**学习project第二期 学习报告**

1. 任务

1.熟悉apt-get的换源方法，并将apt-get的软件源换为国内源，并用apt-get安装vim和git。

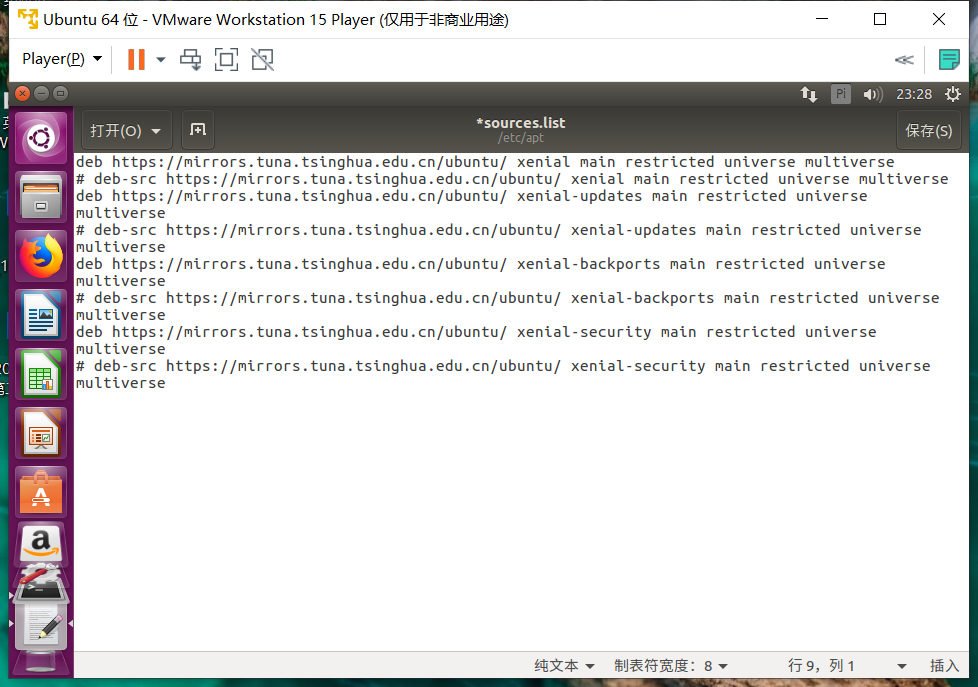
* **先进入开源镜像站获取相应链接，复制地址（清华大学镜像站：https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/）**



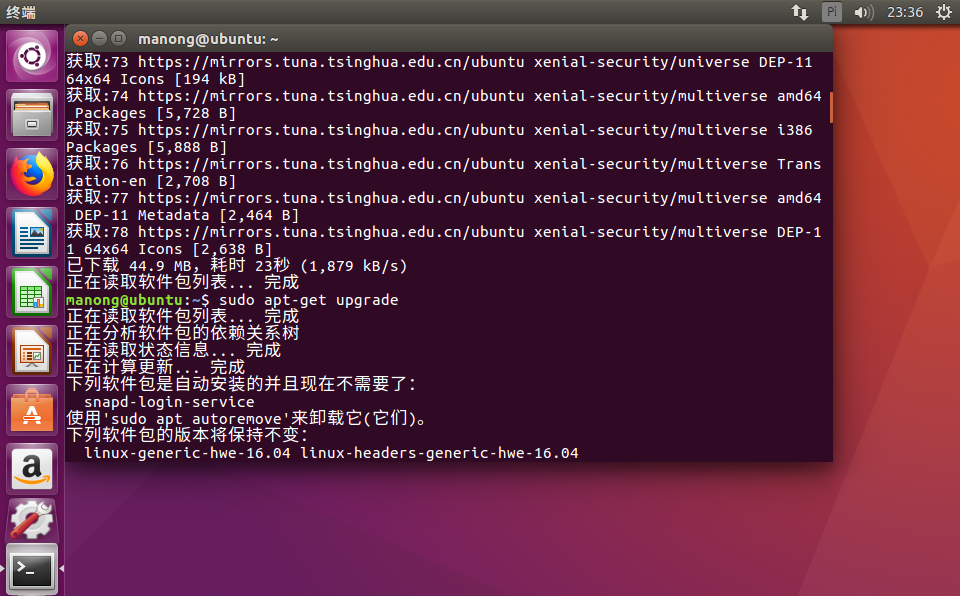
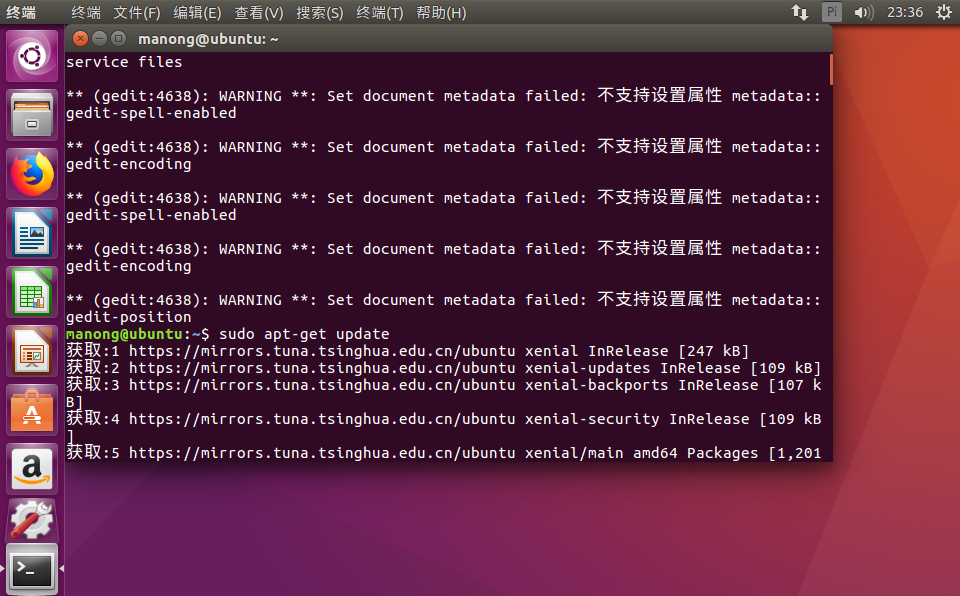
* **备份一下自己系统的源文件（命令：sudo cp /etc/apt/sources.list /etc/apt/sources.bak1）**

