操作前需知

- 所有的测试都必须先运行前面那个模块

```
import os
from cxz.userutils import UserAction

ssh_args = {
    'hostname': '192.168.220.128',
    'port': 22,
    'username': 'lzh',
    'password': '083636'
}

user_task = UserAction(ssh_args)
```

终端快捷方式

- 粘贴（上）下一条命令

  > 方向键上下键

- 翻查历史记录

  > CTRL+R

- 终止当下命令

  > CTRL+C

- 退出终端

  > eixt/quit/CTRL+E

- 清理屏幕

  > clear

- 查看之前打过的命令（可关键词查找）

  > history

- 将光标移到行首

  > CTRL+A

- 将光标移到行尾

  > CTRL+E

- 将光标按单词往前移

  ```
  ALT+B
  ```

- 将光标按单词往后移

  ```
  ALT+F
  ```

- 向前删除一个单词

  ```
  ALT+BACESPACE
  ```

- 向后删除一个单词

  ```
  ALT+D
  ```

Linux一些命令操作

- 列出所有可更新的软件清单命令

  ```
  sudo apt update
  ```

- 升级软件包

  ```
  sudo apt upgrade
  ```

- 查看当前路径下的文件

  ```
  ll
  ```

- 显示指定工作目录下之内容

  ```
  ls
  ```

- 改变终端所在路径

  ```
  cd 路径
  ```

- 打开文件

  ```
  sudo gedit 文件路径 . 文件名//按TAB键显示当下文件夹
  ```

- 查看当前在哪个路径

  ```
  pwd
  ```

- 如无法将本机复制粘贴到虚拟机解决方法

  > sudo apt install open-vm-tools

- 删除包

  > sudo apt remove 文件名

- 删除冗余包

  > sudo apt autoremove

- 查找某文件路径

  > which 文件名

- 查看用户的关键信息

  > cat /etc/passwd

- 下载包

  > sudo apt install 文件名

- 帮助手册

  > man 命令

- 命令用于连接文件并打印到标准输出设备上

  > cat /etc/passwd

- 抄代码 *//搬砖*

  > git clone

- 将硬盘列出来

  > lsblk

ssh测试

步骤

1. 更新系统

```
sudo apt update
```

2. 下载ssh包

```
sudo apt install openssh-server
```

3. 查IP地址

```
ip a
```

补充

- 服务

```
systemctl
```

- 查看ssh服务

```
systemctl status sshd
```

- 重启ssh服务

```
systemctl restart sshd
```

- 开启ssh服务

```
systemctl start sshd
```

- 关闭ssh服务

```
systemctl stop sshd
```

- 检查是否安装成功ssh服务

```
ssh 用户名 @localhost
```

换源测试

步骤

---

1. 打开换源的文件

   ```
   sudo gedit /etc/apt/sources.list
   ```

2. 替换国内源*//这里仅展示中科大源*

   ```
   deb https://mirrors.ustc.edu.cn/ubuntu/ jammy main restricted universe multiverse deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy main restricted universe multiverse deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-security main restricted universe multiverse deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-security main restricted universe multiverse deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
   ```

3. 更新系统

   ```
   sudo apt update
   ```

4. 更新软件

   ```
   sudo apt upgrade
   ```

考试如果禁用剪切板的话，补救方式

1. ```
   wget http://web.tecxz.com:7080/ubuntu/sources.list.ustc
   ```

2. ```
   sudo mv sources.list.ustc /etc/apt/sources.list
   ```

3. 更新系统

```
sudo apt update
```

4. 更新软件

```
sudo apt upgrade
```

基础测试

1. 下载git和zsh

```
sudo apt install git zsh
```

2. 下载curl

```
sudo apt install curl
```

3. 调用老师的zsh

```
curl -fsSL http://web.tecxz.com:7080/file/zsh/install.sh | sh
```

