Git 高级操作和原理

这篇博文是由 *ChatGPT-4o* 协助整理的。 Keynote 可见于 https://github.com/lzwjava/Keynotes.

撤销

```
$ git commit --amend
$ git add *
$ git status
$ git reset HEAD CONTRIBUTING.md
Unstaged changes after reset:
M CONTRIBUTING.md
$ git status
$ git checkout -- CONTRIBUTING.md
```

- 重新提交
- 撤销暂存的文件
- 撤销对文件的修改

命令

- `git revert`，回滚，重新 commit，跟之前的某个 commit 想抵消
- `git cherry-pick`，从多个提交中选择其中的提交
- `git rebase`

filter-branch

- 从每个提交中移除掉某个文件
- 想把它删了，但还是在历史里面
- 如何让所有的 commit 都删掉一个文件?

```
$ git filter-branch --tree-filter 'rm -f passwords.txt' HEAD
Rewrite 6b9b3cf04e7c5686a9cb838c3f36a8cb6a0fc2bd (21/21)
Ref 'refs/heads/master' was rewritten
```

查找

```
$ git grep -n gmtime_r
compat/gmtime.c:3:#undef gmtime_r
compat/gmtime.c:8:return git_gmtime_r(timep, &result);
compat/gmtime.c:11:struct tm *git_gmtime_r(const time_t *timep, struct tm *result)
compat/gmtime.c:16:ret = gmtime_r(timep, result);
```

```
compat/mingw.c:606:struct tm *gmtime_r(const time_t *timep, struct tm *result)
compat/mingw.h:162:struct tm *gmtime_r(const time_t *timep, struct tm *result);
date.c:429:if (gmtime_r(&now, &now_tm))
date.c:492:if (gmtime_r(&time, tm)) {
git-compat-util.h:721:struct tm *git_gmtime_r(const time_t *, struct tm *);
git-compat-util.h:723:#define gmtime_r git_gmtime_r
$ git log -SZLIB_BUF_MAX --oneline
e01503b zlib: allow feeding more than 4GB in one go
ef49a7a zlib: zlib can only process 4GB at a time
```

.Git 目录

- config，项目配置
- info，.gitignore
- objects，Git 数据库里的所有内容
- refs，分支的指针
- HEAD，当前分支的指针
- index，staging area 的信息

Git Objects

```
$ git init test
Initialized empty Git repository in /tmp/test/.git/
$ cd test
$ find .git/objects
.git/objects
.git/objects/info
.git/objects/pack
$ find .git/objects -type f
```

```
$ echo 'test content' | git hash-object -w --stdin
d670460b4b4aece5915caf5c68d12f560a9fe3e4
$ find .git/objects -type f
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
```

- `hash-object`，把数据保存到.git 目录的命令
- `-w`，写入对象，否则只是返回 key
- `--stdin`，从标准输入中读取
- d670···，40 个字符的 checksum
- `cat-file`，查看 Git Object 的瑞士军刀

```
$ git cat-file -p d670460b4b4aece5915caf5c68d12f560a9fe3e4
test content
```

```
$ echo 'version 1’  > test.txt
$ git hash-object -w test.txt
83baae61804e65cc73a7201a7252750c76066a30
$ echo 'version 2' > test.txt
$ git hash-object -w test.txt
1f7a7a472abf3dd9643fd615f6da379c4acb3e3a
$ find .git/objects -type f
.git/objects/1f/7a7a472abf3dd9643fd615f6da379c4acb3e3a
.git/objects/83/baae61804e65cc73a7201a7252750c76066a30
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
$ git cat-file -p 83baae61804e65cc73a7201a7252750c76066a30 > test.txt
$ cat test.txt
version 1
$ git cat-file -p 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a > test.txt
$ cat test.txt
version 2
$ git cat-file -t 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a
blob
$ git cat-file -p master^{tree}
100644 blob a906cb2a4a904a152e80877d4088654daad0c859 README
100644 blob 8f94139338f9404f26296befa88755fc2598c289 Rakefile
040000 tree 99f1a6d12cb4b6f19c8655fca46c3ecf317074e0 lib
$ git cat-file -p 99f1a6d12cb4b6f19c8655fca46c3ecf317074e0
100644 blob 47c6340d6459e05787f644c2447d2595f5d3a54b simplegit.rb
Tree Objects
```

```
$ git update-index --add --cacheinfo 100644 \
83baae61804e65cc73a7201a7252750c76066a30 test.txt
update-index
```