**#第一个参数是拷贝的文件路径和文件名称,第二个是拷贝到(粘贴)的文件路径和文件名**

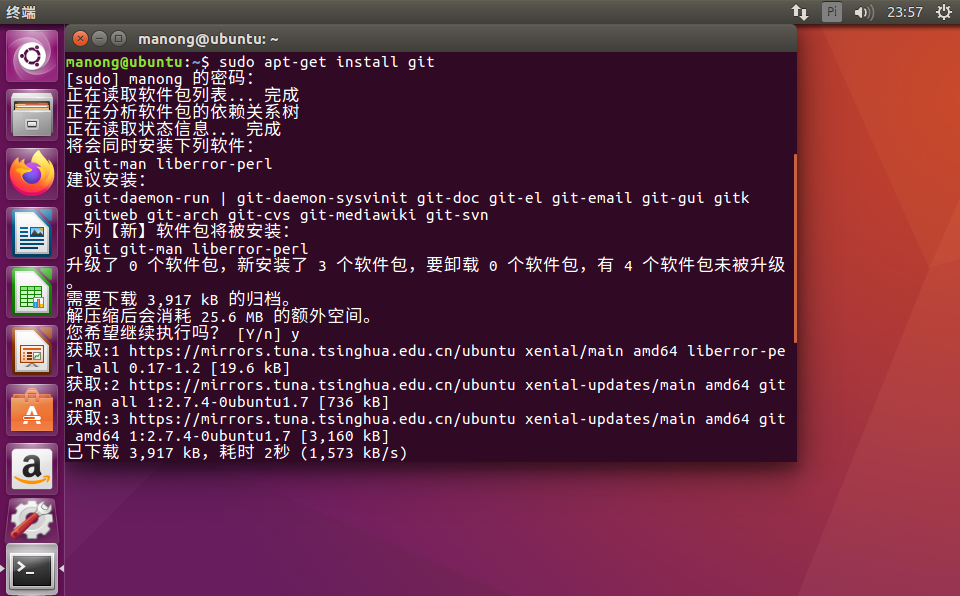
* **替换系统本身的镜像源地址（命令：sudo gedit /etc/apt/source.list）**

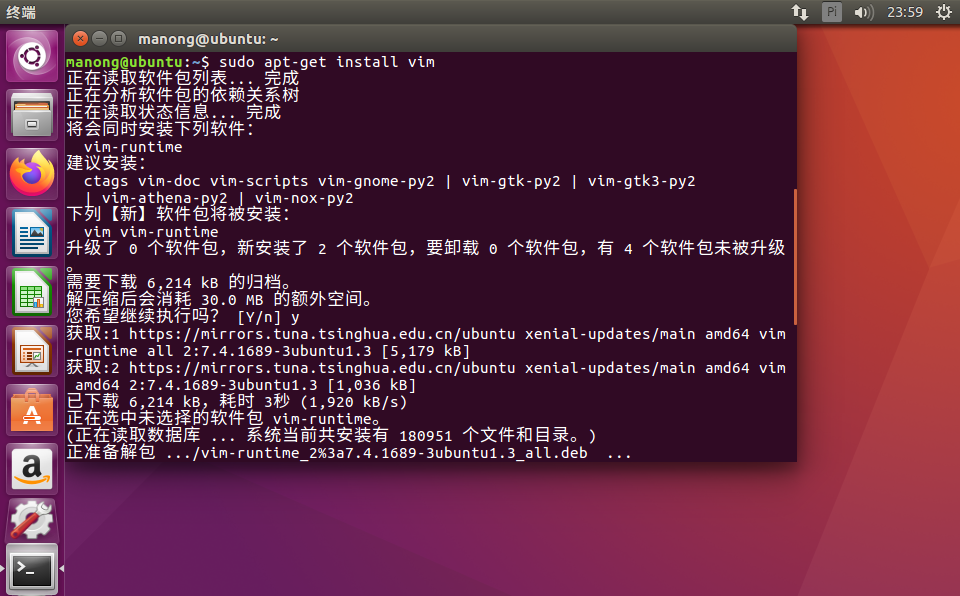


* **更新软件源列表（命令1:sudo apt-get update#这个命令，会访问源列表里的每个网址，并读取软件列表，然后保存在本地电脑。然后输出命令2：sudo apt-get upgrade #这个命令，会把本地已安装的软件，与刚下载的软件列表里对应软件进行对比，如果发现已安装的软件版本太低，就会提示你更新）**



* **用apt-get安装git和vim（命令：sudo apt-get git/vim）**

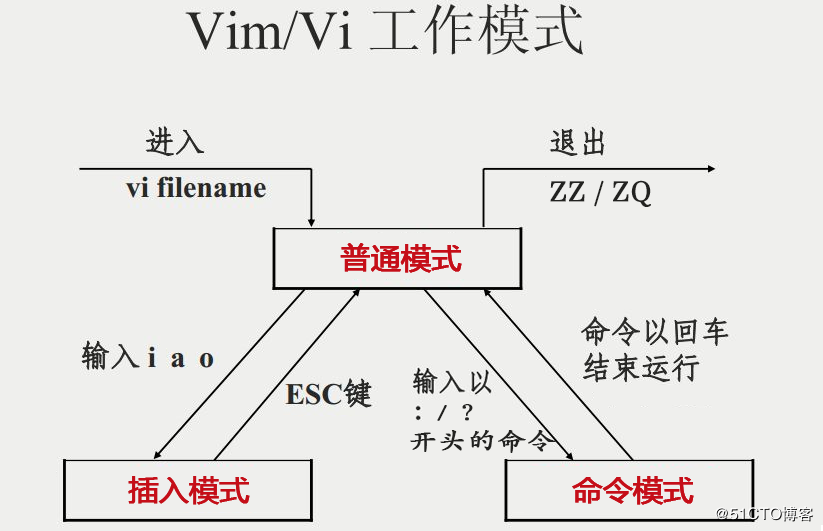




**2.熟悉vim的用法（这东西真的有点反人类...）**

* **三种基本模式**

基本上vim可以分为三种状态，分别是普通模式、插入模式和命令模式，各模式的功能区分如下：  
**1、普通模式**  
控制屏幕光标的移动，字符、字或行的删除，移动复制某区段。  
**2、插入模式**  
只有在插入下，才可以做文字输入、删除，按「ESC」键可回到普通模式。  
可以按i、I、a、A、o、O、r、R即可。其中i和I表示插入（Insert），r和R表示替换（replace）。  
**3、命令模式**  
将文件保存或退出vim，也可以设置编辑环境，如寻找字符串、列出行号等……  
可以按“:”、“/”、“?”。其中“:”表示输入vim命令，“/”表示后面输入要向下搜索的字符串，“?”表示后面输入要向上搜索字符串。



