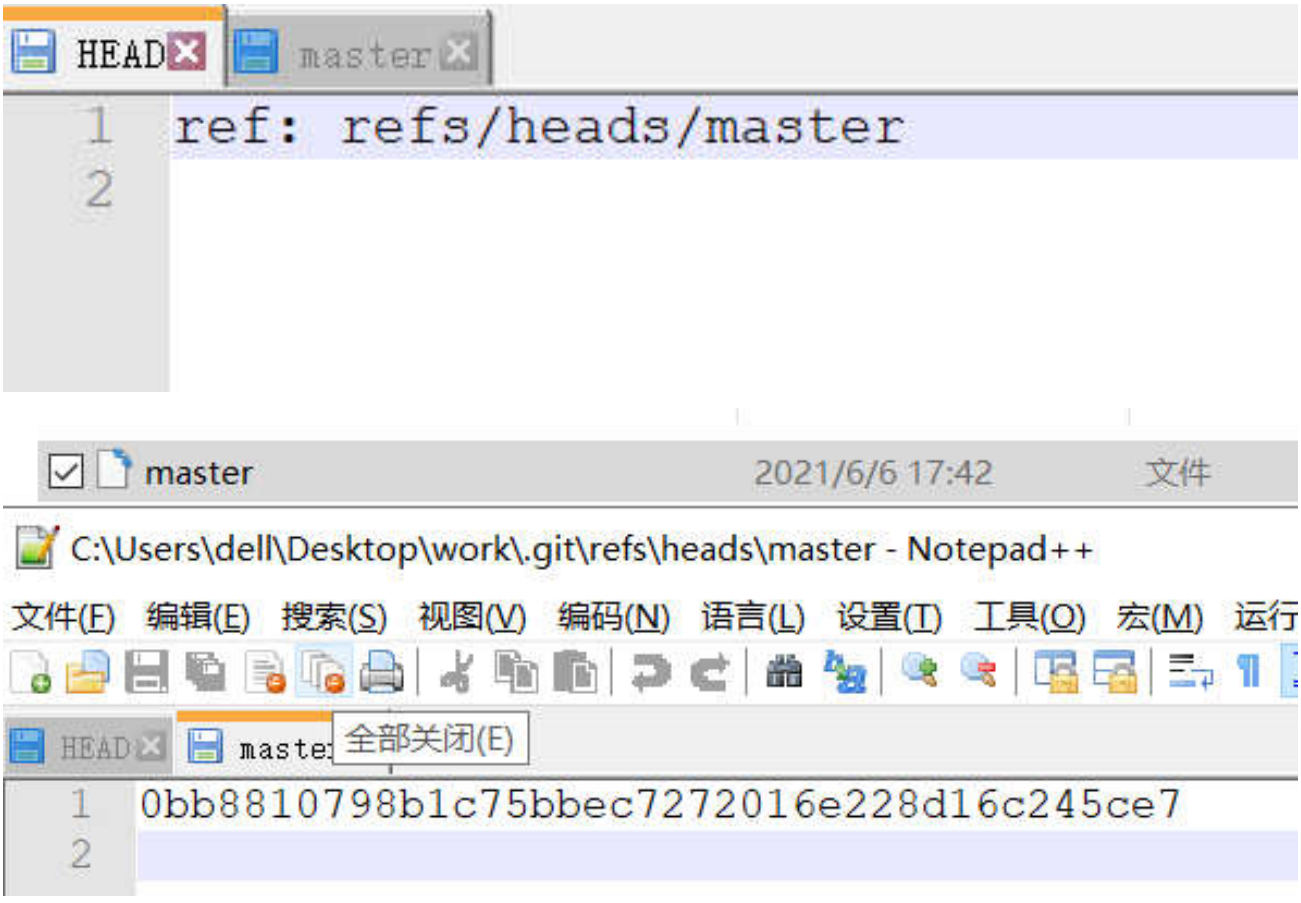
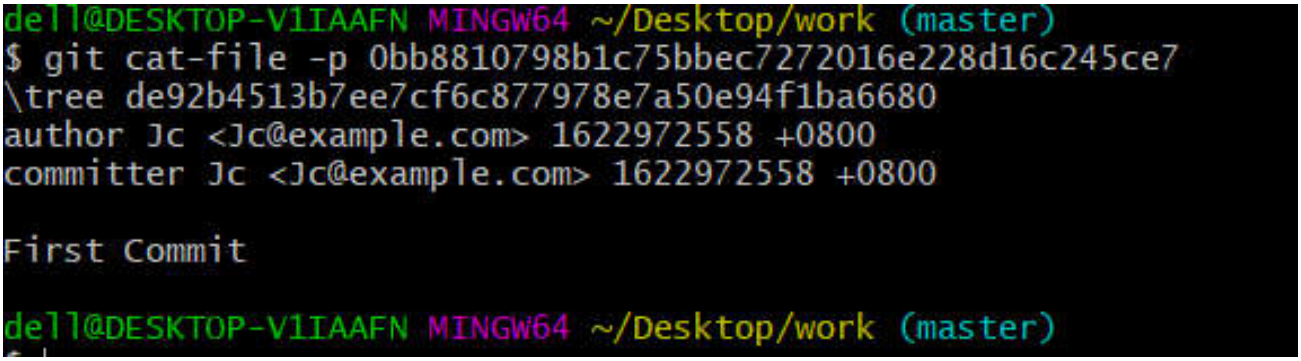
前言

