

# **Лабораторная работа № 2**

**Первоначальная настройка git**

Кадирова Мехрубон

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Создание репозитория курса на основе шаблона.	9
5	Настройка каталога курса	11
6	Выводы	12
	Список литературы	13

## Список иллюстраций

4.1	Создание шаблона рабочего пространства . . . . .	9
4.2	Создание репозитория . . . . .	10
5.1	создаем необходимые каталоги . . . . .	11
5.2	. . . . .	11

## Список таблиц

# 1 Цель работы

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

## 2 Теоретическое введение

- Системы контроля версий. Общие понятия 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.

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 поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными

участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая 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, 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.

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

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. |

Более подробно об Unix см. в [1–6].

### **3 Выполнение лабораторной работы**



## 4 Сознание репозитория курса на основе шаблона.

- Необходимо создать шаблон рабочего пространства (см. Рабочее пространство для лабораторной работы) (рис. 4.2).

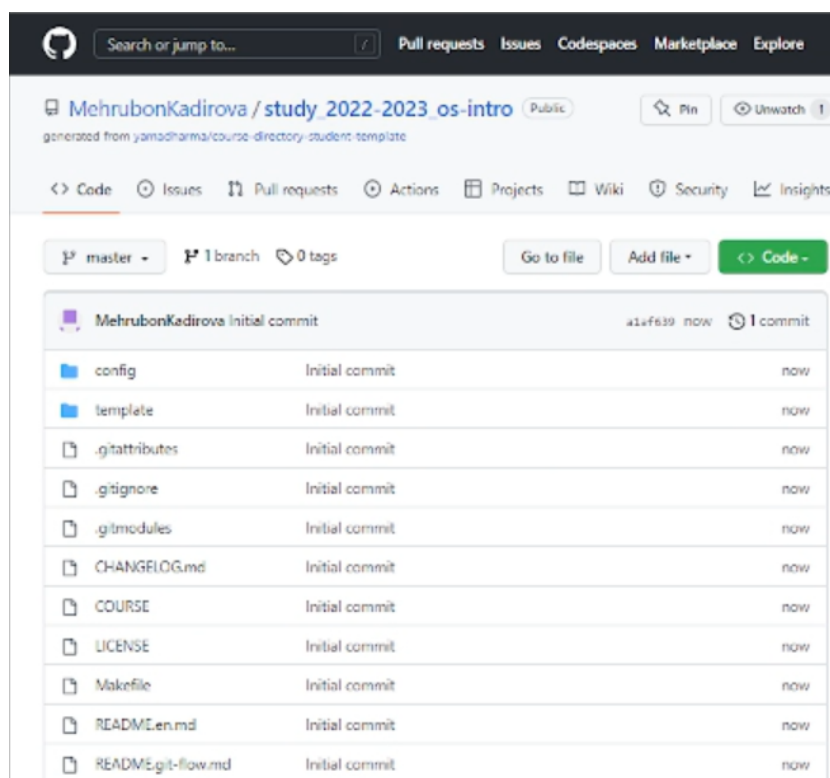


Рис. 4.1: Создание шаблона рабочего пространства

- Например, для 2022–2023 учебного года и предмета «Операционные системы» (код предмета os-intro) создание репозитория примет следующий вид

(рис. ??):

```
[mehrubonkr@mrkadirova ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[mehrubonkr@mrkadirova ~]$ cd ~/work/study/2022-2023/"Операционные системы"
[mehrubonkr@mrkadirova Операционные системы]$ git clone --recursive git@github.com:MehrubonKadirova/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.94 КиБ | 1.54 МБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/mehrubonkr/work/study/2022-2023/Операционные системы/os-intro/template/presentation»...
```

Рис. 4.2: Создание репозитория

## 5 Настройка каталога курса

- Перейдите в каталог курса:
- Удалите лишние файлы:
- Создайте необходимые каталоги (рис. 5.1):

```
[mehrubonkr@mrkadirova Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/os-intro
[mehrubonkr@mrkadirova os-intro]$ rm package.json
[mehrubonkr@mrkadirova os-intro]$ echo os-intro > COURSE
[mehrubonkr@mrkadirova os-intro]$ make
```

Рис. 5.1: создаем необходимые каталоги

- Отправьте файлы на сервер (рис. 5.2):

```
py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
[mehrubonkr@mrkadirova os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (38/38), готово.
Запись объектов: 100% (38/38), 342.40 Киб | 2.00 Миб/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:MehrubonKadirova/study_2022-2023_os-intro.git
alaf639..1f09d03 master -> master
[mehrubonkr@mrkadirova os-intro]$
```

Рис. 5.2:

## 6 Выводы

Изучили идеологию и применение средств контроля версий. Освоили умения по работе с git.

## Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.  
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.