

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютера

Студент: Бызова М.О.

Группа: НПИбд-01-23

МОСКВА

2023 г.

Содержание.

Цель работы.....	3
Теоретическое введение.....	4
Выполнение лабораторной работы.....	5
1. Настройка Github	5
2. Базовая настройка Git.....	6
3. Создание SSH ключа.....	7
4. Создание рабочего пространства и репозитория курса на основе шаблона.....	8
5. Создание репозитория курса на основе шаблона.....	9
6. Настройка каталога курса.....	11
Описание результатов выполнения заданий для самостоятельной работы.....	13
Выводы.....	16

Цель работы.

Целью данной лабораторной работы является изучение идеологии и применение средств контроля версий, приобретение практических навыков по работе с системой git.

Теоретическое введение.

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 изменения, сделанные разными участниками, 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. 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. Система 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 распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией. Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений). Затем можно вносить изменения в локальное дерево и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

Выполнение лабораторной работы.

1. Настройка Github.

Чтобы приступить к настройке, зайдя через браузер на сайт [GitHub](https://github.com), создаем учетную запись или входим в ранее уже созданную. При необходимости заполняем основные данные. Для выполнения лабораторной работы я воспользуюсь ранее созданной учетной записью. (рис. 1-3)

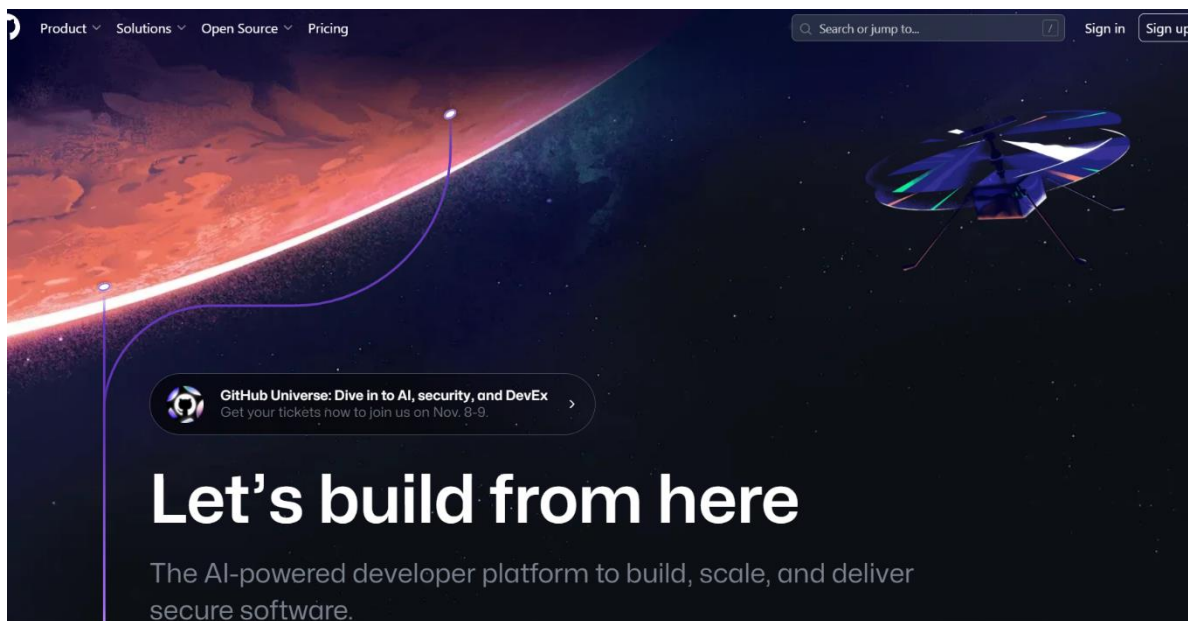


Рисунок 1. Главная страница сайта Github.

