

Radno okruženje



Uvod u GIT

- Git je distribuirani sustav za praćenje promjena izvornog koda tijekom razvoja softvera.
- Dizajniran je za koordiniranje rada među programerima, ali se može koristiti za praćenje promjena u bilo kojem skupu datoteka.
- Njegove prednosti uključuju brzinu, integritet podataka i podršku za distribuirane, nelinearne tijekove rada.
- Git je kreirao Linus Torvalds 2005. godine za razvoj Linux kernela, a drugi su programeri kernela pridonijeli njegovom početnom razvoju.
- Neki od poznatijih servisa koji nude besplatna i plaćena GIT spremišta su: **GitHub**, **GitLab**, **Bitbucket**, itd.

GIT naredbe

- Općenito, git naredbe uvijek imaju sljedeći format:

```
git <naredba> <opcija1> <opcija2> ...
```

- Za svaku git naredbu možemo dobiti pomoć s:

```
git help <naredba>
```

- Nekoliko postavki je poželjno konfigurirati da bismo nastavili normalan rad. Sva git konfiguracija se postavlja pomoću naredbe `git config`. Postavke mogu biti lokalne (odnosno vezane uz jedan jedini projekt) ili globalne (vezane uz korisnika na računalu).

- Globalne postavke se postavljaju s:

```
git config --global <naziv> <vrijednost>
```

- Lokalne postavke se postavljaju s:

```
git config <naziv> <vrijednost>
```

GIT naredbe

- Stvaranje novog lokalnog repozitorija:

```
git init
```

- Provjera stanja repozitorija:

```
git status
```

- Dodavanje promjena u indeks:

```
git add
```

- Spremanje novog stanja projekta u lokalni repozitorij:

```
git commit
```

GIT naredbe

- Stvaranje novog lokalnog repozitorija:

```
git init
```

- Provjera stanja repozitorija:

```
git status
```

- Dodavanje promjena u indeks:

```
git add <datoteka>
```

- Spremanje novog stanja projekta u lokalni repozitorij:

```
git commit -m <message>
```

GIT repozitoriji

- U GIT-u se repozitoriji dijele na lokalne i udaljene.
- Apsolutno svaki direktorij može postati git repozitorij.
- Ne mora uopće postojati udaljeni server i neki centralni repozitorij kojeg koriste i ostali koji rade na projektu.
- U prošlom poglavlju naučili smo GIT naredbu `git init` koja kreira lokalni repozitorij. Međutim što ako taj lokalni repozitorij želimo pohraniti na udaljeni repozitorij?

GIT repozitoriji

Prvu stvar koju ćemo obraditi je kreiranje udaljenog repozitorija.

Svaki udaljeni repozitorij s kojime će git "komunicirati" mora imati svoju adresu.

Povezivanje lokalnog s udaljenim repozitorijem možemo učiniti s:

```
git remote add origin <adresa_udaljenog_repozitorija>
```

Naredba za "prebacivanje" izmjena iz lokalnog u udaljeni repozitorij:

```
git push
```

Naredba za dohvat promjena iz udaljenog repozitorija ali ne i sjedinjavanje s lokalnim repozitorijem:

```
git fetch
```

Naredba za dohvat i sjedinjavanje promjena s udaljenog repozitorija:

```
git pull
```

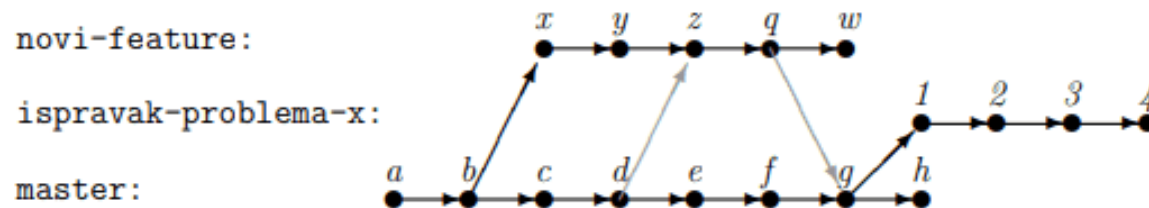
Git repozitoriji - kloniranje

- Kloniranje je postupak kojim kopiramo cijeli repozitorij s udaljene lokacije na naše lokalno računalo. S tako kloniranim repozitorijem možemo nastaviti rad kao s repozitorijem kojeg smo inicirali lokalno.
- Kloniranje udaljenog repozitorija možemo učiniti s:

```
git clone <adresa_udaljenog_repozitorija>
```
- Kloniranje je kopiranje udaljenog repozitorija, ali tako da novi (lokalni) repozitorij ostaje "svjestan" da je on kopija nekog udaljenog repozitorija.
- Klonirani repozitorij čuva informaciju o repozitoriju iz kojeg je kloniran.

Git grananje

- Grananje je umnožavanje objekta (poput datoteke s izvornim kodom ili stabla direktorija), tako da se izmjene mogu odvijati paralelno duž više grana.
- Grane omogućuju paralelno razvijanje dijelova softvera.
- Grane omogućuju suradnicima da izoliraju promjene bez destabiliziranja baze koda, na primjer, ispravke pogrešaka, dodavanje novih značajki i integraciju novih verzija. Te se promjene kasnije mogu spojiti (ponovo sinkronizirati) nakon testiranja.
- Primjer grananja:



Git grananje

- Popis grana projekta:

```
git branch
```

- Stvaranje nove grane:

```
git branch <naziv_nove_grane>
```

- Prebacivanje s grane na granu:

```
git checkout <naziv_grane>
```

- Brisanje grane:

```
git branch -D <naziv_grane>
```

Preuzimanje datoteke iz druge grane:

```
git checkout <naziv_grane> -- <datoteka1> <datoteka2> ...
```

Preuzimanje izmjena iz jedne grane u drugu:

```
git merge <naziv_grane>
```

Hvala na pažnji!

