****

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](#_Toc26268)

[一、 实验目的 3](#_Toc10023)

[二、 实验内容和相应结果 3](#_Toc18816)

[1. 服务器端 3](#_Toc23863)

[2. 客户端 4](#_Toc32187)

[实验十三 虚拟机的使用与 Linux系统的安装 5](#_Toc25957)

[一、 实验目的 5](#_Toc32313)

[二、 实验内容和相应结果 5](#_Toc499)

[1. 下载软件 5](#_Toc4886)

[2. VMware安装ubuntu 5](#_Toc13801)

[3. 安装后的问题解决 19](#_Toc2346)

[实验十四 Hadoop 的安装与部署 27](#_Toc26285)

[一、 实验目的 27](#_Toc25395)

[二、 实验内容和相应结果 27](#_Toc9999)

[1. 安装 JDK 27](#_Toc31776)

[2. 设置目录并安装 Hadoop 27](#_Toc29744)

[3. 单机工作模式 28](#_Toc23209)

[实验七 Internet 应用 30](#_Toc24050)

[1、 学习Tomcat 30](#_Toc2023)

[2、 安装Tomcat 30](#_Toc9928)

[3、 实现第一个任务 30](#_Toc2903)

[4、 实现第二个任务 32](#_Toc5060)

[5、 实现第三个任务 32](#_Toc11045)

[实验心得与感想 33](#_Toc6082)

# 实验一 数据包socket应用

## 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

## 实验内容和相应结果

### 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 inbuffff.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。

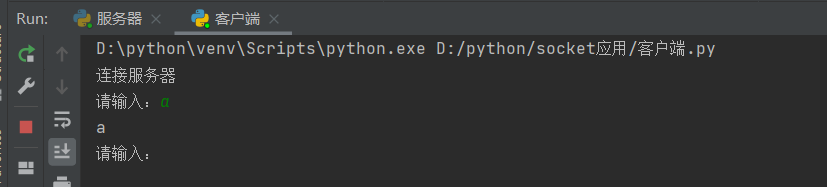
String message = " Server received message !";

byte [] sendData = message . getBytes () ;

DatagramPacket outPacket = new DatagramPacket ( sendData , sendData .

length , inPacket . getSocketAddress () ) ;

socket . send ( outPacket ) ;



### 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 ) ;

}

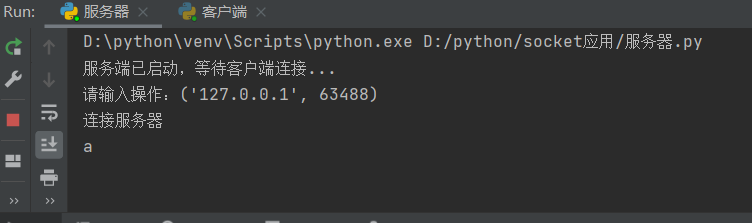
最后接收服务器端发送的收到信息，通过构建一个 DatagramPacket 类的实例inPacket 用于读取 socket 实例中的内容，并打印输出

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

## 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。

## 实验内容和相应结果

### 下载软件

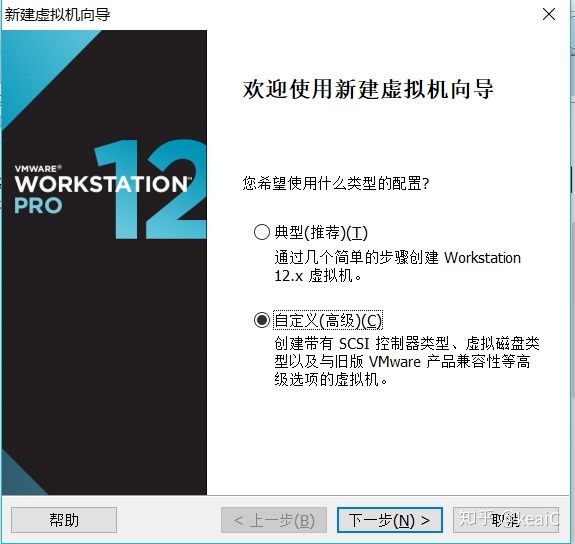
首先需要下载VMware Workstation Pro 12虚拟机。

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](https://link.zhihu.com/?target=https://www.ubuntu.com/download/desktop" \t "https://zhuanlan.zhihu.com/p/_blank)

### VMware安装ubuntu

打开安装好的虚拟机，创建新的虚拟机。

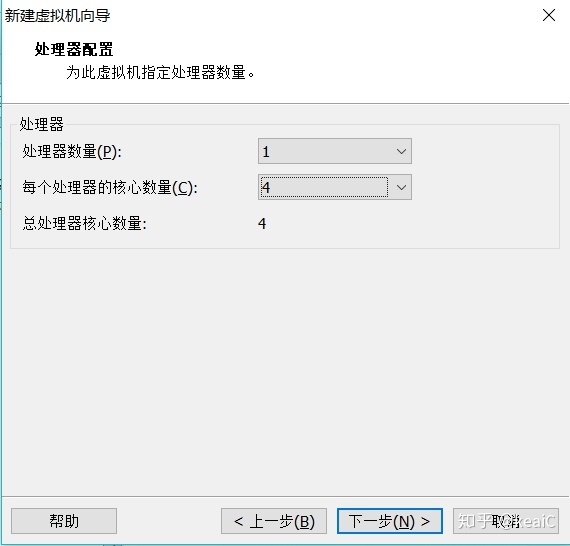
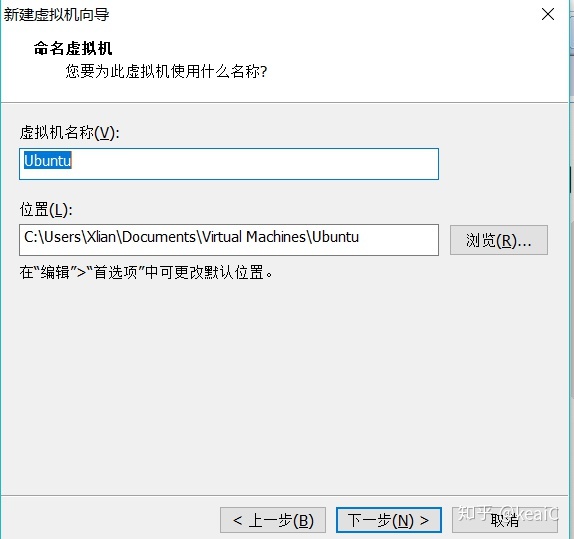