* **VIM的配置**

1. **颜色设置**

|  |  |
| --- | --- |
| **syntax on** | **开启代码高亮** |
| **syntax off** | **关闭代码高亮** |
| **syntax enable** | **开启代码高亮** |

1. **搜索设置**

|  |  |
| --- | --- |
| **set hlsearch** | **开启搜索高亮** |
| **set nohlsearch** | **关闭搜索高亮** |
| **set incsearch** | **输入搜索字符串的同时进行搜索** |
| **set ignorecase** | **搜索时忽略大小写** |

1. **用户界面**

|  |  |
| --- | --- |
| **set ruler** | **开启光标位置提示** |
| **set number** | **显示行号** |
| **set nonumber** | **不显示行号** |
| **set cmdheight=1** | **命令部分高度为1** |

1. **编辑辅助配置**

|  |  |
| --- | --- |
| **set autoindent** | **自动缩进** |
| **set noautoindent** | **不自动缩进** |
| **set smartindent** | **智能缩进** |
| **set autoread** | **当文件在外部被改变时，vim自动更新载入** |
| **set showmatch** | **显示匹配的括号** |

* **常用的状态切换按键**

1. **普通模式 → 插入模式**

|  |  |
| --- | --- |
| **小写字母i** | **在光标位置插入** |
| **大写字母I** | **在光标所在行的第一个非空格处插入** |
| **小写字母a** | **在光标的下一个位置插入** |
| **大写字母A** | **在光标所在行的最后一个字符处插入** |
| **小写字母o** | **在光标所在行的下一行处插入新行** |
| **大写字母O** | **在光标所在行的上一行处插入新行** |
| **小写字母r** | **替换光标所在处的字符一次** |
| **大写字母R** | **持续替换光标所在处的字符，直到按下「ESC」键** |

1. **普通模式 → 命令模式**

|  |  |
| --- | --- |
| **:w** | **保存文件** |
| **:w!** | **强制保存文件（前提是用户有修改文件访问权限的权限）** |
| **:q** | **退出缓冲区** |
| **:q!** | **强制退出缓冲区而不保存** |
| **:wq** | **保存文件并退出缓冲区** |
| **:wq!** | **强制保存文件并退出缓冲区（前提是用户有修改文件访问权限的权限）** |
| **:w [filename]** | **另存为名为filename文件** |
| **:r [filename]** | **将另外一个文件读入到正在编辑的文件** |
| **:n1,n2 w [filename]** | **将n1行到n2行的数据另存为名为filename文件** |
| **:x** | **保存并退出** |
| **ZZ** | **保存并退出** |
| **ZQ** | **不保存退出** |

1. **插入模式 → 视图模式**

按下「ESC」键

1. **命令模式 → 视图模式**

按下「ESC」键

* **常用的光标移动按键**

1. **光标字符操作**

|  |  |
| --- | --- |
| **h** | **向左箭头键(←) 光标向左移动一个字符** |
| **I** | **向右箭头键(→) 光标向右移动一个字符** |
| **j** | **向下箭头键(↓) 光标向下移动一个字符** |
| **k** | **向上箭头键(↑) 光标向上移动一个字符** |
| **$** | **光标移动到行尾（End），注意要按「Shift」键** |
| **0** | **光标移动到行首（Home）** |
| **^** | **光标移动到行首第一个非空白字符（Home），注意要按「Shift」键** |

1. **光标词操作**

|  |  |
| --- | --- |
| **w** | **光标移动到后一词的词首** |
| **W** | **光标移动到后一词的词首且忽略标点符号** |
| **e** | **光标移动到后一词的词尾** |
| **E** | **光标移动到后一词的词尾且忽略标点符号** |
| **b** | **光标移动到前一词的词首** |
| **B** | **光标移动到前一词的词首且忽略标点符号** |

1. **光标行操作**

|  |  |
| --- | --- |
| **G** | **光标移动到文件的最后一行的第一个非空白字符** |
| **nG/ngg** | **光标移动到文件的第n行，相当于”:n”** |
| **gg** | **光标移动到文件的第1行的第一个非空白字符** |

1. **光标页操作**

|  |  |
| --- | --- |
| **Ctrl + f** | **向下翻页（Pagedown）** |
| **Ctrl + b** | **向上翻页（Pageup）** |
| **Ctrl + d** | **向下翻半页** |
| **Ctrl + u** | **向上翻半页** |
| **「H」** | **光标移动到目前显示页的第一行** |
| **「M」** | **光标移动到目前显示页的中间行** |
| **「L」** | **光标移动到目前显示页的最后一行** |

1. **光标自由操作**

**Ctrl + o  回到光标的上一位置**

* **常用编辑操作按键**

1. **删除操作（delete）**

|  |  |
| --- | --- |
| **dd** | **删除光标所在行** |
| **ndd** | **删除从光标所在行开始，向下的n行** |
| **d1G** | **删除从光标所在行开始，到第一行的所有行** |
| **dG** | **删除从光标所在行开始，到最后一行的所有行** |
| **d$** | **删除从光标所在位置，到行尾的所有字符** |
| **d0** | **删除从光标所在位置，到行首的所有字符** |
| **x** | **每按一次，删除光标所在位置的“后面”一个字符** |
| **nx** | **删除光标所在位置的“后面”n个字符** |
| **X** | **大写的X，每按一次，删除光标所在位置的“前面”一个字符。** |
| **nX** | **删除光标所在位置的“前面”n个字符** |

1. **复制操作（yank）**

|  |  |
| --- | --- |
| **yy** | **复制光标所在行** |
| **nyy** | **复制从光标所在行开始，向下的n行** |
| **y1G** | **复制从光标所在行开始，到第一行的所有行** |
| **yG** | **复制从光标所在行开始，到最后一行的所有行** |
| **y$** | **复制从光标所在位置，到行尾的所有字符** |
| **y0** | **复制从光标所在位置，到行首的所有字符** |
| **J** | **将光标所在行与下一行合并成一行** |
| **yw** | **复制当前光标所在的字母到该单词的最后一个字母** |
| **yaw** | **复制当前光标所在的单词** |

