

Структура научной презентации

Простейший шаблон

Поляков Г. С.

2023-02-18

Российский университет дружбы народов, Москва, Россия

Информация

Поляков Глеб Сергеевич

НПИБд-02-22

РУДН, Москва, Россия

```
# Цель работы
```

```
Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.
```

Рис. 1: Добавление цели работы

Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

Рис. 2: Добавление задания

Добавление теоретического введения

```
# Теоретическое введение
## Системы контроля версий. Общие понятия

Система контроля версий (Version Control System, VCS) применяется при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удаленном репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведенные разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определенных команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может создавать не полную версию измененных файлов, а производить так называемую дельта-компрессию – сохранять только изменения между последовательными версиями, что позволяет уменьшать объем хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединять (сплит) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменений. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы. ОС обеспечивает таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохранять общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветки. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределенных системах контроля версий центральный репозиторий не является обязательным.

Среди классически VCS наиболее известны CVS, Subversion, а среди распределенных – Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

## Системы контроля версий Git

Системы контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями.

Благодаря тому, что Git является распределенной системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.
```

Рис. 3: Добавление теоретического введения

Добавление выполнения лабораторной работы

```
# Выполнение лабораторной работы
1. Установка git
К скачиванию, скриншот был утерян
2. Установка gh
! [Установка gh] (image/Screenshot 2023-02-18 at 8.26.48 PM.png) (#fig:001 width=70%)
3. Базовая настройка git
! [Базовая настройка git] (image/Screenshot 2023-02-19 at 11.25.06 AM.png) (#fig:002 width=70%)
4. Создание ключа ssh
! [Создание ключа ssh] (image/Screenshot 2023-02-16 at 2.07.35 PM.png) (#fig:003 width=70%)
5. Настройка gh
! [Настройка gh] (image/Screenshot 2023-02-18 at 8.32.39 PM.png) (#fig:004 width=70%)
6. Создание репозитория курса на основе шаблона
! [Создание репозитория курса на основе шаблона] (image/Screenshot 2023-02-18 at 10.27.14 PM.png) (#fig:005 width=70%)
7. Настройка каталога курса
К скачиванию, скриншот был утерян
```

Рис. 4: Добавление выполнения лабораторной работы

Добавление контрольных вопросов

```
##? Контрольные вопросы:

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?

Системы контроля версий (VCS) – это программное обеспечение, которое позволяет отслеживать изменения в коде, документах и других файлах, которые хранятся в репозитории. Они позволяют контролировать и управлять версиями файлов, отслеживать изменения, внесенные в файлы, и восстанавливать предыдущие версии.

VCS решает следующие задачи:


- Управление версиями: сохранение и отслеживание изменений в коде и документах
- Совместная работа над проектом: позволяет разным разработчикам работать над одним и тем же кодом и контролировать изменения
- Восстановление предыдущих версий: возможность возврата к предыдущим версиям файлов и кода
- Слияние изменений: позволяет объединять изменения, внесенные несколькими разработчиками в один исходный код



2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище (repository) – это место, где хранятся файлы, версии и история изменений в проекте.

Commit (зафиксировать изменения) – это операция, при которой внесенные изменения в файлы исходного кода или документы сохраняются в VCS.

История (history) – это список всех commit'ов, которые были внесены в хранилище.

Рабочая копия (working copy) – это локальная копия файлов исходного кода из хранилища, которая находится на компьютере разработчика.

Отношения между этими понятиями таковы: при работе с VCS разработчик обычно клонирует репозиторий (хранилище) на свой компьютер, создавая рабочую копию проекта. Затем он вносит изменения в файлы, находящиеся в рабочей копии, и фиксирует эти изменения в хранилище с помощью коммитов. Все коммиты хранятся в истории изменений, которую можно использовать для просмотра и восстановления предыдущих версий файлов.

3. Что представляет собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные VCS это тип VCS, в котором все версии файлов исходного кода хранятся в центральном репозитории. Все разработчики работают с этим центральным репозиторием, загружая и скачивая файлы для выполнения задач. Каждый разработчик работает с локальной копией файлов, которая содержит последнюю версию файлов из центрального репозитория. Централизованные VCS могут быть полезными для команд, которые работают над большими проектами и имеют централизованную структуру управления. Пример CVS.



- Subversion (SVN) – один из наиболее известных и широко используемых централизованных VCS.



Децентрализованные VCS это тип VCS, в котором каждый разработчик имеет свою локальную копию полного репозитория. Это позволяет разработчикам работать независимо друг от друга и синхронизироваться с другими разработчиками только при необходимости. Каждый разработчик может легко фиксировать изменения в своем локальном репозитории и сливать изменения с другими разработчиками, что делает работу в команде более эффективной. Пример DVCS:



- Git – наиболее популярная и широко используемая децентрализованная VCS.

```

Рис. 5: Добавление контрольных вопросов

Выполняя данную лабораторную работу, я научился оформлять отчёты с помощью легковесного языка разметки Markdown.