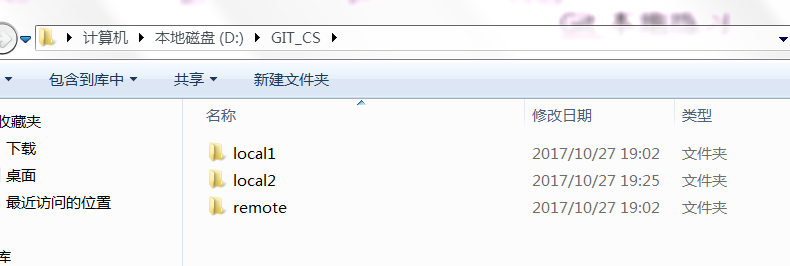
Git本地练习

1. 搭建本地练习环境

为帮助大家尽快熟悉git使用方法，可在本地进行模拟练习，辅助提高开发效率，由于水平有限，这里只介绍常用的部分。Git更多的命令可以直接在终端git --help获取。

git支持多种协议，这里我采取的是file协议来搭建我们本地的模拟环境。

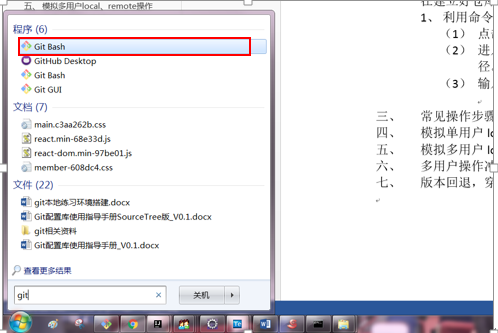
1. 在本地新建一个文件夹（为了目录的简洁，也可跨盘操作），再新建两个子文件夹（local和remote），local文件夹来模拟我们的本地仓库，remote模拟远程仓库。



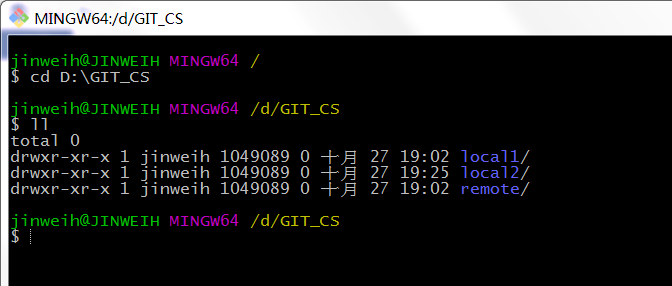
1. 本地环境配置

在建立好仓库的基础上，需要进行环境得到初始化操作，和基本的配置。

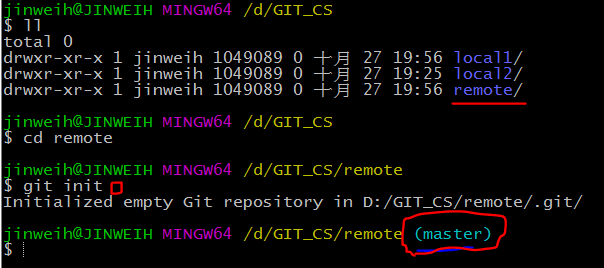
1. 利用命令git init 来初始化一个git仓库。
2. 点击windows键，输入git，点击gitbash。