4. 默认改成zsh

```
chsh -s /usr/bin/zsh
```

5. 改zsh主题（第11行双引号里的改成ys）

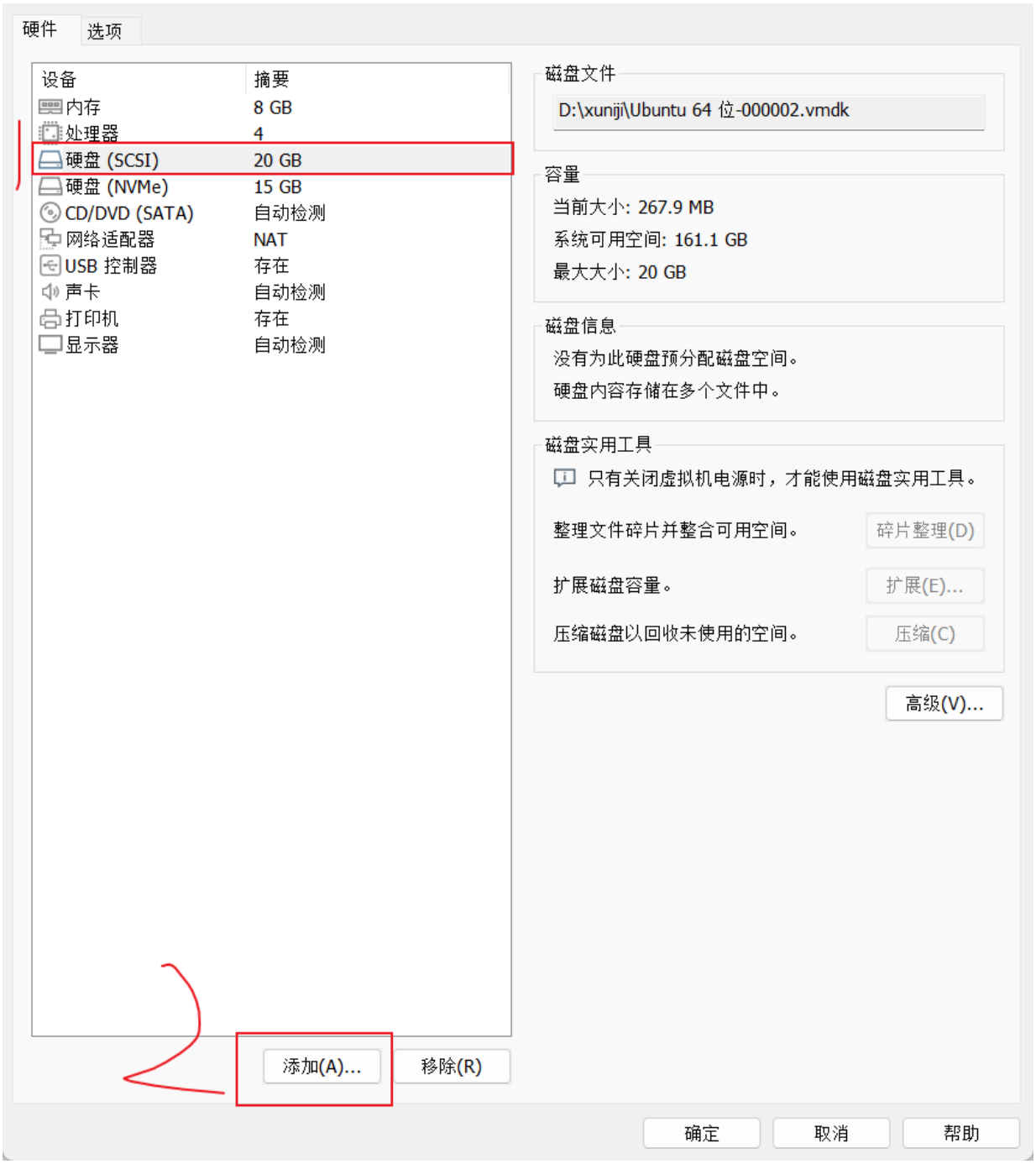
```
gedit ~/.zshrc
```

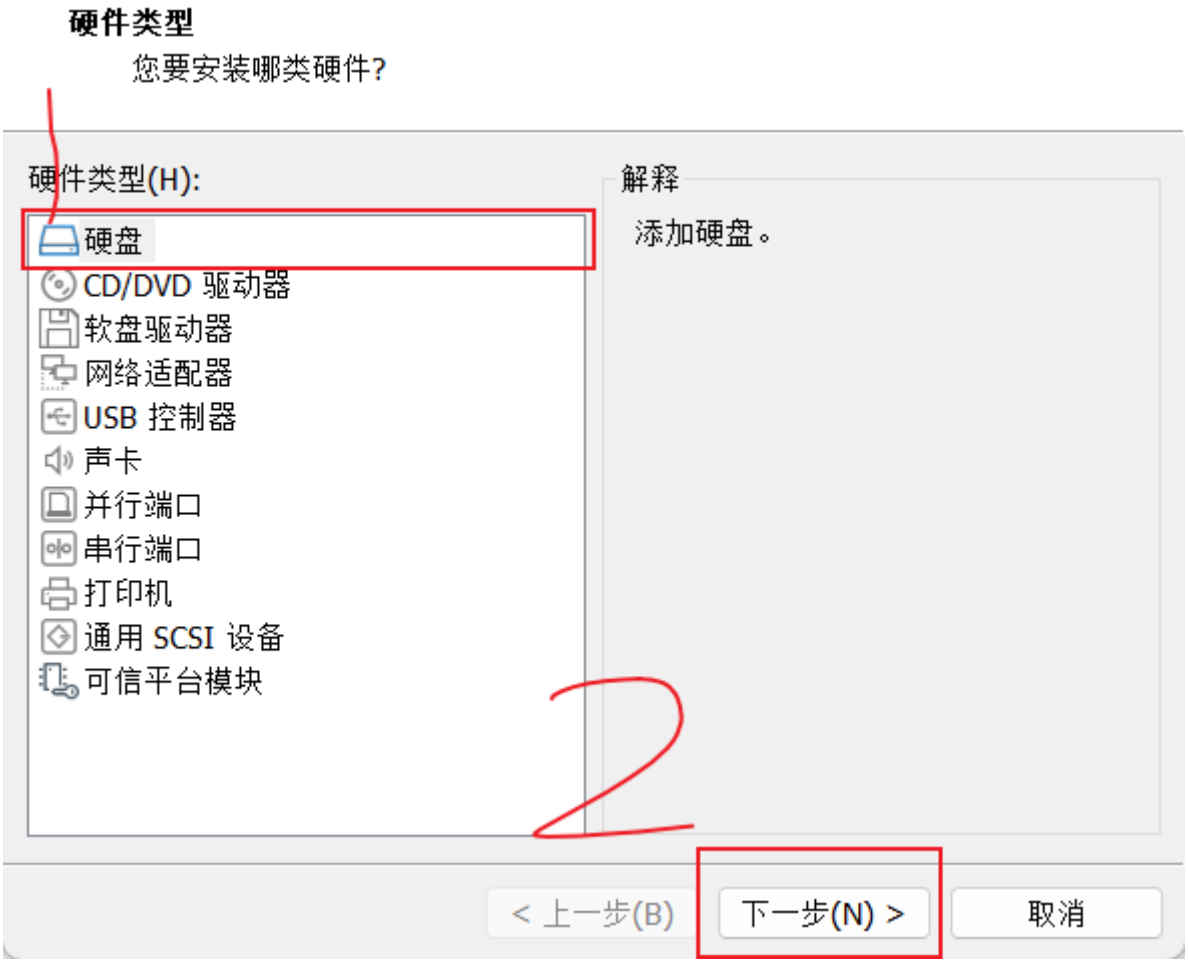
补充

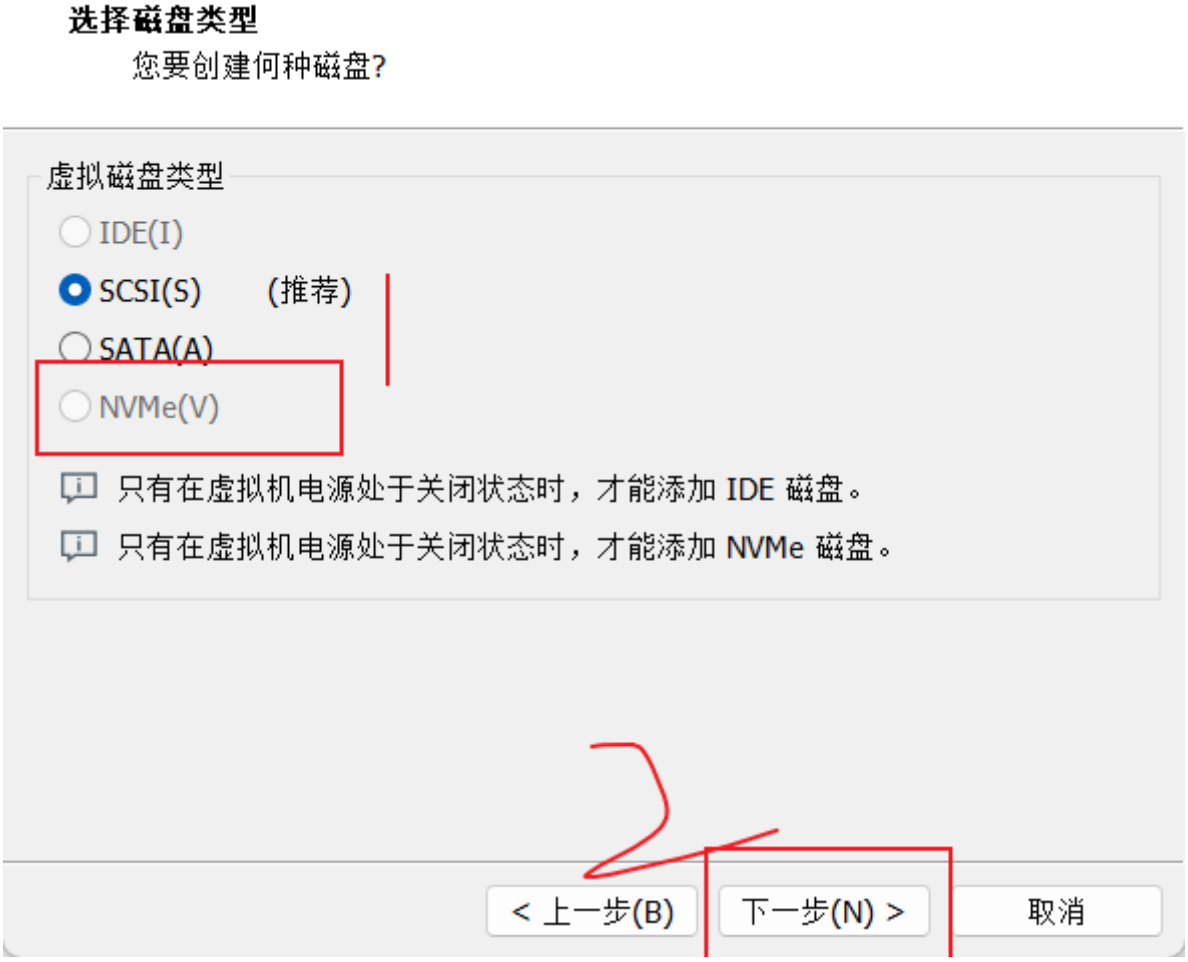
- 改成zsh后必须要登**pyclss**，不然命令运行不了

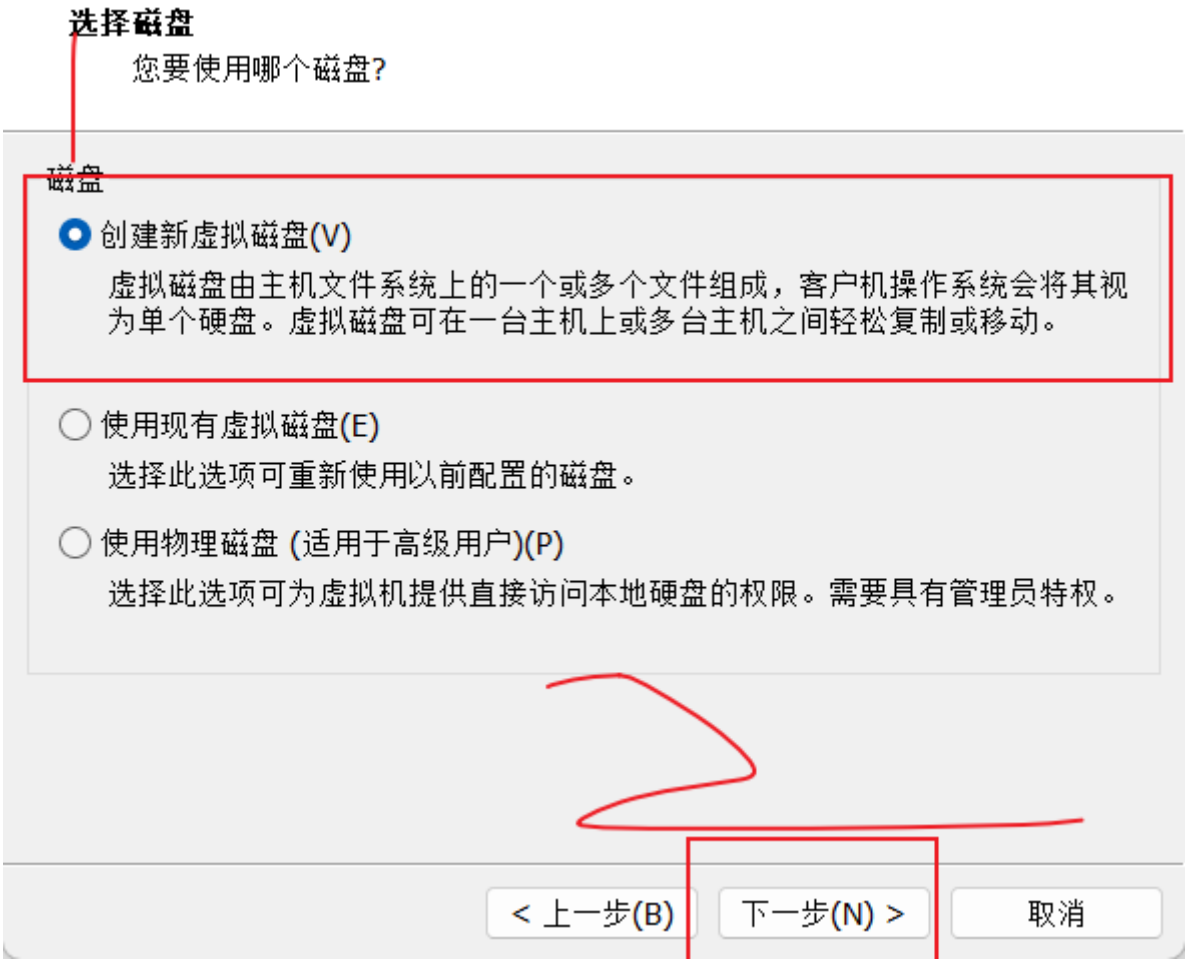
分区创建测试

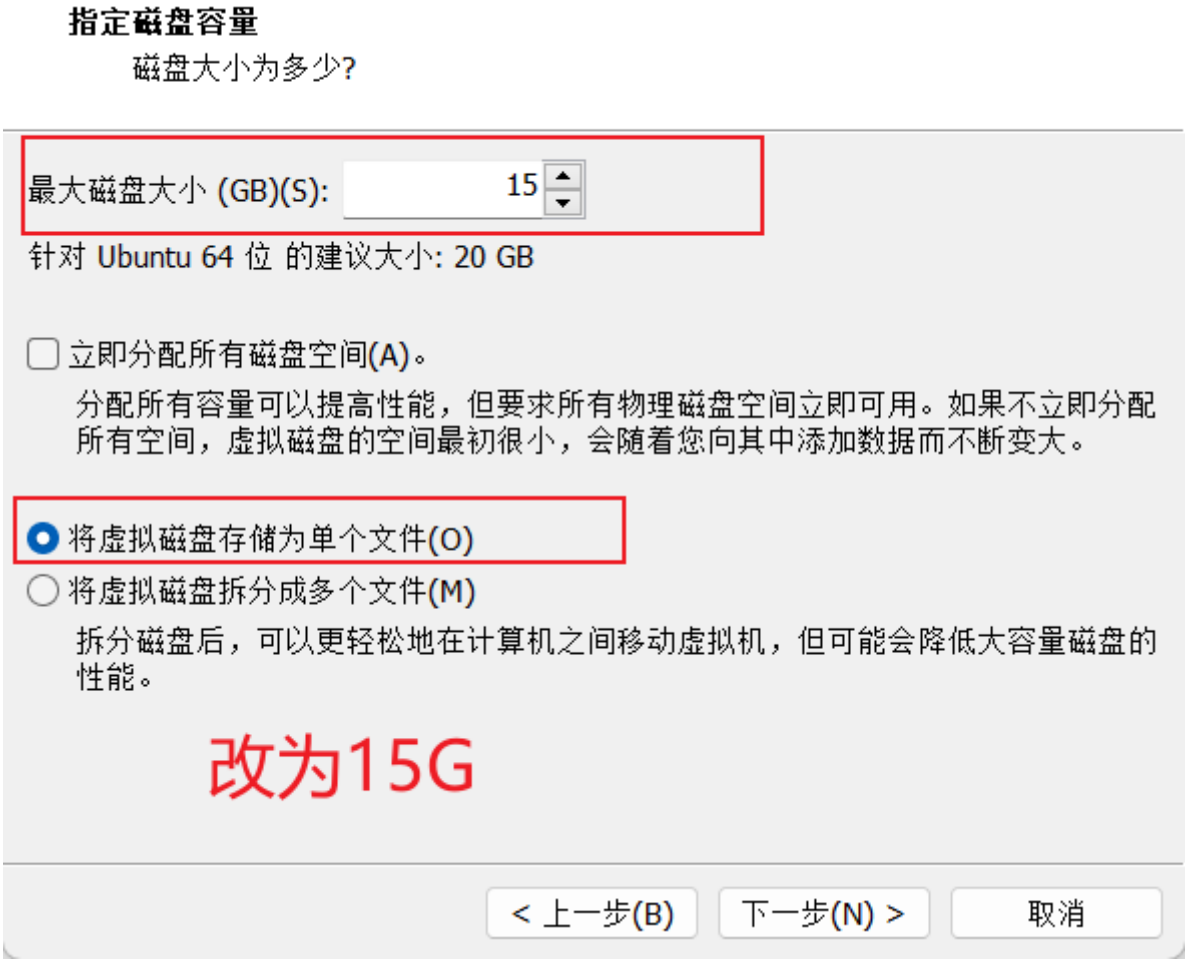
1. 创建新硬盘（15个G）

1. 
硬件
选项
设备
摘要
内存
8 GB
处理器
4
硬盘 (SCSI)
20 GB
硬盘 (NVMe)
15 GB
CD/DVD (SATA)
自动检测
网络适配器
NAT
USB 控制器
存在
声卡
自动检测
打印机
存在
显示器
自动检测
磁盘文件
D:\xunji\Ubuntu 64 位-000002.vmdk
容量
当前大小: 267.9 MB
系统可用空间: 161.1 GB
最大大小: 20 GB
磁盘信息
没有为此硬盘预分配磁盘空间。
硬盘内容存储在多个文件中。
磁盘实用工具
只有关闭虚拟机电源时，才能使用磁盘实用工具。
整理文件碎片并整合可用空间。
碎片整理(D)
扩展磁盘容量。
扩展(E)...
压缩磁盘以回收未使用的空间。
压缩(C)
高级(V)...
添加(A)...
移除(R)
确定
取消
帮助

2. 
硬件类型
您要安装哪类硬件?
硬件类型(H):
硬盘
CD/DVD 驱动器
软盘驱动器
网络适配器
USB 控制器
声卡
并行端口
串行端口
打印机
通用 SCSI 设备
可信平台模块
解释
添加硬盘。
< 上一步(B)
下一步(N) >
取消

3. 
选择磁盘类型
您要创建何种磁盘?
虚拟磁盘类型
IDE(I)
SCSI(S)
(推荐)
SATA(A)
NVMe(V)
只有在虚拟机电源处于关闭状态时，才能添加 IDE 磁盘。
只有在虚拟机电源处于关闭状态时，才能添加 NVMe 磁盘。
< 上一步(B)
下一步(N) >
取消

4. 

5. 

6. 一路确定下去
7.

2. 查找是否有新硬盘

> ls /dev

3. 进入硬盘设置

> 进入nvme0n1分区

1.
```
$ sudo gdisk /dev/nvme0n1
[sudo] lzh 的密码:
GPT fdisk (gdisk) version 1.0.8

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.
```

> 显示当前磁盘的分区表

2.
```
Command (? for help): p
Disk /dev/nvme0n1: 31457280 sectors, 15.0 GiB
Model: VMware Virtual NVMe Disk
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): DF6E6330-8E27-4F44-8DFD-323C30310F71
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 31457246
Partitions will be aligned on 2048-sector boundaries
Total free space is 31457213 sectors (15.0 GiB)

Number  Start (sector)    End (sector)  Size       Code  Name
```

> 增加一个新的分区

3. 输入n后按回车

```
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-31457246, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-31457246, default = 31457246) or {+-}size{KMGTP}: +500M
Current type is 8300 (Linux filesystem)
```

按玩n回车后，按2下回车后，输入+500M回车

4. 输入L回车

```
Hex code or GUID (L to show codes, Enter = 8300): L
```

5. 上一步一直回车到出现此界面，输入ef00

```
Hex code or GUID (L to show codes, Enter = 8300): ef00