- `update-index`，创建树的命令
- `--add`，创建 index
- `--cacheinfo`，从 Git 数据库里读取
- `100644`，文件 mode，为普通文件；100755 为可执行文件；120000 为符号链接
- 83baae，是之前的内容，version 1

- `\`，一行命令拆成两行

write-tree

```
$ git write-tree
d8329fc1cc938780ffdd9f94e0d364e0ea74f579
$ git cat-file -p d8329fc1cc938780ffdd9f94e0d364e0ea74f579
100644 blob 83baae61804e65cc73a7201a7252750c76066a30 test.txt
$ git cat-file -t d8329fc1cc938780ffdd9f94e0d364e0ea74f579
tree
```

- `write-tree`，把暂存区的内容写到 Tree Object 中去

read-tree

```
$ echo 'new file' > new.txt
$ git update-index test.txt
$ git update-index --add new.txt
$ git write-tree
0155eb4229851634a0f03eb265b69f5a2d56f341
$ git cat-file -p 0155eb4229851634a0f03eb265b69f5a2d56f341
100644 blob fa49b077972391ad58037050f2a75f74e3671e92 new.txt
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a test.txt
$ git read-tree --prefix=bak d8329fc1cc938780ffdd9f94e0d364e0ea74f579
$ git write-tree
3c4e9cd789d88d8d89c1073707c3585e41b0e614
$ git cat-file -p 3c4e9cd789d88d8d89c1073707c3585e41b0e614
040000 tree d8329fc1cc938780ffdd9f94e0d364e0ea74f579 bak
100644 blob fa49b077972391ad58037050f2a75f74e3671e92 new.txt
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a test.txt
```

Commit Objects

```
$ echo 'first commit' | git commit-tree d8329f
d5ffe1aa4b7b089eee03dccea5e0439ad6d72037
$ git cat-file -p d5ffe1aa4b7b089eee03dccea5e0439ad6d72037
tree d8329fc1cc938780ffdd9f94


e0d364e0ea74f579
author lzwjava <lzwjava@gmail.com> 1462090215 +0800
committer lzwjava <lzwjava@gmail.com> 1462090215 +0800
```

```
first commit
```

- hash 值会不同，因为创建时间和作者都不同

```
$ echo 'second commit' | git commit-tree 0155eb -p d5ffe1aa4
e946d6367f07de45cac242dca7cd002f5eaa72b1
$ echo 'third commit' | git commit-tree 3c4e9c -p e946d6
09490bf051c34b3693dfd5c7fb63dfe27b295904
$ git log --stat 09490
commit 09490bf051c34b3693dfd5c7fb63dfe27b295904
Author: lzwjava <lzwjava@gmail.com>
Date: Sun May 1 16:38:55 2016 +0800
third commit
bak/test.txt | 1 +
1 file changed, 1 insertion(+)
commit e946d6367f07de45cac242dca7cd002f5eaa72b1
Author: lzwjava <lzwjava@gmail.com>
Date: Sun May 1 16:37:01 2016 +0800
second commit
new.txt | 1 +
test.txt | 2 +-
2 files changed, 2 insertions(+), 1 deletion(-)
commit d5ffe1aa4b7b089eee03dccea5e0439ad6d72037
Author: lzwjava <lzwjava@gmail.com>
Date: Sun May 1 16:10:15 2016 +0800
first commit
test.txt | 1 +
1 file changed, 1 insertion(+)
```

- `commit-tree`，第一个参数指向 blob 或 tree
- `-p` 指向父节点

```
third commit
second commit
first commit
tree
tree
tree
```

“version 2”

“new file”

“version 1”

09490b

e946d6

d5ffe1 d8329f

0155eb

3c4e9c

83baae

fa49b0

1f7a7a

test.txt

new.txt

test.txt

new.txt

test.txt

bak

分支的原理

```
$ find .git/refs
.git/refs
.git/refs/heads
.git/refs/tags
$ find .git/refs -type f
$ echo "09490bf051c34b3693dfd5c7fb63dfe27b295904" > .git/refs/heads/master
$ git log --pretty=oneline master
09490bf051c34b3693dfd5c7fb63dfe27b295904 third commit
e946d6367f07de45cac242dca7cd002f5eaa72b1 second commit
d5ffe1aa4b7b089eee03dccea5e0439ad6d72037 first commit
$ git update-ref refs/heads/master 09490bf051c34b3693dfd5c7fb63dfe27b295904
$ git update-ref refs/heads/test e946d6367f07de45cac242dca7cd002f5eaa72b1
$ git branch
* master
test
```

