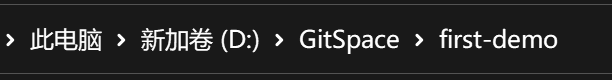
Git使用记录

1. 安装git，安装后在git dash中输入以下命令来设置用户签名，此处名字和邮箱随便填，与github账号无关

$ git config --global user.name wxm

$ git config --global user.email email@example.com

2. 创建仓库文件夹



3. 使用命令定位到仓库目录下（目录名称尽量不要带空格，如果带空格，输入git dash中目录名字需用引号）

cd d:/GitSpace/first-demo/

4. 初始化本地库，使用命令

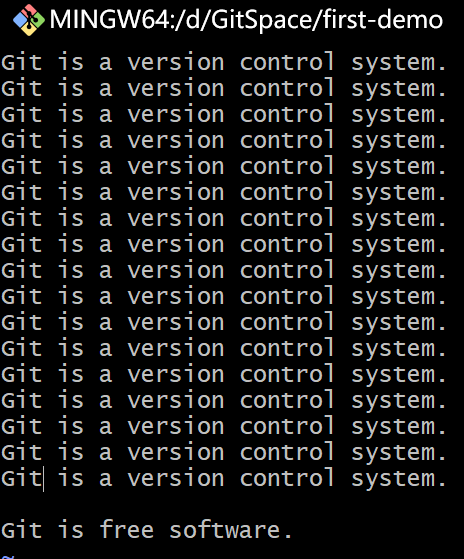
git init

5. 查看本地库状态

git status

创建一个hello.txt文件

vim hello.txt

内容为：

6. 添加暂存区

git add hello.txt

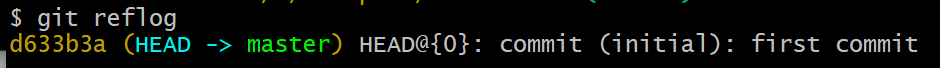
git add -A 将所有文件添加到暂存区

7. 将暂存区的文件提交到本地库，'first commit'为版本号，自己命名， -m后面输入的是本次提交的说明，可以输入任意内容，当然最好是有意义的，这样你就能从历史记录里方便地找到改动记录

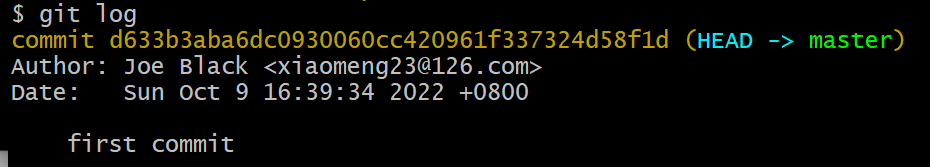
$ git commit -m 'first commit' hello.txt

