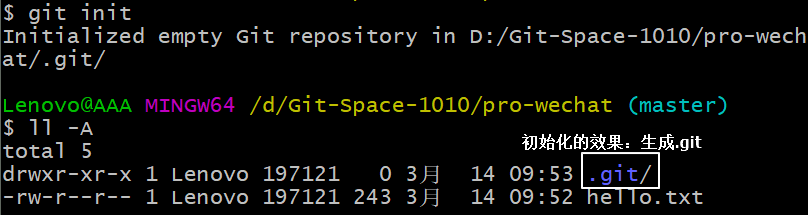
# Git的命令行操作

## 1.1 初始化本地库



## 设置用户签名

### 1.2.1 说明

签名的作用是区分不同操作者身份。用户的签名信息在每一个版本的提交信息中能够看到，以此确认本次提交是谁做的。

※注意：这里设置用户签名和将来登录GitHub（或其他代码托管中心）的账号没有任何关系。

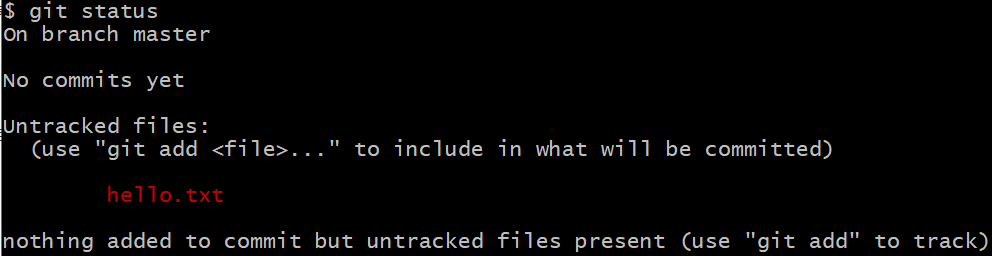
### 1.2.2 操作

全局范围的签名设置：

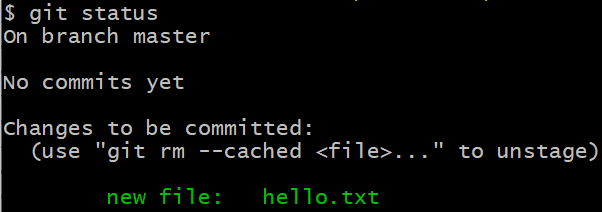


## 1.3 查看本地库状态

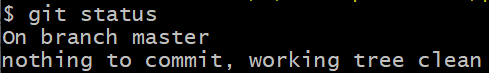
### 1.3.1 检测到未追踪的文件



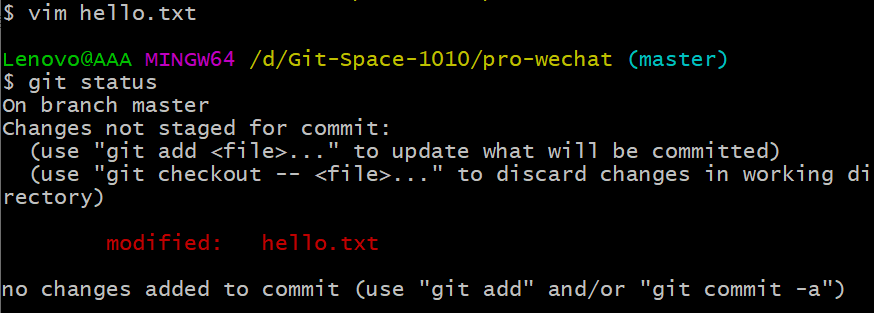
### 1.3.2 检测到有新建的文件添加到了暂存区



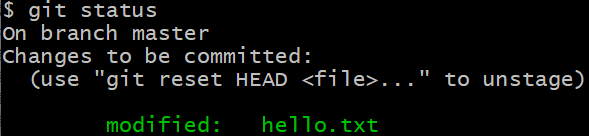
