
Git

Git je distribuovaný verzovací systém.

Získání souborů a kontrola stavu

clone

Slouží k vytvoření lokální kopie vzdáleného repozitáře.

```
git clone <remote address>
```

pull

Synchronizuje lokální verzi repozitáře se vzdáleným repozitářem, tj. stáhne všechny změny.

```
git pull
```

status

Vypíše všechny neuložené změny v lokálním repozitáři (které nejsou v commitu).

```
git status
```

log

Vypíše poslední provedené commity, jejich popis, ID a další informace.

```
git log
```

Nahrání změn

Provedení změn



Prázdný commit



Příprava souborů
`git add <file>`



Zabalení k odeslání
`git commit`



Odeslání do repozitáře
`git push`



Obrázek 1: Proces nahrání lokálních změn do repozitáře

add

Připraví vybrané soubory ke commitu. Dokud se neprovede commit, lze soubory přidávat a odstraňovat.

```
git add <file>
```

commit

V podstatě „zabalí“ všechny připravené soubory a vytvoří nový stav, ke kterému se v budoucnu lze vrátit. Změny provedené v commitu standardně nelze měnit.

Příkaz pro commit otevře editor, ve kterém se na první řádek napíše stručný popis změn s maximálním počtem 72 znaků. Na další řádky lze změny popsat do detailu, ovšem řádky by se měly zalamovat po nejvíce 72 znacích. Editor *vim* může zalamování automatizovat pomocí `set tw=72`.

```
git commit
```

Nechceme-li pracovat v editoru, máme možnost popis zadat rovnou přepínačem `-m`.

```
git commit -m "Kratky popis zmen"
```

push

Nahraje všechny lokálně vytvořené commity na vzdálený repozitář.

```
git push
```

Konflikty

Dojde-li k situaci, kdy dva uživatelé pracovali na stejném souboru a jeden nahrál změny do hlavního repozitáře, musí druhý uživatel před nahráním vlastních změn vyřešit potencionální konflikty.

Konflikt nastane v případě, že oba uživatelé upravili stejný řádek, nebo provedli podobné změny. Druhý uživatel musí tedy ručně (s pomocí *gitu*) zvolit, které řádky chce zachovat a které nahradit

Před vyřešením konfliktu je nutné lokální repozitář synchronizovat na poslední verzi.

Práce s větvemi

Pro výběr větve, se kterou chceme pracovat, použijeme příkaz `git checkout <branch>`, ve kterém musíme zvolit správný název větve. Chceme-li větev vytvořit, můžeme příkaz rozšířit o přepínač `-b`, tj. `git checkout -b <branch>`. Mazání větví provádíme přepínačem `-d`.

Existující větev lze vypsat příkazem `git branch -a`. Aktivní větev je označena hvězdičkou.

Spojení větví se provádí příkazem `git merge <branch>`, který spojí specifikovanou větev s aktivní větví vybranou příkazem `checkout`.