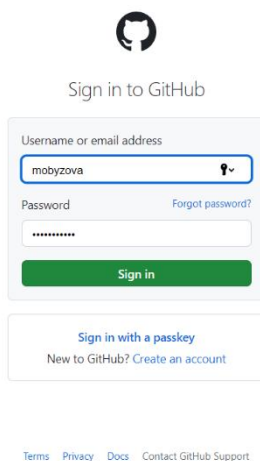


Рисунок 2. Вход в ранее созданный аккаунт.

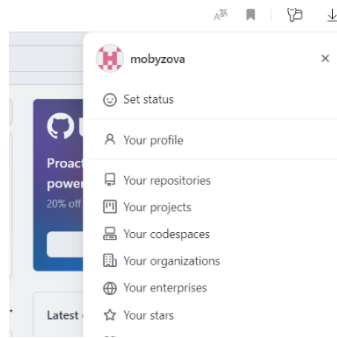


Рисунок 3. Аккаунт Github.

2. Базовая настройка Git.

Для дальнейшей настройки Git открываем терминал и вводим следующие команды: команду `git config --global user.name ""`, указывая свое имя, и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца репозитория (рис. 4).

```
mobihzova@dk8n73 ~ $ git config --global user.name "<mobyzova>"
mobihzova@dk8n73 ~ $ git config --global user.email "<1132236129@pfur.ru>"
mobihzova@dk8n73 ~ $
```

Рисунок 4. Предварительная конфигурация git.

Настраиваем utf-8 в выводе сообщений git для корректного отображения символов (рис. 5).

```
mobihzova@dk8n73 ~ $ git config --global core.quotePath false
```

Рисунок 5. Настройка кодировки

Далее зададим имя начальной ветки. Назовём её «master» (рис. 6).

```
mobihzova@dk8n73 ~ $ git config --global init.defaultBranch master
```

Рисунок 6. Создание имени для начальной ветки.

Зададим параметр `autocrlf` со значением `input` для того, чтобы конвертировать CRLF в LF только при коммитах. Уточним, что CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах (рис. 7)

```
mobihzova@dk8n73 ~ $ git config --global core.autocrlf input
```

Рисунок 7. Параметр autocrlf.

Далее зададим параметр `safecrlf` со значением `warn`. Таким образом, Git будет проверять преобразование на обратимость. В дальнейшей работе при значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации (рис. 8).

```
mobihzova@dk8n73 ~ $ git config --global core.safecrlf warn
```

Рисунок 8. Параметр safecrlf.

3. Создание SSH ключа.

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого вводим команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая имя владельца и электронную почту владельца. Ключ автоматически сохранится в каталоге `~/.ssh/` (рис. 9).

```
mobihzova@dk8n73 ~ $ ssh-keygen -C "mobyzova <1132236129@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh/id_rsa):
Created directory '/afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1j09zAQ1SI8j9goPoo2Mh7sfc1bN0y8dlkZtSRE+R60 mobyzova <1132236129@pfur.ru>
The key's randomart image is:
+---[RSA 3072]---+
|      .oo=o.. |
|      .=.o  .|
|      o oo=...|
|      + =.*+oE |
|      . + S.+o= |
| + + o = o=. . |
|o B +  =+ .   |
| o =   .o.    |
|oo.      ..   |
+----[SHA256]-----+
mobihzova@dk8n73 ~ $
```

Рисунок 9. Создание SSH-ключа.

Далее копируем открытый ключ из директории, в которой он был сохранен, с помощью утилиты `xclip` (рис. 10).

```
mobihzova@dk8n73 ~ $ cat ~/.ssh/id_rsa.pub | xclip -sel clip
mobihzova@dk8n73 ~ $
```

Рисунок 10. Копирование содержимого файла.

Далее необходимо загрузить сгенерённый открытый ключ. Для этого заходим на сайт <http://github.org/> под своей учётной записью и переходим в меню Setting. После этого выбираем в боковом меню SSH and GPG keys и нажимаем кнопку New SSH key. Вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title). Нажимаем «Add SSH-key», чтобы завершить добавление ключа (рис. 11).

