

# Системы контроля версий

## Инструментарий Современного Программиста

Иван Трепаков

NSU Sys.Pro

## Индивидуальная разработка

- Регулярные backup'ы
- История изменений
- Определение правок, на которых что-то сломалось
- Эксперименты в отдельных ветках
- Переносимость окружения разработки

## Командная разработка

- Несколько человек работают с одной кодовой базой
- Разрешение конфликтов
- Ревью изменений
- Выяснение кто, а главное *зачем*, внес какое-то изменение
- Формирование changelog'ов

Система контроля версий  
Version control system, VCS

## Система контроля версий

Version control system, VCS

- **Изменение / changeset / patch:** добавление или удаление строк

a.sh
-echo "Hello World!" +echo "Hello Sys.Pro!"

## Система контроля версий

Version control system, VCS

- **Изменение / changeset / patch**: добавление или удаление строк
- **Коммит / версия / ревизия**: набор изменений

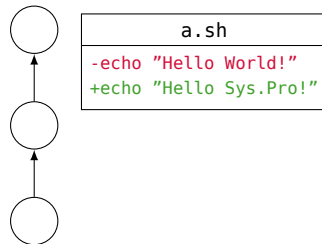


a.sh
<pre>-echo "Hello World!" +echo "Hello Sys.Pro!"</pre>

## Система контроля версий

Version control system, VCS

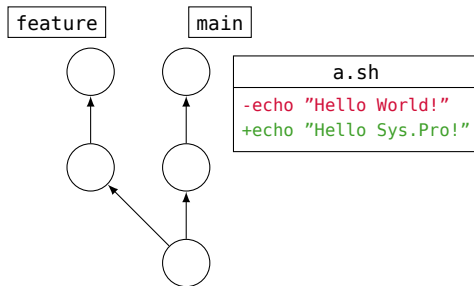
- **Изменение / changeset / patch**: добавление или удаление строк
- **Коммит / версия / ревизия**: набор изменений
- **История**: последовательность коммитов



## Система контроля версий

Version control system, VCS

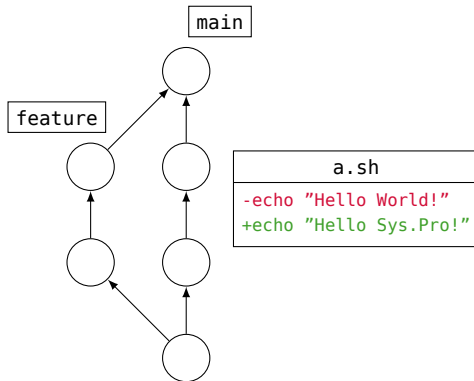
- **Изменение / changeset / patch**: добавление или удаление строк
- **Коммит / версия / ревизия**: набор изменений
- **История**: последовательность коммитов
- **Ветка / branch**: история относящаяся к конкретной задаче / фиче



## Система контроля версий

Version control system, VCS

- **Изменение / changeset / patch**: добавление или удаление строк
- **Коммит / версия / ревизия**: набор изменений
- **История**: последовательность коммитов
- **Ветка / branch**: история относящаяся к конкретной задаче / фиче
- **Слияние**: объединение историй двух веток в одну
- **Конфликт**: изменения в одной и той же строке при слиянии

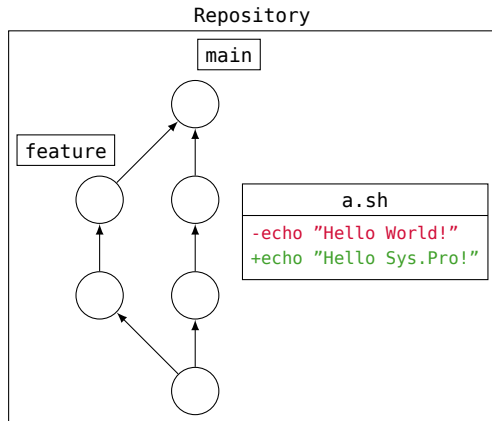




## Система контроля версий

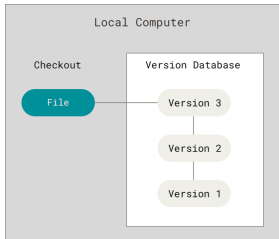
Version control system, VCS

- **Изменение / changeset / patch:** добавление или удаление строк
- **Коммит / версия / ревизия:** набор изменений
- **История:** последовательность коммитов
- **Ветка / branch:** история относящаяся к конкретной задаче / фиче
- **Слияние:** объединение историй двух веток в одну
- **Конфликт:** изменения в одной и той же строке при слиянии
- **Репозиторий:** база данных всех изменений

