

# **Технология и методы программирования**

Лекция 3. Системы контроля версий. Git.

# Определение

Система контроля версий (Version control system) — ПО позволяющее хранить разные версии файлов и директорий

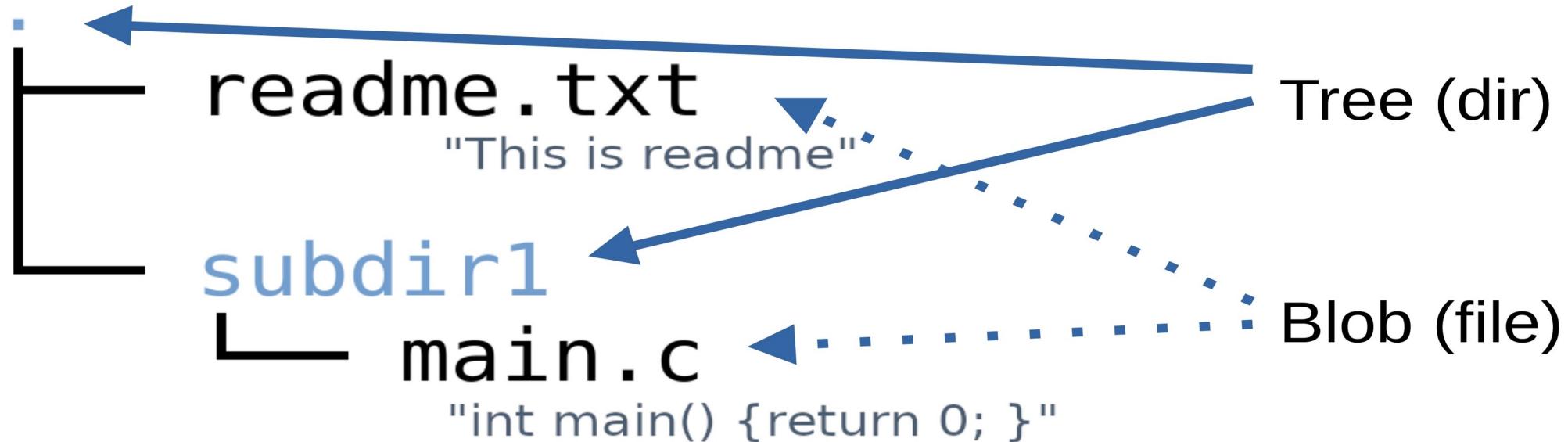
# Где применяются?

- Исходники
- Конфиг-файлы
- XML-документы
- Любые другие текстовые файлы

# Зачем?

- Совместная разработка
- Безопасность (нельзя незаметно изменить прошлые коммиты, можно подписывать коммиты ЭЦП)
- Анализ истории изменений проекта
- Параллельная работа над частями проекта
- Решение конфликтов при изменении одного файла