Add new SSH Key

Title

Key type

Authentication Key ▾

Key

```
htW9b83KDFm2rlwc48fu4lRrSE1umB5ZjHU3A2JQ7CWHhsiUSlaGOKP7KSD6r/9mJWerZxEao3
/xdtBPvuiXFv9XepfPFKDdwLYRbcuwjrCHo
/5Qfii0WbC++NZvVgqvb1GtC9FD9xssMPOwZaFg8aei2hM8rOPmsJqwrCNdMej81gjYfNOIM6tIUeK3dpRprSezvbaVCepR
7ut0qRikN7tP7/BneWwpzF3hQMyYshBqFGF92wDw8iJTkgydT
/2ATICXxj4n+YrAhrLfkWbMVTAvkSghH2Kllmz2Gv2+gybfaoemrc3WXnVEmTlpHvpCpzmqsXjR
/e/PrU1zgWAlhTwG7Z1jF3WOMxrJvBx7pufGTx0osQNaJWmgXDr
/OCUCoAw9PiHhvfpx8JD2jTTFhpZPL7AwzHqc2hUGUGnBXG+tFfJPuWVqs7KI9MXsevygc= mobyzova
<1132236129@pfur.ru>
```

Add SSH key

Рисунок 11. Добавление ключа.

4. Создание рабочего пространства и репозитория курса на основе шаблона.

После проделанных действий возвращаемся в терминал. Создаём каталог для предмета «Архитектура компьютера» с помощью утилиты `mkdir` с ключом `-p`. Далее проверяем работу команды (рис. 12-13) любым удобным способом.

```
mobihzova@dk8n73 ~ $ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
mobihzova@dk8n73 ~ $
```

Рисунок 12. Создание рабочего пространства.



Папка пуста

Рисунок 13. Проверка работы команды.

5. Создание репозитория курса на основе шаблона.

Далее в браузере переходим на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharm/course-directory-student-template>. Теперь выбираем «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. 14).

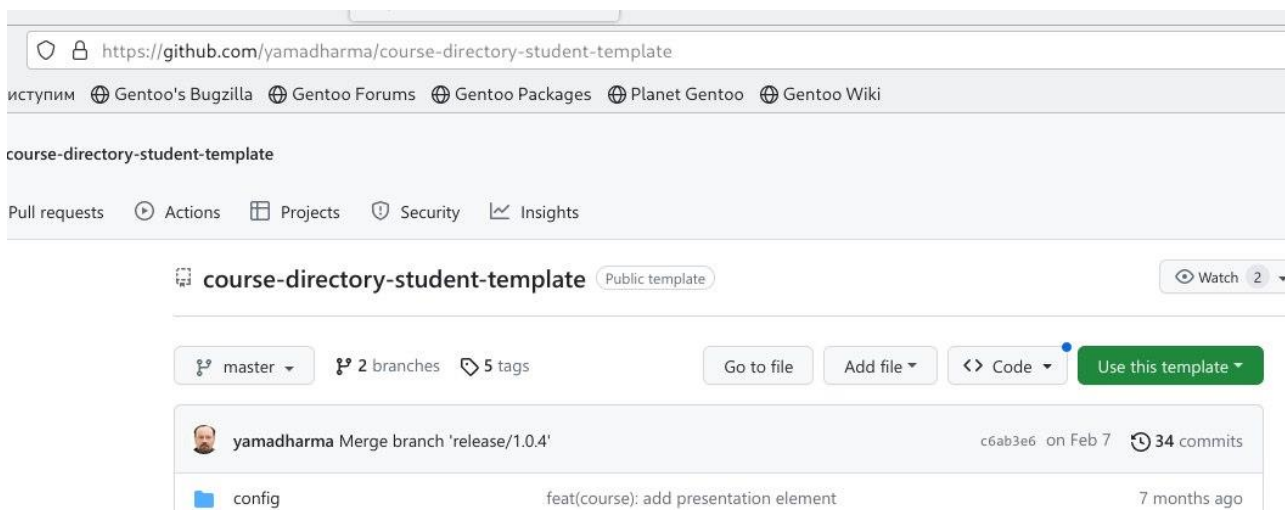


Рисунок 14. Страница шаблона для репозитория.

