



插曲之代码管理 - Github 简单使用

主讲人：曹家英



第一部分：树莓派底层硬件

- 1. 树莓派介绍
- 2. 树莓派环境搭建
- 3. 树莓派与命令行
- 4. 相关知识扩展
- 5. 树莓派基础外设操作
- 6. 树莓派Buildroot SDK环境搭建



Git基本操作

Github远程仓库

Jupyter notebook

Markdown



虚拟机的安装使用



虚拟机——准备工作

软件准备:

Windows

Vmware Workstation

VirtualBox

Mac:

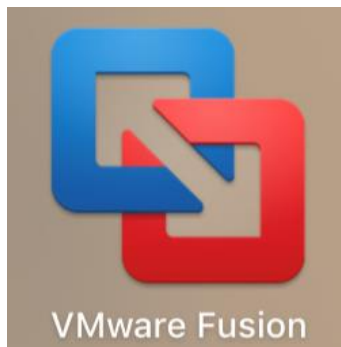
PD

VM Fusion

VirtualBox

安装镜像:

Ubunt 18.04



<https://www.ubuntu.com/desktop>

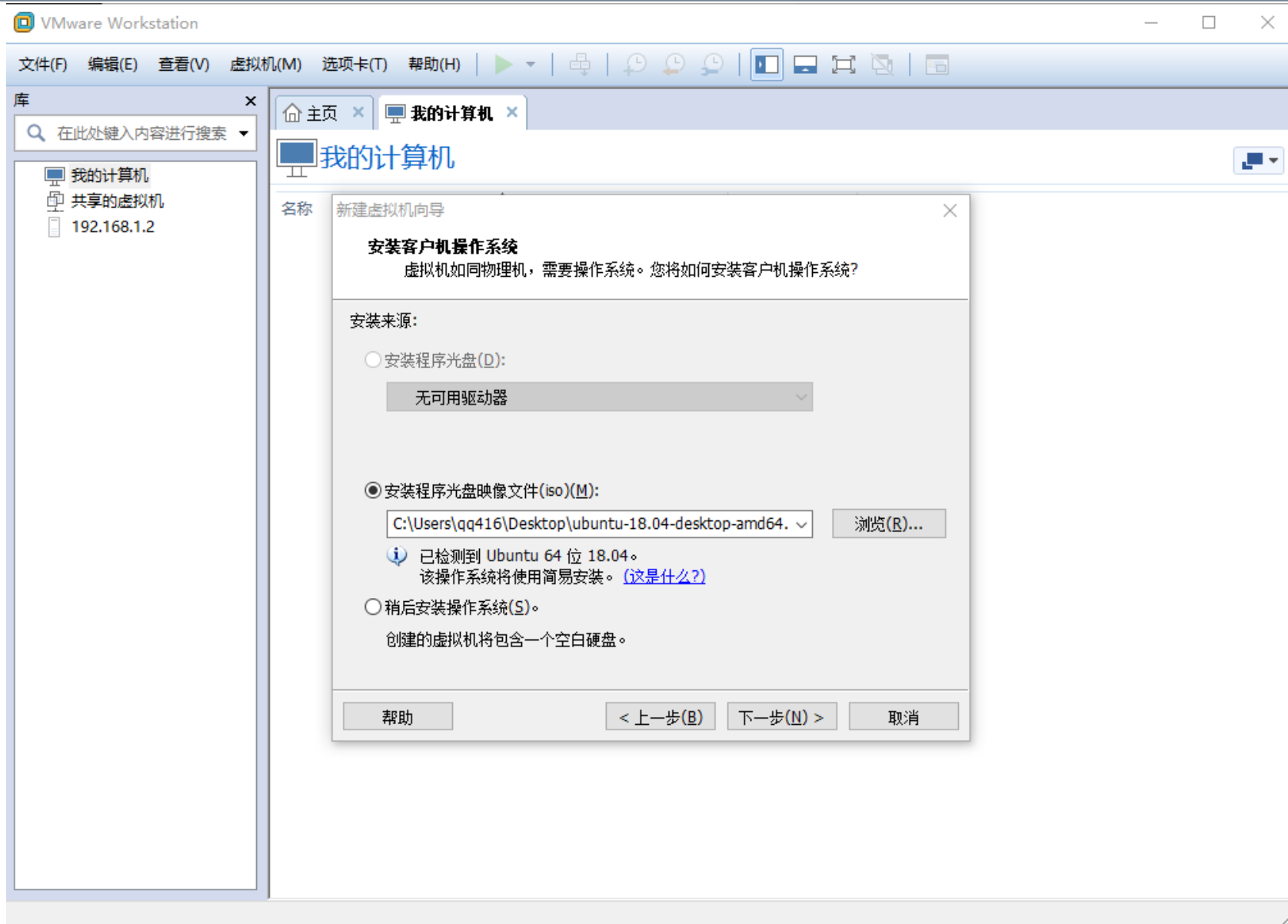


安装虚拟机



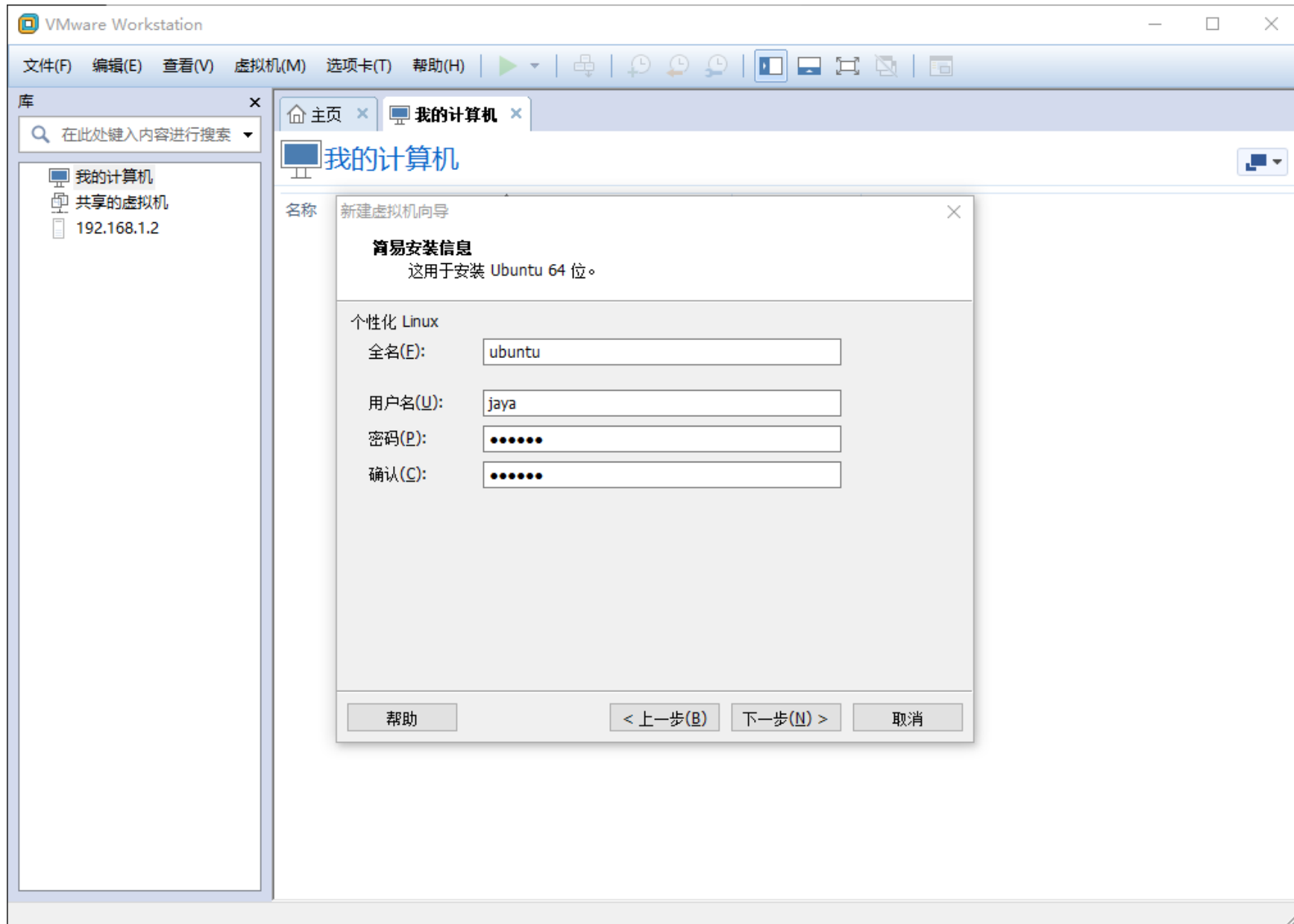


安装虚拟机



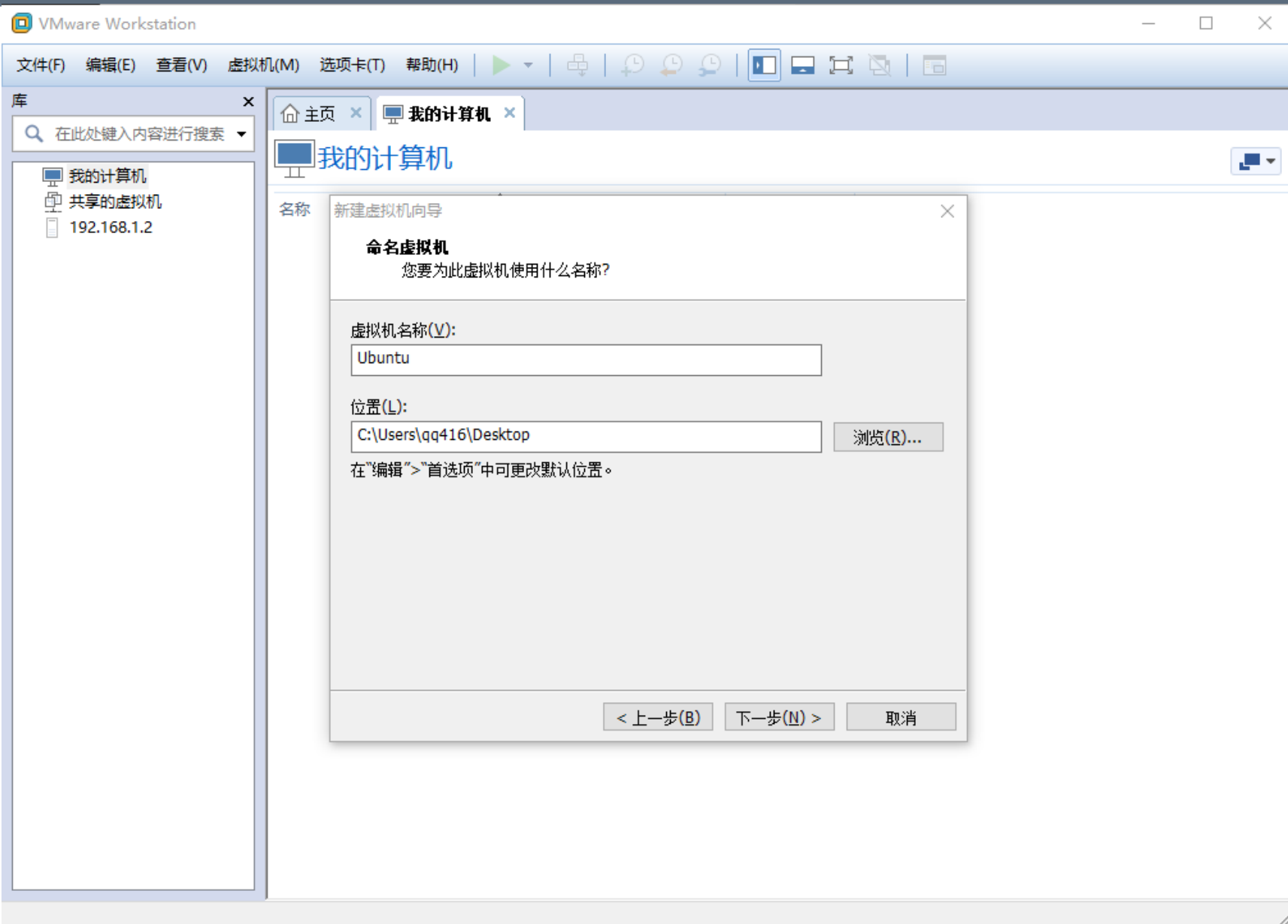


安装虚拟机



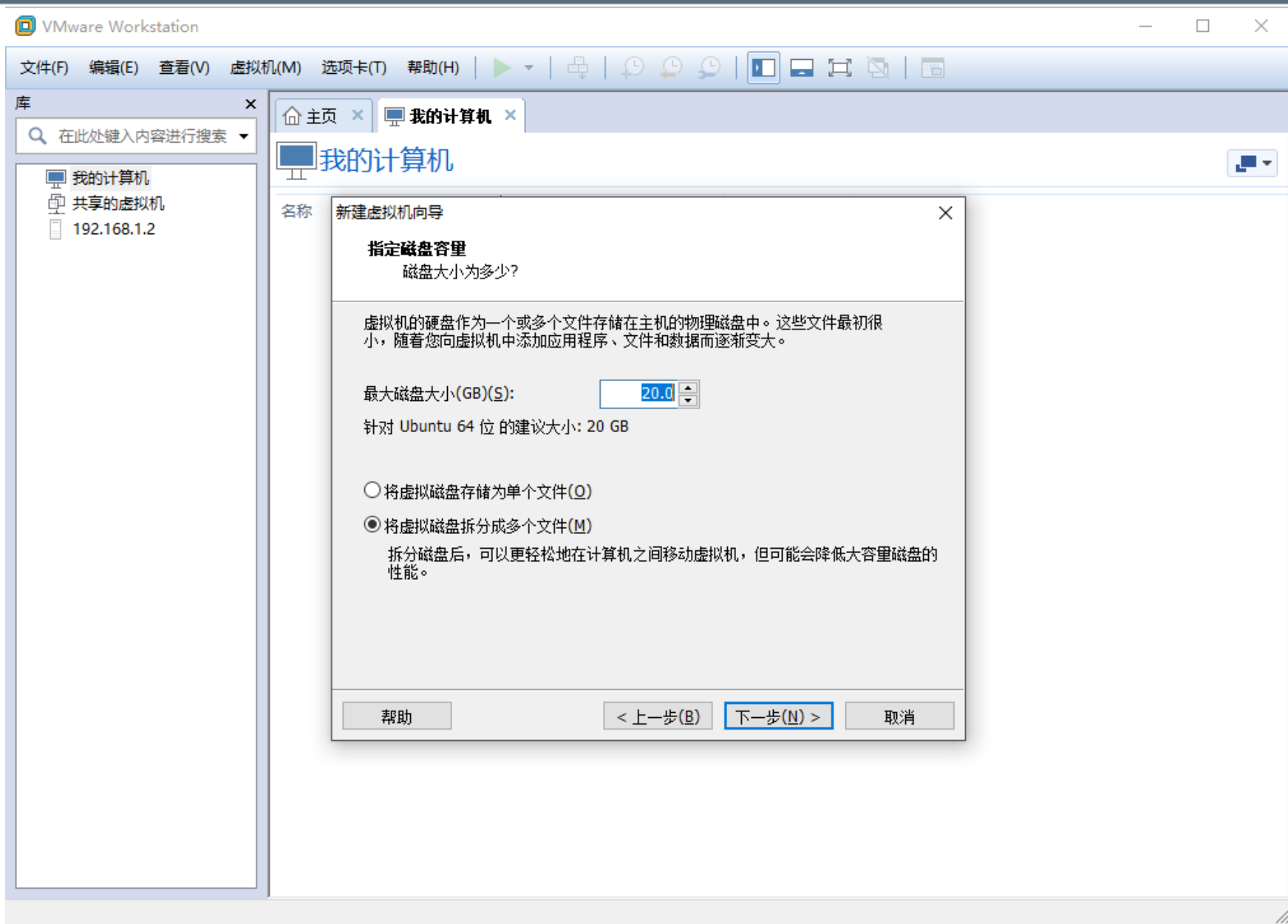


安装虚拟机





安装虚拟机





安装虚拟机

VMware Workstation

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，并开始安装 Ubuntu

