



中南大學  
CENTRAL SOUTH UNIVERSITY

# 分布式系统与云计算 实验报告

学生姓名：王云鹏  
学 院：计算机学院  
专业班级：物联网工程 1802 班  
指导教师：李玺

2021 年 12 月

## 目录

实验一 数据包 socket 应用.....	3
一、 实验目的.....	3
二、 实验内容和相应结果.....	3
1. 服务器端.....	3
2. 客户端.....	4
实验十三 虚拟机的使用与Linux 系统的安装.....	5
一、 实验目的.....	5
二、 实验内容和相应结果.....	5
1. 下载软件.....	5
2. VMware 安装 ubuntu.....	5
3. 安装后的问题解决.....	19
实验十四 Hadoop 的安装与部署.....	27
一、 实验目的.....	27
二、 实验内容和相应结果.....	27
1. 安装 JDK.....	27
2. 设置目录并安装 Hadoop.....	27
3. 单机工作模式.....	28
实验七 Internet 应用.....	30
1、 学习 Tomcat.....	30
2、 安装 Tomcat.....	30
3、 实现第一个任务.....	30
4、 实现第二个任务.....	32
5、 实现第三个任务.....	32
实验心得与感想.....	33

# 实验一 数据包 socket 应用

## 一、实验目的

1. 理解数据包 socket 的应用
2. 实现数据包 socket 通信
3. 了解 JAVA 并行编程的基本方法

## 二、实验内容和相应结果

### 1. 服务器端

创建一个 DatagramSocket 类的实例 socket，选定本机端口 40096，并与本机默认 IP 地址进行绑定。标志该 socket 所在程序为服务器端

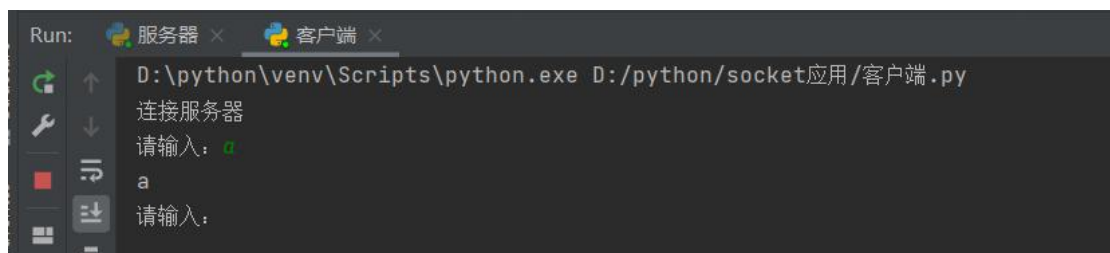
```
DatagramSocket socket = new DatagramSocket (40000)
```

服务器端接收客户端数据。在 Socket 接收数据前，先创建一个 DatagramPacket 类的实例对象 inPacket 用于接收数据，可接收长度为 inbuff.length 的 packets。并调用 receive 方法等待数据报到来。并将缓冲区的接收数据其转化为字符串打印输出。

```
DatagramPacket inPacket = new DatagramPacket ( inBuff , inBuff . length ) ;  
socket . receive ( inPacket ) ;
```

服务器端发送收到给客户端。创建一个 DatagramPacket 类的对象 outPacket，将一个标识服务器端成功接收信息的字符串放入其缓冲区，socket 调用 send 方法将 outPacket 对象发送给客户端，这里通过 DatagramPacket 的 getSocketAddress() 方法接受的客户端的数据报 inPacket 携带的 IP 地址及端口号，以此为目标将数据报发送给客户端。

```
String message = " Server received message !";  
byte [] sendData = message . getBytes () ;  
DatagramPacket outPacket = new DatagramPacket ( sendData , sendData .  
length , inPacket . getSocketAddress () ) ;  
socket . send ( outPacket ) ;
```



## 2. 客户端

客户端的实现与服务器端类似，主要不同在于客户端的 socket 不指定端口，增加了从键盘接收输入的部分。首先创建一个 DatagramSocket 类的实例 socket 作为客户端，并将其与任意一个有效的端口绑定。

```
DatagramSocket socket = new DatagramSocket ();
```

接着创建一个 DatagramPacket 类的实例 outPacket 用于发送数据报，这里设置 IP 地址为本地机器，端口号为先前服务器的端口号。

```
DatagramPacket outPacket = new DatagramPacket ( new byte [0] ,0 ,  
InetAddress . getByName ( " 127.0.0.1 " ) ,40000 );
```

读写键盘数据。通过读入键盘输入并将其放入 outPacket 实例中，通过调用 DatagramSocket 类的 send 方法将内容发送到服务器端。

```
Scanner scanner = new Scanner ( System . in );  
while ( scanner . hasNextLine () ) {  
byte [] buff = scanner . nextLine () . getBytes () ;  
outPacket . setData ( buff ) ;  
socket . send ( outPacket ) ;  
}
```

最后接收服务器端发送的收到信息，通过构建一个 DatagramPacket 类的实例 inPacket 用于读取 socket 实例中的内容，并打印输出

```
byte [] inBuff = new byte [8000];  
DatagramPacket inPacket = new DatagramPacket ( inBuff , inBuff . length ) ;  
socket . receive ( inPacket ) ;  
System . out . println ( new String ( inBuff , 0 , inPacket . getLength () ) ) ;
```



# 实验十三 虚拟机的使用与 Linux 系统的安装

## 一、实验目的

- (1) 理解虚拟机软件的工作原理与运行机制；
- (2) 掌握 VMware Workstation 的下载、安装、配置与使用；
- (3) 掌握 Linux 系统 (Ubuntu) 的配置与使用方法。

## 二、实验内容和相应结果

### 1. 下载软件

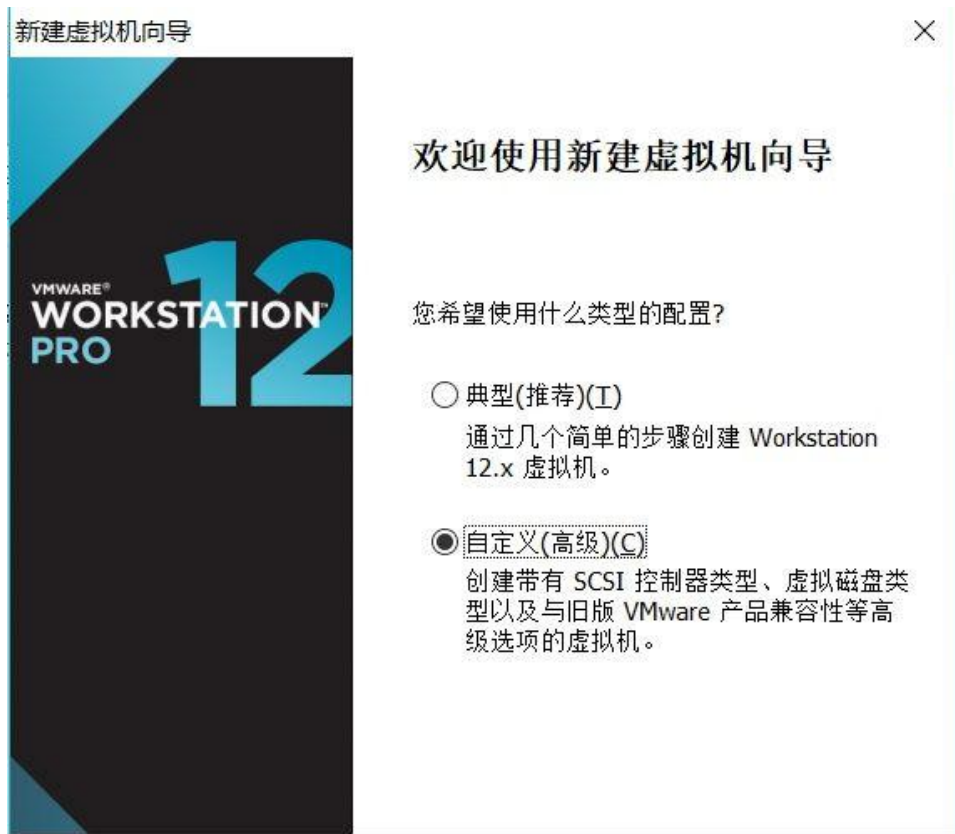
首先需要下载 VMware Workstation Pro 12 虚拟机。

还有 ubuntu 镜像，我这里下载的是 Ubuntu 18.04，下载地址在 ubuntu 官网：  
<https://www.ubuntu.com/download/desktop>

### 2. VMware 安装 ubuntu

打开安装好的虚拟机，创建新的虚拟机。





接下来这步一定先不要选择下载的光映像文件，选了点“下一步”系统就直接帮你配置好后面的配置，要根据自己的要求进行选择配置的虚拟机，所以这步选择“稍后安装操作系统”。

## 安装客户机操作系统

虚拟机如同物理机，需要操作系统。您将如何安装客户机操作系统？

安装来源：

☐ 安装程序光盘(D):

无可用驱动器

☐ 安装程序光盘映像文件(iso)(M):

C:\Users\Xian\Downloads\ubuntu-18.04-desktop-amd

浏览(R)...

☒ 稍后安装操作系统(S)。

创建的虚拟机将包含一个空白硬盘。

## 选择客户机操作系统

此虚拟机中将安装哪种操作系统？

客户机操作系统

☐ Microsoft Windows(W)

☒ Linux(L)

☐ Apple Mac OS X(M)

☐ Novell NetWare(E)

☐ Solaris(S)

☐ VMware ESX(X)

☐ 其他(O)

版本(V)

Ubuntu 64 位

新建虚拟机向导

命名虚拟机

您要为此虚拟机使用什么名称?

虚拟机名称(V):

Ubuntu

位置(L):

C:\Users\Xian\Documents\Virtual Machines\Ubuntu

浏览(B)...

在“编辑”>“首选项”中可更改默认位置。

新建虚拟机向导

处理器配置

为此虚拟机指定处理器数量。

处理器

处理器数量(P):

1

每个处理器的核心数量(C):

4

总处理器核心数量:

4



这步配置给虚拟机的内存,这个要根据自己的电脑配置给虚拟机分内存,内存大,肯定性能好,但是别到时候给虚拟机的内存大了,影响自己主电脑的使用。

新建虚拟机向导

✕

此虚拟机的内存

您要为此虚拟机使用多少内存?

指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。

64 GB

32 GB

16 GB

8 GB

4 GB

2 GB

1 GB

512 MB

256 MB

128 MB

64 MB

32 MB

16 MB

8 MB

4 MB

此虚拟机的内存(M):

2048

MB

最大推荐内存:

6308 MB

推荐内存:

1024 MB

客户机操作系统最低推荐内存:

512 MB

新建虚拟机向导

✕

网络类型

您要添加哪类网络?

