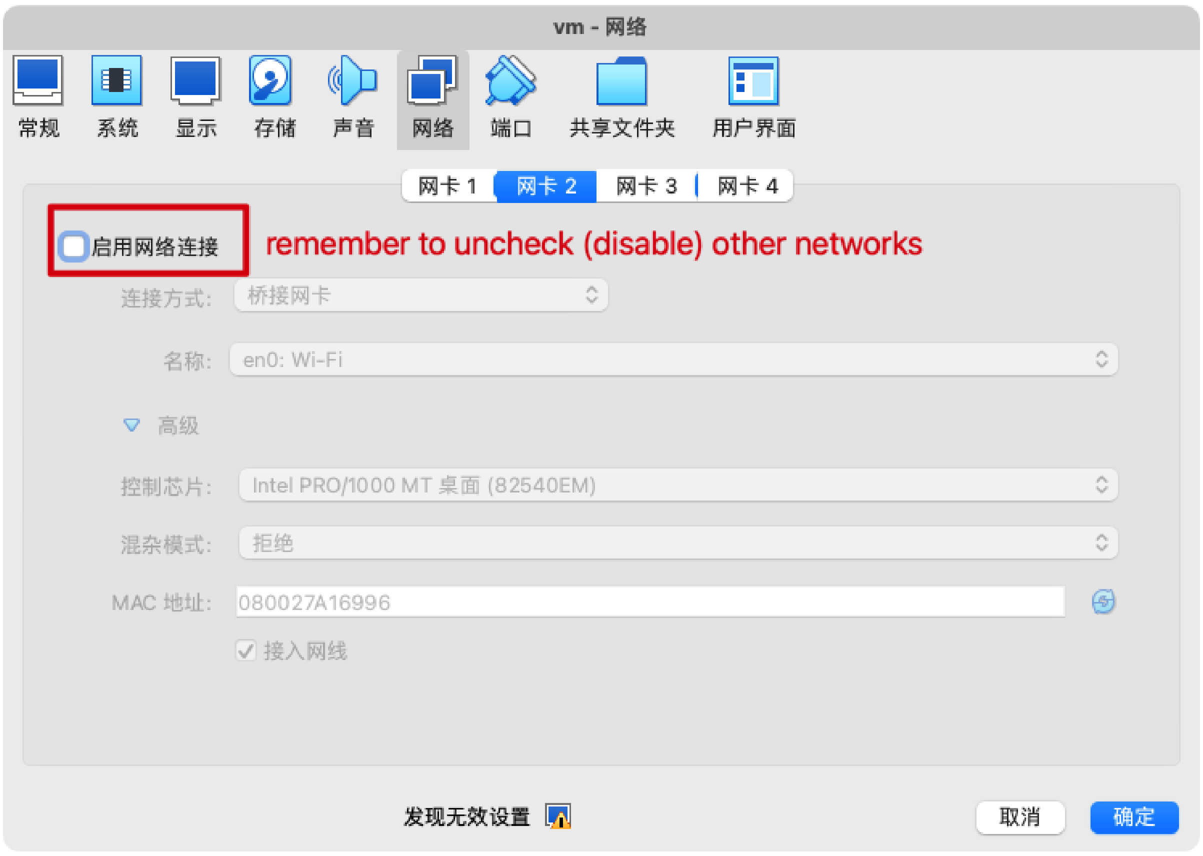
端口转发 (port forwarding)