将使用下列设置创建虚拟机：

名称： Ubuntu
位置： C:\Users\qq416\Desktop
版本： Workstation 12.x
操作系统： Ubuntu 64 位

硬盘： 20 GB, 拆分
内存： 1024 MB
网络适配器： NAT
其他设备： 2 个 CPU 核心, CD/DVD

自定义硬件(C)...

☒ 创建后开启此虚拟机(P)

< 上一步

硬件

设备	摘要
内存	2 GB
处理器	2
新 CD/DVD (SATA)	正在使用文件 C:\Users\qq416\Desktop\...
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

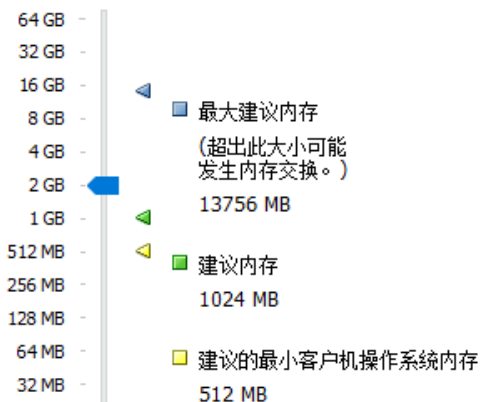
添加(A)...

移除(R)

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。

此虚拟机的内存(M): 2048 MB



虚拟机最多将此内存的 768 MB 用作图形内存。您可以在“显示器”设置页面中更改此数量。

关闭

帮助



安装虚拟机

Ubuntu - VMware Workstation

文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H) | [Icons]

库

在此处键入内容进行搜索

- 我的计算机
 - Ubuntu
 - 共享的虚拟机
 - 192.168.1.2

主页 | 我的计算机 | Ubuntu

```