网络连接

☐ 使用桥接网络(R)

为客户机操作系统提供直接访问外部以太网网络的权限。客户机在外部网络上必须有自己的 IP 地址。

☒ 使用网络地址转换(NAT)(E)

为客户机操作系统提供使用主机 IP 地址访问主机拨号连接或外部以太网网络连接的权限。

☐ 使用仅主机模式网络(H)

将客户机操作系统连接到主机上的专用虚拟网络。

☐ 不使用网络连接(I)

**选择 I/O 控制器类型**

您要使用何种类型的 SCSI 控制器？

## I/O 控制器类型

SCSI 控制器：

- ☐ BusLogic(U) (不适用于 64 位客户机)
- ☒ LSI Logic(L) (推荐)
- ☐ LSI Logic SAS(S)

**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)

**选择磁盘**

您要使用哪个磁盘？

**磁盘**☒ **创建新虚拟磁盘(V)**

虚拟磁盘由主机文件系统上的一个或多个文件组成，客户机操作系统会将其视为单个硬盘。虚拟磁盘可在一台主机上或多台主机之间轻松复制或移动。

☐ **使用现有虚拟磁盘(E)**

选择此选项将重新使用之前配置的磁盘。

☐ **使用物理磁盘(适用于高级用户)(P)**

选择此选项将为虚拟机提供直接访问本地硬盘的权限。

**指定磁盘容量**

磁盘大小为多少？

最大磁盘大小(GB)(S):

20.0

针对 Ubuntu 64 位 的建议大小: 20 GB

☐ **立即分配所有磁盘空间(A)。**

分配所有容量可以提高性能，但要求所有物理磁盘空间立即可用。如果不立即分配所有空间，虚拟磁盘的空间最初很小，会随着您向其中添加数据而不断变大。

☒ **将虚拟磁盘存储为单个文件(O)**☐ **将虚拟磁盘拆分成多个文件(M)**

拆分磁盘后，可以更轻松地计算机之间移动虚拟机，但可能会降低大容量磁盘的性能。

## 新建虚拟机向导



### 已准备好创建虚拟机

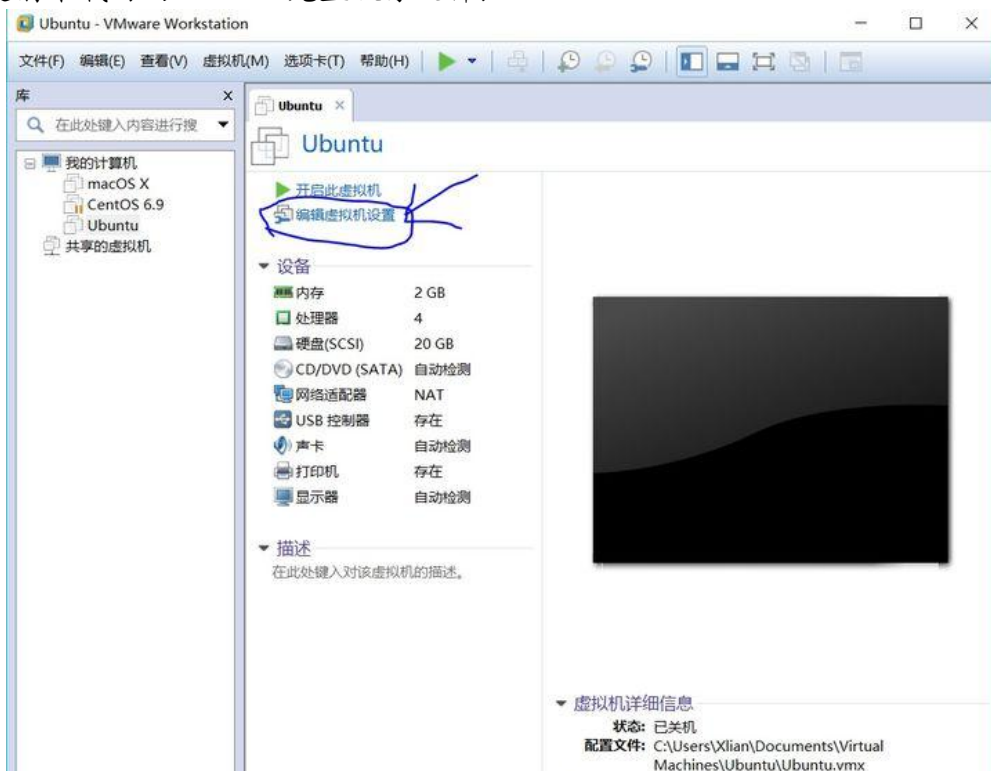
单击“完成”创建虚拟机。然后可以安装 Ubuntu 64 位。

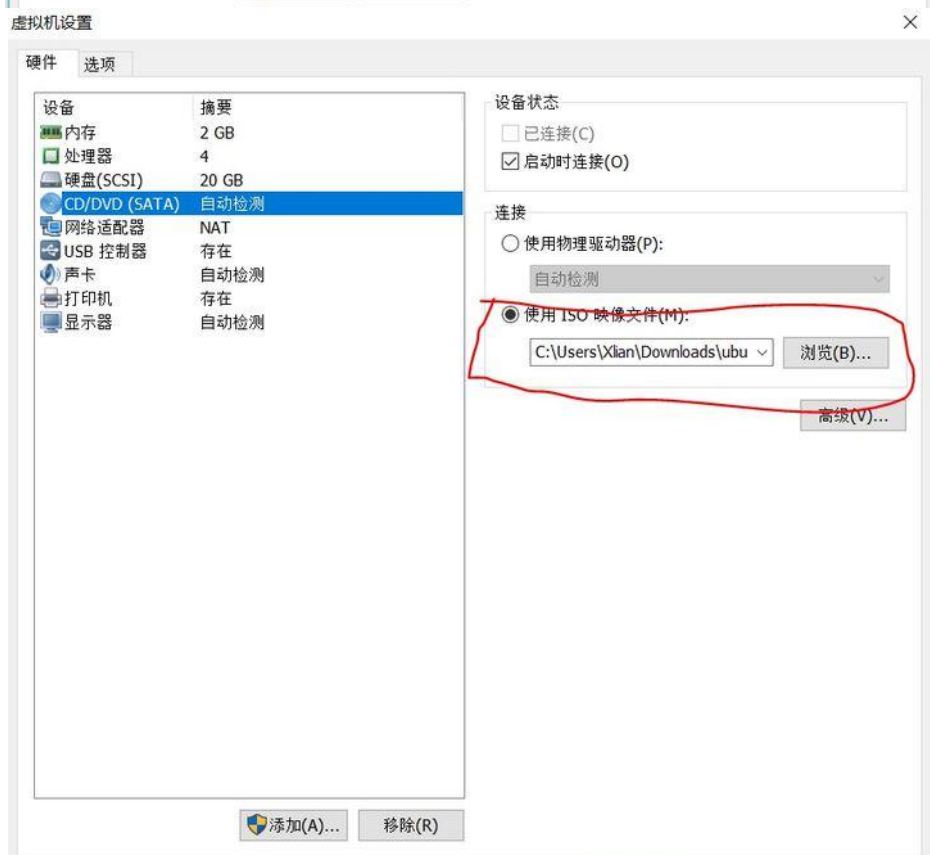
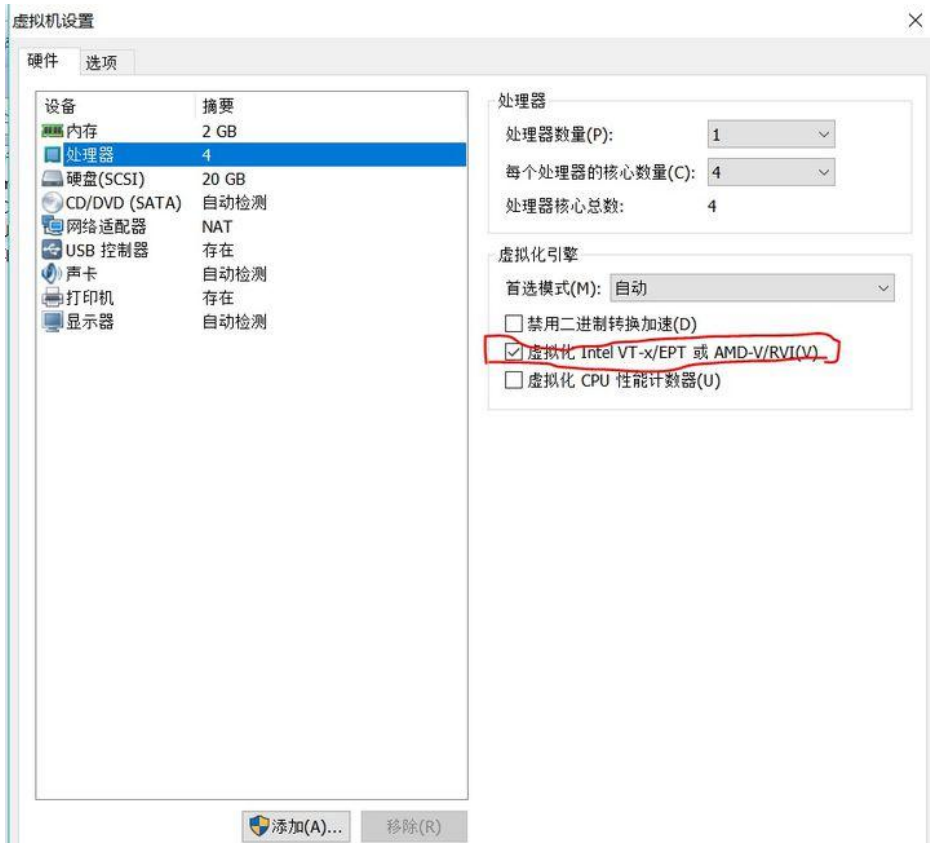
将使用下列设置创建虚拟机:

名称:	Ubuntu
位置:	C:\Users\Xlian\Documents\Virtual Machines\Ubuntu
版本:	Workstation 12.x
操作系统:	Ubuntu 64 位
硬盘:	20 GB
内存:	2048 MB
网络适配器:	NAT
其他设备:	4 个 CPU 核心, CD/DVD, USB 控制器, 打印机, 声卡

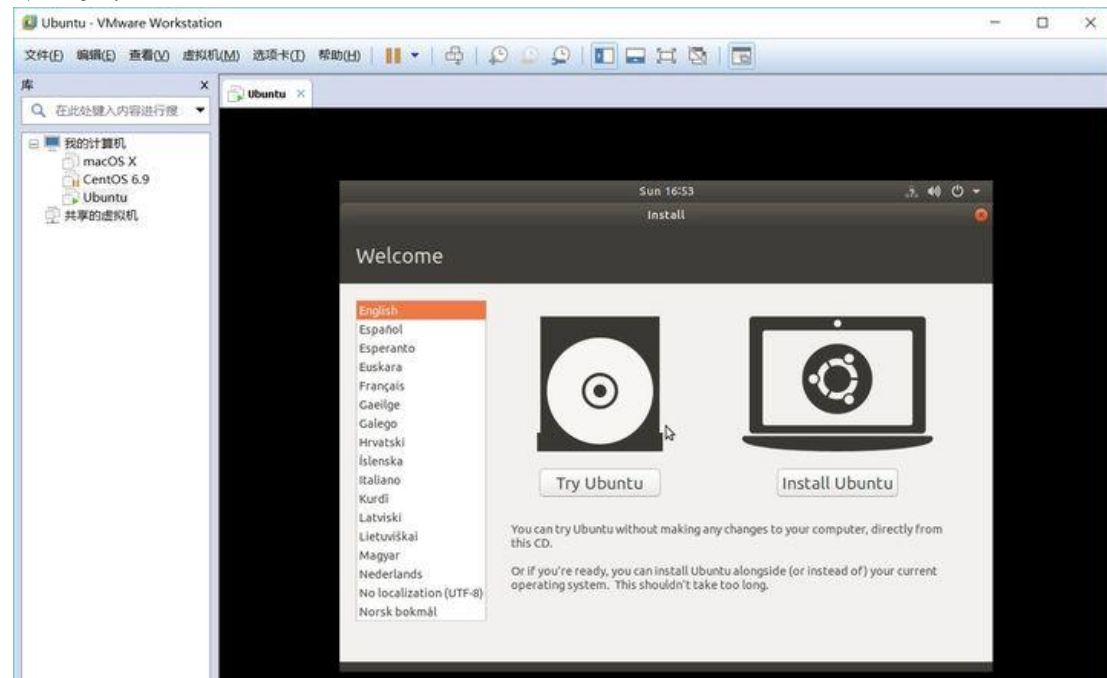
自定义硬件(C)...