1. **粘贴操作（paste）**

**p  小写字母p，粘贴剪贴板中的数据，从光标所在行的下一行开始**

**P  大写字母P，粘贴剪贴板中的数据，从光标所在行的上一行开始**

1. **撤销与重做操作（undo）**

**u   撤销上一个操作**

1. **重复操作**

**.   重复上一操作**

1. **替换操作（replace）**

|  |  |
| --- | --- |
| **r** | **替换光标所在处字符** |
| **R** | **进入替换模式，直至按ESC退出** |
| **cc** | **替换光标所在行** |
| **cw** | **替换光标所在的英文单词** |
| **~** | **转换大小写** |
| **:n1,n2s/old/new/g** | **从n1到n2行将old替换成new，g表示全部替换** |
| **:1,$s/old/new/g** | **将整个文件的old替换成new** |
| **:1,$s/old/new/gc** | **c表示确认** |

**new (y/n/a/q/l/^E/^Y)? 表示如下：  
    y：    替换当前内容  
    n：   不替换  
    a：    替换所有  
    q：    退出  
    l：   替换当前内容，并退出  
    Ctrl + e：向后查找  
    Ctrl + y：向前查找**

1. **排版操作**

|  |  |
| --- | --- |
| **:le** | **光标所在行左对齐** |
| **:ri** | **光标所在行右对齐** |
| **:ce** | **光标所在行居中** |

1. **查看历史**

**:history  显示最近在vim里面输入的历史**

* **常用的搜索快捷键**

1. **搜索字符串**

**/string    向下搜索字符串“string”  
?string    向上搜索字符串“string”**

**2、重复上次搜索**

n   根据上次搜索条件找到下一个匹配的字符串  
N  根据上次搜索条件找到上一个匹配的字符串

* **常用多缓冲区操作按键**

1. **单一缓冲区打开多个文件**

|  |  |
| --- | --- |
| **:files** | **显示目前打开了哪些文件** |
| **:n** | **切换到下一个缓冲区内容** |
| **:N** | **切换到上一个缓冲区内容** |
| **:2n** | **切换到下下个缓冲区内容** |
| **:bn** | **下一个缓冲区内容（buffer next）** |
| **:bp** | **上一个缓冲区内容（buffer previous）** |

1. **多个缓冲区打开多个文件**

|  |  |
| --- | --- |
| **:sp [filename]** | **打开一个新的缓冲区，如果有filename则内容为该文件，否则为当前文件** |
| **Ctrl+w n** | **新建一个缓冲区** |
| **Ctrl+w q** | **退出光标所在的缓冲区** |
| **Ctrl+w j** | **光标移动到下一缓冲区** |
| **Ctrl+w k** | **光标移动到上一缓冲区** |
| **Ctrl+w l** | **光标移动到右一缓冲区** |
| **Ctrl+w h** | **光标移动到左一缓冲区** |
| **Ctrl+w v** | **左右切割窗口新建缓冲区** |
| **Ctrl+w s** | **上下切割窗口新建缓冲区** |
| **Ctrl+w o** | **使光标所在缓冲区最大化，其他缓冲区隐藏** |

* **关于加密**

1. **加密文本文件**  
   当你用vim打开或新建一个文件时，如果加上-x参数，即：

**# vim -x test.txt**  --打开你要加密的文本文件  
那么vim会提示你：  
**Enter entryption key:**  
当你输入了密码后，vim会提示你重复输入以确认：  
**Enter same key again:**  
当你在设置密码的时候，是以\*来显示的，非明文。打开文件之后，:wq保存退出，密码才生效。  
   
当下次有人（包括你自己）打开该文件的时候，vim就会提示：  
**Enter encryption key:**  
这时你只要输入你之前所设置的密码就可以了。如果你输入的密码是错误的，vim并不会提示你密码错误，而是显示一堆无意义的垃圾代码。

