关于分支 branch 的命令

git branch 查看你自己创建的所有分支，分支前带*的代表当前所在分支

git branch -r 查看远程分支

git branch -a 查看所有分支

git branch branch-name 创建新的分支 branch-name

git branch -d branch-name 删除本地分支 branch-name （注：该分支需要已经合并过的分支才删的掉哦）

git branch -D branch-name 删除本地分支 branch-name （注：强制删除分支，无论合并与否都可删除）

git branch -m new-branch-name 将当前所在的分支更名为 new-branch-name

git branch -m old-branch-name new-branch-name 将旧分支名更名为新分支名

git checkout branch-name 将分支切换到 branch-name 分支

git checkout -b branch-name 创建并切换到分支 branch-name (注：此命令与执行 Git branch branch-name 和 Git checkout branch-name 两条命令等效)

关于查看信息的命令

git diff 查看工作区与暂存区的不同，在本地修改了一些文件且文件未加入到缓存，使用此命令可查看你修改了哪些文件

git diff --cached 查看暂存区与本地仓库的不同，在将修改的文件加入暂存区，即 add 之后，用此命令可以查看接下来要提交的内容

git diff HEAD 查看工作区与本地仓库的不同

git log 查看所在分支的所有提交的 log 信息

git log --committer liy 查看所在分支上所有由 liy 提交的 log 信息

git log -2 查看所在分支上最近两次提交的 log 信息

git log -p 此命令不仅可查看提交的 log 信息还能显示每条 commit 具体修改

了哪些地方
git log --grep=“ xxxx” 根据关键字过滤查看想要查看的 log 信息

git show 哈希值 每次 commit 之后都会生成一个对应的哈希值，git log 可以看到。此命令可以展示本次提交具体修改了哪些内容。(注：此命令很实用)

git show filename 可以显示该文件修改了哪些内容

git reflog 此命令可以查看该用户操作的所有动作包括提交信息，切换分支，合并分支，删除分支，分支更名等等，所有你的操作都会被记录下来

关于撤销、反悔的命令

git checkout -- filename 撤销对该文件的修改，要求该文件没有加入到暂存区即 add 之前撤销修改（注：此命令的两个短划线后有空格）

git checkout . 后面直接一点是对所有修改进行撤销，在执行这条命令之前先用 git status 查看一下状态哪些显示红色的文件是否都需要对他们的修改进行撤销，对于自己添加的文件只能手动删除

git reset HEAD -- filename 对于已经加入到暂存区的文件即 add 之后的文件，用此命令可使文件从暂存区撤销即文件由绿色暂存状态变成红色未暂存的状态

git reset 哈希值 对于已经提交了的文件想撤销该次的提交，可用此命令（注：后跟的哈希值不是本次提交的哈希值而是上一次的提交生成的哈希值），此命令之后查看状态这些提交的文件会变成红色未暂存的状态，回退两步

git reset --soft 哈希值 soft 这个参数只回退一步，使提交的文件回到暂存的绿色状态

git reset --hard 哈希值 hard 这个参数回退三步，彻底回退到上一个版本的内容，工作区也不保留修改，这个命令比较危险，撤销后就没有修改了，此命令慎用！！！！

git reset HEAD^ 回退到上一个版本，此命令与 git reset 哈希值是等效的命令

git reset HEAD^ filename 假设你的本次提交中是修改了几个文件，而你想撤销其中的一个文件，可使用此条命令将你想回退的文件回退到未暂存的红色状态

忽略对文件权限的改变

git config --global core.filemode false 有时在拉了分支或仓库后权限不够，我们会对文件 chmod ，在这个操作后用 git status 查看会发现它们都变成了红色修改的状态，这是我们不想看到的，用此命令进行配置，之后就可忽略仅修改权限的文件。
git config 不加参数表示仅对该仓库进行配置

git config --global 该参数表示对该用户的所有仓库进行配置

关于远程仓库和远程分支

git branch -r 在你需要拉服务器上别人推得分支时，你需要先查看一下远程分支的名字，用此命令可查看远程分支名

git checkout -b local-branch-name remote-branch-name 此命令是将远程分支拷贝到本地分支，local-branch-name 是自己命名，remote-branch-name 是远程分支名，它一般形式是 yuxin/branch-name 或者 origin/branch-name

git fetch yuxin 将服务器上更新的分支更新到本地可见，仅显示出来，不与本地的分支合并，只是查看更新

git merge remote-branch-name 将想要合并的某远程的一个分支合并到本地当前所在的分支上

git pull yuxin remote-branch-name:local-branch-name 将服务器上的远程分支直接拉到本地分支并与之合并，相当于执行了 fetch 和 merge 两个命令

git remote -v 查看远程仓库的地址

git remote add liy git@192.168.2.90:A20/lichee/linux-3.4.git 添加单独的远程仓库（注：liy 是一个远程仓库地址的简称，方便我们单独推分支时写分支所去仓库的地址用简称即可）

git push liy local-branch-name:remote-branch-name 往服务器上推单独一个仓库的分支，这里的 liy 就是上面自己添加的 linux-3.4 远程仓库的地址简称，你推上去后这个分支就在这个仓库下。(注：remote-branch-name 的名字要求芯片名_项目名命名)

repo remote add liying git@192.168.2.90:A23/lichee 如果要推这个 lichee 的代码，先为整个 lichee 添加仓库，执行这个命令后，用 git remote -v 查看会发现他为 lichee 下的各个仓库都添加了远程仓库的简称。

repo forall -c git push liying local-branch-name:remote-branch-name
将整个 lichee 代码推到服务器上 ，liying 是上面添加的远程仓库的地址简称

repo remote add liying git@192.168.2.90:A23 为整个 android 添加远程仓库，推整个 android 的命令与推 lichee 相同，整个工程推上去时间花费很比较久。如果只修改了几个仓库，建议只推修改的仓库下的分支。

git remote rename oldname newname 将远程仓库重命名
git remote rm name 删除添加的远程仓库 name

常用的 repo 命令封装

repo start branch-name --all 为整个 android 或者 lichee 创建并切换到此分支上。

repo forall -c git branch branch-name 此命令也是为整个 android 或 lichee 创建分支，但是它与上一条命令有个很大的区别，repo start 创建的分支是以公版分支创建的代码，而本命令是以当前所在分支为基础创建分支。

repo checkout branch-name 对整个 android 或 lichee 进行操作，切换分支

repo branch 查看所有的分支都在哪些目录下

repo abandon branch-name 删除在 android 或 lichee 下创建的分支。

repo status 查看工作区的状态

其他的 git 命令如果需要对整个 android 或者 lichee 进行操作，只需要在 git 命令前加上
repo forall -c
例如 repo forall -c git log

注 ：要经常用 git status 查看工作区的状态，尤其在切换分支时，一定要保证工作区是干净的再切换分支。