查看版本号：

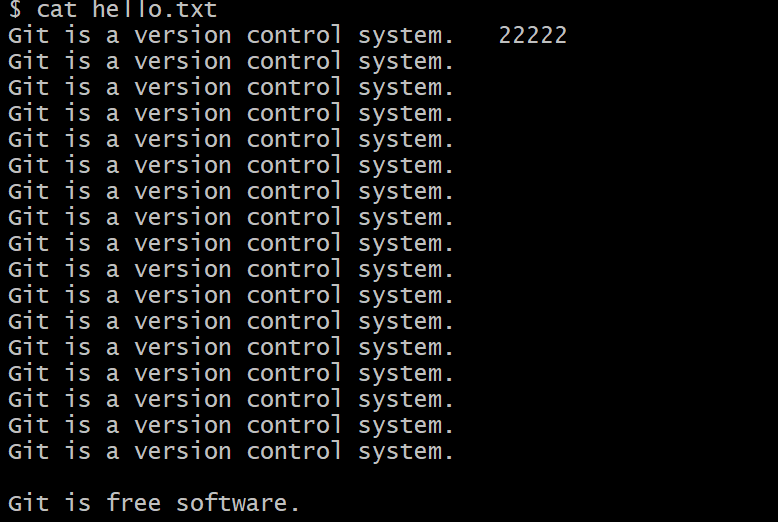
git reflog



git log

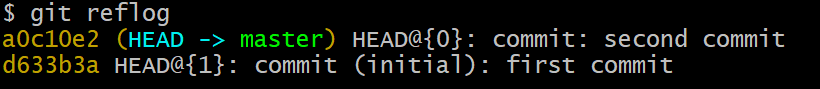


8. 修改文件后需要添加暂存区，添加到本地库



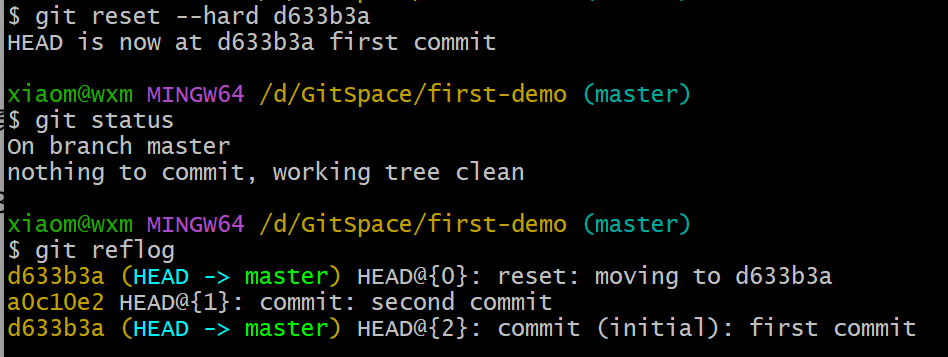
9.再次查看版本号

git reflog

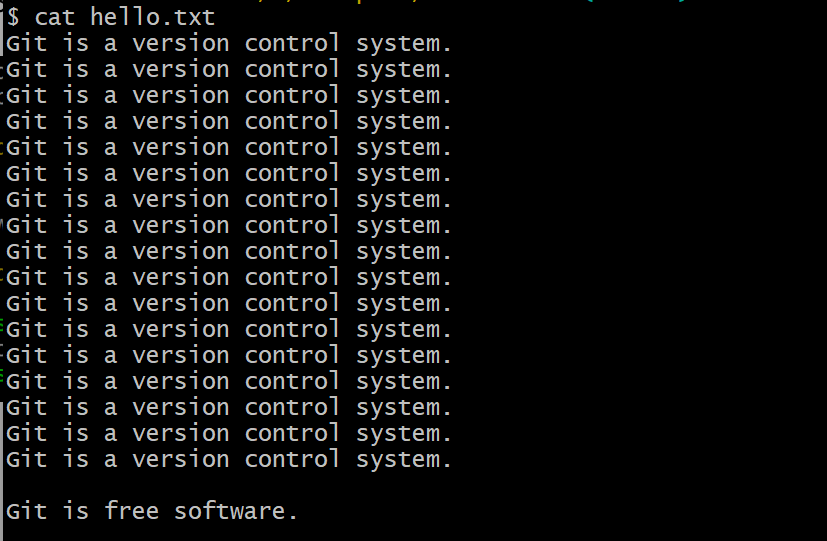


8. 版本穿梭。例如想穿梭到第一个版本，版本号选择为d633b3a

git reset --hard d633b3a



查看内容发现hello.txt文件已修改为第一次的内容

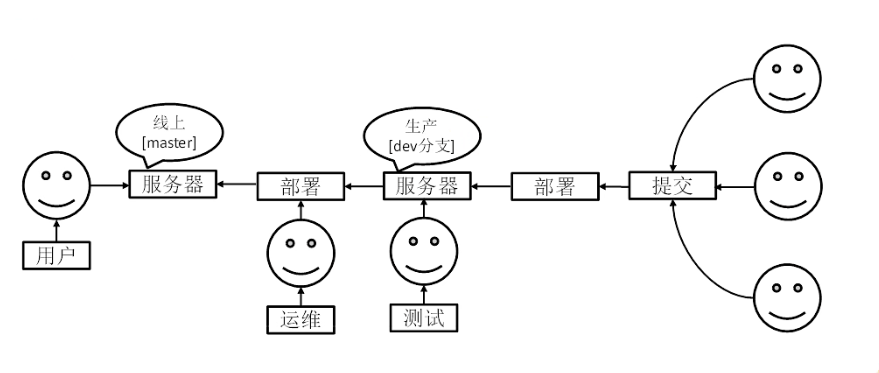


9. git分支。

优点：可以同时推进多个功能开发，提高开发效率；各个分支在开发过程中，如果一个分支开发失败，不会对其他分支有任何影响。失败的分支删除后重新开始即可。

分支的操作：

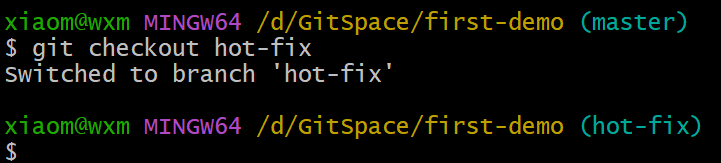
|  |  |
| --- | --- |
| git branch 分支名 | 创建分支 |
| git branch -v | 查看分支 |
| git checkout 分支名 | 切换分支 |
| git merge 分支名 | 把指定的分支合并到当前的分支上 |



