git 双帐号配置

1、生成帐号

#新建SSH key：

$ cd ~/.ssh   # 切换到C:\Users\Administrator\.ssh

ssh-keygen -t rsa -C "mywork@email.com" # 新建工作的SSH key

设置名称为id_rsa_personal

Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):

id_rsa_personal

2、新密钥添加到SSH agent中

因为默认只读取id_rsa，为了让SSH识别新的私钥，需将其添加到SSH agent中：

ssh-add ~/.ssh/id_rsa_personal

如果出现Could not open a connection to your authentication agent的错误，就试着用以下命令：

ssh-agent bash

ssh-add ~/.ssh/id_rsa_personal

3、修改config文件

在~/.ssh目录下找到config文件，如果没有就创建：

Host github.com

 HostName github.com

 User liuzhiwe7829

 IdentityFile C:\\Users\\zhiwei.liu003\\.ssh\\id_rsa_personal

Host china-cri.com

 HostName china-cri.com

 IdentityFile C:\\Users\\zhiwei.liu003\\.ssh\\id_rsa

 User zhiwei.liu003

基本操作

git clone url 克隆

git status 查看本地分支文件信息，确保更新时不产生冲突

git checkout -b

git branch（查看当前分支情况）

git 分支切换

https://www.cnblogs.com/smiler/p/6924583.html

git commit 与 pull区别

https://blog.csdn.net/xiaosi_xiaosi/article/details/81336077

gitlab 新建项目

Command line instructions
Git global setup
git config --global user.name "刘志伟"
git config --global user.email "zhiwei.liu003@china-cri.com"
Create a new repository
git clone git@192.168.26.51:chinacri-b2b/chinacri-b2b-gateway.git

```
cd chinacri-b2b-web
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```
Existing folder
```
cd existing_folder
git init
git remote add origin git@192.168.26.51:chinacri-b2b/chinacri-b2b-gateway.git
git remote add origin git@192.168.26.51:chinacri-b2b/chinacri-b2b-admin.git
git remote add origin git@192.168.26.51:chinacri-b2b/chinacri-b2b-car.git
git remote add origin git@192.168.26.51:chinacri-b2b/chinacri-b2b-web.git
git add .
git commit -m "Initial commit"
git push -u origin master
```
Existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin git@192.168.26.51:chinacri-b2b/chinacri-b2b-
web.git
git push -u origin --all
git push -u origin --tags
```

1. 查看远程分支

```
$ git branch -a
```

我在mxnet根目录下运行以上命令:

```
~/mxnet$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
  remotes/origin/nnvm
  remotes/origin/piiswrong-patch-1
  remotes/origin/v0.9rc1
```

可以看到，我们现在在master分支下

2. 查看本地分支

```
~/mxnet$ git branch
* master
```

3. 切换分支

```
$ git checkout -b v0.9rc1 origin/v0.9rc1
Branch v0.9rc1 set up to track remote branch v0.9rc1 from origin.
Switched to a new branch 'v0.9rc1'

#已经切换到v0.9rc1分支了
$ git branch
  master
* v0.9rc1

#切换回master分支
```

```
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
```
合并

https://blog.csdn.net/boysky0015/article/details/78185879/

git remote add origin https://github.com/leeyongbard/learngit.git

git branch -v

git remote -v

git clone url   检出代码 Shift+Insert 快捷复制

git branch -a 查看本地版本

git checkout 版本   ---确保同develop代码一直

git pull 同步代码

git add . 添加提交代码

git commit -m "modify" 提交本地

git push 提交代码到分支

git branch --set-upstream-to=origin/develop develop

git commit -"modify"

