

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: Ким Илья Еногиевич

Группа: НКАбд-03-25

МОСКВА

2025 г.

Оглавление

1. <u>Цель работы</u>	3
2. <u>Задание</u>	4
3. <u>Теоретическое введение</u>	5
4. <u>Выполнение лабораторной работы</u>	7
4.1 <u>Техническое обеспечение</u>	7
4.2 <u>Базовая настройка Git</u>	7
4.3 <u>Создание рабочего пространства и репозитория курса на основе шаблона</u>	9
4.4 <u>Создание репозитория курса на основе шаблона</u>	10
4.5 <u>Настройка каталога курса</u>	11
5. <u>Задания для самостоятельной работы</u>	12
6. <u>Выводы</u>	14
7. <u>Список литературы</u>	15

1. Цель работы

Приобретение практических навыков по работе с системой контроля версий git.

2. Задание

На основе указания выполнить работу с базовыми командами системы контроля версий git, выучить применение команд для разных случаев использования, настроить GitHub

3. 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 и 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 во все или за заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файлам средствами файловой системы ОС, обеспечивая, таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Команда	Описание
<code>git init</code>	создание основного дерева репозитория
<code>git pull</code>	получение обновлений (изменений) текущего дерева из центрального репозитория
<code>git push</code>	отправка всех произведённых изменений локального дерева в центральный репозиторий
<code>git status</code>	просмотр списка изменённых файлов в текущей директории
<code>git diff</code>	просмотр текущих изменений
<code>git add .</code>	добавить все изменённые и/или созданные файлы и/или каталоги
<code>git add</code> имена_файлов	добавить конкретные изменённые и/или созданные файлы и/или каталоги
<code>git rm</code> имена_файлов	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
<code>git commit -am</code> 'Описание коммита'	сохранить все добавленные изменения и все изменённые файлы
<code>git checkout -b</code> имя_ветки	создание новой ветки, базирующейся на текущей
<code>git checkout</code> имя_ветки	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
<code>git push origin</code> имя_ветки	отправка изменений конкретной ветки в центральный репозиторий
<code>git merge</code> <code>--no-ff</code> имя_ветки	слияние ветки с текущим деревом
<code>git branch -d</code> имя_ветки	удаление локальной уже слитой с основным деревом ветки
<code>git branch -D</code> имя_ветки	принудительное удаление локальной ветки
<code>git push origin</code> :имя ветки	удаление ветки с центрального репозитория

Таблица 3.1 Описание основных команд `git`

4. Выполнение лабораторной работы

4.1. Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Linux Ubuntu

4.2. Базовая настройка Git

Для начала проведём предварительную конфигурацию Git, для этого открываем терминал и вводим команды (рис 4.2.1)

```
iekim@LINUXUBUNTU:~$ git config --global user.name "Ilya Kim"
iekim@LINUXUBUNTU:~$ git config --global user.email "2006.11.09kimilya@gmail.com"
iekim@LINUXUBUNTU:~$
```

Рис. 4.2.1 Предварительная конфигурация Git

Далее настраиваем параметры utf-8, имя начальной ветки, autocrlf и safecrlf (4.2.2)

```
iekim@LINUXUBUNTU:~$ ssh-keygen -C "2006.11.09kimilya@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/iekim/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/iekim/.ssh/id_ed25519
Your public key has been saved in /home/iekim/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:2uqvcvEMvqvIBs30kSaNVPPkr/eqJobQ2byMZr7s1as 2006.11.09kimilya@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|
|   o .
|  . =
|   o
|.* * ..S
|= X B ++.
|. *.O =o=..
| o=+*ooo+
| .+ +*EB+.
+-----[SHA256]-----+
iekim@LINUXUBUNTU:~$
```

Hbc

ilya

Key type

Authentication Key ↕

Key

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIH64/rTK6VdudH34c0yNNG7RABCjivZiT4XI/3zk1I4n 2006.11.09kimilya@gmail.com

Рис. 4.2.4. Добавление SSH ключа

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

Рабочее пространство при выполнении лабораторных работ должно придерживаться определённой структурной иерархии, для этого я создаю директорию на своём

```
iekim@LINUXUBUNTU:~$ mkdir -p work/study/2025-2026/"Архитектура компьютера"  
iekim@LINUXUBUNTU:~$
```

Рис. 4.3.1. Создание директории для работы

4.4. Создание репозитория курса на основе шаблона.

Создаём репозиторий на основе имеющего шаблона через функционал клонирования интерфейса в GitHub (рис. 4.4.1)

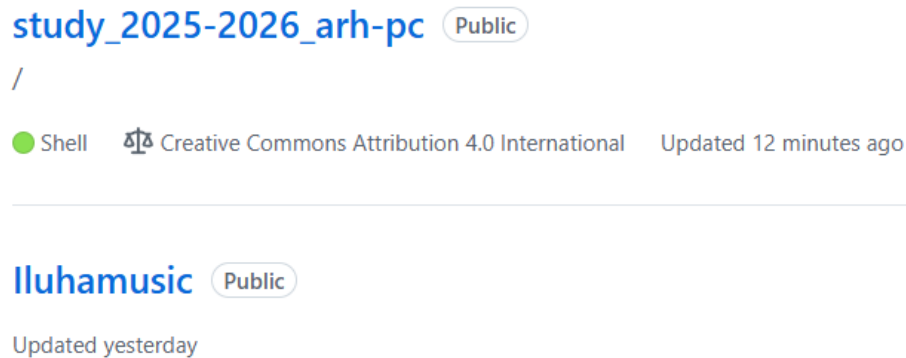


Рис. 4.4.1. Процесс клонирования репозитория через GitHub

Сгенерированный репозиторий на основе шаблона клонируем на свой рабочий компьютер, для этого берём ссылку для клонирования через интерфейс GitHub(рис. 4.4.2.) и вводим в терминале git clone (рис. 4.4.3.)