ISOLINUX 3.82 2009-06-09 ETCD Copyright (C) 1994-2009 H. Peter Anvin et al
[ 7.156821] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
[ 7.240198] sd 2:0:0:0: [sda] Assuming drive cache: write through
```

单击虚拟屏幕
可发送按键

简易安装正在安装 Ubuntu 64 位。

帮助

要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

[Icons]



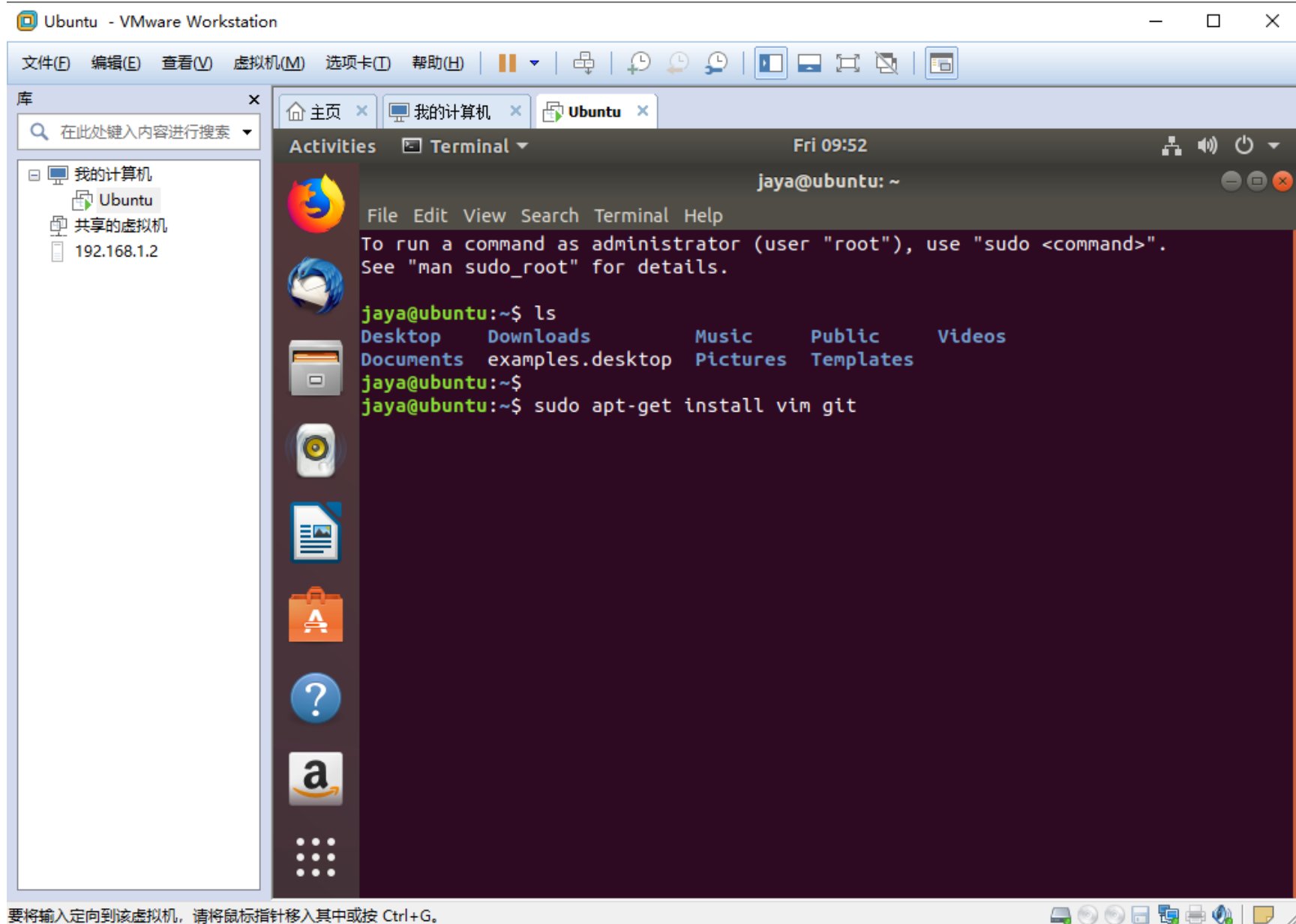
升级到 Ubuntu 18.04 LTS 的理由

1. 默认桌面环境由Unity桌面环境更换为GNOME 3.28
2. 安全性能增加
3. 更新内核, Linux Kernel 4.15





安装虚拟机





安装虚拟机——软件的安装

添加Atom源：

```
sudo add-apt-repository ppa:webupd8team/atom
```

更新源

```
sudo apt-get update
```

安装Atom

```
sudo apt-get install atom
```

必要软件的安装

```
sudo apt-get install vim git aptitude
```

```
sudo aptitude install python3-pip
```

