

# Git internals i implementacija osnovnih git komandi

Adrian Bralic Toth   Anton Frlan   Marin Jansky

Tehnički Fakultet Rijeka

2018

# Interne datoteke

- Git unutar .git direktorija svakog repozitorija sprema datoteke koje sadrže informacije o repozitoriju
- Neke od tih informacija su popis grana, promjene spremljene u index, konfiguracija, trenutno checkoutani commit ili grana i sl.
- Git svim commitovima i podacima pridodaje jedinstvene ključeve (hasheve) koji se koriste za identificiranje i pristup tim podacima i koriste se u internim gitovim datotekama

# Interne datoteke

```
.git/  
├── hooks/  
│   ├── applypatch-msg.sample  
│   ├── commit-msg.sample  
│   ├── post-update.sample  
│   ├── pre-applypatch.sample  
│   ├── pre-commit.sample  
│   ├── prepare-commit-msg.sample  
│   ├── pre-push.sample  
│   ├── pre-rebase.sample  
│   └── update.sample  
├── info/  
│   └── exclude  
├── objects/  
│   ├── info/  
│   └── pack/  
├── refs/  
│   ├── heads/  
│   ├── remotes/  
│   ├── tags/  
│   └── stash  
├── config  
├── description  
├── FETCH_HEAD  
├── HEAD  
└── index
```

Slika: Datotečna struktura .git direktorija

# Datoteka config

- Datoteka config unutar .git direktorija sadrži osnovne podatke o repozitoriju kao što je popis svih povezanih udaljenih repozitorija, poveznicu na master granu zadanog default repozitorija, interne opcije kao što je osjetljivost na velika i mala slova i još nekoliko postavki vezanih uz sam repozitorij

```
[core]
  repositoryformatversion = 0
  filemode = false
  bare = false
  logallrefupdates = true
  symlinks = false
  ignorecase = true
[remote "origin"]
  url = https://github.com/kabelmeister/Git-Internals
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
[remote "main"]
  url = https://github.com/zvonecan/git-internals
  fetch = +refs/heads/*:refs/remotes/main/*
```

Slika: Primjer config datoteke

# Datoteke HEAD i FETCH\_HEAD

- Datoteka HEAD sadrži referencu na trenutno checkoutan commit ili granu
- Ako je checkoutan commit onda datoteka sadrži jedinstveni hash tog commita (npr. 0e9dc3b589624f744f06b34450829382a61bfe44)
- U slučaju da je checkoutana neka grana onda datoteka sadrži referencu do te grane u obliku: ref: refs/heads/igranaž
- Datoteka FETCH\_HEAD sadrži jedinstveni hash commita kad je zadnji put napravljen fetch sa udaljenog repozitorija te ime grane i lokaciju udaljenog repozitorija

# Direktorij refs/

- Direktorij refs/ sadrži poddirektorije heads, remotes i tags te datoteku stash koja sadrži hash privremenog commita sa stashanim promjenama
- Direktorij refs/heads sadrži datoteku za svaku granu u repozitoriju u kojima je zapisan hash zadnjeg commita u pojedinoj grani
- Sličan direktoriju refs/heads je direktorij refs/remotes i razlikuje se po tome što su u njemu zapisani isti podaci, ali za sve povezane udaljene repozitorije
- Unutar direktorija refs/tags nalaze se datoteke sa svim tagovima koji se koriste u repozitoriju a u njima je zapisan hash commita koji je tagiran tim tagovima

# Ostali direktoriji i datoteke

- Najvažniji direktorij git sustava je `objects/`, u tom direktoriju spremaju se sve datoteke, promjene i commitovi
- `hooks/` direktorij sadrži 11 datoteka sa skriptama koje se pokreću na određeni događaj (npr. kada napravimo commit)
- U datoteci `index` pohranjene su sve promjene od zadnjeg commita dodane u stage u binarnom obliku
- Datoteka `info/exclude` radi na sličnom principu kao `.gitignore`, ali se veže globalno za cijeli repozitorij i ne može se commitati kao `.gitignore`

```
git init
```

- Git init stvori .git datoteku te ostale u njoj napravi još neke datoteke potrebne za rad git-a.