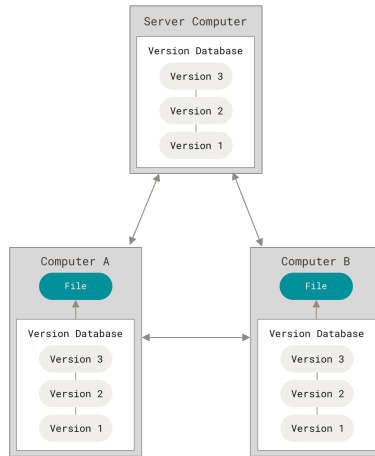


# Системы контроля версий

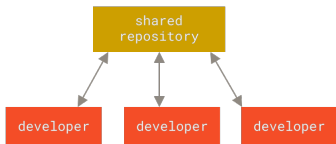
## Локальные



## Распределенные



## Централизованные



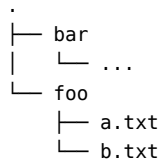
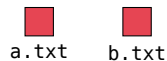
## Хранение данных

- Хранится не разница (**diff**) файлов, а полный снимок (**snapshot**) файловой структуры в виде *дерева*

```
.
├── bar
│   └── ...
└── foo
    ├── a.txt
    └── b.txt
```

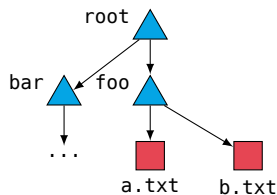
## Хранение данных

- Хранится не разница (**diff**) файлов, а полный снимок (**snapshot**) файловой структуры в виде *дерева*
- **Blob**: содержимое одного файла



## Хранение данных

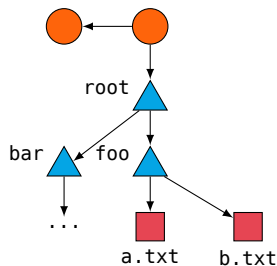
- Хранится не разница (**diff**) файлов, а полный снимок (**snapshot**) файловой структуры в виде *дерева*
- **Blob**: содержимое одного файла
- **Tree**: директория указывающая на blob'ы, находящихся в ней файлов и на tree поддиректорий



```
.
├── bar
├── ...
└── foo
    ├── a.txt
    └── b.txt
```

## Хранение данных

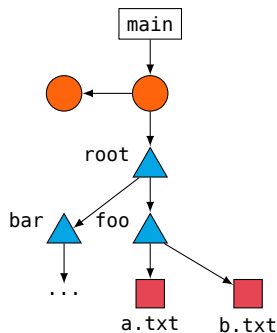
- Хранится не разница (**diff**) файлов, а полный снимок (**snapshot**) файловой структуры в виде *дерева*
- **Blob**: содержимое одного файла
- **Tree**: директория указывающая на blob'ы, находящихся в ней файлов и на tree поддиректорий
- **Коммит**: конкретный снимок, указывающий на соответствующий root tree и на предыдущий коммит в истории



```
.
├── bar
│   └── ...
└── foo
    ├── a.txt
    └── b.txt
```

## Хранение данных

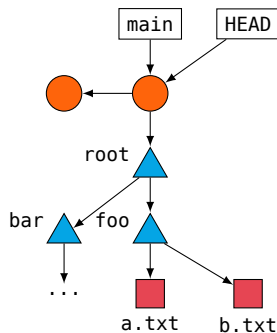
- Хранится не разница (**diff**) файлов, а полный снимок (**snapshot**) файловой структуры в виде *дерева*
- **Blob**: содержимое одного файла
- **Tree**: директория указывающая на blob'ы, находящихся в ней файлов и на tree поддиректорий
- **Коммит**: конкретный снимок, указывающий на соответствующий root tree и на предыдущий коммит в истории
- **Ветка**: указатель на конкретный коммит
- **HEAD**: указатель на *текущий* коммит



```
.  
├── bar  
│   └── ...  
└── foo  
    ├── a.txt  
    └── b.txt
```

