

分支创建

```
git branch br_name
```

分支切换

```
git checkout br_name
```

分支合并

```
git merge br_name    #将br_name合并到当前分支
```

分支列表

```
git branch           #本地分支
```

```
git branch -a        #所有分支
```

分支删除

```
1 | git branch -d br_name
```

查看分支所属

```
1 | git reflog --date=local | grep 分支名
```

查找两个分支的共同祖先

```
1 | git merge-base v1.11.2_a v1.11.4
```

自动同步分支

```
1 | git fetch --prune
```

git删除远程分支后本地同步

```
1 | git remote update origin --prune //更新远程分支列表
2 | git branch -D {branch_name}      //删除本地未同步到远程的分支
```

分支规范

1.git主分支(master)。它是自动建立，用于发布重大版本更新

2.git开发主分支(develop)。日常开发在此分支上进行

3.git临时性分支：用于应对一些特定目的的版本开发(验证OK后，应该删除此分支)，主要有：

- 功能（feature）分支：它是为了开发某种特定功能，从Develop分支上面分出来的。开发完成后，要再并入Develop。可以采用feature-的形式命名。

- 预发布 (release) 分支：指发布正式版本之前（即合并到Master分支之前），我们可能需要有一个预发布的版本进行测试。预发布分支是从Develop分支上面分出来的，预发布结束以后，必须合并进Develop和Master分支。它的命名，可以采用release-的形式
- 修补bug (hotfix) 分支：软件正式发布以后，难免会出现bug。这时就需要创建一个分支，进行bug修补。修补bug分支是从Master分支上面分出来的。修补结束以后，再合并进Master和Develop分支。它的命名，可以采用hotfix-***的形式。

注意事项：一个分支尽量开发一个功能模块，不要多个功能模块在一个分支上开发。feature 分支在申请合并之前，最好是先 pull 一下 develop 主分支下来，看一下有没有冲突，如果有就先解决冲突后再申请合并。

适配openssl 3.0.9