在这里虚拟机就创建完成了,下面打开虚拟机设置,设置安装镜像文件的位置(就是之前下载好的 Ubuntu 光盘映像文件)

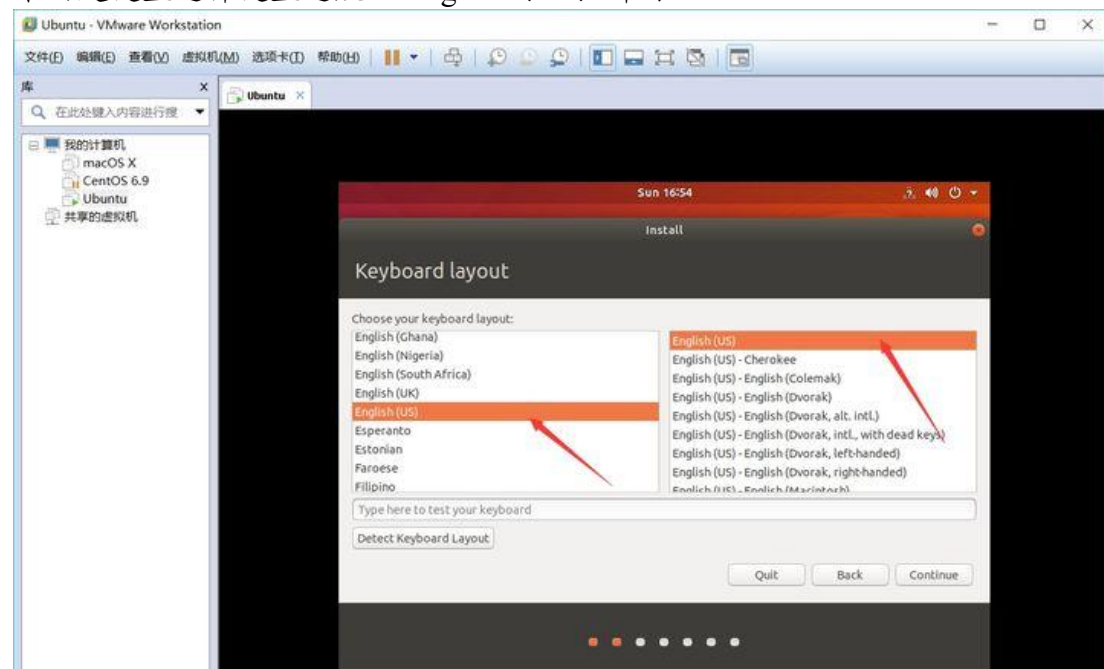




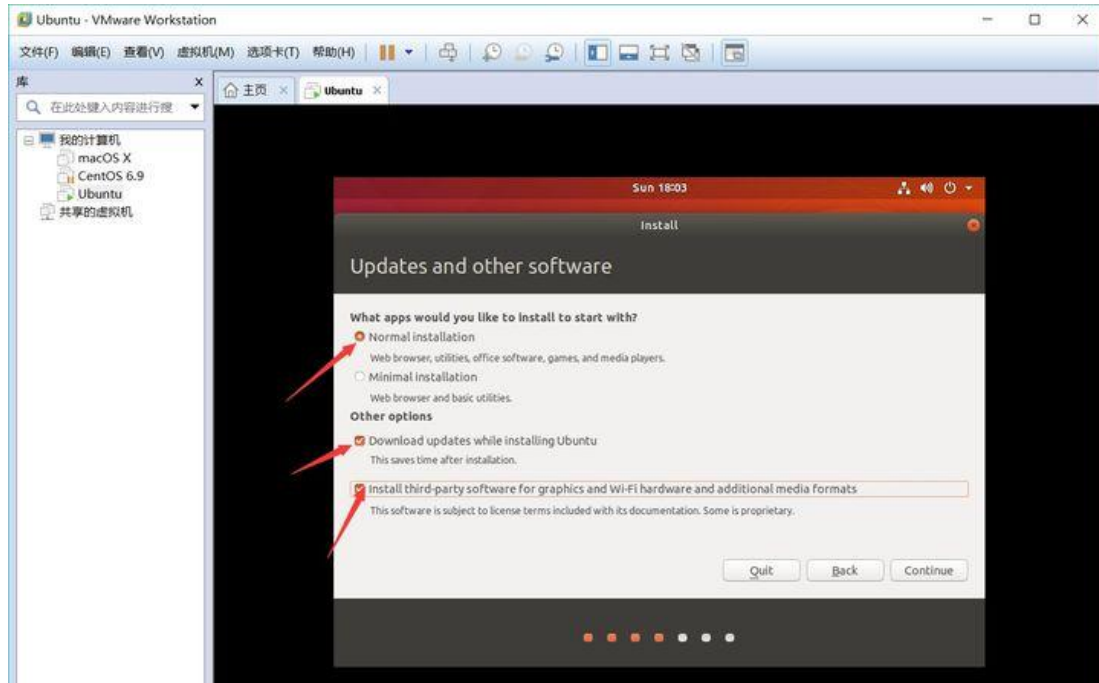
设置好后点击“开启此虚拟机”，在正常安装系统时，要在 BIOS 里设置启动项，把电脑设置为光盘启动，但是这个虚拟机他可以自己识别这台虚拟机有没有安装系统，所以启动时他就可以直接进入系统安装界面。下图，关于语言选择，其实比较推荐选择英文，如果选择中文，安装完成后桌面、文档等文件夹的名字会成为中文，在终端操作时会有不便，使用命令行报错也会显示中文，便于阅读问题，但是不利于进行搜索。进入安装界面后选择“English”，点击 install ubuntu 进入下一步；



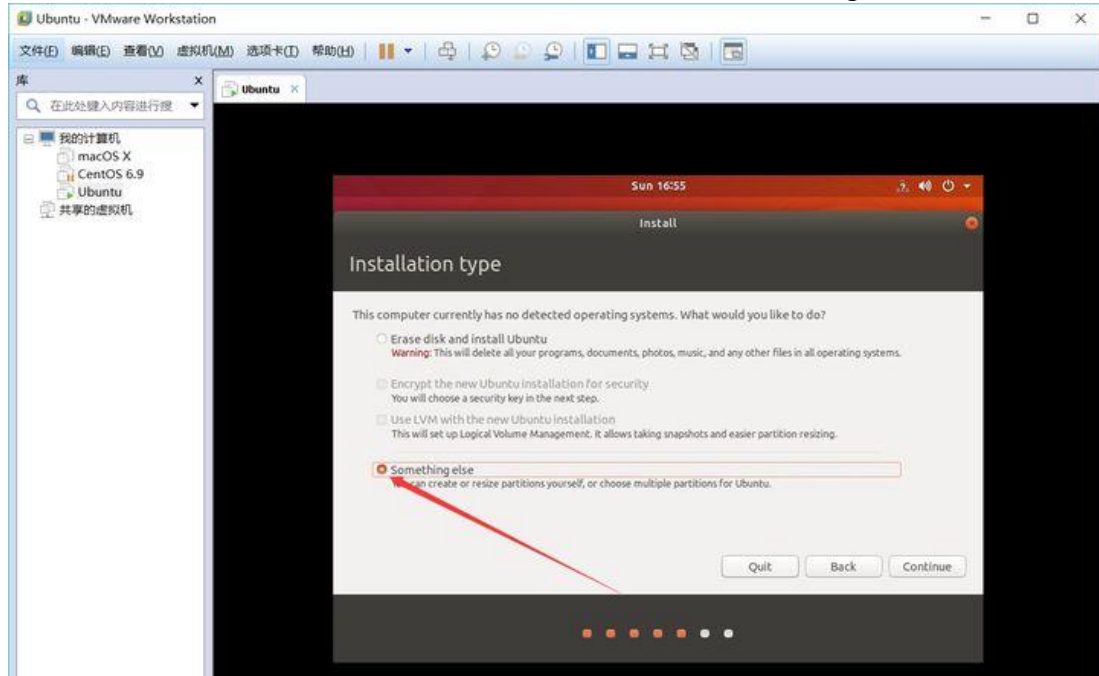
下面是键盘选择键盘选默认 English (US) 即可



下面这两个选项第一个选择的是正常安装，这个安装完后就是会出现一些系统自带的软件，不用的可以卸载掉，第二个选项是最小安装他只会安装系统要用到的基本工具。下面两个都勾选上下面第一个是安装 ubuntu 时下载更新，第二个为图形或无线硬件以及其他的媒体格式安装第三方软件。我这里选择的是正常安装，如果想要一个纯净的系统就可以选择最小安装。

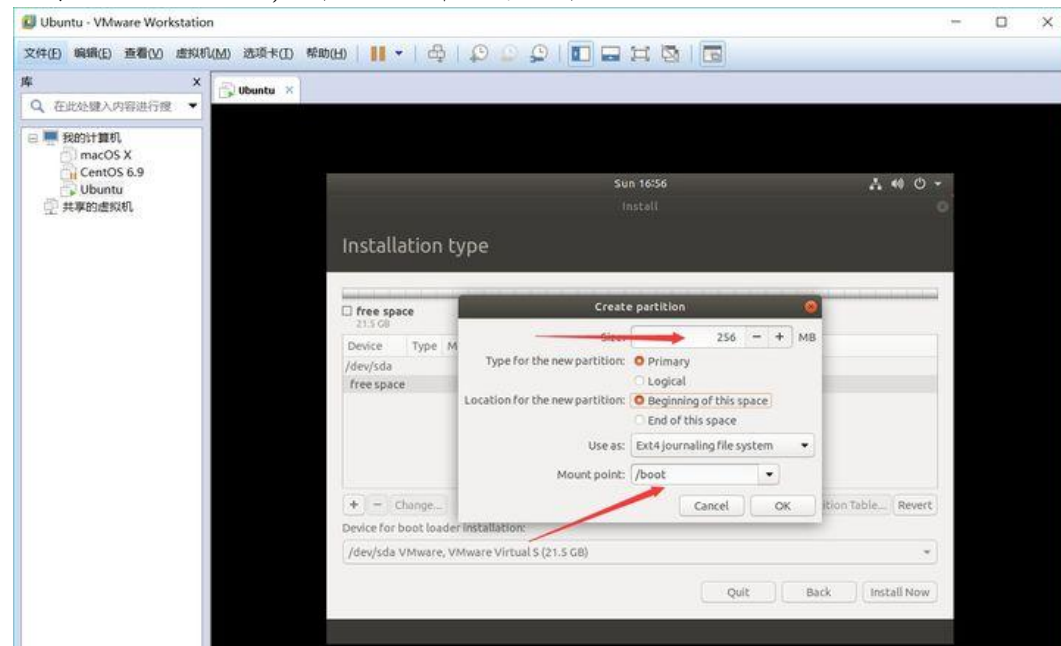


这里是给磁盘分区，进行手动分区，选择最后一个“something else”（其他选项）

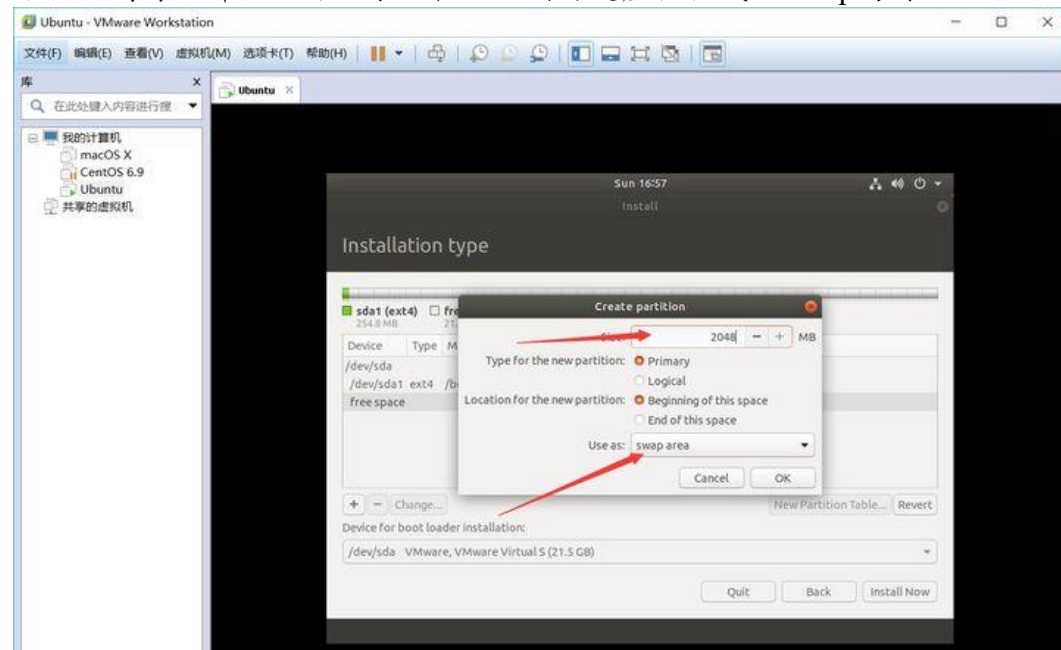




进来后进行分区，先分一个启动分区 boot，一般启动分区的大小为 200M 就可以了。boot 包含了操作系统的内核和在启动系统过程中所要用到的文件，要给他单独分出来。要是不分当磁盘文件写满了磁盘空间不足时或者根分区出现问题了，的系统就没法启动，所以给他单独分出来。

