

实验七: Git实战

配置管理工具git

版本控制对于大中型软件系统的开发非常重要。

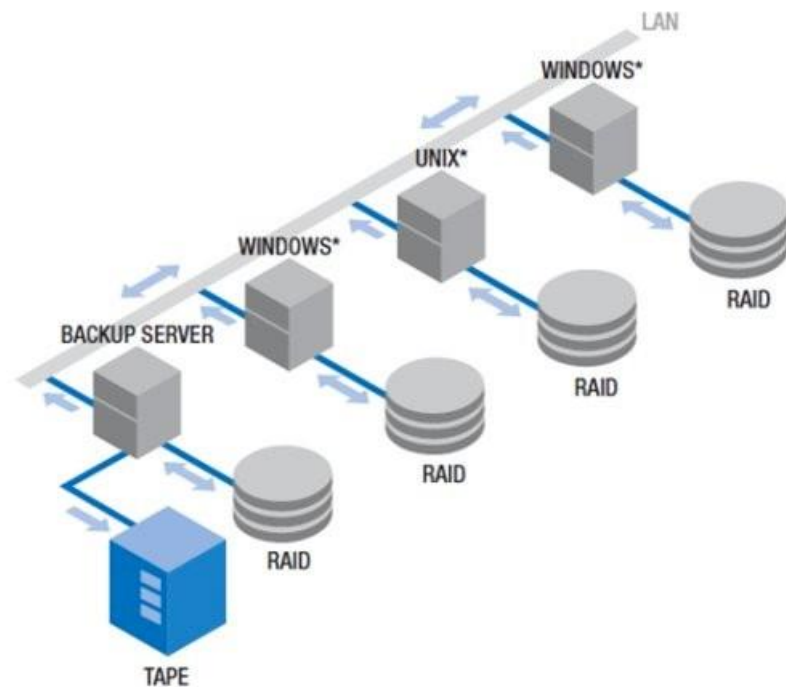
最早为Linux内核项目管理而开发的git工具受到大量其他项目的欢迎。

git是**分布式**的“饭桶”。

/* 自学: <https://try.github.io/> */

Git-it : http://jlord.us/git-it/challenges/get_git.html

实验楼: <https://www.shiyanlou.com/courses/4>



git的基本概念-版本库



远端版本库



本地版本库1



本地版本库2



本地版本库3



本地版本库 n_3

git版本库的操作-创建与提交



```
git init
git status
git add file.txt
git commit -m "msg"
git status
git add *.txt
git commit -m "msg2"
```

git命令行：

Unix系较新的操作系统一般自带，如果没有，

Linux在apt-get可以安装，OS X可以port安装或在<http://sourceforge.net/projects/git-osx-installer/>下载安装。

Windows没有自带git，msysGit项目提供了安装包，可以在<https://git-for-windows.github.io/>下载安装。

git版本库的操作 - 克隆到本地



git clone username@domain:path

比如Github的版本库一般是git@github.com:用户名/版本库名
其实这是ssh格式的URL，也可以用其他的格式，比如Github也提供了https

git版本库的操作 - 从远端拉取



远端版本库



本地版本库2

git pull



本地版本库

git pull

git版本库的操作 - 提交到远端



远端版本库



file4.txt



本地版本库

git pull

git add file4.txt

git commit -m "msg3"

git push

git版本库的操作 - 撤销变动

git add file4.txt

git reset HEAD file4.txt



git checkout -- file4.txt



file4.txt



本地版本库

git版本库的操作 - 修改提交

```
git commit -a -m "msg4"  
git add file4.txt  
git commit --amend
```