A screenshot of a Linux terminal window titled "ivanecan@ivanecan-VirtualBox: ~/Seminar". The terminal shows a series of Git commands being executed in a directory named "Seminar".  
  
The commands and their outputs are:  
1. `$ cd Seminar`  
2. `ivanecan@ivanecan-VirtualBox:~/Seminar$ ls`  
Output: `README.md`  
3. `ivanecan@ivanecan-VirtualBox:~/Seminar$ git branch`  
Output: `* main  
anton  
marin`  
4. `ivanecan@ivanecan-VirtualBox:~/Seminar$ nano pre1`  
5. `ivanecan@ivanecan-VirtualBox:~/Seminar$ git hash-object`  
6. `ivanecan@ivanecan-VirtualBox:~/Seminar$ ls -l`  
Output: `ls: cannot access 'l': No such file or directory`  
7. `ivanecan@ivanecan-VirtualBox:~/Seminar$ ls .git/`  
Output: `.git/COMMIT_EDITMSG .git/description .git/HEAD .git/ORIG_HEAD  
.git/config .git/FETCH_HEAD .git/index`  
8. `ivanecan@ivanecan-VirtualBox:~/Seminar$ git branches`  
9. `ivanecan@ivanecan-VirtualBox:~/Seminar$ git hooks`  
Output: `applypatch.sample pre-applypatch.sample pre-push.sample  
commit-msg.sample pre-commit.sample pre-rebase.sample  
post-update.sample prepare-commit-msg.sample update.sample`  
10. `ivanecan@ivanecan-VirtualBox:~/Seminar$ git info`  
Output: `exclude  
HEAD refs`  
11. `ivanecan@ivanecan-VirtualBox:~/Seminar$ git objects`  
Output: `0e 1b 27 33 56 68 8d 8f 99 ab c8 ed ff pack  
ff 22 2b 54 5a 6d 86 92 9c ae ca f7 info`  
12. `ivanecan@ivanecan-VirtualBox:~/Seminar$ git refs`  
Output: `heads remotes tags`  
13. `ivanecan@ivanecan-VirtualBox:~/Seminar$`



# Clone i config

- Clone funkcioniira vrlo slično git initu, ali kopira već gotov repozitorij s udaljene lokacije.

```
zvonecan@zvonecan-VirtualBox:~/Testing$ ls -a
.
..
zvonecan@zvonecan-VirtualBox:~/Testing$ git clone /home/zvonecan/Seminar
Cloning into 'Seminar'...
done.
zvonecan@zvonecan-VirtualBox:~/Testing$ ls -a
.
..
Seminar
zvonecan@zvonecan-VirtualBox:~/Testing$ cd Seminar
zvonecan@zvonecan-VirtualBox:~/Testing/Seminar$ ls -a
.
..
.git
new.txt
prezi
README.md
Seminar_aux
Seminar.dvi
Seminar.log
Seminar.nav
Seminar.out
Seminar.snn
Seminar.tex
Seminar.toc
slike
test.txt
zvonecan@zvonecan-VirtualBox:~/Testing/Seminar$
```

- Pomoću naredbe config postavljamo username i email od korisnika

- Git commit git izvršava tako da datoteku spremi pod ključ zvan SHA-1.
- SHA-1 je kombinacija od 40 malih slova ili brojeva koja služi kao pointer na spremljenu datoteku.
- Pomoću komande git hash-object možemo spremi neki file u git te dodavanjem -w nam git vraća SHA-1 od tog filea.
- Pomoću git hash-object spremamo neki file pod SHA-1 kao blob i možemo mu pristupiti samo pomoću SHA-1 ključa.