1. 进入git终端，这里支持linux命令，找到建立的远程仓库路径。

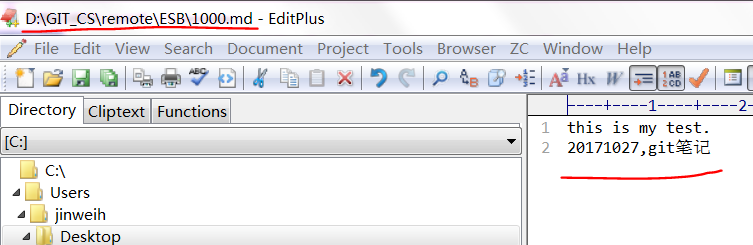


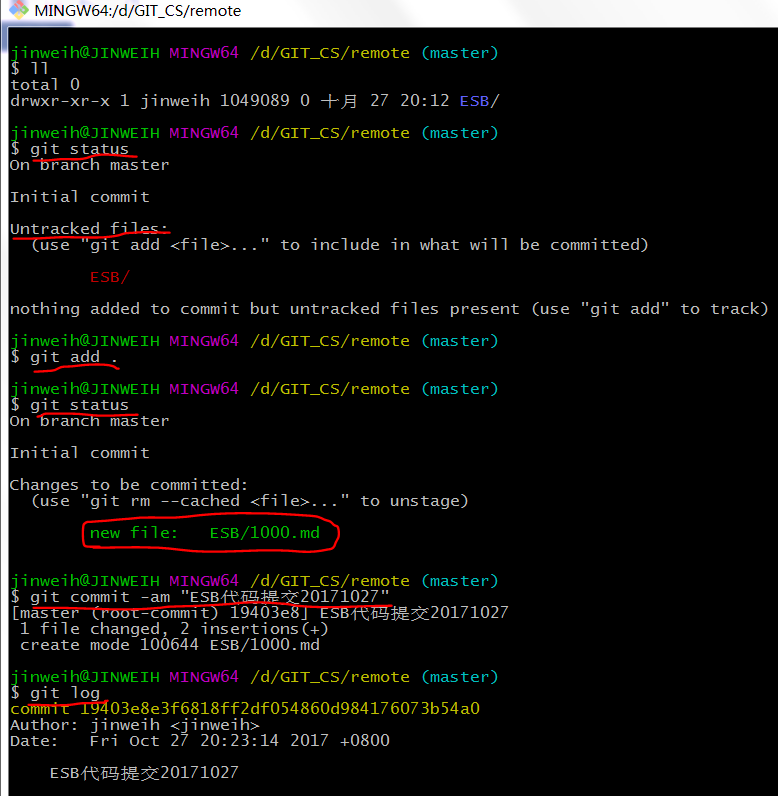
1. 输入git init命令来初始化一个仓库。就会在远程仓库产生一个*.git*的隐藏文件夹。最初初始化的仓库为master分支。



1. 本地仓库配置

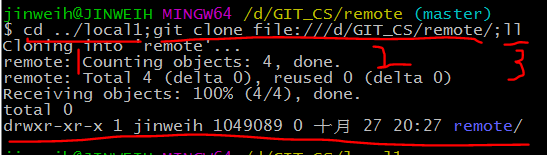
（1）远程仓库建立好之后，我们在远程（remote）建立工程，写入信息，然后下载到本地仓库（local）。从远程克隆工程使用 git clone来实现。





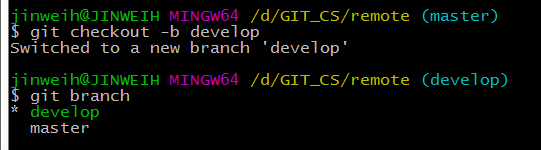
图一 远程仓库信息提交

（2）在本地仓库进行克隆,查看本地仓库已经从远程仓库下载到工程。

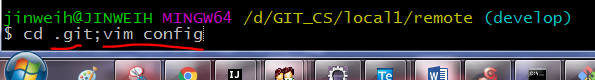


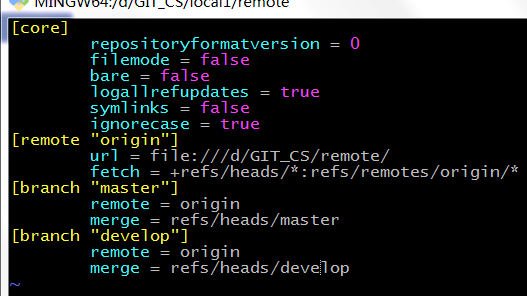
图二 克隆远程仓库到本地

（3）分别在远程仓库和本地仓库检出新的develop分支。



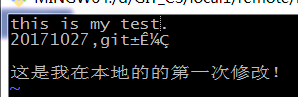
（4）本地develop分支关联远程分支。进入本地 .git隐藏文件夹，打开config文件，按下图进行配置。





