

Git使用说明

前言

考虑到我们不是一项需要持续开发维护的工程（顶多做个半年）我就不详细介绍git的原理与一些比较偏僻的指令了，主要根据不同情况具体介绍如何使用git命令，并结合vscode上的图形化界面。

文件结构

/CPUDesign_nwpu1

 / AXI 代码（暂时为空，加入总线后的代码）

 / no_AXI （当前主要代码目录）

 / study_record （第二个子模块，**这是一个和其他文件夹都不一样的文件夹**，这是一个子模块）

 / AXI （这四个可以放我们的学习日志）

 / Cache

 / TLB

 / Tools

 / reference_code （这个是之前在CPUDesign_nwpu1目录里的参考代码资源，现在放在这了，里面有学长的代码）

下面我们分情况来说一下git的使用，首先说一说在/CPUDesign_nwpu1下的可能操作

1.远处的main分支有更新，自己的分支较为落后。

下午时，每个人都创建了自己的分支，创建时相当于复制了一份当前分支的内容（换了一个分支名字）之后两个分支互不影响了（但还存在联系），这个时候main分支有更新了但自己的分支并没有被更新修改，要想使用到最新代码而又不放弃自己的修改，需要merge操作。

git bash 操作（命令窗口）：

```
# 进入到根目录下，如果已在CPUDesign_nwpu1目录下，则不需要操作
cd /CPUDesign_nwpu1

#切换到有更新的分支（这里以main为例）如果已在分支下，则不需要切换
git checkout main

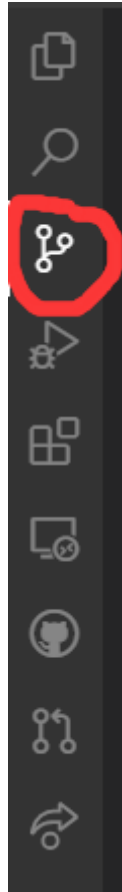
#拉取更新（这个时候只是远程代码更新了，本地main分支还未更新，需要pull一下）
git pull

#切换到自己分支（这里以我的分支为例）merge更新后的main（！！！注意merge的对象，不要merge反了）
git checkout xkx_develop

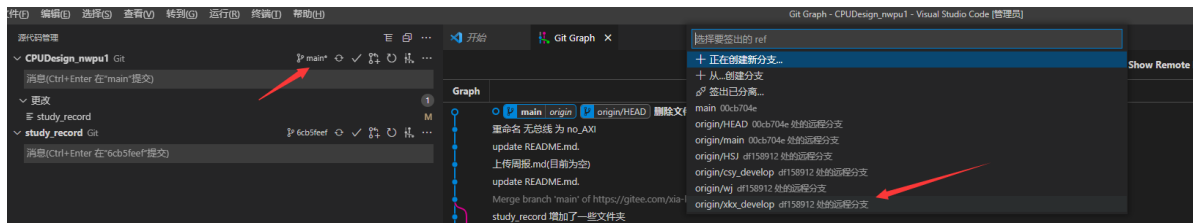
#现在我在自己分支了
git merge main
```

vscode (图形化界面)