# git commit

```
romeo@romeo-VirtualBox:~/Seminar$ ls .git/*
.git/COMMIT_EDITMSG  .git/description  .git/HEAD  .git/HEAD~
.git/config           .git/FETCH_HEAD  .git/index
.git/branches:
.git/hooks:
applypatch-msg.sample  pre-applypatch.sample  pre-push.sample
commit-msg.sample      pre-commit.sample      pre-rebase.sample
post-update.sample     prepare-commit-msg.sample  update.sample
.git/info:
exclude
.git/logs:
HEAD  refs
.git/objects:
4d 1b 27 33 56 68 80 88 99 ab cd ed ff pack
0f 22 2b 54 5a 6d 86 92 9c aa ca f7 info
.git/refs:
HEAD  remotes  logs
romeo@romeo-VirtualBox:~/Seminar$ echo 'Neki tekst' > test.txt
romeo@romeo-VirtualBox:~/Seminar$ ls
git  data.txt  prezj  README.md  test.txt
romeo@romeo-VirtualBox:~/Seminar$ git hash-object -w test.txt
bb913ccba4e4a38c5243b57089339839a974cf
romeo@romeo-VirtualBox:~/Seminar$ git cat-file -p bb913ccba4e4a38c5243b57089339839a974cf
git: 'cat-file' is not a git command. See 'git -help'
romeo@romeo-VirtualBox:~/Seminar$ git cat-file -p bb913ccba4e4a38c5243b57089339839a974cf
Neki tekst
romeo@romeo-VirtualBox:~/Seminar$ echo 'Neki tekst verzija 2' > test.txt
romeo@romeo-VirtualBox:~/Seminar$ git hash-object -w test.txt
2aecec3bf5959982b26912392356384b63f87fa1
romeo@romeo-VirtualBox:~/Seminar$ git cat-file -p 2aecec3bf5959982b26912392356384b63f87fa1
Neki tekst verzija 2
romeo@romeo-VirtualBox:~/Seminar$ find .git/objects -type f
.git/objects/ca/42dba3b52f736c3bf763a7127c407416bb89d
.git/objects/mf/a7f4bcb4c8393c1d6b6913355f3f5d4af6a
.git/objects/54/644059ced7d2afdcdfb5dc812c2193d024d9e
.git/objects/2b/6b829c9ca1319e6d1899fcd8e43793c
.git/objects/2b/6b829c9ca1319e6d1899fcd8e43793c
.git/objects/c8/8f3579871eacfa79c9773e7f2653ac8c1cf
.git/objects/8f/81ffa139e8f3774d6ba1cf589ca5b089e5d
.git/objects/56/c4495Secd10e541fbc3a42857a88941d6b5
.git/objects/22/4b3a69493b0b13960236768f0a993746a3386d
.git/objects/1b/bd2163d68999c74e4d66e335e30e1e5ec4e4
.git/objects/ff/0e0f1623b089532820f1a09956d8f4692c6
.git/objects/0e/963c3b58624f744f08b3450829382a81b7e49
.git/objects/8b/9139c3b0a4aa38c35243b57089339839a974cf
.git/objects/ab/7394d1d8b1e4fec2a9d33899e55a21d717
.git/objects/77329578937412f0c736367cd89914f9b6scb8d
.git/objects/ed/7acc73168f03c1ef4c6e1643dc7f46a9247b5
```

# git commit

- Git rješava taj problem pomoću stabala.
- Blob je neki podatak ili tekst dok je stablo više blobova i barem jedno stablo od kojih jedno stablo ima SHA-1 te datoteke ili stabla koji ga sadrži.

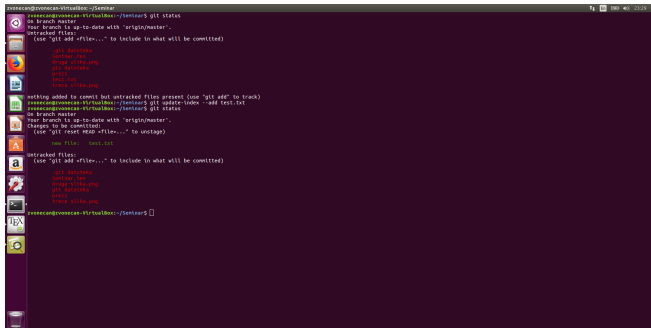
```
zvonecan@zvonecan-VirtualBox:~$ cd Seminar
zvonecan@zvonecan-VirtualBox:~/Seminar$ ls
druga slika.png  git datoteka  prez1  README.md  Seminar.tex  test.txt
zvonecan@zvonecan-VirtualBox:~/Seminar$ git cat-file -p master^{tree}
100644 blob 881cbief1487c8a663c12a3b6af90a15ab8947b2  README.md
zvonecan@zvonecan-VirtualBox:~/Seminar$ clear history

zvonecan@zvonecan-VirtualBox:~/Seminar$ git cat-file -p anton^{tree}
100644 blob 881cbief1487c8a663c12a3b6af90a15ab8947b2  README.md
100644 blob 92591f9a253a9e88f7e7cedcd6d258297c433537  pokusaj
zvonecan@zvonecan-VirtualBox:~/Seminar$
```