至此，本地和远程仓库都已经在develop分支。相当于我们项目现有的开发环境。

1. 模拟单用户local、remote操作
2. 经过上面的的一系列操作，便可以进行本地的文件修改，并提交我们的修改。下面列出我们提交的常见操作命令。

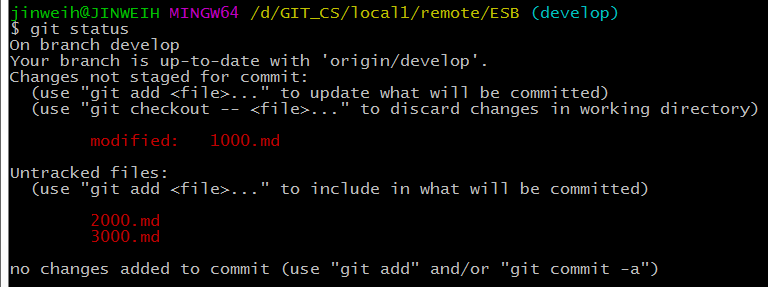


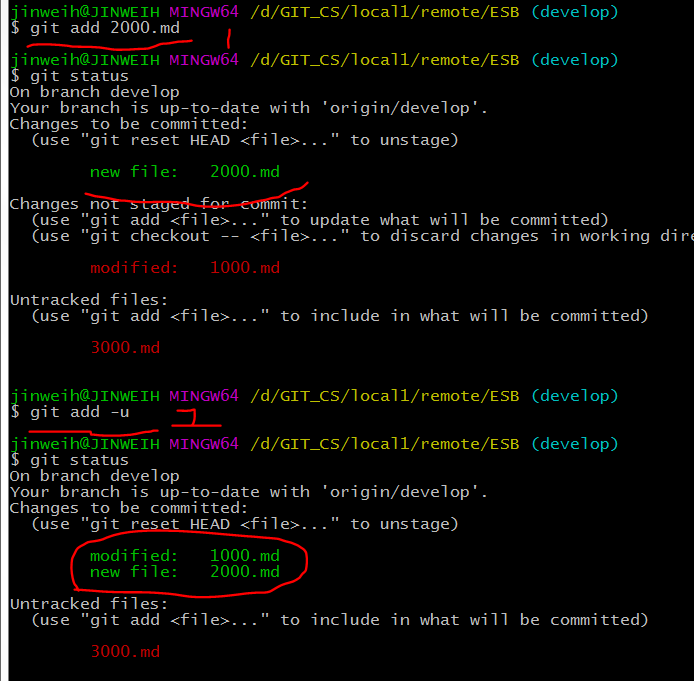
*修改文件：*修改文件之后使用git status，查看修改之后的状态，然后进行提交git commit –am “Message”。

提交之后使用git log，可以查看提交的历史记录。这时想要推送本地提交的更改使用git push,因为是我个人操作，所以不必进行git pull的操作。在推送远程仓库之前，需要使用git checkout develop将你的远程仓库检出到master分支，才能将本地的develop推送到远程develop分支。推送之后可以观察远程仓库此时没有你推送的本地的最新修改，需要将远程仓库使用git checkout develop分支，就可以在远程仓库查看你的推送。

*新增一个文件：*在本地仓库新增文件时，需要先将使用git add .文件加入版本控制。然后在进行提交。

*新增多个文件，修改一个文件：*当我在本地新增了多个文件，修改了一个或多个文件时，我要对新增文件实现部分提交。使用git add <path/filename>,可以看到2000.md已经add到暂存区，使用git add –u将忽略没有加入版本控制的文件。然后就可以进行提交。git commit –am “message”,使用git push就可以推送到远程仓库。查看本地仓库的3000.md还在本地未提交。

