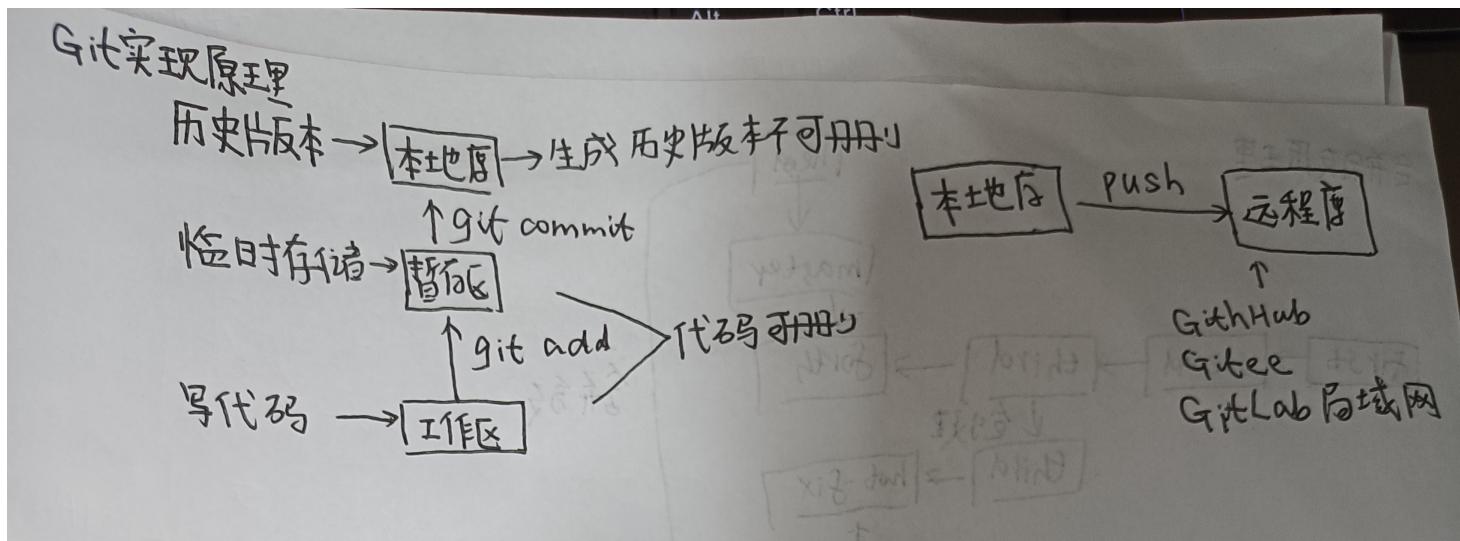


# Git的使用

## Git的工作原理

见图片：



## Git中的常见命令

命令名称	作用
git config --global user.name 用户名	设置用户签名
git config --global user.email 邮箱	设置用户签名
git init	初始化本地库
git status	查看本地库状态
git add 文件名	添加到暂存区
git commit -m "日志信息" 文件名	提交到本地库
git reflog	查看本地记录
git reset --hard 版本号	版本穿梭
git rm --cached 文件名	删除存放在暂存区的文件
git log	查看本地记录的详细信息

