在本地git仓库中，可以通过命令观察到远程Gerrit维护的Git库中的review任务,merge后的任务

git ls-remote origin

git log --name-status branch --not tag --no-merges > merged.txt

git log --name-status 1L88\_DEV --not MV601\_B0\_01.15\_20140306 --no-merges > merged.txt

git log --name-status ?INT80A --not INT80B\_Main\_B0\_01.14\_20140314 --no-merges > merged.txt

git log v2.5.. Makefile fs/ （v2.5以后的log，也可以是commit）

git log -p 显示详细修改的 patch 具体内容

git log --stat 显示那些文件修改了 分别添加和修改了多少行内容

git log --name-status branch --not tag --no-merges > merged.txt

git log --name-status 1L88\_DEV --not MV601\_Main\_B1\_02.28\_20140708 --no-merges > merged.txt

这个命令会显示文件中每一行的作者，最后一次改动后进行的提交(commit)以及该次提交的时间戳。

git blame [file\_name]

git merge FETCH\_HEAD

清楚本地状态信息

git clean -df

git checkout ./

在工作库中可以添加. gitignore文件，让git忽略某些文件

checkout到branchA，再用rebase branchB，等于将A嫁接到B上，注意这里是从A与B最开始出现分歧的commit开始一整枝的接上去。而cherry-pick方向相反，checkout到branchA，输入cherry-pick branchB的结果是仅仅将B所在点的commit内容嫁接到A上。

在Gerrit里点击“publish and submit”提示如下：

Gerrit Code Review

Your change could not be merged due to a path conflict.

Please merge (or rebase) the change locally and upload the resolution for review.

cd ~/projects/pan #切换到pan项目

git branch ＃查看分支情况

git checkout master #选择分支

git fetch origin #fetch与pull的区别，自己再搜吧～

git rebase origin/xxx // git pull --rebase #查看有“CONFLICT (content): ”的地方，手工解决冲突后，下一步

git add dev/controller/web/index.php #这只是一个举例，即要先add操作

git rebase --continue

git push origin HEAD:refs/for/xxx #OK了

git rebase -i HEAD~4 squash

cd ~

curl https://raw.github.com/git/git/master/contrib/completion/git-completion.bash -o ~/.git-completion.bash

然后，在您的~/.bash\_profile文件中添加以下代码

if [ -f ~/.git-completion.bash ]; then

. ~/.git-completion.bash

fi

git diff -w xxx

git log --graph --oneline --all

git fsck --lost-found

ssh -vv -p 29418 qsq@192.168.5.21（测试账号是否可用）

cppcheck --xml-version=2 --enable=all --errorlist . > cppcheck-result.xml

\*\*/cppcheck-result.xml

git show HEAD 当前的commit

git revert HEAD(撤销修改) 但是会产生新的commit

gerrit只接受注释标签。这是相当简单的创建和推动一个annotated tag：

git checkout v2.0

git tag -am "Adding v2.0.1 tag" v2.0.1

git push origin INT80B\_Main\_B0\_01.07\_20140226 HEAD:refs/heads/INT80B\_Main\_B0\_01.07\_20140226

git push origin --tags

git push origin HEAD:refs/for/1L88\_DEV

git config remote.origin.push refs/heads/\*:refs/for/\*

scp -p -P 29418 [account@192.168.5.21:hooks/commit-msg .git/hooks/](mailto:account@192.168.5.21:hooks/commit-msg%20.git/hooks/)

git stash list

git stash pop

git stash apply stash@{x}

git stash apply --index stash{0}（被已经drop的stash）

git format-patch -s HEAD~3(打补丁)

cat \*.patch | git am (应用补丁)

git apply xxxx

git am user-mail-archive

一般遇见错误加参数 -3

git am -3 user-mail-archive

git error: unpack failed: error Missing tree 解压错误，可能是下载的时候索引错误，重新下载

git reflog 查询

git reset --hard

fatal: Path 'packages/apps/Shield/AndroidManifest.xml' is in submodule 'packages/apps/Shield'文件夹错误 重新删除添加

ERROR: In commit 0c9eeaba4c2f8ed64394f1701e78a94c4fd3484e

ERROR: author email address chenxinke@ict.ac.cn

ERROR: does not match your user account.

ERROR: The following addresses are currently registered:

ERROR: wanghongmei@loongson.cn

ERROR: To ssh://wanghongmei\_99@172.17.109.72:29418/pmon-loongson.git

! [remote rejected] HEAD -> ref/for/master (invalid author)

出现此错误需要在

gerrit

权限配置中管理员在gerrit选择

For ge Committer Identity

下面添加registered users

git diff > my.patch

patch -p1 < my.patch

git diff --no-prefix > my.patch(不想在文件名前面增加前缀，添加--no-prefix参数)

patch -p0 < my.patch

gerrit 邮件设置

enable = true

smtpServer = smtp.exmail.qq.com

smtpServerPort = 465

smtpUser = qi.shanqing@bluebank.com.cn

smtpEncryption = ssl

smtpPass = q372232

sslVerify = true

from = qi.shanqing@bluebank.com.cn

[gui]

encoding=utf-8

[i18n]

commitencoding=utf-8

logoutputencoding=utf-8

sysv-rc-conf

svnserve -d --listen-port 3691 -r /home/svnuser/test

repo init -u ssh://liuguanghui@github.marvell.com:29418/platform/manifest.git -b pxa988\_jb4.2.beta3-sp5-01 --repo-url ssh://liuguanghui@github.marvell.com:29418/tools/repo.git