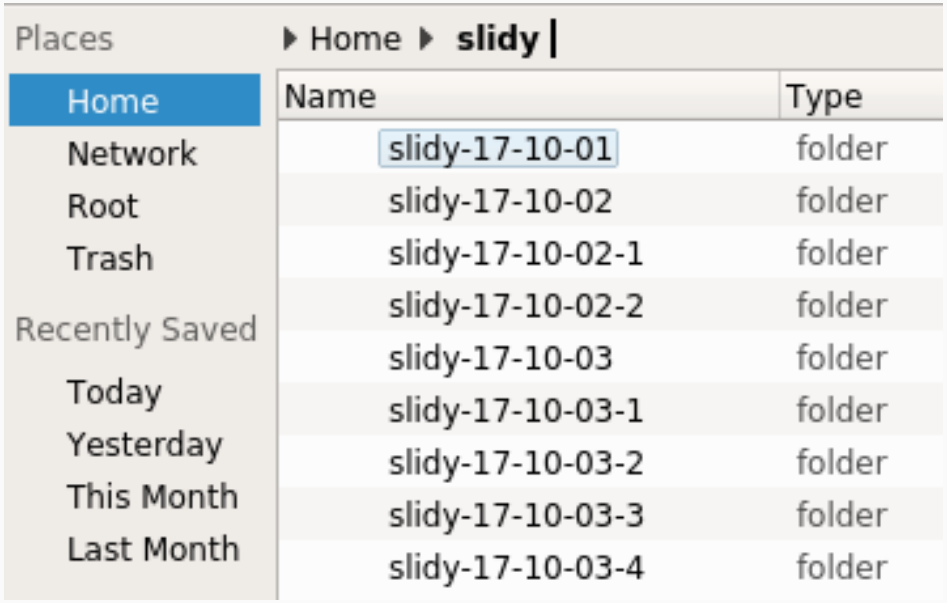
GIT

15. září 2017

FAKOS

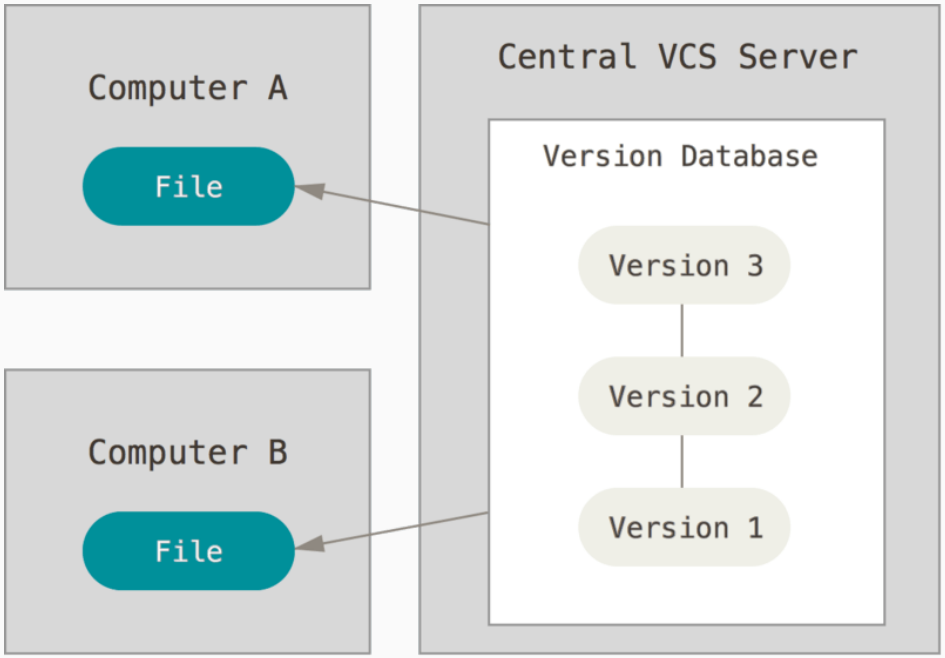
Verzované složky / lokální verzovací systém

- Jednoduché, ale neefektivní a nepraktické
- Jak spolupracovat s ostatními lidmi?



