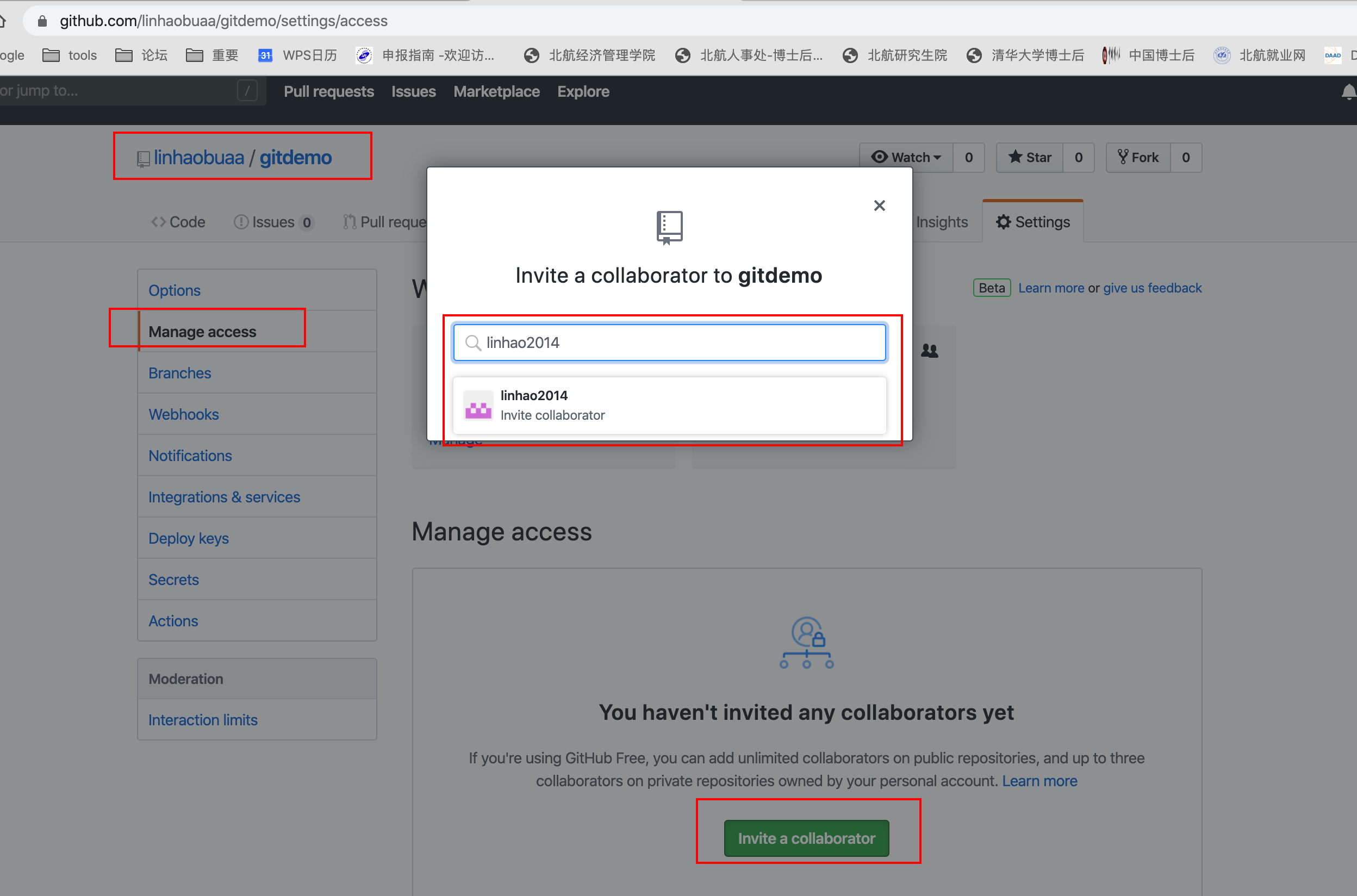
**Git协同开发与版本控制常见问题解答20200318**

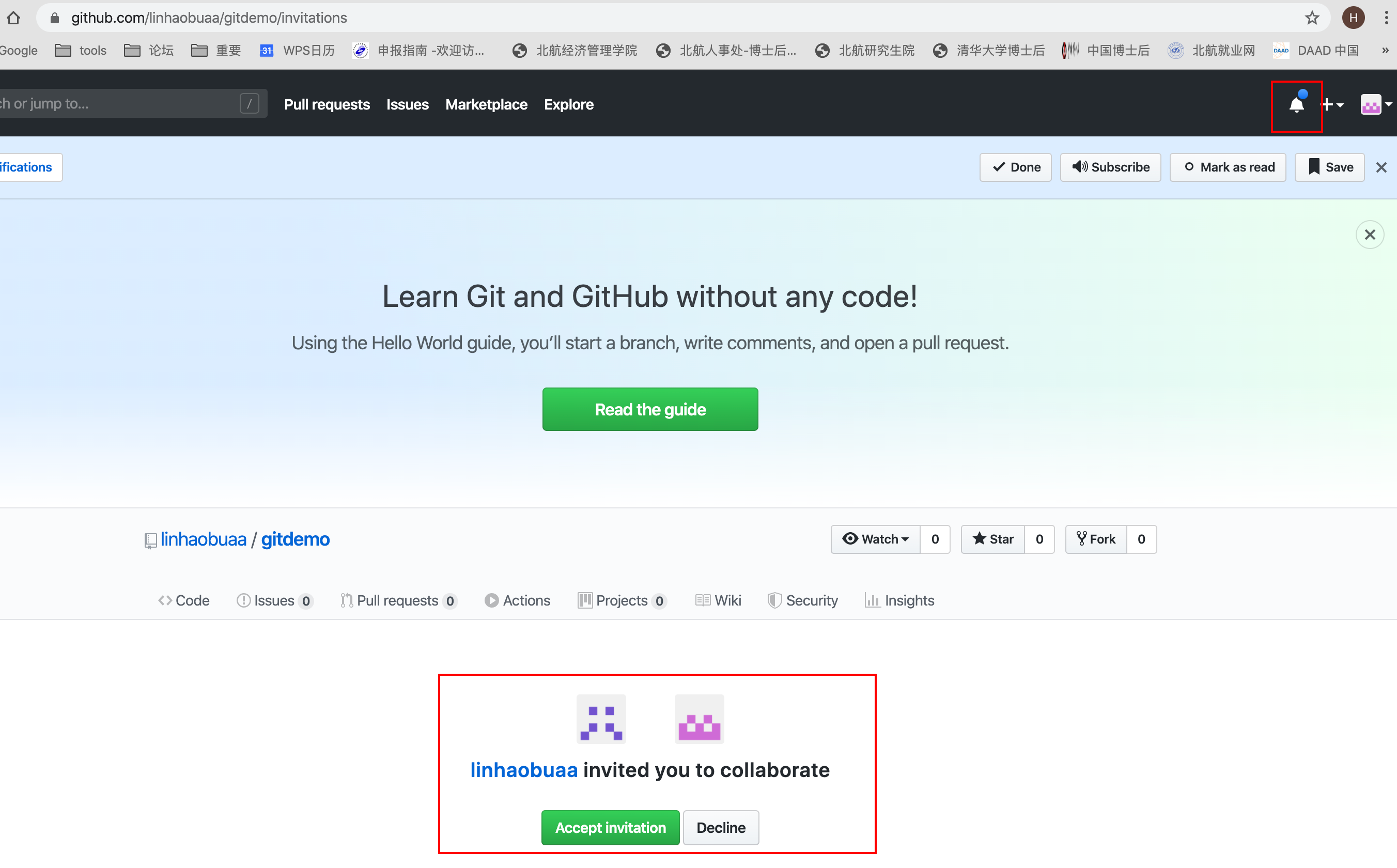
鉴于部分问题同学们给的反馈信息比较有限，缺乏必要的命令行上下文信息，只能给出可能的尝试方向，后面同学们反馈信息的时候可尽可能多地将命令行工具中已执行的其他命令、命令执行返回信息、操作系统信息等提供出来，便于帮助同学更好地解决问题。此外，同学们在利用搜索引擎工具进行问题解决时，推荐借助VPN工具使用Google搜索引擎直接搜索英文报错信息，搜索结果中推荐采纳技术问答网站**Stack Overflow**上建议的解决方案。

1. **git在使用第二个developer进行push操作时会提示fatal: unable to access The requested URL returned a error:403。**

答：403错误一般来说可能是权限的问题，推测可能是由于在使用Ubuntu系统进行git push时提示输入的用户名和密码（即第二个开发者的Github账户）不具备该Repository的写权限有关。实际上课程视频演示的git pull/push操作都是通过https协议进行的，https协议下理论上在进行所有git pull/push操作时都是需要输入Github账户名和密码的。然而，由于课程视频演示中使用了MacOS操作系统，MacOS系统在第一个开发者进行git push操作时已经记住第一个开发者的Github账号用户名和密码，因此后续在模拟第二个开发者进行git push操作时，无需再输入用户名和密码（实际上是用了第一个开发者的Github账户权限进行push），因此不存在Ubuntu系统上git push权限不对的问题。为解决这个问题有如下两个可选解决方案：

