The best way to configur…

Ec2 linux server s… 1

Linux Virtual Console(6)…

dnscrypt-proxy in Archli…

Git as I understand (9): …

Linux Virtual Console(5)…

Git as I understand (8): …

Git as I understand (6): …

Git as I understand (7): …

Git as I understand (5): …

Git as I understand (4): …

Git as I understand (3): …

Git as I understand (2): r…

Git as I understand (1): …

Mac book pro 11.3 Linux…

Mac book pro 11.3 Linux…

Mac book pro 11.3 Linux…

Mac book pro 11.3 Linux…

Mac book pro 11.3 Linux…

Mac book pro 11.3 Linux…
How to debug system p…

Git as I understand (2): reference

As we have seen in previous post, git locate stored objects with the SHA1 (160bit/20byte/40 ascii chars). SHA1 value is not human friendly. Git provide a simple mechanism to assign human friendly names to stored objects called "reference". A reference is just a text file store under ".git/refs" with a user friendly file name. The content of this ascii file is just the SHA1 value of the referred objects. There are four types of objects in Git, but Git only provides two types of references : heads and tags. Heads refer to commit object, and tags refer to tag objects. Why not provide reference to blobs and trees? Because:

a. A blob does not have file name information. We often manipulate blob via a tree object, where file/path name are stored for each blob. We can refer to blobs indirectly via a tree;

b. We generally store tree objects via "commit-tree/commit", where a commit object will be generated side by side with a tree. There is a link to the tree within the commit object. So we can locate tree object via commit;

1. basic ref operation

We can browse existing references via "git for-each-ref" and "git-show-ref":

```
[luke@rmbp project1]$ git for-each-ref
d2f5de55ee3aa1dc90aec01a4e450f61285fece0 commit refs/heads/master
af6aae63b3551d4b650ea52408b4648d1c62ec71 tag refs/tags/v0.1
[luke@rmbp project1]$ git show-ref
d2f5de55ee3aa1dc90aec01a4e450f61285fece0 refs/heads/master
af6aae63b3551d4b650ea52408b4648d1c62ec71 refs/tags/v0.1
[luke@rmbp project1]$ cat .git/refs/heads/master
d2f5de55ee3aa1dc90aec01a4e450f61285fece0
[luke@rmbp project1]$ cat .git/refs/tags/v0.1
af6aae63b3551d4b650ea52408b4648d1c62ec71
```

Git also provides "symbolic link" to reference called "symbolic reference". The most typical symbolic reference is "HEAD".

```
[luke@rmbp project1]$ cat .git/HEAD
ref: refs/heads/master
```

"ref:" signifies this file points to a reference. It is followed by the path names of the reference file, with ".git" as root directory.This is just another layer of indirection. There is a famous saying in computer science: "how to handle a problem? just create another layer of indirection!"

We use "git update-ref" to create/update/delete reference. "git symbolic-ref" will be used to handle symbolic reference.

```
# first create a head reference to a commit
[luke@rmbp project1]$ git update-ref refs/heads/mybranch
```

