

使用 **Git & Repo** 下载代码

客户端安装 Git

安装 git, gitk

网络连接正常的情况下：

```
$ sudo apt-get install git-core gitk git-gui
```

不能上网，有.deb安装包的，请执行：

```
$ sudo dpkg -i XXXX.deb
```

初次运行 Git 前的配置

配置用户信息

```
$ git config --global user.name yourname  
$ git config --global user.email "youremail"
```

配置文本编辑器

```
$ git config --global core.editor vim
```

详细内容请看[Git 安装配置](#)

安装 openssh

上传代码到 gerrit 上，需要安装openssh client，并将本地的 SSH Public Keys 添加到 gerrit 上，gerrit 通过 SSH Public Keys 验证上传权限

安装 openssh client

```
$ sudo apt-get install openssh-client
```

检查 ~/.ssh 目录有没有，没有的话用 **ssh-keygen** 生成。ssh是管理和转换认证的密钥，包括 RSA 和 DSA 两种密钥

生成 rsa 密钥

用下面的命令生成**ssh**私公钥：

```
$ ssh-keygen -t rsa -C "youremail"
```

(最好加上youremail)一路回车,不用输密码,在 ~/.ssh 目录会生成 id_rsa, id_rsa.pub 两个文件

请将 ~/.ssh/id_rsa.pub 文件以邮件附件的形式发给 yuhua.wang@sim.com和y.liu@sim.com

拷贝 ~/.ssh/id_rsa.pub 的内容,在 **Gerrit** 中添加公钥:

首先登入到 http://172.16.6.72:8080 [http://172.16.6.72:8080], 点击右上角的 sign in, 输入用户名, 密码 (Windows登陆账号和密码)。

进入 gerrit 界面之后,点击 admin -> settings (在右上角), 选择左侧 SSH Public Keys 选项, 添加公钥

公钥添加完成后, 测试能否连接到 **gerrit** 服务器:

```
$ ssh -p 29418 yourname@172.16.6.72
```

如果提示 publickey denied, 说明生成的 key 有问题, 请重新生成, 然后重启 ubuntu 机器。

安装 Repo

安装 Repo

下载代码

初始化版本库

先在本地建立一个空的工程目录

例: myandroid

```
$ mkdir myandroid  
$ cd myandroid
```

用 **repo init** 命令初始化:

用 repo init 来下载 manifest Git 库, 具体命令如下:

下载 S55 通用项目请用下面的分支:

```
$ repo init -u git@192.167.100.232:platform/manifest.git -b QRD8926_KK --repo-url=git@192.167.100.232:repo.git
```

下载过程中会出现让你输入姓名和 email, 请一定要输入正确的用户名和 email (执行了git config 的话, 这个地方会自动出现你的名字和mail)

最后会看到 repo initialized in android 这样的提示, 说明本地的版本库已经初始化完毕

同步版本库

使用命令:

```
$ repo sync
```

将整个 android 代码同步到本地

同步单个或选定的**projects**:

```
$ repo sync project1 project2...
```

开发步骤

建立个人feature分支, 修改代码, 解决冲突, 上传分支到 Gerrit

基本流程

1. 用 `repo start branch -all` 创建个人分支
2. 修改文件
3. `git add file`
4. 用 `git commit` 提交更改
5. 用 `repo upload` 把本地修改的分支上传到 Gerrit server

创建个人分支

创建个人分支方法一: (应用所有仓库)

```
$ repo start yourbranch --all
```

`--all` 参数表示应用于所有project, 可以创建出此分支

创建单个project分支

```
$ repo start yourbranch projectlist1
```

projectlist1 表示你创建的分支只应用projectlist1项目

创建个人分支方法二: (只适用于当前仓库, 不能用 **repo upload** 上传 **git branch** 创建的分支名, 所以建议大家不要使用这个命令创建分支)

```
$ git branch yourbranch
```

分支应用技巧:

查看分支:

```
$ git branch
```

可显示当前版本库所有存在的分支, 带*号的分支表示当前定位在哪个分支上

切换分支:

```
$ git checkout yourbranch
```

提示: switched to branch 'yourbranch', HEAD已经指向 yourbranch

修改代码

开发人员可以开始修改代码

修改好之后，可以用下面两个命令来查看你修改文件内容的差异：

```
$ repo status  
$ git diff
```

提交到本地 git 仓库

代码确认修改没问题，可以提交到本地 **git** 仓库：

```
$ git add file
```

file: 需要上传到服务器的文件名，此处也可使用 git-gui 工具来操作

```
$ git commit -m "注释"
```

此处的注释一定要添加,撤销修改使用 git reset

上传代码前同步版本库

同步版本库：

```
$ repo sync
```

同步版本库中的改动到本地仓库，如遇到冲突，代码更新失败，到失败的库中做如下操作：

```
$ git pull --rebase
```

如有冲突，修改冲突后：

```
$ git add 冲突文件  
$ git commit  
$ git rebase --continue
```

上传代码

使用命令：

```
$ repo upload
```

使用命令 repo upload 把本地的分支上传 gerrit 进行 code review, gerrit 系统会自动创建一个 chang-id

如果只修改了某个**projectlist**下的几个文件，就可以直接对这个**project**提交：

```
$ repo upload projectlist
```

上传成功提示：

*[new branch] yourbranch→refs/for/psh100_414007/yourbranch

说明已经上传到 **gerrit** 中，等待 **code review**

如果修改了好几个**projectlist**，请用下面的命令提交：

```
$repo upload
```

弹出 **vim** 编辑窗口界面，找到要上传仓库的分支名，将前面的 **#** 删除，按**esc**键，输入:wq保存退出，即可提交所有项目

如果上传到**gerrit**，发现修改的有问题，但又不想创建新的 **change-id**，可以用如下方式：

保证你本地的分支还是之前**checkout**的那个分支，修改完成之后用：

```
$ git commit --amend -m "注释"  
$ repo upload
```

上传成功，该 **change-id** 会多了一个 **patch set**

关于**repo**的补充：

```
sudo apt-get install phablet-tools
```

E:/xampp/backup/dokuwiki/data/pages/git/git_install_use.txt · 最后更改: 2016/09/12 04:02 由 qiancheng