[**git只clone仓库中指定子目录和指定文件的实现20151020**](http://blog.csdn.net/xuyaqun/article/details/49275477)

标签： [git](http://www.csdn.net/tag/git)[子目录](http://www.csdn.net/tag/%e5%ad%90%e7%9b%ae%e5%bd%95)[svn](http://www.csdn.net/tag/svn)[代码](http://www.csdn.net/tag/%e4%bb%a3%e7%a0%81)

2015-10-20 15:16 11986人阅读 [评论](http://blog.csdn.net/xuyaqun/article/details/49275477#comments)(5) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/xuyaqun/article/details/49275477#report)

 分类：

技术文档（31）

版权声明：本文为博主原创文章，未经博主允许不得转载。

从svn转git也有四五个月的时间了，期间也遇到过一些问题，但也深感git的强大，用者自知，这里就不在多言，git目前唯一不能实现的是：不能像svn那样，针对子目录设置权限，这与git分布式仓库的运行机制有关，svn是基于文件方式的集中存储，Git却是基于元数据方式分布式存储文件信息的，它会在每一次Clone的时候将所有信息都取回到本地，即相当于在你的机器上生成一个克隆版的版本库，既然本地有了完整的版本库，肯定就有所有权限了，所以也就没办法针对子目录的进行权限控制了。

今天说的问题和上边有点关系，理解了上边的内容，这个问题也就简单了：我们想做的是只拉取一个repository中的几个子目录的代码，而非整个库，从上文的说明中也能看出这是不能实现的，对，在git 1.7.0 之前是不能实现的，git认为如果这样做的话，仓库的数据一致性无法保证，即使你真的这样做，完全可以把这些不相关联的子目录放到不同的repository，repository之间是彼此独立的，仔细想想也很有道理。

我的使用场景：

1、想用一颗repository树来保存相互之间没有关联、没有依赖的运维子项目，而每个子项目代码量都很少，每一个子项创建一个repository太没有必要了；

2、公司的所有内部api也想统一放置到一颗repository树上，几十个api不能都创建一个repository吧（我现在是这么认为的，这个需求也可能不太合理）。

如果非要只clone repository中的几个子目录的话，那就用sparse clone，git从1.7.0开始支持，sparse clone也只是一个变通的方法：先拿到整个repository的object等元数据信息，然后在本地加一个叫.git/info/sparse-checkout的文件（即黑名单、白名单，支持正则，参见下文具体操作命令）来控制pull那些目录和文件（类似.gitignore文件，都是本地的概念），变通的实现《git只clone仓库中指定子目录和文件》，如果非要完美的满足这个需求那就用svn吧。

引用stackoverflow上对sparse clone的描述：

Implementing something like this in Git would be a substantial effort and it would mean that the integrity of the clientside repository could no longer be guaranteed. If you are interested, search for discussions on "sparse clone" and "sparse fetch" on the git mailinglist.

In general, the consensus in the Git community is that if you have several directories that are always checked out independently, then these are really two different projects and should live in two different repositories. You can glue them back together using [Git Submodules](http://git-scm.com/book/en/v2/Git-Tools-Submodules).

具体做法：

一、svn的实现：svn因为是基于文件的集中控制方式，所有“原生”就支持只checkout指定子目录，并且还能很好的对子目录进行权限控制

➜  svn-test  svn co http://xxx.xxxx.com/ops/内网服务器情况   test  
A    test/内网机器硬件配置详细  
A    test/内网机器硬件配置详细/192.168.1.147.txt  
A    test/最新全公司网络拓扑图.png  
Checked out revision 251.  
➜  svn-test  
➜  svn-test  svn info  
Path: .  
Working Copy Root Path: /Users/laijingli/svn-test  
URL: http://xxx.xxxx.com/ops/%E8%BF%90%E7%BB%B4%E6%96%87%E6%A1%A3  
Repository Root: http://xxx.xxxx.com/ops  
Repository UUID: 5773cb3d-14e2-48da-bdf0-37bd7e579499  
Revision: 251  
Node Kind: directory  
Schedule: normal

二、git的实现：基于sparse clone变通方法

[root@vm\_test backup]# mkdir devops  
[root@vm\_test backup]# cd devops/  
[root@vm\_test devops]# git init    #初始化空库  
Initialized empty Git repository in /backup/devops/.git/  
[root@vm\_test devops]# git remote add -f origin http://laijingli@192.168.1.1:90/scm/beeper/yunxxx\_ops.git   #拉取remote的all objects信息  
Updating origin  
remote: Counting objects: 70, done.  
remote: Compressing objects: 100% (66/66), done.  
remote: Total 70 (delta 15), reused 0 (delta 0)  
Unpacking objects: 100% (70/70), done.  
From http://192.168.1.1:90/scm/beeper/yunxxx\_ops  
 \* [new branch]      master     -> origin/master  
[root@vm\_test devops]# git config core.sparsecheckout true   #开启sparse clone  
[root@vm\_test devops]# echo "devops" >> .git/info/sparse-checkout   #设置需要pull的目录，\*表示所有，!表示匹配相反的  
[root@vm\_test devops]# more .git/info/sparse-checkout  
devops  
[root@vm\_test devops]# git pull origin master  #更新  
From http://192.168.1.1:90/scm/beeper/yunxxx\_ops  
 \* branch            master     -> FETCH\_HEAD  
[root@vm\_test devops]# ls  
devops  
[root@vm\_test devops]# cd devops/  
[root@vm\_test devops]# ls  
monitor\_in\_web  test.1

截图：



很赞的几篇参考文章（作为后辈，你遇到的很多问题前辈们早已遇到，并且很多已经有了完美的解决方案，做技术一定要勤于google啊）：

http://stackoverflow.com/questions/600079/is-there-any-way-to-clone-a-git-repositorys-sub-directory-only

http://jasonkarns.com/blog/subdirectory-checkouts-with-git-sparse-checkout/

http://schacon.github.io/git/git-read-tree.html#\_sparse\_checkout

http://www.tuicool.com/articles/QjEvQvr