- Kada git commita file on obično uzme sto je na stage-u i spremi u jedno stablo ili kombinaciju stabala koji su svi spremljeni pod jedno stablo.

# git stage

- File u gitu stageamo tako da ga dodamo u pod datoteku index unutar .git datoteke.
- File dodajemo u index pomoću git update-index –add.



```
rosecan@rosecan-VirtualBox: ~/Seminar
rosecan@rosecan-VirtualBox:~/Seminar$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    .gitignore
    README.txt
    image1111a.png
    gitignore
    test.txt
    image1111a.png

nothing added to commit but untracked files present (use "git add" to track)
rosecan@rosecan-VirtualBox:~/Seminar$ git update-index --add test.txt
rosecan@rosecan-VirtualBox:~/Seminar$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   test.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    .gitignore
    README.txt
    image1111a.png
    gitignore
    test.txt
    image1111a.png
rosecan@rosecan-VirtualBox:~/Seminar$
```

- Ako želimo stagea neki prijašnji commit prije imena filea moramo dodati –cached i SHA-1 tog filea.
- Sad mozemo napraviti istu stvar što git radi sa fileovima u indexu , stablo.

# git stage

```
romeoan@romeoan-VirtualBox: ~/Seminar
romeoan@romeoan-VirtualBox:~/Seminar$ git write-tree
4d683166fa57dac024fb240c3d00220dc20b990c
romeoan@romeoan-VirtualBox:~/Seminar$ 4d683166fa57dac024fb240c3d00220dc20b990c
4d683166fa57dac024fb240c3d00220dc20b990c command not found
romeoan@romeoan-VirtualBox:~/Seminar$ git cat-file -t 4d683166fa57dac024fb240c3d00220dc20b990c
tree
romeoan@romeoan-VirtualBox:~/Seminar$ echo 'Drugo Stablo' > new.txt
romeoan@romeoan-VirtualBox:~/Seminar$ git update-index --add new.txt
romeoan@romeoan-VirtualBox:~/Seminar$ git write-tree
32642624f83a94a0203fccc733ecf1323f2a32f
romeoan@romeoan-VirtualBox:~/Seminar$ git cat-file -t 32642624f83a94a0203fccc733ecf1323f2a32f
tree
romeoan@romeoan-VirtualBox:~/Seminar$ git cat-file -p 32642624f83a94a0203fccc733ecf1323f2a32f
100644 blob 681c31ef1e87c8a61c32a306a790a15ab0947b2 README.md
100644 blob 832794f0811c0985c2723aacc919f0811b3 new.txt
100644 blob 4ec19ca0f510914d93b2ca1a3fa65770c6b746a test.txt
romeoan@romeoan-VirtualBox:~/Seminar$
```