前面讲解Git的底层和高层命令，对Git有了初步的认知对于Git来言最厉害就那肯定是分支功能啦，下面主要就是围绕**分支**进行讲解~小板凳准备好了吗？Let's go！

一、Git分支

书接上回，最终我们生成了提交对象，提交对象是用链表串起来的,Git默认会有一个分支(主分支)，这个分支的作用就是指向目前的提交对象。看过我Git学习笔记二的小伙伴，里面我说了.git目录下的内容但是对于分支的我说的很少，现在就重新温故一下关于Git分支的文件啦。**HEAD文件**：文件指出目前被检出的分支(目前处于那个分支)。**refs文件夹**：存储了指向数据的提交对象的分支(指针)。老环节了，直接上图~。



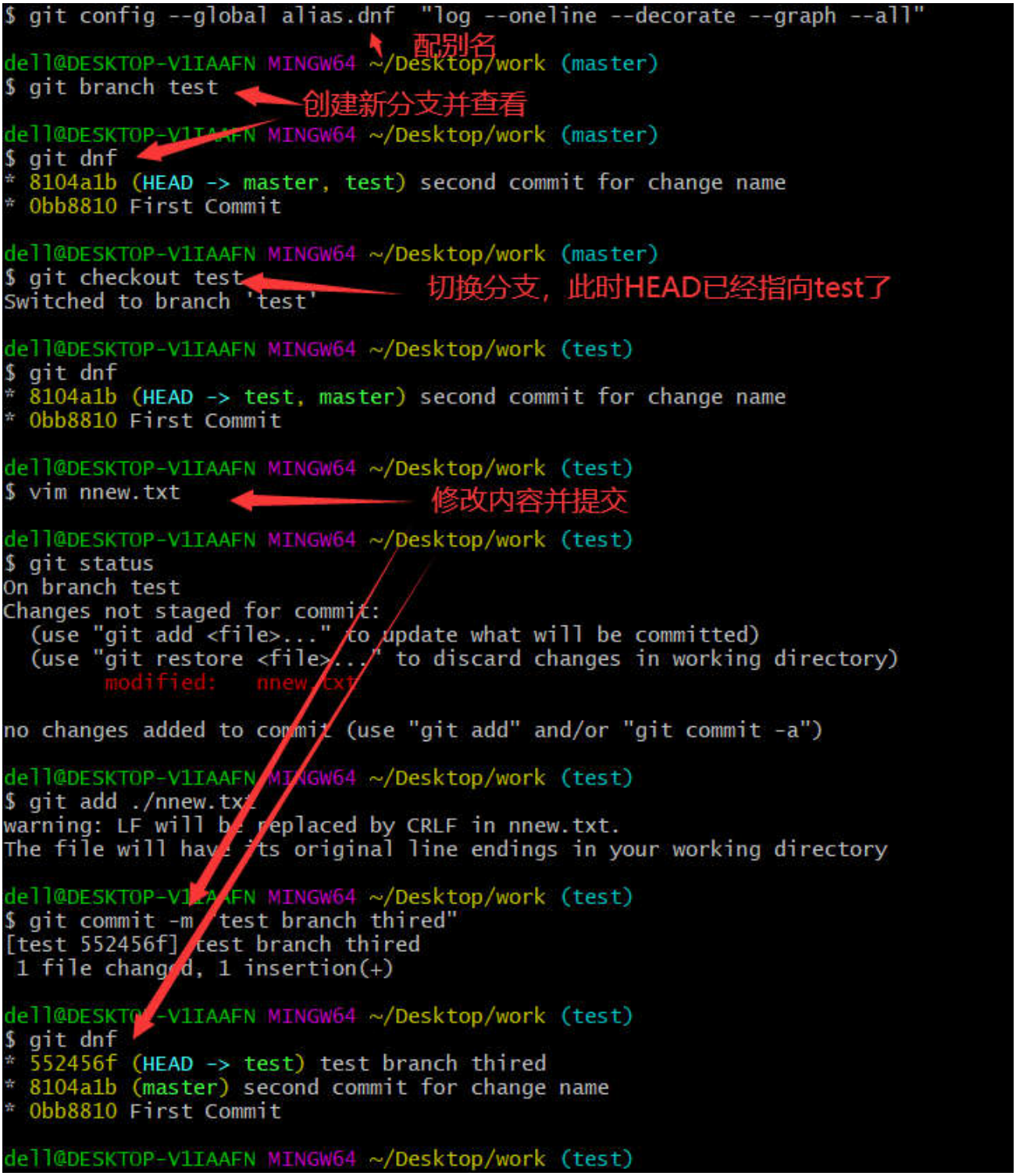


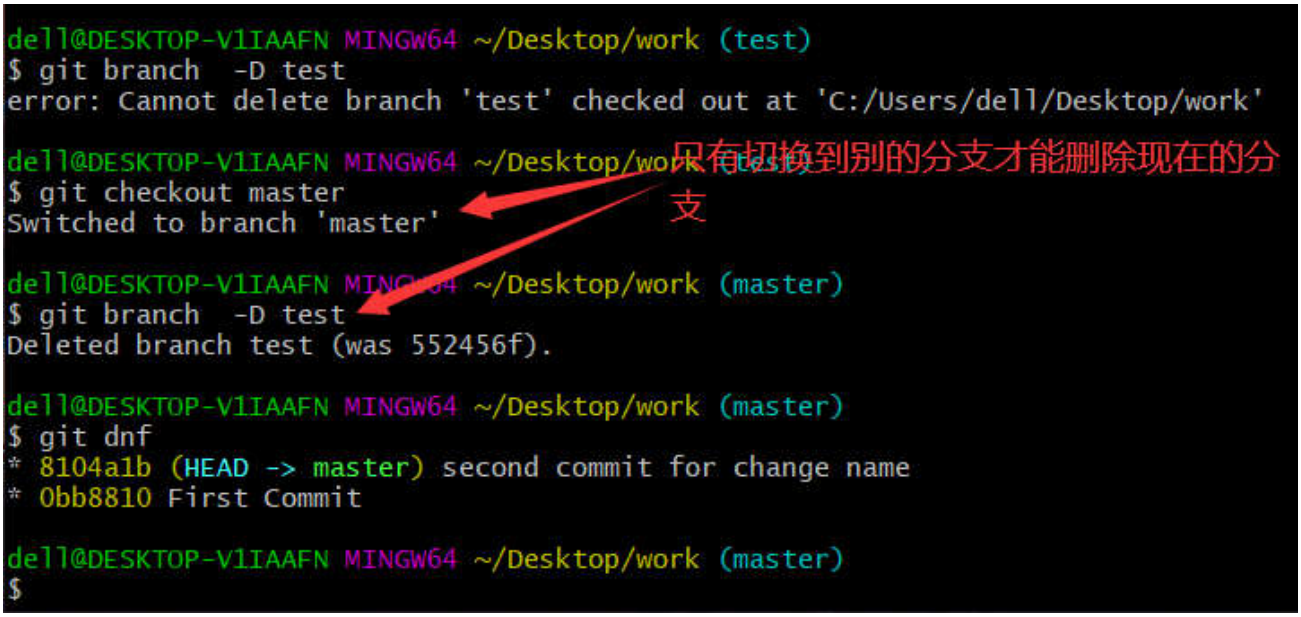
我们通过master可以或者里面存储的hash值，然后通过命令可以检测我们上面的说法，显而易见验证我上面说的。既然有主分支那就肯定有别的呀，带着我们乌黑的头发继续往下面看吧！