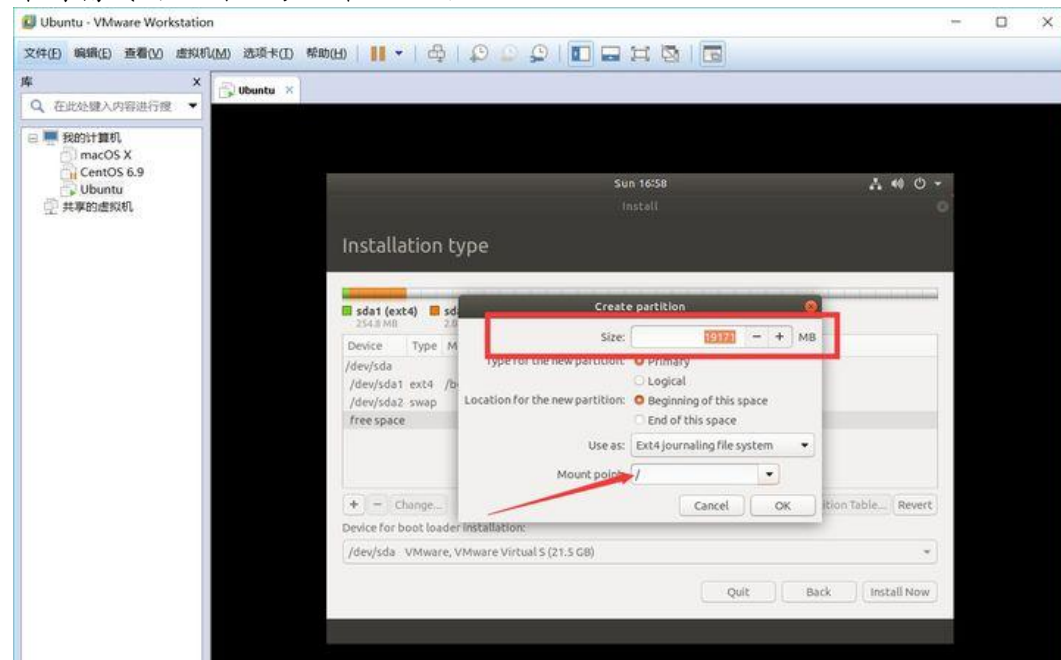


boot 分区完成后分一个交换分区“swap”，swap 分区，在 windows 中就是虚拟内存，在 liunx 中如果没有 swap 分区，系统内存用完后，系统他会杀死一部分程序，这个在使用中是非常可怕的，你正在用电脑，电脑内存满了，你的程序莫名其妙被关闭了，这是绝对不能允许的，所以必须分一个 swap 分区。还有一点 swap 分区的大小，一般电脑的内存小于等于 4G 时，swap 给他是电脑内存的 1.5-2 倍；大于 4G 时，电脑内存多少 swap 就给多少。比如我的电脑内存为 4G，那么 swap 就给 6G=6144M-8G=8192；我的电脑内存为 8G,16G...的 swap 就给 8G,16G...跟电脑内存一样大就可以了。在这里因为是虚拟机我给 swap 分了 2G。






下面给分最后一个分区“/”（根分区），根分区就是系统根目录所在的分区，一般根目录下面只有目录，不要直接有文件。Linux 只有一个根目录，就是“/”，其它目录都是它的子目录。这里把剩下的磁盘空间都给“/”就可以了。（这里说一下“/home”用户的 home 目录所在地，这个分区的大小取决于有多少用户。如果是多用户共同使用一台电脑的话，这个分区是完全有必要的，额外分割出/home 有个最大的好处，当你重新安装系统时，你不需要特别去备份你的个人文件，只要在安装时，选择不要格式化这个分区，重新挂载为/home 就不会丢失你的数据。因为是虚拟机，这台虚拟机是自己使用的，所以/home 没有分出来，如果有需要可以自己分一个/home）



接下来就是输入你的位置，选择上海就可以了



下面是设置用户名密码，“Your name”你的名字，“Your computer's name”你的计算机名，“Pick a username”你的用户名，下面是你的用户名密码“choose a password”确认密码，“confirm your password”确认密码，下面选择自动登录或者登陆时需要密码就可以了。



The image shows the 'Who are you?' screen in the Ubuntu installer. The window title is 'Install' and the time is 'Mon 01:00'. The screen has a dark header with the text 'Who are you?'. Below this, there are several input fields and checkboxes. The 'Your name:' field contains 'liunx\_one' with a green checkmark. The 'Your computer's name:' field contains 'liunx-one-PC' with a green checkmark. Below this is a small text 'The name it uses when it talks to other computers.'. The 'Pick a username:' field contains 'liunx\_one' with a green checkmark. The 'Choose a password:' field contains four dots and is labeled 'Short password'. The 'Confirm your password:' field contains four dots with a green checkmark. At the bottom, there are two radio buttons: 'Log in automatically' (unselected) and 'Require my password to log in' (selected). At the very bottom, there are 'Back' and 'Continue' buttons.

Mon 01:00

Install

Who are you?

Your name: liunx\_one ✓

Your computer's name: liunx-one-PC ✓

The name it uses when it talks to other computers.

