Система контроля версий Git

Преподаватель
Пинежанин Евгений
Сергеевич

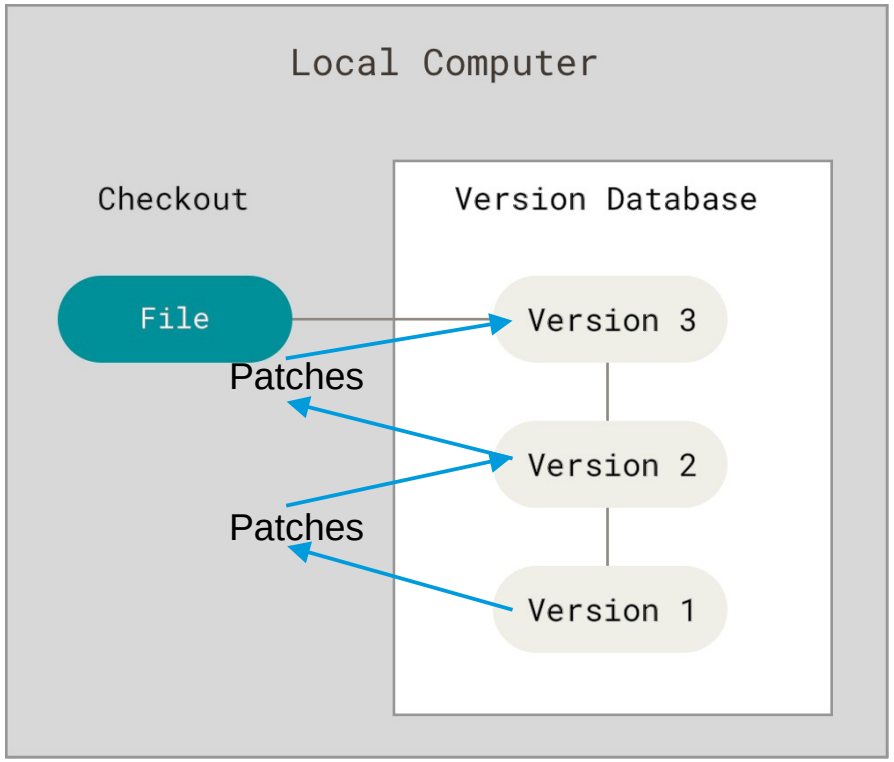
Нижний Новгород
2025г.

Локальные системы контроля версий

Система контроля версий (VCS) — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии.

Локальная VCS — база данных, которая хранит записи о всех изменениях в файлах, осуществляя тем самым контроль ревизий.

Патч (англ. **Patch**) — различия между файлами

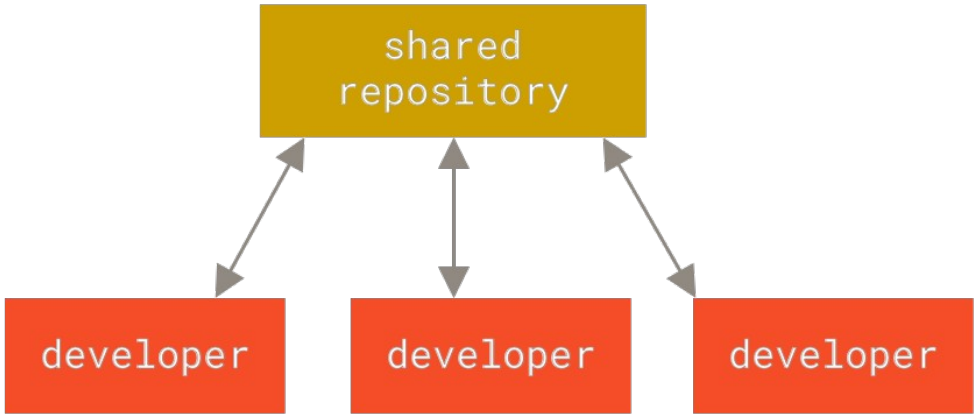


Централизованные системы контроля версий

Централизованные системы контроля версий (**CVCS**) используют единственный сервер, содержащий все версии файлов, и некоторое количество клиентов, которые получают файлы из этого централизованного хранилища.

Плюсы: Все разработчики проекта знают, чем занимается каждый из них. Администраторы имеют контроль над тем, кто и что может делать, им гораздо проще администрировать CVCS, чем оперировать локальными базами.

Минусы: Если сервер выйдет из строя, то в течение этого времени никто не сможет использовать контроль версий для сохранения изменений, над которыми работает.