Changed type of partition to 'EFI system partition'
```

6.

```
Command (? for help): p
Disk /dev/nvme0n1: 31457280 sectors, 15.0 GiB
Model: VMware Virtual NVMe Disk
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): DF6E6330-8E27-4F44-8DFD-323C30310F71
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 31457246
Partitions will be aligned on 2048-sector boundaries
Total free space is 30433213 sectors (14.5 GiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         1026047   500.0 MiB   EF00  EFI system partition
```

7. 按完n回车后，一直回车

```
Command (? for help): n
Partition number (2-128, default 2):
First sector (34-31457246, default = 1026048) or {+-}size{KMGTP}:
Last sector (1026048-31457246, default = 31457246) or {+-}size{KMGTP}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'
```

8.
```
Command (? for help): p
Disk /dev/nvme0n1: 31457280 sectors, 15.0 GiB
Model: VMware Virtual NVMe Disk
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): DF6E6330-8E27-4F44-8DFD-323C30310F71
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 31457246
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         1026047   500.0 MiB   EF00  EFI system partition
   2         1026048        31457246   14.5 GiB    8300  Linux filesystem
```

上一步一直回车到出现此界面，输入p回车

9. 分区1命名

```
Command (? for help): c
Partition number (1-2): 1
Enter name: classesp
```

10. 分区2命名

```
Command (? for help): c
Partition number (1-2): 2
Enter name: classroot
```

11.
```
Command (? for help): p
Disk /dev/nvme0n1: 31457280 sectors, 15.0 GiB
Model: VMware Virtual NVMe Disk
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): DF6E6330-8E27-4F44-8DFD-323C30310F71
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 31457246
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         1026047   500.0 MiB   EF00  classesp
   2         1026048        31457246   14.5 GiB    8300  classroot
```

12.
```
Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!
```
按完w回车后，按y回车
```
Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/nvme0n1.
The operation has completed successfully.
```