# git stage

```
romecan@romecan-VirtualBox: ~/Seminar
clean
romecan@romecan-VirtualBox:~/Seminar$ git clean
fatal: clean.requireforce defaults to true and neither -i, -e, nor -f given; refusing to clean
romecan@romecan-VirtualBox:~/Seminar$ git clear
git: 'clear' is not a git command. See 'git --help'.

Did you mean this?
clean
romecan@romecan-VirtualBox:~/Seminar$ git history
git: 'history' is not a git command. See 'git --help'.
romecan@romecan-VirtualBox:~/Seminar$ git clear history
git: 'clear' is not a git command. See 'git --help'.

Did you mean this?
clean
romecan@romecan-VirtualBox:~/Seminar$ clear history
romecan@romecan-VirtualBox:~/Seminar$ clear history

romecan@romecan-VirtualBox:~/Seminar$ git read-tree --prefix=kak 6683166fa5fdac024fb240c3d602260c29b790c
romecan@romecan-VirtualBox:~/Seminar$ git write tree
git: 'write' is not a git command. See 'git --help'.

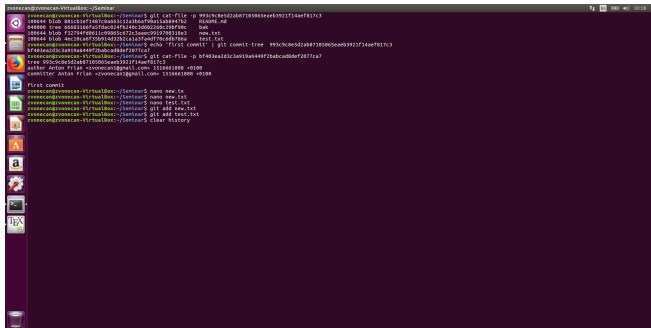
Did you mean one of these?
writeTree
write-tree
romecan@romecan-VirtualBox:~/Seminar$ git write-tree
9b3c8c65d2ab07105865eaab3921f14ae7817c3
romecan@romecan-VirtualBox:~/Seminar$ git cat-file --p93c8c65d2ab07105865eaab3921f14ae7817c3
error: unknown switch 'p'
usage: git cat-file [-t [-allow-unknown-type]] [-s [-allow-unknown-type]] [-e] [-p] [-type] [--textconv] <object>
or: git cat-file [--batch | --batch-check] [--follow-symlinks]

<type> can be one of: blob, tree, commit, tag
-t          show object type
-s          show object size
-e          exit with zero when there's no error
-p          pretty-print object's content
--textconv  for blob objects, run textconv on object's content
--allow-unknown-type allow -s and -t to work with broken/corrupt objects
--buffer    buffer --batch output
--batch[-formats] show info and content of objects fed from the standard input
--batch-check[-formats] show info and content of objects fed from the standard input
--follow-symlinks follow in-tree symlinks (used with --batch or --batch-check)
--batch-all-objects show all objects with --batch or --batch-check

romecan@romecan-VirtualBox:~/Seminar$ git clear history
romecan@romecan-VirtualBox:~/Seminar$ git cat-file -p 9b3c8c65d2ab07105865eaab3921f14ae7817c3
100644 blob 081c1e1f487c4e60c123b0e4f9b05ab047b2 README.md
400000 tree 6683166fa5fdac024fb240c3d602260c29b790c bak
100644 blob 71779f0861e391065c97233aac0959760316d3 new.txt
100644 blob decc18ca0f331914e32b2ca1a3fa5d778c6b786a Test.txt
romecan@romecan-VirtualBox:~/Seminar$
```

# git commit

- Sad samo commitamo stablo koje smo napravili te smo izvršili istu stvar koju git napravi kada nešto addamo i commitamo.