### 1.3.3 提交完成后查看状态



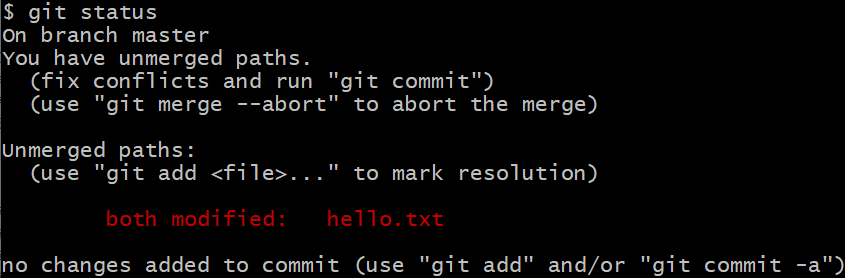
### 1.3.4 检测到工作区有文件被修改



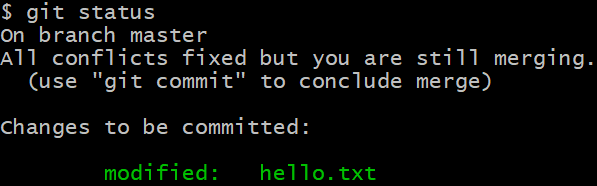
### 1.3.5 工作区的修改添加到了暂存区



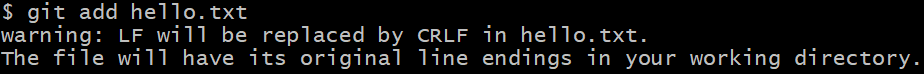
### 1.3.6 产生冲突



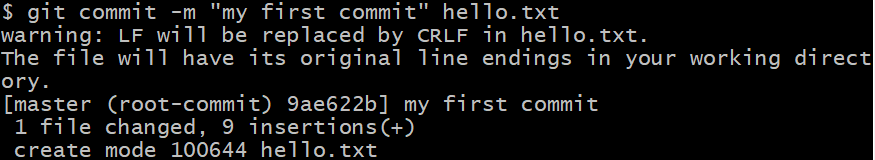
### 1.3.7 把冲突文件修复后添加到暂存区



## 1.4 将工作区的修改添加到暂存区

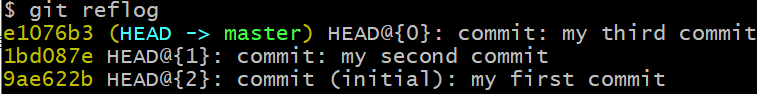


## 1.5 将暂存区的修改提交到本地库

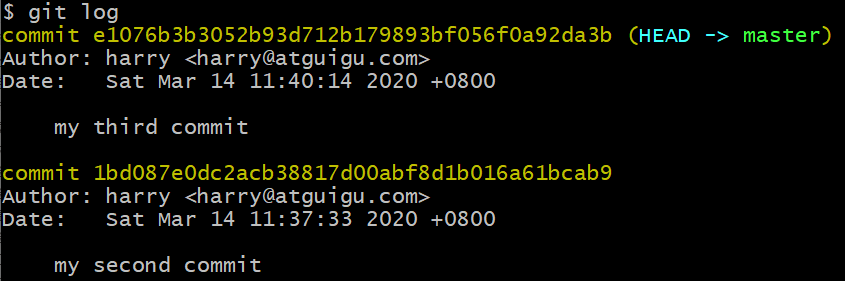