13. 检查是否硬盘分区成功

```
$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0         7:0    0    62M  1 loop /snap/core20/1587
loop1         7:1    0     4K  1 loop /snap/bare/5
loop2         7:2    0 346.3M  1 loop /snap/gnome-3-38-2004/119
loop3         7:3    0 240.6M  1 loop /snap/firefox/2356
loop4         7:4    0  63.3M  1 loop /snap/core20/1822
loop5         7:5    0 163.3M  1 loop /snap/firefox/1635
loop6         7:6    0 400.8M  1 loop /snap/gnome-3-38-2004/112
loop7         7:7    0  91.7M  1 loop /snap/gtk-common-themes/1535
loop8         7:8    0  45.9M  1 loop /snap/snap-store/582
loop9         7:9    0   284K  1 loop /snap/snapd-desktop-integration/14
loop10        7:10   0  49.8M  1 loop /snap/snapd/18357
loop11        7:11   0  45.9M  1 loop /snap/snap-store/638
loop12        7:12   0   304K  1 loop /snap/snapd-desktop-integration/49
sda           8:0    0    20G  0 disk
├─sda1        8:1    0   512M  0 part /boot/efi
└─sda2        8:2    0  19.5G  0 part /var/snap/firefox/common/host-hunspell
                                      /
sr0          11:0    1  1024M  0 rom
nvme0n1     259:0    0    15G  0 disk
├─nvme0n1p1 259:3    0   500M  0 part
└─nvme0n1p2 259:4    0  14.5G  0 part
```

磁盘挂载测试(挂载必须先挂载p2，再挂载p1)

1. 创建B系统的目录

```
sudo mkdir /mnt/usb
```

2. 对p1盘格式化

```
# lzh @ mechrev in ~/Desktop [20:10:30] C:1
$ sudo mkfs.vfat -F32 /dev/nvme0n1p1
[sudo] lzh 的密码：
mkfs.fat 4.2 (2021-01-31)
```

3. 对p2盘格式化

```
# lzh @ mechrev in ~/Desktop [20:11:54]
$ sudo mkfs.ext4 /dev/nvme0n1p2
mke2fs 1.46.5 (30-Dec-2021)
创建含有 3803899 个块（每块 4k）和 950976 个 inode 的文件系统
文件系统 UUID：ef8e182c-f6bd-442e-b264-3f46e16ff42b
超级块的备份存储于下列块：
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

正在分配组表：  完成
正在写入 inode表：  完成
创建日志（16384 个块）：  完成
写入超级块和文件系统账户统计信息：  已完成
```

4. 挂载p2盘

```
sudo mount /dev/nvme0n1p2 /mnt/usb
```

5. 创建/mnt/usb/boot/efi目录

```
sudo mkdir -p /mnt/usb/boot/efi
```

6. 挂载p1盘

```
sudo mount /dev/nvme0n1p1 /mnt/usb/boot/efi
```

7. 查看是否挂载成功

```
lsblk
```

B系统预创建测试

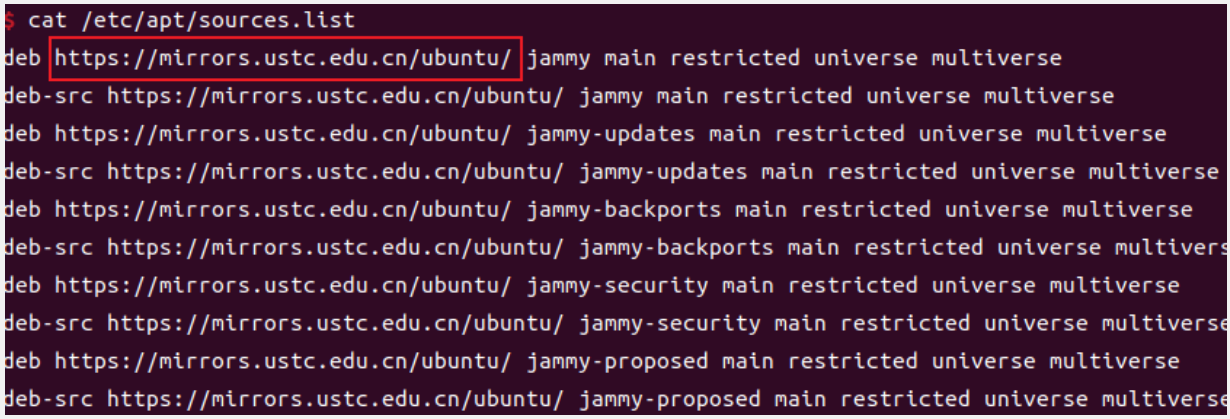
1. B系统根目录

```
ls /mnt/usb
```

2. 构建一套基本的系统(根文件系统)

- sudo apt install debootstrap
- sudo apt search arch-install
- sudo apt install arch-install-scripts

3. 查看源的网站地址

> cat /etc/apt/sources.list
>
> 
> ```
> $ cat /etc/apt/sources.list
> deb https://mirrors.ustc.edu.cn/ubuntu/ jammy main restricted universe multiverse
> deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy main restricted universe multiverse
> deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
> deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
> deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
> deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-backports main restricted universe multivers
> deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
> deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
> deb https://mirrors.ustc.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
> deb-src https://mirrors.ustc.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
> ```

4. 下载根文件系统

> sudo debootstrap --arch=amd64 jammy /mnt/usb https://mirrors.ustc.edu.cn/ubuntu/

5. 创建fstab

> genfstab /mnt/usb | sudo tee /mnt/usb/etc/fstab

6. 检查一下

> cat /mnt/usb/etc/fstab

7. 复制A系统sources.list文件至B系统nt/usb/etc/apt 目录下

> sudo cp /etc/apt/sources.list /mnt/usb/etc/apt/sources.list

B系统安装基础测试

1. 检查一下

> cat /mnt/usb/etc/apt/sources.list

2. 将A系统的dev目录与B系统的dev目录绑定

> sudo mount -o bind /dev /mnt/usb/dev

3. 验证是否绑定成功

```
ls /mnt/usb/dev
```

4. 将系统信息绑定

```
sudo mount -o bind /sys /mnt/usb/sys
```

5. 将用户信息绑定

```
sudo mount -o bind /proc /mnt/usb/proc
```

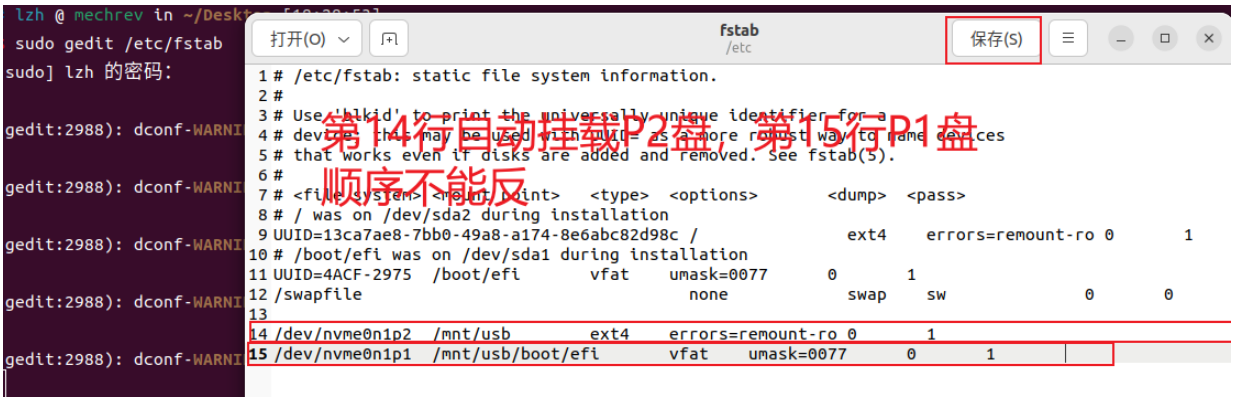
6. 转换到B系统

```
sudo chroot /mnt/usb
```

7. 在A系统修改启动项自动挂载P2盘（不能自动挂载P1盘）

    1. sudo gedit /etc/fstab

    2. /dev/nvme0n1p2 /mnt/usb ext4 errors=remount-ro 0 1



字符界面使用文本编辑器

- nano编辑器

```
nano //q启动编辑器
CTRL+X //保存 sudo nano 文件的绝对路径
```

- vim编译器 //重要
    - 下载vim编译器

```
sudo apt install vim
```

- vi编译器
    - 插入(ESC退出)