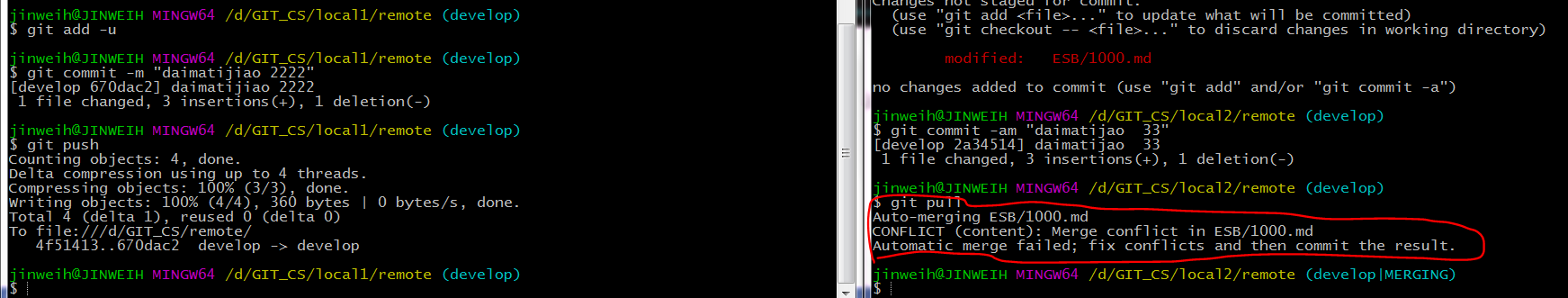




1. 模拟多用户local、remote操作

和前面的操作流程一样，我们在本地建立local2仓库。然后下载远程仓库的工程文件。可以看你的到local1仓库的提交文件。此时文件只有1000.md，然后我将local2检出新分支git checkout –b develop ,这时候再进行拉取，就可以看见2000.md

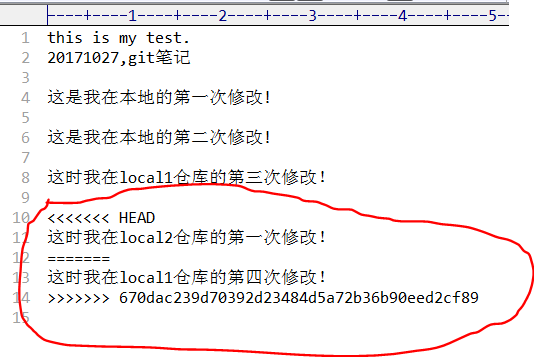
在local1进行修改1000.md，进行提交git commit，推送git push到远程分支。在local2进行修改1000.md，然后提交git commit，在进行git pull拉取操作，这时就会冲突。



在进行 git pull产生冲突

1. 多用户操作冲突解决

在local2 git pull产生冲突，这时候我们需要解决冲突文件。打开local2仓库的1000.md文件就可以看见冲突。我们进行手动解决。我进行选择留下我自己认为正确的部分代码。