file4.txt



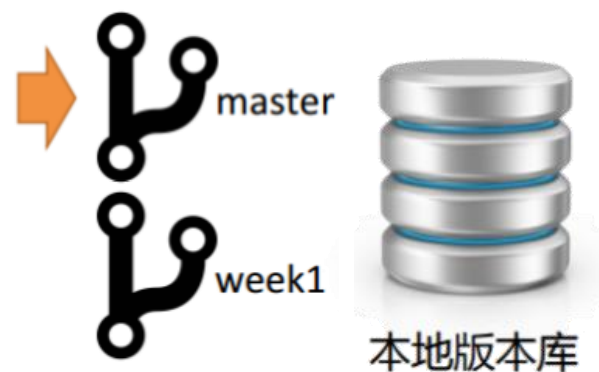
本地版本库

Commit: msg4

git的基本概念 - 分支



远端版本库



```
git checkout -b week1  
git commit -a -m "msg5"  
git push -u origin week1
```

git分支的操作 - 合并



```
git checkout week1  
git commit -a -m "msg6"  
git checkout master  
git merge week1
```

git分支的操作 - 冲突处理



```
<<<<<<< HEAD
Content of current branch
=====
Content of current branch
Merged with Content of week1
>>>>>>> week1
```

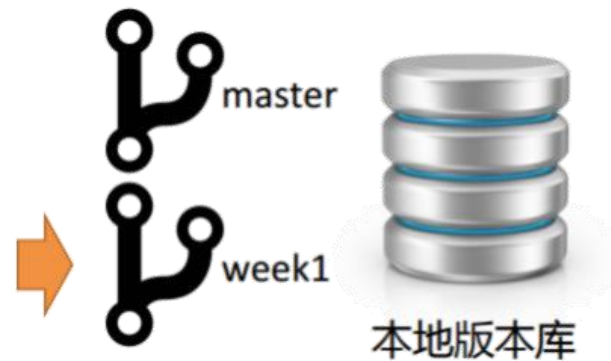


```
git checkout week1
git commit -a -m "msg6"
git checkout master
git merge week1
git add file4.txt
git commit -m "msg7"
```

