

Система контроля версий Git

- Git решает проблему: одна человека изменяет один файл — нужна синхронизация
- Сист. контроль версий:
 - блок
 - синхронизация разработчиков
 - UNDO
 - влагаем правки
 - Sandboxing
 - branching

Version Control Strategies

Lock-modify-unlock strategy

- разработчики не спешат
- блокируют файл для записи
- следующий разработчик
- одновременное обновление можно

Copy-modify-merge

- все разработчики могут одновременно править файл
- если версия на сервере и локальная конфликтуют → разрешение конфликта

Centralized VCS

- У каждого только отдельные файлы
- Проще

Distributed VCS

- У каждого все истории версий
- Сложнее, но быстрее.

Git

opensource

free

самая популярная VCS

быстрее других VCS

гарантия сохр. целостности файла

Хэширование

могут создавать много веток

это называем создавать
разные схемы управления
проектами.

Никогда не шарят private key. — Он
подписывается commits

команды:

git push — где комба залог.

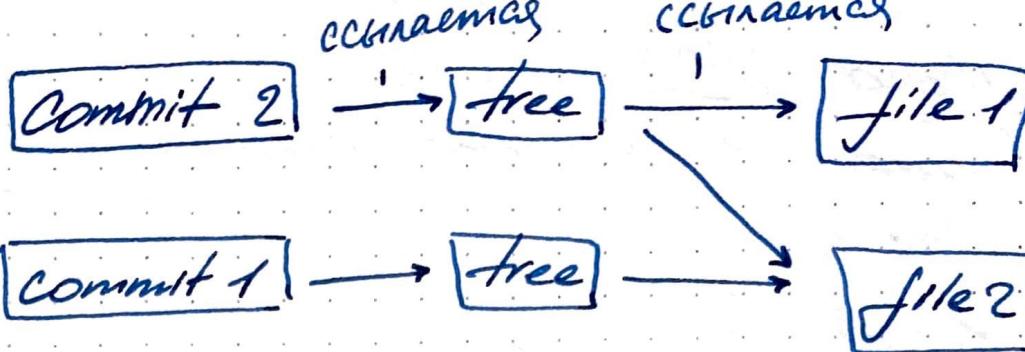
gitk — где отображиваются
истории коммитов

