

git 使用总结

W.J.Z

2019.4.9

1 git 安装与配置

1.1 安装

1. 发行版基础安装包安装 `sudo yum install git`
2. 二进制文件安装: `sudo yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel sudo yum install asciidoc xmlto docbook2x`
下载地址: <https://github.com/git/git/releases>
`tar -zxf git-2.0.0.tar.gz`
`cd git-2.0.0`
`make configure`
`./configure --prefix=/usr`
`make all doc info`
`sudo make install install-doc install-html install-info`

1.2 配置

1. `/etc/gitconfig` 文件: 包含系统上每一个用户及他们仓库的通用配置。如果使用带有 `-system` 选项的 `git config` 时, 它会从此文件读写配置变量
2. `/.gitconfig` 或 `/.config/git/config` 文件: 只针对当前用户。可以传递 `-global` 选项让 Git 读写此文件
3. 当前使用仓库的 Git 目录中的 `config` 文件 (就是 `.git/config`): 针对该仓库
4. 用户信息: `git config --global user.name "JohnDoe"` `git config --global user.email john-doe@example.com`
5. 文本编辑器: `git config --global core.editor emacs`
6. 检查配置信息: `git config --list`
7. 获取帮助: `git help config`
8. 忽略文件 `cat .gitignore`

2 简单命令

1. 在现有目录中初始化仓库: `git init`
2. 跟踪文件: `git add *.c`
3. 提交文件: `git commit -m 'message'`
4. 克隆现有仓库: `git clone https://github.com/libgit2/libgit2`
5. 检查当前文件状态: `git status`
6. 比较工作目录文件和暂存区文件快照的差异: `git diff`
7. 比较已经暂存和上次提交之间的差异: `git diff -cached`
8. 跳过使用暂停区域: `git commit -a -m "`
9. 从工作目录中删除文件: `git rm`
如果已经放入暂存区, 则使用 `git rm -f` 强制删除
10. 从暂存区和仓库中删除: `git rm -cached`
11. 移动文件: `git mv`
12. 查看提交历史: `git log`
13. 撤销操作: `git commit --amend`
14. 取消暂存的文件: `git reset HEAD`
15. 插销对文件的修改: `git checkout -`
16. 创建标签: `git tag -a v1.4 -m 'version 1.4'`
17. 查看标签对应的提交信息: `git show v1.4`
18. git 别名: `git config --global alias.ci commint`

3 分支

1. 分支创建: `git branch testing`
2. 分支切换: `git checkout testing`
3. 创建分支并转移: `git checkout -b testing`
4. 删除分支: `git branch -d testing`
5. 分支合并: `git merge testing`
6. 分支管理: `git branch`

4 gitHub 开源项目贡献演示

1. 将派生的副本复制到本地: `https://github.com/sealgull/hellorworld.git`
2. 切换到项目文件夹下: `cd helloworld/`
3. 创建分支: `git checkout -b author`
4. 进行修改提交仓库: `git commit -a -m "`
5. 推送到远程服务器: `git push origin author` 输入用户名和密码即可
6. 进行比较并推送: `compare & pull request`
7. 创建后项目原有者就会收到合并请求, 之后进行交流。
8. permission denied 解决方法

1. permission denied 错误

```
1 remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
2 To https://github.com/vy2014/git_lesson.git
3 ! [remote rejected] master -> master (permission denied)
4 error: failed to push some refs to 'https://github.com/vy2014/git_lesson.git'
```

复制

Type command:

```
git config --global --edit
```

Add these lines of configuration at the end of file:

```
1 [credential]
2   helper = osxkeychain
3   useHttpPath = true
```

5 同步开源项目别人的更新

1. `git remote -v` 查看远程主机
2. `git remote add upstream git@github.com:xxx/xxx.git` 添加远程主机
3. `git fetch upstream` 从源库更新代码
4. `git merge upstream/master` 合并到本地代码
5. `git push` 同步自己的远程代码
6. `git remote remove upstream` 删除远程主机