

实验环境的搭建与Vim编辑器使用

一、资料准备

1. VirtualBox
2. Ubuntu系统镜像

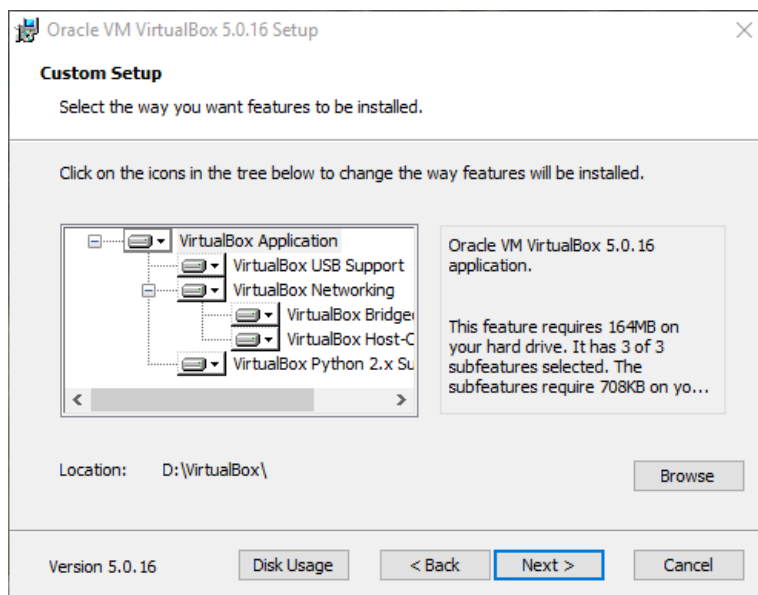
二、本次实验内容

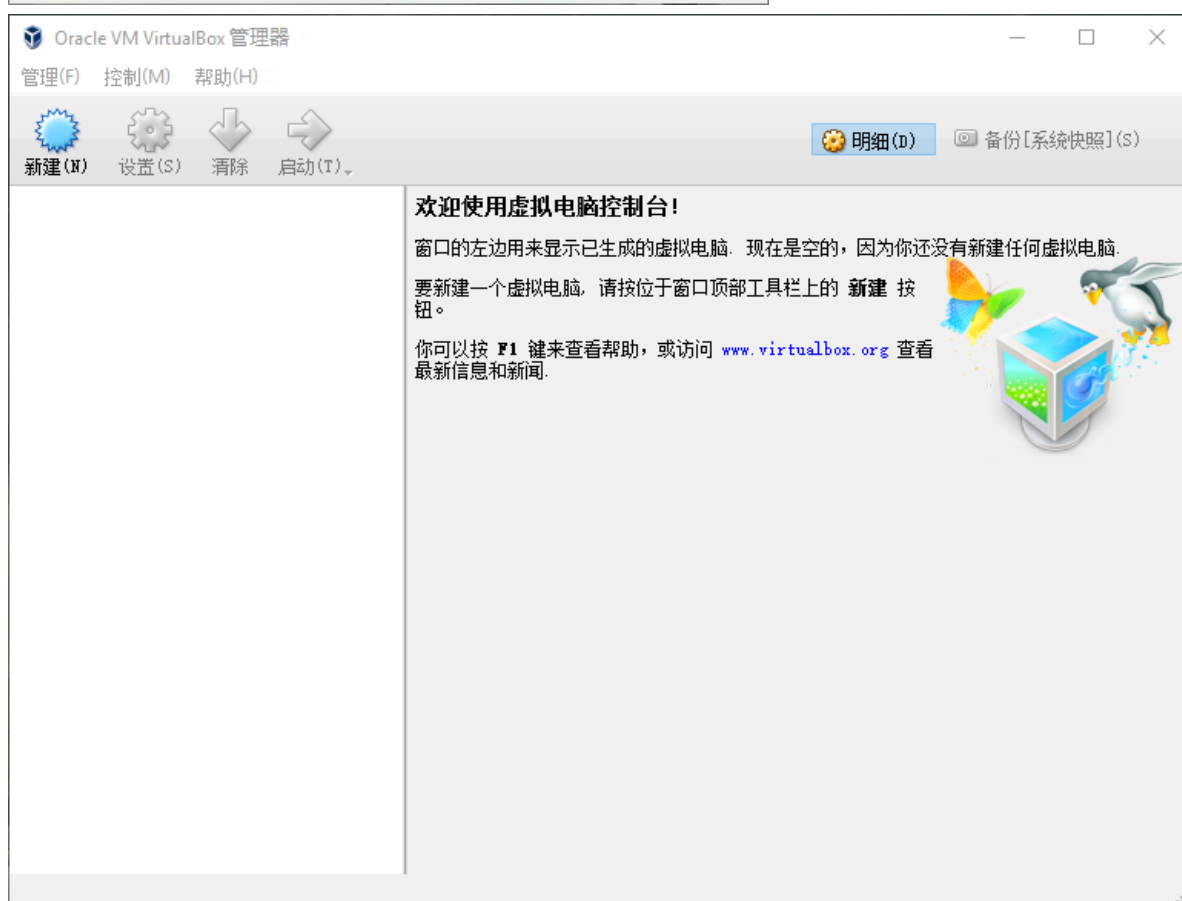
1. 在VirtualBox上安装Ubuntu
2. Vim Editor的使用入门

三、实验第一项:在VirtualBox上安装Ubuntu

1、安装VirtualBox

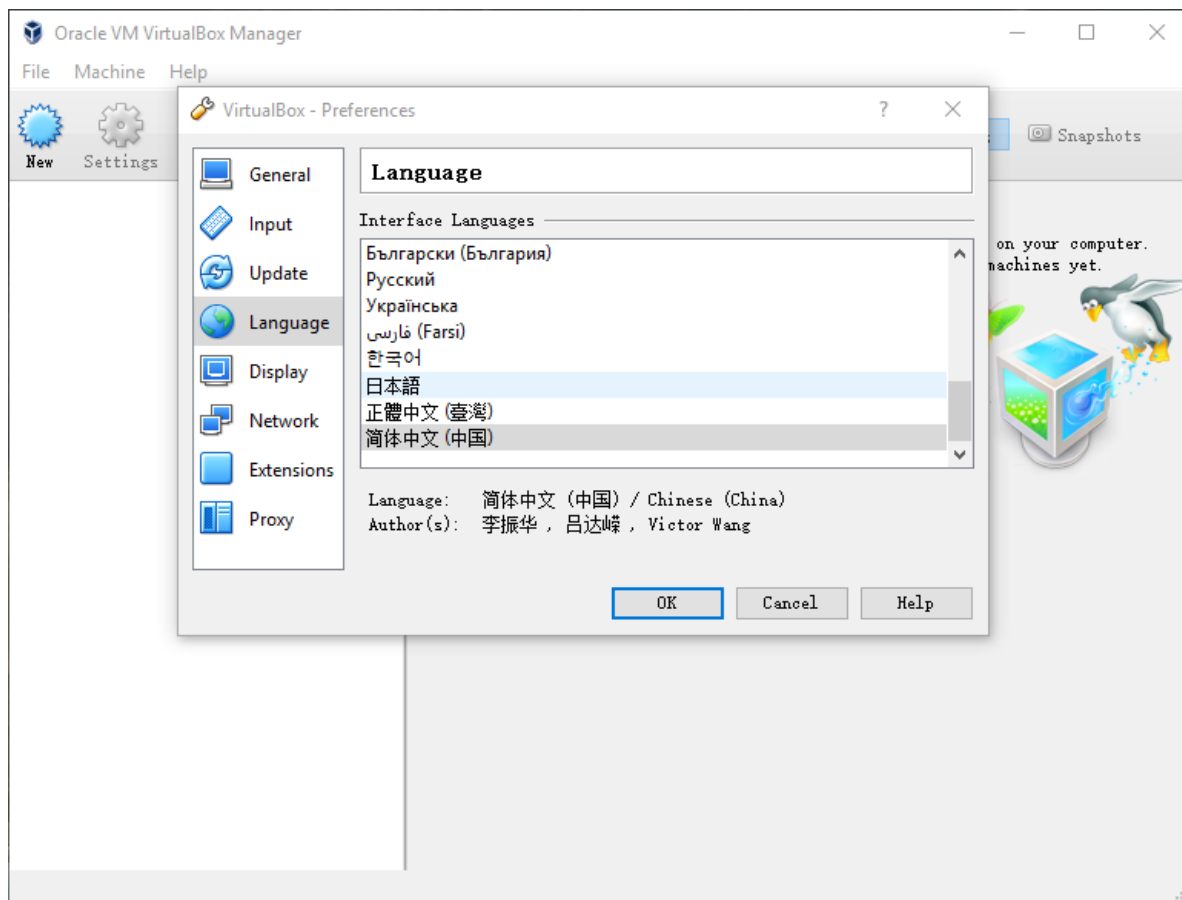
Step 1:





至此VirtualBox已安装成功

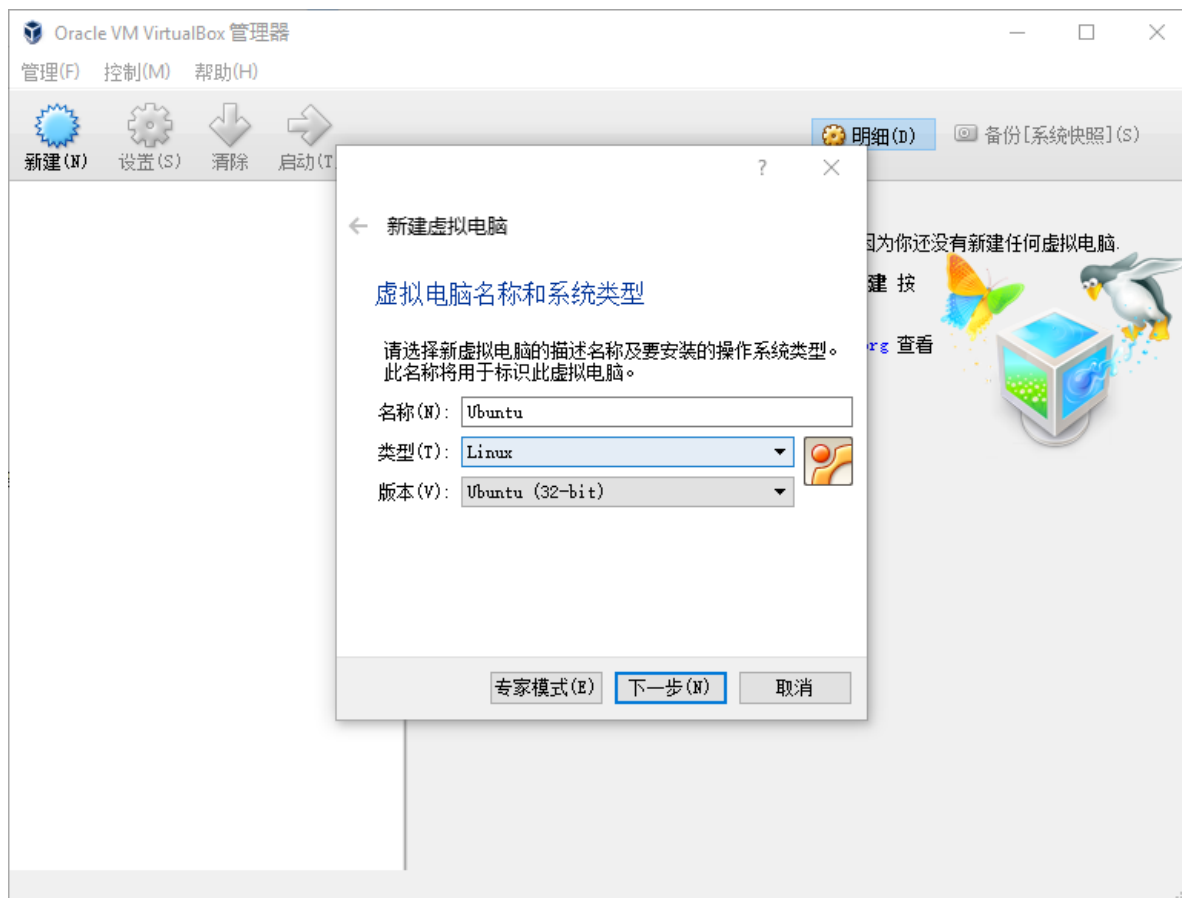
如果想将英文改成中文：File->Preferences->Language 切换至中文