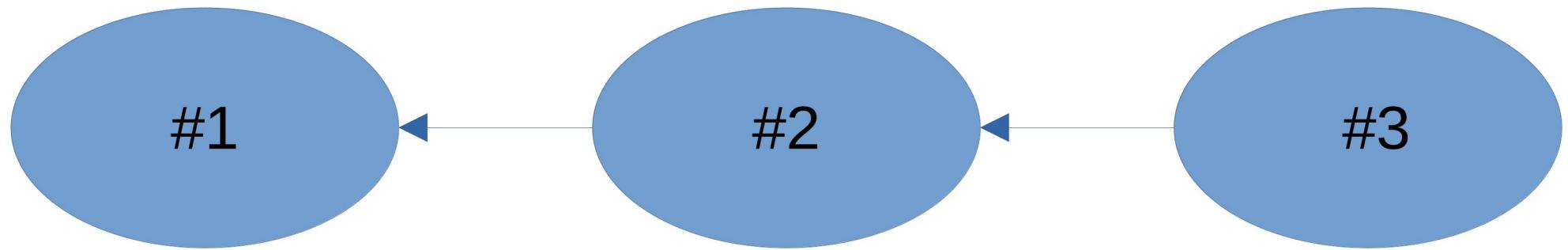
# Терминология git. Файловая система



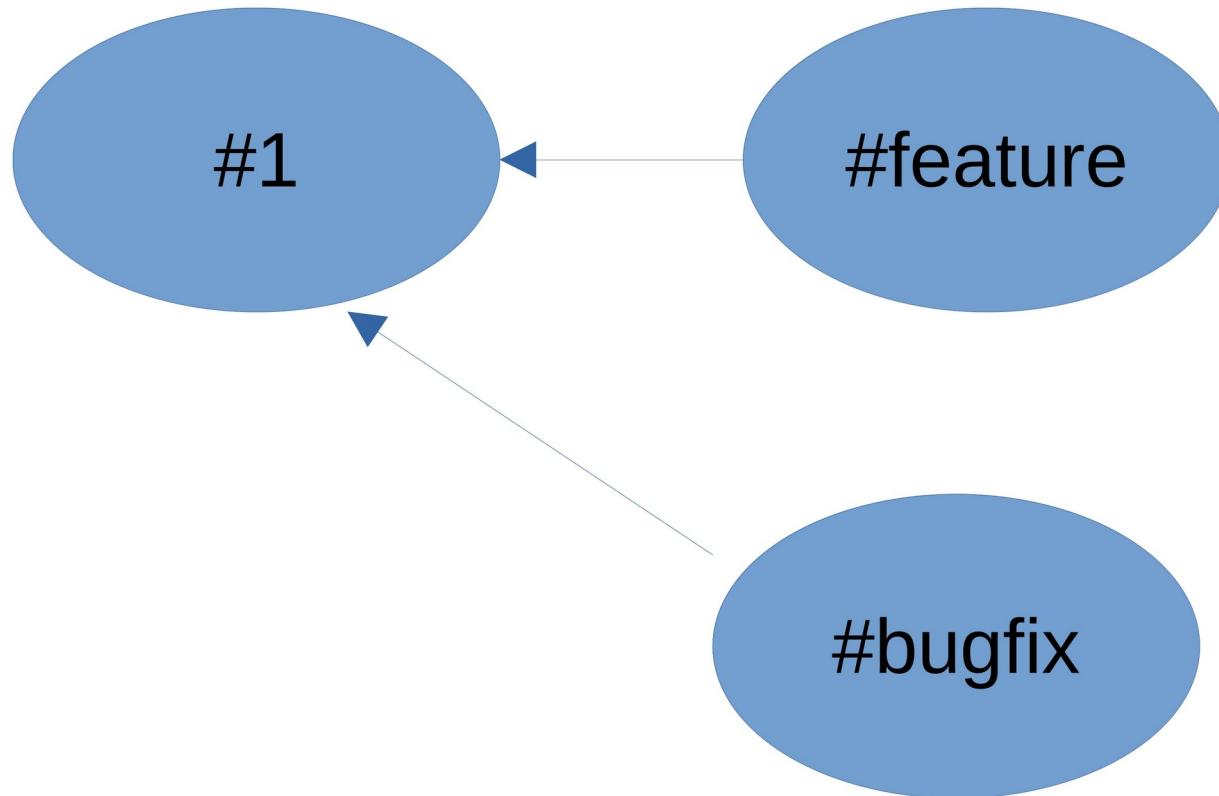
# Определения v2

- Снапшот (Snapshot) — снимок состояния отслеживаемых директорий и файлов
- Коммит (Commit) — процедура создания нового снапшота
- Дерево (Tree) — директория в git репозитории
- Блоб (Blob) — файл
- Ветка (Branch) — последовательность снапшотов

