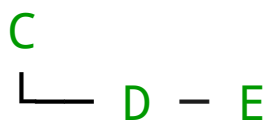
Git 原理简介

- Git 是一种分布式版本控制系统
  - 每个克隆的仓库都包含有整个仓库的所有数据
  - 任何修改先提交在本地，再推送到服务器
  - 创建和切换分支、合并修改相当方便而且快速

- 本文档中第一次出现的术语将以 *蓝色斜体* 标出
  - 术语的含义是单一的、明确的、不可替代的

Git 仓库结构举例

- git@my.server:bob/my.git
  - master
  - dev-1



Git 仓库结构

- 相对本地已克隆的仓库而言，公共的 Git 仓库称为 *remote*
  - 可以用专门的服务器软件管理
  - 也可以直接存放在本机目录中，这样的仓库称为 bare repository
    - SVN 必须使用专门的服务器软件管理
- 每一个仓库是一系列 *commit* 的集合
  - 每一个 commit 都有一个 SHA-1 作为唯一标识符
  - 类似 SVN 的 revision
    - 记录了作者，提交时间，提交日志，所有变更数据等
  - 每一个 commit 都记录了 parent commit
    - 允许非单线提交历史

Git 仓库结构

- Git 仓库中也有 *branch*, *tag* 等等
  - branch 实际是指向一个 branch 中最后的 commit
  - tag 指向某一个 commit
- 选取某一个 commit 的语法称为 *ref*
  - 可以用 SHA-1 指定单个 commit
  - 也可以用 branch 或 tag

git clone git@my.server:bob/my.git

- git@my.server:bob/my.git
  - master A – B – C
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C
    - dev-1 └── D – E
  - master A – B – C
  - HEAD C ──────┘

本地 Git 仓库结构

- 远程分支（ *remote branch* ）
  - `ref: origin/master`
  - 是 remote 中同名分支的镜像
  - push 或 fetch 时会自动更新
- 本地分支（ *local branch* ）
  - `ref: master`
  - 只在本地存在，可自由创建或删除
  - 可以有一个 *upstream branch*，表示是该分支将要与谁同步

HEAD

- HEAD 是一个表示工作区内容基点的 ref
- 如果 HEAD 指向某个 local branch，
  - 进行 commit 或 reset 操作会令该分支指向新的 commit
  - 这种情况下我们说 HEAD 指向的分支为“当前分支”
- 否则，
  - 进行 commit 或 reset 操作不影响本地分支
  - 这种状态称为 detached HEAD

进行修改并提交

- git@my.server:bob/my.git
  - master　A – B – C
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C
    - dev-1　　└── D – E
  - master　A – B – C
  - HEAD　C ──────┘

git commit

- git@my.server:bob/my.git
  - master A – B – C
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C
    - dev-1 └── D – E
  - master A – B – C – m – n
  - HEAD n ──────────────┘

git push

- git@my.server:bob/my.git
  - master A – B – C – m – n
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – m – n
    - dev-1 └── D – E
  - master A – B – C – m – n
  - HEAD n ──────────────┘

如果远程仓库被其他人更新了

- git@my.server:bob/my.git
  - master　A – B – C – H – I
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C
    - dev-1　　└── D – E
  - master　A – B – C – m – n
  - HEAD　n ──────────┘

git fetch

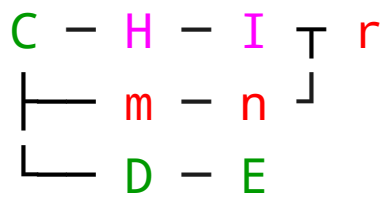
- git@my.server:bob/my.git
  - master A – B – C – H – I
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – H – I
    - dev-1 └── D – E
  - master A – B – C – m – n
  - HEAD n ──────────────┘

git merge origin/master

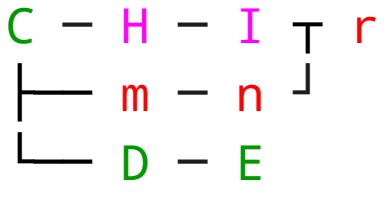
- git@my.server:bob/my.git
  - master A – B – C – H – I
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – H – I
    - dev-1 └── D – E
  - master A – B – C – m – n ┬ r
    └── H – I ┘ |
  - HEAD r ──────────────────┘

git push

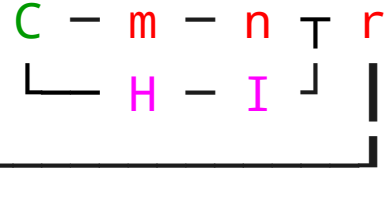
- git@my.server:bob/my.git
  - master
  - dev-1



- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master
    - dev-1



  - master

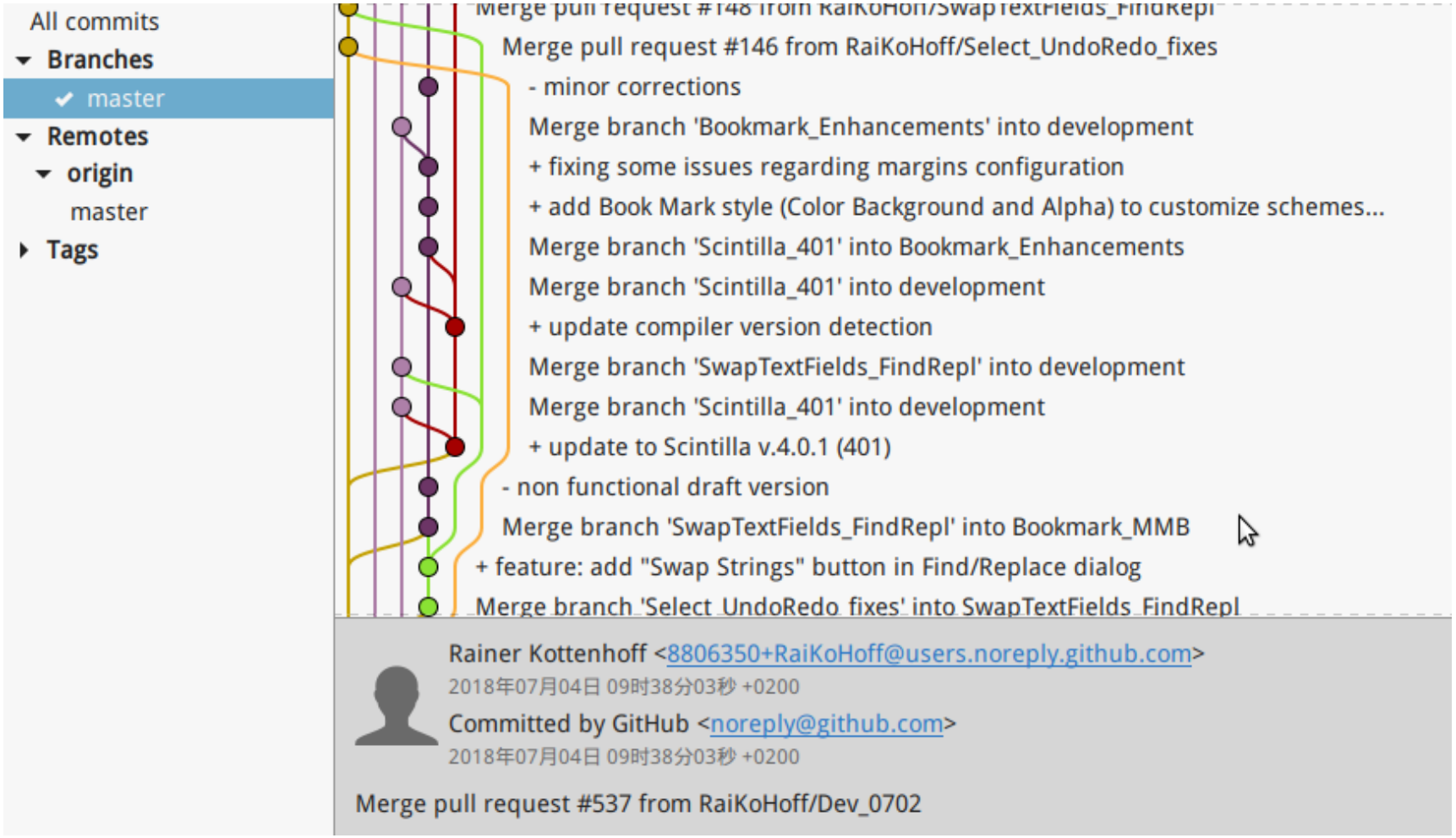


  - HEAD　r

Git 非单线提交历史

- Git 允许多个不同分支合并成为同一条
- 优点
  - 不同分支中的所有 commit 的校验和不变
    - 数字签名也不变
  - 各自的提交历史被全部保留
  - 不会发生“重写历史”的行为
- 缺点
  - 版本回退困难，难以追踪问题

Git 非单线提交历史


All commits
Branches
master
Remotes
origin
master
Tags
Merge pull request #148 from RaiKoHoff/SwapTextFields_FindRepl
Merge pull request #146 from RaiKoHoff/Select_UndoRedo_fixes
- minor corrections
Merge branch 'Bookmark_Enhancements' into development
+ fixing some issues regarding margins configuration
+ add Book Mark style (Color Background and Alpha) to customize schemes...
Merge branch 'Scintilla_401' into Bookmark_Enhancements
Merge branch 'Scintilla_401' into development
+ update compiler version detection
Merge branch 'SwapTextFields_FindRepl' into development
Merge branch 'Scintilla_401' into development
+ update to Scintilla v.4.0.1 (401)
- non functional draft version
Merge branch 'SwapTextFields_FindRepl' into Bookmark_MMB
+ feature: add "Swap Strings" button in Find/Replace dialog
Merge branch 'Select_UndoRedo_fixes' into SwapTextFields_FindRepl
Rainer Kottenhoff <8806350+RaiKoHoff@users.noreply.github.com>
2018年07月04日 09时38分03秒 +0200
Committed by GitHub <noreply@github.com>
2018年07月04日 09时38分03秒 +0200
Merge pull request #537 from RaiKoHoff/Dev_0702

如何保持单线提交历史

- git@my.server:bob/my.git
  - master　A – B – C – H – I
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C – H – I
    - dev-1　　└── D – E