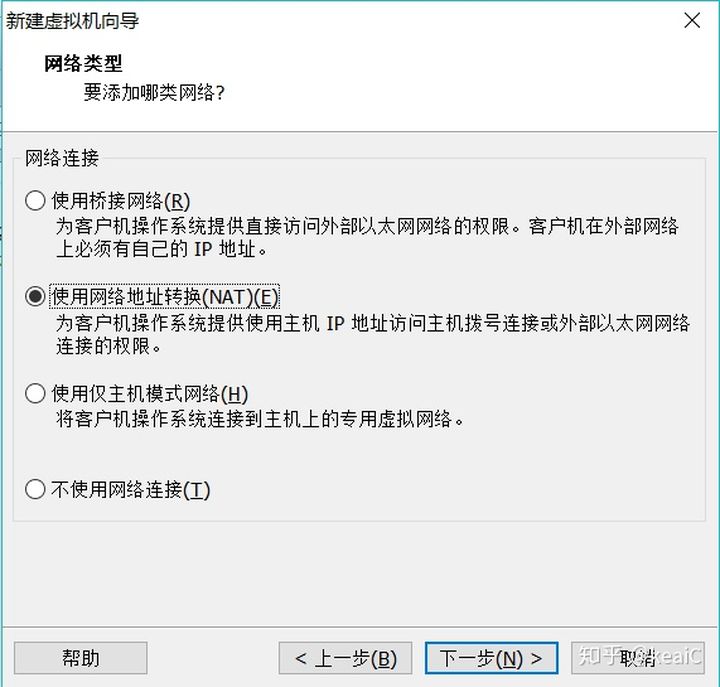
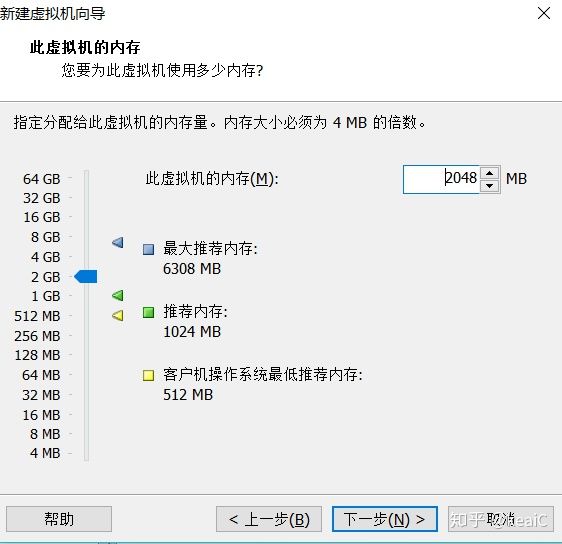
接下来这步一定先不要选择下载的光映像文件，选了点“下一步”系统就直接帮你配置好后面的配置，要根据自己的要求进行选择配置的虚拟机，所以这步选择“稍后安装操作系统”。

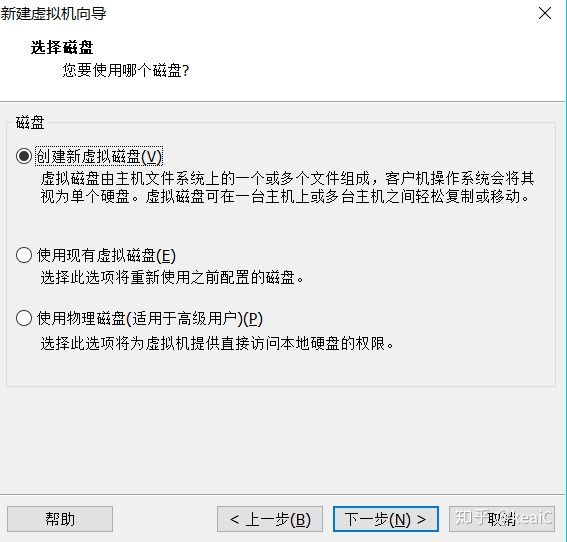
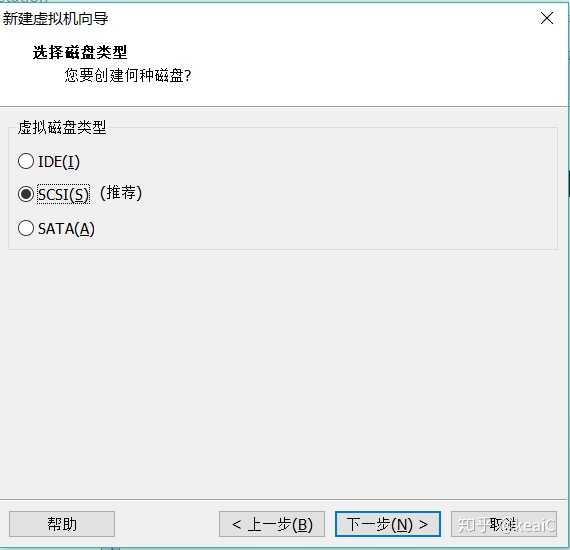
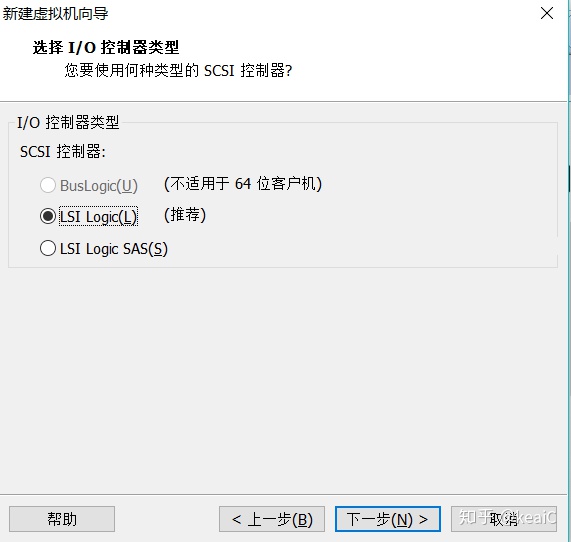


