

开源软件之——git 学习篇

主讲人 朱亚菲

幻灯片制作 朱亚菲

中国海洋大学 信息科学与工程学院

2014 年 9 月 19 日

Contents

1 Git 简介

- Git 是什么?
- Git 的诞生

2 Git 使用

- Github
- My Project
- 分支管理

Contents

1 Git 简介

- Git 是什么?
- Git 的诞生

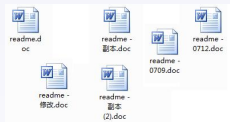
2 Git 使用

- Github
- My Project
- 分支管理

Git

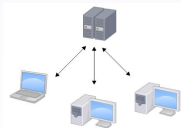
Git 是目前世界上最先进的分布式版本控制系统!

版本控制

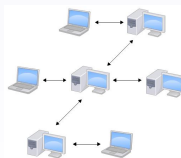


版本	用户	说明	日期
1	张三	删除了实验室规则说明 5	7/12 10:38
2	李四	加入一幅图片	7/12 18:09
3	张三	修改了 README.md	7/13 9:51
4	李四	新增一种绘图颜色	7/14 15:17

集中式 vs 分布式



集中式



分布式

Contents

1 Git 简介

- Git 是什么?
- Git 的诞生

2 Git 使用

- Github
- My Project
- 分支管理

- 1991-2002 年间，世界各地的志愿者把源代码文件通过 diff 的方式发给 Linus，然后由 Linus 本人通过手工方式合并代码！
- 到 2002 年，整个项目组开始启用分布式版本控制系统 BitKeeper 来管理和维护代码。
- 到了 2005 年，开发 BitKeeper 的商业公司同 Linux 内核开源社区的合作关系结束，他们收回了免费使用 BitKeeper 的权力。这就迫使 Linux 开源社区不得不吸取教训，只有开发一套属于自己的版本控制系统才不至于重蹈覆辙。
- Linus 花了两周时间自己用 C 写了一个分布式版本控制系统，这就是 Git！一个月之内，Linux 系统的源码已经由 Git 管理了！

Contents

1 Git 简介

- Git 是什么?
- Git 的诞生

2 Git 使用

- Github
- My Project
- 分支管理

Prearrange

① 安装

② 设置

```
$git config --global user.name "Your Name"
```

```
$git config --global user.email "email@example.com"
```

③ 创建版本库

```
[16:20]fei@CVBI:~/repository[0]  
$mkdir LearnGit  
  
[16:20]fei@CVBI:~/repository[0]  
$cd LearnGit/  
  
[16:20]fei@CVBI:~/repository/LearnGit[0]  
$git init  
初始化空的 Git 版本库于 /home/fei/repository/LearnGit/.git/
```


Contents

1 Git 简介

- Git 是什么?
- Git 的诞生

2 Git 使用

- Github
- My Project
- 分支管理

Git 的杀手级功能之一：远程仓库

服务器

找一台电脑充当服务器的角色，每天 24 小时开机，其他每个人都从这个“服务器”仓库克隆一份到自己的电脑上，并且各自把各自的提交推送到服务器仓库里，也从服务器仓库中拉取别人的提交。

Github

Github 是一个共享虚拟主机服务，用于存放使用 Git 版本控制的软件代码和内容项目。

Contents

1 Git 简介

- Git 是什么?
- Git 的诞生

2 Git 使用

- Github
- My Project
- 分支管理

My Project

Contents

1 Git 简介

- Git 是什么?
- Git 的诞生

2 Git 使用

- Github
- My Project
- 分支管理

Git 的杀手级功能之二：分支模型

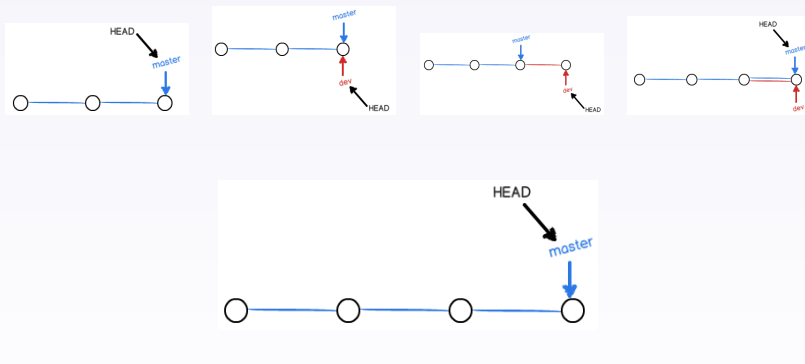
分支

每次提交，Git 都把它们串成一条时间线，这条时间线就是一个分支。

应用场景

假设你准备开发一个新功能，但是需要两周才能完成，如果代码还没写完就提交，不完整的代码库会导致别人无法工作，如果等代码全部写完再一次提交，又存在丢失每天进度的巨大风险。

创建与合并分支



创建与合并分支

- 查看分支: `git branch`
- 创建分支: `git branch name`
- 切换分支: `git checkout name`
- 创建 + 切换分支: `git checkout -b name`
- 合并某分支到当前分支: `git merge name`
- 删除分支: `git branch -d name`

分支管理策略

- master 分支应该是非常稳定的，也就是仅用来发布新版本，平时不能在上面工作。
- dev 分支是不稳定的，平时在 dev 分支上工作，到某个时候，比如 1.0 版本发布时，再把 dev 分支合并到 master 上，在 master 分支发布 1.0 版本。
- 团队中的每个人都在 dev 分支上干活，每个人都有自己的分支，时不时地往 dev 分支上合并。

Thanks

Yafei Zhu

Ocean University of China

2014.09