GIT : Các Th□ Thu□t C□ B□n

Gi□i thi□u: GIT hi□n đ□□c s□ d□ng nhi□u đ□ làm công c□ qu□n lý mã ngu□n và version d□ án thay cho SVN, nh□t là các d□ án ngu□n m□, vì v□y xin gi□i thi□u s□ l□□c □ đây.

Thay vì lao vào c□ m□t bi□n l□nh v□i Git, b□n hãy s□ d□ng các ví d□ c□ b□n đ□ b□t đ□u. M□c dù chúng r□t đ□n gi□n, nh□ng t□t c□ chúng đ□u r□t h□u d□ng. Qu□ th□c là v□y, trong tháng đ□u tiên s□ d□ng Git tôi ch□a bao gi□ v□□t qua nh□ng gì nói trong ch□□ng này.

Ghi l□i Tr□ng thái

B□n mu□n th□ th□c hi□n m□t s□ l□nh gì đó v□i Git? Tr□□c khi làm đi□u đó, th□c hi□n các l□nh sau trong th□ m□c hi□n hành ch□a các mã ngu□n hay văn b□n mà b□n mu□n qu□n lý:

```
$ git init  $ git add .  $ git commit -m "B□n sao l□u đ□u tiên"
```

Bây gi□ n□u nh□ các s□a đ□i v□a xong c□a b□n không nh□ mong đ□i, hãy ph□c h□i l□i b□n cũ:

```
$ git reset --hard # Đ□t l□i tr□ng thái và d□ li□u nh□ l□n commit cu□i
```

Sau đó s□a n□i dung cho đúng ý b□n r□i ghi l□i thành m□t tr□ng thái m□i:

```
$ git commit -a -m "B□n sao l□u khác"
```

Thêm, Xóa, Đ□i Tên

L□nh □ trên ch□ gi□ d□u v□t các t□p tin hi□n di□n t□i th□i đi□m b□n ch□y l□nh **git add**. N□u b□n thêm các t□p tin hay th□ m□c, thì b□n s□ ph□i thông báo v□i Git:

```
$ git add readme.txt Documentation
```

T□□ng t□ nh□ v□y, n□u b□n mu□n Git b□ đi các t□p tin nào đó:

```
$ git rm kludge.h obsolete.c  $ git rm -r incriminating/evidence/ #g□ b□ m□t cách đ□ qui
```

Git xóa b□ nh□ng t□p tin n□u nh□ b□n ch□a làm.

Đ□i tên t□p tin thì cũng gi□ng nh□ là vi□c b□n g□ b□ tên cũ và đ□t vào nó cái tên m□i. S□

d□ng l□nh **git mv** có cú pháp r□t gi□ng l□nh **mv** c□a h□ th□ng Linux. Ví d□:

```
$ git mv bug.c feature.c
```

Ch□c Năng Undo/Redo

Đôi khi b□n ch□ mu□n quay tr□ l□i và b□ đi nh□ng thay đ□i trong quá kh□ t□i m□t th□i đi□m nào đó b□i vì chúng t□t c□ đã sai. Thì l□nh:

```
$ git log
```

s□ hi□n th□ cho b□n danh sách các l□n commit g□n đây cùng v□i giá tr□ băm SHA1:

```
commit 766f9881690d240ba334153047649b8b8f11c664 Author: Bob <bob@example.com> Date: Tue Mar 14 01:59:26 2000 -0800 Replace printf() with write(). commit 82f5ea346a2e651544956a8653c0f58dc151275c Author: Alice <alice@example.com> Date: Thu Jan 1 00:00:00 1970 +0000 Initial commit.
```

Ch□ vài ký t□ c□a giá tr□ băm là đ□ đ□ ch□ ra m□t commit c□ th□; m□t cách khác là chép và dán giá tr□ băm. Gõ:

```
$ git reset --hard 766f
```

đ□ ph□c h□i l□i tr□ng thái đã đ□□c ch□ ra và xóa b□ t□t c□ các l□n commit m□i h□n k□ t□ đó.

M□t lúc nào đó b□n l□i mu□n nh□y t□i m□t b□n cũ h□n. Trong tr□□ng h□p này thì gõ:

```
$ git checkout 82f5
```

Nó giúp b□n quay l□i đúng th□i đi□m đó, trong khi v□n gi□ l□i nh□ng l□n commit m□i h□n. Tuy nhiên, gi□ng nh□ c□ máy th□i gian trong các b□ phim khoa h□c vi□n t□□ng, n□u bây gi□ b□n s□a sau đó commit, b□n s□ □ trong m□t th□c t□i khác, b□i vì hành đ□ng c□a b□n bây gi□ đã khác v□i khi chúng ta l□n đ□u tiên □ t□i đây.