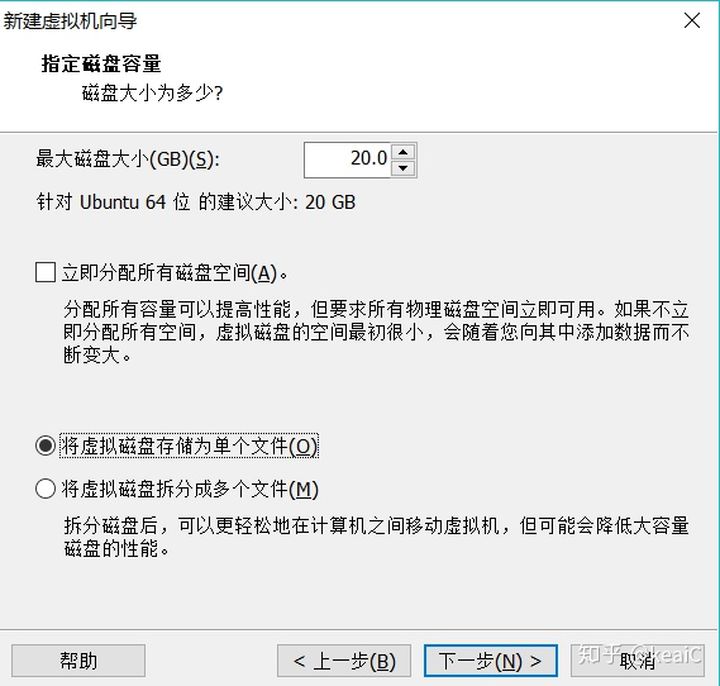




这步配置给虚拟机的内存，这个要根据自己的电脑配置给虚拟机分内存，内存大，肯定性能好，但是别到时候给虚拟机的内存大了，影响自己主电脑的使用。

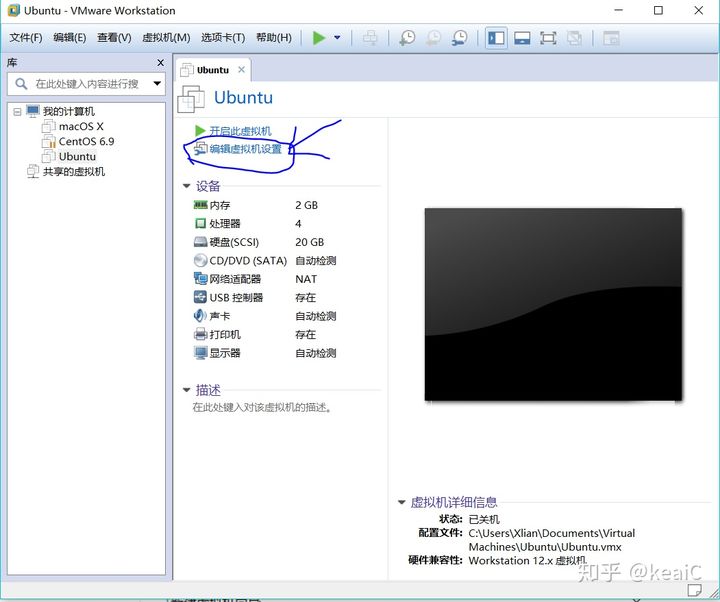


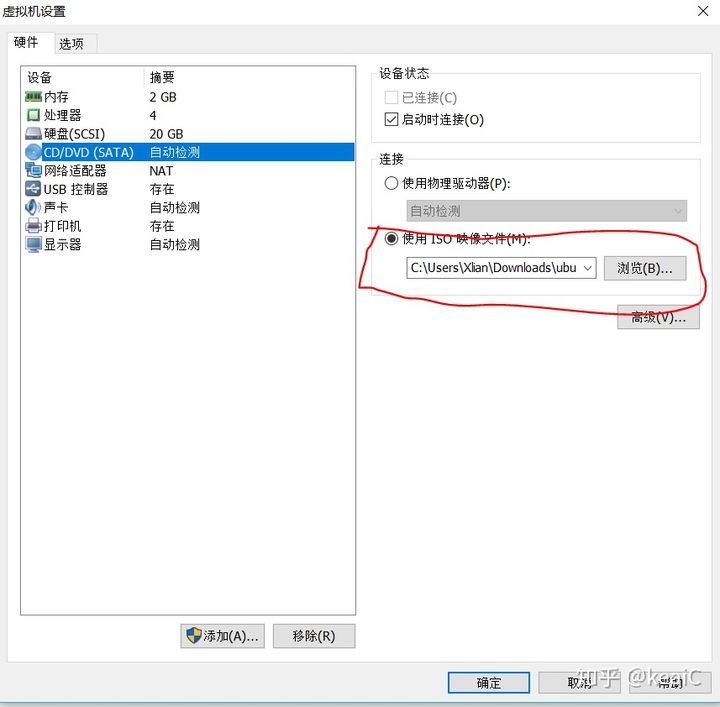
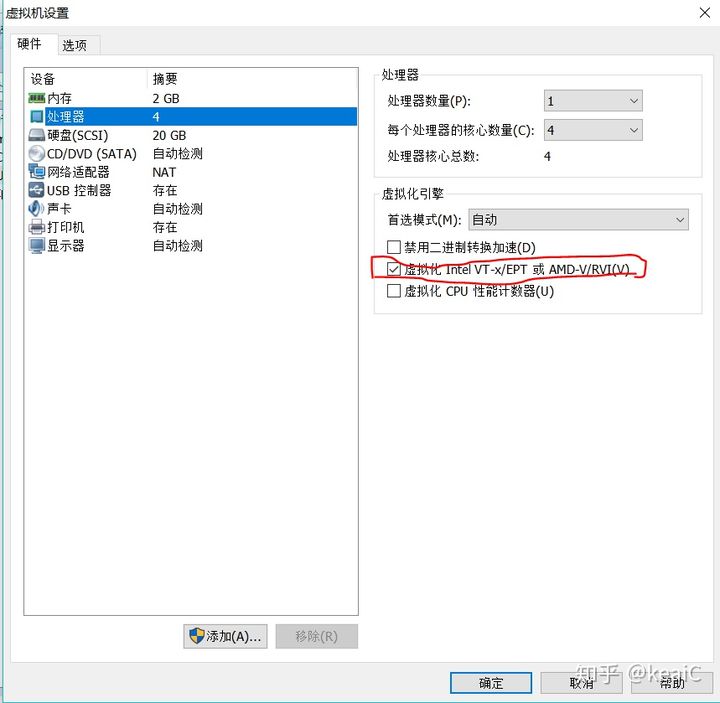