Mac book pro 11.3 Linux…
The best way to configur…

Linux From Scratch for …
Ec2 linux server s… 1

Linux From Scratch for …
Linux Virtual Console(6)…

Linux From Scratch for …
dnscrypt-proxy in Archli…

Linux From Scratch for …
Git as I understand (9): …

Linux From Scratch for …
Linux Virtual Console(5)…

Linux From Scratch for …
Git as I understand (8): …

Linux From Scratch for …
Git as I understand (6): …

Linux From Scratch for …
Git as I understand (7): …

Linux From Scratch for …
Git as I understand (5): …

Linux From Scratch for …
Git as I understand (4): …

Linux From Scratch for …
Git as I understand (3): …

Git as I understand (2): r…

Linux Virtual Console(4)…
Git as I understand (1): …

Linux Virtual Console(3)…
Mac book pro 11.3 Linux…

Regular Expression: A …
Mac book pro 11.3 Linux…

Automatic Scripts… 2
Mac book pro 11.3 Linux…

Github via SSH: a use c…
Mac book pro 11.3 Linux…

Linux Virtual Console(3)…
Mac book pro 11.3 Linux…

Linux Virtual Console(2)…
Mac book pro 11.3 Linux…
How to debug system p…

```
d2f5de55ee3aa1dc90aec01a4e450f61285fece0
# then we create a symbolic reference to this reference
[luke@rmbp project1]$ git symbolic-ref myhead refs/heads/mybranch
[luke@rmbp project1]$ cat .git/myhead
ref: refs/heads/mybranch
```

After we create the references, we can use them to refer to corresponding objects instead of SHA1 value:

```
[luke@rmbp project1]$ git log mybranch
commit d2f5de55ee3aa1dc90aec01a4e450f61285fece0
Author: Luke Luo <luke.jf.luo@gmail.com>
Date:   Tue May 20 15:27:36 2014 +0800

    initial commit
[luke@rmbp project1]$ git cat-file -p mybranch
tree 68aba62e560c0ebc3396e8ae9335232cd93a3f60
author Luke Luo <luke.jf.luo@gmail.com> 1400570856 +0800
committer Luke Luo <luke.jf.luo@gmail.com> 1400570856 +0800

initial commit
[luke@rmbp project1]$ git cat-file -p myhead
fatal: Not a valid object name myhead
[luke@rmbp project1]$ ls
hello.txt
[luke@rmbp project1]$ git log mybranch
commit d2f5de55ee3aa1dc90aec01a4e450f61285fece0
Author: Luke Luo <luke.jf.luo@gmail.com>
Date:   Tue May 20 15:27:36 2014 +0800

    initial commit
[luke@rmbp project1]$ git cat-file -p mybranch
tree 68aba62e560c0ebc3396e8ae9335232cd93a3f60
author Luke Luo <luke.jf.luo@gmail.com> 1400570856 +0800
committer Luke Luo <luke.jf.luo@gmail.com> 1400570856 +0800

initial commit
[luke@rmbp project1]$ git ls-tree mybranch
100644 blob 3b18e512dba79e4c8300dd08aeb37f8e728b8dad hello.txt
[luke@rmbp project1]$

# finally we delete our newly created references
[luke@rmbp project1]$ git update-ref -d refs/heads/mybranch
[luke@rmbp project1]$ git show-ref
d2f5de55ee3aa1dc90aec01a4e450f61285fece0 refs/heads/master
af6aae63b3551d4b650ea52408b4648d1c62ec71 refs/tags/v0.1
[luke@rmbp project1]$ git symbolic-ref -d myhead
```

GRUB2 How To (6): PX…
Mac book pro 11.3 Linux…
The best way to configur…
GRUB2 How To (5): Buil…
Linux From Scratch for …
Ec2 linux server s… 1
GRUB2 How To (… 2
Linux From Scratch for …
Linux Virtual Console(6)…
GRUB2 How To (3) : UE…
Linux From Scratch for …
dnscrypt-proxy in Archli…
GRUB2 How To (… 4
Linux From Scratch for …
Git as I understand (9): …
Debugging with QEMU …
Linux From Scratch for …
Linux Virtual Console(5)…
GRUB2 How To (1) : Ma…
Linux From Scratch for …
Git as I understand (8): …
QEMU 1.5 build configur…
Linux From Scratch for …
Git as I understand (6): …
Linux From Scratch for …
Git as I understand (7): …
Linux From Scratch for …
Git as I understand (5): …
Linux From Scratch for …
Git as I understand (4): …
Linux From Scratch for …
Git as I understand (3): …
Git as I understand (2): r…
Linux Virtual Console(4)…
Git as I understand (1): …
Linux Virtual Console(3)…
Mac book pro 11.3 Linux…
Regular Expression: A …
Mac book pro 11.3 Linux…
Automatic Scripts… 2
Mac book pro 11.3 Linux…
Github via SSH: a use c…
Mac book pro 11.3 Linux…
Linux Virtual Console(3)…
Mac book pro 11.3 Linux…
Linux Virtual Console(2)…
Mac book pro 11.3 Linux…
How to debug system p…

commit objects. Git organize all artifacts via commit objects.