Host Port Guest Port

	协议	主机IP	主机端口	子系统IP	子系统
1	TCP		2200		22

取消 确定



☐ 启用网络连接 remember to uncheck (disable) other networks

连接方式: 桥接网卡

名称: en0: Wi-Fi

高级

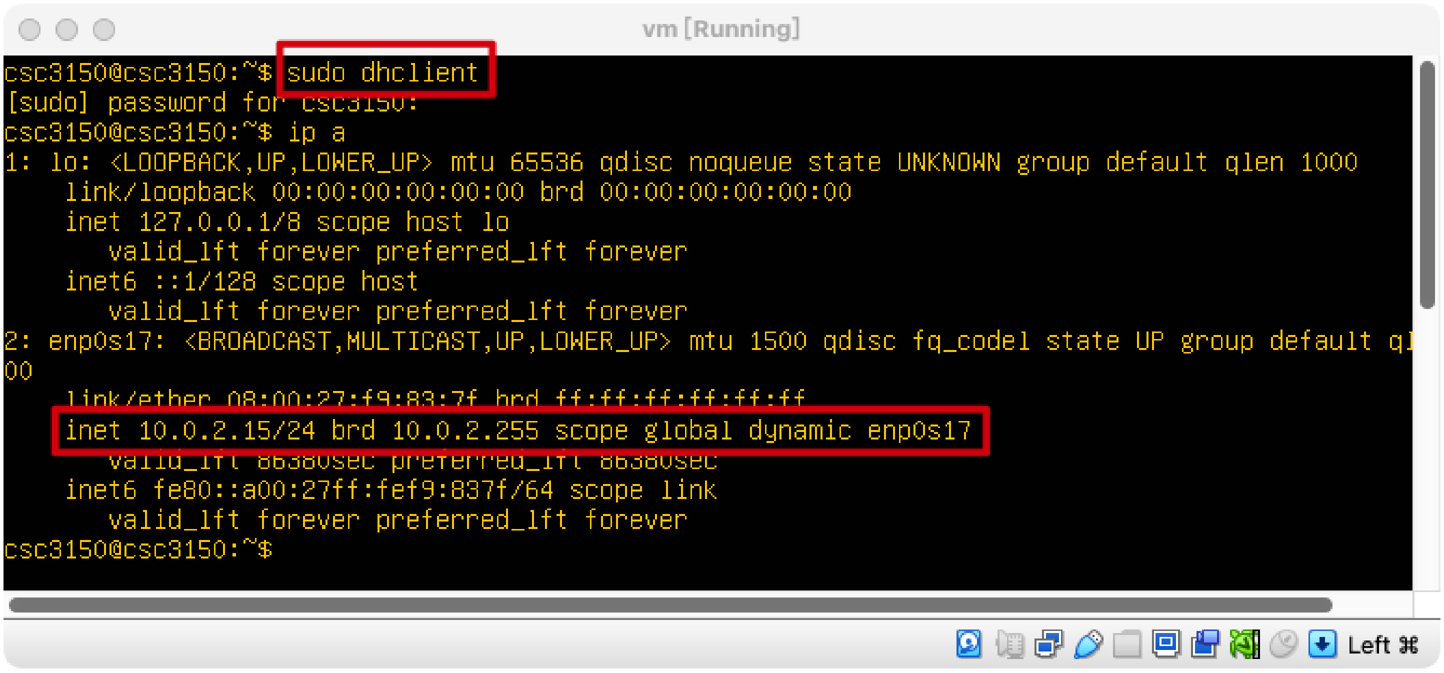
控制芯片: Intel PRO/1000 MT 桌面 (82540EM)