## 1.6 查看历史版本

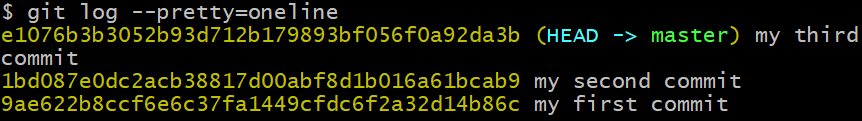
### 1.6.1 git reflog



### 1.6.2 git log

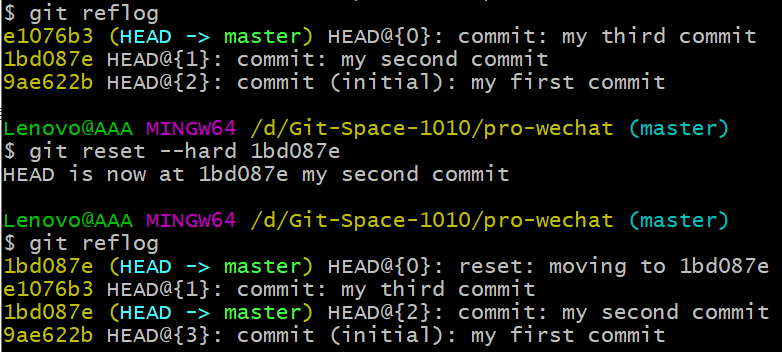


### 1.6.3 git log --pretty=oneline



## 1.7 版本穿梭

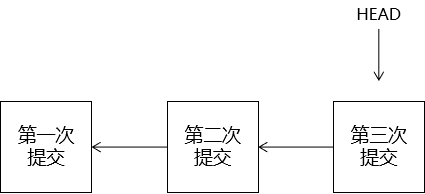
### 1.7.1 版本后退



### 1.7.2 版本前进

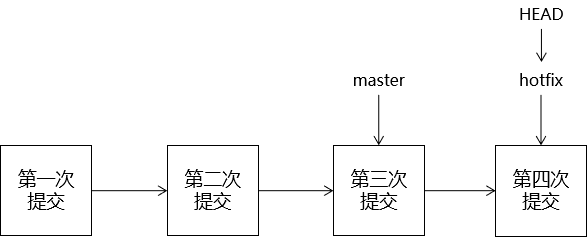


### 1.7.3 底层其实是移动HEAD指针



## 1.8 分支操作

### 1.8.1 创建分支和切换分支图解



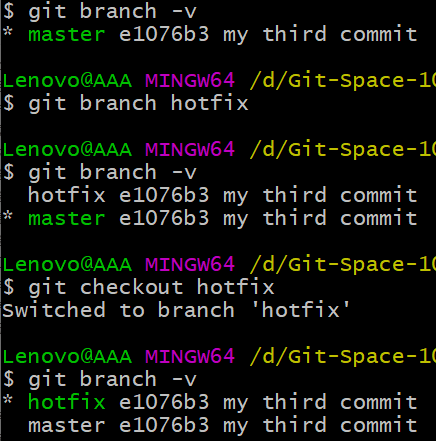
master、hotfix其实都是指向具体版本记录的指针。当前所在的分支，其实是由HEAD决定的。所以创建分支的本质就是多创建一个指针。