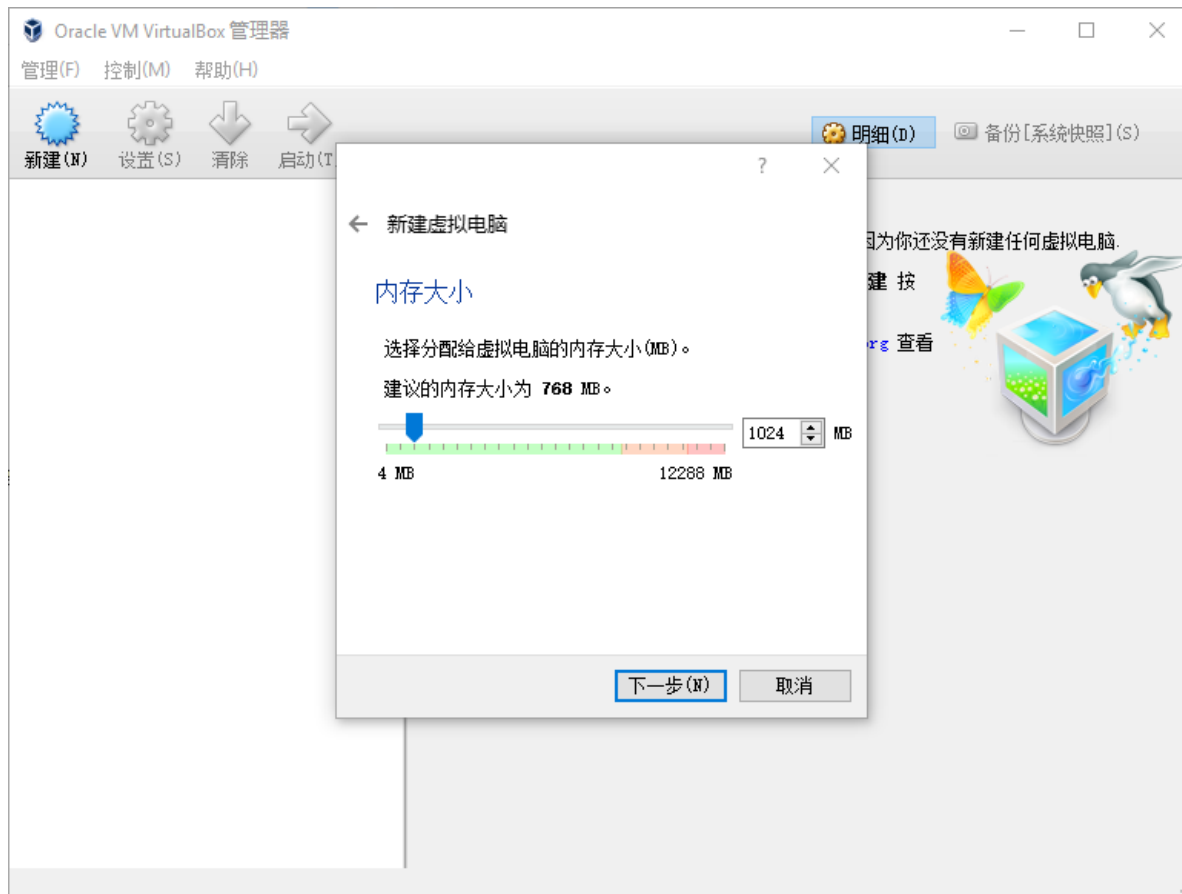
2、在VirtualBox上安装Ubuntu系统

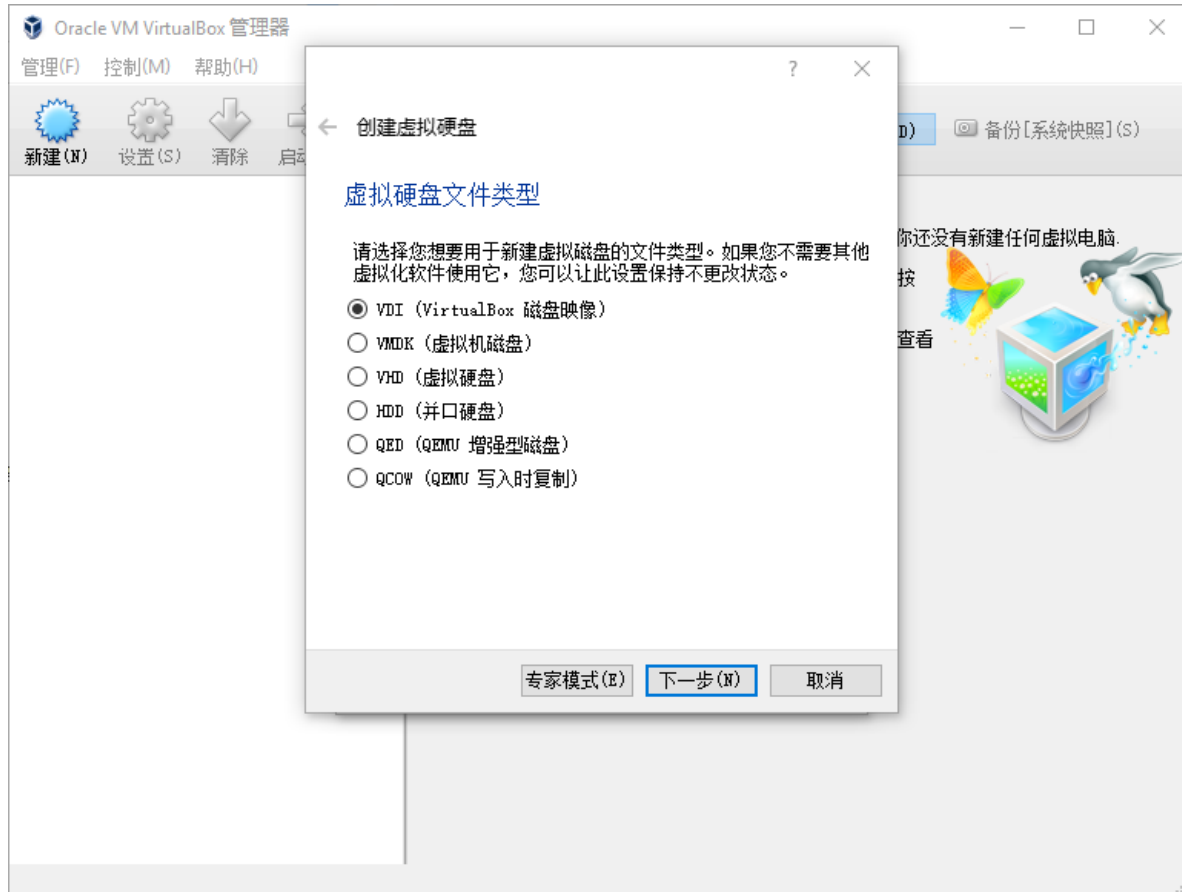
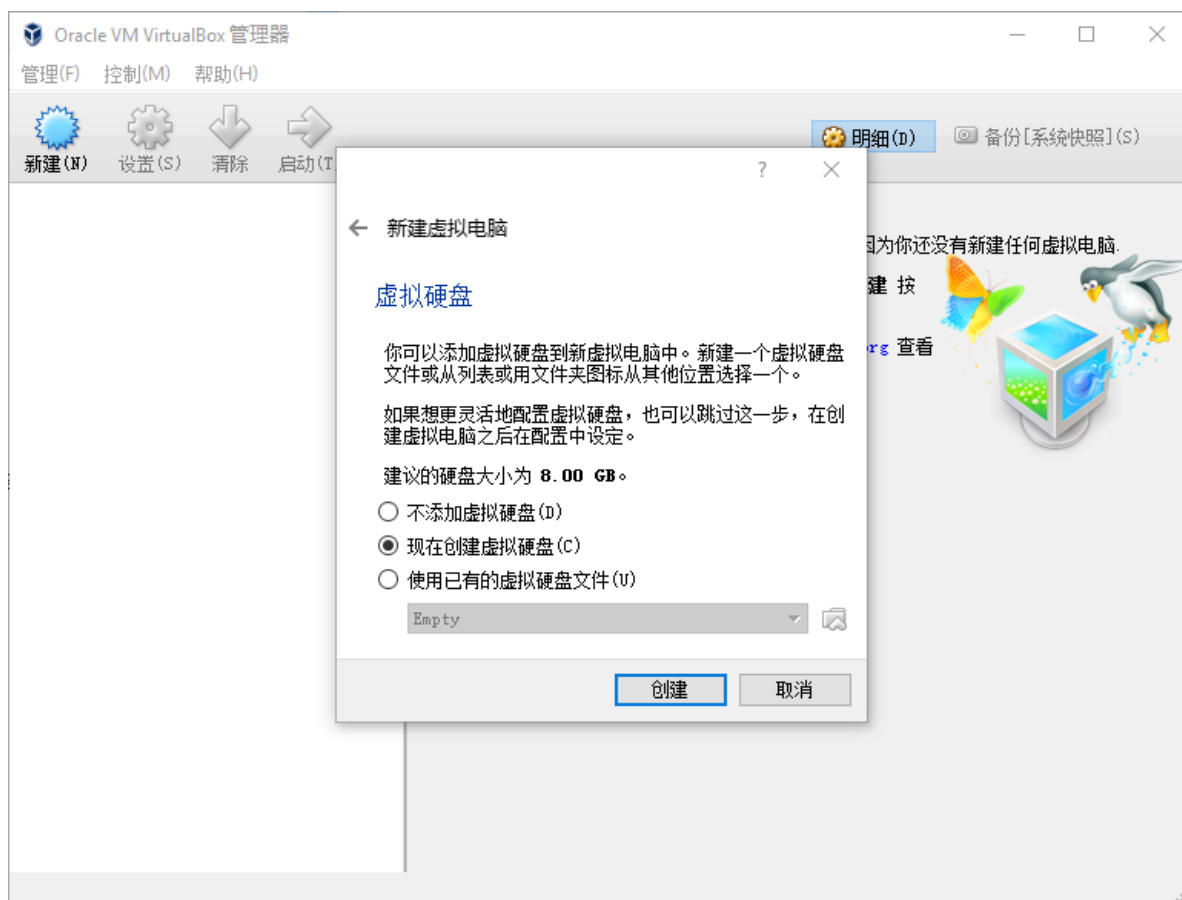
Step 1: 创建虚拟机