Centrální verzovací systém

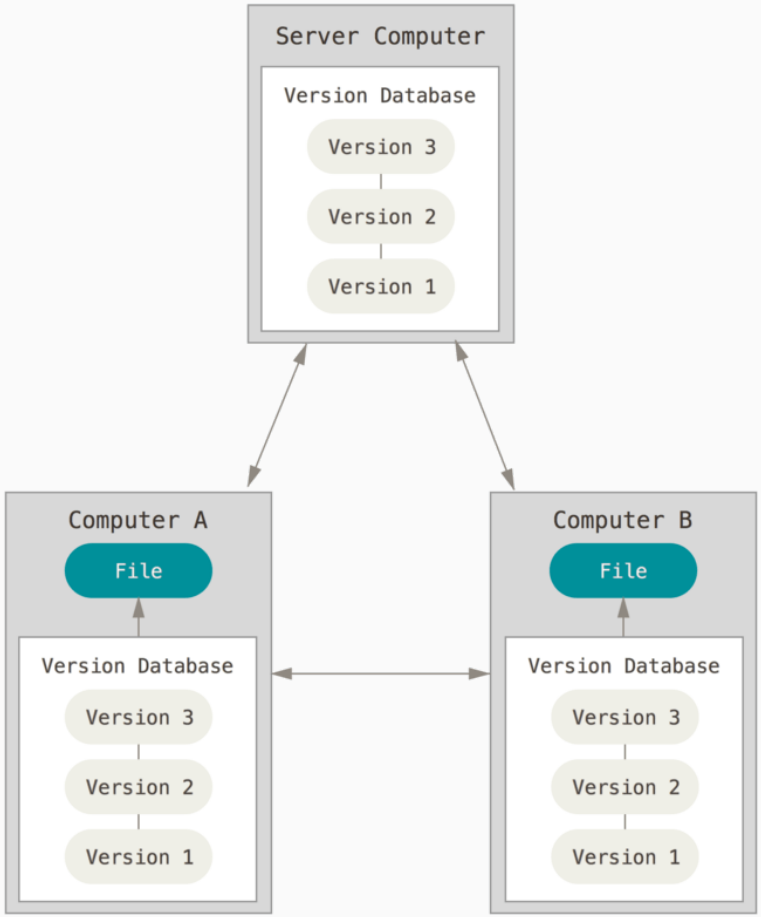
- Server je bottle-neck
- Co když není internetové spojení?



https://git-scm.com/book/en/v2/images/centralized.png

Distribuovaný verzovací systém

- Každý počítač (klient i server) má celou kopii repozitáře



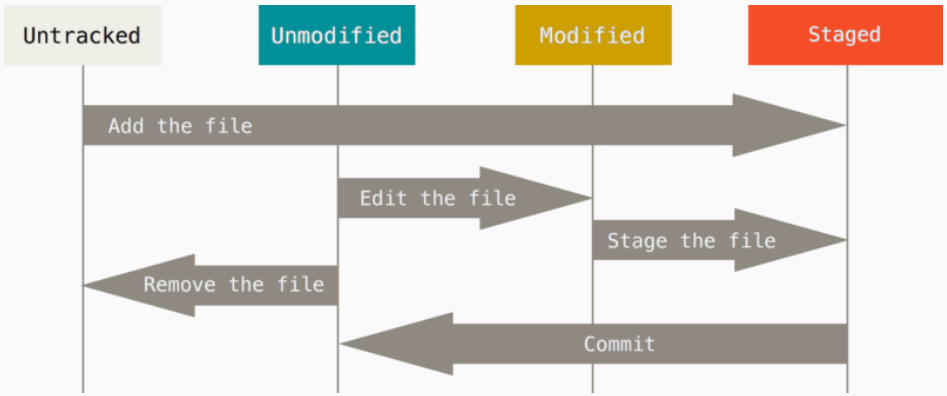
https://git-scm.com/book/en/v2/images/distributed.png

GIT

- Distribuovaný verzovací systém
- Většina operací jsou lokální
- Snímky souborového systému (“snapshot”), neukládá rozdíly
- “Commit often, perfect later, publish once.”