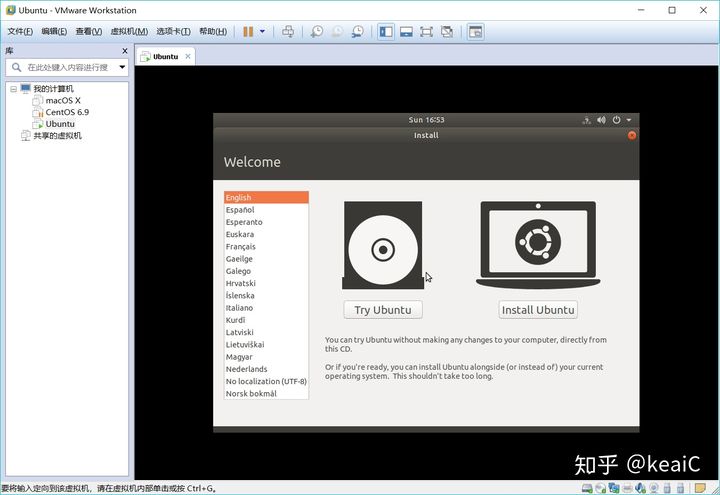


在这里虚拟机就创建完成了，下面打开虚拟机设置，设置安装镜像文件的位置（就是之前下载好的Ubuntu光盘映像文件）

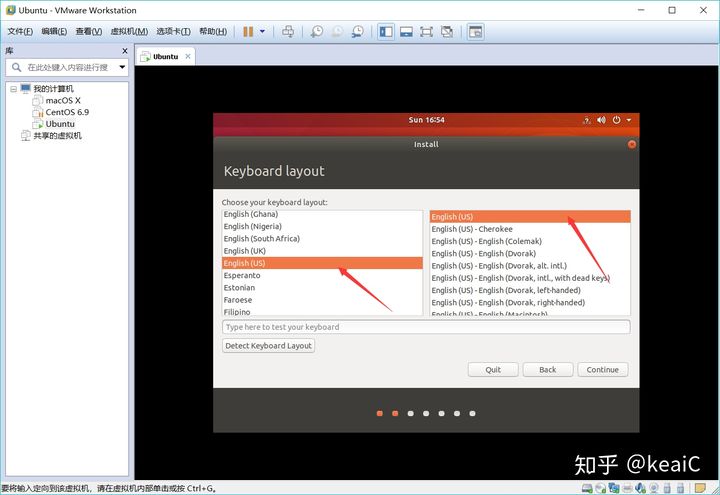




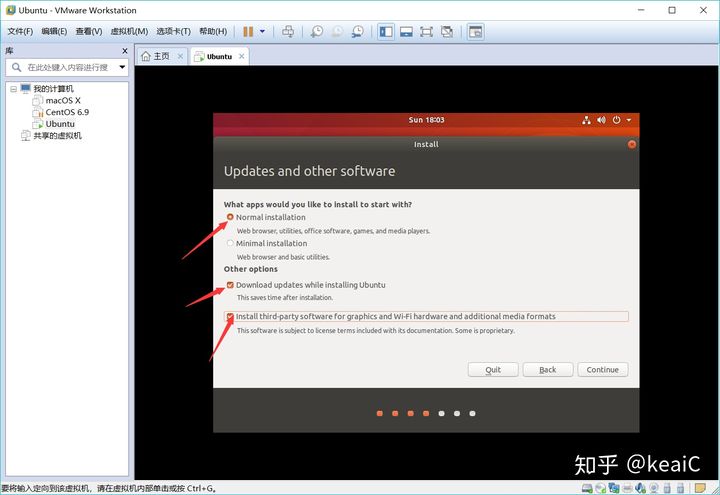
设置好后点击“开启此虚拟机”，在正常安装系统时，要在BIOS里设置启动项，把电脑设置为光盘启动，但是这个虚拟机他可以自己识别这台虚拟机有没有安装系统，所以启动时他就可以直接进入系统安装界面。下图，关于语言选择，其实比较推荐选择英文，如果选择中文，安装完成后桌面、文档等文件夹的名字会成为中文，在终端操作时会有不便，使用命令行报错也会显示中文，便于阅读问题，但是不利于进行搜索。进入安装界面后选择“English”，点击install ubuntu进入下一步；



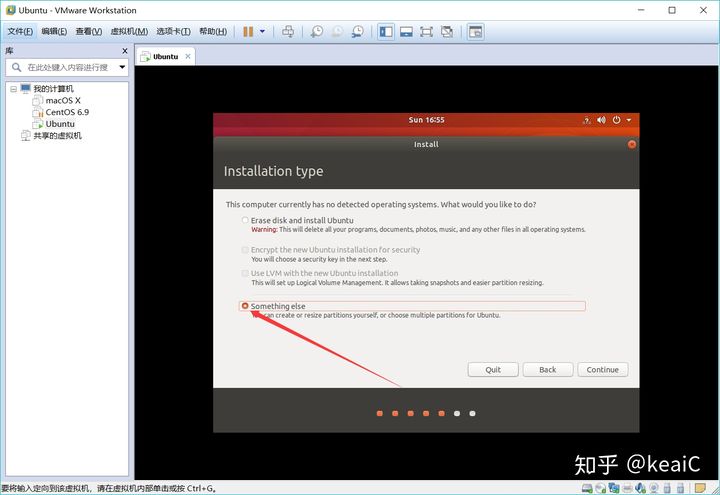
下面是键盘选择键盘选默认English（US）即可



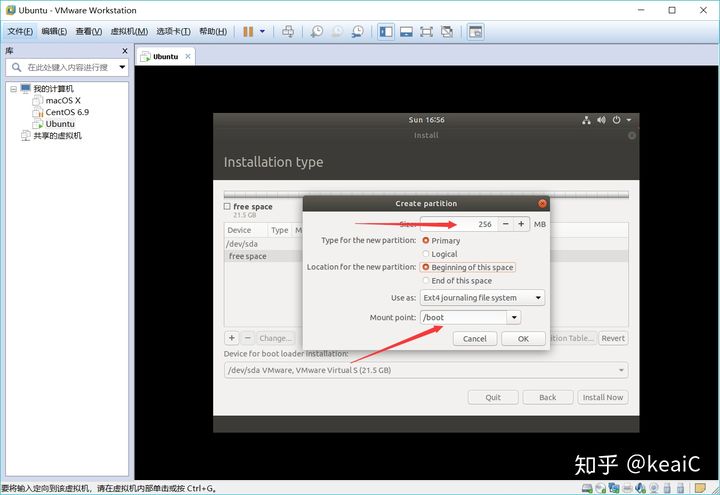
下面这两个选项第一个选择的是正常安装，这个安装完后就是会出现一些系统自带的软件，不用的可以卸载掉，第二个选项是最小安装他只会安装系统要用到的基本工具。下面两个都勾选上下面第一个是安装ubuntu时下载更新，第二个为图形或无线硬件以及其他的媒体格式安装第三方软件。我这里选择的是正常安装，如果想要一个纯净的系统就可以选择最小安装。



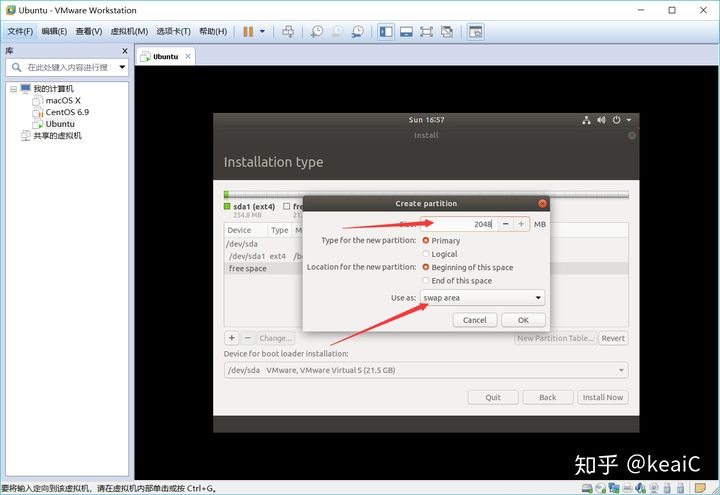
这里是给磁盘分区，进行手动分区，选择最后一个“something else”（其他选项）



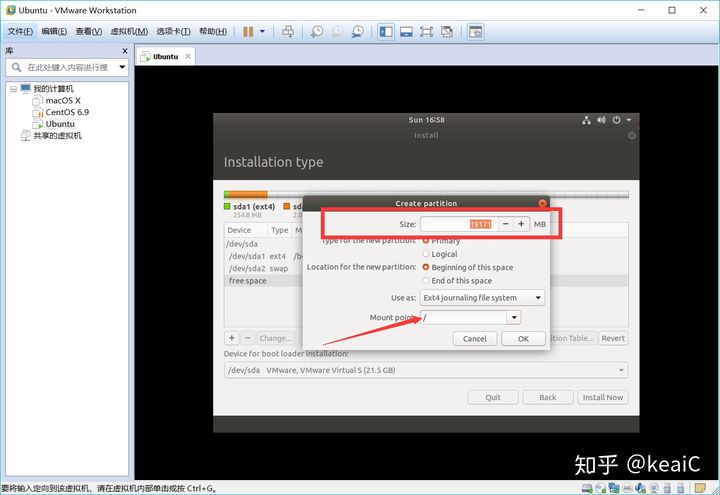
进来后进行分区，先分一个启动分区boot，一般启动分区的大小为200M就可以了。boot包含了操作系统的内核和在启动系统过程中所要用到的文件，要给他单独分出来。要是不分当磁盘文件写满了磁盘空间不足时或者根分区出现问题了，的系统就没法启动，所以给他单独分出来。