由于我们的Ubuntu系统镜像是32位，所以这里选择32位的版本

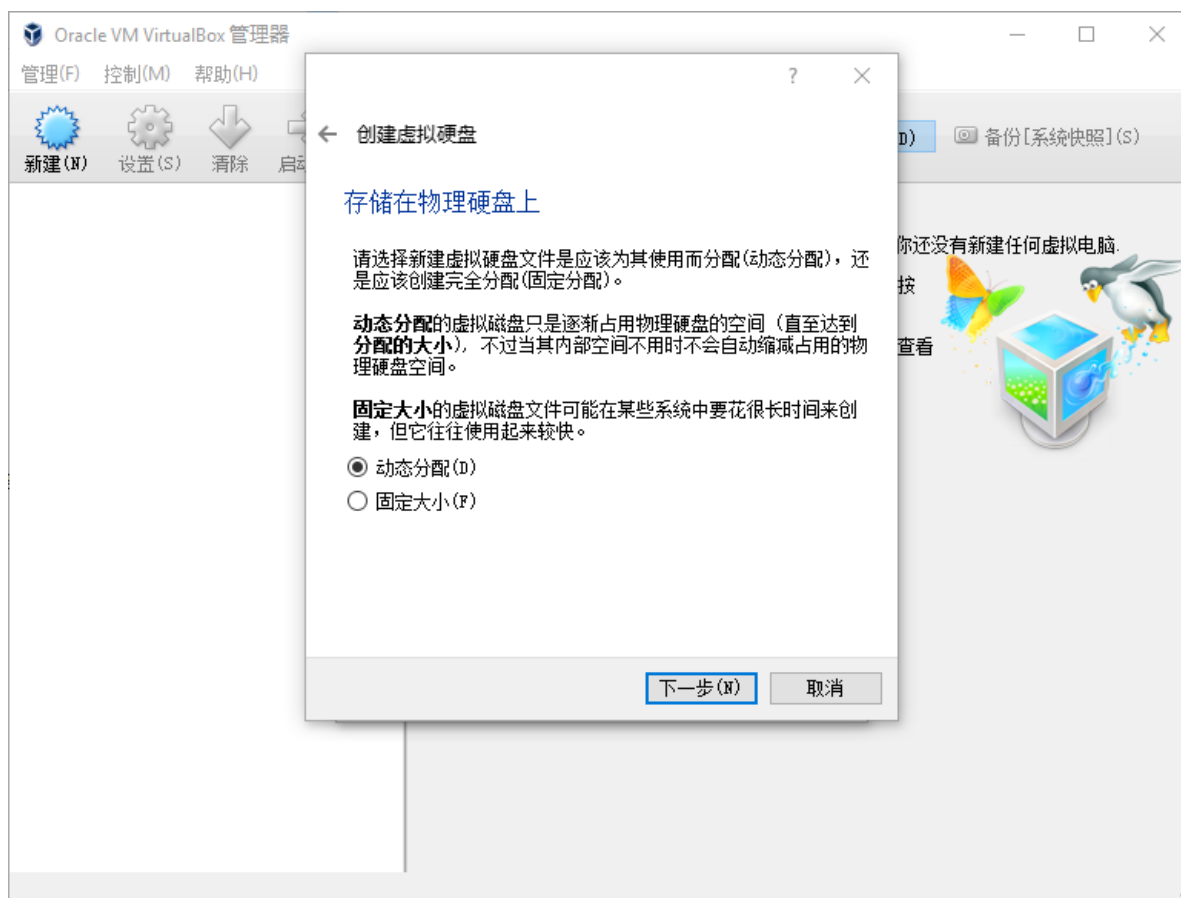


设置内存大小为1G





建议将硬盘大小设置为20G左右



Step 2: 在建好的虚拟机上安装 Ubuntu 系统

在建好的虚拟上右键，点击启动 出现如下界面



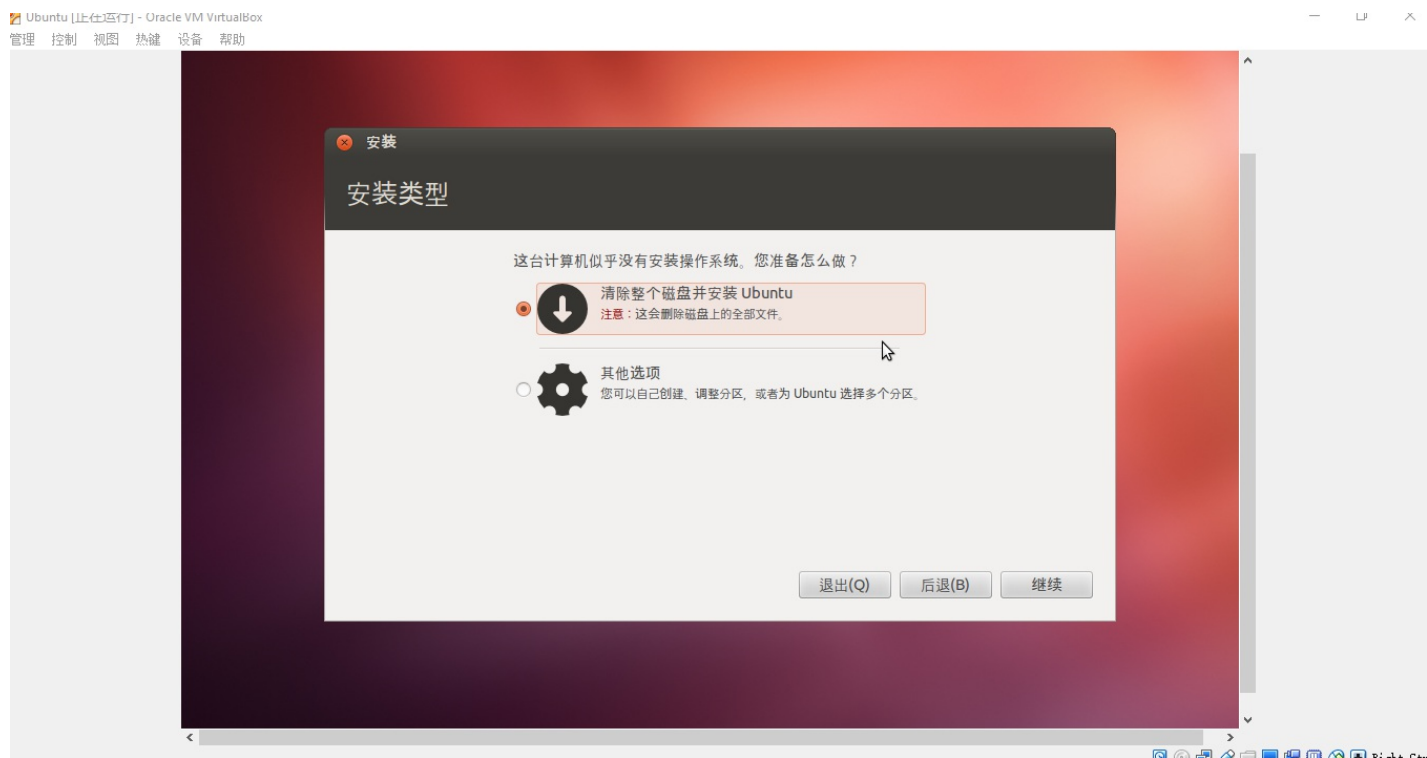
选择语言，然后点击"安装Ubuntu"