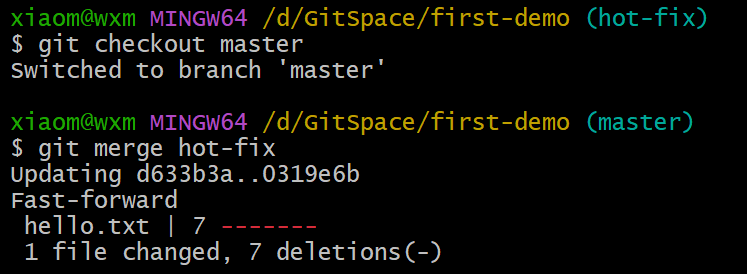
9.1 创建分支



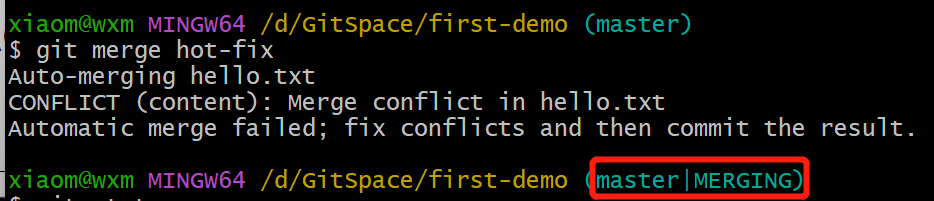
9.2 切换分支



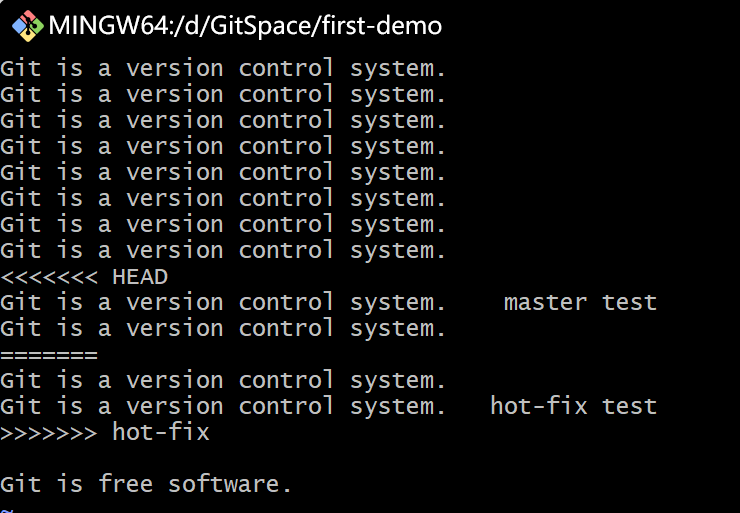
9.2 正常合并分支：1.切换到master分支，2.合并分支



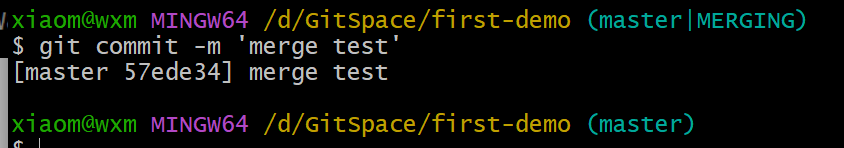
9.3 冲突合并，会发现合并失败，同时当前处于合并中。



打开文件需要手动调整不同的地方。



保存后提交暂存区，提交版本号，不要带文件名。合并成功。

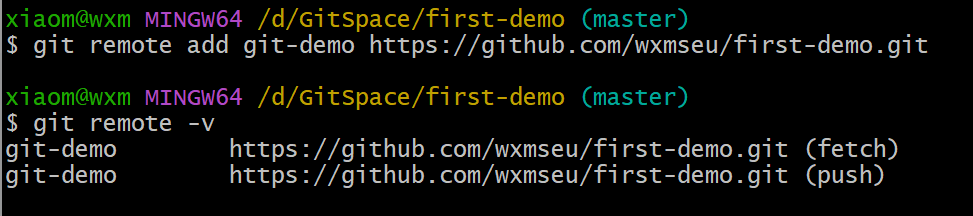


10. 创建远程库，在github中创建仓库时最好库名与本地库名相同，本例中为first-demo。