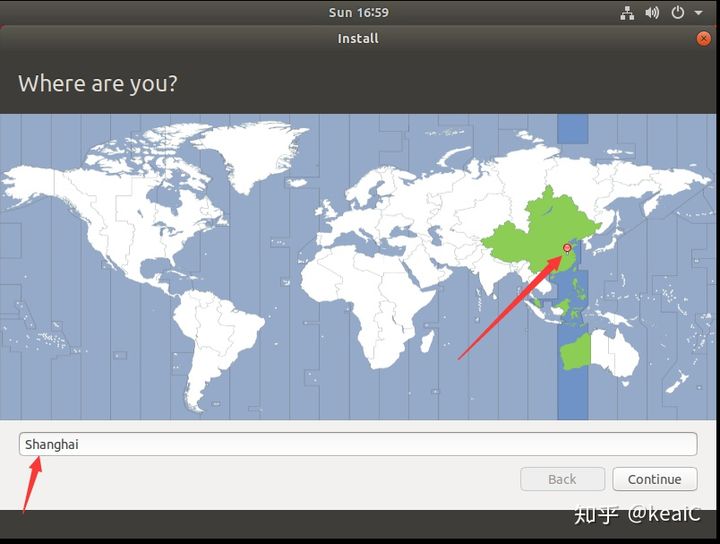
boot分区完成后分一个交换分区“swap”，swap分区，在windows中就是虚拟内存，在liunx中如果没有swap分区，系统内存用完后，系统他会杀死一部分程序，这个在使用中是非常可怕的，你正在用电脑，电脑内存满了，你的程序莫名其妙被关闭了，这是绝对不能允许的，所以必须分一个swap分区。还有一点swap分区的大小，一般电脑的内存小于等于4G时，swap给他是电脑内存的1.5-2倍；大于4G时，电脑内存多少swap就给多少。比如我的电脑内存为4G，那么swap就给6G=6144M-8G=8192；我的电脑内存为8G,16G...的swap就给8G,16G...跟电脑内存一样大就可以了。在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



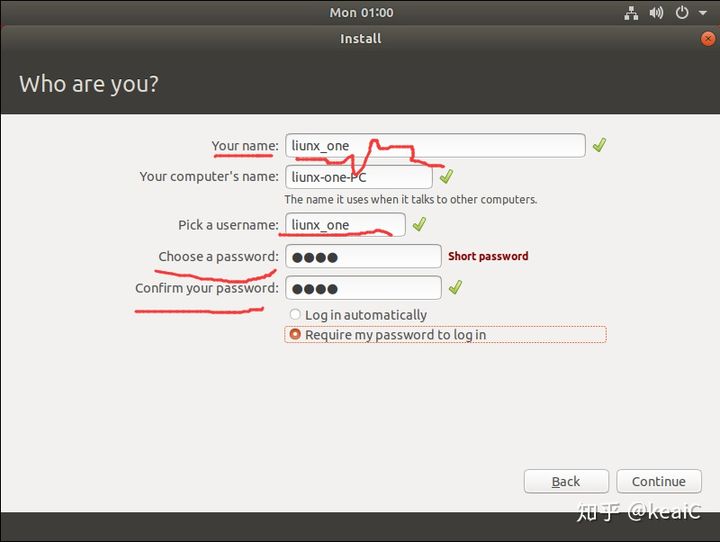
下面给分最后一个分区“/”（根分区），根分区就是系统根目录所在的分区，一般根目录下面只有目录，不要直接有文件。Linux只有一个根目录，就是“/”，其它目录都是它的子目录。这里把剩下的磁盘空间都给“/”就可以了。（这里说一下"/home" 用户的home目录所在地，这个分区的大小取决于有多少用户。如果是多用户共同使用一台电脑的话，这个分区是完全有必要的，额外分割出/home有个最大的好处，当你重新安装系统时，你不需要特别去备份你的个人文件，只要在安装时，选择不要格式化这个分区，重新挂载为/home就不会丢失你的数据。因为是虚拟机，这台虚拟机是自己使用的，所以/home没有分出来，如果有需要可以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



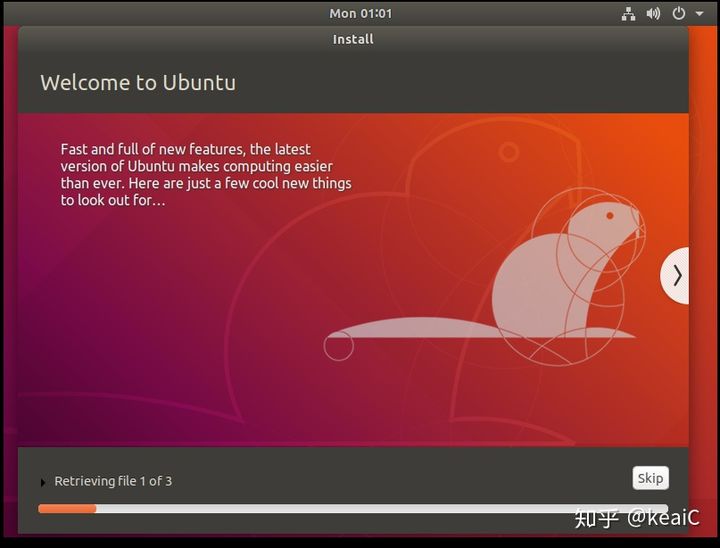
接下来就是输入你的位置，选择上海就可以了



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。



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。



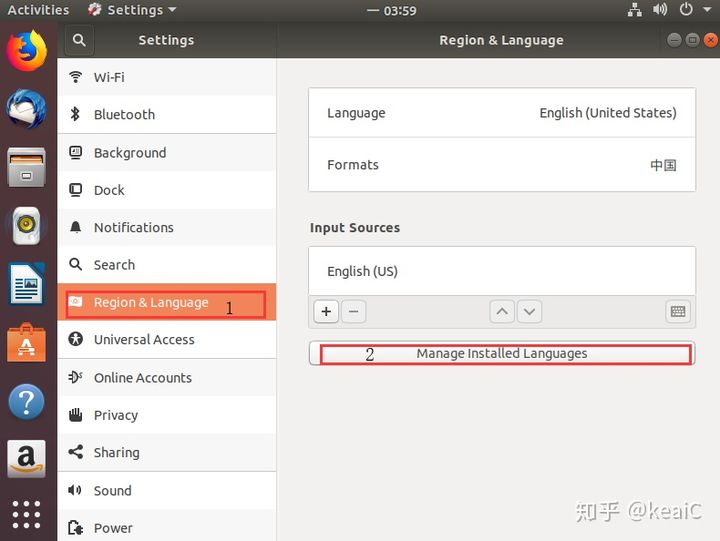
### 安装后的问题解决

**设置语言环境**

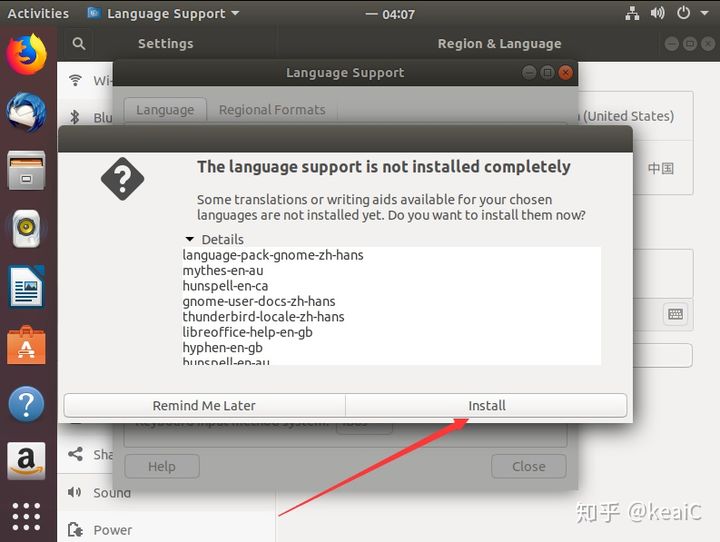
安装时语言选择是英文，那有的人他就用不惯，接下来说一下怎么把ubuntu从英文环境切换成中文环境。先在ubuntu桌面右上角“1”那点一下，再再“2”那点一下进入设置界面