还原

git checkout -- filename

git reset --filename

git reset --hard origin/master https://www.jianshu.com/p/fb5a271d2cc0

git clean -df

git remote add origin git@192.168.26.51:chinacri-b2b/chinacri-b2b-web.git

差异比较

git diff version version

合并到master

git pull origin master

git merge develop

回退版本

git reset xxxx

git push -f

删除分支

删除远程分支
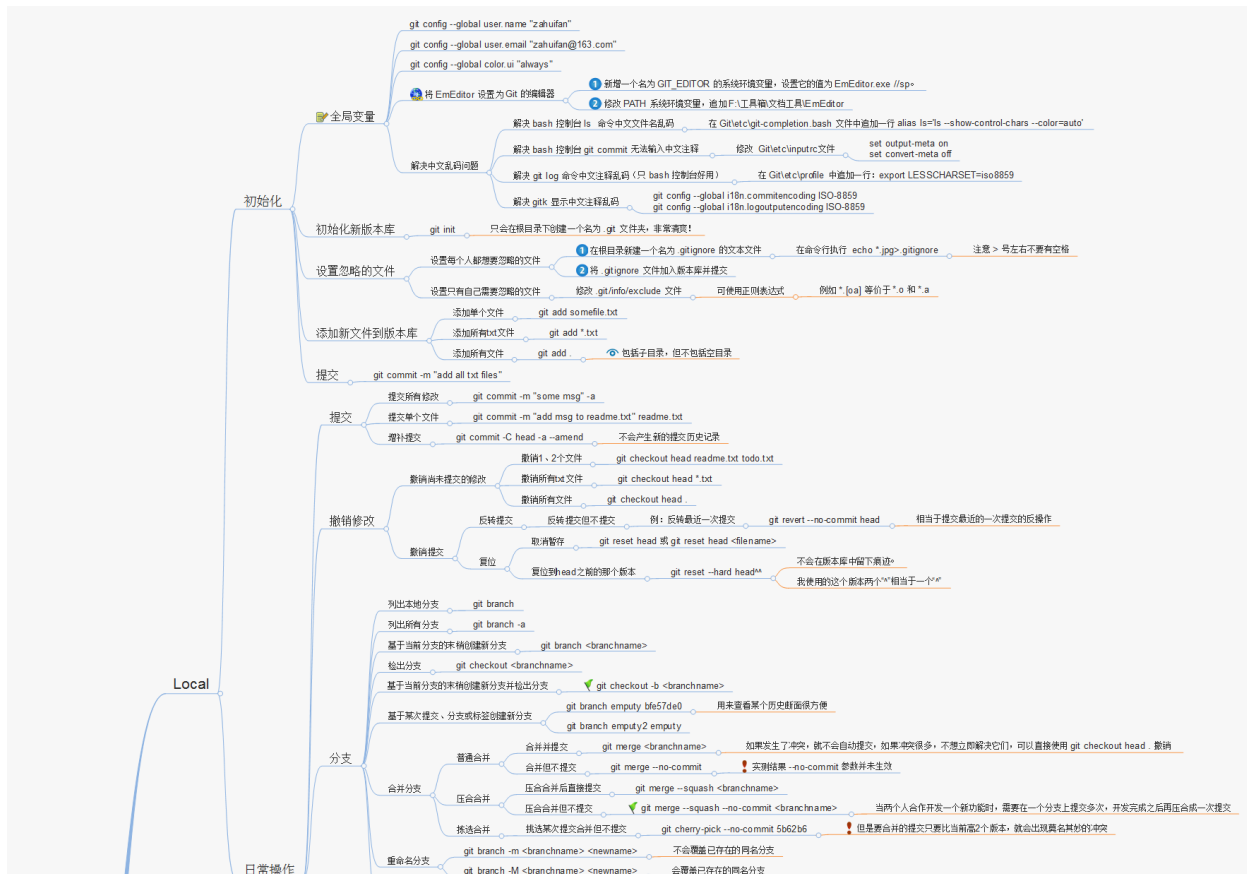
git push origin --delete xxx
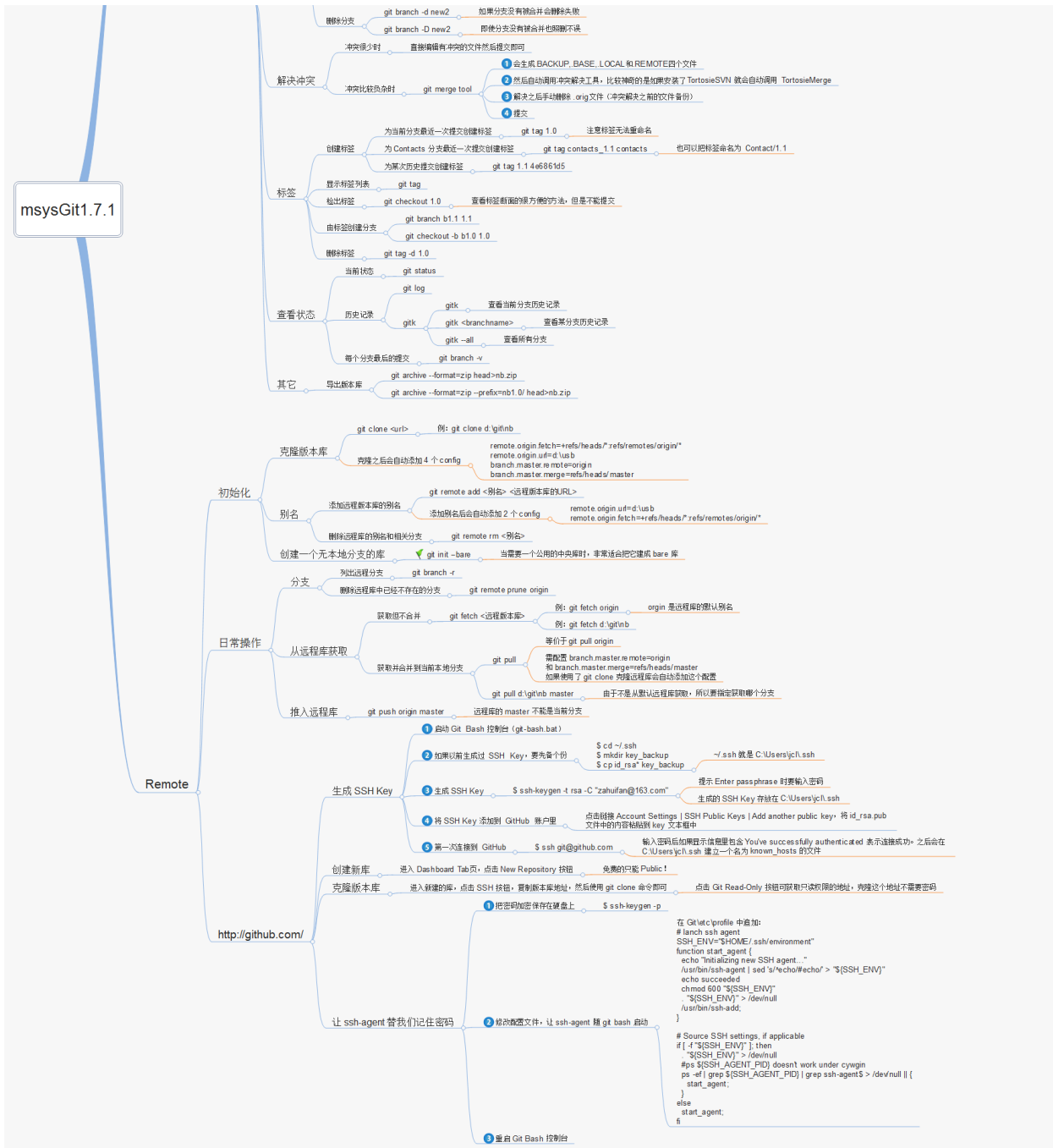
删除本地分支