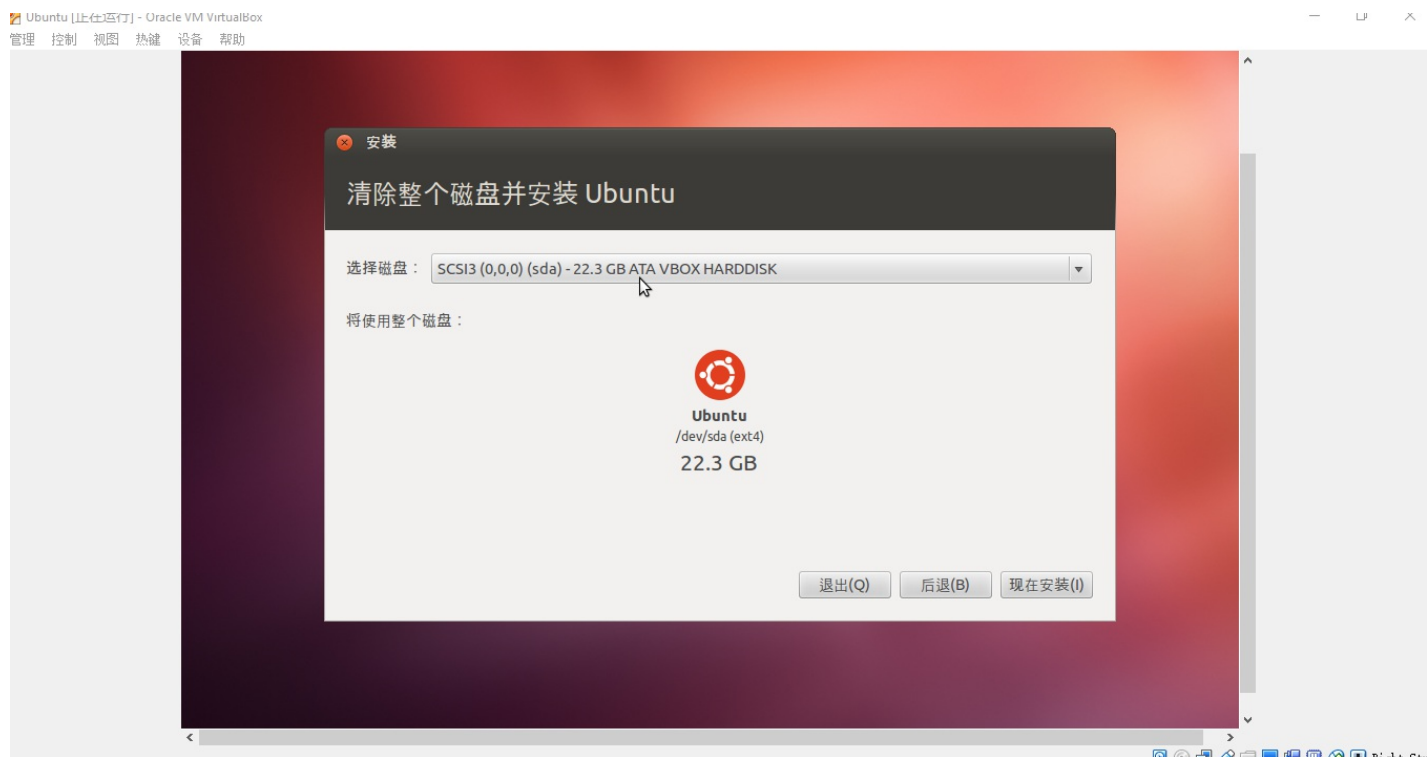
注意，不要选择下面的两个选择框。然后点击继续



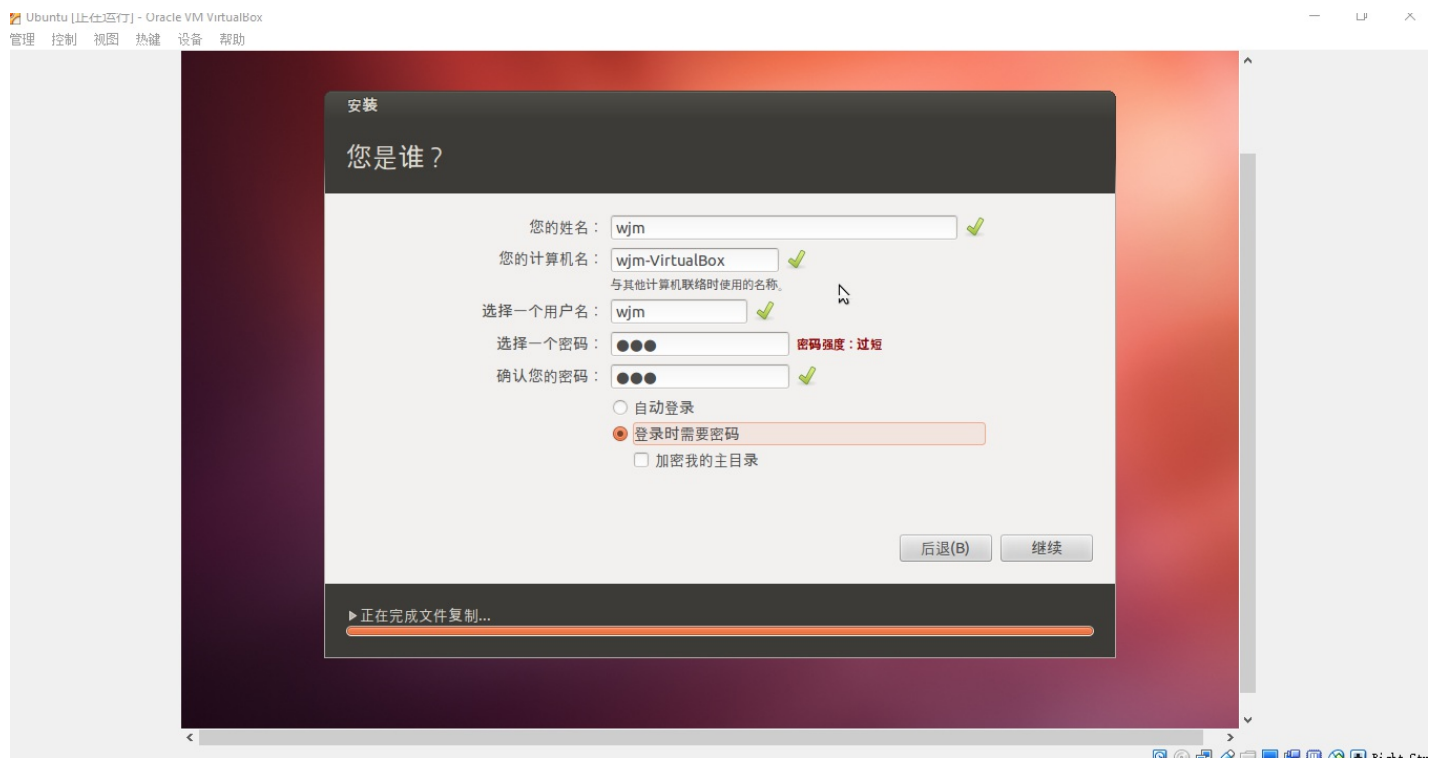