Pick a username: liunx\_one ✓

Choose a password: ●●●● Short password

Confirm your password: ●●●● ✓

☐ Log in automatically

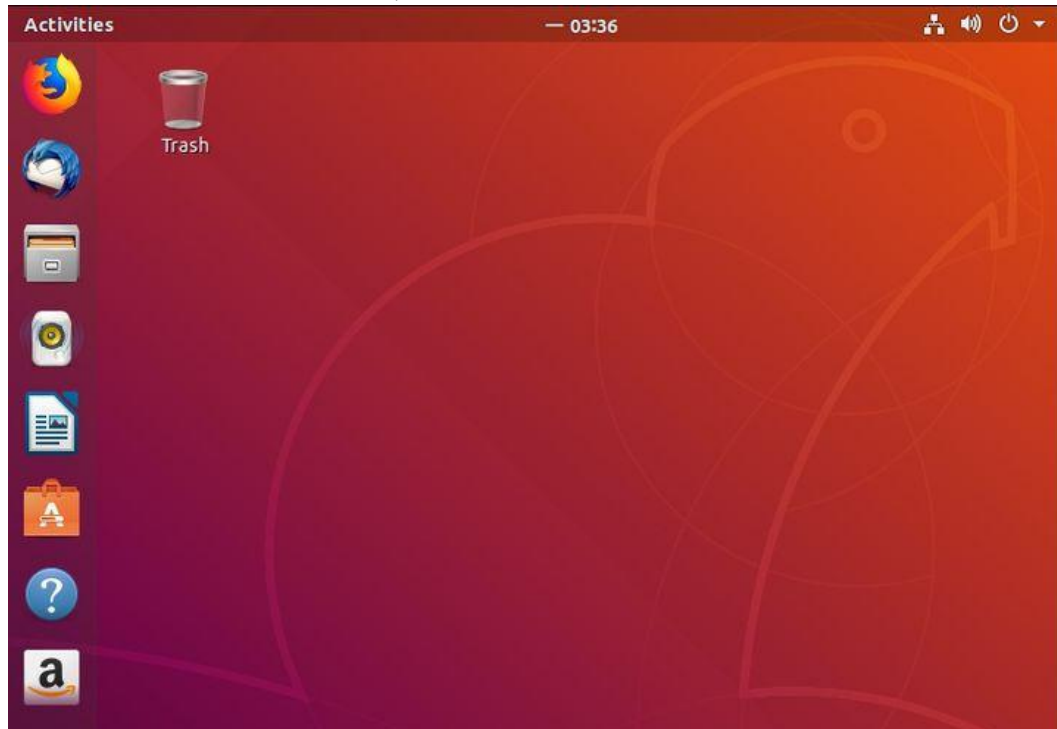
☒ Require my password to log in

Back Continue

以上配置好后就正是安装了，接下就是等，可以去喝喝茶，等待安装完成。



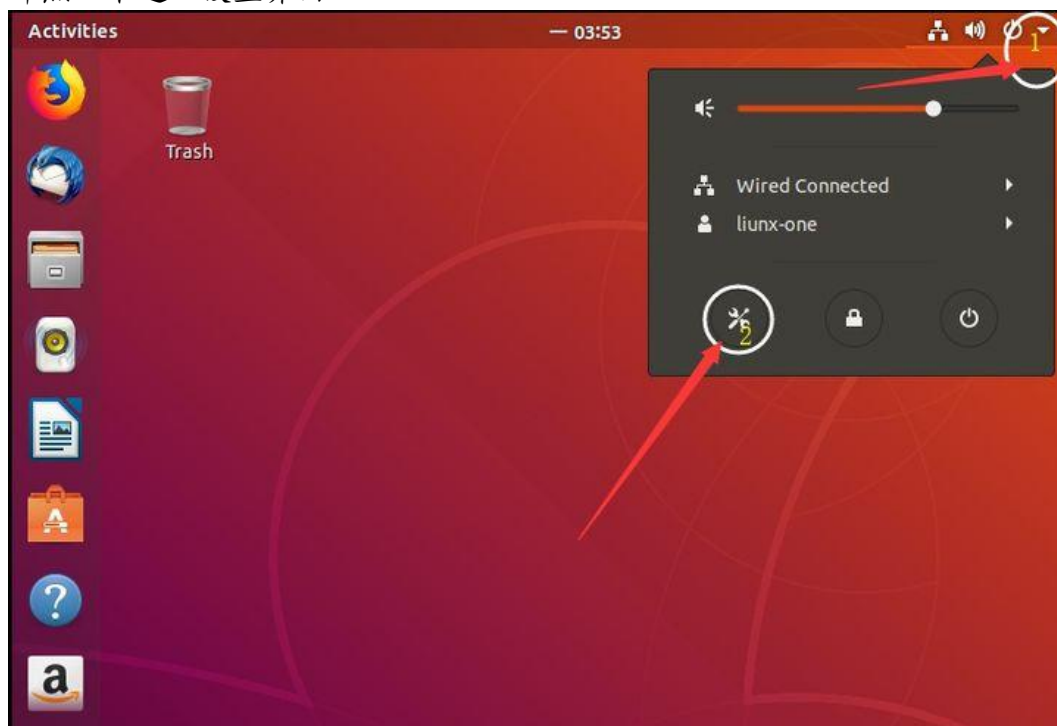
安装完成后会提示重新启动，启动后输入用户名密码就进入 ubuntu 桌面了。



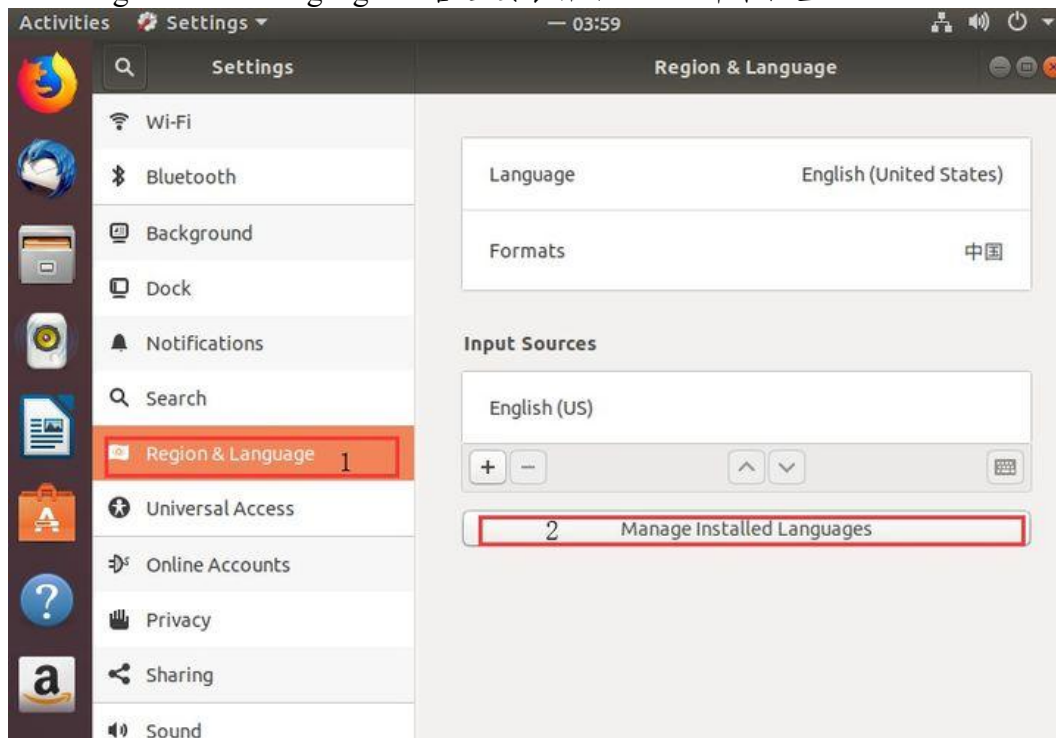
### 3. 安装后的问题解决

#### 设置语言环境

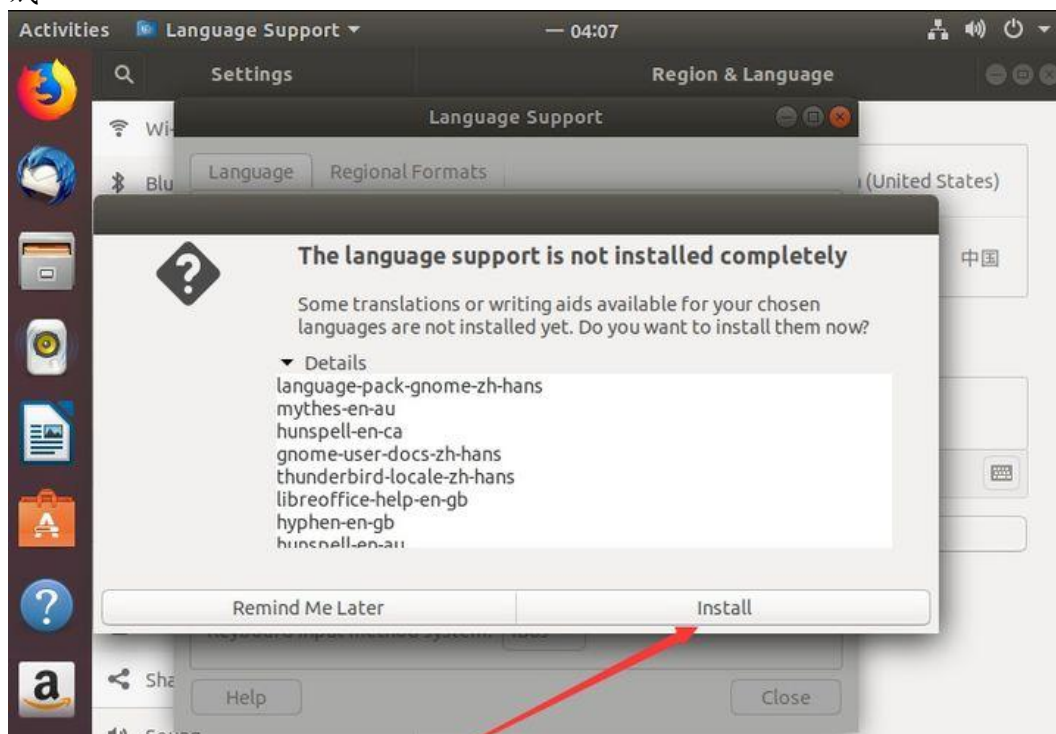
安装时语言选择是英文，那有的人他就用不惯，接下来说一下怎么把 ubuntu 从英文环境切换到中文环境。先在 ubuntu 桌面右上角“1”那点一下，再再“2”那点一下进入设置界面



进入设置界面后点击“Region&language”语言区域“1”那个位置，然后点击“Manage installed languages”管理安装语言“2”那个位置



点击“Manage installed languages”后会弹出一个窗口，他提示有语言支持需要安装，点击安装“install”，系统会提示输入用户密码，输入密码后就等待安装完成

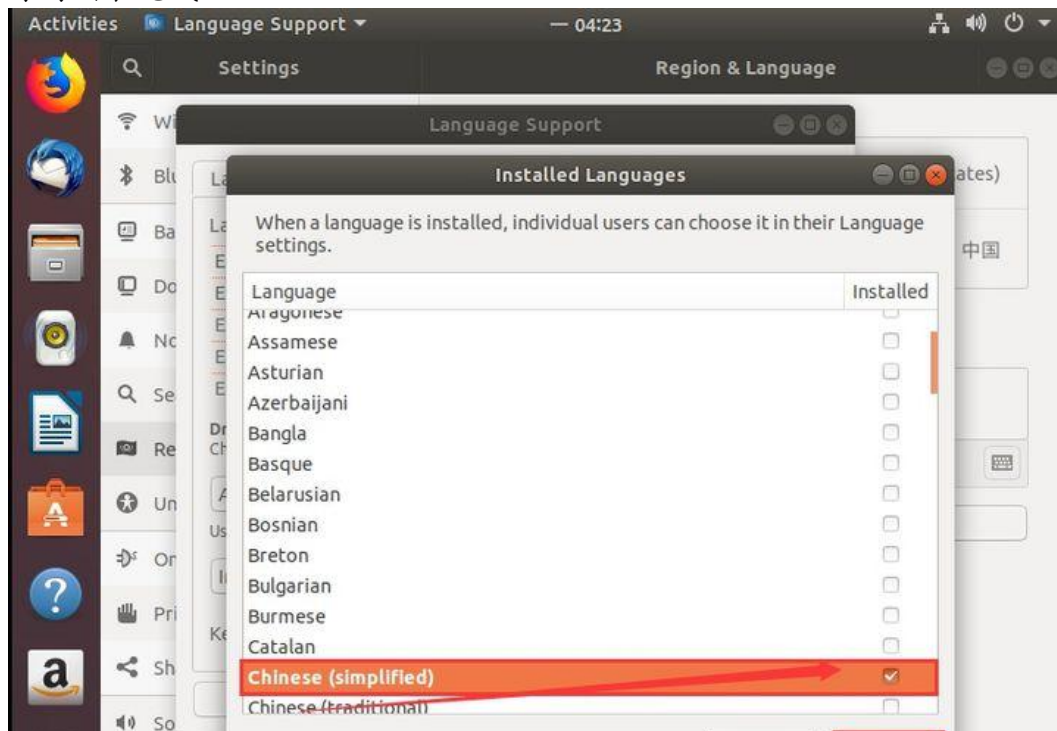




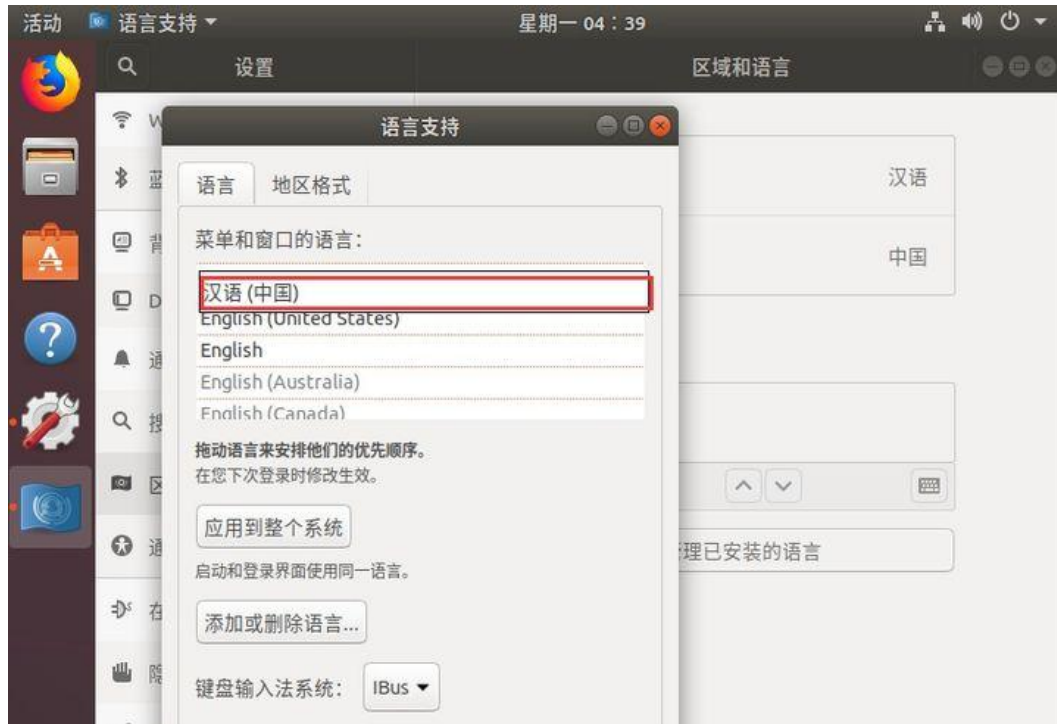
安装完成后会弹出一个窗口下面这张图，选择安装或卸载语言这个选项



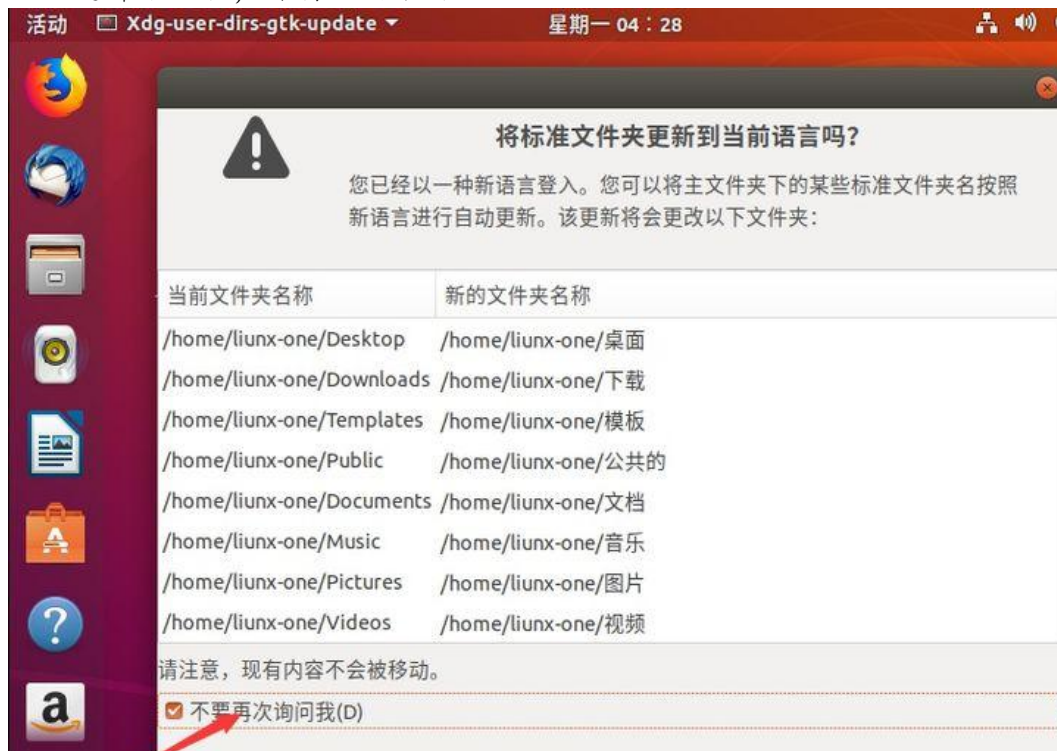
点击后会弹出下面这个窗口，找到简体中文这个选项在后面打勾，点击 Apply，等待安装完成