Có cách th□c t□ h□n là s□ d□ng *branch*, và chúng ta có nhi□u đi□u đ□ nói v□ nó sau này. Bây gi□, ch□ c□n nh□ là:

```
$ git checkout master
```

s□ mang chúng ta tr□ v□ hi□n t□i. Ngoài ra, đ□ tránh r□i ro khi s□ d□ng Git, thì luôn luôn commit hay reset các thay đ□i c□a b□n tr□□c khi ch□y l□nh checkout.

S□ t□□ng đ□ng v□i game trên máy tính:

- **git reset --hard**: l□y cái cũ đã đ□□c l□u l□i và xóa t□t c□ các games m□i h□n cái v□a l□y.
- **git checkout**: l□y m□t cái cũ, nh□ng ch□ ch□i v□i nó, tr□ng thái c□a game s□ tách riêng v□ phía m□i h□n ch□ mà b□n đã ghi l□i l□n đ□u tiên. B□t k□ game nào b□n t□o t□ bây gi□ s□ là b□n cu□i cùng trong nhánh riêng r□ t□□ng □ng v□i m□t th□c t□i khác mà b□n đã gia nh□p vào. chúng tôi s□ nói sau .

B□n có th□ ch□n ch□ ph□c h□i l□i các t□p tin hay th□ m□c b□n mu□n b□ng cách thêm vào chúng vào ph□n sau c□a câu l□nh:

```
$ git checkout 82f5 some.file another.file
```

B□n ph□i c□n th□n khi s□ d□ng các l□nh, nh□ là l□nh **checkout** có th□ âm th□m ghi đè lên các t□p tin. Đ□ ngăn ng□a r□i ro nh□ th□, hãy commit tr□□c khi ch□y l□nh checkout, nh□t là khi m□i h□c s□ d□ng Git. Tóm l□i, b□t k□ khi nào b□n không ch□c ch□n v□ m□t l□nh nào đó, dù có là l□nh c□a Git hay không, đ□u tiên hãy ch□y l□nh **git commit -a** .

B□n không thích vi□c c□t dán □? Hãy s□ d□ng:

```
$ git checkout :/"My first b"
```

đ□ nh□y t□i l□n commit mà ph□n chú thích c□a nó b□t đ□u v□i chu□i b□n đã cho. B□n cũng có th□ yêu c□u tr□ng thái th□ 5 k□ t□ cái cu□i cùng:

```
$ git checkout master~5
```

S□ quay l□i

Trong m□t phiên tòa, m□i s□ ki□n đ□□c g□n v□i m□t b□n ghi. Cũng gi□ng th□, b□n có th□ ch□n l□nh commit đ□ undo.

```
$ git commit -a  $ git revert 1b6d
```

s□ ch□ undo l□n commit v□i giá tr□ băm đã ch□ ra. S□ quay tr□ l□i đ□□c ghi nh□n nh□ là m□t l□n commit m□i, b□n có th□ xác nh□n l□i đi□u này b□ng l□nh**git log**.

T□o Nh□t Ký các thay đ□i

M□t s□ d□ án yêu c□u có m□t changelog . T□o m□t cái b□ng cách gõ:

```
$ git log > ThayĐ□i
```

T□i v□ các T□p tin

L□y v□ m□t b□n sao c□a m□t d□ án qu□n lý b□ng Git b□ng cách gõ:

```
$ git clone git://server/path/to/files
```

Ví d□, đ□ l□y t□t c□ các t□p tin mà tôi đã dùng đ□ t□o ra cho quy□n sách này là:

```
$ git clone git://git.or.cz/gitmagic.git
```

Chúng ta s□ có nhi□u đi□u đ□ nói v□ l□nh **clone** s□m thôi.

Th□ Nghi□m

N□u b□n đã t□i v□ m□t b□n sao c□a m□t d□ án b□ng l□nh **git clone**, b□n có th□ l□y v□ phiên b□n cu□i cùng v□i l□nh:

```
$ git pull
```

Xu□t B□n

Gi□ s□ b□n đã t□o đ□□c m□t kho Git và b□n mu□n chia s□ nó v□i ng□□i khác. B□n có th□ b□o h□ t□i v□ t□ máy tính c□a mình, nh□ng n□u h□ làm nh□ th□ trong khi b□n đang c□i ti□n nó hay có nh□ng thay đ□i mang tính th□ nghi□m, h□ có th□ g□p tr□c tr□c. Dĩ nhiên, đây là lý do t□i sao mà chu k□ phát hành ph□n m□m l□i t□n t□i ph□i không nào. Nh□ng ng□□i phát tri□n có th□ làm vi□c th□□ng xuyên trên m□t d□ án, nh□ng h□ ch□ xu□t b□n nh□ng đo□n mã mà h□ c□m th□y nó có th□ dùng đ□□c đ□ tránh □nh h□□ng đ□n ng□□i khác.