Git基本操作



Git——安装

Git是目前最先进的分布式版本控制系统。

下载地址：

<https://git-scm.com/downloads>

Windows：

下载Windows安装包进行安装。



Linux Ubuntu：

使用sudo apt-get install git

MacBook：

安装Xcode就会自动安装Git

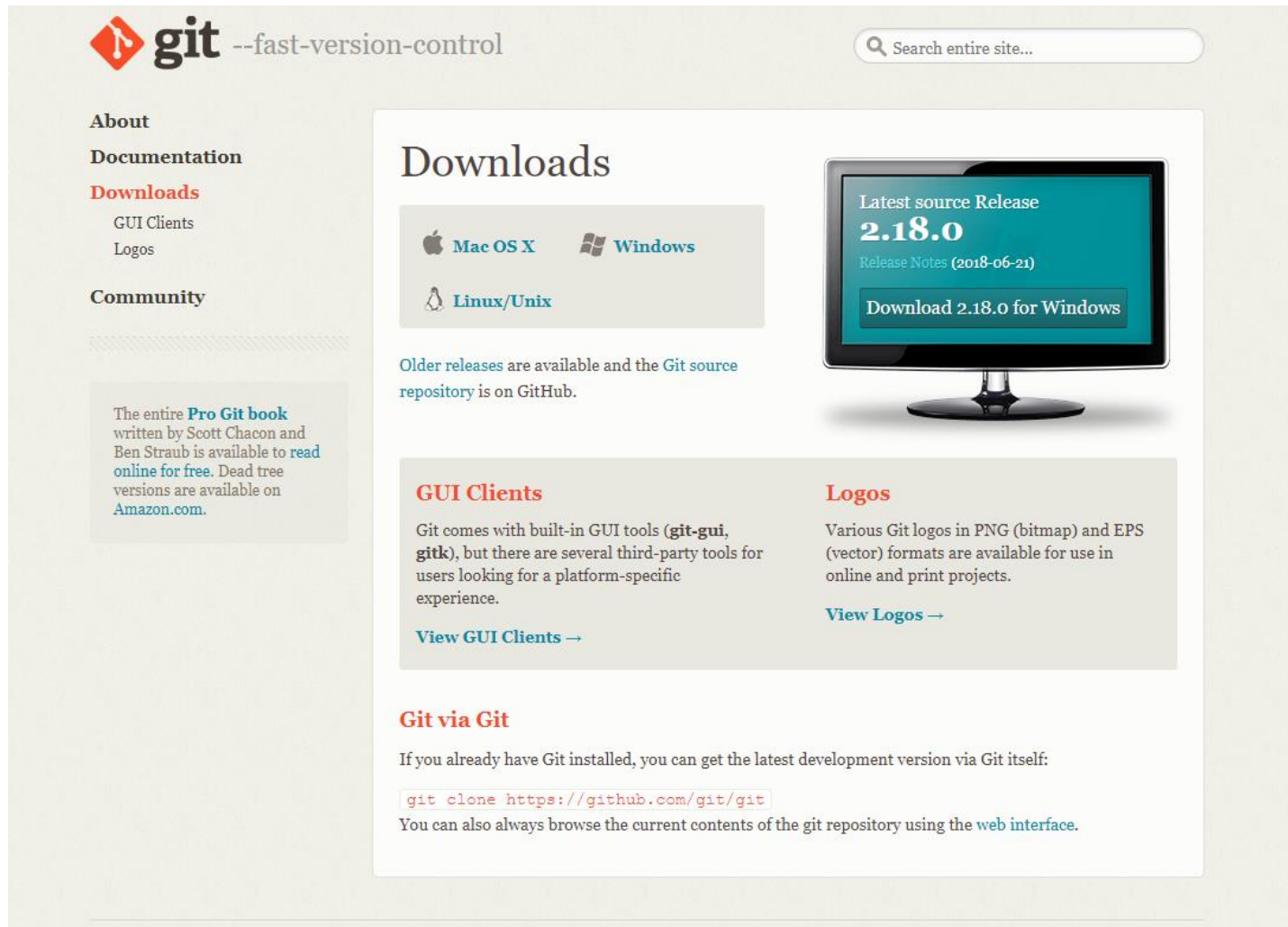
也可以使用：

brew install git

安装完成后：

Windows可以右键Git Bash here

MAC Linux 都可以直接在Sheel中操作





Git简单使用——建立第一个版本库

基本配置：

\$ git config --global user.name "用户名"

\$ git config --global user.email "邮箱"

\$ git config --list

GIT 简单使用：

进行初始化Git仓库： git init

```
MINGW64/c/Users/qq416/Desktop/testgit
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop
$ mkdir testgit
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop
$ cd testgit/
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop/testgit
$ ls
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop/testgit
$ git init
Initialized empty Git repository in C:/Users/qq416/Desktop/testgit/.git/
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop/testgit (master)
$ ls -al
total 36
drwxr-xr-x 1 qq416 197609 0 7月 28 11:31 ./
drwxr-xr-x 1 qq416 197609 0 7月 28 11:31 ../
drwxr-xr-x 1 qq416 197609 0 7月 28 11:31 .git/
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop/testgit (master)
$
```

```
MINGW64/c/Users/qq416/Desktop/testgit
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop/testgit (master)
$ git config --global user.name "jaya_cao"
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop/testgit (master)
$ git config --global user.email "416296378@qq.com"
qq416@Jaya_Nootbook MINGW64 /c/Users/qq416/Desktop/testgit (master)
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=C:/RJ/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.required=true
credential.helper=manager
merge.tool=kdiff3
diff.guitool=kdiff3
core.editor="C:/RJ/GitExtensions/GitExtensions.exe" fileeditor
core.autocrlf=true
difftool.kdiff3.path=C:/RJ/KDiff3/kdiff3.exe
mergetool.kdiff3.path=C:/RJ/KDiff3/kdiff3.exe
user.name=jaya_cao
user.email=416296378@qq.com
i18n.filesencoding=utf-8
```