 git branch -d xxx

问题处理

1、Filename too long

git config --global core.longpaths true

2、github代码下载慢

- DNS解析 http://tool.chinaz.com/dns/
- C:\Windows\System32\drivers\etc\hosts
- 13.250.177.223 github.com

msysGit1.7.1

- 解决冲突
  - 删除分支
    - git branch -d new2 — 如果分支没有被合并会删除失败
    - git branch -D new2 — 即使分支没有被合并也照删不误
  - 冲突很少时 — 直接编辑有冲突的文件然后提交即可
  - 冲突比较负杂时 — git merge tool
    - ① 会生成 BACKUP、BASE、LOCAL 和 REMOTE四个文件
    - ② 然后自动调用冲突解决工具，比较神奇的是如果安装了 TortosieSVN 就会自动调用 TortosieMerge
    - ③ 解决之后手动删除 .orig 文件（冲突解决之前的文件备份）
    - ④ 提交
- 标签
  - 创键标签
    - 为当前分支最近一次提交创键标签 — git tag 1.0 — 注意标签无法重命名
    - 为 Contacts 分支最近一次提交创键标签 — git tag contacts_1.1 contacts — 也可以把标签命名为 Contact/1.1
    - 为某次历史提交创键标签 — git tag 1.1 4e6861d5
  - 显示标签列表 — git tag
  - 检出标签 — git checkout 1.0 — 查看标签面面的很方便的方法，但是不能提交
  - 由标签创建分支
    - git branch b1.1 1.1
    - git checkout -b b1.0 1.0
  - 删除标签 — git tag -d 1.0
- 查看状态
  - 当前状态 — git status
  - 历史记录
    - git log
    - gitk — 查看当前分支历史记录
    - gitk <branchname> — 查看某分支历史记录
    - gitk --all — 查看所有分支
  - 每个分支最后的提交 — git branch -v
- 其它
  - 导出版本库
    - git archive --format=zip head>nb.zip
    - git archive --format=zip --prefix=nb1.0/ head>nb.zip