安装完成后，在菜单和窗口语言里找到“汉语（中国）”拖到第一个位置，然后重启 ubuntu。

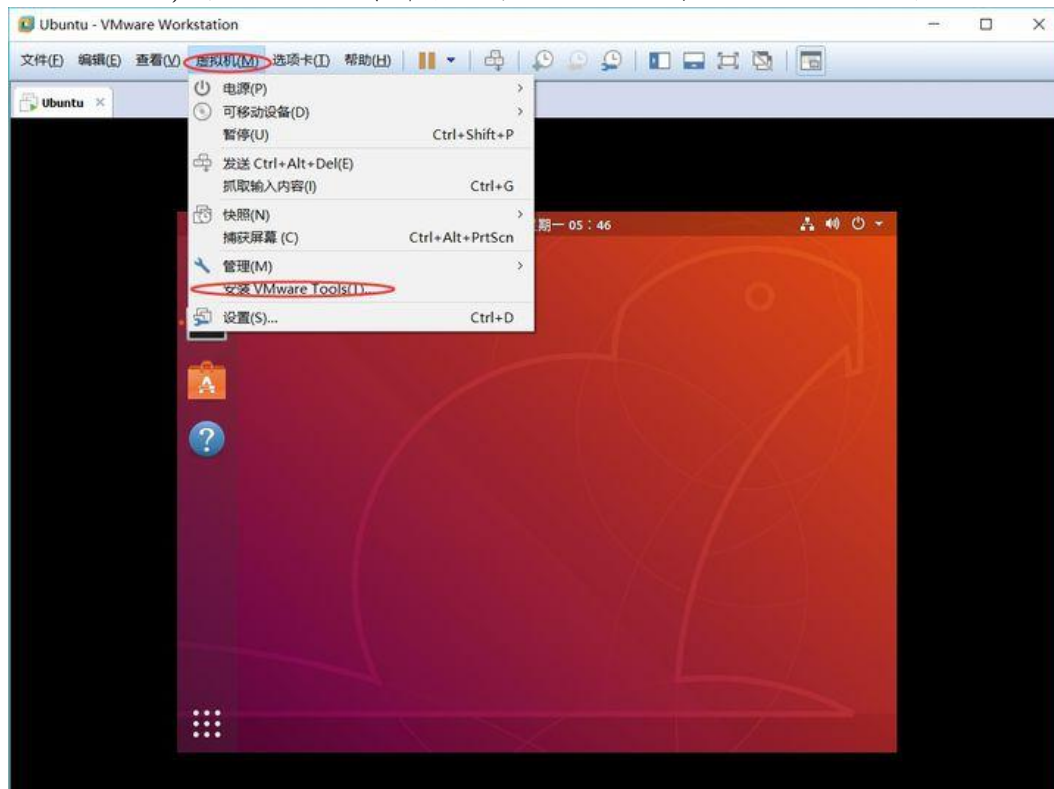


重新启动之后，会弹出一个窗口，问你要不要把英文状态下的文件名改为中文，这里选择不改动，保留旧的名称。

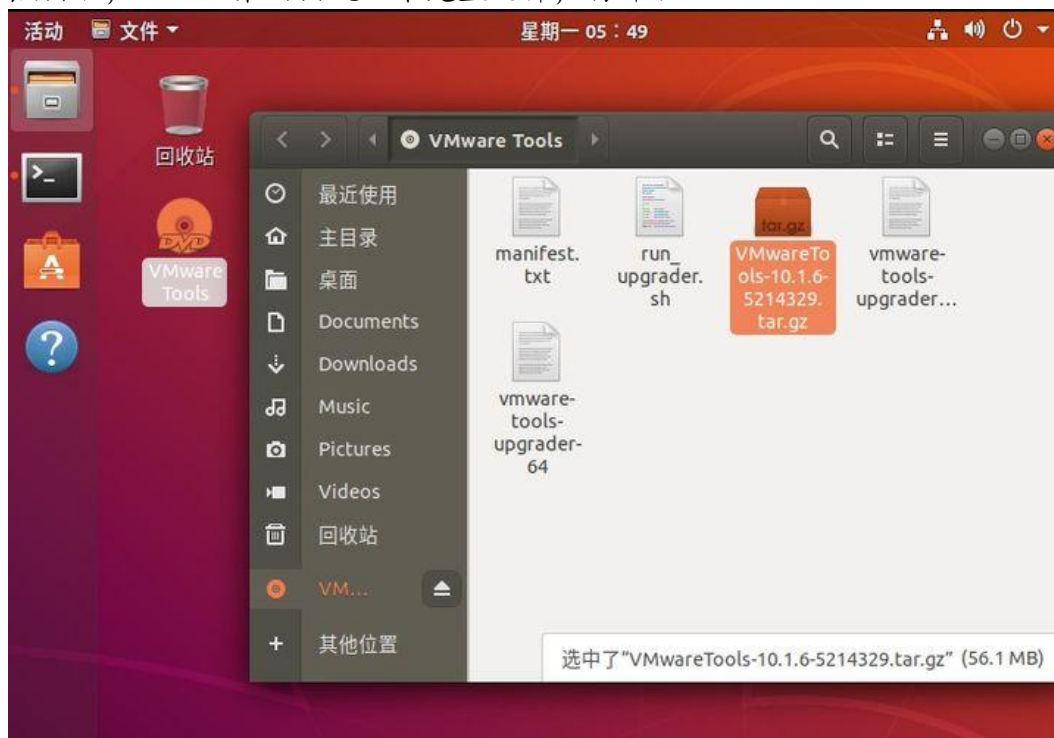


## 安装 VMware Tools 工具

启动 ubuntu，在 VMware 菜单栏 - 虚拟机 - 安装 VMware Tools 点击



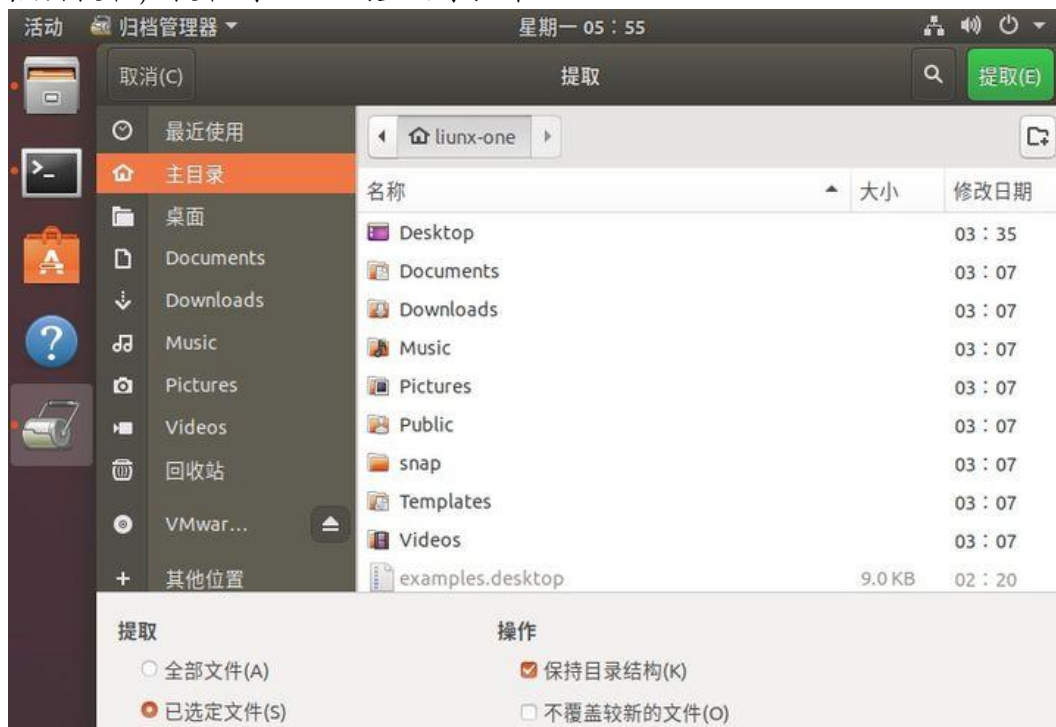
点击后，ubuntu 桌面出现一个光盘文件，打开后



把 VMware Tools-10.1.6-5214329.tar.gz 这个压缩包打开



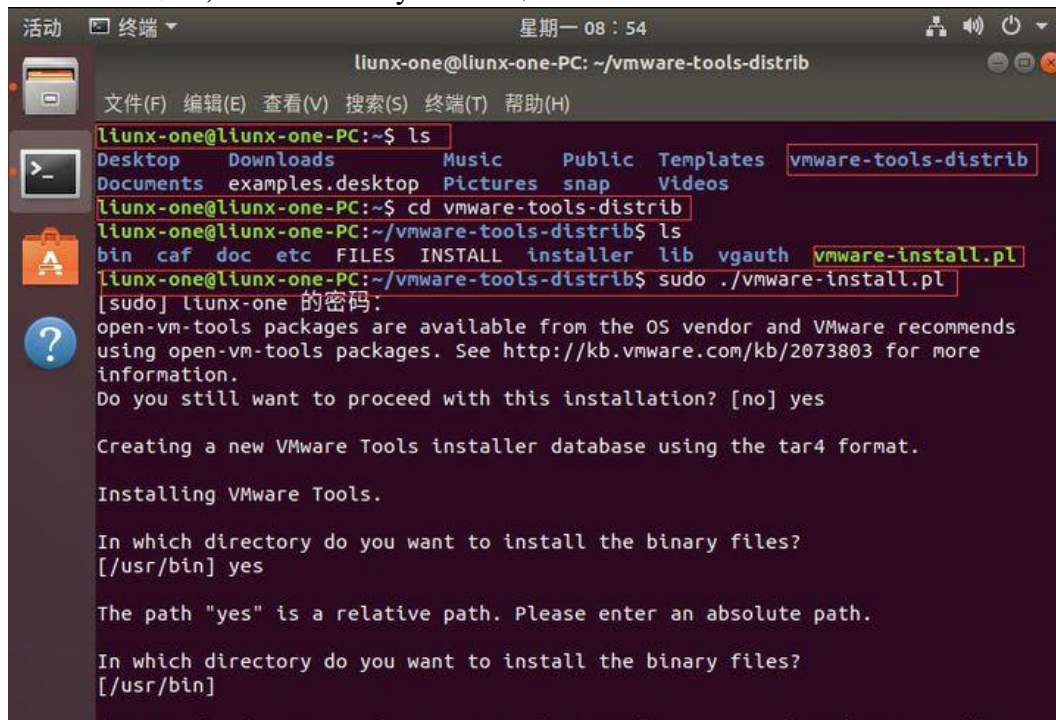
点击提取，提取到 home（主目录）下





然后在 ubuntu 中打开终端，输入 “ls” --> “cd vmware-tools-distrib” --> “sudo ./vmware-install.pl”

