

git a github

martin žatkulák

/ *git*

/ distribuovaná povaha a výkon

/ požiadavky na diskovú kapacitu

/ prístupové práva a číslovanie verzií

/ užívateľské rozhrania

/ *github*

/ základy

/ počiatočné nastavenie

/ jednoduchá správa *repository*

/ správa vetiev

/ tagovanie

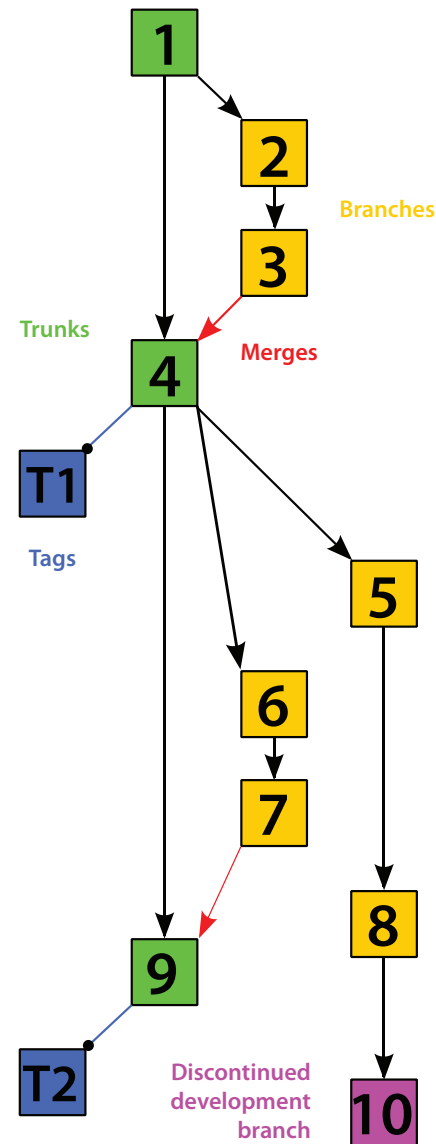
/ publikovanie na *github*

git

git je distribuovaný systém riadenia revízií

pôvodne navrhnutý len ako nízkoúrovňový engine, ktorý by ostatní mohli využiť na napísanie vlastných frontendov

základný *git* projekt sa odvtedy pretransformoval do kompletného systému na správu revízií, ktorý sa dá používať priamo



distribučovaná povaha a výkon

v distribuovaných vcs /version control systems/ akým je *git* má každý užívateľ kompletnú kópiu *repository*

toto umožňuje pristupovať k súborom extrémne rýchlo a bez nutnosti pripojenia k sieti

všetky operácie sú vykonávané lokálne /*init, add, commit, log, status, diff, branch, checkout, merge*/ a až na konci propagované na server /*push*/

pri poškodení *repository* u *git* máme de facto počet záloh rovný počtu užívateľov

v centralizovaných vcs ako *subversion* je treba pre vykonanie každej operácie pristupovať ku centrálnej *repository* na server

ak sa poškodí centrálna *repository* pri použití *subversion* to obvykle znamená stratu všetkých zmien od poslednej zálohy

požiadavky na diskovú kapacitu

git má oveľa nižšie požiadavky na kapacitu ako *subversion*

pracovný adresár u *git* vyžaduje iba jeden indexovací súbor ktorý obsahuje približne 100 bytov pre každý sledovaný objekt

pracovný adresár u *subversion* obsahuje dve verzie toho istého súboru, jeden na ktorom sa pracuje a druhý v skrytom adresári *.svn/* na vykonávanie operácií *diff* a *commit*

