git使用方法

作者：滑国青

历史：2023.07.10 v1.0

# 前言

# 常用合版本操作

## 查看所有分支

git branch -a

## 切到分支或主干

git checkout main

git checkout [分支名]

## 重置修改过的文件或目录

git checkout [文件或目录] 可以将文件或目录里的修改，恢复为原文件，即删除修改。

在Git中，git checkout命令用于切换分支或恢复文件。然而，当你在切换分支时，如果当前分支上的文件有未提交的修改，Git会给出一个警告，因为切换分支可能会导致未提交的修改丢失。

这种情况下，你有几种选择来解决语义冲突问题：

1. 提交或保存修改：如果你想保留当前分支上的修改，可以先提交或保存修改，然后再切换分支。你可以使用git add和git commit命令来提交修改，或者使用git stash命令来保存修改并将工作区恢复到干净的状态。
2. 丢弃修改：如果你不关心当前分支上的修改，可以使用git checkout命令加上-f或--force选项来强制切换分支并丢弃修改。这样做会导致未提交的修改永久丢失，请谨慎使用。
3. 创建新分支：如果你想保留当前分支上的修改，并且不想提交或保存它们，你可以使用git checkout -b <new-branch>命令创建一个新的分支，并在新分支上继续工作。这样做可以避免丢失修改，并且可以在需要时回到原来的分支。

总之，当你在切换分支时遇到语义冲突问题，你需要权衡是否要保留当前分支上的修改，并选择适当的方法来解决冲突。

1. 清除单个文件的修改：

复制代码

git checkout -- <file>

其中，<file>是要清除修改的文件路径。

例如，要清除对example.txt文件的修改，可以使用以下命令：

复制代码

git checkout -- example.txt

1. 清除所有文件的修改：

复制代码

git checkout -- .

这将清除所有已修改的文件。

请注意，这些命令将会丢弃所有未提交的修改，恢复文件到最近一次提交的状态。在执行这些命令之前，请确保你不再需要这些修改，并且已经备份了重要的更改。

如果你只是想撤销最近的提交并保留修改，可以使用git reset命令。例如，要撤销最近的提交并保留修改，可以使用以下命令：

复制代码

git reset HEAD~

这将撤销最近的提交，并将修改保留在工作区中，以便进行进一步的修改或提交。

## 将主干代码同步到分支上

1. 先切到主干，同步主干的代码
2. 再切回分支，将主干代码merge到分支

## 如何记录提交的帐户

git config --global credential.helper store

## git只下载部分文件夹

例如：<https://adasgitlab.autel.com/shrd/shrd100_ps_linux.git> 项目

如果只下载linux\_app目录,则按下步骤，将.git/info/sparse-checkout文件内容，改为/linux\_app/\*

$ git init shrd100\_ps\_linux

Initialized empty Git repository in /home/a23286/git\_src/test/shrd100\_ps\_linux/.git/

$ cd shrd100\_ps\_linux/

$ git sparse-checkout init

$ vi .git/info/sparse-checkout

内容改为：

/\*

!opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/drivers/gpu/drm/nouveau/nvkm/subdev/i2c/aux.h

!opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/drivers/gpu/drm/nouveau/nvkm/subdev/i2c/aux.c

!opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/include/soc/arc/aux.h

$ git sparse-checkout reapply

$ git remote add origin https://adasgitlab.autel.com/shrd/shrd100\_ps\_linux.git

$ git pull origin master

原始问题：在windows上,下载shrd100\_ps\_linux时，会出现

问题现象是:error: invalid path 'opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/drivers/gp  
u/drm/nouveau/nvkm/subdev/i2c/aux.c'  
error: invalid path 'opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/drivers/gp  
u/drm/nouveau/nvkm/subdev/i2c/aux.h'  
error: invalid path 'opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/include/so  
c/arc/aux.h'

## Git之回退已经提交到远程仓库的代码

即：已经push的代码

1）、我们用git log命令找到commit Id

d6434e39e842a40b79119b2b88e76dd1498f8654是我最后一次提交，

然后我们现在需要回退到145c4fc7f237176acba1bca515e965f55eb8ca53

2)、先回退本地的这次提交

git reset --hard [commit id]

3）、将本地修改强制push到远程仓库

git push --force

## 清除缓存的\_\_pycache\_\_

如果我想保留硬盘上的这个文件，而只删除版本管理中的文件，就需要加入--cached参数。

find . -name '\_\_pycache\_\_' -type d -exec git rm -r -f –cached {}

## 本地分支推送到远程并关联

git push origin huaguoqing:user/huaguoqing

## 如何rebase

Merge blocked: fast-forward merge is not possible. To merge this request, first rebase locally.

To resolve this issue, you need to perform a rebase locally before attempting to merge the request again. Here's a step-by-step guide to help you through the process:

1. Start by checking out the branch that you want to merge into. For example, if you want to merge a feature branch into the main branch, run the following command:

复制代码

git checkout main

1. Next, pull the latest changes from the remote repository to ensure your local branch is up to date:

复制代码

git pull origin main

1. Now, switch back to the branch you want to merge. For instance, if you want to merge a feature branch named "feature-branch," run:

复制代码

git checkout feature-branch

1. Perform the rebase operation on your local branch using the following command:

复制代码

git rebase main

This command will apply the changes from the main branch on top of your feature branch, ensuring that your branch is up to date with the latest changes.

1. During the rebase process, you might encounter conflicts if there are conflicting changes between the two branches. Git will pause the rebase and allow you to resolve these conflicts manually. Open the conflicting files, resolve the conflicts, and save the changes.
2. After resolving the conflicts, stage the changes by running:

复制代码

git add .