混杂模式: 拒绝

MAC 地址: 080027A16996

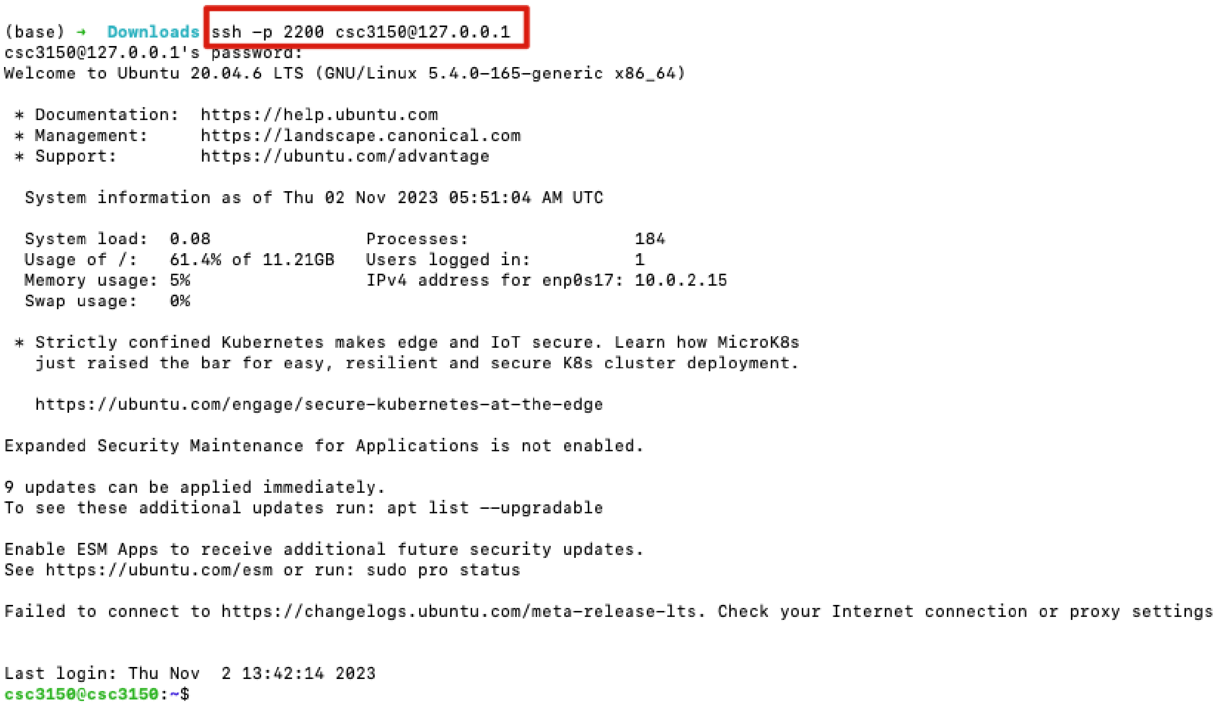
✓ 接入网线

- Log in the virtual machine through the username csc3150 and password csc3150
- Use '**sudo dhclient**' command to assign an IP address. Then try 'ip a' again.



```
csc3150@csc3150:~$ sudo dhclient
[sudo] password for csc3150:
csc3150@csc3150:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default ql
00
    link/ether 08:00:27:f9:83:7f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s17
       valid_lft 86380sec preferred_lft 86380sec
    inet6 fe80::a00:27ff:fef9:837f/64 scope link
       valid_lft forever preferred_lft forever
csc3150@csc3150:~$
```