  - master　A – B – C – m – n
  - HEAD　n ──────────────┘

git rebase origin/master

- git@my.server:bob/my.git
  - master A – B – C – H – I
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – H – I
    - dev-1 └── D – E
  - master A – B – C – H – I – m'– n'
  - HEAD n'──────────────────────┘

git push

- git@my.server:bob/my.git
  - master A – B – C – H – I – m'– n'
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – H – I – m'– n'
    - dev-1 └── D – E
  - master A – B – C – H – I – m'– n'
  - HEAD n'──────────────────────┘

撤销本地 commit

- git@my.server:bob/my.git
  - master　A – B – C
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C
    - dev-1　　└── D – E
  - master　A – B – C – m – n
  - HEAD　　n ──────────────┘

git reset HEAD~1

- git@my.server:bob/my.git
  - master　A – B – C
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C
    - dev-1　　└── D – E
  - master　A – B – C – m
  - HEAD　　m ──────────┘

混乱的 git checkout

- `git checkout dev1`
  - 令 HEAD 指向 dev1，即切换到 dev1 分支
  - 指定 `-` 可以切换回上次切换以前的分支
- `git checkout -b dev1`
  - 基于当前 HEAD 创建新的本地分支 dev1
- `git checkout -- dev1`
  - 从 HEAD 中提取文件 dev1 覆盖工作区中的对应文件

基于远程分支创建本地分支

- git@my.server:bob/my.git
  - master　A – B – C
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C
    - dev-1　　└── D – E
  - master　A – B – C – m – n
  - HEAD　n ──────────────┘

git checkout dev-1

- git@my.server:bob/my.git
  - master A – B – C
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C
    - dev-1 └── D – E
  - master A – B – C – m – n
  - dev-1 └── D – E
  - HEAD E ──────────────┘