Remote

- 初始化
  - 克隆版本库
    - git clone <url> — 例：git clone d:\git\nb
    - 克隆之后会自动添加 4 个config
      - remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
      - remote.origin.url=d:\usb
      - branch.master.remote=origin
      - branch.master.merge=refs/heads/master
  - 别名
    - 添加远程版本库的别名 — git remote add <别名> <远程版本库的URL>
    - 添加别名后会自动添加 2 个config
      - remote.origin.url=d:\usb
      - remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
    - 删除远程库的别名和相关分支 — git remote rm <别名>
  - 创建一个无本地分支的库 — git init --bare — 当需要一个公用的中央库时，非常适合把它建成 bare 库
- 日常操作
  - 分支
    - 列出远程分支 — git branch -r
    - 删除远程库中已经不存在的分支 — git remote prune origin
  - 从远程库获取
    - 获取但不合并 — git fetch <远程版本库> — 例：git fetch origin — origin 是远程库的默认别名
      - 例：git fetch d:\git\nb
    - 获取并合并到当前本地分支 — git pull — 等价于 git pull origin
      - 需配置 branch.master.remote=origin 和 branch.master.merge=refs/heads/master
      - 如果使用了 git clone 克隆远程库会自动添加这个配置
    - git pull d:\git\nb master — 由于不是从默认远程库领取，所以要指定获取哪个分支
  - 推入远程库 — git push origin master — 远程库的 master 不能是当前分支
- http://github.com/
  - 生成 SSH Key
    - ① 启动 Git Bash 控制台 (git-bash.bat)
    - ② 如果以前生成过 SSH Key，要先备个份
      - $ cd ~/.ssh — ~/.ssh 就是 C:\Users\jcl\.ssh
      - $ mkdir key_backup
      - $ cp id_rsa* key_backup
    - ③ 生成 SSH Key — $ ssh-keygen -t rsa -C "zahuifan@163.com"
      - 提示 Enter passphrase 时要输入密码
      - 生成的 SSH Key 存放在 C:\Users\jcl\.ssh
    - ④ 将 SSH Key 添加到 GitHub 账户里 — 点击链接 Account Settings | SSH Public Keys | Add another public key，将 id_rsa.pub 文件中的内容粘贴到 key 文本框中
    - ⑤ 第一次连接到 GitHub — $ ssh git@github.com — 输入密码后如果显示信息里包含 You've successfully authenticated 表示连接连接成功。之后会在 C:\Users\jcl\.ssh 建立一个名为 known_hosts 的文件
  - 创建新库 — 进入 Dashboard Tab页，点击 New Repository 按钮 — 免费的只能 Public！
  - 克隆版本库 — 进入建建的库，点击 SSH 按钮，复制版本库地址，然后使用 git clone 命令即可 — 点击 Git Read-Only 按钮可获取只读权限的地址，克隆这个地址不需要密码
  - 让 ssh-agent 替我们记住密码
    - ① 把密码加密保存在硬盘上 — $ ssh-keygen -p
    - ② 修改配置文件，让 ssh-agent 随 git bash 启动 — 在 Git\etc\profile 中追加:

```
# lanch ssh agent
SSH_ENV="$HOME/.ssh/environment"
function start_agent {
  echo "Initializing new SSH agent..."
  /usr/bin/ssh-agent | sed 's/^echo/#echo/' > "${SSH_ENV}"
  echo succeeded
  chmod 600 "${SSH_ENV}"
  . "${SSH_ENV}" > /dev/null
  /usr/bin/ssh-add;
}

# Source SSH settings, if applicable
if [ -f "${SSH_ENV}" ]; then
  . "${SSH_ENV}" > /dev/null
  #ps ${SSH_AGENT_PID} doesn't work under cywgin
  ps -ef | grep ${SSH_AGENT_PID} | grep ssh-agent$ > /dev/null || {
    start_agent;
  }
else
  start_agent;
fi
```

- ③ 重启 Git Bash 控制台

.ssh

?	config
	375B

T	git.txt
	1013B

id_rsa
1.63KB

id_rsa.pub
409B

id_rsa_personal
1.63KB

id_rsa_personal.pub
398B

id_rsa_sl
3.16KB

id_rsa_sl.pub
753B

known_hosts
2.2KB

```
git pull origin master --allow-unrelated-histories
```