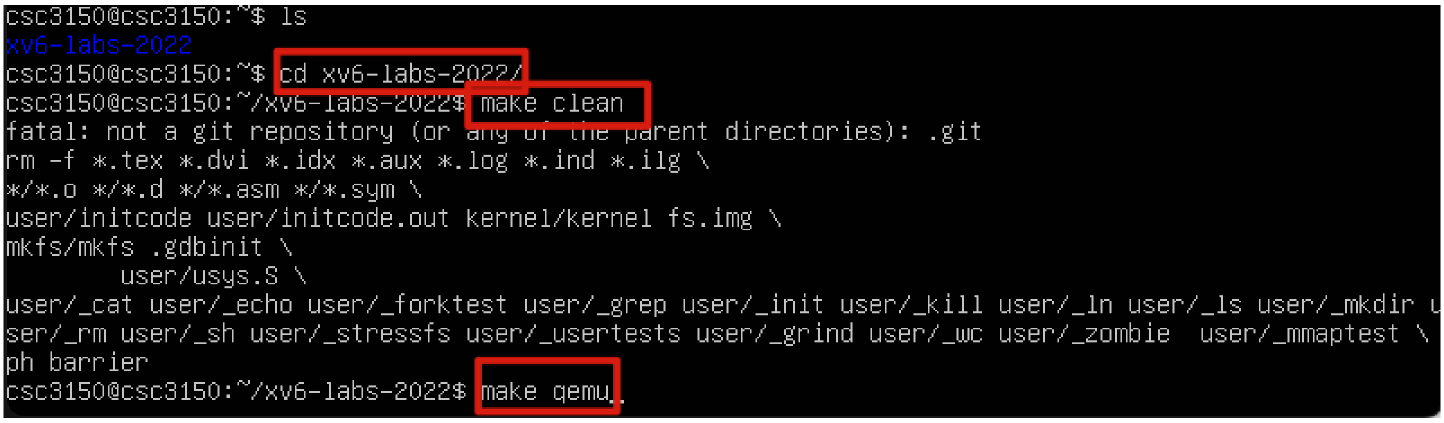
- If you can see the assigned IP shown in the above figure, dhclient works. Then, we use the following command to connect. (***updated on 11/02/2023***)
    - ***ssh -p 2200 csc3150@127.0.0.1***



```
(base) → Downloads ssh -p 2200 csc3150@127.0.0.1
csc3150@127.0.0.1's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-165-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu 02 Nov 2023 05:51:04 AM UTC

  System load:  0.08              Processes:               184
  Usage of /:   61.4% of 11.21GB  Users logged in:         1
  Memory usage: 5%                IPv4 address for enp0s17: 10.0.2.15
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Thu Nov  2 13:42:14 2023
csc3150@csc3150:~$
```

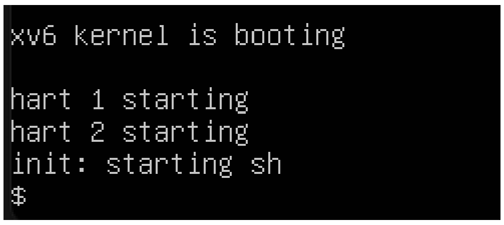
For All Students

1. We have put the template under the default directory. Go to the 'xv6-labs-2022' directory and try the following command to compile and run the project.
   a. (xv6-labs-2022 is the assignment template we gonna to use in assignment 3 and 4, have a try if the following works well)



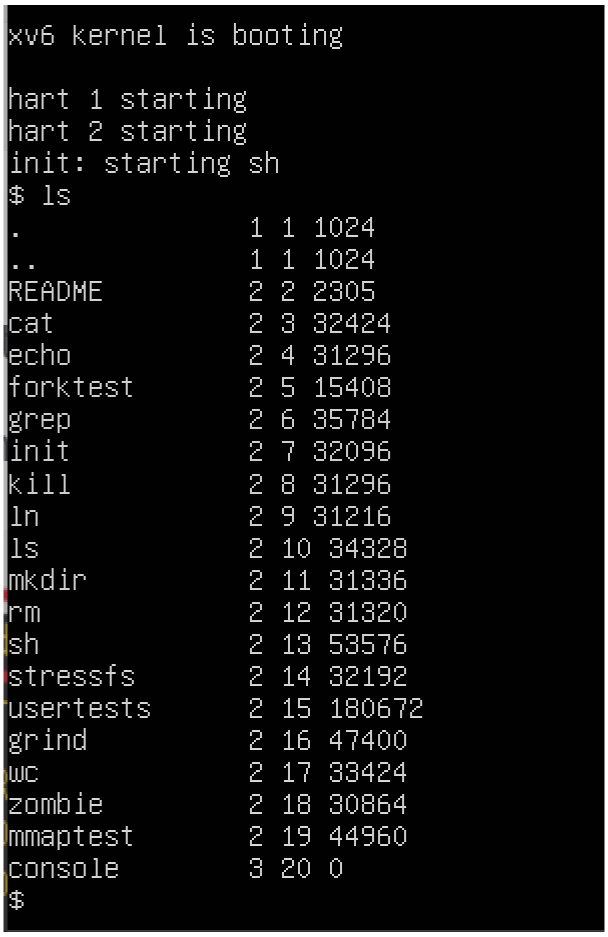
```
csc3150@csc3150:~$ ls
xv6-labs-2022
csc3150@csc3150:~$ cd xv6-labs-2022/
csc3150@csc3150:~/xv6-labs-2022$ make clean
fatal: not a git repository (or any of the parent directories): .git
rm -f *.tex *.dvi *.idx *.aux *.log *.ind *.ilg \
*/*.o */*.d */*.asm */*.sym \
user/initcode user/initcode.out kernel/kernel fs.img \
mkfs/mkfs .gdbinit \
        user/usys.S \
user/_cat user/_echo user/_forktest user/_grep user/_init user/_kill user/_ln user/_ls user/_mkdir u
ser/_rm user/_sh user/_stressfs user/_usertests user/_grind user/_wc user/_zombie  user/_mmaptest \
ph barrier
csc3150@csc3150:~/xv6-labs-2022$ make qemu
```