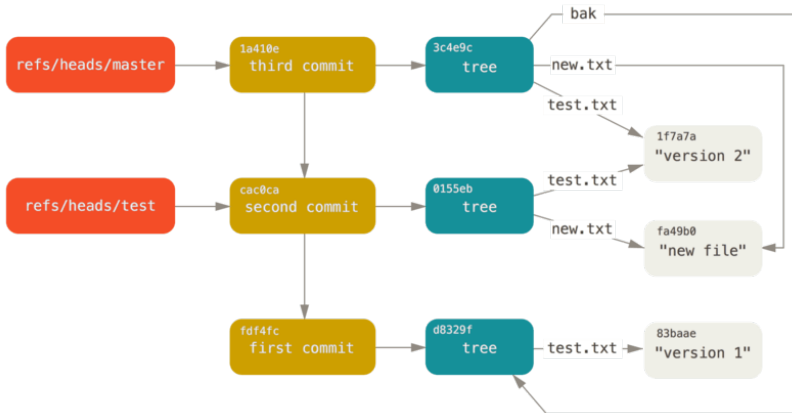
```
romecan@romecan-VirtualBox: ~/Seminar
romecan@romecan-VirtualBox:~/Seminar$ git cat-file -p 9939c9e3d2ab07189b8a5eab3921f14aef817c3
400000 tree 0568116f6c5f0ac247b240c3d0b208c28f56c      hak
100000 blob 9813c15f9f401c0a0d3c22a3a6cf0ea12084873d      hak
100000 blob 0568116f6c5f0ac247b240c3d0b208c28f56c      hak
100000 blob 92724f6e11109952c3e72c2e0c293770d310a2      new.txt
100000 blob ac19c9eef3501643292c4c3a5fa0bf70c4eb786a      test.txt
romecan@romecan-VirtualBox:~/Seminar$ echo "first commit" | git commit -m "first commit"
romecan@romecan-VirtualBox:~/Seminar$ git cat-file -p bf403ea2E3ca019a6440f28acadb0f207ca7
tree 9939c9e3d2ab07189b8a5eab3921f14aef817c3
author Anton Frlan <avroucari@gmail.com> 153661806
committer Anton Frlan <avroucari@gmail.com> 153661806 -v09a
first commit
romecan@romecan-VirtualBox:~/Seminar$ nano new.txt
romecan@romecan-VirtualBox:~/Seminar$ nano new.txt
romecan@romecan-VirtualBox:~/Seminar$ nano test.txt
romecan@romecan-VirtualBox:~/Seminar$ git add new.txt
romecan@romecan-VirtualBox:~/Seminar$ git add test.txt
romecan@romecan-VirtualBox:~/Seminar$ git add test.txt
romecan@romecan-VirtualBox:~/Seminar$ Clear History
```



# Branching

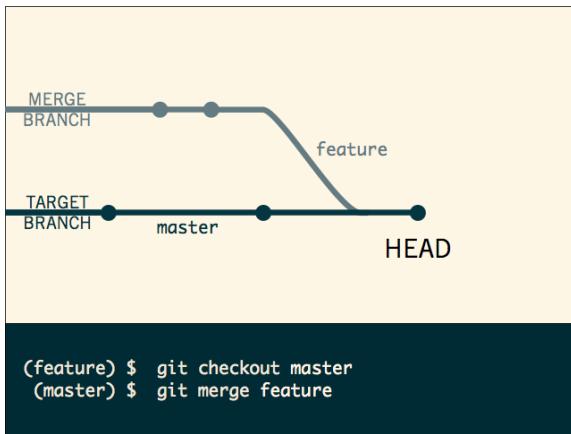
- Jedna od najbitnijih stvari u gitu je grananje.
- Ono omogućuje razvijanje i debugiranje zasebnih featura bez remećenja glavnog programa.
- Za ovo se mogu koristiti reference koje zamijene neki SHA-1 proizvoljnim imenom i odonda se toj novoj grani može pristupiti tim imenom, to jest prati HEAD te grane.

# Branching





# Merge



# HEAD

- Odmah nakon prvog commita u git, prvi se i onda svaki sljedeći commit oznacuje kao HEAD.
- HEAD je referenca koja pokazuje na granu na kojoj se trenutno nalazimo, makar postojala samo jedna.

- Tagovi su vrsta referenci i postoje lightweight and annotated.
  - Lightweight su vrsta referenci koji se ne miču, umjesto ostajući na jednom commitu
  - Annotated su složeniji tagovi koji stvore vlastiti objekt koji ima svoju referencu.

delete pokusa]

```
zvonecan@zvonecan-VirtualBox:~/Seminar$ git tag -a v.34 b85e06c32e4c0cf8517c5be8f480398d8db3c08e -m test tag
fatal: too many params
zvonecan@zvonecan-VirtualBox:~/Seminar$ git tag -a v.34 b85e06c32e4c0cf8517c5be8f480398d8db3c08e
zvonecan@zvonecan-VirtualBox:~/Seminar$ cat .git/refs/tags/v.34
5eba41754830deb19bf740f3483bc6dd37aba661
zvonecan@zvonecan-VirtualBox:~/Seminar$ git update-ref refs/tags/v.42 3cd591aab2835207569878a4d97efb1b83b7990d
zvonecan@zvonecan-VirtualBox:~/Seminar$ cat .git/refs/tags/v.42
3cd591aab2835207569878a4d97efb1b83b7990d
zvonecan@zvonecan-VirtualBox:~/Seminar$
```

- Remote je vrsta referenci koja se pokazuje na udaljeni repozitorij pomoću kojih se može fetchati (povući s udaljenog repozitorija da mozemo pogledati sadržaj), pushati (poslati na repozitorij) i pullati (povucć s udaljenog repozitorija te mergati) sadržaj tih repozitorija.