HEAD указывает на последний коммит

Объекты git: :commit

:tree

:blob — Binary Large
Object —
макс. еденица в git



деревья состоят из один или
больше blob

git все построено на графах

commit может содержать и файлы,

команды и директории

ssh-keygen - создание пары ssh
private, public

git config - конфигурация

cd - переход в папку

git clone - копирование репозитория
в текущую папку

ls - выводить содержимое текущей
папки

git remote -v - выводить список
данных

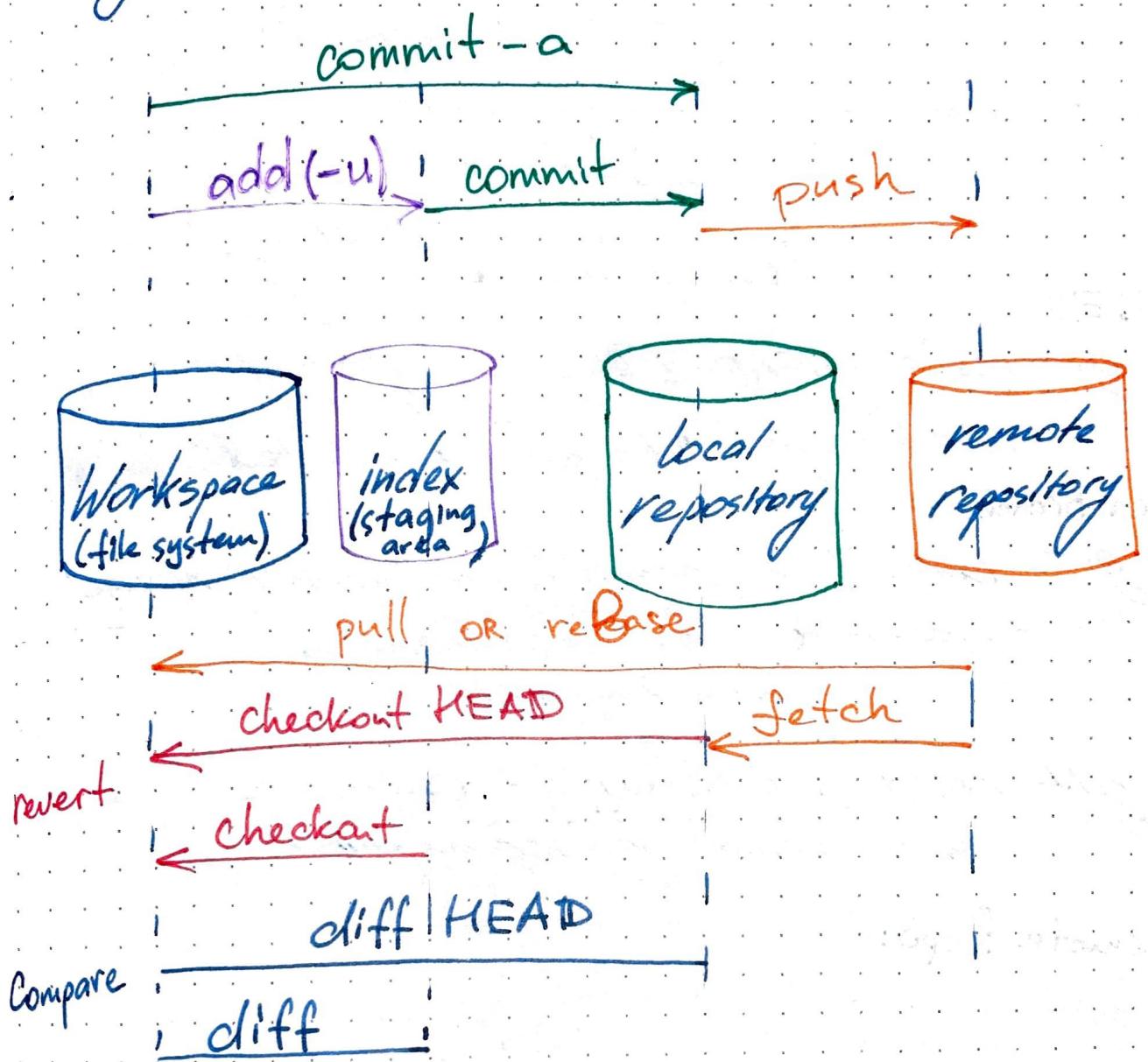
git status - состояние синхро-
низации текущей папки
с репо на сервере.
изменение/новых/измененных

git add -
добавить файлы
в коммит

git commit - произвести
(на локальном) коммит

git log - выводит историю коммитов
git push - сбрасывает коммит на сервере

git data transport commands



git show - подробная информация о коммите или

git ls-tree - инфом. о дереве

В .git/objects ищута пакеты
это не более 2 символов в sha1

Undoing Changes

FILE SYSTEM:

- git checkout - отмена изменений на локальном уровне (Working directory)
 - подсвечены все файлы в текущем рабочем каталоге
- git clean -f - удаление всех новых файлов в Working directory
 - Необходимо

INDEX:

- git reset HEAD <file> - удаляем файл из "to be committed"

Local Branch:

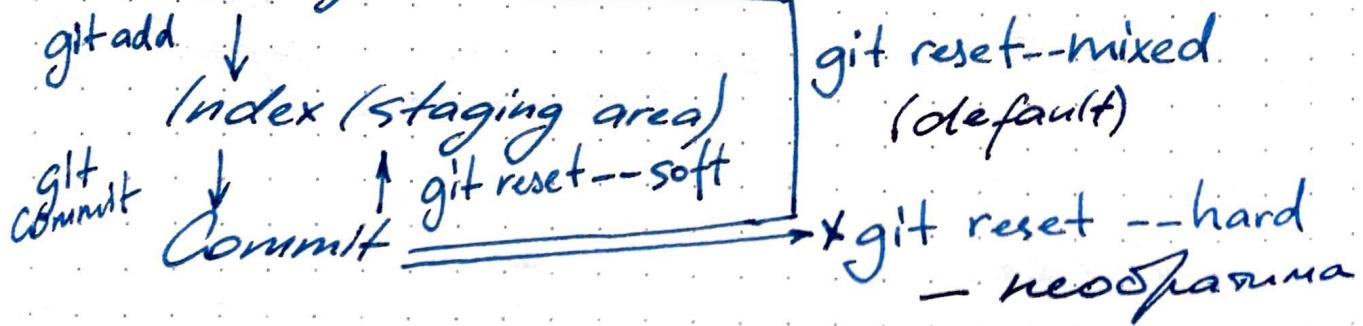
- git commit --amend -m "commit message"
 - изменение последующей коммитной
- git reset HEAD~n - отмена последних n коммитов

Remote Repo:

- git revert

Git Reset

File System



През ".(морма) - Всегда все

Gitignore

gitignore - файл назначенный указать, какие файлы не должны отображаться в базе git.