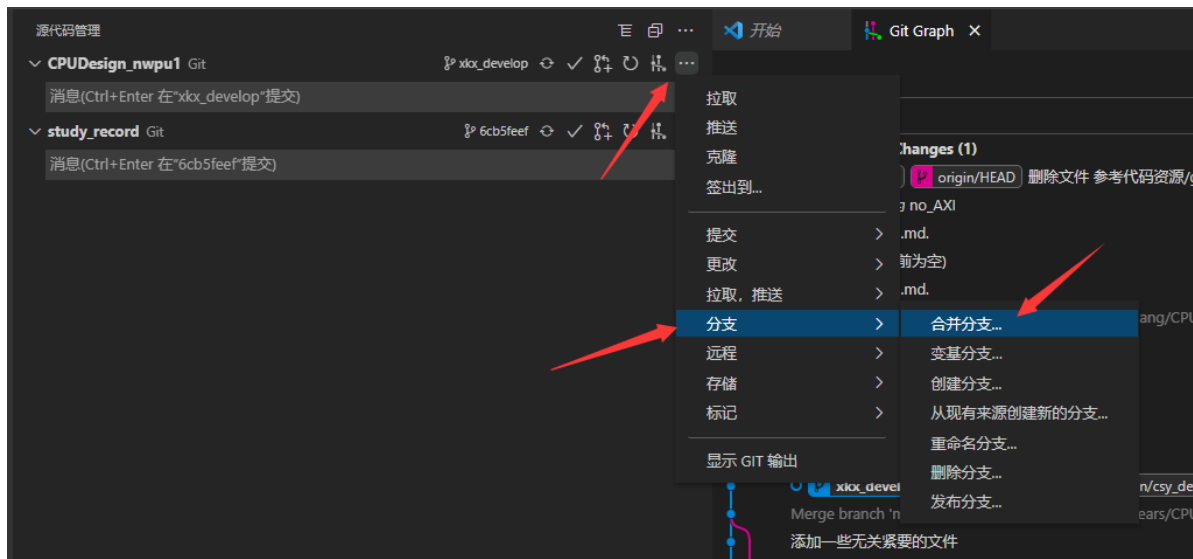
这个就比较简单了，我上图吧



vscode里面应该默认有这个东西，点开它，会发现有两个模块，并且所处分支名字也显示出来了。点击可以切换到你的分支。



然后合并分支，选择main (!!! 一定要切到你的分支，然后合并main，不要反掉)



PS:

合并一般情况下，会有冲突，这很正常，原因是你与更新的人在同一地方做了不同的开发，导致冲突，这时候就需要你在合并的时候选择保留一个更改，这种情况运用vscode的图形化界面比较方便，（遇见了再说，比较简单，很好解决）。这意味着如果你在命令窗口输入git merge main提示有合并冲突之后，你依然可以回到vscode，运用图形化界面解决冲突，如果你是用的vscode合并，那当然也可以使用vscode解决。

2.工作了一天了，想把工作成果上传到云端保存(自己分支)