选择第一项，点击继续



点击现在安装



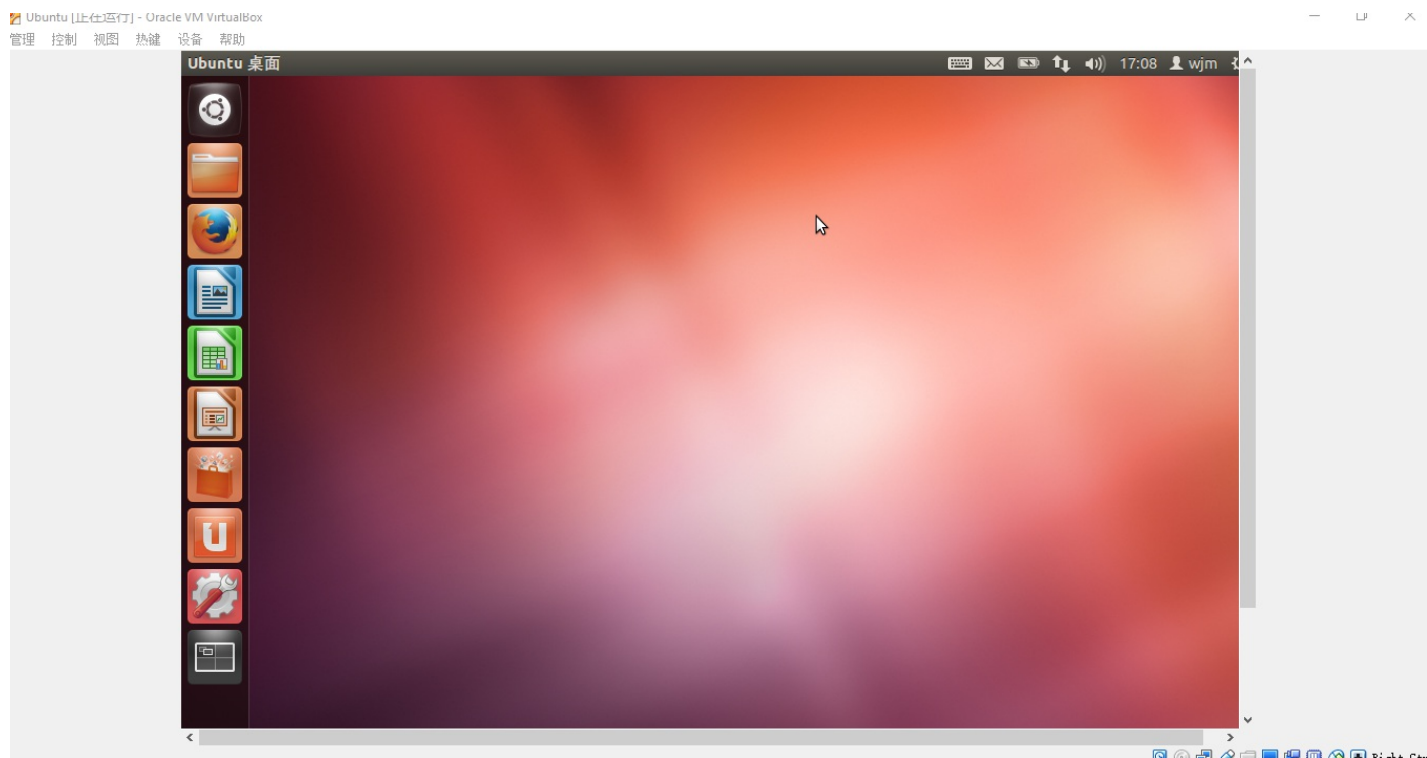
填写相关信息后，点击继续



进入正式的安装状态，等待ing.....