Git简单使用——常用语句

`git status`

查看当前状态

`git add Readme.md`

将要追踪的文件添加进暂存区域

`git commit -m "Old Version!"`

将暂存区域的文件保存在仓库中

`git log`

log命令可以显示所有提交过的版本信息。

`git reset --hard 87aba1`

回退到87aba1版本。

--hard 彻底回退，

--soft 回退到某个版本，只回退了commit的信息

`git reflog`

可以查看所有分支的所有操作记录

`git checkout 87aba1`

回到老的版本

`git checkout -- file`

撤销某个文件的修改(和暂存区域对比)

`git rm file`

从暂存区域删除文件

`git diff file`

1.当暂存区中没有文件时，`git diff`比较的是，工作区中的文件与上次提交到版本库中的文件。

2.当暂存区中有文件时，`git diff`则比较的是，当前工作区中的文件与暂存区中的文件。

`git diff HEAD -- file`

比较的是工作区中的文件与版本库中文件的差异。HEAD指向的是版本库中的当前版本

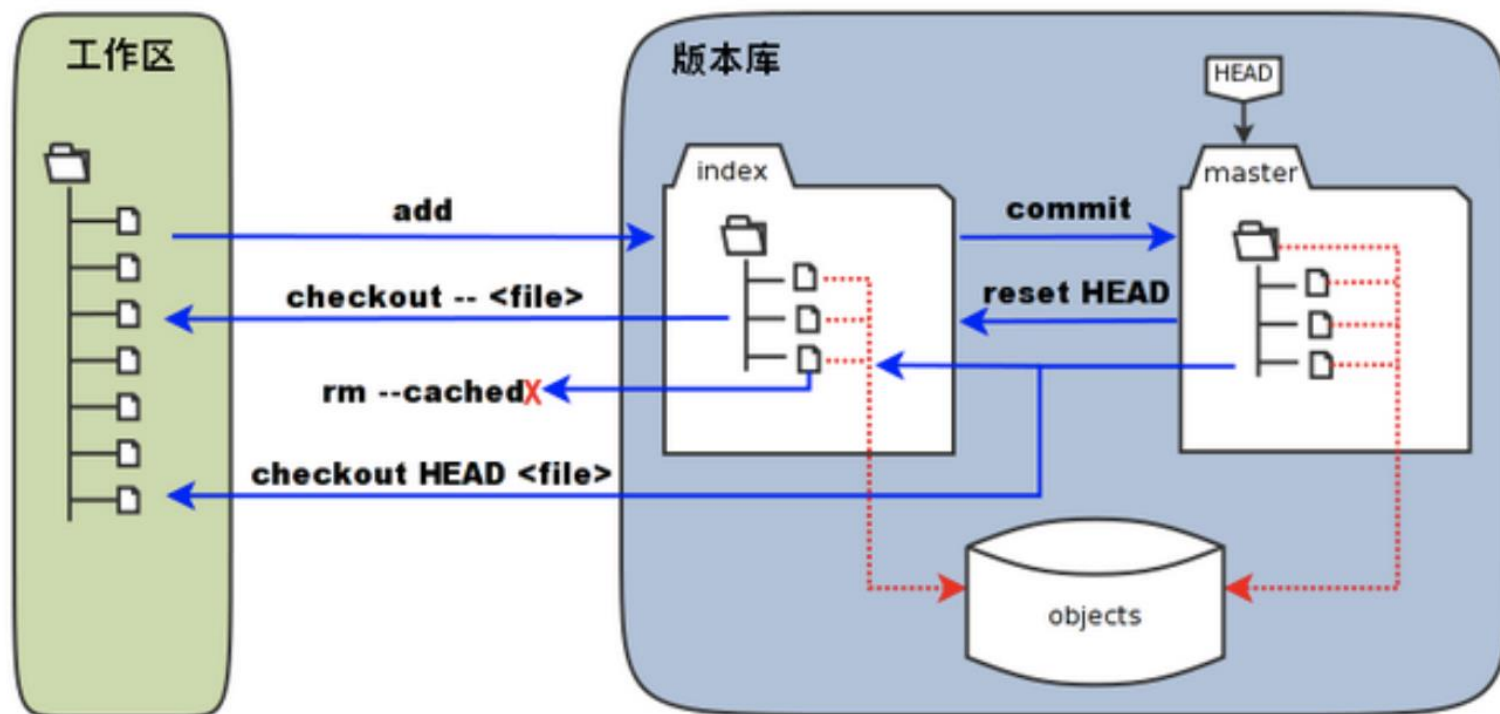


Git简单使用——原理介绍

Working Directory：用户空间，即用户修改代码存储的位置。

Index：暂存区域，引用的指针，会指向最新提交的快照。

HEAD：仓库，历史提交都在这里。



Github远程仓库



Github——准备工作

gitHub: 是一个面向开源及私有软件项目的托管平台, 因为只支持git 作为唯一的版本库格式进行托管, 故名gitHub。
全球最大的社交编程及代码托管网站。

2018年6月4日, 微软宣布, 通过75亿美元的股票交易收购代码托管平台GitHub。

2013年1月20日左右, GitHub在中国大陆被官方的防火长城封锁(封锁手段为域名污染和关键词过滤), 李开复等一些微博大号呼吁解禁GitHub, 微博转发近10万条。2013年1月23日github网站恢复正常访问。