# Git的基本使用

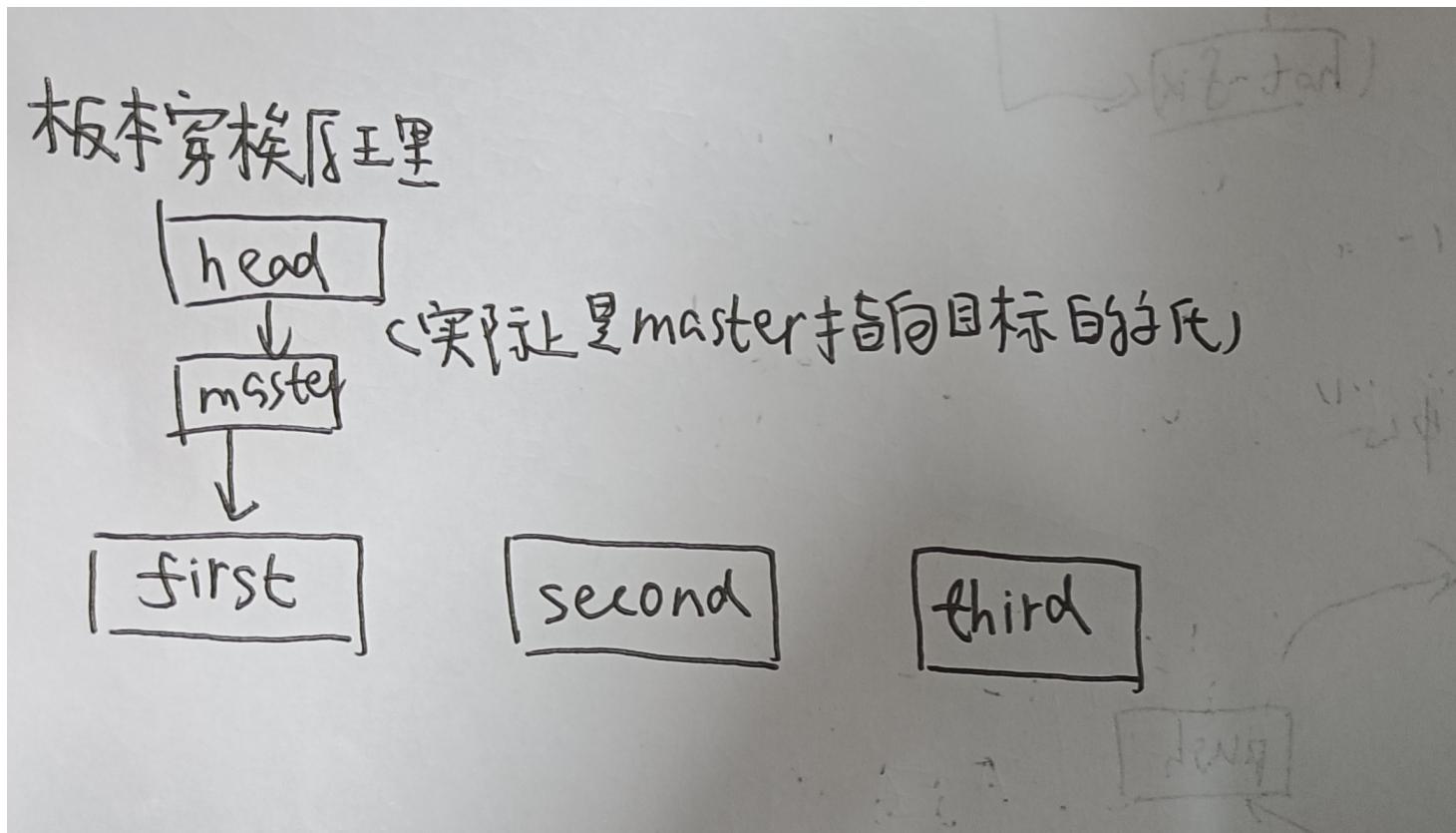
1. 编写工作区代码
2. 使用git add 文件名 命令添加工作区代码到暂存区中
3. 使用git commit -m "日志信息" 文件名 的命令去添加暂存区代码到本地库中

## 其他操作

1. 如果要删除暂存区中的代码，可以使用git rm --cached 文件名 的命令
2. 可以使用git status 命令去查看本地仓库的状态
3. 可以使用git reflog 命令去查看各个版本的详细信息，比如版本号等
4. 可以使用git log 命令去查看各个版本的详细信息，包含提交人和版本号
5. 可以使用git reset --hard 版本号 命令进行版本穿越