В открывшемся окне задаем имя репозитория (Repository name) «study_2023-2024_arh-pc» и создаем репозиторий (кнопка Create repository from template) (рис. 15).

https://github.com/new?template_name=course-directory-student-template&template_owner=yamadharma

иступим Gentoo's Bugzilla Gentoo Forums Gentoo Packages Planet Gentoo Gentoo Wiki

Q Type [7] to search

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *

study_2023-2024_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about [upgraded-bassoon](#)?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

📘 You are creating a public repository in your personal account.

[Creating repository...](#)

Рисунок 15. Окно создания репозитория.

Таким образом, репозиторий успешно создан (рис. 16).

https://github.com/mobyzova/study_2023-2024_arh-pc

иступим Gentoo's Bugzilla Gentoo Forums Gentoo Packages Planet Gentoo Gentoo Wiki

dy_2023-2024_arh-pc

Q Type [7] to search

Pull requests Actions Projects Wiki Security Insights Settings

study_2023-2024_arh-pc Public

generated from [yamadharma/course-directory-student-template](#)

[Pin](#) [Unwatch](#) [Fork](#) [Star](#)

[master](#) [1 branch](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

File	Commit	Time
mobyzova Initial commit	1048bea	now
config	Initial commit	now
template	Initial commit	now
.gitattributes	Initial commit	now
.gitignore	Initial commit	now
.gitmodules	Initial commit	now
CHANGELOG.md	Initial commit	now
COURSE	Initial commit	now
LICENSE	Initial commit	now
Makefile	Initial commit	now

About

No description, website, or topics provided

[Readme](#)

[CC-BY-4.0 license](#)

[Activity](#)

[0 stars](#)

[1 watching](#)

[0 forks](#)

Releases

No releases published

[Create a new release](#)

Packages

Рисунок 16. Созданный репозиторий.

Для дальнейшей работы через терминал переходим в созданный каталог курса с помощью уже известной нам утилиты `cd` (рис. 17).

```

mobihzova@dk8n73 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера $

```

Рисунок 17. Перемещение между директориями.

Теперь клонируем созданный репозиторий с помощью команды `git clone --recursive git@github.com:/study_2023-2024_arh-pc.git arch-pc`, предварительно скопировав ссылку для клонирования на странице созданного репозитория: сначала переходим в окно «code», далее выбираем в окне вкладку «SSH» (рис. 18-19).

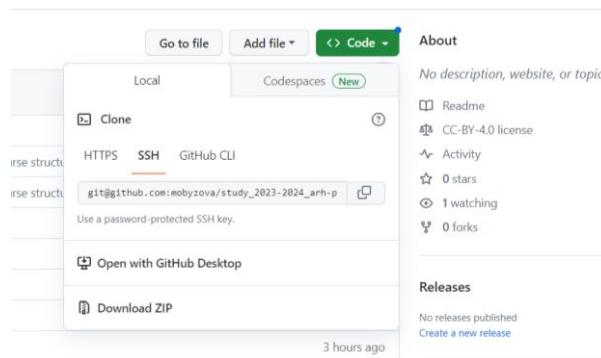


Рисунок 18. Окно с ссылкой для копирования репозитория.

```

mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:mobyzova/study_2023-2024_arh-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:DiY3wvV6TujhbpZisF/zLDA0zPMSVhdKr4UvCoQu.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
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), 16.93 КиБ | 8.46 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 «/afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 1.21 МБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 КиБ | 2.32 МБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера $

```

Рисунок 19. Клонирование репозитория.

Таким образом, репозиторий успешно скопирован.

6. Настройка каталога курса.

Для дальнейшей работы переходим в каталог курса с помощью утилиты `cd` (рис. 20).

```

mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc

```

Рисунок 20. Перемещение между директориями.

Теперь удалим лишние файлы с помощью утилиты `rm` (рис. 21).

```

mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $

```

Рисунок 21. Удаление файлов.