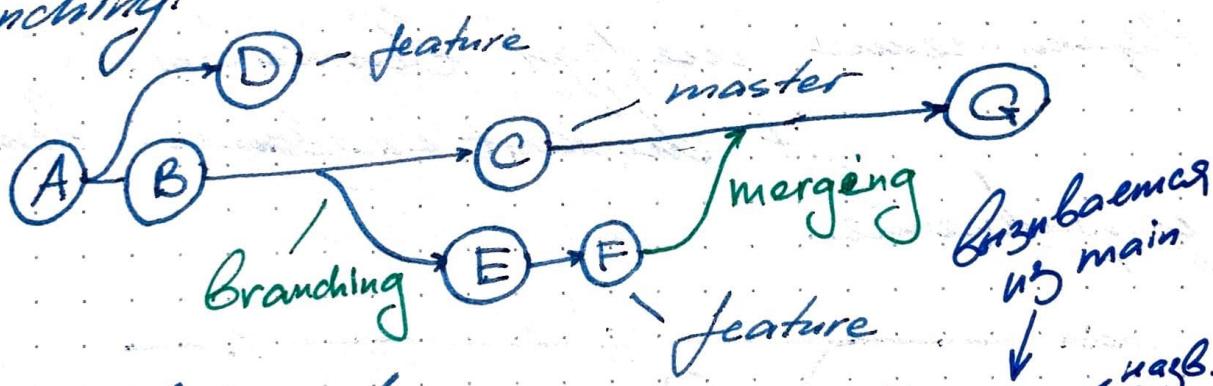
Branching and Merge

parent
child

Каждый коммит считается на git носителем предыдущего коммита и имеет предыдущий коммит в качестве родителя.

git назначает коммиту родительский коммит

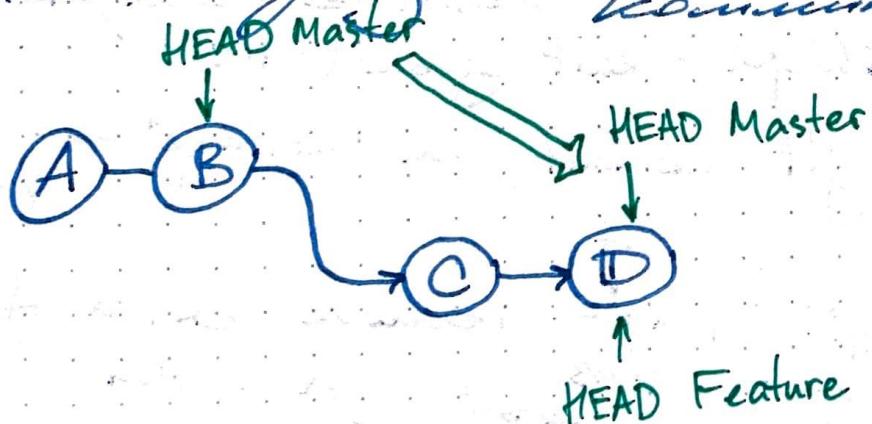
Branching:



Fast-forward merge: команда: git merge <бета>

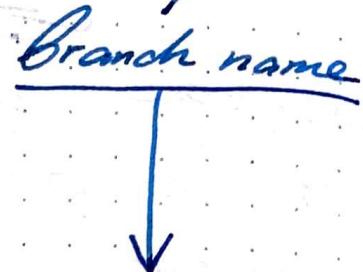
Head master переименовывает на
меню head feature

HEAD - указывает на последний
коммит

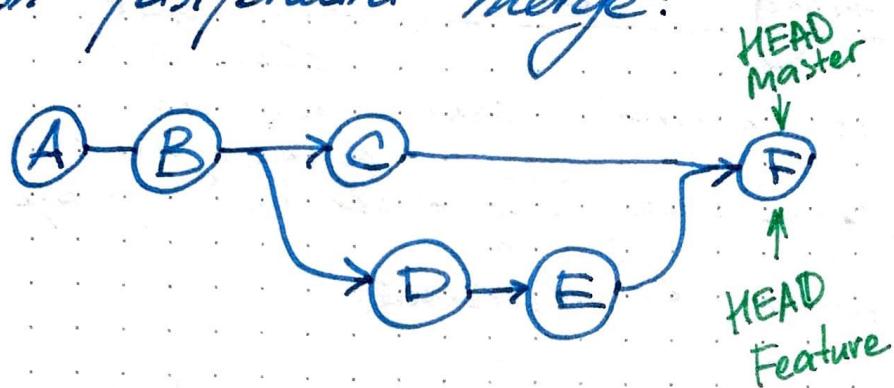


Branching: git checkout -b feature
command: →

- cocaine
branch
"feature"