git分支的操作 - 删除



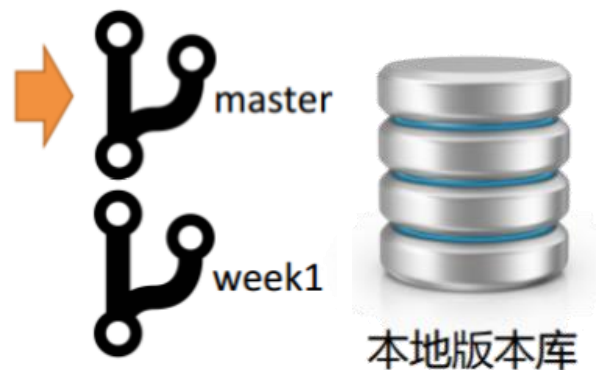
远端版本库



git分支的操作 - 删除



远端版本库



本地版本库

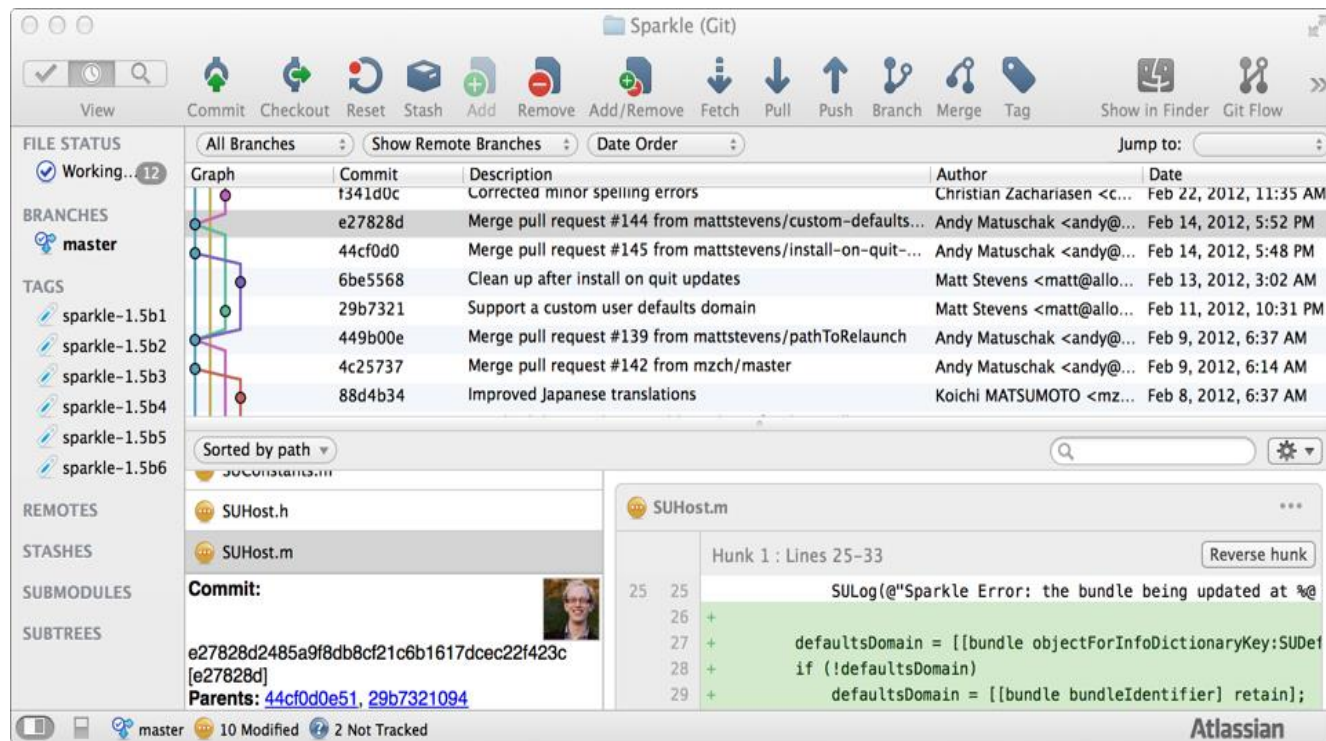
```
git checkout master  
git branch -d week1  
git push origin --delete week1
```


记不住？先用工具偷个懒



<https://www.sourcetreeapp.com>

(Windows & OS X)



GitHub和SourceTree入门教程：

<http://blog.csdn.net/colonn/article/details/39259227>

实验要求

- 熟练掌握git的基本指令和分支管理指令；
- 掌握git支持软件配置管理的核心机理；
- 在实践项目中使用git/github管理自己的项目源代码。

实验内容及过程

○ 安装Git

➤ 本地机器上安装git

如果在linux系统下安装git，给出安装命令和安装后的运行界面；如果在windows下安装git，给出安装的git版本号和在本地上安装git后的运行界面（给出主要界面即可）；

对各个界面做出必要的解释。

➤ 申请github帐号

给出自己在github上申请的帐号名称和本次实验中涉及的项目的URL地址。

给出github网站上你的账号信息和项目信息的截图。

实验内容及过程

○ Git操作过程

- (1) 仓库创建与提交
- (2) 分支管理
- (3) 远程分支管理

○ 在Eclipse中安装和使用Git Plugin

➤ Eclipse中Git plugin的安装和配置

采用屏幕截图的方式给出自己在Eclipse中安装和配置Git plugin的过程。

➤ Eclipse中使用Git plugin

采用屏幕截图和文字说明的方式，给出在Eclipse中使用Git plugin向github提交实验一代码的过程。

实验内容及过程

○ 小结与体会

对本次实验过程和结果的思考：

- 比较之前的开发经验，使用git的优点？
- 之前是否用过其他的版本控制软件？如果有，同git相比有哪些优缺点？ □
- 在什么情况下适合使用git、什么情况下没必要使用git？
- 其他相关思考。