2. After 'make qemu', the xv6 system powers on.



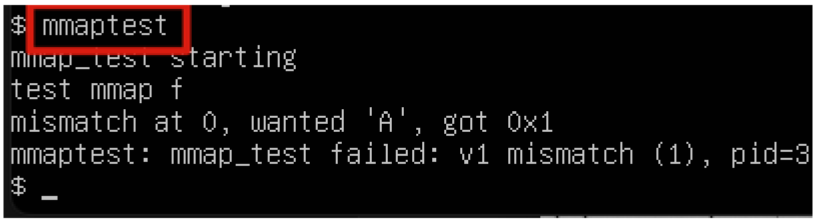
```
xv6 kernel is booting

hart 1 starting
hart 2 starting
init: starting sh
$
```

3. Try 'ls'. You are expected to have the following output like this



```
xv6 kernel is booting

hart 1 starting
hart 2 starting
init: starting sh
$ ls
.              1 1 1024
..             1 1 1024
README         2 2 2305
cat            2 3 32424
echo           2 4 31296
forktest       2 5 15408
grep           2 6 35784
init           2 7 32096
kill           2 8 31296
ln             2 9 31216
ls             2 10 34328
mkdir          2 11 31336
rm             2 12 31320
sh             2 13 53576
stressfs       2 14 32192
usertests      2 15 180672
grind          2 16 47400
wc             2 17 33424
zombie         2 18 30864
mmaptest       2 19 44960
console        3 20 0
$
```

4. Try 'mmaptest'. You are expected to have the following output.

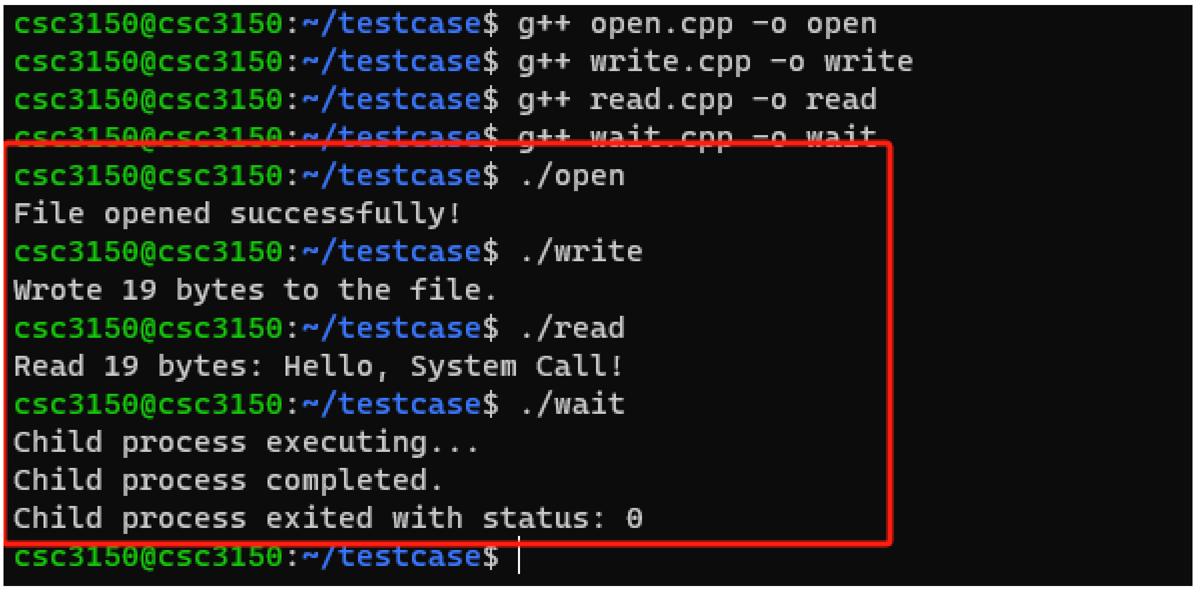


```
$ mmaptest
mmap_test starting
test mmap f
mismatch at 0, wanted 'A', got 0x1
mmaptest: mmap_test failed: v1 mismatch (1), pid=3
$ _
```