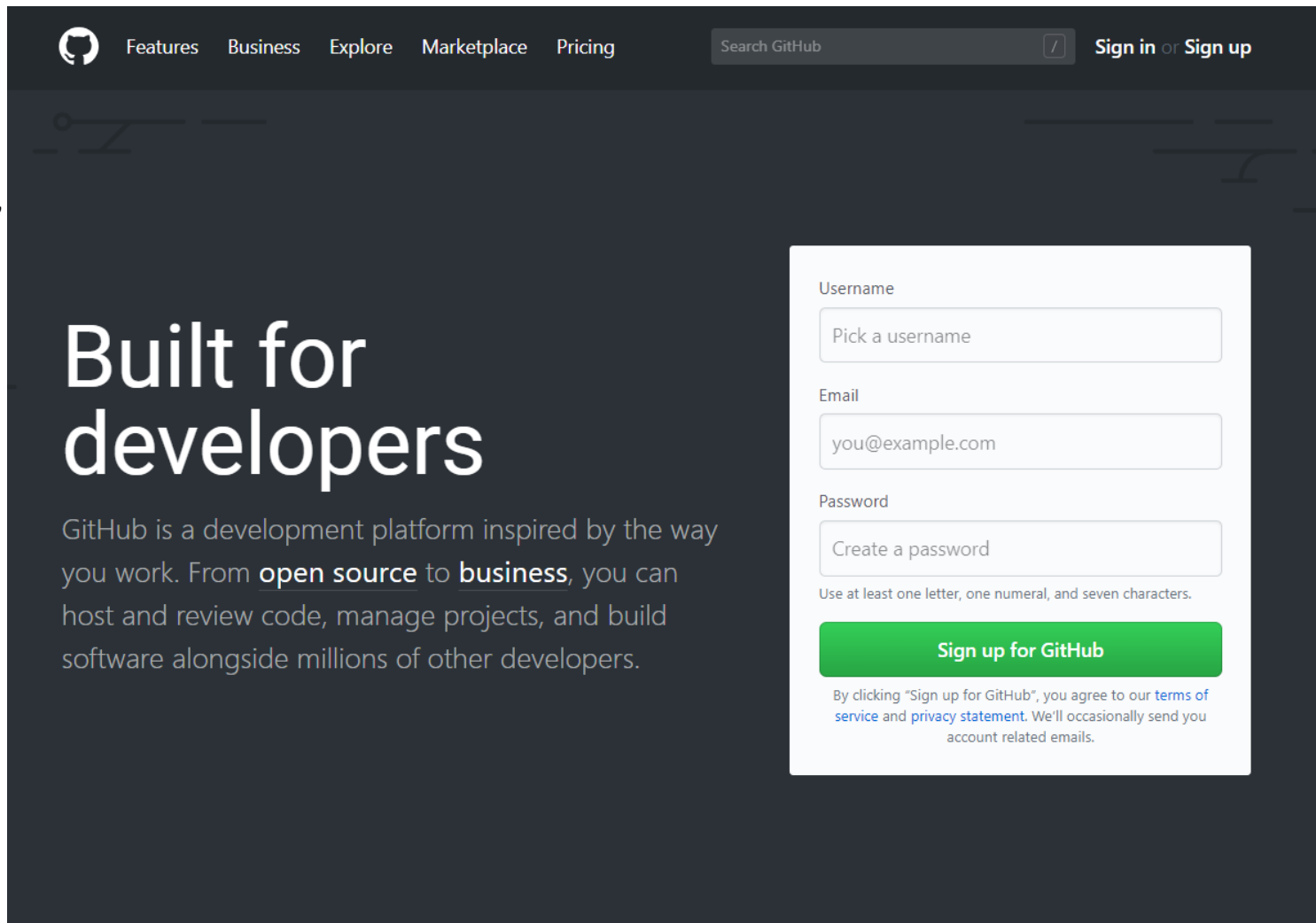
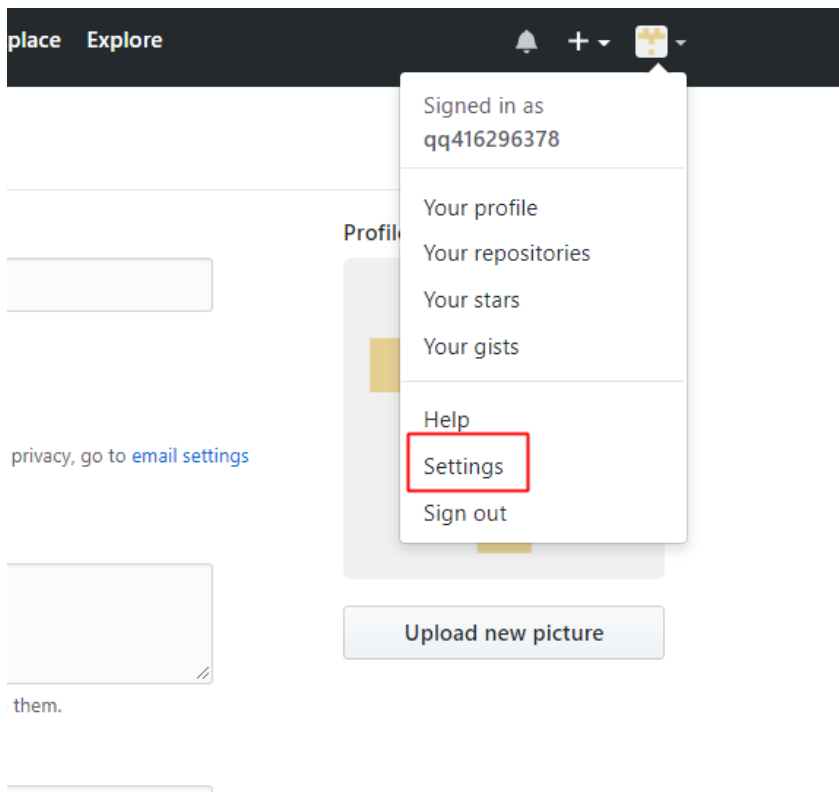
<https://www.github.com/>





Github——准备工作

1. 注册Github用户。
2. 生成本地SSH密钥
`ssh-keygen.exe -t rsa -C "416296378@qq.com"`
3. 打开设置添加密钥。





Github——准备工作

4. 复制密钥：~/.ssh/cat id_rsa.pub

```
qq416@Java_Nootbook MINGW64 ~/.ssh
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQD0qmV2jluJ1VVjRryTuScz80fBKv4DkBmawzjzv2s1
OViPm1Ygua+qgL75SMGUAY78H10ej+R6wFndIvNbRo/MuCCDZ+yRyx0+cUV1q7Jj2C6o0vNF1ZN1C81
w6K+PuQ+a0z8Cfs5jOn6fjJY10xbMbHTRRBEiSh4oyAniL41mE3aZYTTTrTbjs+yxm0mcBx0QFmfHMu19
D8I26TpHLypadr1TkfE8EVetndG48k4bSufk9e0dE42N/AULoGSyD16HtjlucCn8JrQjzLz8GDbqKH21
9I68Adaqp86b4gPZs6JIWmg5xX2A5fhy0cKJt++vf7WBhTpIoZC1zXNx1pM9 416296378@qq.com
```

5. 添加在Key的栏目中。

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Saved replies

Applications

Developer settings

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

notebook

Fingerprint: b9:60:12:25:8c:7e:2f:2c:cd:5f:b5:f1:6a:fd:fb:a0

Added on 28 Jul 2018

Last used within the last week — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).



Github——本地版本推送到Github

1. 创建一个版本库

2. 添加远程仓库

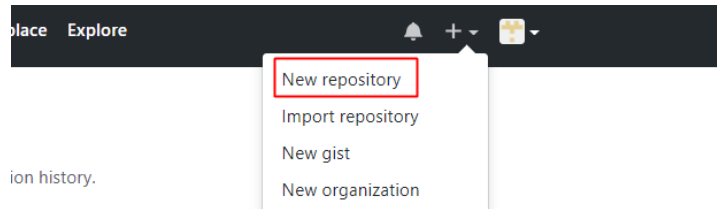
```
git remote add origin git@github.com:qq416296378/RPi_test.git
```

3. 将本地版本推送到Github

```
git push -u origin master
```

4. 拉取远端版本。

```
git clone git@github.com:qq416296378/RPi_test.git
```



How about [bug-free-potato](#).

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

qq416296378

Repository name

Great repository names are short and memorable. Need inspiration? How about [bug-free-potato](#).

Description (optional)

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

☐ Initialize this repository with a README

Quickly clone the repository to your computer. Skip this step if you're importing an existing repository.