1. 第一种方案：第二个开发者在Ubuntu系统上进行git push提示输入用户名和密码时，输入第一个开发者的Github账户用户名和密码，相当于用第一个开发者的账号进行了代码push操作，就不会有权限问题，这在模拟环境下执行没有问题。
2. 第二种方案：如果在真实的开发环境下，第二个开发者在自己的仓库下进行Git push操作时不可能知道第一个开发者的Github账户用户名和密码，怎么办？按下图的方式，在第一个开发者（linhaobuaa）的Github project中邀请第二个开发者的Github账户（linhao2014）加为项目的Collaborator，第二个开发者登录自己的账号(linhao2014账户)同意邀请后，成为项目协作开发者，随后第二个开发者在命令行操作git push时，直接输入第二个开发者自己的Github账户的用户名和密码即可。





注：

* 第一张图是以linhaobuaa登录向linhao2014账户发送邀请，第二张图是以linhao2014登录并点击右上角消息按钮即可看到邀请信息。
* 如果按照上述方案进行git push操作时仍出现403问题，可检查是否正确输入了Github账号用户名和密码，以及输入的账号是否对repository具备access权限。
* 其他问题可参考<https://stackoverflow.com/questions/7438313/pushing-to-git-returning-error-code-403-fatal-http-request-failed>进行尝试。

1. **git注册不成功，网址打开后不加载，在网上查也没查明白。**

答：根据现有的信息，猜测是机器与Github的连接状况较差，提供如下方案进行git注册的相关尝试：

* 尝试不同的机器，例如换不同的PC机或手机
* 尝试不同的浏览器，例如firefox、chrome、safari
* 切换成不同网络，例如连接手机热点、连接境外VPN

1. **现有一文件，status是modified红色未commit，使用git stash，虽然stash记录多了一条，但是status还有这个文件（理论上应该是没有了），git stash pop会提示conflict 环境：Windows wsl子系统Ubuntu。**

答：git相关命令非常繁多，本课程只涉及其中一些核心命令。git stash相关操作本课程未进行介绍，无法给出具体的建议。鼓励感兴趣的同学们进行尝试，但同时建议同学们先参考相关命令的官方文档进行学习，理解命令背后的一些内在机制之后进行尝试可能效果更好（例如Git Stash可参考<https://github.com/progit/progit2-zh/releases/download/2.1.36/progit_v2.1.36.pdf> 中229-232页相关内容）。

1. **在windows系统中操作是一样的吗，为什么我按视频教程，在cmd中操作出现ls不是内部或外部命令，也不是可运行程序这样的异常。谢谢老师解答。**

答：该问题可参考上一次解答“Git协同开发与版本控制常见问题解答20200306”（后文随附）中“一、反馈问题情况概述”和“二、具体问题解答”第2个问题的回答。

**Git协同开发与版本控制常见问题解答20200306**

**一、反馈问题情况概述**

从同学们的反馈来看，目前主要问题**集中于Windows系统上Git的安装与命令行工具的使用问题**。鉴于PPT第5页已给出Git安装的官方说明链接**<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>**（涵盖Linux、MacOS、Windows三大主流操作系统的Git详细安装说明），分析原因在于部分同学对Windows系统上命令行工具的选择不太清楚。下面对PPT第5页Git的安装过程进行进一步阐述，便于同学完成后续演示过程。

演示过程中使用的主要工具包括命令行工具、文本编辑器和浏览器，其中命令行工具主要用于Git相关命令的执行、文件目录的切换等，文本编辑器主要用于项目代码编写与查看，浏览器主要用于Github相关操作的执行。

**Step 1：选择操作系统、命令行工具、文本编辑器。**

命令行工具是后续Git相关命令执行的载体，因此需要首先选择合适的命令行工具。

**情况一**：如果本地机器操作系统是MacOS，命令行工具建议选择如下两种方式之一（这两种命令行工具上相关命令用法基本上是一致的）：

* （1）系统自带的“终端”工具；
* （2）更加流行的iTerm 2工具（需要安装，可参考官网说明<https://www.iterm2.com/>，课堂演示用的就是这种方案）。