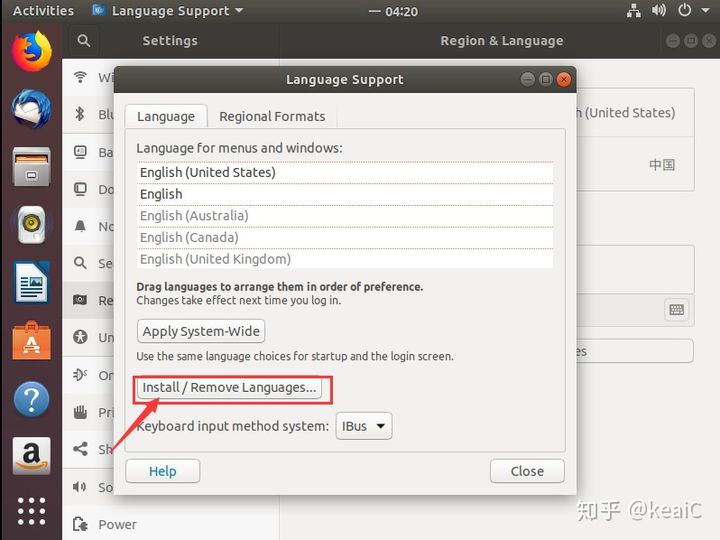
进入设置界面后点击“Region&language”语言区域“1”那个位置，然后点击“Manage installed languages”管理安装语言“2”那个位置



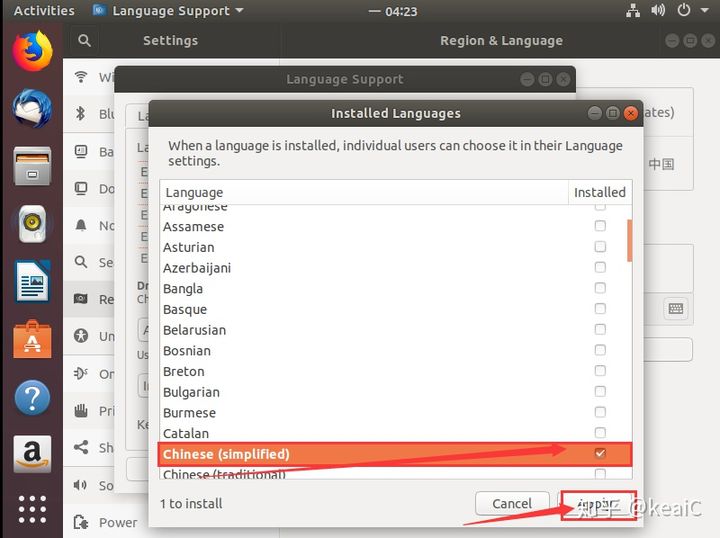
点击“Manage installed languages”后会弹出一个窗口，他提示有语言支持需要安装，点击安装“install”，系统会提示输入用户密码，输入密码后就等待安装完成



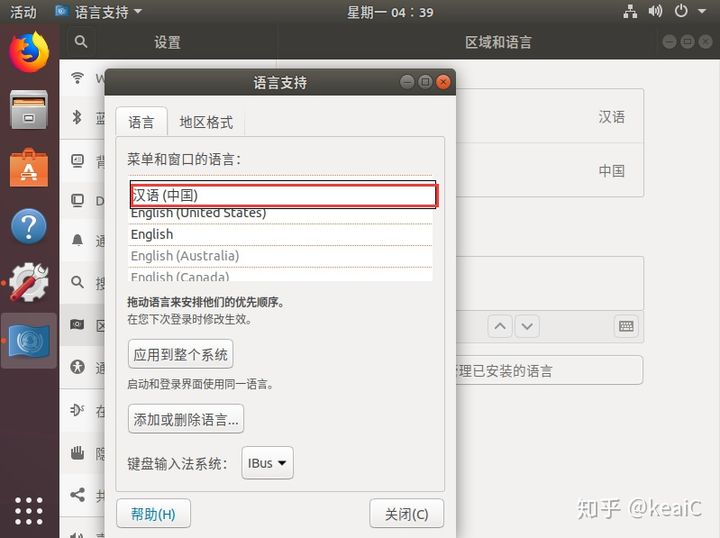
安装完成后会弹出一个窗口下面这张图，选择安装或卸载语言这个选项



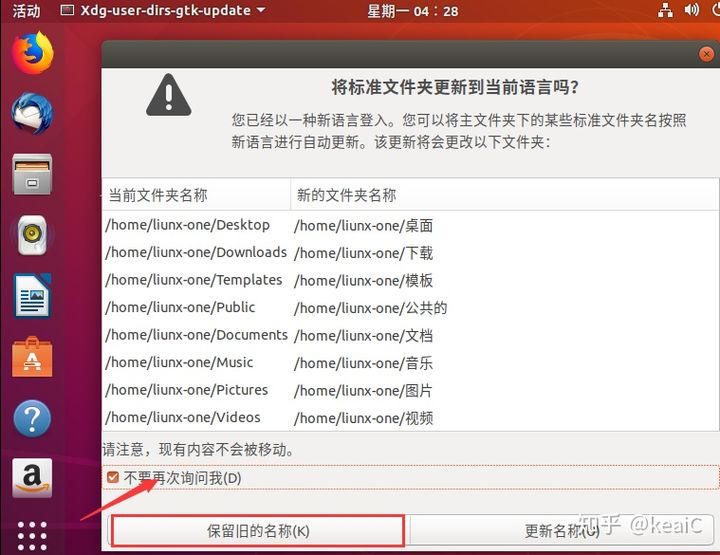
点击后会弹出下面这个窗口，找到简体中文这个选项在后面打勾，点击Apply，等待安装完成