Создадим все необходимые каталоги при помощи ряда команд (рис. 22).

```

mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ echo arch-pc > COURSE
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $

```

Рисунок 22. Создание каталогов.

Отправим созданные нами каталоги на наш сервер с локального репозитория: для этого добавляем все созданные ранее каталоги при помощи `git add`, комментируем и сохраняем изменения на сервере как добавление курса с помощью `git commit` (рис. 23).

```

mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add .
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am "feat(main): make course structure"
[master 2942e9a] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/_init_.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile

```

Рисунок 23. Добавление и сохранение изменений на сервере.

При помощи `push` отправляем всё на сервер (рис. 24).

```

mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.61 МиБ/с, готово.
Всего 35 (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:mobyzova/study_2023-2024_arh-pc.git
  1048bea..2942e9a master -> master
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $

```

Рисунок 24. Выгрузка изменений на сервер.

В конце концов проверим правильность создания иерархии рабочего пространства в локальном репозитории и на странице `github` (рис. 25-26).

mobyzova feat(main): make course structure		2942e9a 3 hours ago 2 commits
config	Initial commit	3 hours ago
labs	feat(main): make course structure	3 hours ago
presentation	feat(main): make course structure	3 hours ago
template	Initial commit	3 hours ago
.gitattributes	Initial commit	3 hours ago
.gitignore	Initial commit	3 hours ago
.gitmodules	Initial commit	3 hours ago
CHANGELOG.md	Initial commit	3 hours ago
COURSE	feat(main): make course structure	3 hours ago
LICENSE	Initial commit	3 hours ago
Makefile	Initial commit	3 hours ago
README.en.md	Initial commit	3 hours ago
README.git-flow.md	Initial commit	3 hours ago
README.md	Initial commit	3 hours ago
prepare	feat(main): make course structure	3 hours ago

Рисунок 25. Страница репозитория (1).

mobyzova feat(main): make course structure		2942e9a 3 hours ago History
Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	3 hours ago
lab02	feat(main): make course structure	3 hours ago
lab03	feat(main): make course structure	3 hours ago
lab04	feat(main): make course structure	3 hours ago
lab05	feat(main): make course structure	3 hours ago
lab06	feat(main): make course structure	3 hours ago
lab07	feat(main): make course structure	3 hours ago
lab08	feat(main): make course structure	3 hours ago
lab09	feat(main): make course structure	3 hours ago
lab10	feat(main): make course structure	3 hours ago
lab11	feat(main): make course structure	3 hours ago
README.md	feat(main): make course structure	3 hours ago
README.ru.md	feat(main): make course structure	3 hours ago

Рисунок 26. Страница репозитория (2).

Описание результатов выполнения заданий для самостоятельной работы.

1. Для создания отчета по выполнению лабораторной работы в соответствующем каталоге рабочего пространства переходим в директорию labs/lab02/report с помощью утилиты cd. Создаем в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch (рис. 27-28).

```
mobihszova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab02/report
mobihszova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ touch
```

Рисунок 27. Перемещение между директориями.


```

mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ touch Л02_Бызова_Отчёт
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $

```

Рисунок 28. Создание файла.

2. Для копирования отчетов по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства проверим местонахождение файла с отчетом по первой лабораторной работе - он должен находиться в подкаталоге домашней директории «Загрузки», для проверки используем команду `ls` (рис. 29). Затем копируем необходимый файл с помощью утилиты `cp` (рис. 30). Проверяем правильность выполнения команды `cp` с помощью `ls` (рис. 31).

```

mobihzova@dk8n73 ~ $ ls ~/Загрузки
'Telegram Web_files' 'Telegram Web.html' Отчёт_Лабораторная_работа_№1_Бызова_М_О_.pdf
mobihzova@dk8n73 ~ $

```

Рисунок 29. Проверка местонахождения файла.

```

mobihzova@dk8n73 ~ $ cp ~/Загрузки/Отчёт_Лабораторная_работа_№1_Бызова_М_О_.pdf ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
mobihzova@dk8n73 ~ $

```

Рисунок 30. Копирование файл.

```

mobihzova@dk8n73 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report $ ls
bib image Makefile pandoc report.md Отчёт_Лабораторная_работа_№1_Бызова_М_О_.pdf
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report $

```

Рисунок 31. Проверка выполнения команды.

