Отчёт по лабораторной работе №2

Операционные системы

Ведьмина Александра Сергеевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить идеологию и применение средств контроля версий. Освоить работу с гит.

2 Задание

1. Ознакомиться с теоретическим введением.
2. Создать репозиторий на гитхаб по имеющемуся шаблону.

3 Теоретическое введение

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

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 определённых 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 сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. 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.

4 Выполнение лабораторной работы

Я уже имею репозиторий на гитхаб, поэтому приступаю сразу к созданию каталога для курса.

Создаю репозиторий с названием os-intro, переходя по “use this template” в репозитории-шаблоне.

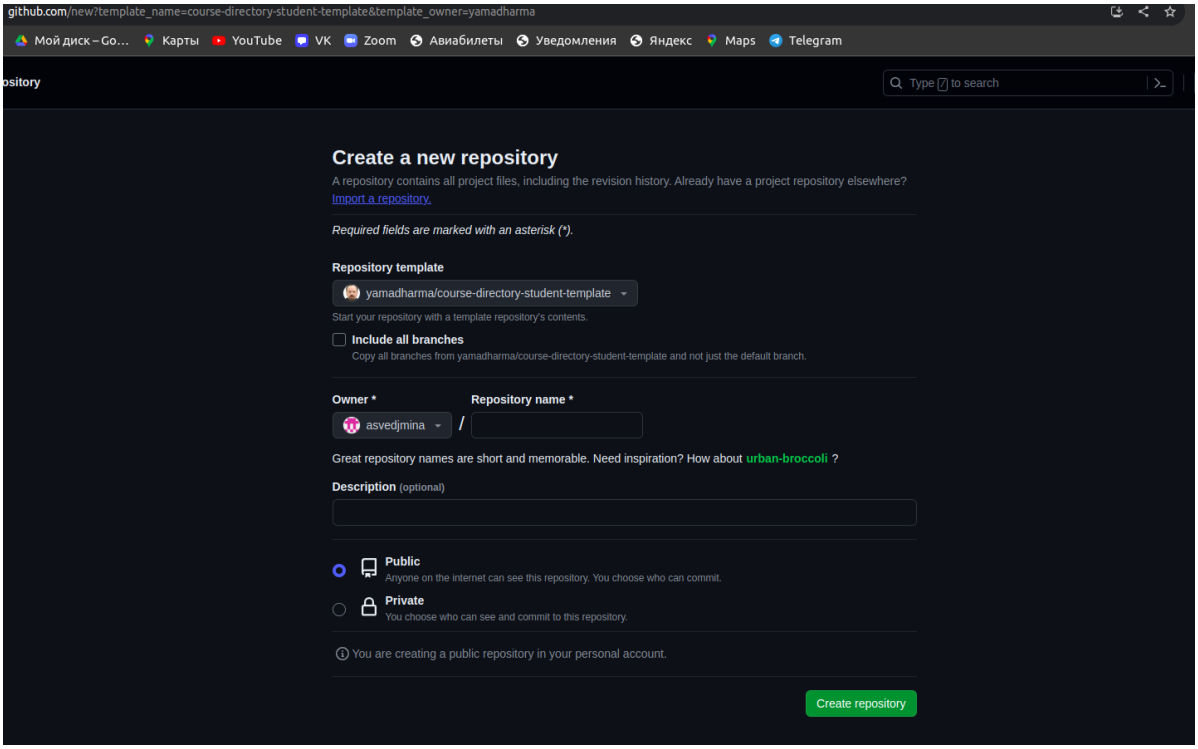

Рис. 4.1: Создание репозитория

Создаю папку “Операционные системы” в work/study/2023-2024. Клонирую репозиторий с гитхаба.

```
asvedjmina@asvedjmina-Modern-15-B12M:~/work/study/2023-2024/Операционные системы$ git c
lone --recursive git@github.com:asvedjmina/os-intro.git
Cloning into 'os-intro'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Receiving objects: 100% (32/32), 18.60 KiB | 340.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-
markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-t
emplate.git) registered for path 'template/report'
Cloning into '/home/asvedjmina/work/study/2023-2024/Операционные системы/os-intro/templ
ate/presentation'...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Cloning into '/home/asvedjmina/work/study/2023-2024/Операционные системы/os-intro/templ
ate/report'...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Receiving objects: 100% (126/126), 335.80 KiB | 1.62 MiB/s, done.
Resolving deltas: 100% (52/52), done.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca72c60a3
```

Рис. 4.2: Клонирование репозитория

Удаляю все файлы типа json.

```
asvedjmina@asvedjmina-Modern-15-B12M:~/work/study/2023-2024/Операционные системы$ cd '/
home/asvedjmina/work/study/2023-2024/Операционные системы/os-intro'
asvedjmina@asvedjmina-Modern-15-B12M:~/work/study/2023-2024/Операционные системы/os-int
ro$ rm package.json
```

Рис. 4.3: Удаление лишних файлов

Создаю требуемую структуру каталога с помощью команды make.