- 保存一个指针引用
- update-ref，用来更改、添加引用

third commit

second commit

first commit

tree

tree

tree

“version 2”

“new file”

“version 1”

09490b

e946d6

d5ffe1 d8329f

0155eb

3c4e9c

83baae

fa49b0

1f7a7a

test.txt

new.txt

test.txt

new.txt

test.txt

bak

refs/heads/master

refs/heads/test

```
$ cat .git/HEAD
ref: refs/heads/master
$ git symbolic-ref HEAD
refs/heads/master
$ git symbolic-ref HEAD refs/heads/test
$ cat .git/HEAD
ref: refs/heads/test
```

symbolic-ref

tag 原理

```
$ git update-ref refs/tags/v1.0 e946d6367f07de45cac242dca7cd002f5eaa72b1
$ git tag
```

```
v1.0
$ git tag -a v1.1 e946d6367f07de45cac242dca7cd002f5eaa72b1 -m 'test tag’
$ cat .git/refs/tags/v1.1
2766532f03289bc5e158629a8b3faffa5f80b8b6
$ git cat-file -p 276653
object e946d6367f07de45cac242dca7cd002f5eaa72b1
type commit
tag v1.1
tagger lzwjava <lzwjava@gmail.com> 1462103203 +0800
test tag
```

remotes

```
$ git remote add origin git@github.com:schacon/simplegit-progit.git
$ git push origin master
Counting objects: 11, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 716 bytes, done.
Total 7 (delta 2), reused 4 (delta 1)
To git@github.com:schacon/simplegit-progit.git
a11bef0..ca82a6d master -> master
$ cat .git/refs/remotes/origin/master
ca82a6dff817ec66f44342007202690a93763949
```

- 放在 refs/remotes 目录
- Remotes 的引用和分支的区别在于，它们是只读的
- 可以 `git checkout`，但 HEAD 不变，所以用 commit 无法更改 remotes 的引用

```
$ curl https://raw.githubusercontent.com/mojombo/grit/master/lib/grit/repo.rb > repo.rb
$ git checkout master
$ git add repo.rb
$ git commit -m 'added repo.rb'
[master 484a592] added repo.rb
3 files changed, 709 insertions(+), 2 deletions(-)
delete mode 100644 bak/test.txt
create mode 100644 repo.rb
rewrite test.txt (100%)
$ git cat-file -p master^{tree}
100644 blob fa49b077972391ad58037050f2a75f74e3671e92 new.txt
```

```
100644 blob 033b4468fa6b2a9547a70d88d1bbe8bf3f9ed0d5 repo.rb
100644 blob e3f094f522629ae358806b17daf78246c27c007b test.txt
$ git cat-file -s 033b4468fa6b2a9547a70d88d1bbe8bf3f9ed0d5
22044
$ echo '# testing' >> repo.rb
$ git commit -am 'modified repo a bit'
[master 2431da6] modified repo.rb a bit
1 file changed, 1 insertion(+)
$ git cat-file -p master^{tree}
100644 blob fa49b077972391ad58037050f2a75f74e3671e92 new.txt
100644 blob b042a60ef7dff760008df33cee372b945b6e884e repo.rb
100644 blob e3f094f522629ae358806b17daf78246c27c007b test.txt
$ git cat-file -s b042a60ef7dff760008df33cee372b945b6e884e
22054
$ git gc
Counting objects: 18, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (18/18), done.
Total 18 (delta 3), reused 0 (delta 0)
$ find .git/objects -type f
$ git verify-pack -v .git/objects/pack/pack-978e03944f5c581011e6998cd0e9e30000905586.idx
```

Refspec

```
$ git remote add origin https://github.com/schacon/simplegit-progit
[remote "origin"]
url = https://github.com/schacon/simplegit-progit
fetch = +refs/heads/*:refs/remotes/origin/*
$ git log origin/master
$ git log remotes/origin/master
$ git log refs/remotes/origin/master
fetch = +refs/heads/master:refs/remotes/origin/master
[remote "origin"]
url = https://github.com/schacon/simplegit-progit
fetch = +refs/heads/master:refs/remotes/origin/master
fetch = +refs/heads/qa/*:refs/remotes/origin/qa/*
```

push = refs/heads/master:refs/heads

/qa/master

- <src>:<dst>
- refs/remotes 本地的位置

https://github.com/schacon/simplegit-progit

https://github.com/schacon/simplegit-progit

参考资料

- Pro Git