**情况二**：如果本地机器操作系统是Linux（如Ubuntu），命令行工具建议选择系统自带的“终端”工具。Linux“终端”的相关命令用法与MacOS中两种命令行工具的命令用法是类似的，同时Linux操作系统的整体布局、相关用法与课堂视频演示用的MacOS系统也是非常类似的。

**情况三**：如果本地机器操作系统是Windows（如Windows 7或Windows 10），可使用如下两种方式之一：

* （1）直接在Windows系统上进行操作：当在Windows系统上安装完成exe版本的Git安装包之后，可选择Git Bash作为命令行工具。这种方案**较简单，但由于使用MinGW命令集合，可能有部分命令与MacOS、Linux中命令行工具的用法不一致，采用这种方案作命令行工具的同学可尝试按课堂视频进行操作，有可能但不保证后续所有操作能成功**；
* （2）选择在Windows上安装Vmware，再安装Ubuntu虚拟机，安装完Ubuntu虚拟机之后，后续所有操作（包括Git安装、命令行操作、代码编写、浏览器使用等）都可在该虚拟机上执行，虚拟机中命令行工具可参考情况二，因此后续整个操作过程与课堂演示的MacOS的操作过程也会非常类似。**注意Ubuntu操作系统一定要选择Ubuntu Desktop版本而非Ubuntu Server版本**，因为Desktop指代的是含图形界面的操作系统，相比Server版本操作起来较为便利，安装过程可百度搜索“Windows Vmware Ubuntu”，这里仅给出其中一个博客说明：<https://blog.csdn.net/u012611644/article/details/80728806>。

注：

1、操作系统选择：**Linux和MacOS操作系统是个人编程和团队实际开发中最常见和流行的平台**，同时结合上述三种情况后续操作的便利性的分析，给出如下建议：如果操作系统是情况一、情况二，那么无需过多操作即可参考课堂视频进行后续操作；如果操作系统是情况三，那么建议可先尝试使用情况三中的第（1）种方案进行后续操作，若失败，则强烈建议根据情况三中的第（2）种方案安装Ubuntu虚拟机后进行后续操作（Ubuntu操作系统与MacOS类似，有一定的学习成本）。

2、文本编辑器的选择：无论是哪种操作系统，都建议安装sublime编辑器（可参考官网<https://www.sublimetext.com/>中相应操作系统对应安装说明进行安装）。

3、本解答只能为同学们列出一些关键安装建议，无法做到面面俱到，如安装过程中遇到问题（预料到同学们可能会遇到一些问题诸如Ubuntu/Windows如何打开命令行工具、Ubuntu虚拟机怎么无法连接互联网、某某安装包为何下载不了、Mac/Ubuntu操作系统怎么用命令行安装某某package等等），仍需各位同学积极地利用搜索引擎工具寻求解决方案，锻炼解决实际问题的能力。

**Step 2：安装Git。**

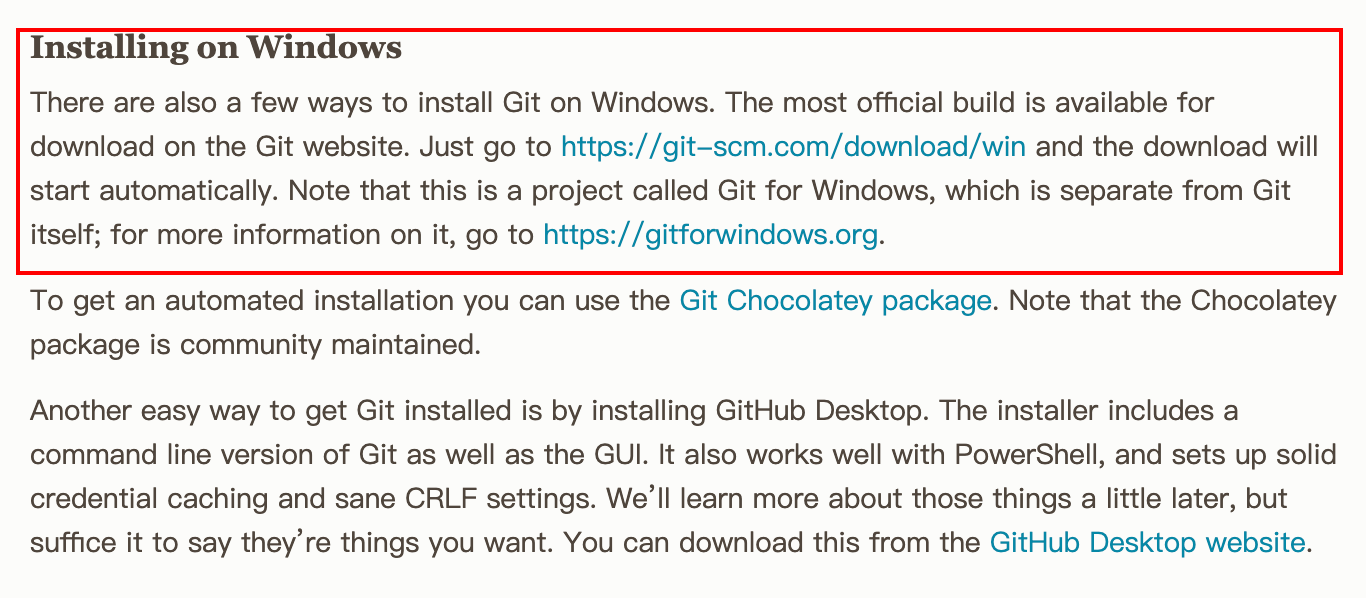
参考Git官网安装说明<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>，根据自己的操作系统进行相应安装。注意不要安装Github Desktop（它是Git的图形界面版本，建议同学们从Git的命令行工具开始学习，便于理解Git的用法与设计理念）。