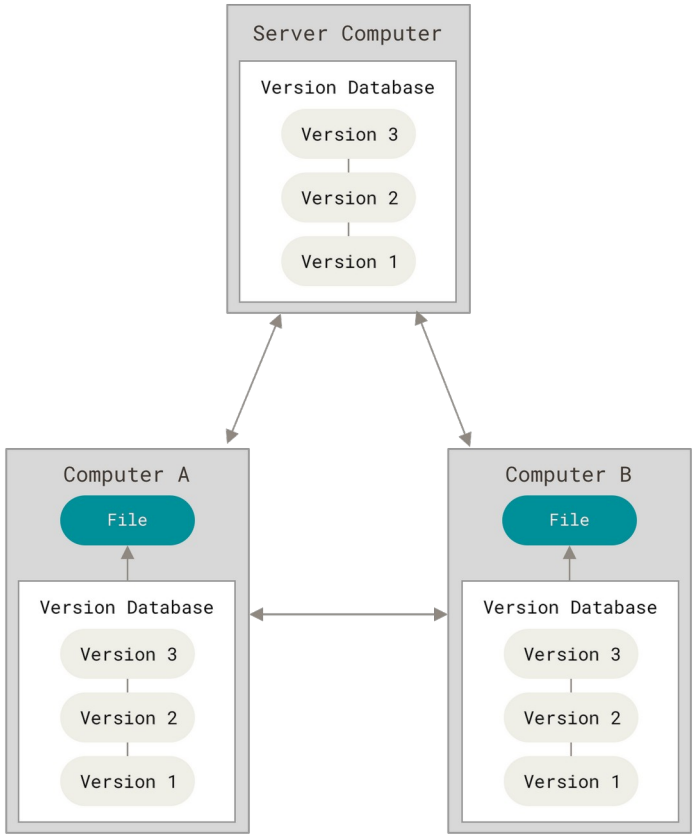


Распределённые системы контроля версий

В **распределённых системах контроля версий** (**DVCS**) клиенты не просто скачивают снимок всех файлов (состояние файлов на определённый момент времени) — они полностью копируют репозиторий.

В случае, если один из серверов, через который разработчики обменивались данными, умрёт, любой клиентский репозиторий может быть скопирован на другой сервер для продолжения работы. Каждая копия репозитория является полным бэкапом всех данных.

Git — распределённая системах контроля версий

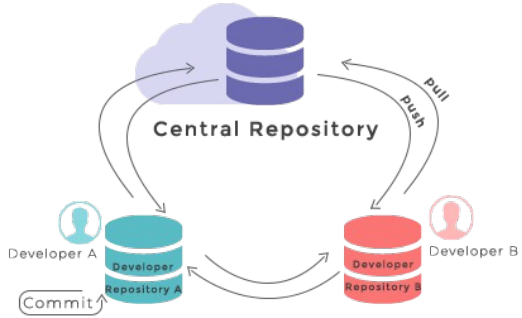


Основные понятия в Git

Репозиторий (от англ. **repository — «хранилище»**) — место, где хранятся и поддерживаются какие-либо данные.

Коммит (от англ. **commit — «зафиксировать»**) в системе контроля версий Git — это запись изменений, внесённых в репозиторий.

Ветка (англ. **branch**) — это последовательность коммитов, которые упорядочены по времени.



У **Git** есть три основных состояния, в которых могут находиться ваши файлы: изменён (**modified**), индексирован (**staged**) и зафиксирован (**committed**):

Изменённые файлы, те которые поменялись, но ещё не были зафиксированы.
Индексированный — это изменённый файл в его текущей версии, отмеченный для включения в следующий коммит.
Зафиксированный значит, что файл уже сохранён в вашей локальной базе.

Начало работы с Git

0. Установка Git на компьютер (https://git-scm.com/downloads/win)
1. Инциализация репозитория:

Создание репозитория (**git init**)

Клонирование репозитория (**git clone <path>**)

2. Добавление кода
3. Индексация кода (**git add “files”**)
4. Создание коммита (**git commit -m "Description of your changes"**)
5. Если итоговая задача не решена то снова переходим к шагу 2

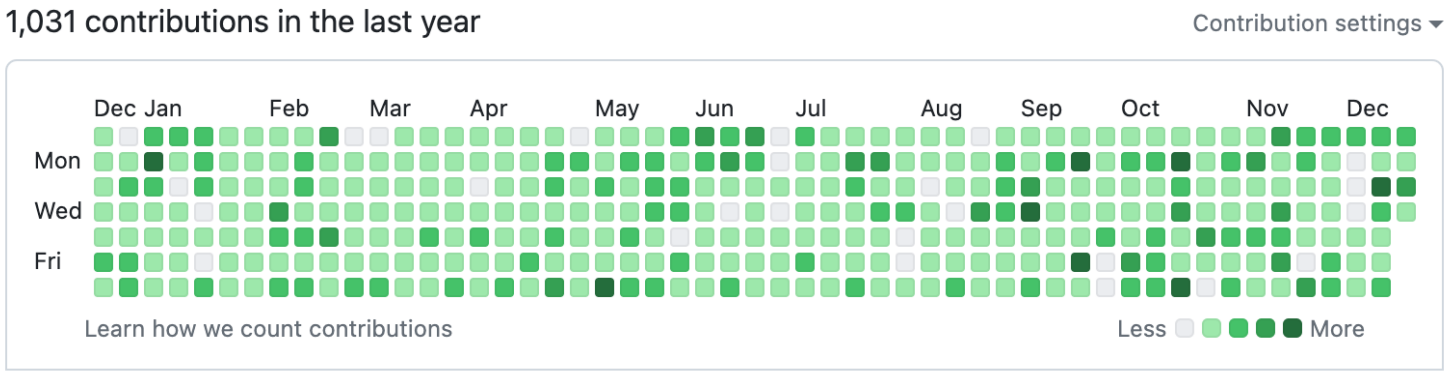
git status — команда для просмотра статуса файлов в репозитории