2. lightweight tag and branch

lightweight tag and branch are just normal ref (not symbolic) stored under "refs/tags" and "refs/heads". git also provide specific command to manage tag and branch. For lightweight tag, it generally points to a commit. For annotated tag, we need to use "git tag -a" to create an tag object, then create a lightweight tag pointing to this tag object.

"git branch" can manage local branch as "update-ref". The more import role is to set up tracking relation between local and remote branches. We will talk about remote repo sync later.

3. reflog
git log the value change of ref under ".git/logs/". "git reflog" will show the logs of HEAD by default. If you like to see the reflog:

```
[luke@rmbp project1]$ git reflog
6612aab HEAD@{0}: checkout: moving from master to test
4b3ac84 HEAD@{1}: commit: master.txt anot
daa6797 HEAD@{2}: checkout: moving from test to master
6612aab HEAD@{3}: checkout: moving from master to test
daa6797 HEAD@{4}: checkout: moving from test to master
6612aab HEAD@{5}: commit: c1
daa6797 HEAD@{6}: checkout: moving from master to test
daa6797 HEAD@{7}: commit (initial): c0
[luke@rmbp project1]$ git reflog HEAD
6612aab HEAD@{0}: checkout: moving from master to test
4b3ac84 HEAD@{1}: commit: master.txt anot
daa6797 HEAD@{2}: checkout: moving from test to master
6612aab HEAD@{3}: checkout: moving from master to test
daa6797 HEAD@{4}: checkout: moving from test to master
6612aab HEAD@{5}: commit: c1
daa6797 HEAD@{6}: checkout: moving from master to test
daa6797 HEAD@{7}: commit (initial): c0

[luke@rmbp project1]$ git reflog master
4b3ac84 master@{0}: commit: master.txt anot
daa6797 master@{1}: commit (initial): c0
og of individual branch, use "git reflog <branchname>"
```

Posted 20th May 2014 by Luke Luo

0 Add a comment

I think, therefore I am

search

Classic Flipcard Magazine Mosaic Sidebar Snapshot Timeslide


GRUB2 How To (6): PX…
Mac book pro 11.3 Linux…
The best way to configur…


GRUB2 How To (5): Buil…
Linux From Scratch for …
Ec2 linux server s… 1


GRUB2 How To (… 2
Linux From Scratch for …
Linux Virtual Console(6)…


GRUB2 How To (3) : UE…
Linux From Scratch for …
dnscrypt-proxy in Archli…


GRUB2 How To (… 4
Linux From Scratch for …
Git as I understand (9): …


Debugging with QEMU …
Linux From Scratch for …
Linux Virtual Console(5)…


GRUB2 How To (1) : Ma…
Linux From Scratch for …
Git as I understand (8): …


QEMU 1.5 build configur…
Linux From Scratch for …
Git as I understand (6): …


Linux From Scratch for …
Git as I understand (7): …


Linux From Scratch for …
Git as I understand (5): …


Linux From Scratch for …
Git as I understand (4): …


Linux From Scratch for …
Git as I understand (3): …

Git as I understand (2): r…


Linux Virtual Console(4)…
Git as I understand (1): …

Linux Virtual Console(3)…
Mac book pro 11.3 Linux…


Regular Expression: A …
Mac book pro 11.3 Linux…


Automatic Scripts… 2
Mac book pro 11.3 Linux…


Github via SSH: a use c…
Mac book pro 11.3 Linux…

Linux Virtual Console(3)…
Mac book pro 11.3 Linux…

Linux Virtual Console(2)…
Mac book pro 11.3 Linux…

Comment as: bharathvenna.v Sign out

Publish Preview Notify me