基于 HEAD 创建本地分支

- git@my.server:bob/my.git
  - master　A – B – C
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C
    - dev-1　　└── D – E
  - master　A – B – C – m – n
  - HEAD　　n ──────────────┘

git checkout -b dev-2

- git@my.server:bob/my.git
  - master A – B – C
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C
    - dev-1 └── D – E
  - master A – B – C – m – n
  - dev-2 └── m – n
  - HEAD n ──────────┘

git push origin dev-2

- git@my.server:bob/my.git
  - master A – B – C
  - dev-1 ├── D – E
  - dev-2 └── m – n
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C
    - dev-1 ├── D – E
    - dev-2 └── m – n
  - master A – B – C – m – n
  - dev-2 └── m – n
  - HEAD n ──────────┘

git push origin dev-2:master

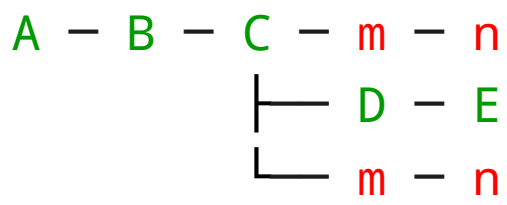
- git@my.server:bob/my.git
  - master　A – B – C – m – n
  - dev-1　　├── D – E
  - dev-2　　└── m – n
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C – m – n
    - dev-1　　├── D – E
    - dev-2　　└── m – n
  - master　A – B – C – m – n
  - dev-2　　└── m – n
  - HEAD　n ──────────┘

删除分支

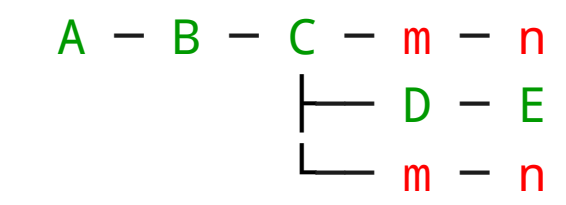
- 删除本地分支
  - `git branch -d dev-2`
- 删除远程分支
  - 本质是推送一个空分支到远程仓库
  - `git push origin :dev-2`
- 更新本地仓库，并删除远程仓库中不再存在的分支
  - `git fetch --prune origin`

git checkout master

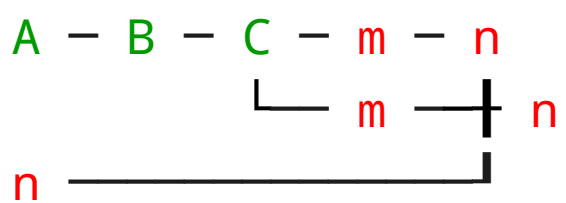
- git@my.server:bob/my.git
  - master
  - dev-1
  - dev-2



- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master
    - dev-1
    - dev-2



  - master
  - dev-2
  - HEAD



git branch -d dev-2

- git@my.server:bob/my.git
  - master A – B – C – m – n
  - dev-1 ├── D – E
  - dev-2 └── m – n
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – m – n
    - dev-1 ├── D – E
    - dev-2 └── m – n
  - master A – B – C – m – n
  - HEAD n ──────────────┘

git push origin :dev-2

- git@my.server:bob/my.git
  - master A – B – C – m – n
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – m – n
    - dev-1 └── D – E
  - master A – B – C – m – n
  - HEAD n ──────────────┘

提取其它分支中的 commit

- git@my.server:bob/my.git
  - master　A – B – C – m – n
  - dev-1　　└── D – E
- 本地文件夹　my
  - origin　git@my.server:bob/my.git
    - master　A – B – C – m – n
    - dev-1　　└── D – E
  - master　A – B – C – m – n
  - HEAD　　n ──────────────┘

git cherry-pick D E

- git@my.server:bob/my.git
  - master A – B – C – m – n
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – m – n
    - dev-1 └── D – E
  - master A – B – C – m – n – D'– E'
  - HEAD E'──────────────────────┘

git push

- git@my.server:bob/my.git
  - master A – B – C – m – n – D'– E'
  - dev-1 └── D – E
- 本地文件夹 my
  - origin git@my.server:bob/my.git
    - master A – B – C – m – n – D'– E'
    - dev-1 └── D – E
  - master A – B – C – m – n – D'– E'
  - HEAD E'──────────────────────┘