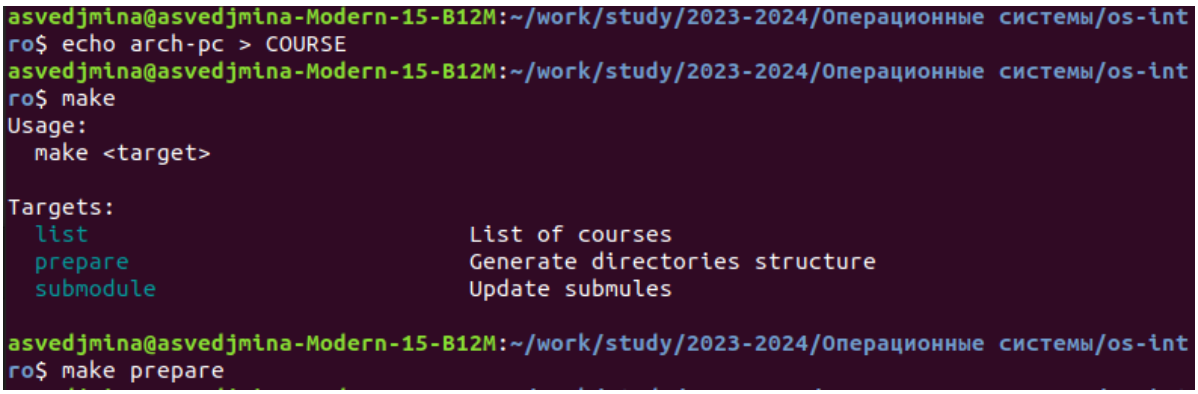

Рис. 4.4: Создание структуры каталога

Загружаю все изменения на гитхаб.

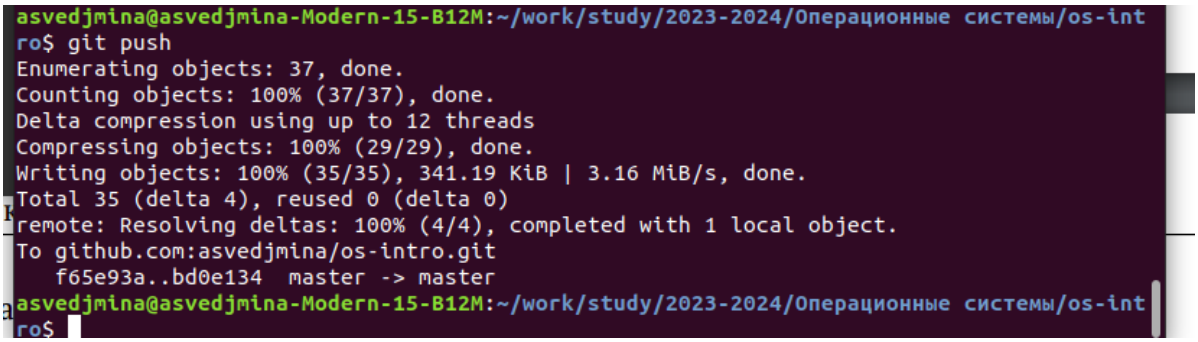

Рис. 4.5: Загрузка изменений на гитхаб

5 Выводы

В ходе лабораторной работы я освоила навыки работы с git и создала каталог для выполнения заданий на курсе.

6 Ответы на контрольные вопросы

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

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

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

Хранилище - единый репозиторий для хранения файлов. Commit позволяет сохранять все добавленные изменения и все изменённые файлы. История - история изменений, внесённых в проект. Рабочая копия - копия с которой непосредственно работают.

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

При централизованной VCS репозиторий хранится на одном сервере, все разработчики работают с ним (например, Apache Subversion). В децентрализованных VCS каждый пользователь имеет локальную копию всей истории работы (например, Monotone).

4. Опишите действия с VCS при единоличной работе с хранилищем.

Создание локальной копии репозитория, внесение изменений в неё, коммит изменений в репозиторий.

5. Опишите порядок работы с общим хранилищем VCS.

Клонирование репозитория, внесение изменений в проект, коммит изменений в репозиторий, отправка их в общий репозиторий.

6. Каковы основные задачи, решаемые инструментальным средством git?

Фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

7. Назовите и дайте краткую характеристику командам git.

git init - создание основного дерева репозитория git pull - получение обновлений (изменений) текущего дерева из центрального репозитория git push - отправка всех произведённых изменений локального дерева в центральный репозиторий git status - просмотр списка изменённых файлов в текущей директории git diff - просмотр текущих изменения git add . - добавить все изменённые и/или созданные файлы и/или каталоги git add имена_файлов - добавить конкретные изменённые и/или созданные файлы и или каталоги git rm имена_файлов - удалить файл и/или каталог из индекса репозитория и другие

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

git add . git commit -am ‘feat(main): make course structure’

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветви - версии одного файла, имеющие общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви.

10. Как и зачем можно игнорировать некоторые файлы при commit?

Чтобы игнорировать некоторые файлы при коммите в VCS, можно использовать файл .gitignore (для Git).