**GIT**

**Uvod u Git**

Git je sistem kontrole verzija za praćenje promena u računarskim fajlovima i za koordinisan rad više ljudi na tim fajlovima. Primarno se koristi za verzionisanje koda pri razvoju softvera, ali se može koristiti za praćenje promena u bilo kojim fajlovima. To je distribuirani sistem kontrole verzija čija je glavna prednost brzina, integritet podataka, i podrška za distribuirane, nelinearne poslove.

Razvio ga je Linus Torvalds, 2005. godine za potrebe razvoja Linuks jezgra, uz doprinos i drugih programera jezgra njegovom prvobitnom razvoju.

Kao i kod većine drugih distribuiranih sistema kontrole verzija, a za razliku od većine klijent-server sistema, svaki Git radni direktorijum, na svakom računaru predstavlja kompletan repozitorijum sa kompletnom istorijom i upotpunjenom mogućnošću za praćenje verzija, nezavisno od pristupa mreži ili centralnog servera.

**Osnovne komande**

1. Komanda **git log** se koristi da prikaže zabeleženu istoriju projekta do koje može da se dođe kada se iz poslednjeg komit snimka krene unazad. Podrazumevano će da prikaže samo istoriju grane na kojoj se trenutno nalazite, ali joj možete proslediti različitu, ili čak i više glava ili grana iz kojih treba da krene u obilazak. Takođe se često koristi za prikaz razlika između dve ili više grana na nivou komita.
2. Komanda **git revert** se može smatrati komandom tipa 'poništi', međutim, to nije tradicionalna operacija poništavanja. Umesto da ukloni urezivanje iz istorije projekta, on smišlja kako da invertuje promene unete urezivanjem i dodaje novo urezivanje sa rezultujućim inverznim sadržajem. Ovo sprečava Git da izgubi istoriju, što je važno za integritet vaše istorije revizija i za pouzdanu saradnju.
3. Komanda **git branch** je alat za upravljanje granama. Može da vam ispiše postojeće grane, kreira novu granu, obriše grane i promeni im ime.Veći deo grananje u programu Git je posvećen komandi branch koja se koristi kroz celo poglavlje. Najpre je uvodimo u pravljenje nove grane i kroz većinu njenih ostalih mogućnosti (ispis i brisanje) prolazimo u upravljanje granama.

U Grane za praćenje koristimo opciju git branch -u da postavimo granu za praćenje.

1. Komanda **git merge** se koristi za spajanje jedne ili više grana u granu koju ste odjavili. Zatim će pomeriti unapred tekuću granu na rezultat spajanja.
2. Komanda za uploadovanje izmena na server je **git push**
3. Komanda za preuzimanje izmena sa servera je **git pull**
4. **SSH** je mrezni protokol koji korisnicima omogucava uspostavljanje sigurnoog komunikacionog kanala izmedju dva racunara putem nesigurne racunarske mreze. Tradicionalni mrezni protokoli kao sto su TELNET,FTP,RSH.. iako su vrlo prakticni I jednostavni za koriscenje sadrze jedan veliki nedostatak. Radi se o vrlo niskom nivou sigurnosti koji je implementiran kod većine takozvanih „starijih“ mrežnih protokola, budući da se u vreme njihovog nastanka nije previše pažnje obraćalo na sigurnost računarskih sistema. U današnje vreme kada sigurnosni aspekt predstavlja jedan od najvažnijih elemenata svake računarske mreže taj problem svakodnevno dolazi sve više do izražaja.

Literatura:

<https://www.atlassian.com/git/tutorials/undoing-changes/git-revert>

[https://git-scm.com/book/sr/v2/Додатак-В%3А-Гит-команде-Гранање-и-спајање](https://git-scm.com/book/sr/v2/Додатак-В%253А-Гит-команде-Гранање-и-спајање)

https://www.mcloud.rs/blog/git-komande-za-pocetnike/