分支的高层命令

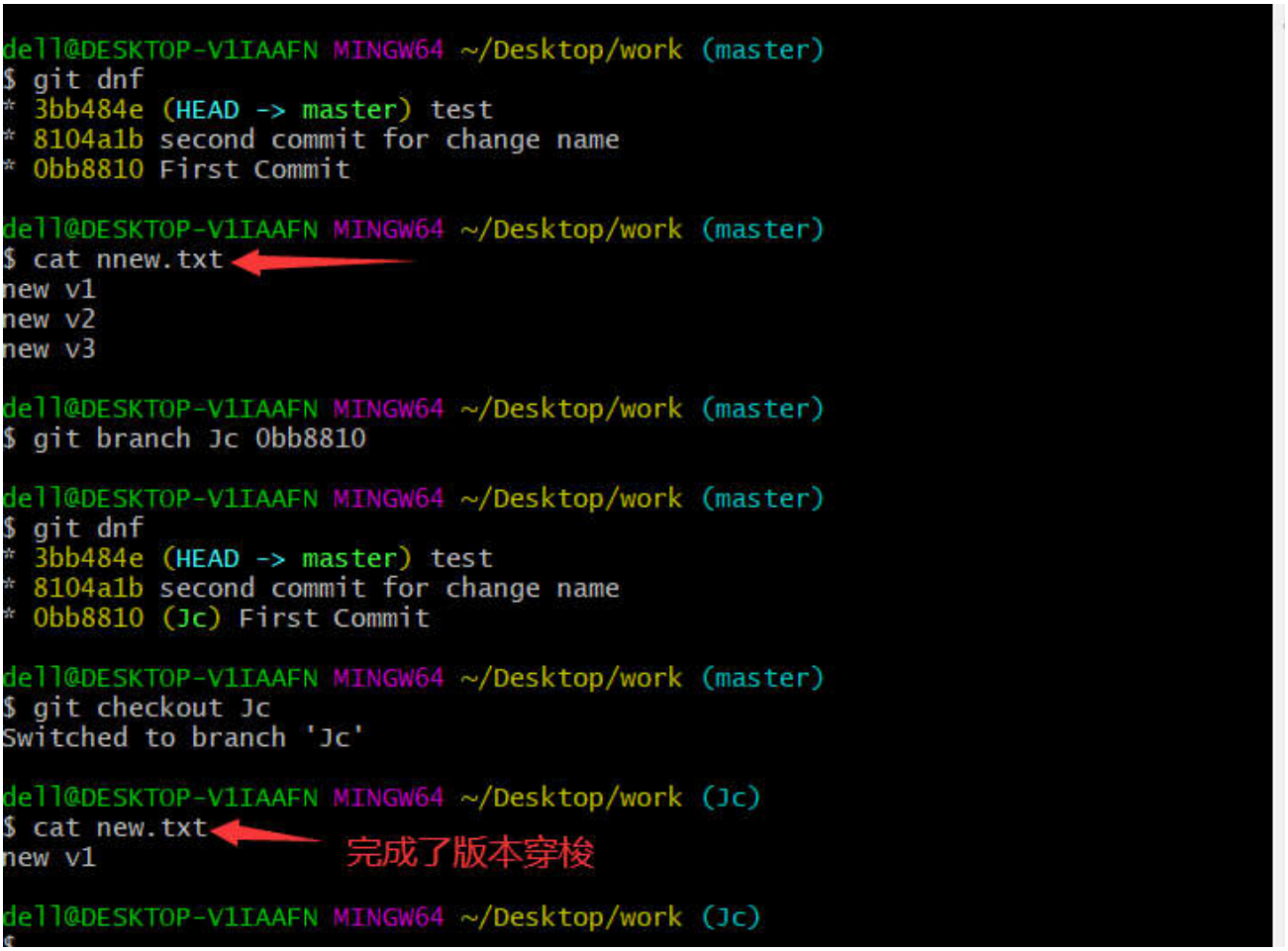
- 命令: git branch name
  作用: 创建一个可以移动新的分支(指针)。
- 命令: git config --global alias.别名 "命令"
  作用: 配别名。
- 命令: git branch -D name(强制删除用于未合并的分支),git branch -d name
  作用: 删除分支。
- 命令: git log --oneline --decorate --graph -all(显然合格名字太长了我们需要通过命令给起一个简单的名字)
  git config --global alias.dnf "log --oneline --decorate --graph -alli"(此时就可以用git dnf啦~)
  作用: 查看详细分支图
- 命令: git checkout name
  作用: 切换到名字为name的分支上面。
- 命令: git branch -D name(强制删除用于未合并的分支),git branch -d name
  作用: 删除分支。
- 使用场景：比如你先有一个项目运行的稳定，但是你突然脑壳一亮想到一个新的点子，我们肯定不能在原来的项目上面进行修改这个

时候我们就应该创建一个新分支并在这个新分支上面进行工作，如果你在新分支上面实现了自己的想法那么可以通过分支合并让它于项目融为一体，如果没有实现自己的想那就删除新分支会到原来的的分支就可以。话不多说，上图演示！(Git新手要自己尝试的哦！所以我也是新手哈哈哈)





- 命令：git branch commitHash
  作用：新建一个分支并使该分支指向对于的提交对象，这个命令很牛皮他可以用来版本穿梭,不过别忘了使用git checkout进行分支切换哦~上图演示。



Git分支总结： Git分支的本质就是指向提交对象的一个指针。分支名字其实就是给当前的提交对象命名。通过HEAD(活动指针)的指向进行分支切换版本穿梭。

二、 结尾

至此呢Git分支功能基础讲解完毕，下一篇文章肯定是Git分支的实践，学到了新知识那肯定要练一练手哈哈哈，如果感觉有用可以点个赞的哦！我会持续更新，如果有错误还请指出来,感谢观众老爷的赏脸。

若想获得上述内容的PDF版本移步到GitHub下载。

地址：Git 学习笔记专区。

-----缱绻