## Хранение данных

- Хранится не разница (**diff**) файлов, а полный снимок (**snapshot**) файловой структуры в виде *дерева*
- **Blob**: содержимое одного файла
- **Tree**: директория указывающая на blob'ы, находящихся в ней файлов и на tree поддиректорий
- **Коммит**: конкретный снимок, указывающий на соответствующий root tree и на предыдущий коммит в истории
- **Ветка**: указатель на конкретный коммит
- **HEAD**: указатель на *текущий* коммит

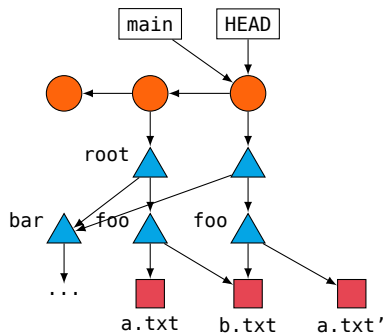


```
.  
├── bar  
│   └── ...  
└── foo  
    ├── a.txt  
    └── b.txt
```



## Хранение данных

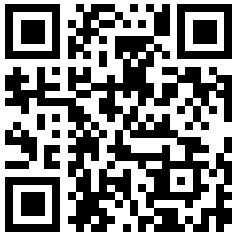
- Хранится не разница (**diff**) файлов, а полный снимок (**snapshot**) файловой структуры в виде *дерева*
- **Blob**: содержимое одного файла
- **Tree**: директория указывающая на blob'ы, находящихся в ней файлов и на tree поддиректорий
- **Коммит**: конкретный снимок, указывающий на соответствующий root tree и на предыдущий коммит в истории
- **Ветка**: указатель на конкретный коммит
- **HEAD**: указатель на *текущий* коммит



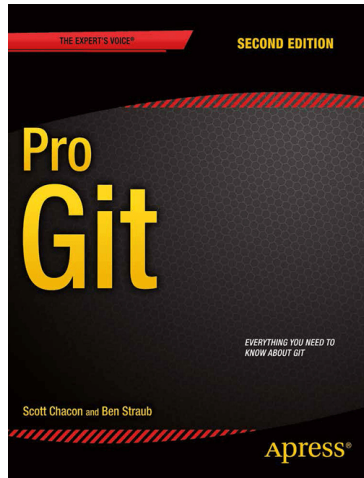
```
.
├── bar
│   └── ...
└── foo
    ├── a.txt
    └── b.txt
```

## Pro Git

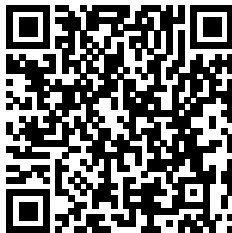
Основополагающая книга по работе с Git и его внутреннему устройству



<https://git-scm.com/book/en/v2>



## Pro Git Глава 3. Git Branching



<https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

- `git init / git clone`
- `git status`
- `git log`
- `git add`
- `git restore`
- `git commit`
- `git push`
- `git switch / git checkout`
- `git merge`
- `git cherry-pick`
- `git rebase --interactive`
- `git remote ...`

# Git config

- ~/.gitconfig

```
[user]
```

```
email = <email>
```

```
name = <first-name> <last-name>
```

```
[diff]
```

```
tool = vimdiff
```

```
[merge]
```

```
conflictstyle = zdiff3
```

```
tool = vimdiff
```

```
[mergetool]
```

```
keepBackup = false
```

```
[alias]
```

```
st = status
```

```
graph = log --graph --oneline --decorate --all
```

```
...
```

- git config set --global <option> <value>

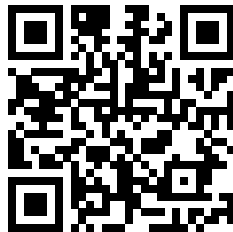
## TUI

- `git log --graph --oneline --decorate`
- `vimdiff`
- `delta`
- `lazygit`

## GUI

- `gitk + git gui`
- Sublime Merge
- GitKraken

## Git clients



<https://git-scm.com/downloads/guis>

Q&A