# Remov, diff i status

- Remove služi za brisanje pojedinih fileova iz trenutačne datoteke.
- Status uspoređuje zadnju commitanu verziju s onim što je napisano poslije i gleda jesu li te razlike stageane kako bi se mogle commitati.
- Diff uspoređuje razlike između zadnje commitane verzije i prvo stageanih promjena, ako nema onda svih promjena te ih ispisuje, a dodatci su zeleni i označeni s plusom, a što je izbrisano je crveno s minusom.

```
zvonecan@zvonecan-VirtualBox:~/Seminar
zvonecan@zvonecan-VirtualBox:~/Testing$ ls -la
...
zvonecan@zvonecan-VirtualBox:~/Testing$ git clone /home/zvonecan/Seminar
Cloning into 'Seminar'...
done.
zvonecan@zvonecan-VirtualBox:~/Testing$ ls -la
... Seminar
zvonecan@zvonecan-VirtualBox:~/Testing$ cd Seminar
zvonecan@zvonecan-VirtualBox:~/Testing/Seminar$ ls -la
... .git new.txt prez1 README.md Seminar.aux Seminar.dvi Seminar.log Seminar.nav Seminar.out Seminar.smm Seminar.tex Seminar.toc slike test.txt
zvonecan@zvonecan-VirtualBox:~/Testing/Seminar$ cd ..
zvonecan@zvonecan-VirtualBox:~/Testing$ cd ..
zvonecan@zvonecan-VirtualBox:~/Seminar$ cd Seminar
zvonecan@zvonecan-VirtualBox:~/Seminar$ nano test
zvonecan@zvonecan-VirtualBox:~/Seminar$ nano test.txt
zvonecan@zvonecan-VirtualBox:~/Seminar$ git diff test.txt
diff --git a/test.txt b/test.txt
index c340e66..a7b9e9a 100644
--- a/test.txt
+++ b/test.txt
@@ 1,1,1 @@
-neki tekst verzija 2

drugi commit

d
+nuzja
-neso commit

nala promjena
+urabovic

nala git "zabava"
zvonecan@zvonecan-VirtualBox:~/Seminar$ git log --oneline --decorate
7776ae5 HEAD -> anton, origin/anton merge plz
7883b9e adrian
8c0c916 plz help
0cfb983 Update Seminar.tex
2e0d809 bore ponoci
703b9e8 update Seminar.tex
3e0cb85 Merge branch 'anton' of https://github.com/Zvonecan/Umutaraja-struktura-Git-a-i-implementacija-osnovnih-komandi-Git-Internals- into anton
3cd591a (tag: v.42) svedno
3d2e425 Delete pokusaj 1
36c4409 pokusaj 1
ed7acc7 (origin/master, master) Update README.md
99a77f4 Update README.md
54e4481 Update README.md
1bbd216 Update README.md
f7f056f formatiranje 1
ca3208e clonovi tina
zvonecan@zvonecan-VirtualBox:~/Seminar$
```

# Log i stash

- Log pokazuje povijest pojedinih grana. Tu se broje commitovi, autor, datum, vrijeme i reference.
- Stash privremeno sakriva necommitane promjene ako se iz nekih razloga treba nešto napraviti bez njih. Promjene ostaju skrivene u stashu dok ih se ne povрати nazad na stage.



[1] Merge,

<http://vimcasts.org/episodes/fugitive-vim-resolving-merge-conflicts-with-vimdiff/>



[2] Git internals i implementacija osnovnih komandi,

<https://git-scm.com/book/en/v2/Git-Internals-Plumbing-and-Porcelain>