Th□c hi□n đi□u này v□i Git, trong th□ m□c làm vi□c c□a Git:

```
$ git init  $ git add .  $ git commit -m "B□n phát hành đ□u tiên"
```

Sau đó nói v□i nh□ng ng□□i cùng s□ d□ng hãy ch□y:

```
$ git clone your.computer:/path/to/script
```

đ□ t□i d□ li□u v□. Gi□ đ□nh là h□ truy c□p thông qua ssh. N□u không, ch□y **git daemon** và nói v□i ng□□i s□ d□ng là ch□y l□nh sau đ□ thay th□:

```
$ git clone git://your.computer/path/to/script
```

K□ t□ lúc này, b□t c□ khi nào mã ngu□n c□a b□n đã có th□ s□ d□ng đ□□c, ch□ vi□c th□c hi□n:

```
$ git commit -a -m "B□n phát hành ti□p"
```

và nh□ng ng□□i s□ d□ng có th□ c□p nh□t d□ li□u c□a h□ b□ng cách chuy□n t□i th□ m□c làm vi□c t□□ng □ng và gõ:

```
$ git pull
```

Nh□ng ng□□i s□ d□ng s□ không bao gi□ th□y đ□□c d□ li□u cu□i cùng c□a b□n mà b□n không mu□n h□ th□y.

Tôi Đã Làm Đ□□c Gì?

Tìm t□t c□ các thay đ□i k□ t□ l□n b□n commit l□n cu□i b□ng l□nh:

```
$ git diff
```

Hay t□ hôm qua:

```
$ git diff "@{yesterday}"
```

Hay gi□a m□t b□n nào đó và b□n tr□□c đây 2 b□n:

```
$ git diff 1b6d "master~2"
```

Trong t□ng tr□□ng h□p, đ□u ra là m□t mi□ng vá mà nó có th□ đ□□c s□ d□ng v□i l□nh **git apply** .
Cũng có th□ dùng l□nh:

```
$ git whatchanged --since="2 weeks ago"
```

Th□□ng th□□ng, tôi duy□t l□ch s□ b□ng qgit đ□ thay th□ cách □ trên, b□i vì nó có giao di□n đ□ h□a bóng b□y, hay tig , có giao di□n dòng l□nh làm vi□c r□t t□t v□i các máy có k□t n□i m□ng ch□m. M□t l□a ch□n khác là cài đ□t máy ch□ web, ch□y l□nh **git instaweb**
và s□ d□ng b□t k□ trình duy□t web nào.

Bài T□p

Coi A, B, C, D là 4 l□n commit thành công, n□i mà B gi□ng A ngo□i tr□ m□t s□ t□p tin b□ xóa b□. Chúng ta mu□n thêm các t□p tin đó tr□ l□i D. Chúng ta th□c hi□n đi□u này b□ng cách nào?

□ đây chúng ta có ít nh□t 3 gi□i pháp. Gi□ thi□t chúng ta đang □ D:

1.

S□ khác nhau gi□a A và B là vi□c các t□p tin đã b□ g□ b□. Chúng ta có th□ t□o mi□ng vá t□□ng □ng v□i s□ khác bi□t này và apply nó:

```
$ git diff B A | git apply
```

2.

K□ t□ sau khi chúng ta ghi l□i các t□p tin t□i A tr□ đi, chúng ta có th□ l□y l□i:

```
$ git checkout A foo.c bar.h
```

3.

Chúng ta có th□ xem s□ di chuy□n t□ A t□i B gi□ng nh□ là m□t thay đ□i mà chúng ta mu□n undo:

```
$ git revert B
```

L□a ch□n nào là t□t nh□t? Cách nào b□n thích nh□t. Th□t d□ dàng đ□ có đ□□c th□ mà b□n mu□n v□i Git, và th□□ng là có nhi□u h□n m□t cách đ□ th□c hi□n đ□□c m□t th□ b□n mu□n.

NGU□N :

http://vnwildman.users.sourceforge.net/gitmagic/ch02.html