GIT – pozadí

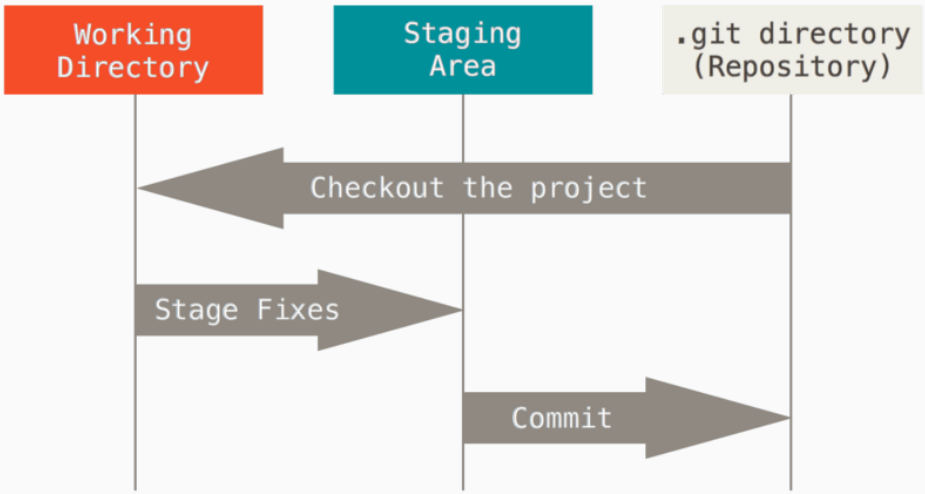
- Git sleduje danou složku a soubory v ní
- Soubor je buď sledovaný nebo nesledovaný (tracked/untracked)
- Každý sledovaný soubor může být ve třech stavech:
  - unmodified / modified / staged (indexed)



https://git-scm.com/book/en/v2/images/lifecycle.png

První tři příkazy

- `git init` – initializuj repozitář v prázdné složce
- `git add` – přidej obsah do následujícího commitu
- `git commit` – vytvoř commit



https://git-scm.com/book/en/v2/images/areas.png

Prohlížení změn a historie

- `git log`
- GUI nástroje: `tig`, `gitk`, `gitkraken`
- Zkusíte si s Vláďou ;)

Vrácení změn

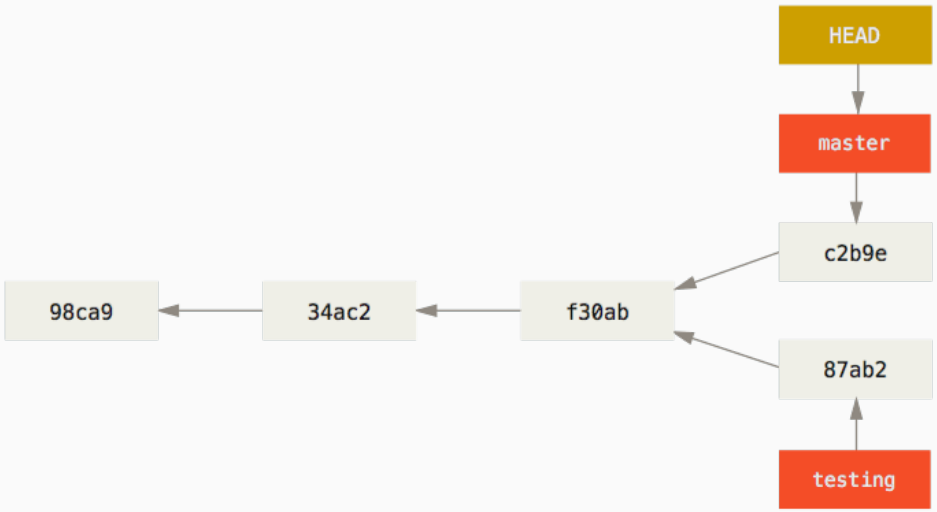
- `git reset HEAD` – odznač indexované změny (“staged” se stane “modified”)
- `git checkout -- <soubor>` – vrať všechny změny provedené na souboru
- `git commit --amend` – přidej indexovaný obsah do předchozího commitu