1. Once all conflicts are resolved and staged, continue the rebase by running:

复制代码

git rebase --continue

1. If there are no further conflicts, the rebase will complete successfully. At this point, you can push the rebased branch to the remote repository:

复制代码

git push origin feature-branch

1. Finally, you can attempt to merge the request again, either through the Git hosting platform or by running:

复制代码

git merge feature-branch

By following these steps, you should be able to rebase your local branch and resolve the "fast-forward merge is not possible" issue.

# gitignore文件匹配规则

## 作用域

gitignore只匹配其所在目录及子目录的文件。

已经被git track的文件不受gitignore影响。

子目录的gitignore文件规则会覆盖父目录的规则。

## 文件与文件夹匹配

在表达式前头出现/则匹配从当前目录（gitignore所在目录）开始的相应文件/文件夹。

在表达式后加/来匹配文件夹。如表达式hello/只会匹配文件夹hello, 而不会匹配文件hello。

\*\*匹配任意文件/文件夹，或者空内容。如a/\*\*/b匹配a/b,a/x/b以及a/x/y/b。

## 模糊匹配

\*匹配除了/之外任意数量的任意内容。

?匹配除了/之外的任意一个字符。

[]匹配包含在[]范围内的任意字符。

## 取消匹配

在表达式前使用!来取消匹配，然而当其父文件夹已经被其他表达式匹配时，则该条表达式无效。

## 特殊字符

在git中有特殊含义的字符如!, #等在匹配时在前面加\来转义。

表达式后面的空格会在匹配时被忽略掉，如果想要匹配名字后面带有空格的文件/文件夹时，可以使用\加空格来匹配空格。

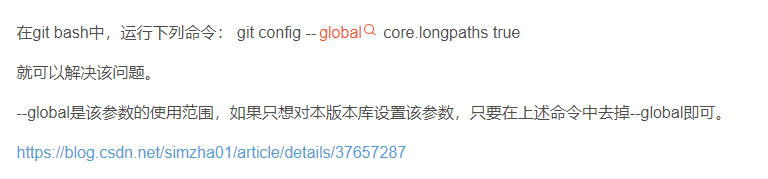
## 其他

使用#添加注释。

# 常见问题

### git pull 时显示Filename too long的解决办法

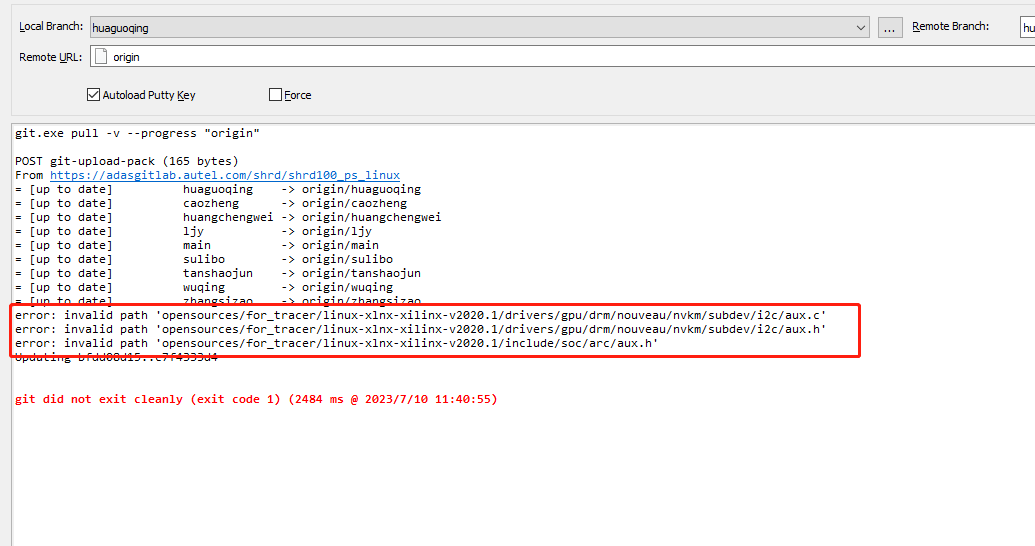
fatal: cannot create directory at 'kernel/modules/RTL8821CS\_WiFi\_linux\_v5.14.2-28-g6011b0372\_COEX20210319-5555.20220328/driver/rtl8821CS\_WiFi\_linux\_v5.14.2-28-g6011b0372.20220328\_COEX20210319-5555.hua.bak/hal/halmac/halmac\_88xx/halmac\_8821c': Filename too long



### error: RPC failed; HTTP 413 curl 22 The requested URL returned error: 413 Request Entity Too Large

不要上传超大文件。

### windows下获取不下来linux下的一些文件



解决办法：使用git的sparse-checkout功能，取消下载aux文件

$ git init shrd100\_ps\_linux

Initialized empty Git repository in /home/a23286/git\_src/test/shrd100\_ps\_linux/.git/

$ cd shrd100\_ps\_linux/

$ git sparse-checkout init

$ vi .git/info/sparse-checkout

内容改为：

/\*

!opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/drivers/gpu/drm/nouveau/nvkm/subdev/i2c/aux.h

!opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/drivers/gpu/drm/nouveau/nvkm/subdev/i2c/aux.c

!opensources/for\_tracer/linux-xlnx-xilinx-v2020.1/include/soc/arc/aux.h

$ git sparse-checkout reapply

$ git remote add origin https://adasgitlab.autel.com/shrd/shrd100\_ps\_linux.git

$ git pull origin main

# 结束