**二、具体问题解答**

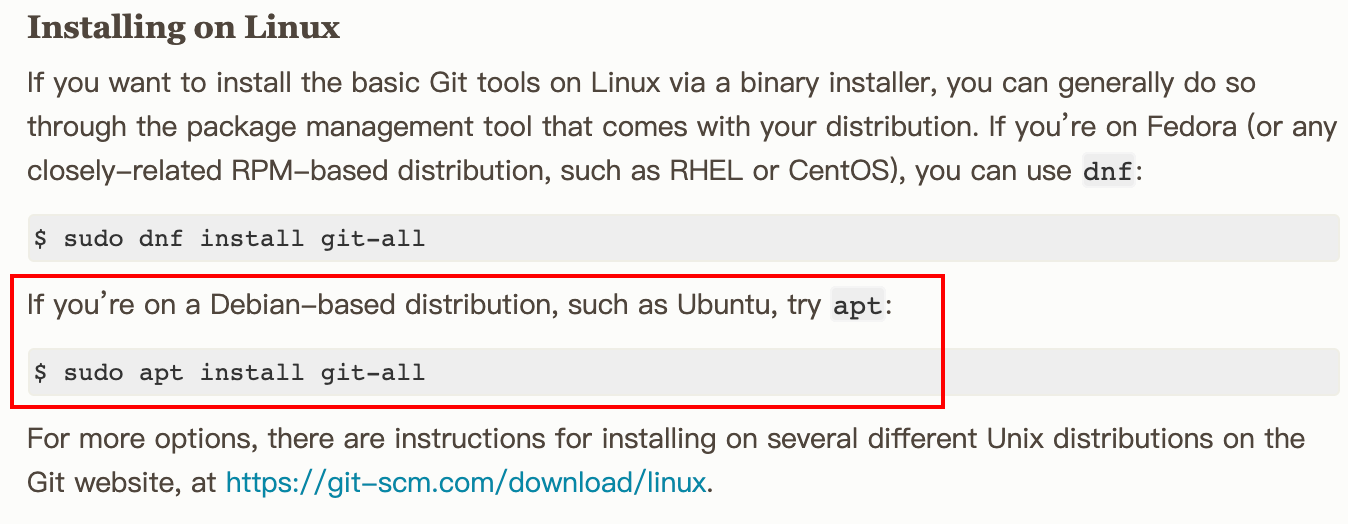
1. **Windows系统如何安装Git、使用Git?**

答：正如上面情况三所述，Windows系统上有如下两种操作方案：

（1）直接在Windows系统上进行相关操作：安装Git可参考PPT第5页中给出的Git官方说明链接进行操作<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>，下载相应的Git exe文件之后，双击打开exe文件，安装过程中一直选择next即可。安装Git完成之后，随后在windows系统搜索框（开始菜单右侧）中输入git bash，打开Git Bash命令行工具，在Git Bash中执行Git相关命令即可。



（2）在Windows上安装Ubuntu虚拟机，所有操作包括Git安装和使用、Github协同等都在Ubuntu虚拟机上执行。安装Git可参考PPT第5页中给出的Git官方说明链接进行操作<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>。其中安装Git命令“sudo apt install git-all”需要在Ubuntu虚拟机自带的终端工具上进行执行。