Vzdálené repozitáře

- `git clone` – stáhni novou kopii vzdáleného repozitáře
- `git pull` – stáhni commity ze vzdáleného repozitáře
- `git push` – nahraj svoje commity do vzdáleného repozitáře

Větve

- Způsob jak bezpečně upravovat obsah bez rozbití “publikovaných” dat
- Odštěpíme větev, provedeme změny, spojíme větve zpátky
- Větvení je levné – používejte často



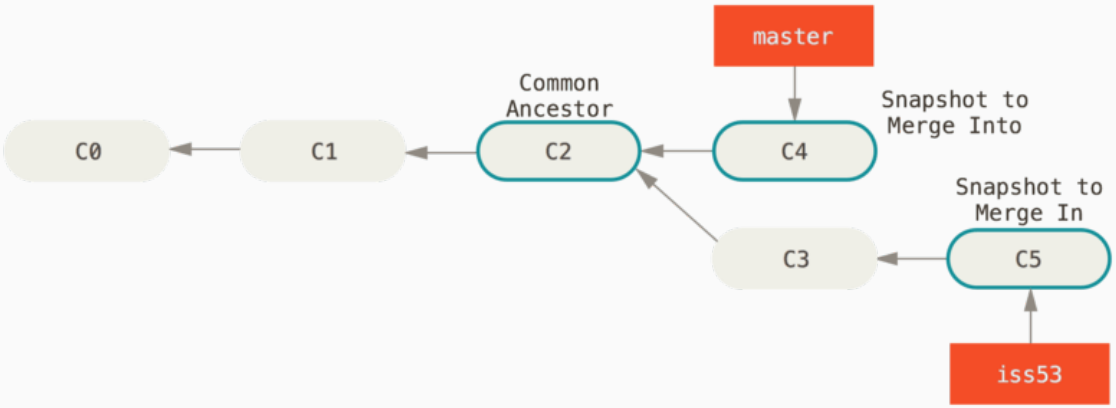
https://git-scm.com/book/en/v2/images/advance-master.png

Větve – pokračování

- `git checkout <nazev vetve>` – přepínání mezi větvemi
- `git branch <nazev vetve>` – vytvoření nové větve
- `git branch` – vypsání větví
- `git pull <repo> <vetev>` – stáhni commity z vetvě repozitáře `repo`
- `git push <repo> <vetev>` – pošli commity do větve v repozitáři `repo`

Větve – spojování

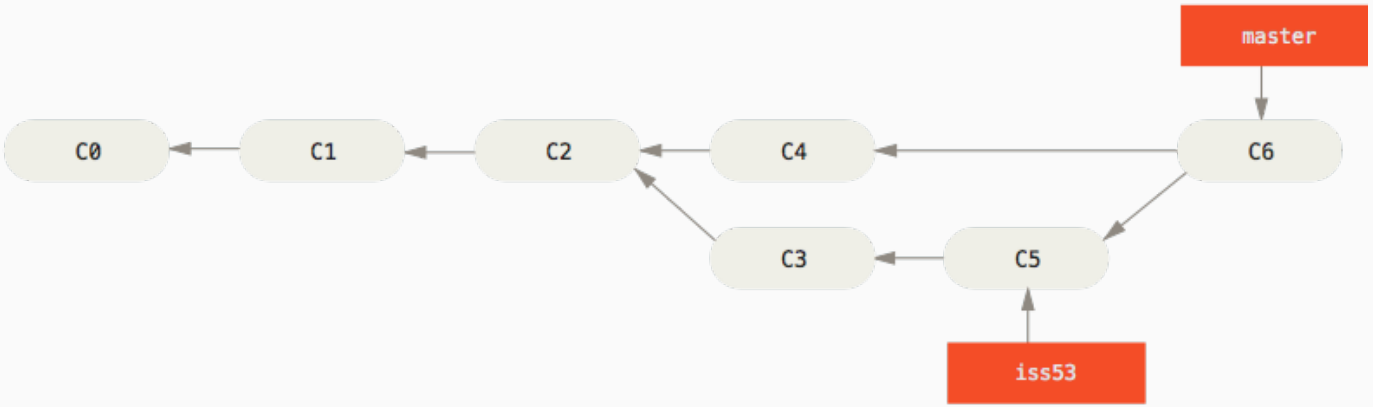
- `git merge <branch>` – spoj větev `branch` do aktualní větve
- Vytvoří se “merge commit”



