1.3 git命令使用说明

2018年9月14日

9:01

**1、git的常用6个命令**

Remote 
pull 
checkout 
fetch/clone 
Repository 
Index 
push 
commit 
workspace 
add 
• Workspace: 
• Index/ Stage: 
Repository: 
• Remote: 

**2、git bash创建工程**

<http://www.cnblogs.com/chenwolong/p/GIT.html>

（1）github上创建一个库，即repository

（2）git bash定位到存储repository的路径下，通过git init创建本地仓库

（3）通过命令：git clone <https://github.com/KevinLiao888/aris_luoshi.git> 将github上面的库下载到**本地仓库**（或者采用git remote add [shortname] [url]，git pull origin dev:dev）

（4）可以在**本地仓库**中进行工程文件夹、文件、cmakelist等创建

（5）创建完成后，可以通过git add 文件名，将文件添加到本地**暂存区**；然后通过 git commit -m "cc"（引号内的内容可以随意改动），将文件上传到**本地仓库**中；最后可以通过git push origin dev：dev，将本地的dev上传到远程仓库的根目录dev分支下下

（6）将远程仓库的代码下载到本地工作区，git pull origin dev:dev

**3、linux端下载、编译安装aris库**

（1）在Desktop创建aris和build\_aris文件夹

（2）cd到aris文件夹，shell端输入git init

（3）git remote add origin <https://github.com/py0330/aris>

（4）git pull origin dev:dev

（5）shell端命令切到build\_aris文件夹

（6）sudo cmake ../aris/ -DUSE\_ETHERLAB=ON -DUSE\_XENOMAI=ON

（7）sudo make install -j8

**4、其他git命令使用说明**

（1）git merge kaanh：将kaanh分支合并到当前分支

G 过 示 范 
我 们 准 备 了 两 个 分 支 ， 每 个 分 支 上 各 有 一 个 独 有 的 交 。 这 
意 味 着 没 有 一 个 分 支 包 含 了 我 们 修 改 的 所 有 内 容 。 咱 们 涌 过 
合 并 这 两 个 分 支 来 解 决 这 个 间 题 。 
我 们 要 把 
合 并 到 
glt merge bugF1x 
哇 哦 ！ 看 见 了 吗 ？ 苜 先 ， 
父 节 点 的 交 记 录 。 假 如 从 
现 在 指 向 了 一 个 拥 有 两 个 
开 始 沿 着 箭 头 向 上 看 ， 
bugF±x 
0 
m 亡 已 
在 到 达 起 点 的 路 上 会 经 过 所 有 的 交 记 录 。 这 意 味 着 
包 含 了 对 代 码 库 的 所 有 修 改 。 
0 

（2）git rebase master：从本地master拉取代码，更新当前分支。与上一条指令功能相同，唯一区别是，本指令可以让代码仓库的提交历史按照顺序标记，非常清晰

G 过 示 范 
现 在 我 们 切 换 到 了 
上 。 扌 巴 它 rebase 到 
好 了 ！ 由 于 
， 所 以 Git 只 是 简 单 的 
0 
分 支 的 引 用 向 前 移 动 了 一 下 而 已 。 
BugF±x 
0 
0 

（3）git branch -D kaanh：删除kaanh分支

（4）git push origin --delete kaanh：删除远程(github)上的kaanh分支

（5）git push origin --all：推送所有分支到远程仓库

（6）git push origin --force：强行推送当前分支到远程仓库，即使有冲突

（7）git log：列出当前分支的版本历史（可以通过"Q"键，退出显示）

（8）git tag：列出所有tag

（9）git commit -a：提交工作区自上次commit之后的变化，直接到仓库区。相当于执行了git add .和git commit -m两条语句

（10）git reset --hard：重置暂存区与工作区，与上一次commit保持一致

（11）git reset --hard b1c778f：重置暂存区与工作区，与commit——b1c778f保持一致

（12）git reset main.cpp：重置暂存区的指定文件，与上一次commit保持一致，但工作区不变

（13）git stash,git stash pop：暂时将未提交的变化移除，稍后再移入