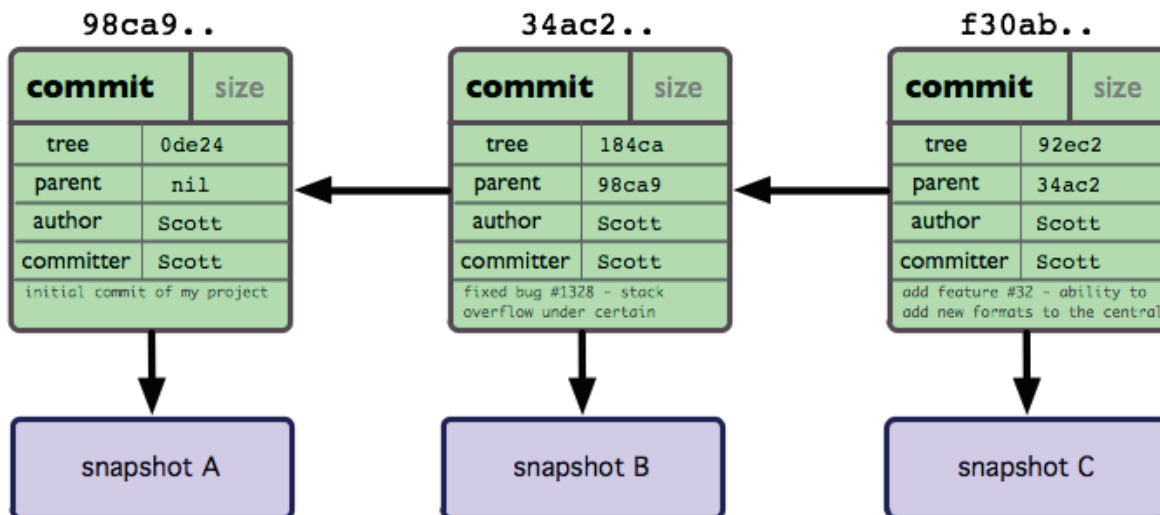
prístupové práva a číslovanie verzií

pre *git* nie je treba nastavovať prístupové práva, sačí sa rozhodnúť ako vykonávať *merge* /od koho a kedy/

git používa jednoznačný 40 bitový sha1 identifikátor na označenie verzie a vetvy z ktorej pochádza

pri *subversion* je možné nastaviť prístupové práva pre jednotlivé lokácie

verzie sú číslované sekvenčne a globálne pre celú *repository* /všetky vetvy/, z čoho plynie nevýhoda nemožnosti určiť z ktorej vetvy ktorá verzia pochádza



užívateľské rozhrania

primárne užívateľské rozhranie je príkazový riadok

pri inštalácii sú zavedené dve rozhrania grafické */git-gui, gitk/*

grafické rozhrania sú vo vývoji */tortoisegit, portované tortoissvn* pre windows/

subversion má viacero /grafických/ užívateľských rozhraní, hlavne pre systémy mac os x a windows v podobe inštalačných balíčkov

github

github je miesto pre založenie verejnej /súkromnej/ online *repository* pre projekt

najjednoduchšia cesta ako participovať na spolupráci a výmene jednotlivých verzií

základy / počítačové nastavenie

globálne nastavenia identity

```
$ git config --global user.name "zollko"
```

```
$ git config --global user.email "zatkulak@gmail.com"
```

kontrola

```
$ cat ~/.gitconfig
```

```
[user]
```

```
    name = zollko
```

```
    email = zatkulak@gmail.com
```

nastavenie primárneho editora /implicitne vi/

```
$ git config --global core.editor 'mate'
```

vygenerovanie dvojice kľúčov /upload verejného na *github*/

```
$ ssh-keygen -t rsa -C "zatkulak@gmail.com"
```


základy / jednoduchá správa *repository*

založenie *repository*

```
$ cd rails_space  
$ git init
```

clone *repository* z existujúceho projektu

```
$ git clone git@github.com:zollko/funeralis.git
```

pridanie súborov projektu a *commit*

```
$ git add .  
$ git commit -a -m "initial import"
```

log pre *repository*

```
$ git log  
commit 1cd3968b62123a2eceacb27aca6ba3ae96b60737  
Author: zollko <zatkulak@gmail.com>  
Date: Tue Mar 2 12:39:58 2010 +0100
```

```
initial import
```