> 按I

- 进入命令界面

> 按冒号

- 保存并退出

> wq

- 退出

> q

B系统必备软件测试

1. 在B系统更新软件

> apt update

2. 在B系统安装Linux的内核

- 在A系统拷贝 sources.list到B系统上

> sudo cp /etc/apt/sources.list /mnt/usb/etc/apt/sources.list

- 转换到B系统更新系统

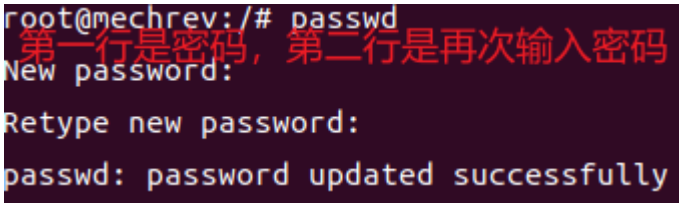
> sudo apt update

- 下载Linux内核到B系统

> apt install linux-image-generic

3. 设置root密码

> passwd



4. 创建一个用户

```
useradd joker -m
```

5. 验证是否用户有home目录

```
ls /home
```

6. 给刚创建的用户越权权限

```
usermod -aG sudo 用户名
```

7. 安装网络工具软件

```
apt install net-tools network-manager
```

8. 安装nano编译器

```
sudo apt install nano
```

B系统引导测试

1. 安装grub

```
apt install efibootmgr grub-efi-amd64
```

2. 更新系统

```
apt update
```

3. 挂载

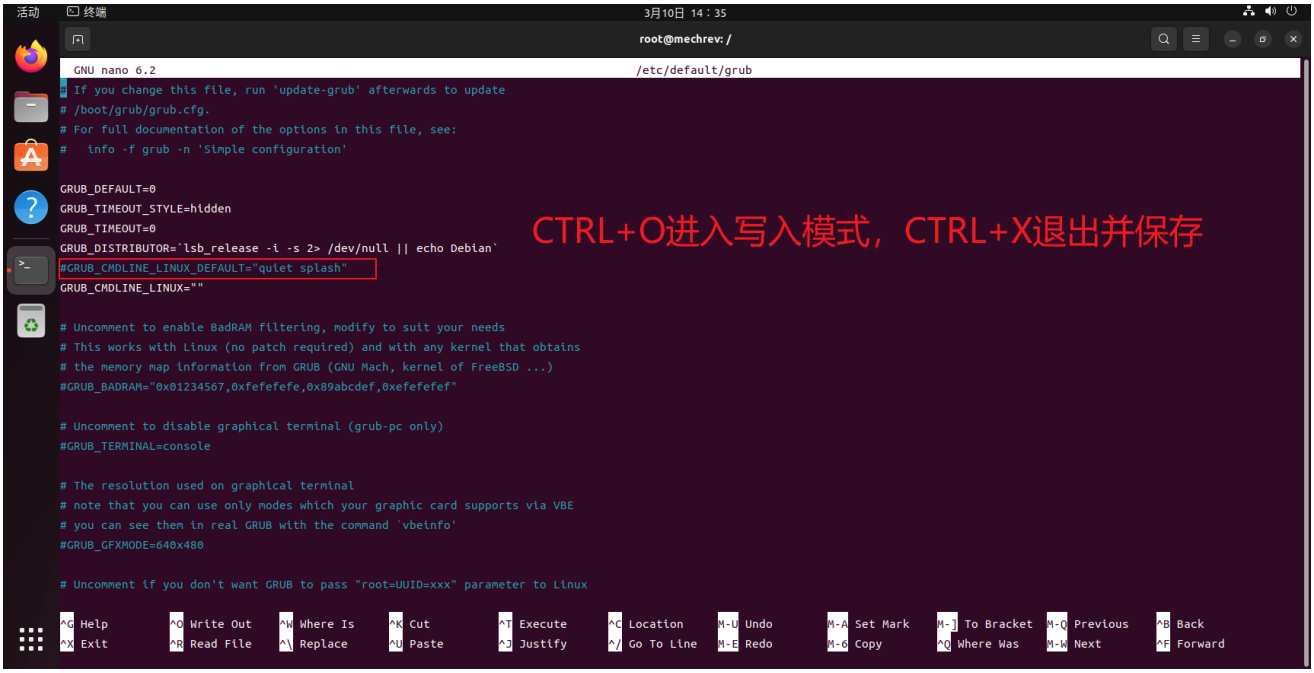
```
mount -t efivarfs efivarfs /sys/firmware/efi/efivars/
```

4. 下载grub

```
grub-install -v --target=x86_64-efi --recheck /dev/nvme0n1
```

5. 使用nano编辑器编译

```
nano /etc/default/grub
```



6. 更新grub

> update-grub

7. 下载NetworkManager

> apt install network-manager

B系统ssh测试(关机后进入A系统才能跑100分，在字符界面安装)

1. 重启进入B系统
2. 输入账户密码
3. 使用bash

> bash

4. 使用指令查看

> systemctl status NetworkManager

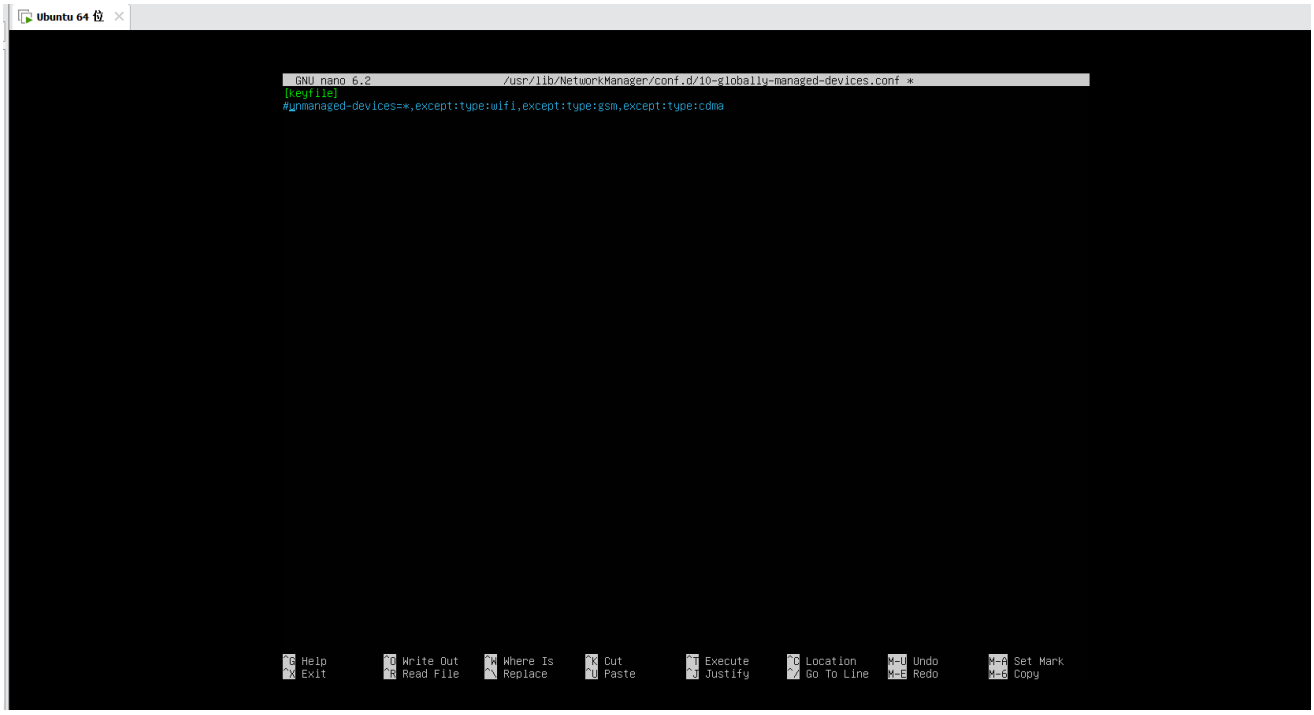
```
joker@joker-virtual-machine:~$ systemctl  status NetworkManager
● NetworkManager.service - Network Manager
     Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-01-02 10:41:11 UTC; 4min 31s ago
       Docs: man:NetworkManager(8)
   Main PID: 532 (NetworkManager)
      Tasks: 3 (limit: 4584)
     Memory: 10.0M
        CPU: 34ms
     CGroup: /system.slice/NetworkManager.service
             └─532 /usr/sbin/NetworkManager --no-daemon

Warning: some journal files were not opened due to insufficient permissions.
joker@joker-virtual-machine:~$
```

