**服务器**

添加用户：

1、root登录后，点击左上角菜单-管理员-左侧用户

图形用户界面, 应用程序

描述已自动生成

2、点击新用户，输入用户信息

电脑屏幕截图

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

3、设置账户密码，重新登陆账户重新设置密码

**提交代码**

服务器代码存放于10.127.109.3:/home/ubuntu/git-code/

第一步： 建立本地仓库

sudo git init



第二步： 设置用户名及邮箱

sudo git config --global user.email "1144097453@qq.com"

sudo git config --global user.name "Leo.liu"



第三步：建立远程仓库关联

sudo git remote add [自定义远程仓库名] [远程仓库ssh地址]

sudo git remote add origin <http://10.127.109.3:9800/Leo.liu/openwrt-ipq6000.git>

如果输出fatal: remote origin already exists.则

sudo git remote rm origin



Sudo git pull [远端仓库名] [远端分支名]

sudo git pull origin main --allow-unrelated-histories

文本

描述已自动生成

git fetch origin tag v2.2.0

查看是否关联

sudo git remote -v



如关联错误或已有关联

sudo git remote rm origin 执行删除关联



第四步：把文件夹加入暂存区并提交

sudo git add .

（如果需要忽略.gitignore的作用，需要添加-f强制提交文件至暂存区）

sudo git add -f .



sudo git commit -m " SE1100/SE1000-V2.3.5-emmc-dl"



第五步：把本地仓库分支关联到上游远程仓库

查看本地仓库文件

sudo git ls-files

查看分支

sudo git branch -vv



创建tag

sudo git tag -a V2.3.0-ucmd -m 'v2.3.0-ucmd'

查看tag

sudo git tag

要显示附注信息,我们需要用 show 指令来查看

sudo git show V1.2

但是目前这个标签仅仅是提交到了本地git仓库.如何同步到远程代码库

sudo git push origin --tags

如果刚刚同步上去,你缺发现一个致命bug ,需要重新打版本,现在还为时不晚.

git tag -d V1.2

到这一步我们只是删除了本地 V1.2的版本,可是线上V1.2的版本还是存在,如何办?这时我们可以推送的空的同名版本到线下,达到删除线上版本的目标:

git push origin :refs/tags/V1.2

如何获取远程版本?

git fetch origin tag V1.2

查看远程仓库分支

图形用户界面, 应用程序

描述已自动生成

git push <远程主机名> <本地分支名>:<远程分支名>

sudo git push -u origin master:main



**成员：**

克隆仓库

git clone --branch v2.2.0 <http://127.0.0.1:9800/root/R-Dtest.git>

文本

描述已自动生成

1、分支操作：

使用分支意味着你可以把你的工作从开发主线上分离开来，以免影响开发主线。

查看当前所有分支

sudo git branch -a

**第一步：创建分支**

sudo git branch [分支名称]

sudo git branch testing

如果要切换到一个已存在的分支，你需要使用 git checkout 命令。我们现在切换到新创建的 testing 分支去：

sudo git checkout testing

这样 HEAD 就指向 testing 分支了

**第二步：合并分支**

当我们在 testing 分支上工作并形成了一些列提交后，可以切换回 master 分支，然后运行 git merge命令：

sudo git checkout master

sudo git merge testing

有时候合并操作不会如此顺利。如果你在两个不同的分支中，对同一个文件的同一个部分进行了不同的修改，Git 就没法干净的合并它们。此时会出现Automatic merge failed; fix conflicts and then commit the result.，

文本

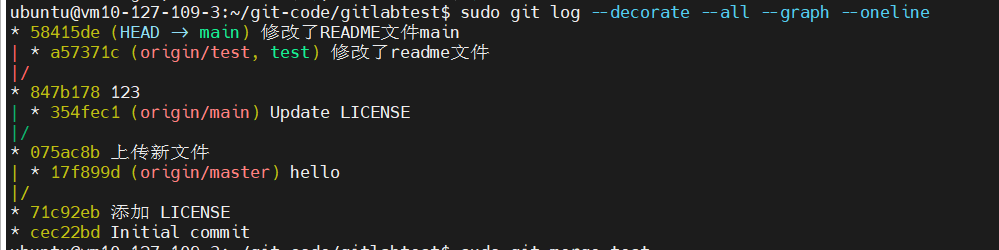
描述已自动生成

你可以在合并冲突后的任意时刻使用 git status 命令来查看那些因包含合并冲突而处于未合并（unmerged）状态的文件：

文本

描述已自动生成

sudo git log --decorate --all --graph --oneline



也可使用sudo git mergetool来处理冲突

日程表

描述已自动生成

跳转到下一个diff点：使用]c命令

跳转到上一个diff点：使用[c命令

如果希望在两个文件之间来回跳转，可以用下列命令序列：Ctrl-w,w

在比较和合并告一段落之后，可以用下列命令对两个文件同时进行操作。比如同时退出：

qa

如果希望保存全部文件：wa

或者是两者的合并命令，保存全部文件，然后退出：wqa

撤销指令：Ese+u