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。

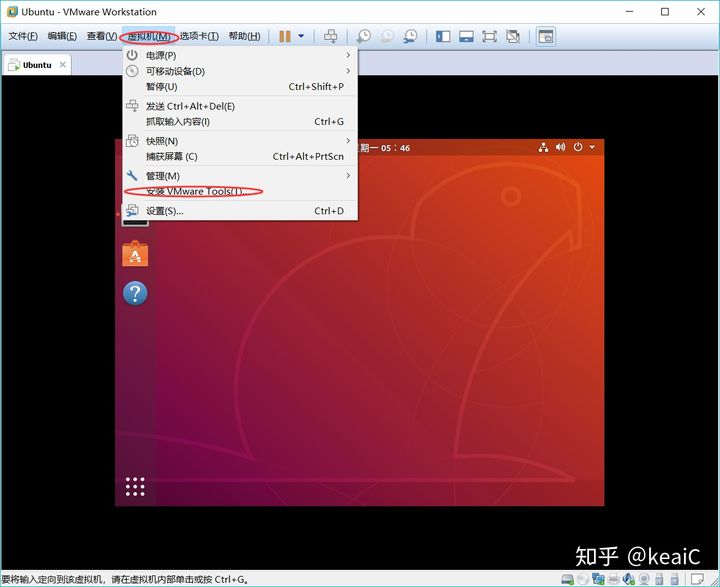


重新启动之后，会弹出一个窗口，问你要不要把英文状态下的文件名改为中文，这里选择不改动，保留旧的名称。



**安装VMware Tools工具**

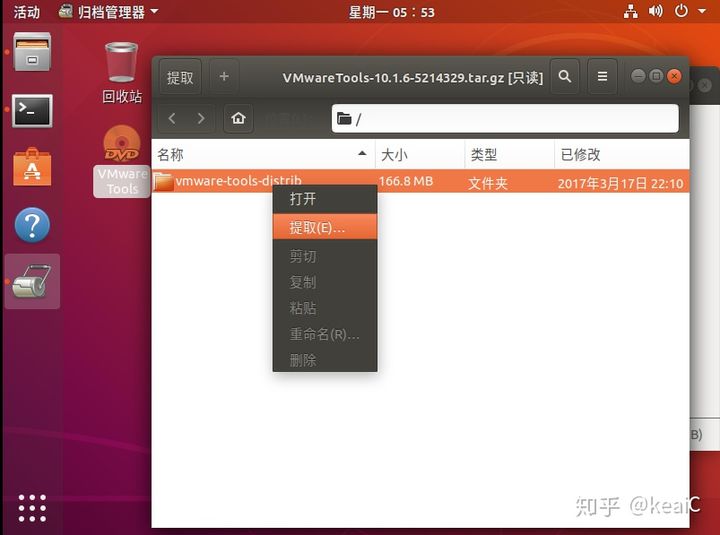
启动ubuntu，在VMware菜单栏 - 虚拟机 - 安装VMware Tools点击



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

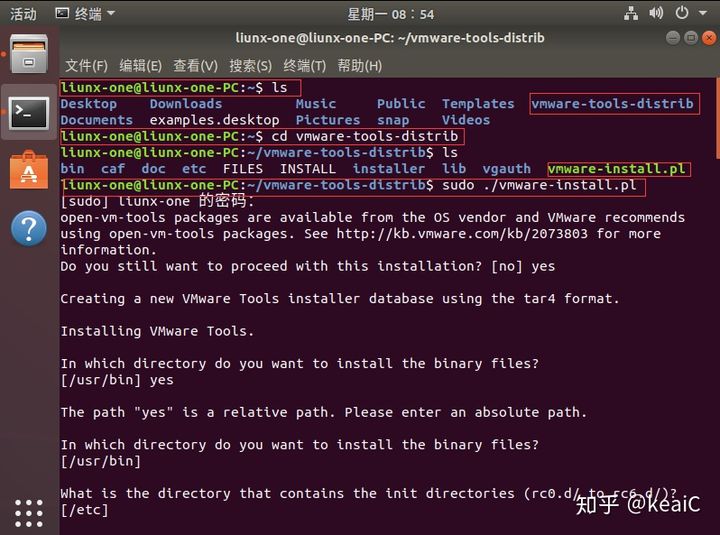


点击提取，提取到home（主目录）下

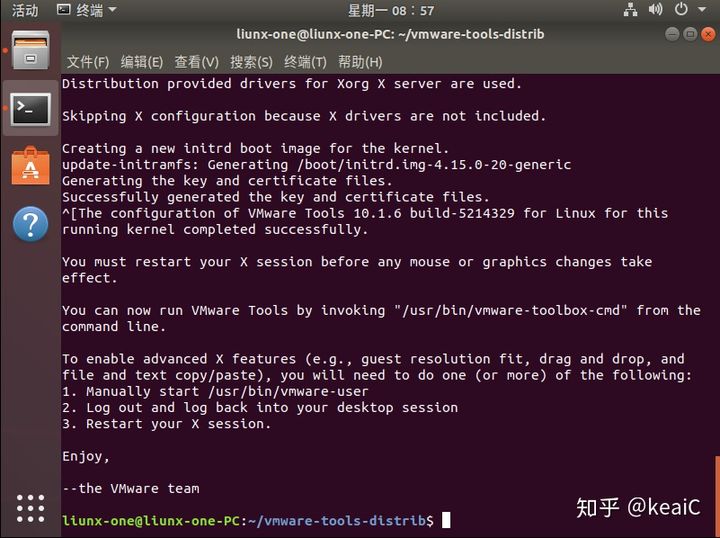


然后在ubuntu中打开终端，输入“ls”-->“cd vmware-tools-distrib”-->“sudo ./vmware-install.pl”

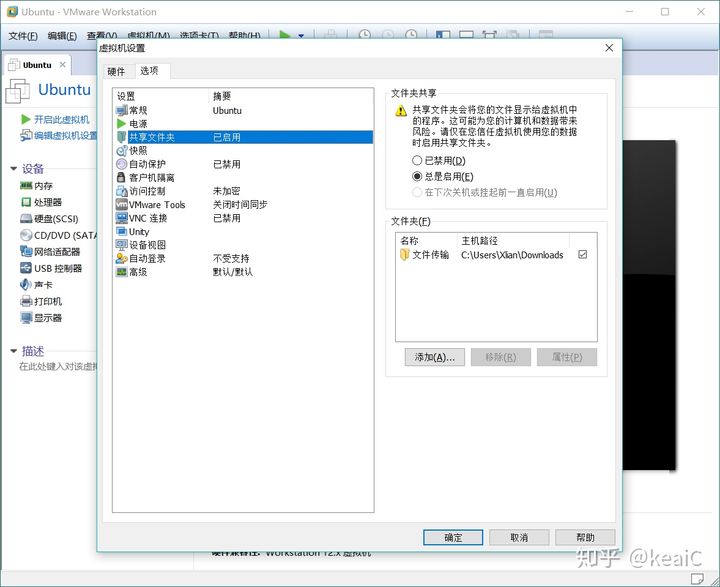
输入root密码，然后一路“y”加回车



等到下边这样就安装完成，然后重启ubuntu就OK了。



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

## 实验目的

（1） 理解 Hadoop 的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。

## 实验内容和相应结果

### 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.

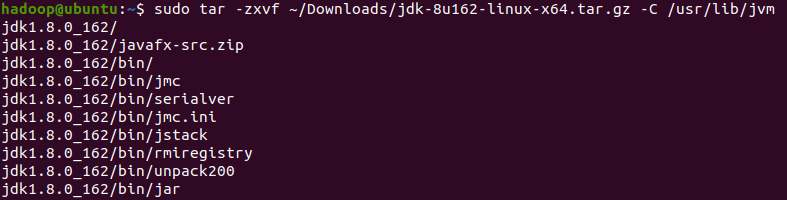
在 root 用户下:

$ 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.