1. 版本回退，穿越时空

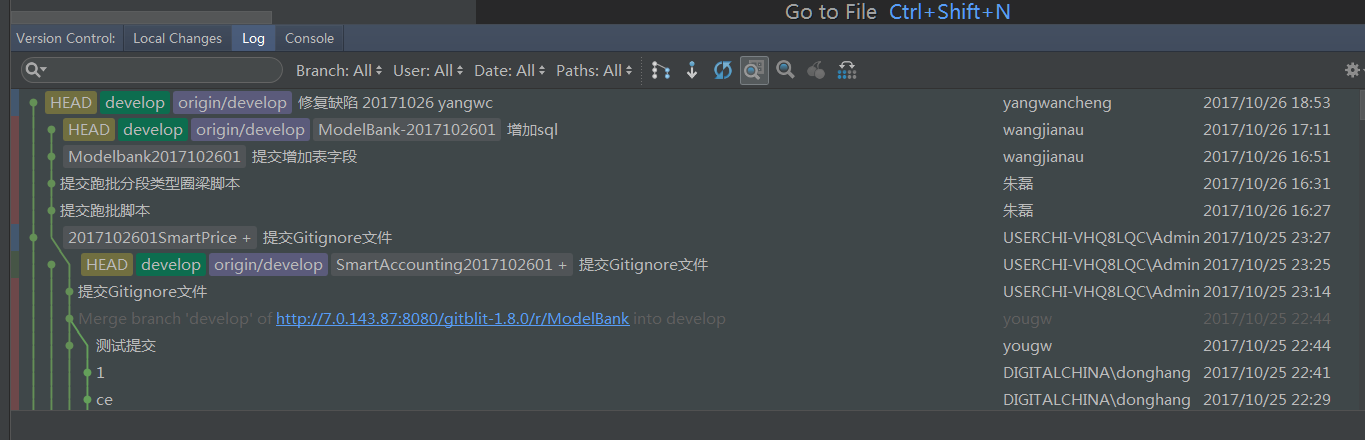
经常我们会遇到版本回退的问题，提交了一个修改，想要回去，可以使用git reset命令，这个操作最好谨慎，否则会删掉自己的代码，这个就不给大家介绍了。感兴趣可以查看git的说明，在git bash终端里面输入 git reset --help会有详细的说明。

1. Git自己的图形化界面

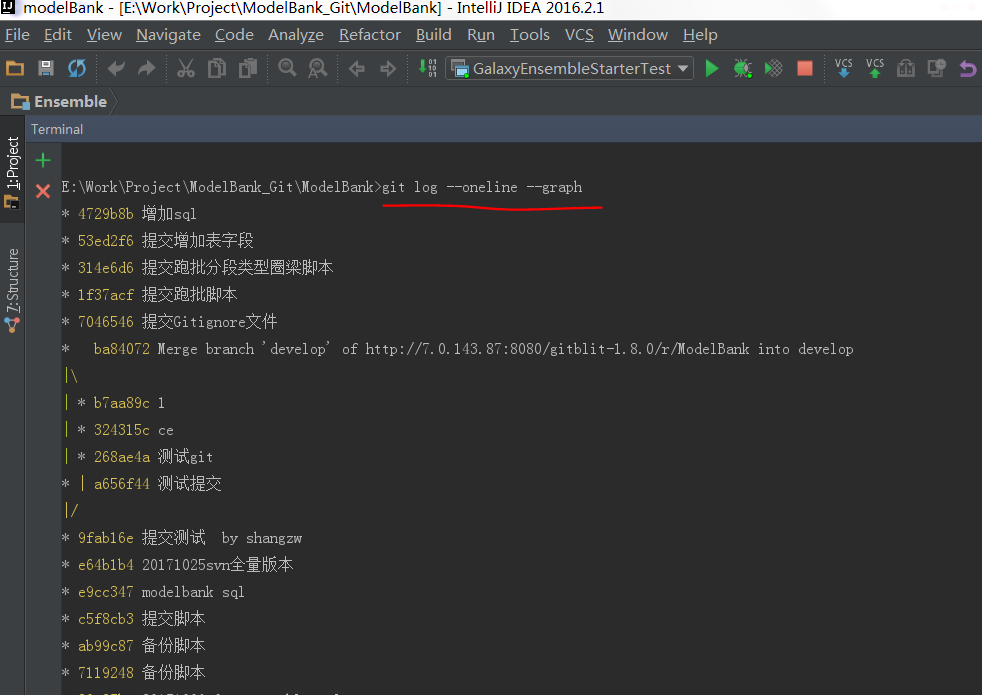
（1）在git终端输入gitk命令就可以看见。

（2）git终端的图形化日志提交信息，git log --oneline --graph

（3）配置git的环境变量，在核心IDEA里面可以直接进行代码更新，不需要切换到git终端。



核心git日志界面



核心cmd使用效果

1. 注意

1、

git pull操作是一个不安全的操作。它分为两部分 ：

git fetch和 git merge ，常见的标红部分就是有gitpull操作表现出来



2、对于暂时不需要提交的文件，可以先进行 git stash贮藏，代码更新之后使用git stash pop将代码还原.