输入 root 密码，然后一路 “y” 加回车



```
liunix-one@liunix-one-PC: ~/vmware-tools-distrib
liunix-one@liunix-one-PC:~$ ls
Desktop  Downloads  Music      Public  Templates  vmware-tools-distrib
Documents examples.desktop Pictures  snap    Videos
liunix-one@liunix-one-PC:~$ cd vmware-tools-distrib
liunix-one@liunix-one-PC:~/vmware-tools-distrib$ ls
bin  caf  doc  etc  FILES  INSTALL  installer  lib  vgauth  vmware-install.pl
liunix-one@liunix-one-PC:~/vmware-tools-distrib$ sudo ./vmware-install.pl
[sudo] liunix-one 的密码:
open-vm-tools packages are available from the OS vendor and VMware recommends
using open-vm-tools packages. See http://kb.vmware.com/kb/2073803 for more
information.
Do you still want to proceed with this installation? [no] yes

Creating a new VMware Tools installer database using the tar4 format.

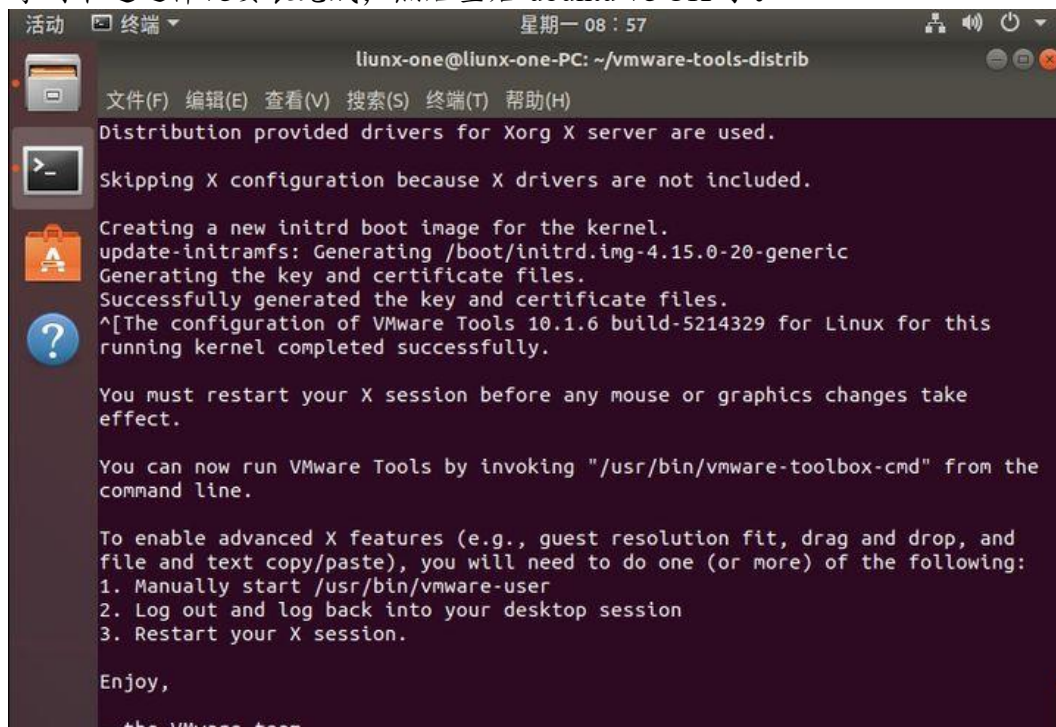
Installing VMware Tools.

In which directory do you want to install the binary files?
[/usr/bin] yes

The path "yes" is a relative path. Please enter an absolute path.

In which directory do you want to install the binary files?
[/usr/bin]
```

等到下边这样就安装完成，然后重启 ubuntu 就 OK 了。



```
liunix-one@liunix-one-PC: ~/vmware-tools-distrib
Distribution provided drivers for Xorg X server are used.

Skipping X configuration because X drivers are not included.

Creating a new initrd boot image for the kernel.
update-initramfs: Generating /boot/initrd.img-4.15.0-20-generic
Generating the key and certificate files.
Successfully generated the key and certificate files.
^[The configuration of VMware Tools 10.1.6 build-5214329 for Linux for this
running kernel completed successfully.

You must restart your X session before any mouse or graphics changes take
effect.

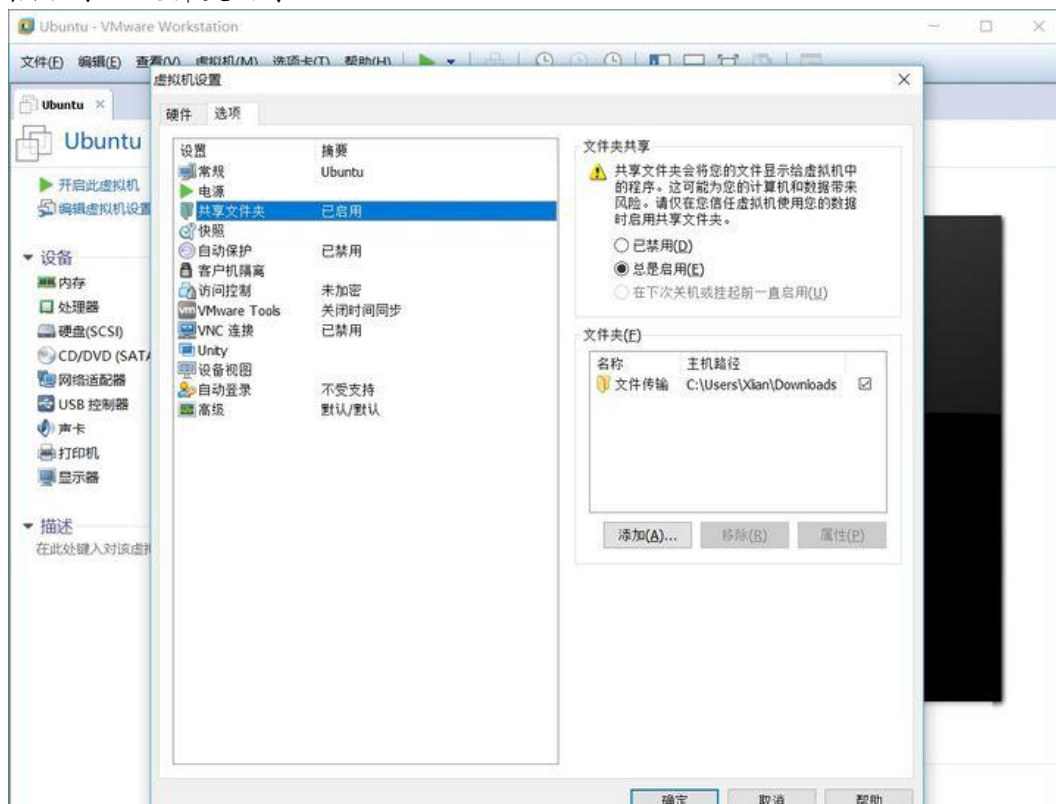
You can now run VMware Tools by invoking "/usr/bin/vmware-toolbox-cmd" from the
command line.

To enable advanced X features (e.g., guest resolution fit, drag and drop, and
file and text copy/paste), you will need to do one (or more) of the following:
1. Manually start /usr/bin/vmware-user
2. Log out and log back into your desktop session
3. Restart your X session.

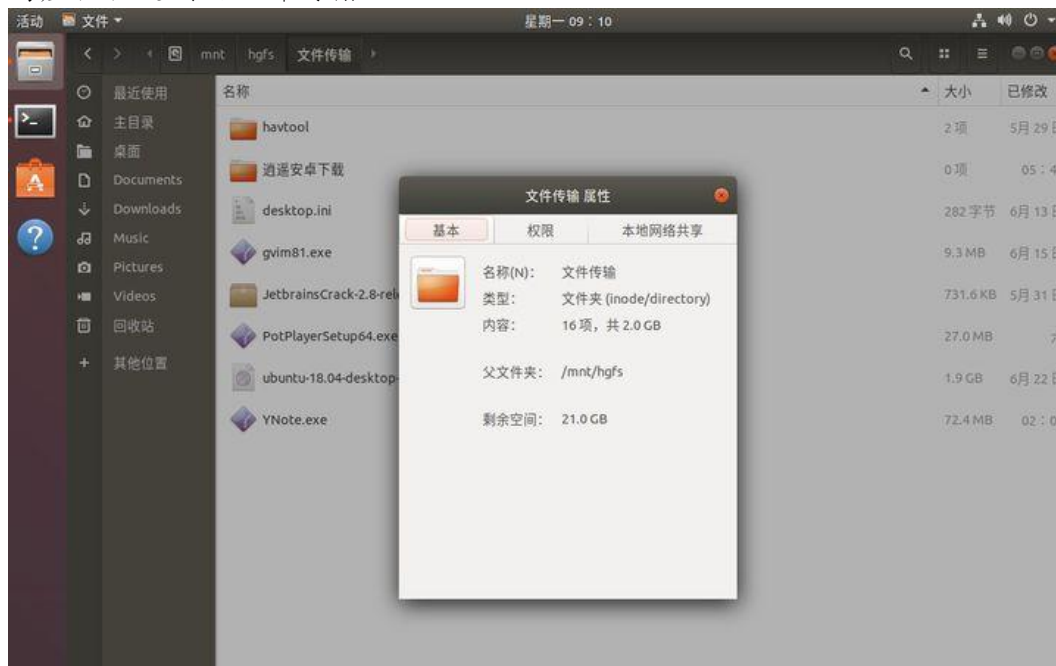
Enjoy,

--the VMware team
```

VMware Tools 工具安装完成后就可以设置共享文件夹，共享文件夹可以帮助主机与虚拟机之间进行文件传输。在虚拟机设置中找到共享文件夹启用共享文件夹，然后添加文件夹目录



在 ubuntu 文件管理/mnt/hgfs 中就是创建的共享文件夹，可以用这个来进行主机与虚拟机之间的文件传输。



这里的 ubuntu 就安装完成了。

# 实验十四 Hadoop 的安装与部署

## 一、实验目的

- (1) 理解 Hadoop 的工作原理与工作模式。
- (2) 掌握基于 Linux 平台的 JDK 6.0、Hadoop 的安装与配置；
- (3) 掌握 Hadoop 软件的单机工作模式的配置与使用方法。

## 二、实验内容和相应结果

### 1. 安装 JDK

到 sun 网站下载 JDK 安装包 jdk-6u11-linux-i586.bin, 复制到机器的 usr 目录中,并在每台机器的 root 用户下面安装.

在 root 用户下:

```
$ cd /usr
```

```
$ chmod +x jdk-6u11-linux-i586.bin 给安装文件增加执行权限.
```

```
$ ./jdk-6u11-linux-i586.bin,按提示按几个空格健后,输入 yes 后开始安装 jdk6.
```

安装好后,将目录名修改为 jdk6.

```
hadoop@ubuntu:~$ sudo tar -zxvf ~/Downloads/jdk-8u162-linux-x64.tar.gz -C /usr/lib/jvm
jdk1.8.0_162/
jdk1.8.0_162/javafx-src.zip
jdk1.8.0_162/bin/
jdk1.8.0_162/bin/jmc
jdk1.8.0_162/bin/serialver
jdk1.8.0_162/bin/jmc.ini
jdk1.8.0_162/bin/jstack
jdk1.8.0_162/bin/rmiregistry
jdk1.8.0_162/bin/unpack200
jdk1.8.0_162/bin/jar
```

设置 JDK 的环境变量,考虑到 JDK 可能会有其他系统用户也会用到,建议将环境变量直接设置在/etc/profile 中具体内容(如果没有则直接在 profile 文件中添加):

```
export JAVA_HOME=/usr/jdk6
```

```
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

```
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH:$HOME/bin
```

```
$ source /etc/profile 使用 java 环境生效.
```

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_162
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

```
hadoop@ubuntu:~$ java -version
java version "1.8.0_162"
Java(TM) SE Runtime Environment (build 1.8.0_162-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, mixed mode)
```

### 2. 设置目录并安装 Hadoop

用 hadoop 用户登录 namenode, 并新建一个目录, 用于存放所有 hadoop 相关内容。本例中在/home/hadoop 目录下新建 HadoopInstall



下载 hadoop 安装包并 copy 至 namenode 的 hadoop 用户的 /home/hadoop/HadoopInstall 并解压缩:

`tar zxvf hadoop-0.16.3.tar.gz` 考虑到今后升级以及其他操作的方便性, 建议建一个名称为 hadoop 的链接, 指向 hadoop-0.16.3 目录

`ln -s hadoop-0.16.3 hadoop` 新建目录: /home/hadoop/HadoopInstall/hadoop-conf 将 /home/hadoop/HadoopInstall/hadoop-conf

目录下的 `hadoop_site.xml,slaves,hadoop_env.sh,masters` 文件拷贝到 /home/hadoop/HadoopInstall/hadoop-conf 目录 在 /home/hadoop/.bashrc 文件中设置环境变量 `$HADOOP_CONF_DIR`:

`export HADOOP_CONF_DIR=$HOME/HadoopInstall/hadoop-conf/`