Git Bash — интерфейс командной строки (CLI), который сочетает систему контроля версий Git с интерфейсом в стиле Unix для Windows. Рекомендую для использования

GitHub

Сам по себе Git не так эффективен, если вы работаете с разработчиками, которые не находятся с вами в одном офисе. Необходимо хранить репозитории на сервере, который доступен из любой точки мира. С этим помогают веб-сервисы для хостинга IT-проектов.

Примером такого веб-сервиса является **GitHub**. В рамках нашего курса он будет использоваться для сдачи задач и лабораторных работ.



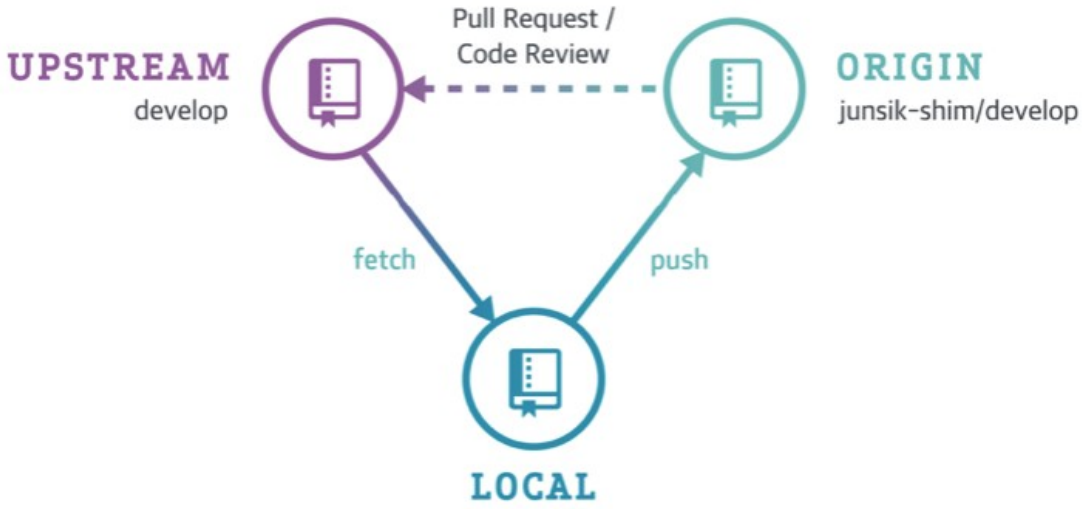
Основные понятия в GitHub

Fork (**форк**) — это создание копии репозитория (набора файлов и истории изменений) в аккаунте пользователя. Это позволяет работать над проектом независимо от оригинала, вносить изменения и предлагать их через запрос на включение (**pull request**)

upstream — это обозначение для исходного репозитория, от которого сделан форк на GitHub

origin в свою очередь указывает на форк репозитория **upstream**

Pull request (**PR**) на **GitHub** — это запрос на слияние изменений в основной код проекта. Он используется для представления изменений (нового кода, исправлений ошибок или улучшений) другим членам команды для проверки, обсуждения и одобрения.

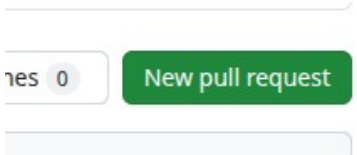


Начало работы с GitHub

0.1 Регистрация на платформе **GitHub** (https://github.com)
0.2 Задание имени пользования в **Git**

git config --global user.name "your_nickname"
git config --global user.email "mail@email.com"

1. Если вы работаете с чужим репозиторием, сделайте **fork** к себе
2. … Работа внутри локального репозитория ...
3. После после работы с локальным репозиторием переносим результаты на сервер
   3.1 Для избежания конфликтов сначала копируем все изменения с удаленной ветки в локальную: **git pull**
   3.2 Затем из локальной на удаленную: **git push <branch>**
4. При необходимости открыть Pull request в upstream репозиторий



Более подробное обсуждение Git и GitHub будет после начала работы над лабораторными задачами

Благодарю за внимание!