1. **解除加密与重设密码**

**# vim test.txt**   --打开加密文件之后，输入下面内容  
**:set key=**

这样，你的密码就被设置为空了，今后进入就不再需要输入密码了。而如果你向重设密码的话，你可以输入：:X

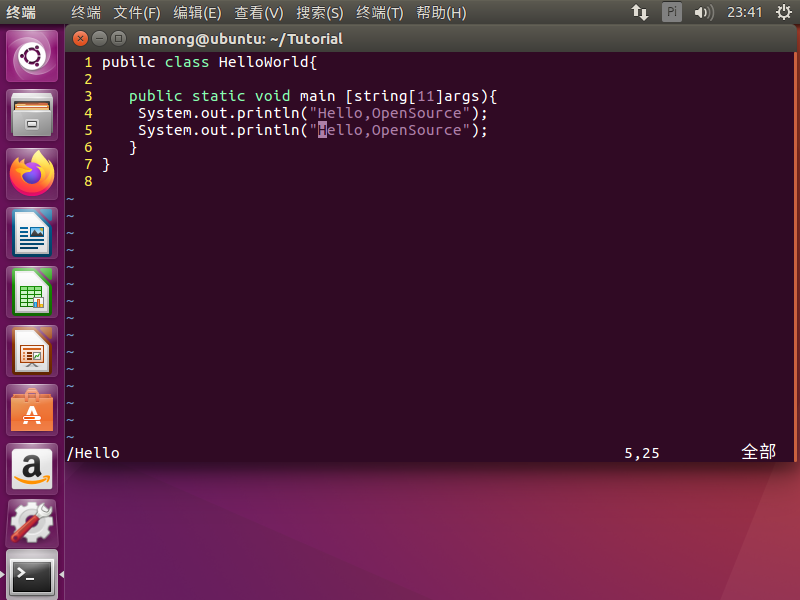
注意是个大写字母X。这时vim会提示你，只要反复输入两次新的密码就可以了：

Enter entryption key: \*\*\*\*\*  
Enter sanme key again: \*\*\*\*\*

如果是清除密码，连续敲两次键盘就可以了

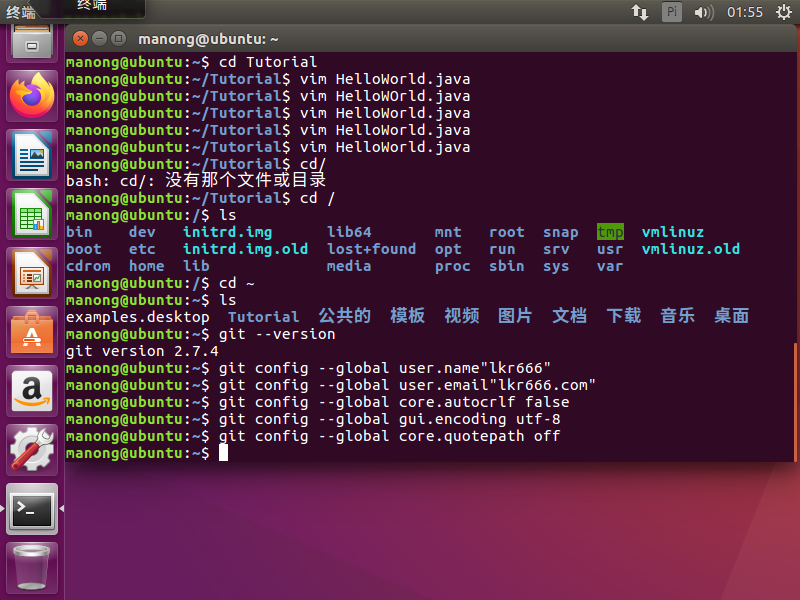
**注意：**  
加密、解除密码、重设密码之后，一定要保存之后才能生效。  
如果密码输入错误，vim不会拒绝访问者编辑文件。在这种显示乱码的情况下编辑文件并保存，整个文件就会变得混乱不堪，无法辨识。今后即使使用正确的密码打开文件，也看不到正确的文档了。

**（我写mkdir Tutorial创建一个名为Tutorial的新目录,接着cd Tutorial打开这个目录,vim HelloWorld.java 创建一个Vim文件然后写出了这个并保存）**



**3.熟悉git的用法**

* + - * **Git的配置：**
* (1) 配置用户名（提交时会引用）  
  git config --global user.name “lkr666”  
  #请把lkr替换成自己的用户名
* (2) 配置邮箱（提交时会引用）  
  git config --global user.email "1418165438@qq.com"
* (3) 其他配置  
  git config --global merge.tool “kdiff3”  
  #要是没安装KDiff3就不用设这一行
* git config --global core.autocrlf false  
  #让Git不要管windows/Unix换行符转换的事
* (4) 编码配置  
  git config --global gui.encoding utf-8  
  #避免git gui中的中文乱码
* git config --global core.quotepath off  
  #避免git status显示的中文文件名乱码
* Windows上还需要配置：  
  git config --global core.ignorecase false



* **创建shh（这个是真的看不懂）**

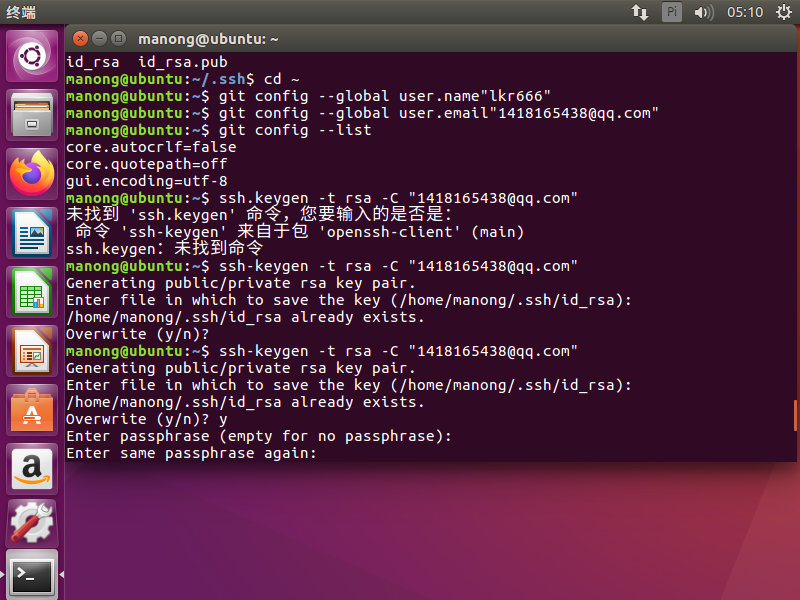
1. 安装完git查看版本 git –version
2. **配置用户名邮箱**

**git config --global user.name "lkr666"**

**git config --global user.email** [**1418165438@qq.com**](mailto:1418165438@qq.com)**（邮箱要写有效邮箱）**

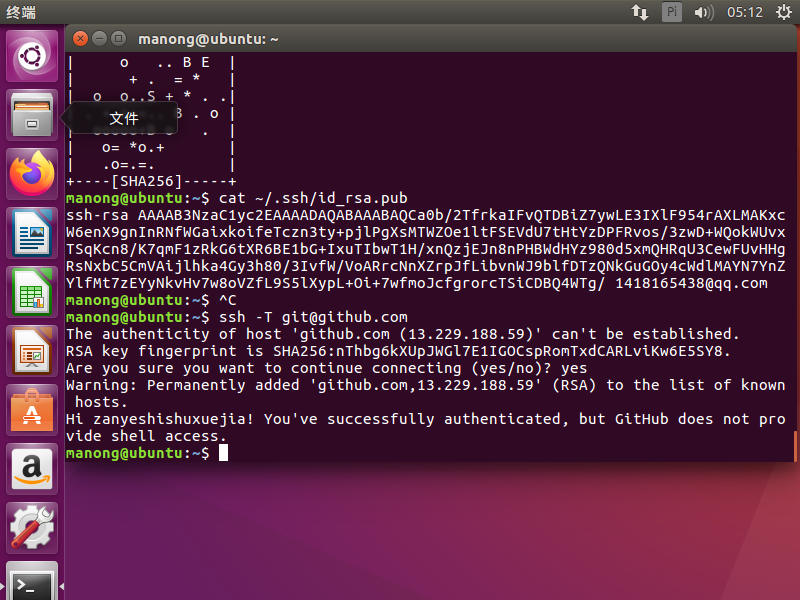
**3、**创建SSH key

$ ssh-keygen -t rsa -C "1418165438@qq.com"