```
betty@ubuntu:~$ sudo useradd -m hadoop -s /bin/bash
[sudo] password for betty:
betty@ubuntu:~$ sudo passwd hadoop
New password:
Retype new password:
passwd: password updated successfully
betty@ubuntu:~$ sudo adduser hadoop sudo
Adding user `hadoop' to group `sudo' ...
Adding user hadoop to group sudo
Done.
```

```
hadoop@ubuntu:~$ cd /usr/local/hadoop
hadoop@ubuntu:~$ ./bin/hadoop version
Hadoop 3.3.0
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r aa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled by brahma on 2020-07-06T18:44Z
Compiled with protoc 3.7.1
From source with checksum 5dc29b802d6ccd77b262ef9d04d19c4
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.3.0.jar
```

### 3. 单机工作模式

单机模式下 Hadoop 使用的是本地文件系统, Hadoop 中有几个示例程序并且已经打包成了 `hadoop-0.20.1-examples.jar`。其中有一个 WordCount 程序, 功能是统计一批文本文件中 各个单词出现的次数

`$ cd /usr/hadoop/hadoop-0.20.1 $ mkdir test-in`

`$ cd test-in`

`$ echo "hello world bye world" >file1.txt`

`$ echo "hello hadoop goodbye hadoop" >file2.txt`

#在 test-in 目录下创建两个文本文件, WordCount 程序将统计其中各个单词出现次数

`$ cd ..`

`$ bin/hadoop jar hadoop-0.20.1-examples.jar wordcount test-in test-out`

注意事项: test-out 目录是程序生成的,运行前必须先删除 test-out 目录 # 执行完毕, 下面 查看执行结果:

`$ cd test-out`

\$ cat part-00000 bye 1  
goodbye 1

hadoop 2

hello 2

world 2

```
hadoop@ubuntu:/usr/local/hadoop$ cd test-out
hadoop@ubuntu:/usr/local/hadoop/test-out$ cat part-r-00000
bye      1
goodbye  1
hadoop   2
hello    2
world    2
```

# 实验七 Internet 应用

## 1、学习 Tomcat

参考：<https://www.bilibili.com/video/BV1dJ411N7Um?p=4>

## 2、安装 Tomcat

参考：<https://zhuanlan.zhihu.com/p/37852137>

安装成功截图



## 3、实现第一个任务

使用 socket API 实现简单的 HTTP 客户

(1) 使用 `sendMessage` 方法发送符合 http 协议定义的消息给 web 服务器以获得想要的页面

(2) 使用 `receiveMessage` 方法接收返回的页面源文件并显示。

代码

```
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.SocketAddress;

public class HttpSocketClient {
```

```

private Socket mSocket;

public static void main(String[] args) {
    HttpClient client = new HttpClient();
    try {
        client.sendGet("www.baidu.com",80,"/");
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public HttpClient() {
    this.mSocket = new Socket();
}

/** 在百度服务器面前，这就是一个正常的浏览器
 * @param host
 * @param port
 * @param path
 * @throws IOException
 */
void sendGet(String host, int port, String path) throws IOException {
    SocketAddress dest = new InetSocketAddress(host, port);
    mSocket.connect(dest);
    OutputStreamWriter streamWriter = new
OutputStreamWriter(mSocket.getOutputStream());
    BufferedWriter bufferedWriter = new BufferedWriter(streamWriter);

    bufferedWriter.write("GET " + path + " HTTP/1.1\r\n");
    bufferedWriter.write("Host: " + host + "\r\n");
    bufferedWriter.write("Connection: " + "keep-alive" + "\r\n");
    bufferedWriter.write("User-Agent: " + "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140
Safari/537.36" + "\r\n");
    bufferedWriter.write("Accept: "
+ "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;
q=0.8" + "\r\n");
    bufferedWriter.write("Accept-Language: "
+ "zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7" + "\r\n");
    bufferedWriter.write("\r\n");
    bufferedWriter.flush(); //flush 一下很重要，等于说已经写完了

    BufferedInputStream stream = new
BufferedInputStream(mSocket.getInputStream());
    BufferedReader bufferedReader = new
BufferedReader(new
InputStreamReader(stream));
    String line = null;

```

```
while ((line = bufferedReader.readLine())!=null) {
    System.out.println(line);
}
bufferedReader.close();
bufferedWriter.close();
mSocket.close();
}
}
```

参考: <https://www.cnblogs.com/ranyonsue/p/5984001.html>

#### 4、实现第二个任务

使用隐式表单域传输会话状态数据

- (1) 编写客户端请求页面 form.html
- (2) 编写 form.html 所触发的脚本 hiddenform.c
- (3) 编写 hiddform.cgi 动态生成页面中触发的脚本 hidden.form2.c

学习 html: <https://www.runoob.com/html/html-intro.html>

学习 html 中的表单: <https://www.runoob.com/html/html-forms.html>

什么是隐式表单: <https://www.debugease.com/javaweb/1013492.html>

隐式表单的作用: <https://www.jianshu.com/p/5bdf753acef2>

学习 xml: <https://www.runoob.com/xml/xml-tutorial.html>

学习如何使用 Tomcat 开发:

[https://www.bilibili.com/video/BV1gJ411q7Lf?from=search&seid=600581961118443894&spm\\_id\\_from=333.337.0.0](https://www.bilibili.com/video/BV1gJ411q7Lf?from=search&seid=600581961118443894&spm_id_from=333.337.0.0)

#### 5、实现第三个任务

##### 3. 使用 cookie 传递状态数据

- (1) 编写请求页面 cookie.html
- (2) 编写 cookie.html 所触发的脚本 cookie.c
- (3) 编写由 cookie.c 动态生成页面中触发的脚本 cookie2.c

学习 cookie:

Cookie、Session、Token 的区别:

[https://www.bilibili.com/video/BV1ob4y1Y7Ep?from=search&seid=8681360534596789764&spm\\_id\\_from=333.337.0.0](https://www.bilibili.com/video/BV1ob4y1Y7Ep?from=search&seid=8681360534596789764&spm_id_from=333.337.0.0)

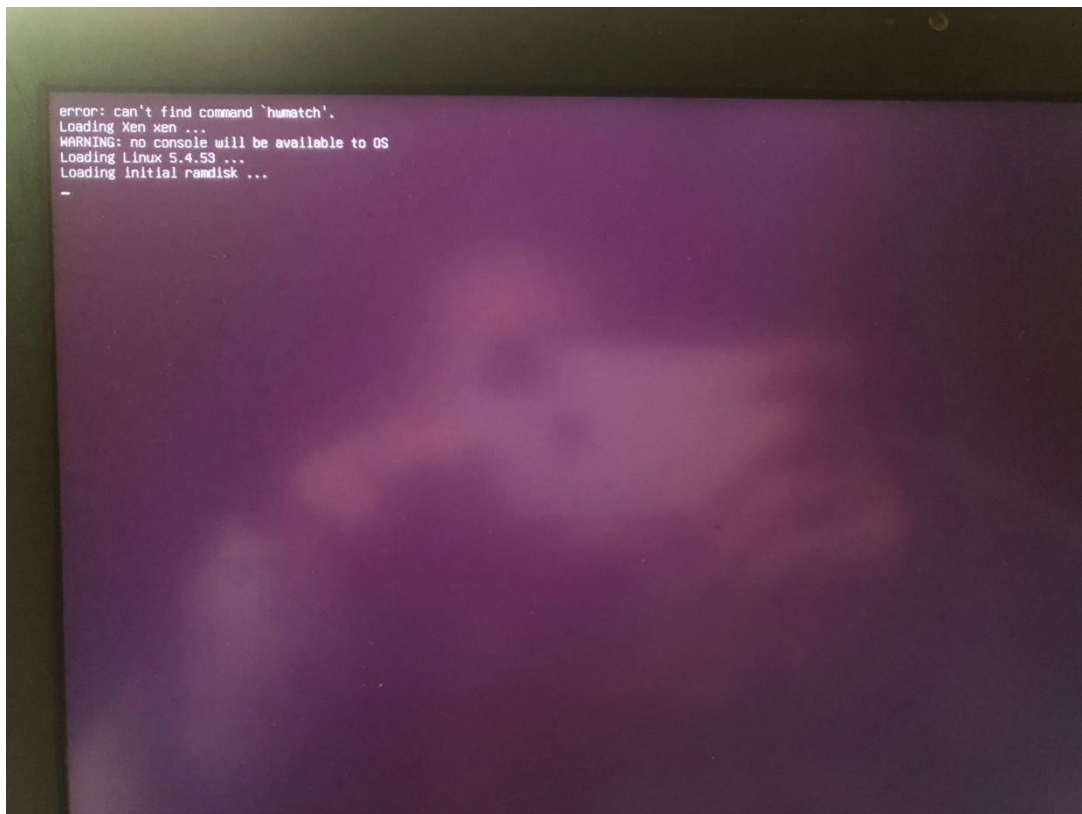
Cookie 使用步骤与 Cookie 原理:

[https://blog.csdn.net/weixin\\_43691058/article/details/104052875?utm\\_medium=distri-bute.pc\\_aggpage\\_search\\_result.none-task-blog-2~aggregatepage~first\\_rank\\_ecpm\\_v1~rank\\_v31\\_ecpm-1-104052875.pc\\_agg\\_new\\_rank&utm\\_term=%E4%BD%BF%E7%94%A8cookie%E4%BC%A0%E9%80%92%E7%8A%B6%E6%80%81%E6%95%B0%E6%8D%AE&spm=1000.2123.3001.4430](https://blog.csdn.net/weixin_43691058/article/details/104052875?utm_medium=distri-bute.pc_aggpage_search_result.none-task-blog-2~aggregatepage~first_rank_ecpm_v1~rank_v31_ecpm-1-104052875.pc_agg_new_rank&utm_term=%E4%BD%BF%E7%94%A8cookie%E4%BC%A0%E9%80%92%E7%8A%B6%E6%80%81%E6%95%B0%E6%8D%AE&spm=1000.2123.3001.4430)



# 实验心得与感想

此次实验颇为不易，除了我在实验报告中呈现的三个实验，我此前还失败了一些实验。除了做实验指导书中的“实验一 数据包 socket 的应用”，“实验十三 虚拟机的使用与 Linux 系统的安装”，“实验十四 Hadoop 的安装与部署”之外。我之前先是在 Ubuntu18.04 的双系统上安装了 hadoop，再然后打算做“实验十 安装 Xen 熟悉常用命令”，可惜经历千辛万苦安装完成 Xen 后重启系统时，Ubuntu 无法重启了，它卡在了开机的某一个界面，如下图所示。



查询并询问同学后，发现除了重装外没有什么好办法。于是，我损失了一个 Ubuntu 的系统。顺带的，还有里面安装好的各种环境，比如 python 与 cuda 的环境。这两者尤其难以安装，印象十分深刻。因此，我不能再访问我笔记本上的一半存储空间以及在 linux 上的所有环境了。

于是我开始重新做分布式的实验，然后我打算做“实验七 Internet 应用”以及后面的几个实验。然而这个实验我钻研良久也没弄清楚要怎么弄，我想是因为确实太缺少前置知识了。应该是 web 开发的课程知识，然而我并没选这门课。最终，这个实验我做了一半没做完。将我写到一半的实验报告也呈现了出来，想来以后补充了前置知识再来钻研这个实验。

虽然过程曲折，但分布式系统与云计算这门课还是让我学到了很多知识。除了李玺老师精彩课堂讲述外，老师让我们做的案例调研也让我们学到了很多较新和工程上实际应用的框架和知识。非常感谢有机会上这门课，也非常感激老师的教导，将我们领入了分布式系统领域的大门。