## 版本穿梭原理

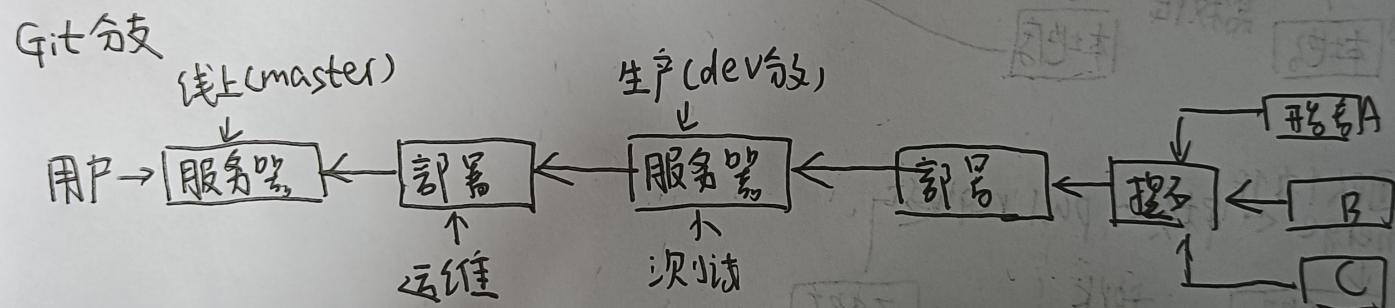
见图片：



## Git的分支

分支：即不同的开发任务不在一条线上进行，提高的开发的效率

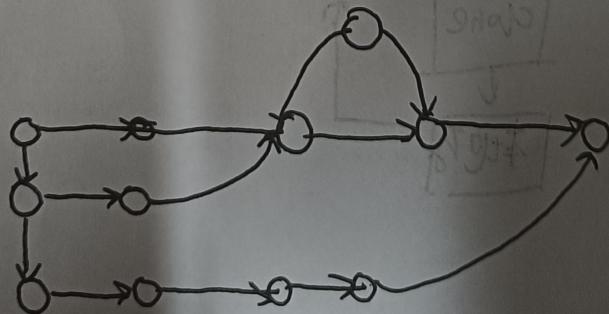
分支的结构如图：



如开发过程：

hot-fix

master  
feature-blue  
feature-game



## 对于分支的操作命令

命令名称	作用
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分支上

## 对于分支的其他注意事项

1. 可以使用git branch 分支名创建分支，可以使用git branch -v 命令查看当前分支
2. 不同分支下的工作步骤相同，都是工作区编写代码，提交到暂存区再提交到本地仓库，并且不同分支之间不会相互影响，各自完成自己的工作
3. 创建新的分支实际上就是复制原来的master分支，再进行自己的操作

4. 只有在用git merge 分支名命令合并分支时，不同分支的本地仓库内容才会合并，并且被合并的分支的内容进入主分支

## 分支冲突的解决

### 产生问题的原因

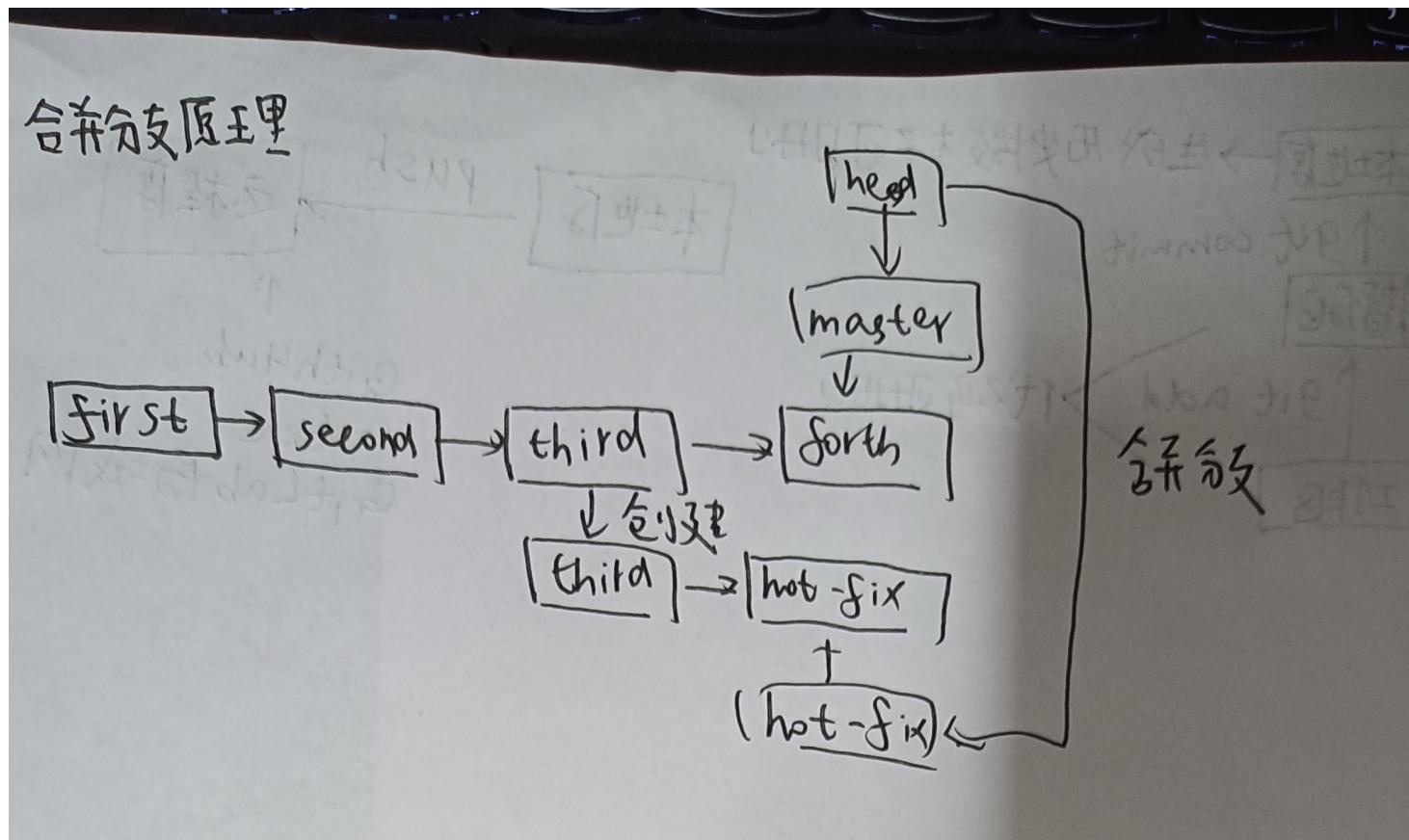
合并分支时两个分支在同一个位置进行不同的修改，Git无法决定哪一个对，需要人为决定