5. To quit the xv6 system, press Ctrl a. Leave it. Then type x.

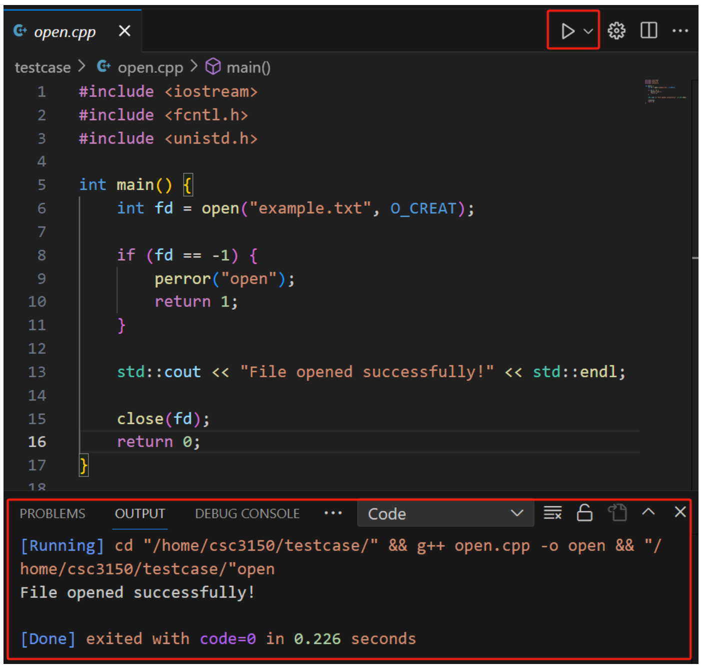
Try the System Call with provided scripts

1. Upload the provided scripts that invoke system calls (open, write, read, wait) written in cpp. Here provide two simple uploading methods:
    a. You can directly copy the code to the VM or drag and drop it by using vscode.
    b. You can use SCP to upload files/folders to the specified directory on the VM. For example,
        i. `scp -P 2200 read.cpp csc3150@127.0.0.1:/home/csc3150/testcase/`
2. Execute the following compilation, normally expecting the output provided below.



```
csc3150@csc3150:~/testcase$ g++ open.cpp -o open
csc3150@csc3150:~/testcase$ g++ write.cpp -o write
csc3150@csc3150:~/testcase$ g++ read.cpp -o read
csc3150@csc3150:~/testcase$ g++ wait.cpp -o wait
csc3150@csc3150:~/testcase$ ./open
File opened successfully!
csc3150@csc3150:~/testcase$ ./write
Wrote 19 bytes to the file.
csc3150@csc3150:~/testcase$ ./read
Read 19 bytes: Hello, System Call!
csc3150@csc3150:~/testcase$ ./wait
Child process executing...
Child process completed.
Child process exited with status: 0
csc3150@csc3150:~/testcase$ 
```