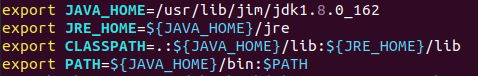
设置 JDK 的环境变量，考虑到 JDK 可能会有其他系统用户也会用到，建议将环境变量直 接设置在/etc/profile 中具体内容(如果没有则直接在 profile 文件中添加)：

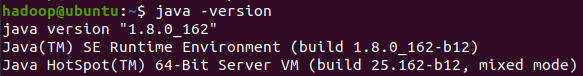
export JAVA\_HOME=/usr/jdk6

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.





### 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

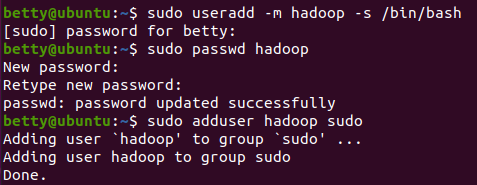
下载 hadoop 安装包并 copy 至 namenode 的 hadoop 用户的/home/hadoop/HadoopInstall 并解 压缩：

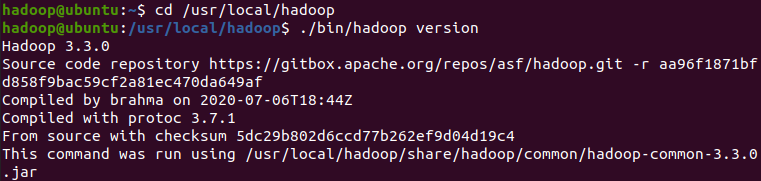
tar zxvf hadoop-0.16.3.tar.gz 考虑到今后升级以及其他操作的方便性，建议建一个名称为 hadoop 的链接，指向 hadoop-0.16.3 目录

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/





### 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

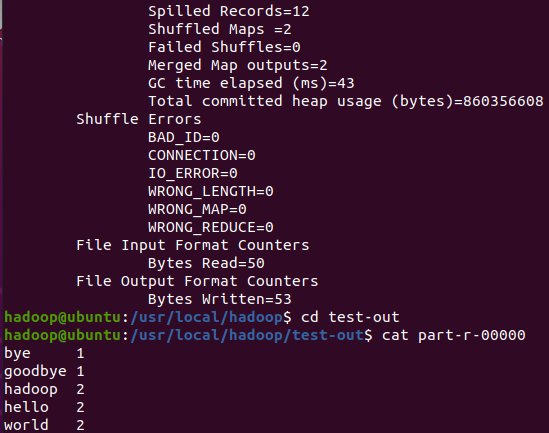
$ cat part-00000 bye 1

goodbye 1

hadoop 2

hello 2

world 2



# 实验七 Internet 应用

1. 学习Tomcat

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

1. 安装Tomcat

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

安装成功截图



1. 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) {  HttpSocketClient client = new HttpSocketClient();  try {  client.sendGet("www.baidu.com",80,"/");  } catch (IOException e) {  e.printStackTrace();  }  }  public HttpSocketClient() {  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>

1. 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>

1. 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

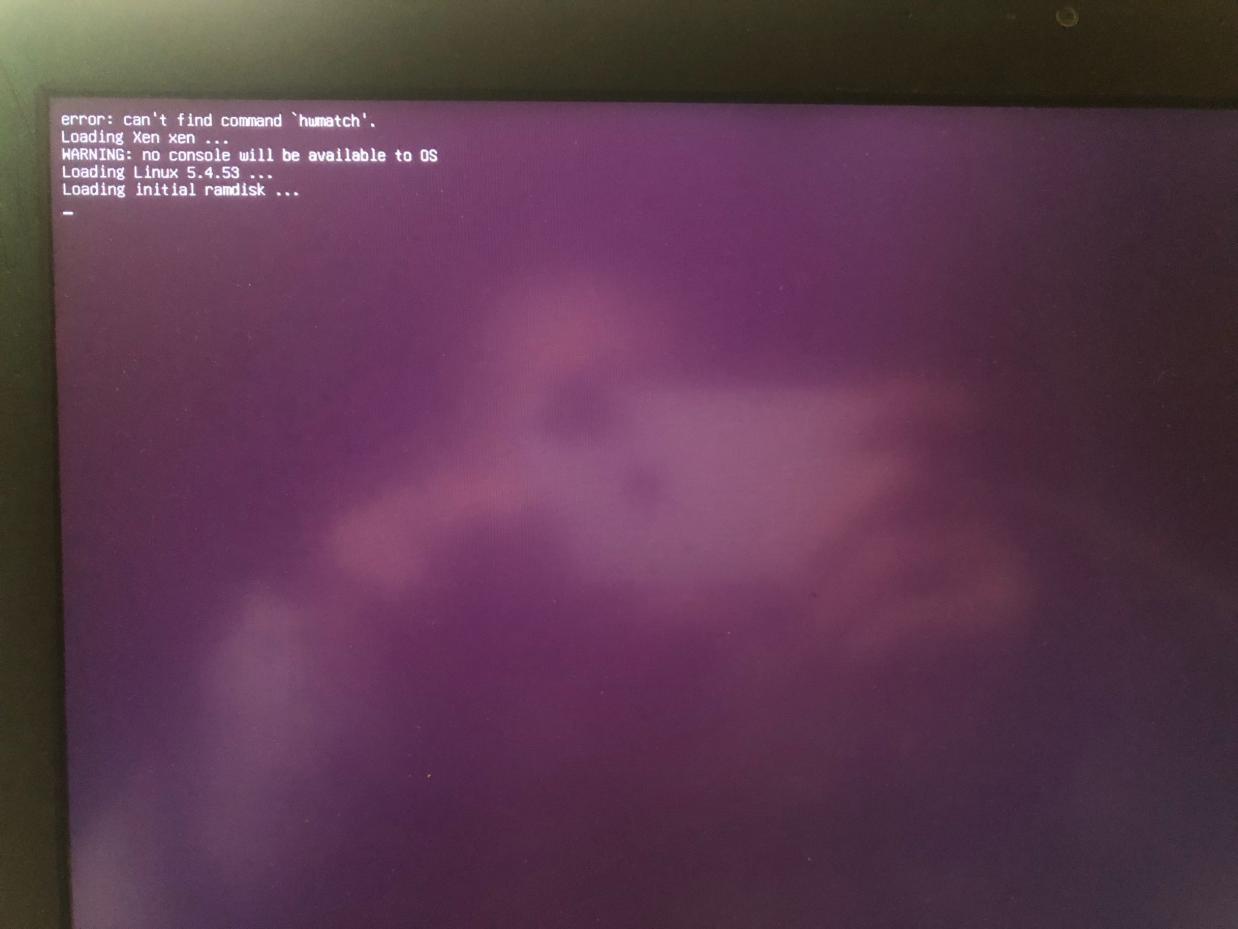
学习cookie：

Cookie、Session、Token的区别: <https://www.bilibili.com/video/BV1ob4y1Y7Ep?from=search&seid=8681360534596789764&spm_id_from=333.337.0.0>

Cookie使用步骤与Cookie原理: https://blog.csdn.net/weixin\_43691058/article/details/104052875?utm\_medium=distribute.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，我此前还失败的了一些实验。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。