5. 输入（可用TAB补全）

> sudo nano /usr/lib/NetworkManager/conf.d/10-globally-managed-devices.conf

6. 注释掉内容



7. 输入

> sudo systemctl restart NetworkManager

8. 查看ip

> ip a

9. 更新

> sudo apt update

10. 安装ssh

```
sudo apt install openssh-server
```

11. 查看ip

```
ip a
```

B系统基础测试（关机后进入A系统才能得100分，在字符界面安装）

1. 安装zsh和git

```
sudo apt install zsh git
```

2. 安装zsh环境

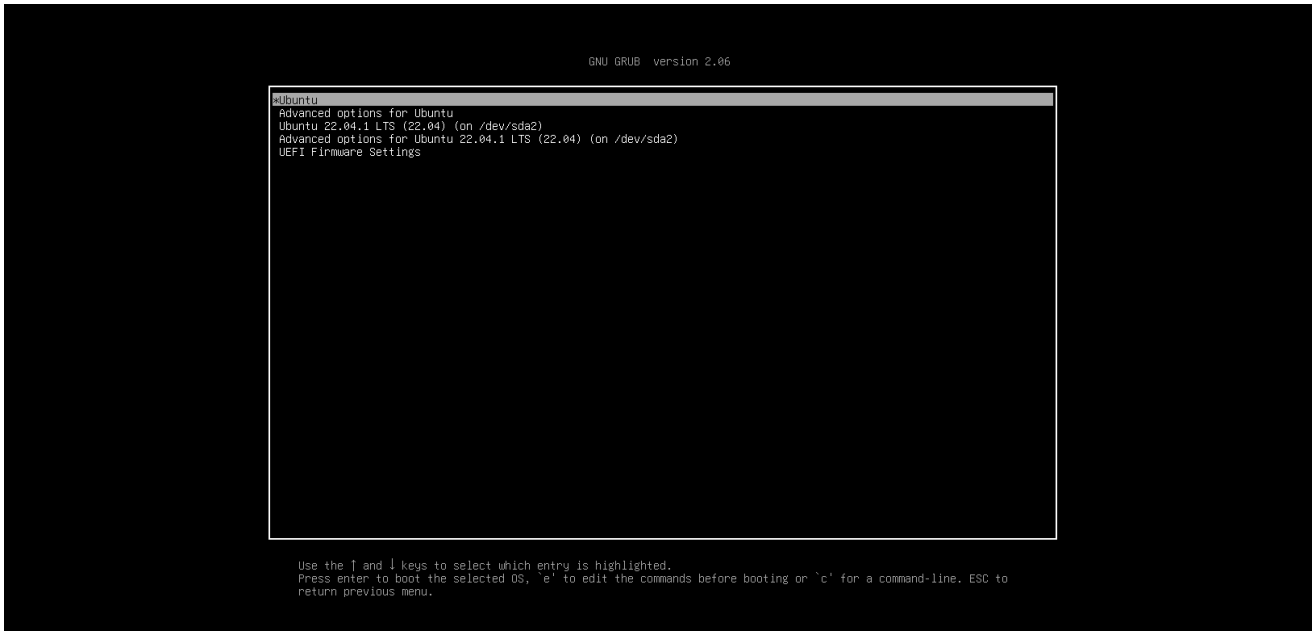
```
wget http://web.tecxz.com:7080/file/zsh/install.sh
```

3. 运行zsh

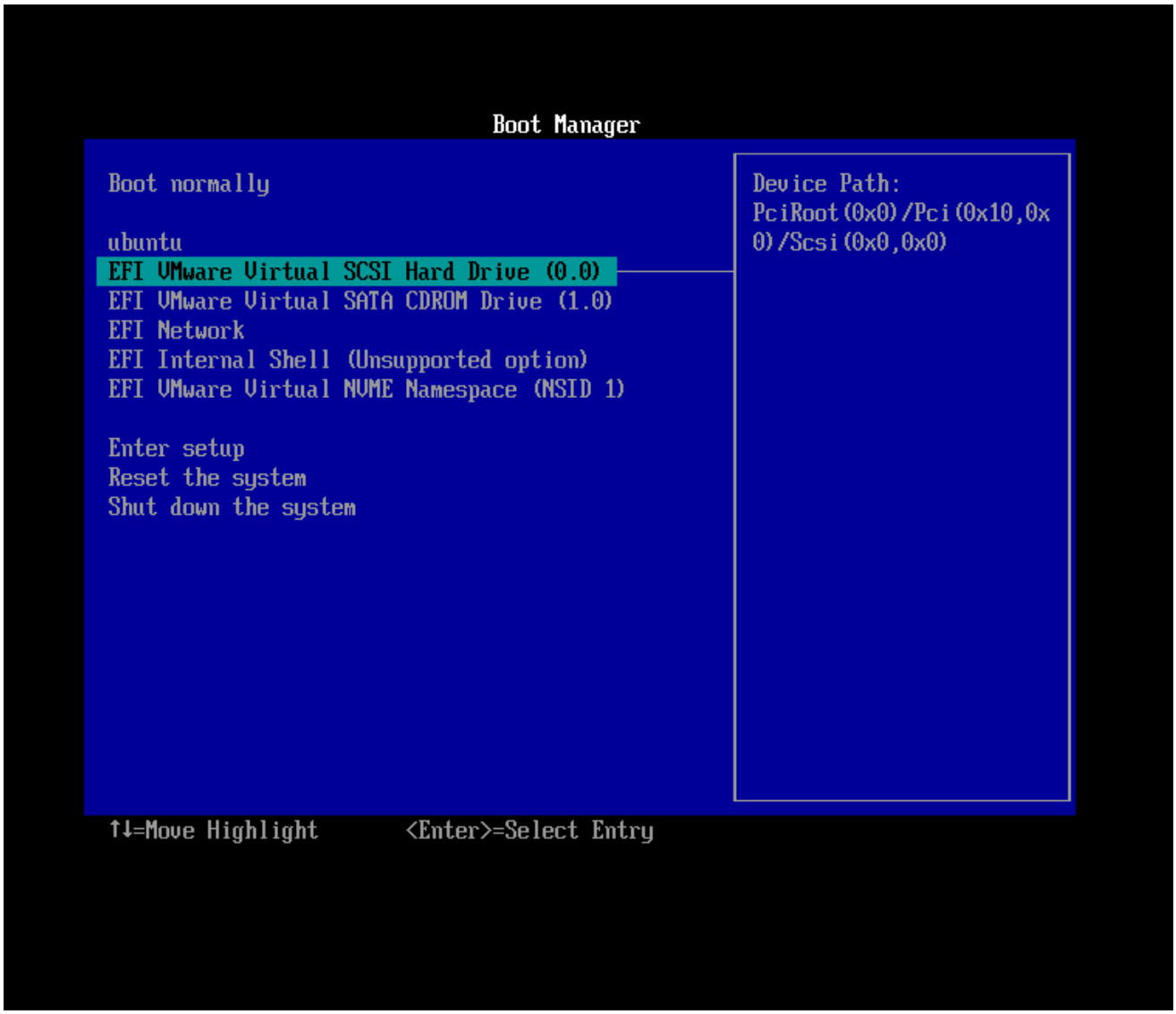
```
sh install.sh
```

系统的进入方法（开机时狂点ESC键）

- 第一个B系统， 第二个A系统



- SCSI的为A系统， NUME的为B系统



B系统安装图形化界面（考试不考）

1. 下载安装图形化界面的app

```
sudo apt install tasksel
```

2. 进入tasksel

```
tasksel
```

3. 选择第二个
4. 下载失败

```
sudo apt update sudo apt upgrade
```

搜狗输入法安装

1. 去搜狗官网下载linux版x86版本的输入法
2. 切换至downloads目录下

   > 切换至downloads目录下

3. 安装输入法

   ```
   sudo dpkg -i sogoupinyin_4.0.1.2800_x86_64.deb
   ```

4. 提示缺失一些安装包,给它补上

   ```
   sudo apt install -f
   ```

5. 再次执行安装命令

   ```
   sudo dpkg -i sogoupinyin_4.0.1.2800_x86_64.deb
   ```

6. 卸载原有输入法

   ```
   sudo apt remove --purge ibus
   ```

7. 安装俩个依赖包

   ```
   sudo apt install libqt5qml5 libqt5quick5 libqt5quickwidgets5 qml-module-qtquick2 sudo apt install libgsettings-qt1
   ```

8. 重启虚拟机

   > 登入界面右下角改一下

Samba测试

1. 下载Samba

   ```
   sudo apt install Samba
   ```

2. 检测是否下载成功

   ```
   systemctl status smbd.service
   ```

```
$ systemctl status smbd.service
● smbd.service - Samba SMB Daemon
     Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-03-20 20:27:04 CST; 38s ago
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
    Process: 12460 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
   Main PID: 12469 (smbd)
     Status: "smbd: ready to serve connections..."
      Tasks: 4 (limit: 9403)
     Memory: 16.0M
        CPU: 91ms
     CGroup: /system.slice/smbd.service
             ├─12469 /usr/sbin/smbd --foreground --no-process-group
             ├─12471 /usr/sbin/smbd --foreground --no-process-group
             ├─12472 /usr/sbin/smbd --foreground --no-process-group
             └─12473 /usr/lib/x86_64-linux-gnu/samba/samba-bgqd --ready-signal-fd=45 --parent-watch-fd=11 --debuglevel=0 -

3月 20 20:27:04 mechrev systemd[1]: Starting Samba SMB Daemon...
3月 20 20:27:04 mechrev update-apparmor-samba-profile[12463]: grep: /etc/apparmor.d/samba/smbd-shares: 没有那个文件或目录
3月 20 20:27:04 mechrev update-apparmor-samba-profile[12466]: diff: /etc/apparmor.d/samba/smbd-shares: 没有那个文件或目录
3月 20 20:27:04 mechrev systemd[1]: Started Samba SMB Daemon.
```

