git 使用记录

1. 生成密匙：

    ssh-keygen -t rsa 几次回车，在.ssh/ 下的 id_rsa.pub 即为密匙文件；

2. 获取 git 分支：

    A. 远程获取：(路径须绝对路径)

        git clone git@192.168.1.50:/home/git/repositories/A20-Android4_2.git .

        (注意在最后加个点是为了避免 clone 时在当前目录下新建一个 git 目录)

    B. 本地获取：(路径为相对路径即可)

        git clone git@192.168.1.50:A20-Android4_2.git .

    C. 查看所获取分支的路径：

        git remote -v

    D. 切换到某个分支：

        git checkout $branchname

3. 分支的新建与删除：

    A. 新建：

        git branch $branchname (注意在哪个分支上执行就是基于哪个分支新建)

        git push origin $branchname (推到服务器仓库)

    B. 删除：

        git branch -D $branchname (删除本地的分支)

        git branch -rd origin/$branchname (删除服务器仓库分支)

        git push origin :$branchname (注意冒号)

        git remote prune origin (同步远端已删除分支)

4. 修改内容查看及提交：

A. 查看未提交的修改：

git status/git status . （查看修改的文件）

git diff/git diff . （查看修改的内容）

B. 查看已提交修改：

git log （查看提交信息）

git whatchanged （查看每个提交修改的文件）

git diff $2 $1 （查看莫个"提交 ID" $1 的修改内容）

C. 还原被修改文件：

git checkout -f */$fileanme

D. 提交修改：

git add */$filename （将新建文件加入仓库）

git commit */$filename -m "***" /git commit -a -m "" （提交修改/提交当前所有修改，删除一个文件也可以）

git push origin $branchname

E. 还原到某个提交 ID 前：

git reset “$提交 ID” （注意避免冲突：如果本地有修改过即将还原的文件，可以先备份 it，然后 git checkout -f $it）

git push origin $branchname --force

git pull origin $branchname

F. 提取同一仓库不同分支的修改：

git cherry-pick “$提交 ID”

G. 解决冲突：

git diff .

vi */$冲突文件 （注意去掉 ----> HEAD 之类的冲突提示）

git add */$冲突文件

git commit -c "产生冲突的$提交 ID"

git push origin $branchname

H. 清除当前所有修改：

git checkout -f && git clean -df

5. 忽略一些不需要管理的文件/类型：

vi .gitignore

6. 一些常见的异常处理：

A. 分支在别处又提交，本地提交不了：

git pull origin $branchname 然后再提交；

B. 新建仓库时，遇到漏提交文件：

find . -name ".git*" | xargs rm -rf {}

7. 打 git patch：

patch -p1 < $patchfile.patch （用过可以，但是不能新建.a/.so 等库文件）

或者

git apply $patchfile