1. For students using VSCode for editing, an optional solution is to install plugins such as Code Runner to achieve more convenient compilation and debugging. (Instructions on how to connect VS Code to a virtual machine are provided later.)
   a. Example: use code runner to have a more convenient run and debug with VS Code

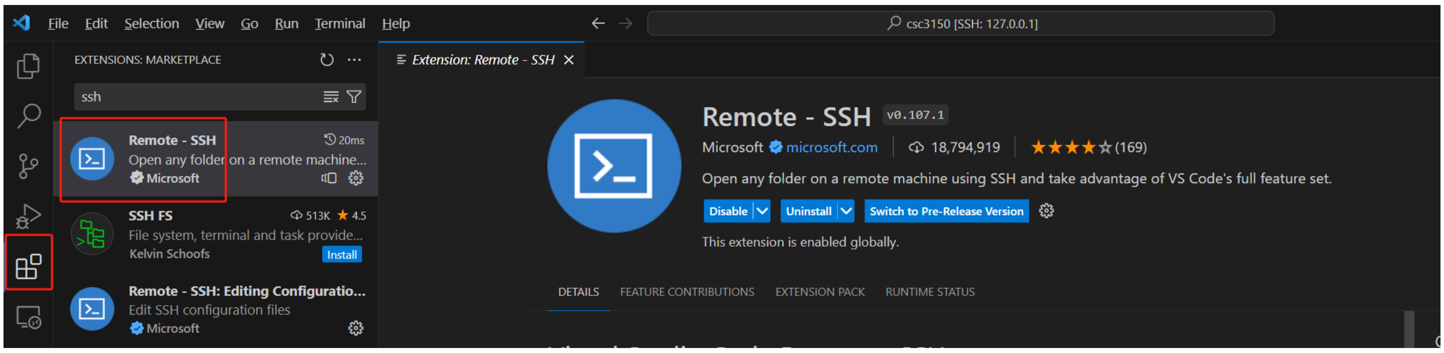


```cpp
#include <iostream>
#include <fcntl.h>
#include <unistd.h>

int main() {
    int fd = open("example.txt", O_CREAT);

    if (fd == -1) {
        perror("open");
        return 1;
    }

    std::cout << "File opened successfully!" << std::endl;

    close(fd);
    return 0;
}
```

```
[Running] cd "/home/csc3150/testcase/" && g++ open.cpp -o open && "/
home/csc3150/testcase/"open
File opened successfully!

[Done] exited with code=0 in 0.226 seconds
```

(Optional) Use VSCode as Code Editor through SSH Connection to VM on Local Port

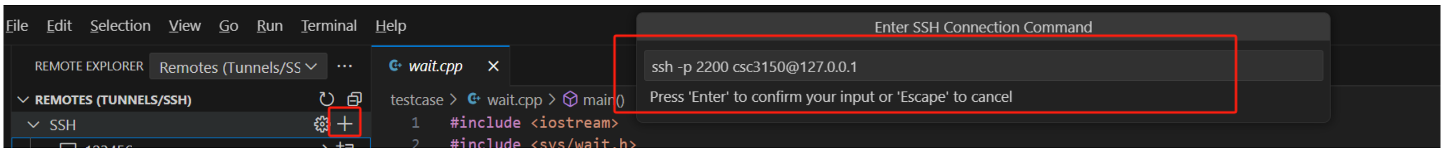
1. Download and Install VSCode
2. Install ssh Plugin



Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set.

This extension is enabled globally.