**4、**然后用cat查看id\_rsa.pub文件内的内容，粘帖到github帐号管理的添加SSH key界面中。

cat ~/.ssh/id\_rsa.pub

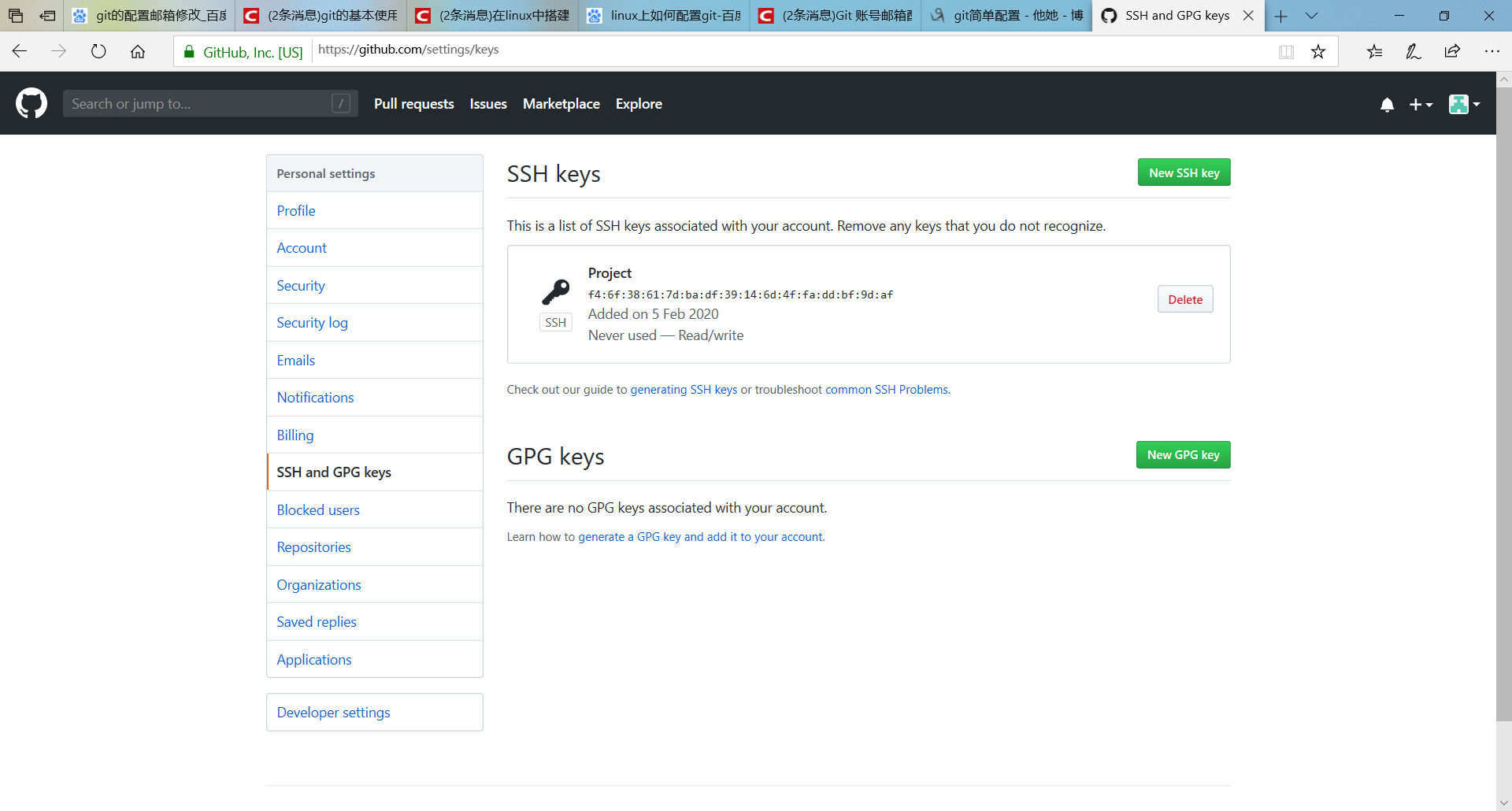


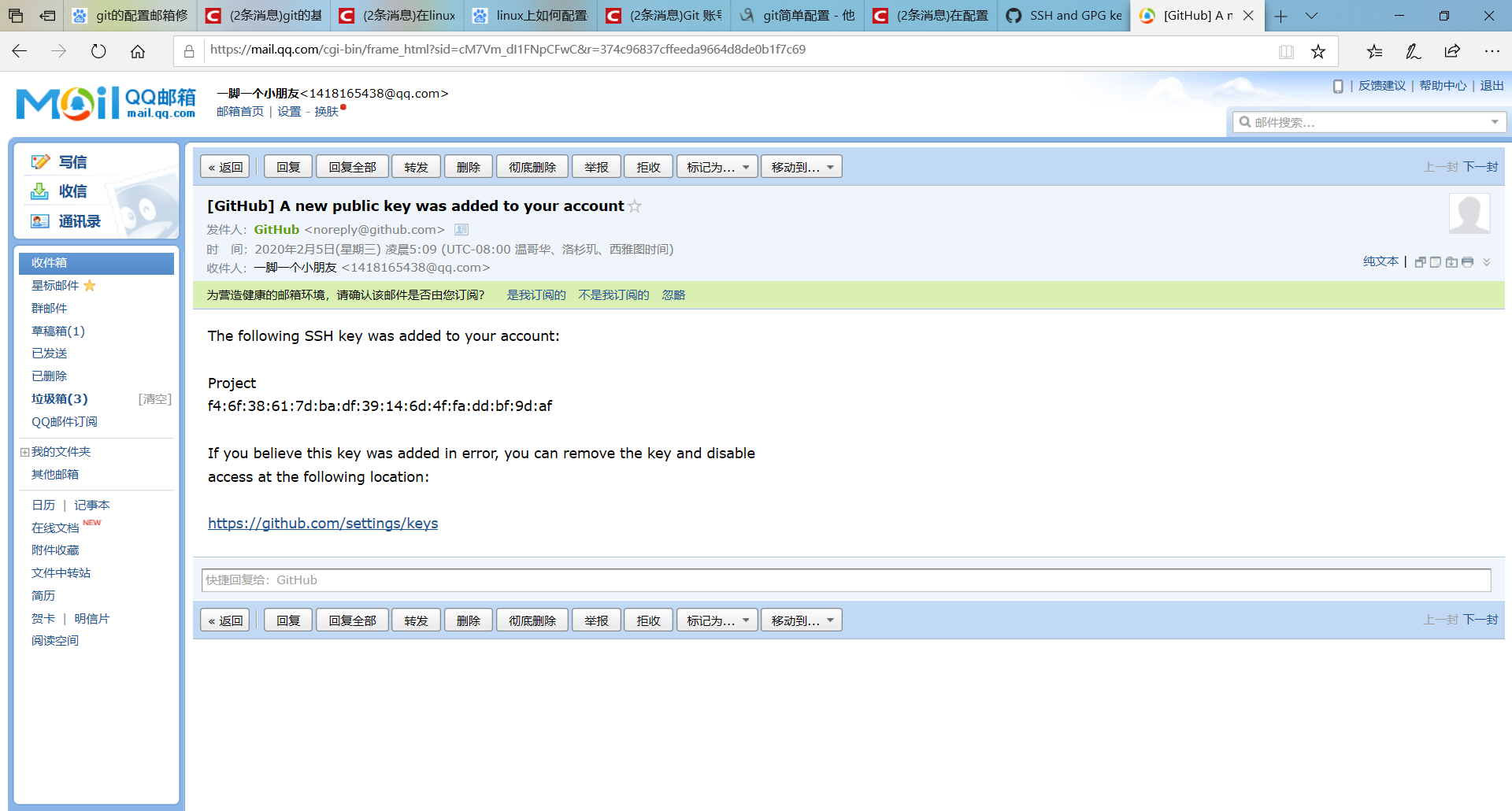