Add a license: [None](#)

qq416296378 / RPi_test

Watch 0

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) `git@github.com:qq416296378/RPi_test.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# RPi_test" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:qq416296378/RPi_test.git
git push -u origin master
```

Jupyter notebook



Jupyter notebook——安装

Jupyter:

Jupyter Notebook（此前被称为 IPython notebook）是一个交互式笔记本。Jupyter Notebook 的本质是一个 Web 应用程序，便于创建和共享文学化程序文档，支持实时代码，数学方程，可视化和 markdown。用途包括：数据清理和转换，数值模拟，统计建模，机器学习等等。

安装:

1. 安装Python3 & PIP:

```
sudo apt-get install python3 python3-pip
```

2. 安装Jupyter:

```
sudo pip3 install jupyter
```

速度很慢，会安装失败可以尝试国内的源

```
sudo pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple jupyter
```

3. 设置登录密码:

```
jupyter notebook password
```





Jupyter notebook——启动

启动： `jupyter notebook -ip=0.0.0.0 -port=8888 -allow-root`

启动命令太长不好记，创建一个启动脚本：

```
sudo vi jupyter_setup.sh
```

```
#!/bin/bash
```

```
jupyter notebook -ip=0.0.0.0 -port=8888 -allow-root
```

修改脚本执行权限：

```
chmod 755 jupyter_setup.sh
```

执行脚本：(PS 在执行语句后面加“&”表示在后台执行)

```
./jupyter_setup.sh &
```

```
pi@raspberrypi:~ $ ./jupyter_setup.sh &
[1] 1843
pi@raspberrypi:~ $ [I 05:22:15.397 NotebookApp] Serving notebooks from local dir
ectory: /home/pi
[I 05:22:15.397 NotebookApp] The Jupyter Notebook is running at:
[I 05:22:15.398 NotebookApp] http://(raspberrypi or 127.0.0.1):8888/
[I 05:22:15.398 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[w 05:22:15.400 NotebookApp] No web browser found: could not locate runnable bro
wser.
```





Jupyter notebook——使用介绍

1. 访问：http://“树莓派IP”:8888
2. 输入Password
3. 我们打开了jupyter notebook的页面



Password:

Log in

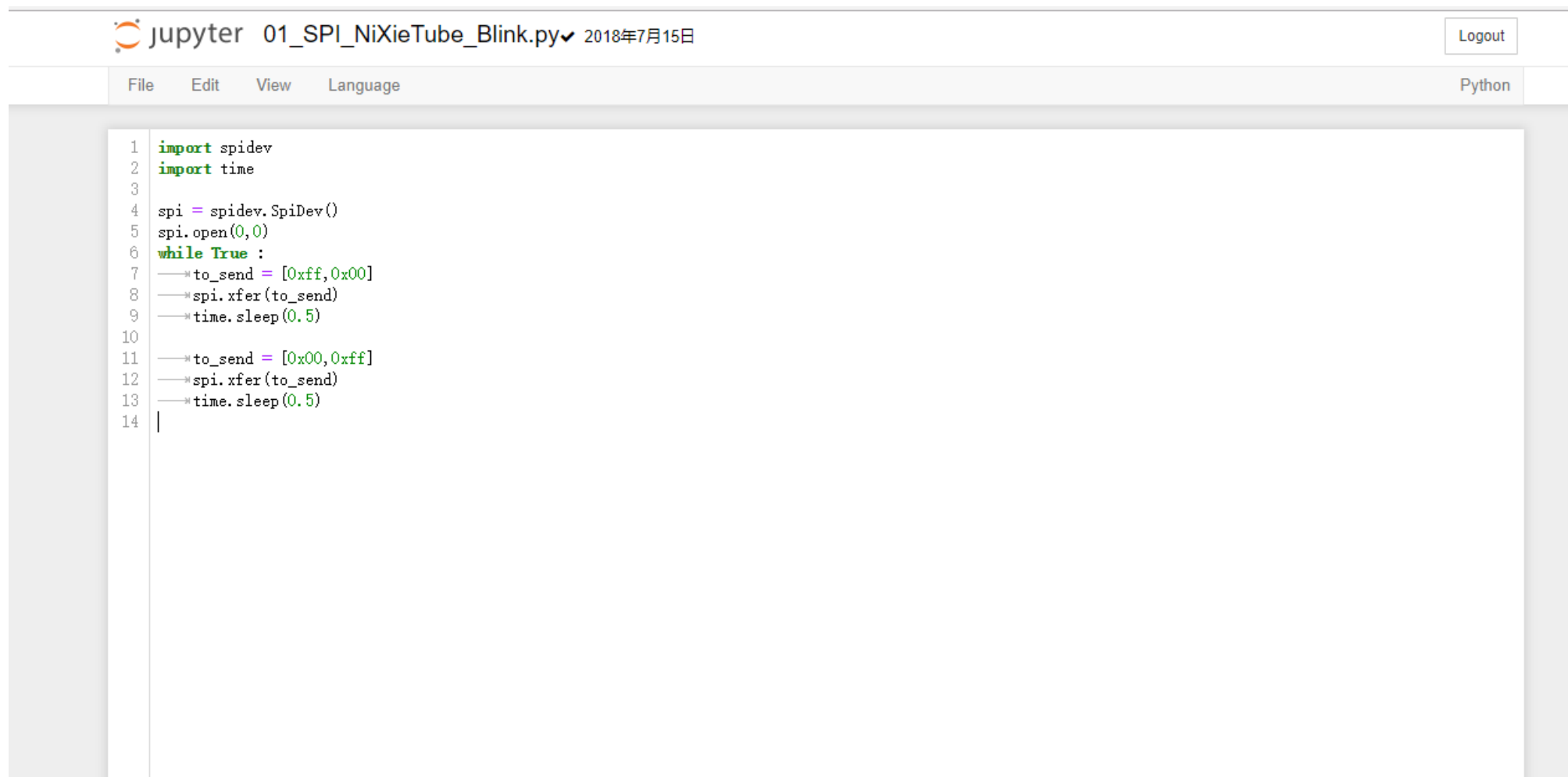
The screenshot shows the Jupyter Notebook web interface. At the top, there's a header with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a file browser. The browser displays a list of files and folders in the current directory. The list includes folders like Desktop, Documents, Downloads, Music, Pictures, Public, python_games, seeed-voicecard, Templates, test, and Videos. It also lists several Python files: 01_SPI_NiXieTube_Blink.py, 02_SPI_NiXieTube_Static.py, 03_SPI_NiXieTube_Dynamic.py, 04_SPI_NiXieTube_Dynamic_Def.py, and 05_SPI_NiXieTube_Dynamic_time.py. Each file entry shows its name, last modified time (e.g., '21 天前'), and file size (e.g., '195 B').

Name	Last Modified	File size
Desktop	21 天前	
Documents	15 天前	
Downloads	21 天前	
Music	21 天前	
Pictures	21 天前	
Public	21 天前	
python_games	3 个月前	
seeed-voicecard	3 天前	
Templates	21 天前	
test	21 天前	
Videos	21 天前	
01_SPI_NiXieTube_Blink.py	13 天前	195 B
02_SPI_NiXieTube_Static.py	13 天前	219 B
03_SPI_NiXieTube_Dynamic.py	13 天前	583 B
04_SPI_NiXieTube_Dynamic_Def.py	13 天前	716 B
05_SPI_NiXieTube_Dynamic_time.py	13 天前	877 B



Jupyter notebook——使用介绍

编辑文件：



```
1 import spidev
2 import time
3
4 spi = spidev.SpiDev()
5 spi.open(0,0)
6 while True :
7     to_send = [0xff,0x00]
8     spi.xfer(to_send)
9     time.sleep(0.5)
10
11     to_send = [0x00,0xff]
12     spi.xfer(to_send)
13     time.sleep(0.5)
14 |
```



Jupyter notebook——使用介绍

Python3的使用和笔记

The screenshot shows a Jupyter Notebook interface. At the top, the title bar says "jupyter test" with a "Last Checkpoint: 12 小时前 (autosaved)" status. On the right, there is a "Logout" button and a Python version indicator "Python 3". Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area contains a code cell with the following code:

```
In [1]: import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
GPIO.output(17, GPIO.HIGH)
time.sleep(2)
GPIO.output(17, GPIO.LOW)
time.sleep(2)
GPIO.output(17, GPIO.HIGH)
time.sleep(2)
GPIO.output(17, GPIO.LOW)
time.sleep(2)
print("can use LED")
```

The output of the code cell is "can use LED". Below the code cell, there is a text box with the following text:

RPi.GPIO为上层操作底层硬件提供接口

GPIO.setwarnings(False)

如果RPi.GPIO检测到一个引脚已经被设置成了非默认值，那么你将看到一个警告信息。你可以通过下列代码禁用警告：

```
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
```

At the bottom, there is an input field for the next code cell, labeled "In []:".



Jupyter notebook——使用介绍

映射terminal

jupyter

Logout

```
pi@raspberrypi:~ $ ls
01_SPI_NiXieTube_Blink.py      123.wav      Documents      LED.py      python_games      Test
02_SPI_NiXieTube_Static.py     APA102LED.py Downloads      Music      seeed-voicecard  Untitled.ipynb
03_SPI_NiXieTube_Dynamic.py    Button_LED.py I2C-OLED.py    Pictures    shu.py           Videos
04_SPI_NiXieTube_Dynamic_Def.py Button.py      jupyter_setup.sh Public      Templates
05_SPI_NiXieTube_Dynamic_time.py Desktop       led.py         __pycache__ test

pi@raspberrypi:~ $ ping baidu.com
PING baidu.com (220.181.57.216) 56(84) bytes of data.
64 bytes from 220.181.57.216 (220.181.57.216): icmp_seq=1 ttl=55 time=30.4 ms
64 bytes from 220.181.57.216 (220.181.57.216): icmp_seq=2 ttl=55 time=33.7 ms

--- baidu.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 30.495/32.144/33.794/1.659 ms
^C
pi@raspberrypi:~ $ curl google.co,
^C
pi@raspberrypi:~ $ curl google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
pi@raspberrypi:~ $
```




Jupyter notebook——使用介绍

Markdown功能

```
In [1]: import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17,GPIO.OUT)
GPIO.output(17,GPIO.HIGH)
time.sleep(2)
GPIO.output(17,GPIO.LOW)
time.sleep(2)
GPIO.output(17,GPIO.HIGH)
time.sleep(2)
GPIO.output(17,GPIO.LOW)
time.sleep(2)
print("can use LED")

can use LED
```

RPi.GPIO为上层操作底层硬件提供接口
GPIO.setwarnings(False)

如果RPi.GPIO检测到一个引脚已经被设置成了非默认值，那么你将看到一个警告信息。你可以通过下列代码禁用警告：

```
python3
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
```

In []:

```
In [1]: import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17,GPIO.OUT)
GPIO.output(17,GPIO.HIGH)
time.sleep(2)
GPIO.output(17,GPIO.LOW)
time.sleep(2)
GPIO.output(17,GPIO.HIGH)
time.sleep(2)
GPIO.output(17,GPIO.LOW)
time.sleep(2)
print("can use LED")

can use LED
```

RPi.GPIO为上层操作底层硬件提供接口

GPIO.setwarnings(False)

如果RPi.GPIO检测到一个引脚已经被设置成了非默认值，那么你将看到一个警告信息。你可以通过下列代码禁用警告：

```
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
```

In []:

Markdown



Markdown——介绍

Markdown是一种轻量级标记语言，它允许人们“使用易读易写的纯文本格式编写文档，然后转换成有效的XHTML（或者HTML）文档”。这种语言吸收了很多在电子邮件中已有的纯文本标记的特性。

推荐使用Markdown语言的软件：
Typora

官方网站：
<https://typora.io/>



```
1  Linux kernel
2  =====
3
4  There are several guides for kernel developers and users. These guides can
5  be rendered in a number of formats, like HTML and PDF. Please read
6  Documentation/admin-guide/README.rst first.
7
8  In order to build the documentation, use ``make htmldocs`` or
9  ``make pdfdocs``. The formatted documentation can also be read online at:
10
11     https://www.kernel.org/doc/html/latest/
12
13  There are various text files in the Documentation/ subdirectory,
14  several of them using the Restructured Text markup notation.
15  See Documentation/00-INDEX for a list of what is contained in each file.
16
17  Please read the Documentation/process/changes.rst file, as it contains the
18  requirements for building and running the kernel, and information about
19  the problems which may result by upgrading your kernel.
```



Markdown——语法

标题分级：

输入：

```
1 # 一级标题
  ## 二级标题
  ### 三级标题
  #### 四级标题
  ##### 五级标题
  ##### 六级标题
```

效果：

一级标题

二级标题

三级标题

四级标题

五级标题

六级标题

标题大小：

输入：

```
标题大
===
标题小
---
```

效果：

标题大

标题小

强调：

输入：

```
*强调斜体*
**强调粗体**
删除线
`ctrl+a`
```

效果：

强调斜体

强调粗体

~~删除线~~

ctrl+a



Markdown——语法

分割线：

输入：

分割线

效果：

分割线

有序排列：

输入：

- 列表1
- 列表2
- 列表3 ↓

效果：

- 列表1
- 列表2
- 列表3

无序排列：

输入：

1. 123
2. 345
3. 456
4. 789

效果：

1. 123
2. 345
3. 456
4. 789



Markdown——语法

插入图片：

输入：

![GitHub Octocat]
(http://github.global.ssl.fastly.net/images/modules/logos_page/Octocat.png)

效果：



插入连接：

输入：

[[百度](http://www.baidu.com)](<http://www.baidu.com>)

效果：

[百度](http://www.baidu.com)

引用：

输入：

> 一盏灯，一片昏黄；一简书，一杯淡茶。守着那一份淡定，品读属于自己的寂寞。保持淡定，才能欣赏到最美丽的风景！保持淡定，人生从此不再寂寞。

效果：

一盏灯，一片昏黄；一简书，一杯淡茶。守着那一份淡定，品读属于自己的寂寞。保持淡定，才能欣赏到最美丽的风景！保持淡定，人生从此不再寂寞。



Markdown——语法

表格:

输入:

```
| Tables          | Are          | Cool  |
| -----|:-----:| -----|
| col 3 is       | right-aligned| $1600 |
| col 2 is       | centered    | $12   |
| zebra stripes  | are neat    | $1    |
```

效果:

Tables	Are	Cool
col 3 is	right-aligned	\$1600
col 2 is	centered	\$12
zebra stripes	are neat	\$1

代码区块:

输入:

```
```c
void main()
{
 printf("Hello, Markdown.");
}
```
```

效果:

```
void main()
{
    printf("Hello, Markdown.");
}
```

转义符:

输入:

>\\ 反斜杠 ↓
\\ 反引号 ↓
* 星号 ↓
_ 下划线 ↓
\\{\\} 大括号 ↓
\\[\\] 中括号 ↓

效果:

\\ 反斜杠 ↓
` 反引号 ↓
* 星号 ↓
_ 下划线 ↓
{ } 大括号 ↓
[] 中括号 ↓

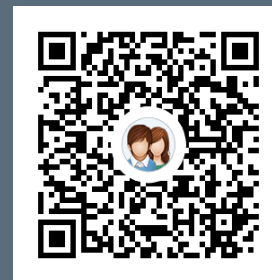


感谢您的观看

www.moore8.com



微信公众平台:
moore_8



QQ群: 327350729