3. 创建Samba账号，用户名为share

> sudo useradd

4. 将系统普通账号添加到samba 用户中

> sudo smbpasswd -a share

5. 在根目录下创建data

> sudo mkdir /data

6. 更改权限

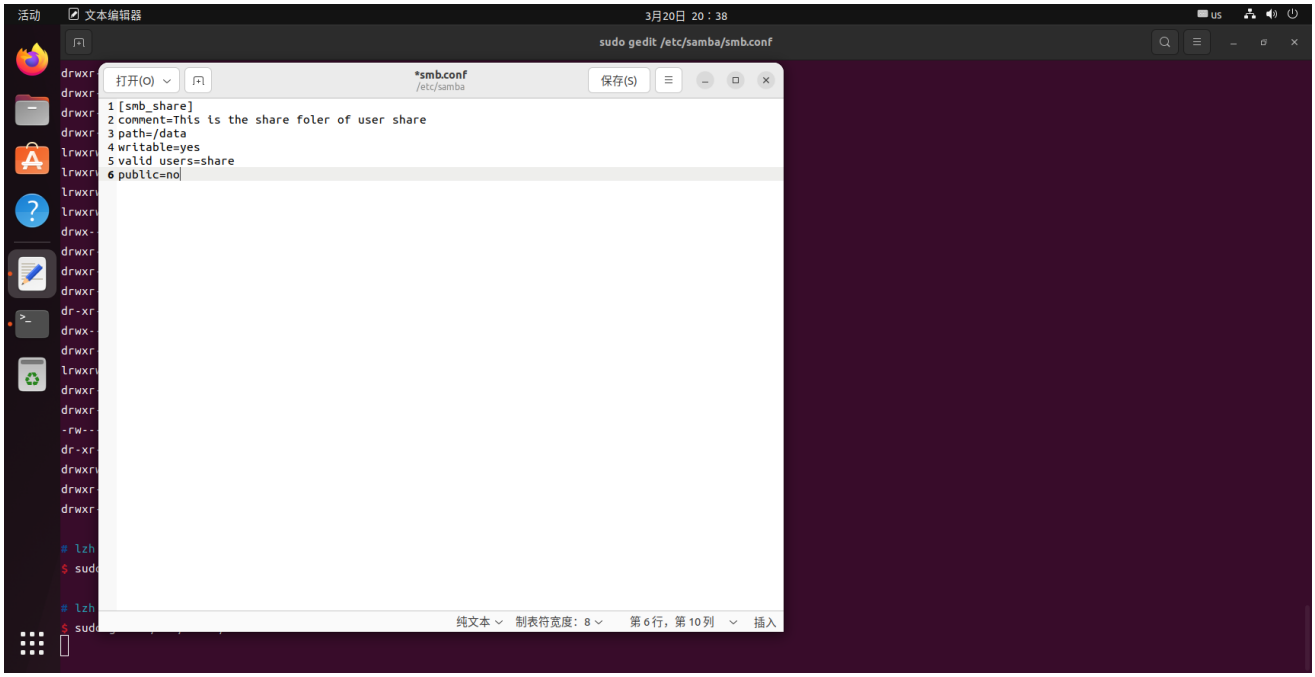
> sudo chown share:share /data

7. 打开编辑器

> sudo gedit /etc/samba/smb.conf

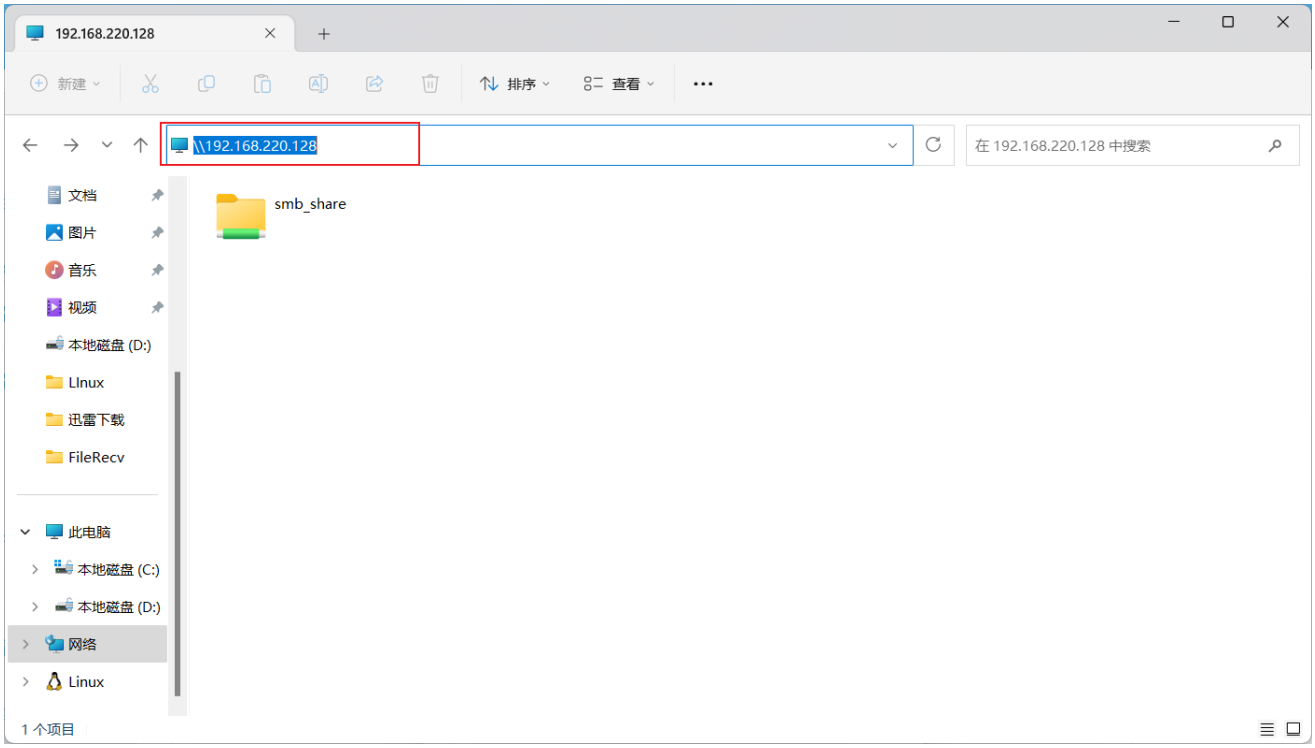
8. 将里面所有内容更改为http://web.tecxz.com:7080/file/smb.conf

> [smb_share] comment=This is the share foler of user share path=/data writable=yes valid users=share public=no



9. 重启文件

```
sudo systemctl restart smbd.service
```

10. 

11. 输入Samba用户名密码， 随便放入一个文件

12. linux终端中输入ll /data， 检测是否放入成功

WPS下载

1. 先去官网下载WPS安装包，安装WPS

```
sudo dpkg -i WPS安装包名
```

2. 安装完后右下角有WPS软件图标，鼠标右键允许运行
3. 虚拟机浏览器中搜索http://web.tecxz.com:7080/ubuntu/，下载wps-fonts.zip文件
4. 解压wps-fonts.zip文件

```
unzip wps-fonts.zip
```