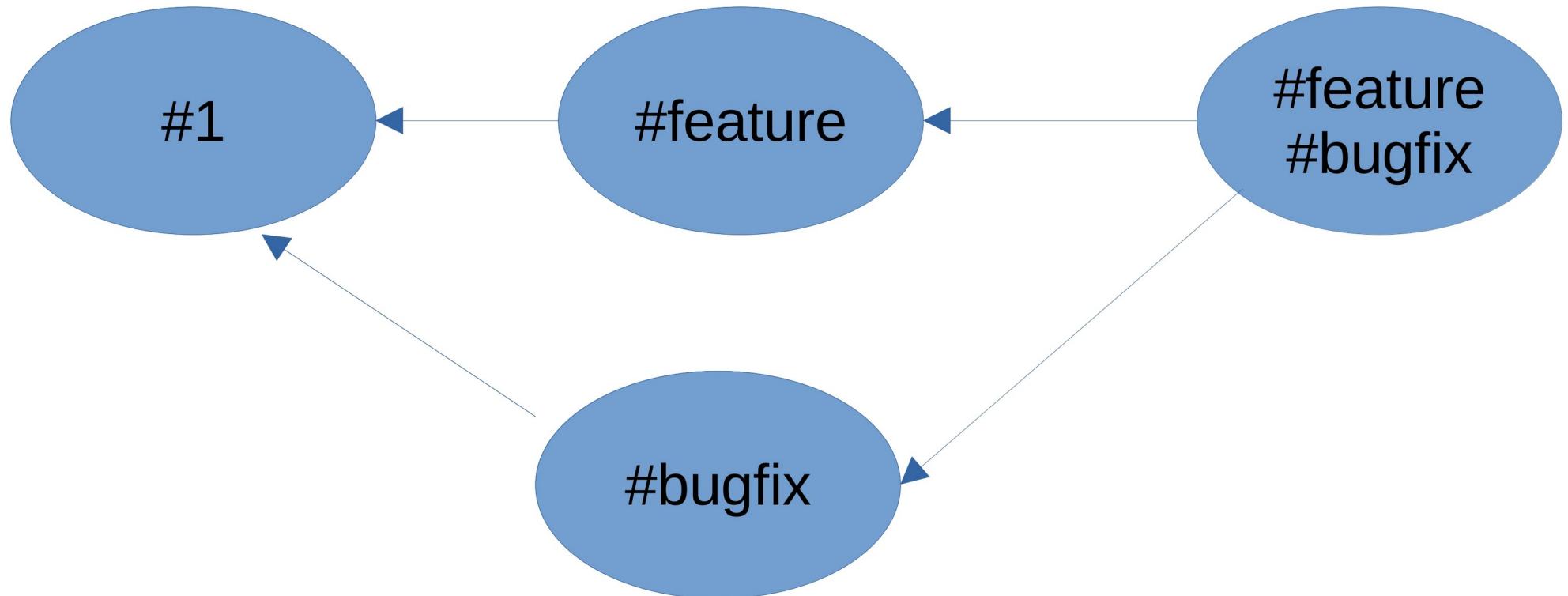
# Снапшоты



# Снапшоты. fork



# Снапшоты.Merge



# Git. Internals 1

```
type blob = array<byte>

type tree = map<string, tree | blob>

type commit = struct {
    parents : array<commit>
    author : string,
    message : string,
    snapshot: tree
}

type object = blob | tree | commit
```



# Git. Internals 2

```
type objectStore = map<string, object>

def store(o):
    id = sha1(o)
    storage[id] = o

def load(id):
    return storage[id]
```

# Git. Internals 3

```
references = map<string, string>
```

# Demo

- Git шпаргалка (на русском)

# Доп информация

- Про git — очень подробная книжка. Рекомендуется полистать первые 5 глав.
- [Youtube. Missing semester. Git tuto](#)
- [Git gui clients list](#)

# Задачки

- [https://learngitbranching.js.org/?locale=ru\\_RU](https://learngitbranching.js.org/?locale=ru_RU)

Склонировать <https://github.com/git/git.git>

- Кто и когда создал файл entry.h?
- Найти файл, где **определена** struct tree. Выяснить кто и когда модифицировал её в последний раз.
- Показать top 5 пользователей, которые сделали больше всех коммитов (\*)

**The end**