上传的操作还是先在自己分支操作，如果你想上传main分支最好还是先在群里和大伙说一声你的代码通过了测试啥的。

这里额外说一声，不管你在本地电脑对分支内的内容做了什么操作，只要你没有在分支下运行git push指令，就不会影响远程代码，也就不会对他人代码造成影响。

git bash 操作（命令窗口）：

```
#到达你想上传的分支下的文件中（确保分支与位置正确）
#.....

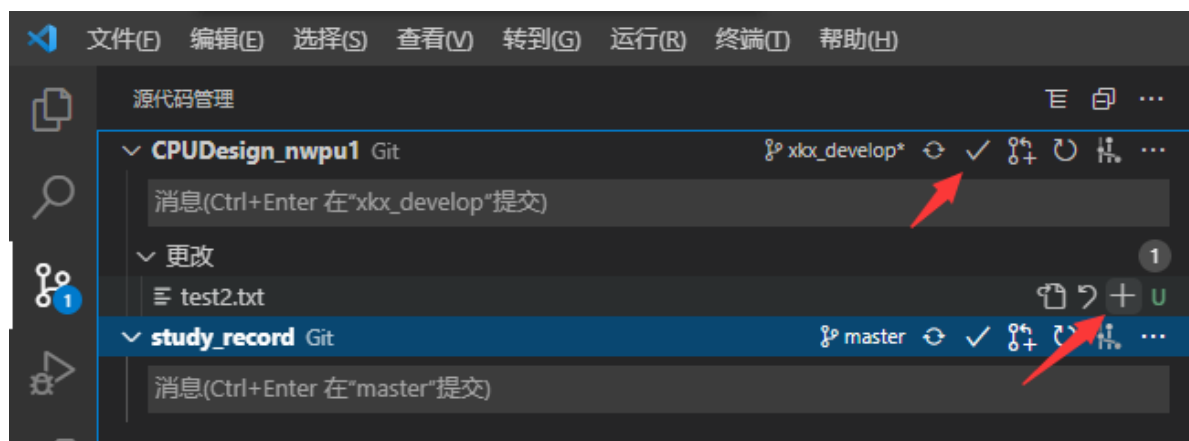
#暂存更改
git add .

#提交说明
git commit -m "对xxx文件进行了修改"

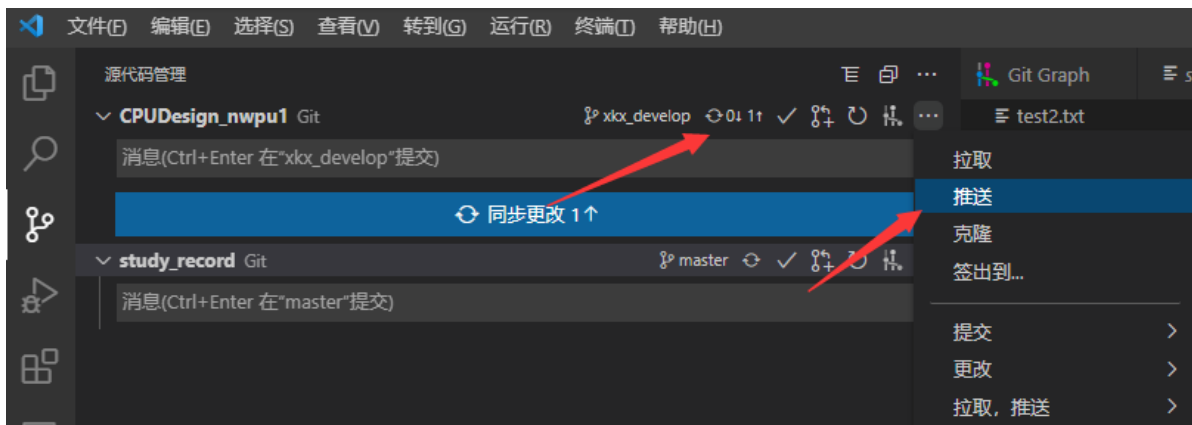
#推送（如果你下面日志上传操作参考了此处自己分支上传的流程，这个地方可能要加一条git pull 指令，根据报错，你大概能懂是什么原因）
git push
```

vscode（图形化界面）：

+相当于add操作，√相当于commit操作，√之后会让你写提交信息。



接下来2选1，都可以推送。



接下来说一说比较特殊的study_record文件夹

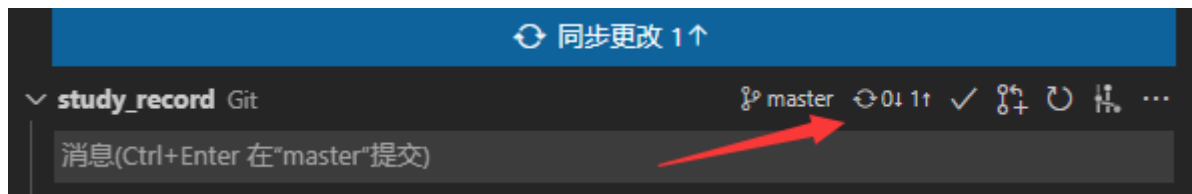
这是一个子模块，相对而言，学习日志的上传应该比代码更改更加频繁。比较神奇的是，如果你直接把/CPUDesign_nwpu1克隆下来的话，会发现study_record文件夹是一个空文件夹，此时你需要在/CPUDesign_nwpu1下运行一行命令

```
git submodule update --init --remote
```

```
Xiakx@LAPTOP-ANVL9AV5 MINGW64 ~/Desktop/CPUDesign_nwpu1 (main)
$ git submodule update --init --remote
Submodule 'study_record' (https://gitee.com/xia-kangxiang/study_record.git) regi
stered for path 'study_record'
Cloning into 'C:/Users/Xiakx/Desktop/CPUDesign_nwpu1/study_record'...
Submodule path 'study_record': checked out '6cb5feef3e7372cffaabb40079397c9462cb
475e'
Xiakx@LAPTOP-ANVL9AV5 MINGW64 ~/Desktop/CPUDesign_nwpu1 (main)
```

这里study_record就只有一个分支了，不需要自己独立创建分支了，不然互相阅读学习还要切来切去
说一下日常上传日志的操作（用vscode举例，bash命令参考上面）：

先点+然后点√，最后点这个：



然后点确定。