**5、**添加到GitHub

登录GitHub，点击如下图倒三角，然后点击“Settings”

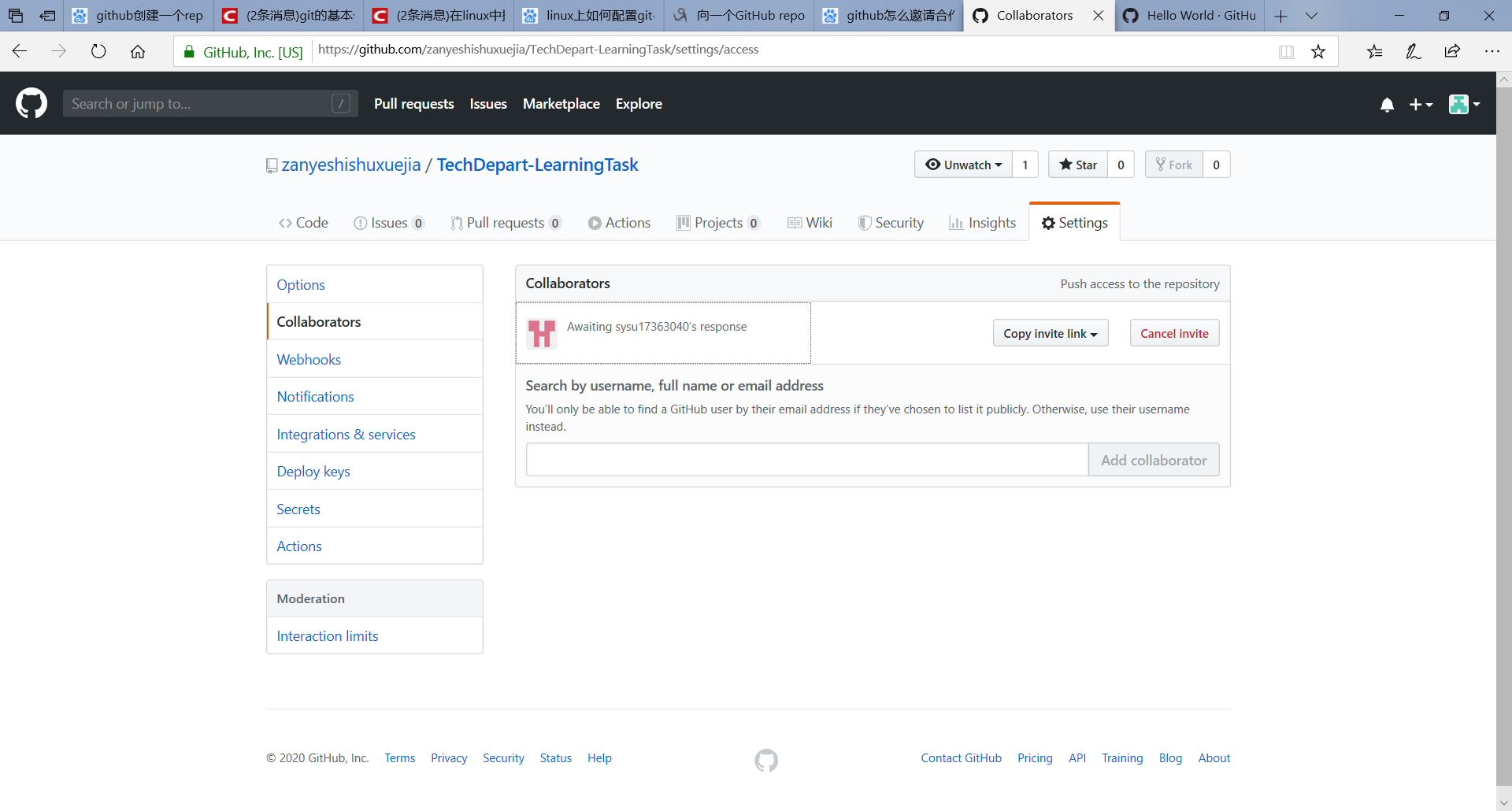
1. 将步骤2中cat ~/.ssh/id\_rsa.pub内容粘贴在下面，点击Add keys

