Git 学习笔记

高睿昊

2017 年 2 月 9 日

1 安装 git

一提到 git，大佬们常说：

“Git 是目前世界上最先进的分布式版本控制系统”

所以我们必须理解 Git 实际上是一个软件，同样需要安装。

在 Ubuntu 上安装 Git 的命令：`$sudo apt-get install git` 。（Windows 和 Mac 上面的安装很简单，问度娘就好了）安装完成以后还需要配置账户信息

$ git config −−global user.name ”YourName”

$ git config −−global user.email ”email@example.com”

2 创建版本库

在需要 Git 管理的目录下执行命令：

$ git init

对于需要 Git 管理的目录需要使用 add 命令进行处理

$ git add 文件名或者目录名

最后使用 commit 命令即可提交到版本库

$ git commit −m “用于识别的附加在提交后面的消息”

3 Git 时光机

git status 命令可以让我们时刻掌握仓库当前的状态，上面的命令告诉我们，readme.txt 被修改过了，但还没有准备提交的修改。

git diff 顾名思义就是查看 difference，显示的格式正是 Unix 通用的 diff 格式，可以从上面的命令输出看到，我们在第一行添加了一个 “distributed” 单词。

当我们对文件修改后再提交时，也要经过添加新的文件的 add 和 commit 命令。

3.1 版本回退

在 Git 中，可以用 git log 命令查看提交的版本信息，例如在编写个笔记过程中使用的 log 是这样的：

```
commit a0a5fe4e08f77bc5bf609b27575206cb7cb1ccb2
Author: Acinonyx Tungsten <gaoruihao@outlook.com>
Date:   Thu Feb 9 19:54:30 2017 +0800

    1

commit 047711edf118019cd76d9628d3d61b6022f6b008
Author: Acinonyx Tungsten <gaoruihao@outlook.com>
Date:   Thu Feb 9 08:39:34 2017 +0800

    add a line

commit 41320d5047bf318ab067352bfaa5ce658646d188
Author: Acinonyx Tungsten <gaoruihao@outlook.com>
Date:   Thu Feb 9 08:33:59 2017 +0800

    first time
```

这样看起来比较烦的话可以用 `$ git log --pretty=oneline`，得到的结果是这样的：

```
wolf-tungsten@wolftungsten-ThinkPad-T460p:~/NotePapers$ git log --pretty=oneline
a0a5fe4e08f77bc5bf609b27575206cb7cb1ccb2 1
047711edf118019cd76d9628d3d61b6022f6b008 add a line
41320d5047bf318ab067352bfaa5ce658646d188 first time
```

前面一串串的数字就是十六进制表示的版本号了（官方名称是 commit id）。当我们想回退到上一个版本的时候，就可以用命令`$ git reset --hard HEAD^`。其中 HEAD 代表当前版本，就^ 代表上一个版本——以此类推`$ git reset --hard HEAD^^` 自然就是回退到两个版本啦。

现在我们相当于从 21 世纪回到了 19 世纪，那我们要如何再从 19 实际回到 21 世纪呢？这时候刚才看起来烦人的 commit id 就有大用了。首先我们使用`$ git reflog` 命令来查看各个提交版本的 commit id：

```
wolf−tungsten@wolftungsten−ThinkPad−T460p:~/NotePapers$ git reflog
a0a5fe4 HEAD@{0}: commit: 1
047711e HEAD@{1}: commit: add a line
41320d5 HEAD@{2}: commit (initial): first time
```

然后`$ git reset --hard 047711e`，就会回到这个版本。这样的话，时光机就可以任意穿梭了。

3.2 撤销修改

首先我们必须了解 git 的**工作区**和**暂存区**：我们实际操作的文件处于工作区，add 命令的执行就是将工作区的内容添加到了暂存区，commit 命令将暂存区的内容正式提交至版本仓库。

git 的存在相当于给文件修改添加了撤销功能，`$ git checkout -- 文件名或者目录名` 可以让文件恢复到最近一次 commit 或者 add 的状态。如果改动已经提交到了暂存区，就要使用`$ git reset HEAD 文件名或者目录名`把文件撤回到工作区进一步修改之后再进行 add 和 commit 的操作。

撤销更改的几种情况

场景 1 当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令 git checkout – file。

场景 2 当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令 git reset HEAD file，就回到了场景 1，第二步按场景 1 操作。

场景 3 已经提交了不合适的修改到版本库时，想要撤销本次提交，参考版本回退一节，不过前提是没有推送到远程库。

3.3 删除文件

删除文件是一种常见操作。简单的来说我们可以直接在工作区删除一个文件，然后作为一个新版本提交到版本库，这样我们可以随时利用前面的版本回退，或者撤销更改的方法去“一键还原”。

或者，我们可以采用 `$ git rm 文件或者目录` 命令直接从工作区、版本库删除这个文件。