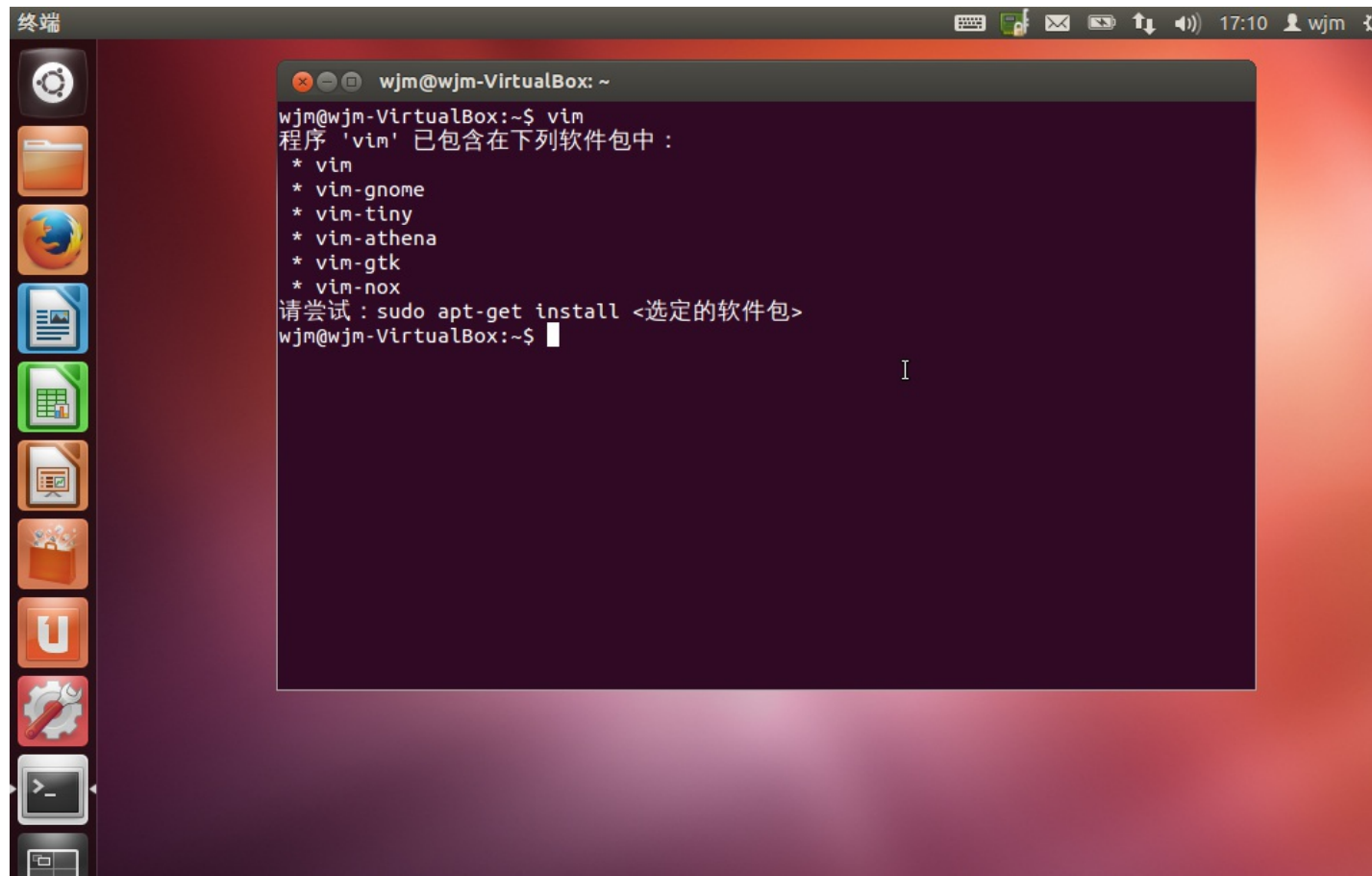
安装成功后，重启，进入系统（忘了账户密码的不要找我！）



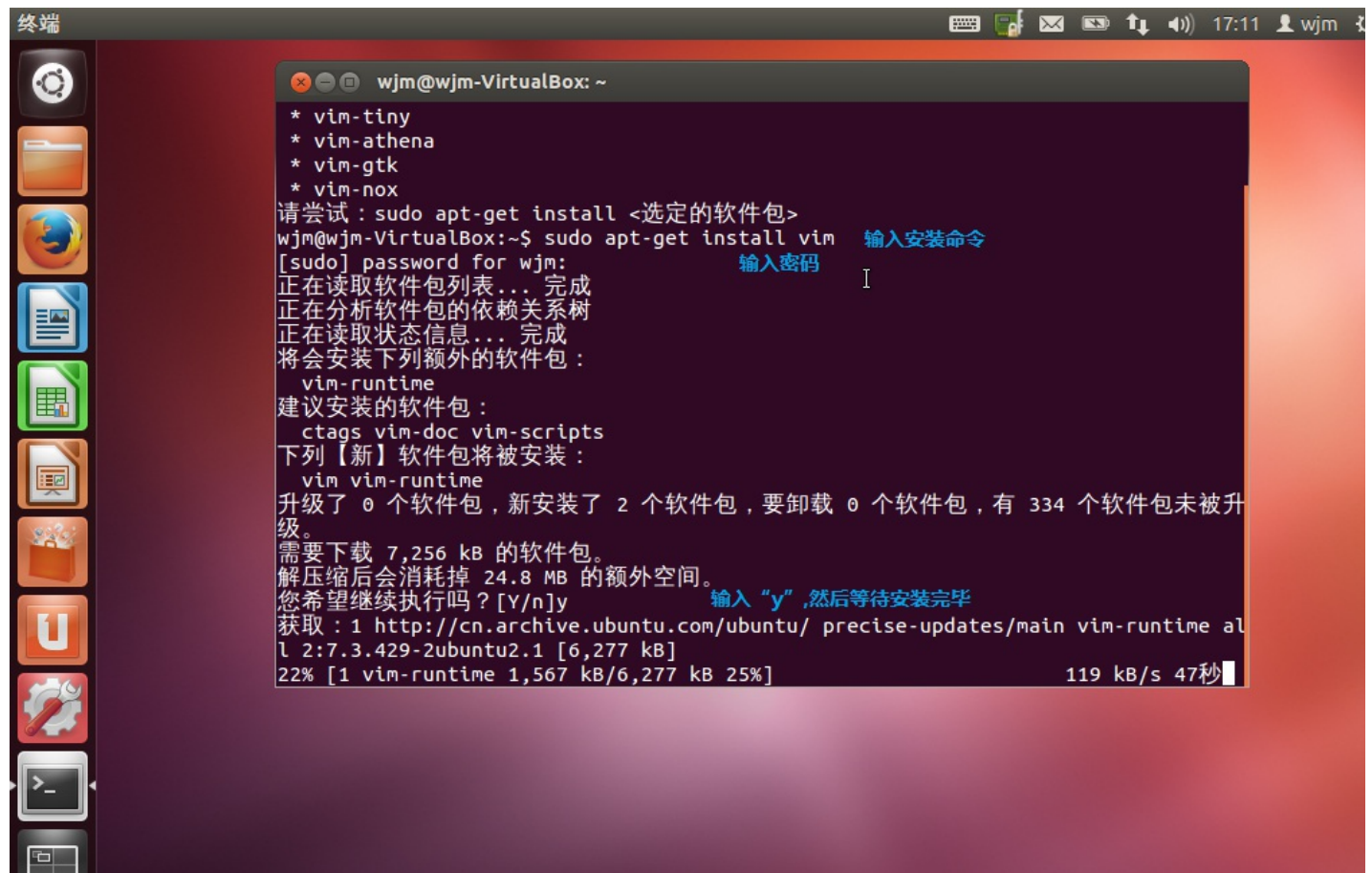
四、实验第二项:Vim Editor的安装与使用

1、Vim编辑器的安装

新安装的Ubuntu上并没有自带Vim编辑器



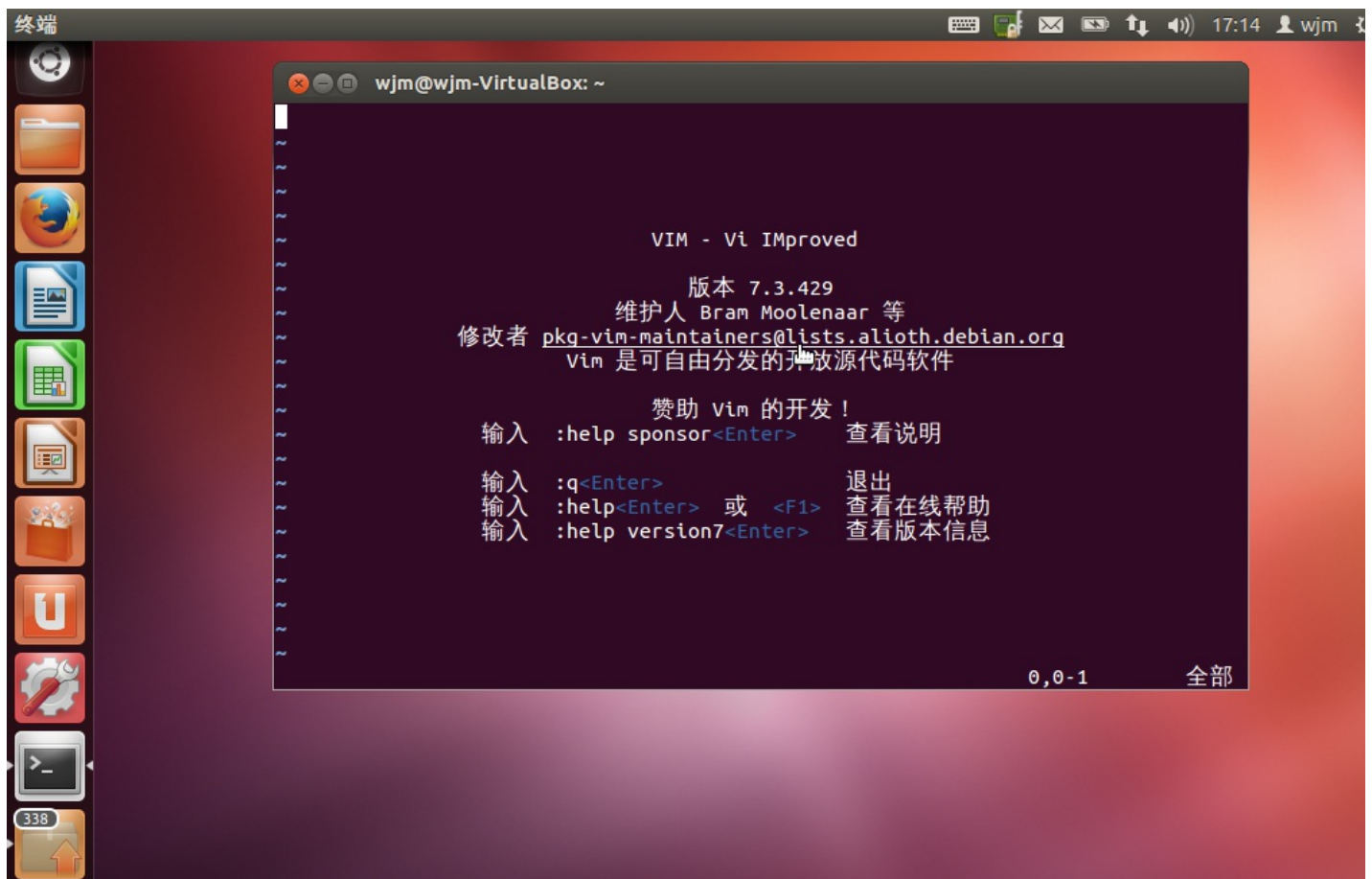
输入命令:sudo apt-get install vim 安装



The image shows a terminal window titled "wjm@wjm-VirtualBox: ~" with a dark background. The terminal output shows the command "sudo apt-get install vim" being executed. It lists available vim packages (vim-tiny, vim-athena, vim-gtk, vim-nox) and prompts the user to select one. The user selects "vim". The terminal then shows the password prompt, the progress of installing dependencies, and the final installation of vim and vim-runtime. It also shows the download progress of the packages.

```
wjm@wjm-VirtualBox: ~  
* vim-tiny  
* vim-athena  
* vim-gtk  
* vim-nox  
请尝试 : sudo apt-get install <选定的软件包>  
wjm@wjm-VirtualBox:~$ sudo apt-get install vim 输入安装命令  
[sudo] password for wjm: 输入密码  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
将会安装下列额外的软件包 :  
  vim-runtime  
建议安装的软件包 :  
  ctags vim-doc vim-scripts  
下列【新】软件包将被安装 :  
  vim vim-runtime  
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 334 个软件包未被升级。  
需要下载 7,256 kB 的软件包。  
解压缩后会消耗掉 24.8 MB 的额外空间。  
您希望继续执行吗？[Y/n]y 输入 "y", 然后等待安装完毕  
获取：1 http://cn.archive.ubuntu.com/ubuntu/ precise-updates/main vim-runtime al  
l 2:7.3.429-2ubuntu2.1 [6,277 kB]  
22% [1 vim-runtime 1,567 kB/6,277 kB 25%] 119 kB/s 47秒
```

在终端中输入:vim 命令即可进入编辑器界面



2、Vim编辑器的使用入门

要问Ubuntu上的编辑器哪家强，就找我Vim!

Step 1:认识Vim编辑器