添加成功之后，你的邮箱会收到一封邮件。





* **创建一个github账号，并创建一个repo，命名为TechDepart-LearningTask并把sysu17363040设为合作者。**

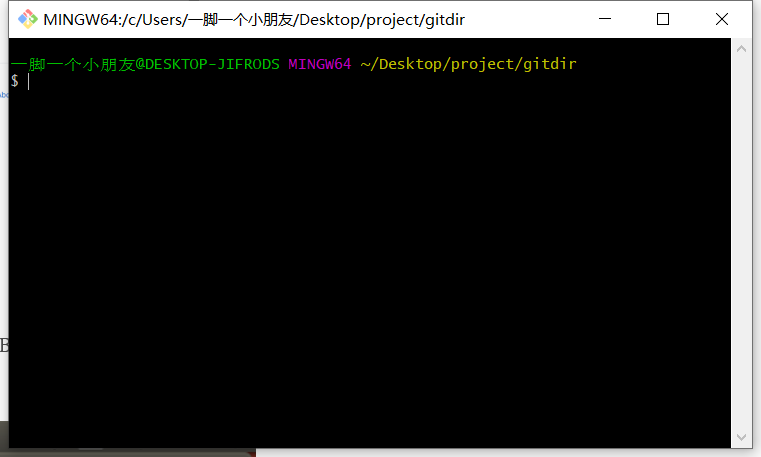


* **在客户端配置git**

首先需要下载 Windows 版本的 git，下载地址：[点击下载](https://git-scm.com/download/win)

下载完成后点击安装，并根据提示点击「下一步」，直到安装完成。

在客户端创建一个工作目录，比如我的是 gitdir，在工作目录内右键并点击 「Git Bash Here」。



一.安装并配置git

$ git config -l       //按下enter键，此时会显示undefined，没有关系，说明你是第一次运行git

$ git config --global user.name'XXX'

$git config --global user.email'XXX@xx.xx'     //请填写正确的邮箱

二．创建仓库（windows）

新建一个文件夹，打开文件夹，右键点击Git Bash Here

$ git init      //此时在当前所建的文件夹下会生成“.git”文件，请不要删除

三．将你所需要上传的文件放置新创建的文件夹下.

$ git add  -A      //将文件提交到暂存区

$ git commit -m'此处为描述，可以不用添加'  //将文件提交到历史区    使用 $ git log  命令可以查看提交记录

四．创建中央仓库

打开github    通过 new repository   创建一个仓库 此时会生成一个SSH

五．建立链接

$ git remote add origin [仓库地址，生成的SSH]       //此时会输入github账号和密码

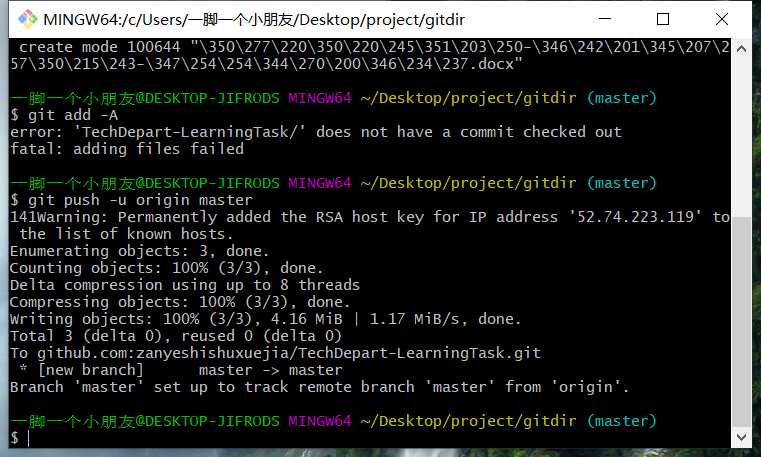
$ git remote -v 查看本地仓库和拿些仓库链接

六．上传文件

$ git pull origin master     //提交前可以先拉取一下

$ git push origin master   //此时就上传完成了 刷新一下github仓库即可看见上传文件

下图是呕心沥血才得到的成功图



**附Git学习网站：<https://blog.csdn.net/qq_33316091/article/details/82414389>**

1. **下载并安装pyhton3.7.3**

**补充知识：**

**源码的安装一般由3个步骤组成：配置（configure）、编译（make）、安装（make install）。**

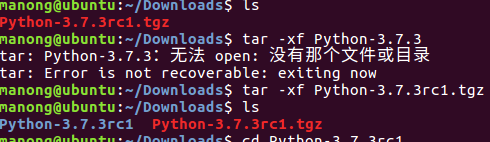
**configure是一个可执行脚本，它有很多选项，使用命令./configure --help输出详细的选项列表。最常用的参数是 --prefix=目录，这个目录就是软件最后的安装目录。**ls -l /usr/bin | grep python这个命令用来查看python版本。

1. **首先去**[**https://www.python.org/ftp/python/3.7.3/Python-3.7.3rc1.tgz**](https://www.python.org/ftp/python/3.7.3/Python-3.7.3rc1.tgz)**下载源码**

**这一步可以在虚拟机中通过命令行实现，具体步骤：切换到要下载的文件夹下(cd ~/Downloads),下载文件（wget** [**https://www.python.org/ftp/python/3.7.3/Python-3.7.3rc1.tgz**](https://www.python.org/ftp/python/3.7.3/Python-3.7.3rc1.tgz)**）**



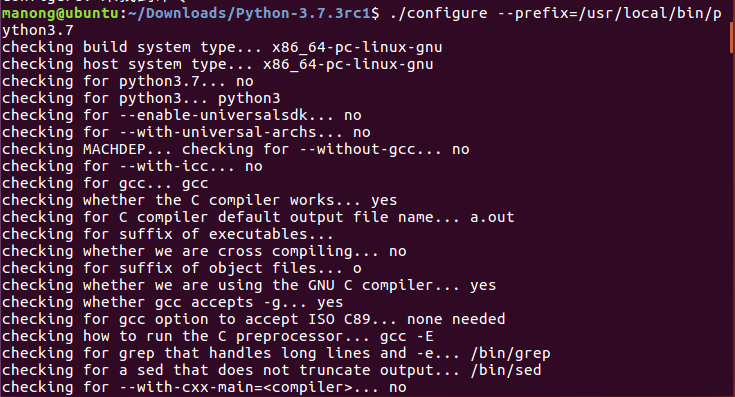
1. **解压文件夹(tar -xf Python-3.7.3rc1)**



1. **到要安装的文件夹下创建目录(mkdir /usr/local/bin/python3.7)**



1. **切近已解压的目录下（cd Python-3.7.3rc1）**
2. **指定安装位置（./configure –prefix=/usr/local/bin/python3.7）**



1. **下载安装依赖(sudo apt-get libffi-dev)**



1. **make**
2. **make install**