### 解决方法

在合并分支中查看文件内容，修改冲突的内容，修改冲突在重新提交代码，注意使用git commit -m "日志信息" 时之后不要带文件名

### 分支实现原理

如图：

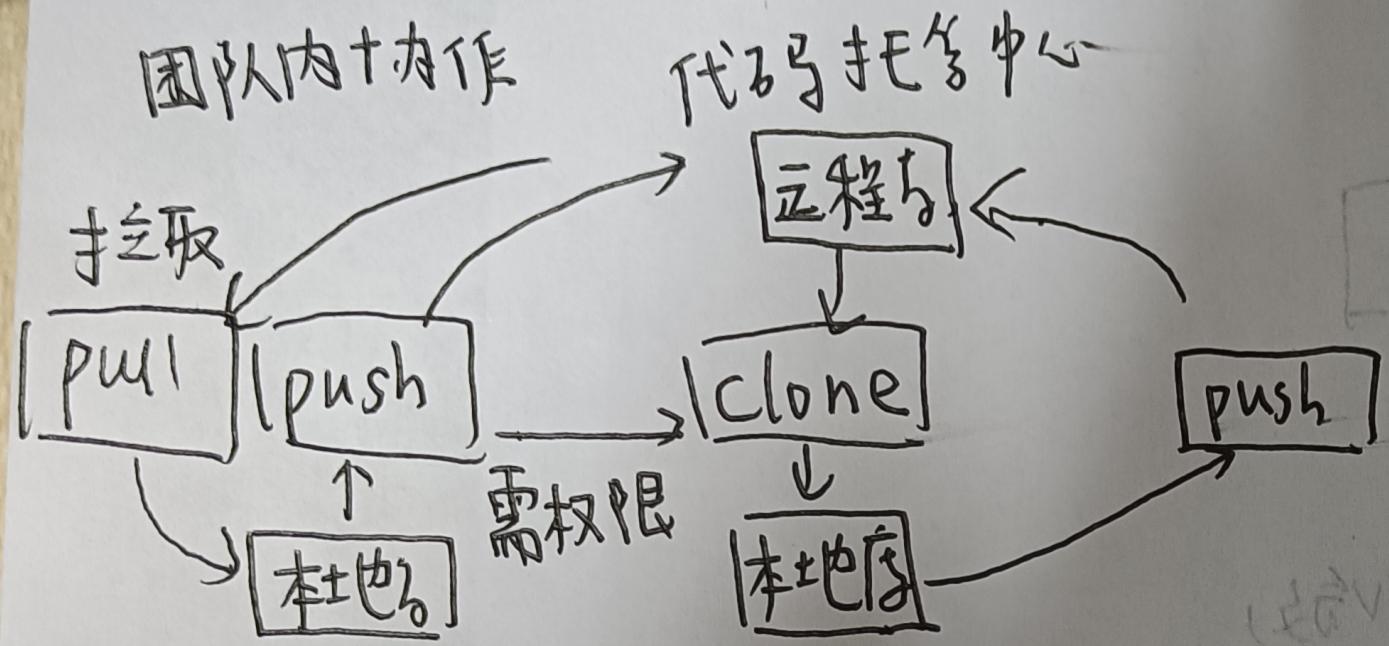


## 团队协作

### 团队内协作

如图：

## 团队协作机制



## 团队之间协作

如图：

跨团队协作审核 < pull request  
merge