1. **如果是windows系统，操作和视频展示中的命令代码一样吗？可以把涉及到的命令统一在一个文档里方便学习吗？**

答：课堂操作的核心是命令行工具，关于Windows上命令行工具的选择已经在第一部分“反馈问题情况概述”中阐明，如果采用Windows上安装Ubuntu虚拟机的方式进行后续操作，在Ubuntu虚拟机上的操作方式与视频MacOS系统中相关操作非常类似。

Windows操作系统中，如果采用Windows自带的cmd作为命令行工具（DOS命令集合），基本上无法完成后续视频教程演示的操作，因为DOS命令用法与MacOS或Linux的命令行用法差异非常之大。一个替代方案是在Windows上采用Git Bash作为自带的命令行工具，Git Bash（MinGW命令集合）相比于cmd（DOS命令集合）更接近于MacOS和Ubuntu上的命令行工具，但仍可能存在差异。若Git Bash的方案操作失败，还是建议在Windows上安装Ubuntu虚拟机，再将Ubuntu虚拟机上的“终端”作为命令行工具来使用。

本课程主要涉及到的命令包括两大类，其一是Git相关命令，如git add/commit/pull/push等，详细文档参考官方链接（<https://git-scm.com/book/en/v2>中Chapter 2 Git Basics）；其二是基本的Bash命令（MacOS和Linux操作系统命令行工具中执行的一种脚本语言，如cd、pwd、touch、ls等）。下面列示本演示过程中用到的命令的含义以及在各命令行工具上的可用性（**可用性是指能否在各命令行工具上正常使用相应命令，能使用这些命令的前提是已正确安装Git，并且首次执行各命令之前需重启各操作系统的命令行工具**）。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **命令** | **含义** | Windows | | Ubuntu  终端 | MacOS  终端或iTerm2 |
| cmd | Git Bash |
| git相关命令 | git add/commit/pull/push等 | √ | √ | √ | √ |
| cd | 切换目录 | √ | √ | √ | √ |
| pwd | 显示当前所在目录 | × | √ | √ | √ |
| touch | 创建文件 | × | √ | √ | √ |
| ls | 查看当前目录下的所有文件信息 | × | √ | √ | √ |

1. **mac的一些操作名命令和windows不一样，为什么我按视频教程，在cmd中操作出现ls不是内部或外部命令，也不是可运行程序这样的异常。**

答：Windows中自带cmd的操作与视频教程中的命令行工具的操作差异非常大，在Windows自带的cmd中基本上只能使用cd命令进行文件路径的切换，别的命令包括ls执行不了，详情可见上一问的回答。

1. **好像没有听到关于版本回滚的操作。**

答：考虑到不同同学的学习基础存在差异的现实情况，本课程重点介绍Git提交代码到版本库和Git协同开发的Git基础内容，未介绍Git版本回滚（https://git-scm.com/book/en/v2/Git-Basics-Undoing-Things）、Git分支管理（https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell）等Git进阶内容。作为可选项，鼓励感兴趣的同学对相关内容进行课下自学、演示，并将操作流程、心得体会提交到最终作业报告中。

1. **编程新手（零基础）如何下载、如何操作、如何学习Git的使用？**

答：本次课程是一个实践性较强的教学环节。考虑到部分零基础同学的情况，本解答将Git相关的安装说明在第一部分“反馈问题情况概述”中进行进一步说明，以保证同学可以按照相关材料进行Git及相关工具安装。一旦相关环境安装完成，同学即可参考教学视频内容，一步一步进行操作，以项目回顾式开发的方式，动手体验Git本地代码提交、Git协同开发的流程，希望同学在此过程中思考并感受Git相关的设计理念，并完成相关作业内容。

1. **git pull为何每次需要输入用户名和密码？**

答：不同操作系统上git操作的表现略有不同，目前来看Ubuntu系统上每次操作需要输入用户名和密码，Mac上是不需要的。

1. **为何模拟两个人操作时，github上只能看到一个contributor？**

答：在本地两个工作目录下需要通过git config配置不同的user.email(对应两个Github账号的注册邮箱)，在Github的commits和contributors里面就可以看到两个用户了，其中contributors如果显示一个，那么可能是因为缓存。

1. **Git pull操作比较卡？**

答：可能与本地机器的网络连接状况较差有关，不同网络下本地机器与github通信速度不同，可尝试切换到手机热点/VPN等。