创建远程库别名

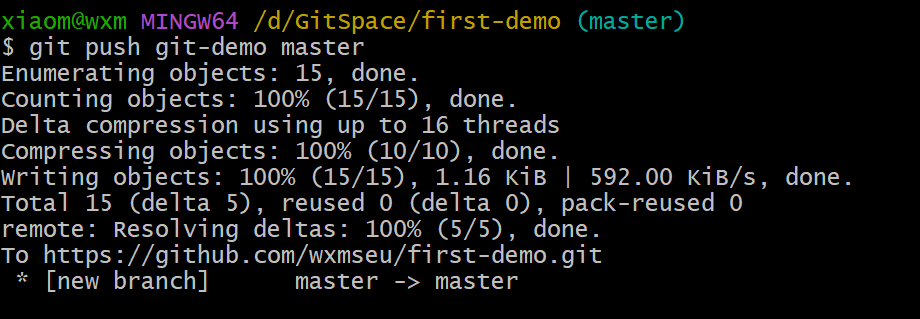
git remote -v 查看当前所有远程地址别名

git remote add 别名 远程地址



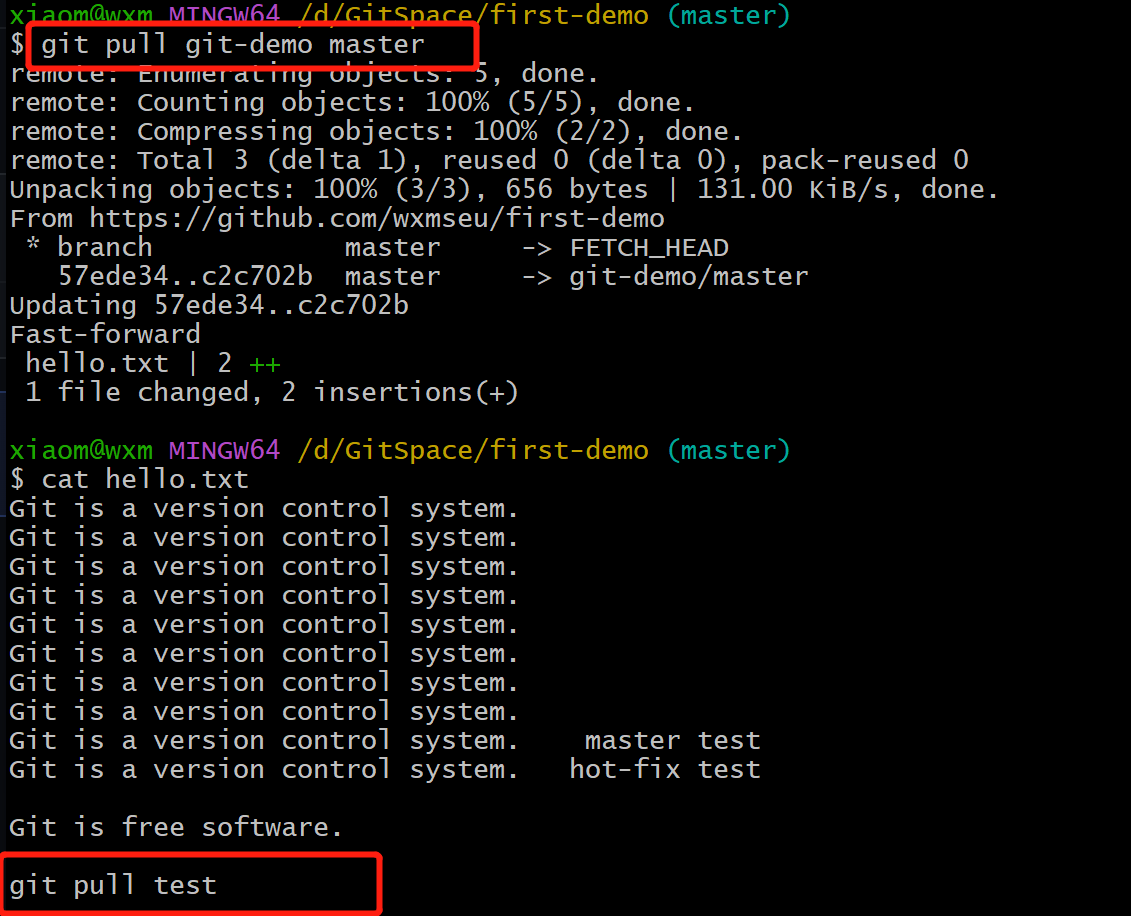
10.1 推送本地库到远程库

git push 别名 分支



10.2 在github远程库中修改文件后，应当从远程库拉取到本地库。

git pull 别名（地址） 分支



10.3 克隆远程库

git clone 地址（https，ssh）

克隆公共远程库不需要账号密码

11. 解决git pull和push连接远程库失败

取消全局代理

git config --global --unset http.proxy

git config --global --unset https.proxy