HEAD如果指向master，那么我们现在就在master分支上。

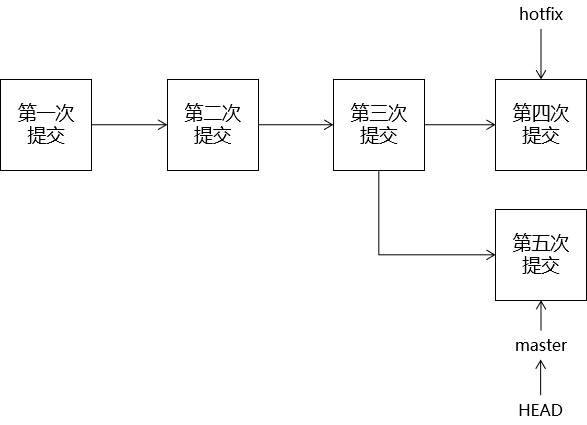
HEAD如果执行hotfix，那么我们现在就在hotfix分支上。

所以切换分支的本质就是移动HEAD指针。

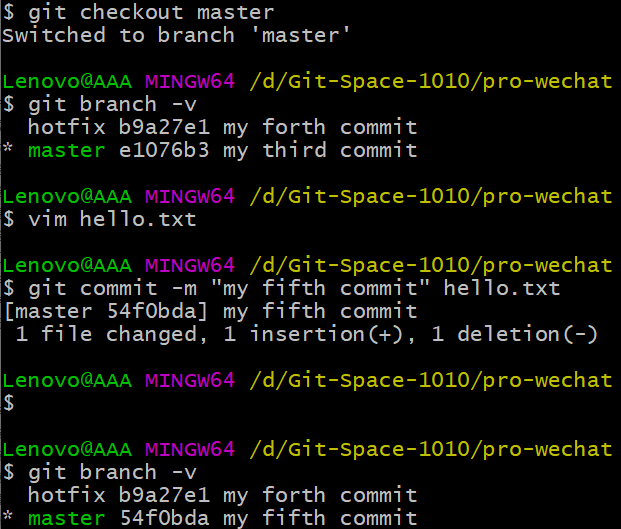
### 1.8.2 创建分支和切换分支的操作截图



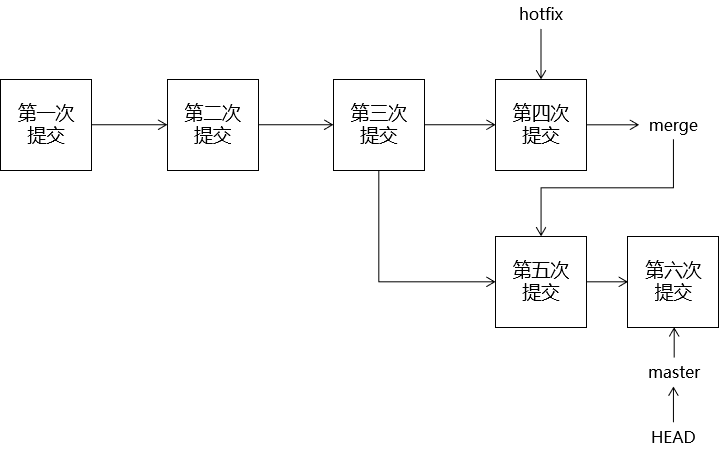
### 1.8.3 在两个不同分支分别做不同修改图解



### 1.8.4 在两个不同分支分别做不同修改操作截图



### 1.8.5 合并分支图解



合并分支时一定是涉及到两个分支。这两个分支一个是“当前所在分支”，一个是“目标分支”。

命令写法：git merge 目标分支

所以分支合并命令的本质就是把“目标分支”合并到“当前分支”。

例如：把hotfix合并到master

git merge hotfx

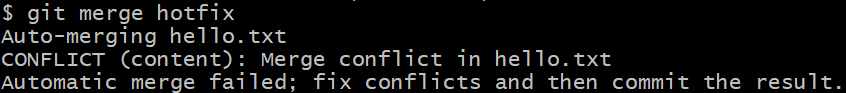
需要确保当前所在的分支是master

例如：把master合并到hotfix

git merge master

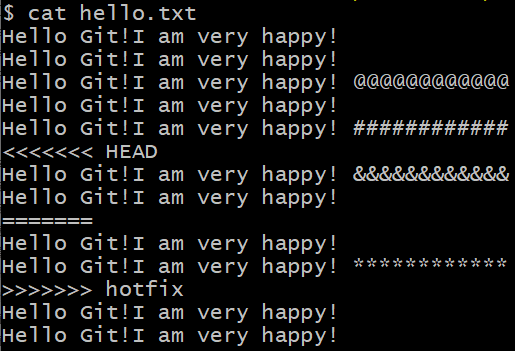
需要确保当前所在的分支是hotfix

### 1.8.6 合并分支操作截图



## 1.9 分支合并时产生冲突

### 1.9.1 冲突产生的表现



Git使用“<<<<<<<、=========、>>>>>>>>>>”符号帮我们标记出来，现在产生冲突的内容。

|  |
| --- |
| <<<<<<< HEAD  Hello Git!I am very happy! &&&&&&&&&&&&  Hello Git!I am very happy!  =======  表示HEAD指针指向的位置在冲突中的内容 |

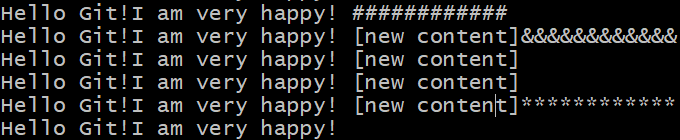
|  |
| --- |
| =======  Hello Git!I am very happy!  Hello Git!I am very happy! \*\*\*\*\*\*\*\*\*\*\*\*  >>>>>>> hotfix  表示hotfix指针指向的位置在冲突中的内容 |

### 1.9.2 冲突产生的原因

合并分支时，两个分支在**同一个文件的同一个位置**有两套完全不同的修改。Git无法替我们决定使用哪一个。必须人为决定新代码内容。

### 1.9.3 冲突的解决

①编辑有冲突的文件，删除特殊符号，决定要使用的内容



②添加到暂存区



③执行提交（注意：使用git commit命令时不能带文件名）