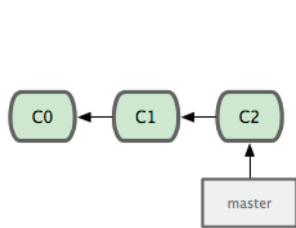
status pre *repository*

```
$ git status  
# On branch master  
nothing to commit (working directory clean)
```

základy / správa vetiev

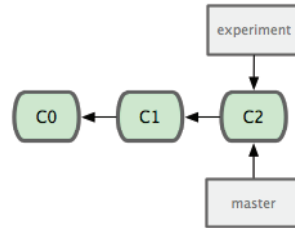
aktuálne vetvy

```
$ git branch  
* master
```



pridanie novej vetvy

```
$ git branch experiment
```

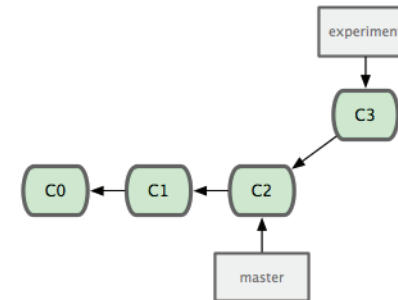


zmena pracovnej vetvy

```
$ git checkout experiment  
Switched to branch "experiment"  
$ git branch  
* experiment  
  master
```

pridanie súboru do vetvy *experiment*

```
$ vi zmena_one.rb  
$ git add .  
$ git commit -a -m "added zmena_one"  
[experiment 43416fb] added zmena_one  
1 files changed, 1 insertions(+), 0 deletions(-)  
create mode 100644 zmena_one.rb
```



zmena pracovnej vetvy

```
$ git checkout master  
Switched to branch "master"  
$ git branch  
  experiment  
* master
```

základy / správa vetiev

pridanie súboru do vetvy *master*

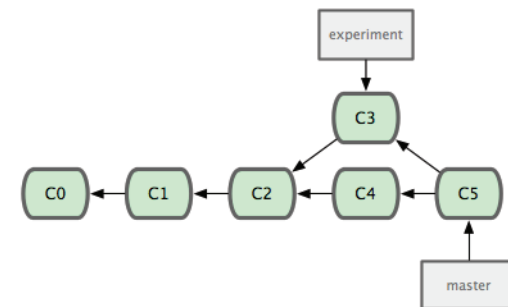
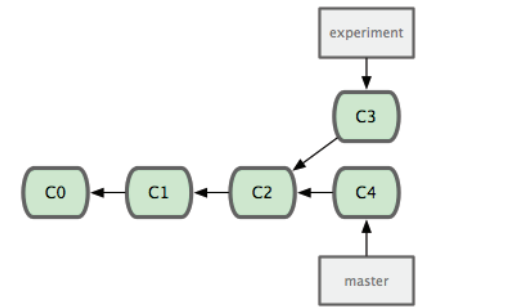
```
$ vi zmena_two.rb
$ git add .
$ git commit -a -m "added zmena_two"
[master f0117ba] added zmena_two
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 zmena_two.rb
```

merge vetvy *experiment* do vetvy *master*

```
$ git merge experiment
Merge made by recursive.
 zmena_one.rb | 1 +
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 zmena_one.rb
```

odstránenie vetvy *experiment*

```
$ git branch -d experiment
```



základy / tagovanie

list tagov

```
$ git tag
```

pridanie tagu

```
$ git tag -a v1.4 -m 'version 1.4'
```

detail verzie

```
$ git show v1.4
```

```
tag v1.4
```

```
Tagger: zollko <zatkulak@gmail.com>
```

```
Date: Wed Mar 3 10:15:05 2010 +0100
```

```
version 1.4
```

```
commit 691dedb70cea0e7d35c7cec6b507d4869dd76fb2
```

```
Author: zollko <zatkulak@gmail.com>
```

```
Date: Tue Mar 2 17:48:40 2010 +0100
```

```
titleize city, state
```

diff a tagy

```
$ git diff v1.4 v1.5
```

základy / publikovanie na *github*

pridanie vzdialenej *repository*

```
$ git remote add origin git@github.com:zollko/rails_space.git
```

odoslanie dát na *github*

```
$ git push origin master  
Counting objects: 198, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (179/179), done.  
Writing objects: 100% (198/198), 114.61 KiB, done.  
Total 198 (delta 32), reused 0 (delta 0)  
To git@github.com:zollko/rails_space.git  
* [new branch]      master -> master
```