https://git-scm.com/book/en/v2/images/basic-merging-1.png

Větve – spojování

- `git merge <branch>` – spoj větev `branch` do aktualní větve
- Vytvoří se “merge commit”



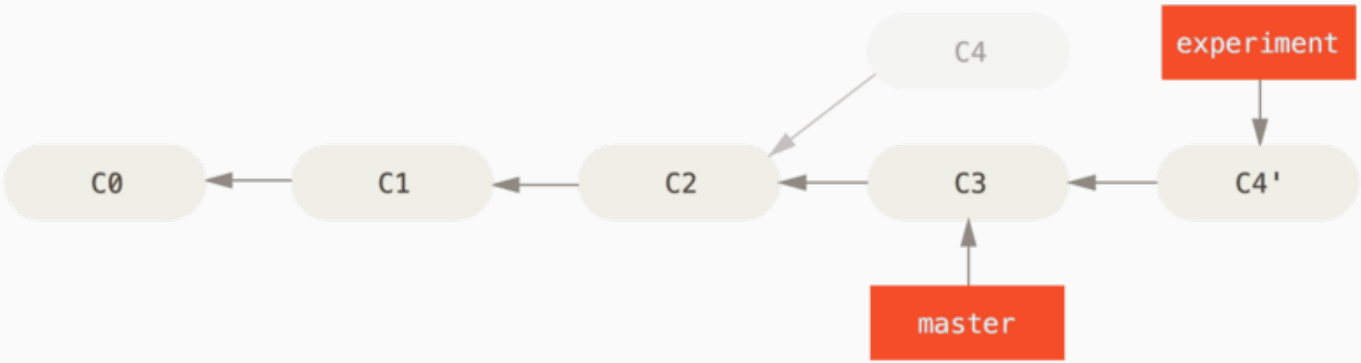
https://git-scm.com/book/en/v2/images/basic-merging-2.png

Větve – spojování

- Pokud je spojovaná větev přímým pokračováním aktuální větve, tak se nevytváří “merge commit” (provede se “fast-forward”)
- Jednotlivé commity se dají aplikovat pomocí `git cherry-pick`

Větve – rebase

- Alternativa k `merge`
- Přehrání změn (commitů) na danou větev



https://git-scm.com/book/en/v2/images/basic-rebase-3.png

Větve – rebase

- `git rebase <branch>` – na větev `branch` přehraj aktuální větev
- `git rebase <branch1> <branch2>` – na vět `branch1` přehraj větev `branch2`
- `git rebase --onto <branch1> <branch2> <branch3>` – na větev `branch1` přehraj větev `branch3`, ale až od místa, kde se odštěpila z `branch2`

Tagování

- Tag je (neměnný) ukazatel na určitý commit
- Chovají se podobně jako větve (pull/push/checkout)
- Může obsahovat dodatečné informace (kdo udělal tag, kdy, komentáře, . . .) – annotovaný tag
- `git tag <nazev>`
- `git tag -a <nazev> -m "komentar"`

Dobré praktiky

- Používat větve
- Konzistentní malé commity (jeden commit = jedna logická změna)
- Struktura commit zprávy – další forma dokumentace
- Neměnit již publikované commity
- Nedělat rebase s commity, které existují mimo lokální adresář

Zdroje

- Online kniha: https://git-scm.com/book/en/v2
- Cheatsheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf
- https://sethrobertson.github.io/GitBestPractices/