3. Далее с помощью ранее изученных команд загружаем файлы на GitHub: добавляем с помощью `git add` нужный файл, сохраняем изменения с помощью `git commit`, отправляем в центральный репозиторий сохраненные изменения командой `git push`. Проверяем правильность работы (рис. 32-35).

```

mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report $ git add .
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report $ git commit -am "add lab01"
[master faebf2c] add lab01
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "labs/lab01/report/320\236\321\202\321\207\321\221\321\202_\320\233\320\260\320\261\320\276\321\200\320\260\321\202\320\276\321\200\320\260\321\202\320\276\321\200\320\275\320\260\321\217_\321\200\320\260\320\261\320\276\321\202\320\260_\342\204\226\320\221\321\213\320\267\320\276\320\262\320\260_\320\234_\320\236_.pdf"
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report $ git push
Перечисление объектов: 100% (10/10), готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 1.33 МБ | 2.10 МБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:mobihzova/study_2023-2024_arh-pc.git
 2942e9a..faebf2c master -> master
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report $

```

Рисунок 32. Загрузка файла с отчетом по лабораторной работе №1 на GitHub.

<div> <div>master</div> <div>study_2023-2024_arh-pc/labs/lab01/report</div> </div> <div> <div>mobihzova add lab01</div> <div>faebf2c · 8 minutes ago</div> <div>History</div> </div>		
Name	Last commit message	Last commit date
...		
bib	feat(main): make course structure	last week
image	feat(main): make course structure	last week
pandoc	feat(main): make course structure	last week
Makefile	feat(main): make course structure	last week
report.md	feat(main): make course structure	last week
Отчёт_Лабораторная_работа_№1_Бызова_М_О_.pdf	add lab01	8 minutes ago

Рисунок 33. Проверка правильности работы команд.

```

mobihzova@dk6n52 ~ $ cd work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab02/report
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ git add .
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ git commit -am "add lab02 first"
[master ec4a693] add lab02 first
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "labs/lab02/report/\320\23302_\320\221\321\213\320\267\320\276\320\262\320\260_\320\236\321\202\321\207\321\221\321\202"
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 486 байтов | 486.00 КиБ/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:mobyzova/study_2023-2024_arh-pc.git
faebf2c..ec4a693 master -> master
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $

```

Рисунок 34. Загрузка файла с отчетом по лабораторной работе №2 на GitHub.

study_2023-2024_arh-pc / labs / lab02 / report			Q Go to file	Add file	...
mobyzova add lab02 first			ec4a693 · 2 minutes ago History		
Name	Last commit message	Last commit date			
..					
bib	feat(main): make course structure	last week			
image	feat(main): make course structure	last week			
pandoc	feat(main): make course structure	last week			
Makefile	feat(main): make course structure	last week			
report.md	feat(main): make course structure	last week			
/02_Euzova_Ovch	add lab02 first	2 minutes ago			

Рисунок 35. Проверка правильности работы команд.

Мы видим, что пояснение к совершенным действиям отображается, а отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report, по второй – в lab02/report. Убеждаемся, что всё выполнено верно.

Выводы:

В ходе выполнения данной лабораторной работы были получены практические навыки работы с системой git, а также была изучена идеология и применение средств контроля версий посредством выполнения различных заданий, а именно:

1. Настройки GitHub.
2. Базовой настройки Git.
3. Создания SSH-ключа.
4. Создания рабочего пространства и репозитория курса на основе шаблона.
5. Создания репозитория курса на основе шаблона.
6. Настройки каталога курса.
7. Выполнения заданий для самостоятельной работы для подведения итогов и закрепления полученных знаний.