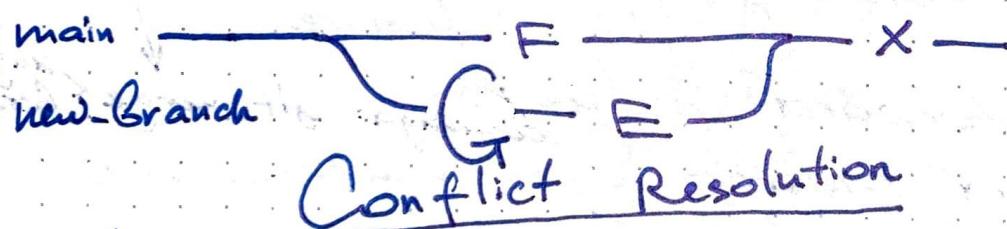
merging command: git merge feature
non-fastforward merge:



Reparenting merge branch:

git checkout <newname> branch

represents:



git merge tool — command to
resolve conflicts

· :wqa — exit from VIM

· git commit -m "message" — commit
merge

· git clean — remove extra files

pagnum merge tools
use pagnum documents from vim

diff tool

· git checkout --ours ← omaburz bepuso grywus

· git checkout --theirs ← omaburz bepuso obos

LLL HEAD - укаживаем на main бенч

Conflict Avoidance

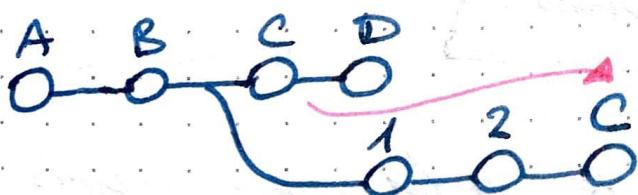
- small commits
- no edits to whitespaces, only necessary edits
- often merging

Rebase



- Hence generate Rebase ~~сомните~~^{publ/c} branch
main main. Delete
- Rebase можно организовать
feature-branch

Cherry Pick



Tags

- menu - номенклатура коммитов
- команда git tag <name>
нужно использовать библиотеку ssh
для обмена с аутентифицированным
- git push --tags - push tags to
REMOTE

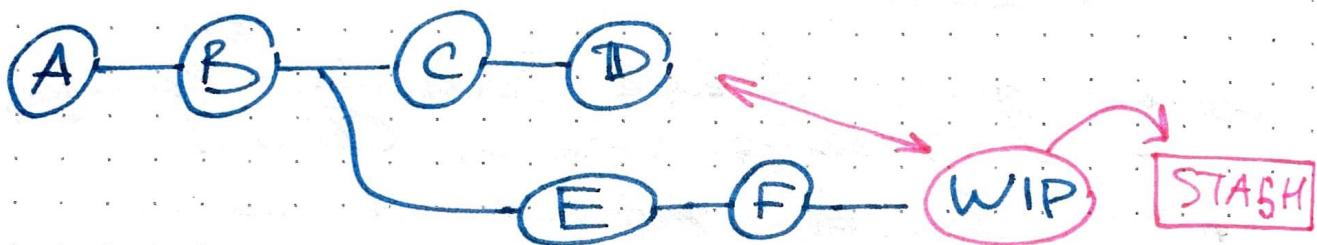
Stashing

Stash - скріншот змін

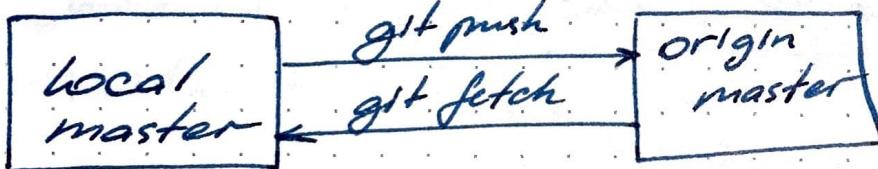
Зберігання змін у змінний

Stashing - зберігання неповністю виконаних змін на певний час в відокремленому від work in progress чи feature-бранче

Команда git stash



Remotes



Add remote: `git remote add <name> <url>`

View remotes: `git remote -v`

Remove remote: `git remote remove`

`git push` - синхронизація більш ранніх змін на певному Remote

Branching Strategies

Centralized strategy:

master



— Alex' commit

Ivan's
commit

Feature-Branch workflow



large
feature

small
feature

Gitflow

• Каминнок върху feature-бранч
нужен бъмб develop.

• Бъмбъ master нужен за
комбиниране във production release

Integration Manager Workflow

• Разработчици pull върху "Blessed repo".

• В "Blessed repo" push наструи
съзидателният бранч
integration manager, който pull

из public repos разработчиков

