**git版本管理使用手册**

马兴 2011.7.6

git是一个分布式的版本管理软件，所谓分布式指没有服务器端，每一个开发者都拥有完整的代码版本历史，包括从第一个版本到最后更新的版本。这样做的好处是，就是可以从任何一个代码库中获取拥有完整版本历史的代码。

git的版本库包含三个部分：remote(有时和origin相同) copy， local copy，working copy。他们之间的关系如图所示：



图 1 git三个副本关系图

# git常用命令

1. git clone url 将url提供的git库同步到本地
2. git commit -a 提交修改到本地git库
3. git add . 添加当前目录及子目录中所有没被git管理的文件到git库中
4. git mv a b 将文件a重命名为文件b
5. git push 将本地git库同步到服务器
6. git branch b1 新开分支b1
7. git branch -d b1 删除分支b1
8. git pull 将服务器git库同步到本地
9. git merge b1 合并b1分支的代码
10. git checkout b1 切换到b1分支
11. git checkout file 将file从本地git库中提取出来(比如误删了某个文件，可以重新从本地库中提取出来)
12. git status 查看当前状态
13. git stash 隐藏当前修改，回到checkout时的状态(当修改了一部分内容，想尝试另一种方法，又不想放弃当前修改，可以先隐藏当前修改)
14. git apply 应用已经隐藏的修改
15. git help command 查看某条命令的帮助，比如要看stash命令的详细用法：git help stash

# 使用git开发流程

约定：

服务器git库url：ssh://git@saintnoah.org

本地工作目录：/home/work/noah/

流程：

1. 先从服务器同步代码到本地工作目录：cd /home/work/noah/ && git clone ssh://git@saintnoah.org
2. 提倡分支开发：
   1. 先创建分支(分支名字随意，这里用dev)：git branch dev
   2. 切换到dev分支：git checkout dev
   3. 开发
      1. 想尝试用另外一种方法，又不想放弃当前的修改：git stash(相当于回到了2.2时的状态)
      2. 发现此种方法不行，打算用原先的方法：git apply(相当于回到了2.3.1使用git stash命令之前的状态)
   4. 提交修改：git commit -a
   5. 重复2.2和2.3步骤，直到某个小功能点完成，编译通过
3. 将修改同步到服务器之前，先看看别人有没有将修改同步到服务器了
   1. 切换到master：git checkout master
   2. 同步服务器版本到本地：git pull
   3. 如果其他人有修改，需先合并其他人的修改
      1. 切换到dev: git checkout dev
      2. 合并修改：git merge master
      3. 如果有冲突，则用git mergetool进行merge(注意不要将合并过程中产生的.orig等文件添加到库中)，然后再commit一次
      4. 切换回master: git checkout master
   4. 将修改同步到服务器：git push
4. 从2.2开始进行下一个功能点的开发。或者从2.1开始重新开一个分支，然后开发。

# git常用工具

* tortoiseGit：windows环境下很方便的一个工具，和svn类似。
* qgit: linux环境下方便查看git日志的工具。
* kdiff3: 第三方图形化merge工具，linux下用起来相对还行，但是还不如tortoiseGit提供的merge工具好用。

# git基本配置

Linux配置文件：~/.gitconfig

windows：git安装目录，或者用tortoiseGit进行配置

附上我的.gitconfig：

[user]

name=maxing

email=xm.crazyboy@gmail.com

[gui]

encoding=utf-8

[merge]

tool=kdiff3

[il8n]

commitEncoding=utf-8

logOutputEncoding=utf-8

# 注意事项

1. 使用git add .命令添加文件时注意不要将不需要的文件添加进来
2. 需要给文件重命名时不要直接mv，用git mv
3. 不小心误删文件时，小心操作不要同步到服务器，以免引起其他人的回滚。

# noah团队git使用规范

1. 文件名不能包含中文字符，包含中文字符会在不同环境下出现乱码显现，网上提供的解决方案不能很好的解决这个问题，或者有人能解决此问题欢迎提供方案~
2. 文件字符编码统一为utf-8
3. 尽量不提交doc等文件到git库中
4. 同步到服务器的代码必须是编译通过的
5. 提倡小修改，多提交
6. 每次commit必须要有short message，大概说明此次修改的内容即可。
7. 使用了新的第三方库时需及时通报其他成员，同时说明如何安装与使用，放置其他成员编译通不过
8. 欢迎补充~