- Vim是从 vi 发展出来的一个文本编辑器。代码补全、编译及错误跳转等方便编程的功能特别丰富，在程序员中被广泛使用，和Emacs并列成为类Unix系统用户最喜欢的文本编辑器

Step 2:理解Vim的三种模式

- 插入模式（输入模式）：进入Vim后，按下i进入插入模式，按下键从插入模式切换至编辑模式
- 编辑模式：进入Vim后默认进入编辑模式，该模式用于移动和操纵文本
- 命令模式：用于执行冒号命令，比如保存，查找/替换以及配置vim等

Step 3: 熟记Vim常用命令

命令历史

以:和/开头的命令都有历史纪录，可以首先键入:或/然后按上下箭头来选择某个历史命令

启动Vim

- 在命令行窗口（终端）中输入以下命令即可
 - vim 直接启动vim
 - vim filename 打开vim并创建名为filename的文件

文件命令

- vim file 打开单个文件
- vim file1 file2 file3 ... 同时打开多个文件

- :open file 在vim窗口中打开一个新文件
- :split file 在新窗口中打开文件
- :bn 切换到下一个文件
- :bp 切换到上一个文件

插入命令

- i 在当前位置前插入
- I 在当前行首插入
- a 在当前位置后插入
- A 在当前行尾插入
- o 在当前行之后插入一行
- O 在当前行之前插入一行

查找命令

- /text 查找text, 按n键查找下一个, 按N键查找前一个。
- ?text 查找text, 反向查找, 按n键查找下一个, 按N键查找前一个。

替换命令

- ra 将当前字符替换为a, 当期字符即光标所在字符。
- s/old/new/ 用old替换new, 替换当前行的第一个匹配
- s/old/new/g 用old替换new, 替换当前行的所有匹配
- %s/old/new/ 用old替换new, 替换所有行的第一个匹配
- %s/old/new/g 用old替换new, 替换整个文件的所有匹配

删除命令

- x 删除当前字符
- 3x 删除当前光标开始向后三个字符
- X 删除当前字符的前一个字符。X=dh
- dl 删除当前字符, dl=x
- dh 删除前一个字符
- dd 删除当前行
- dj 删除上一行
- dk 删除下一行
- 10d 删除当前行开始的10行。

粘贴拷贝命令

- yy 拷贝当前行
- nyy 拷贝当前后开始的n行, 比如2yy拷贝当前行及其下一行。
- p 在当前光标后粘贴,如果之前使用了yy命令来复制一行, 那么就在当前行的下一行粘贴。
- shift+p 在当前行前粘贴

退出命令

- :wq 保存并退出
- ZZ 保存并退出
- :q! 强制退出并忽略所有更改
- :e! 放弃所有修改, 并打开原来文件。

窗口命令

- :split或new 打开一个新窗口, 光标停在顶层的窗口上
- :split file或:new file 用新窗口打开文件
- Ctrl+ww 移动到下一个窗口
- Ctrl+wj 移动到下方的窗口
- Ctrl+wk 移动到上方的窗口

- :close 最后一个窗口不能使用此命令，可以防止意外退出vim。
- :q 如果是最后一个被关闭的窗口，那么将退出vim。
- ZZ 保存并退出。
- :only 关闭所有窗口，只保留当前窗口

帮助命令

- :help or F1 显示整个帮助
- :help xxx 显示xxx的帮助，比如 :help i, :help CTRL-[（即Ctrl+[的帮助）。
- :help 'number' Vim选项的帮助用单引号括起
- :help 特殊键的帮助用<>扩起
- :help -t Vim启动参数的帮助用