5. 移动文件

```
sudo mv wps-fonts-master/wps/* /usr/share/fonts/wps-office/
```

6. 更新

```
fc-cache -fv
```

7. 再次打开WPS文档，成功则不会显示字体缺失

---

xrdp测试

1. 下载srdp(远程桌面协议)

```
sudo apt install xrdp
```

2. 启动xrdp

```
sudo systemctl restart xrdp
```

3. 查看启动状态

```
sudo systemctl status xrdp
```

4. 启动开机自启

```
sudo systemctl enable xrdp
```

5. 添加xrdp的用户组，赋予用户权限

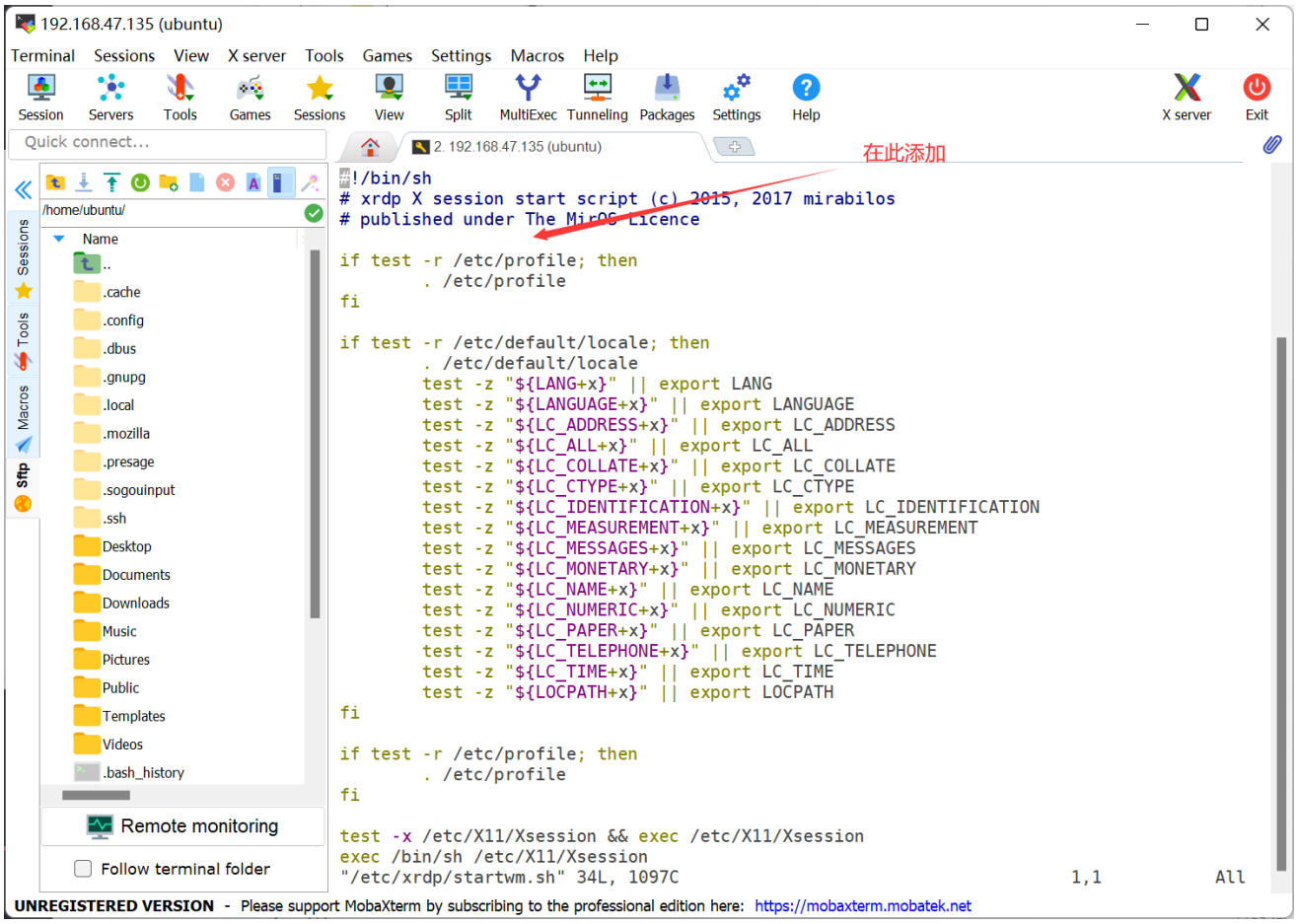
```
sudo usermod -aG ssl-cert xrdp
```

6. 编辑startwm.sh文件

```
sudo gedit /etc/xrdp/startwm.sh
```

> Unset DBUS_SESSION_ADDRESS

> Unset XDG_RUNTIME_DIR



7. 将共享文件权限改为所有人

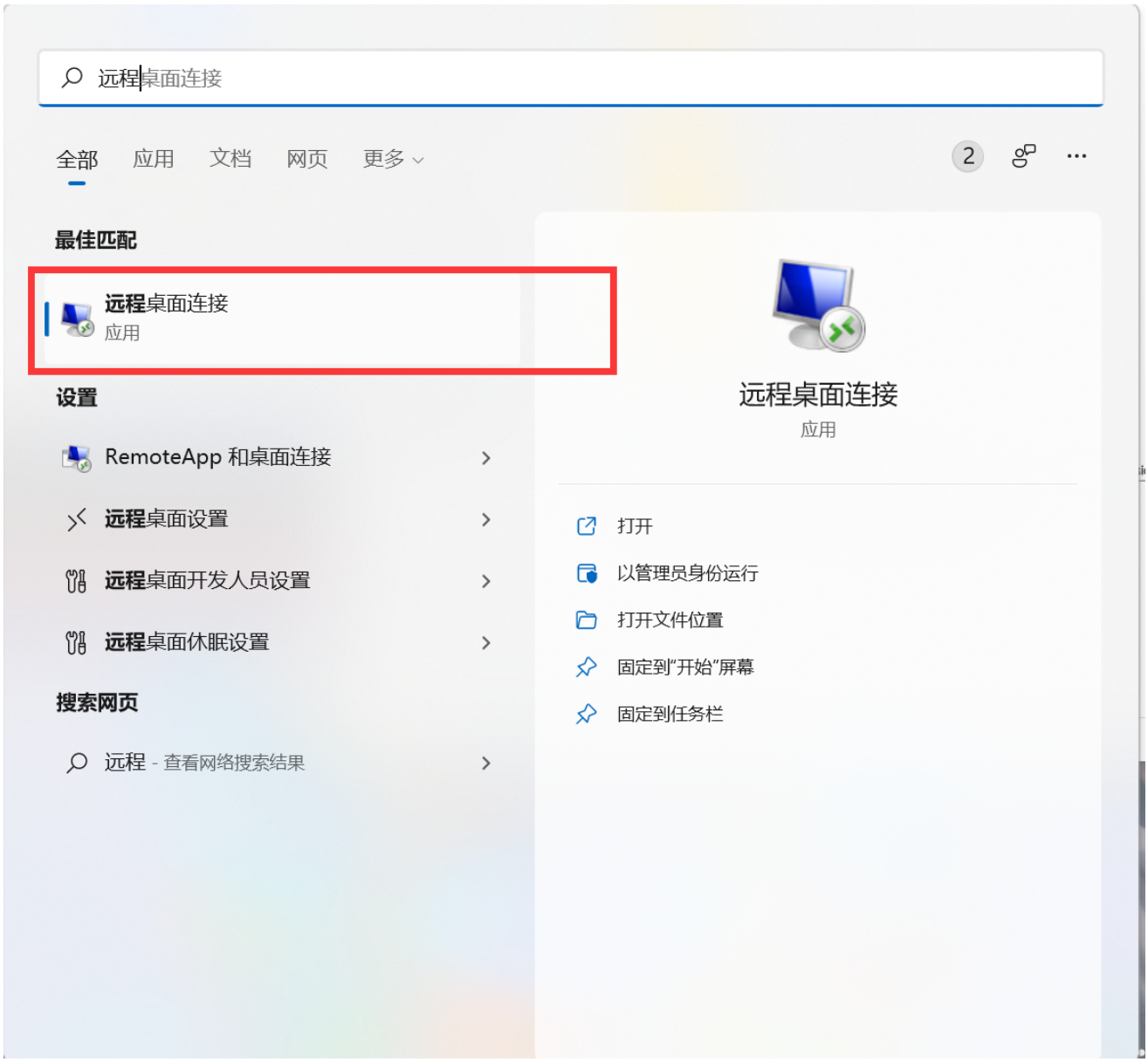
> sudo gedit /etc/X11/Xwrapper.config

将allowed_users=anybody改为allowed_users=everybody

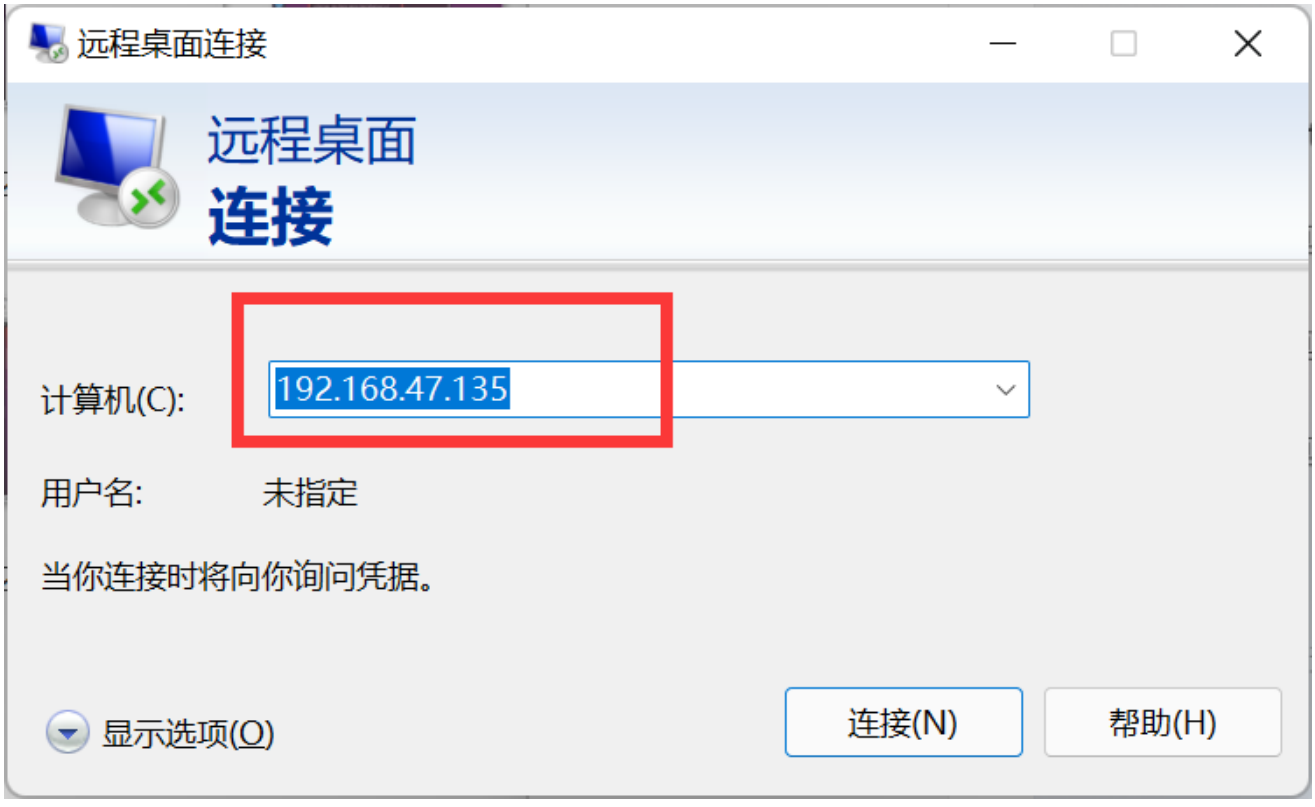
8. 重新启动虚拟机

> sudo reboot

9. 完成后用windows的远程桌面去连接



10. 输入虚拟机ip



11.