3. Connect to VM Using SSH, Assuming you have the IP address assigned, use the following command in the terminal to connect to the VM: ssh -p 2200 csc3150@127.0.0.1



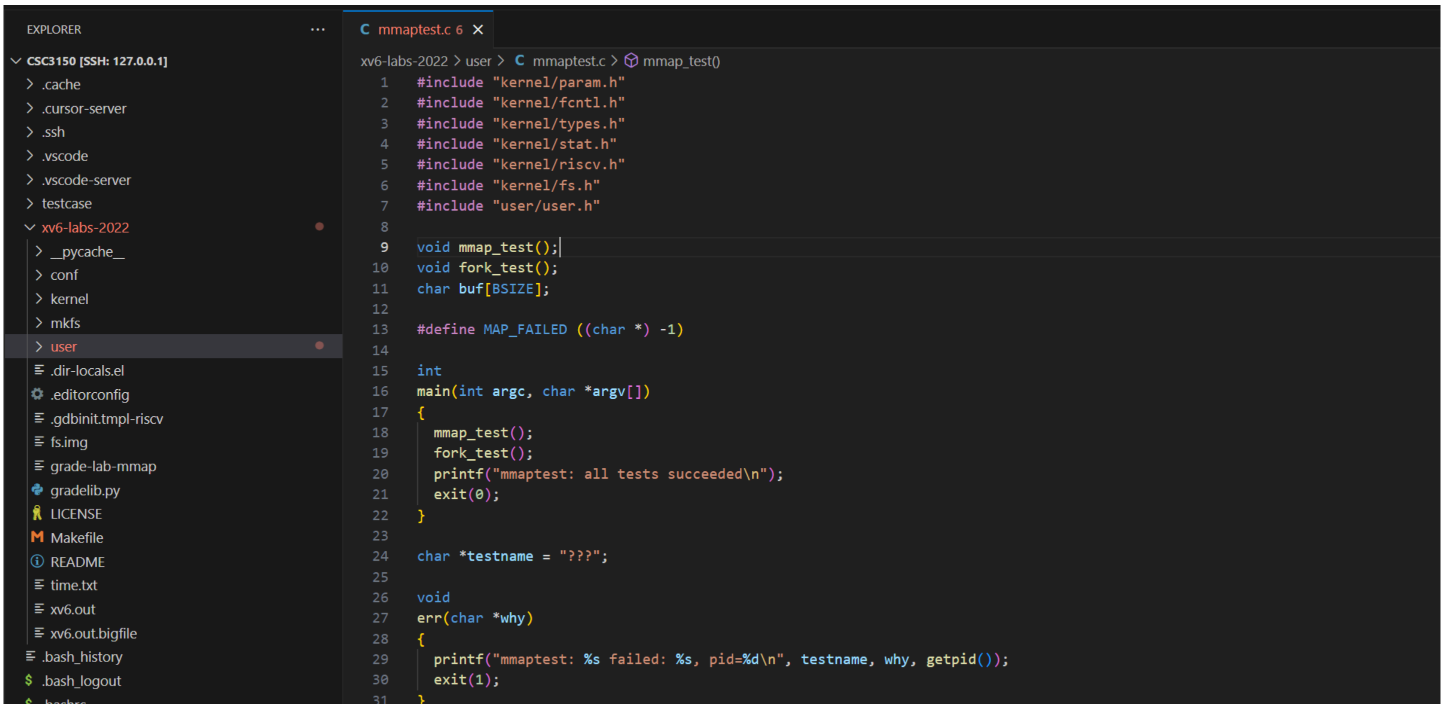
Enter SSH Connection Command

```
ssh -p 2200 csc3150@127.0.0.1
```

Press 'Enter' to confirm your input or 'Escape' to cancel

```
#include <iostream>
#include <sys/wait.h>
```

4. Edit/Debug Codes from VM on VS code.



```c
#include "kernel/param.h"
#include "kernel/fcntl.h"
#include "kernel/types.h"
#include "kernel/stat.h"
#include "kernel/riscv.h"
#include "kernel/fs.h"
#include "user/user.h"

void mmap_test();
void fork_test();
char buf[BSIZE];

#define MAP_FAILED ((char *) -1)

int
main(int argc, char *argv[])
{
  mmap_test();
  fork_test();
  printf("mmaptest: all tests succeeded\n");
  exit(0);
}

char *testname = "???";

void
err(char *why)
{
  printf("mmaptest: %s failed: %s, pid=%d\n", testname, why, getpid());
  exit(1);
```