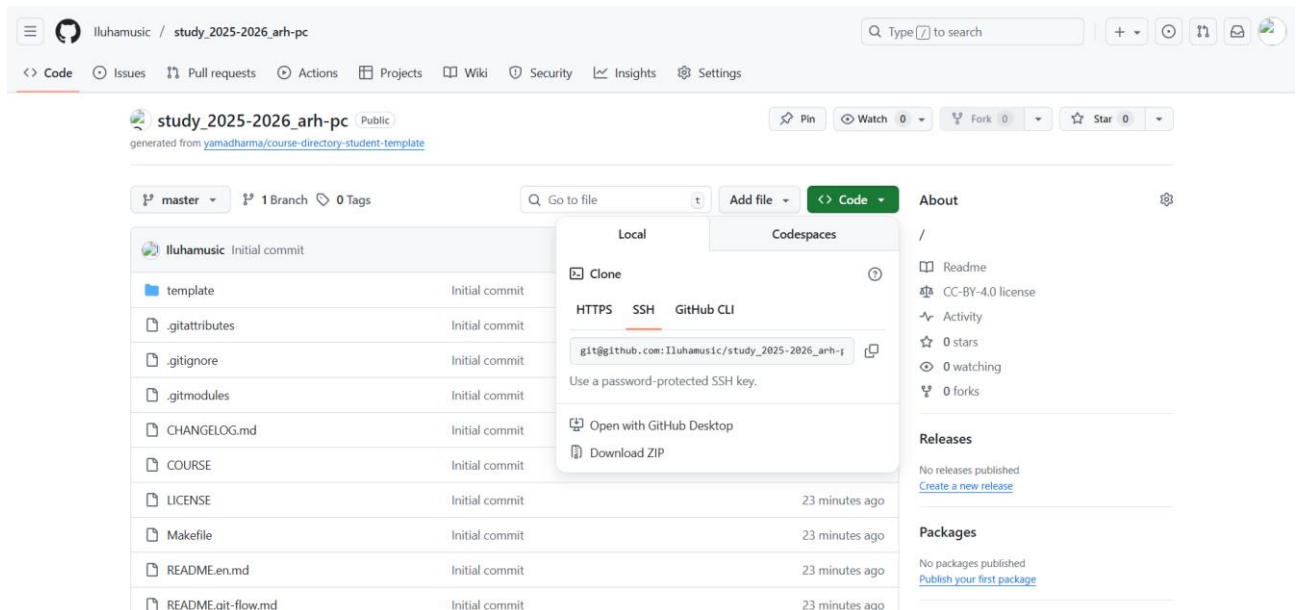


Рис. 4.4.2. Клонирование ссылки вставки в терминал.

```

iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера$ git clone --recursive git@github.com:iluhamusic/study_2025-2026_arh-pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 38 (delta 1), reused 27 (delta 1), pack-reused 0 (from 0)
Получение объектов: 100% (38/38), 23.45 КиБ | 615.00 КиБ/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/iekim/work/study/2025-2026/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 161, done.
remote: Counting objects: 100% (161/161), done.
remote: Compressing objects: 100% (111/111), done.
remote: Total 161 (delta 60), reused 142 (delta 41), pack-reused 0 (from 0)
Получение объектов: 100% (161/161), 2.65 МиБ | 3.80 МиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Клонирование в «/home/iekim/work/study/2025-2026/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 221, done.
remote: Counting objects: 100% (221/221), done.
remote: Compressing objects: 100% (152/152), done.
remote: Total 221 (delta 98), reused 180 (delta 57), pack-reused 0 (from 0)
Получение объектов: 100% (221/221), 765.46 КиБ | 2.82 МиБ/с, готово.
Определение изменений: 100% (98/98), готово.
Submodule path 'template/presentation': checked out '6efd5c4ee78e4456caff3dc7062cfcad26058ca6'
Submodule path 'template/report': checked out '89a9622199b4df88227b9b3fa3d4714c85f68dd2'

```

Рис. 4.4.3. Копирование репозитора на рабочий компьютер.

4.5. Настройка каталога курса.

В каталоге курса формируем необходимые каталоге (рис. 4.5.1.)

```

iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ make prapare

```

Рис. 4.5.1. Создание каталогов.

Далее отправляем файлы на сервер в GitHub (рис. 4.5.2.)

```
create mode 100644 labs/lab11/report/Makefile
create mode 100644 labs/lab11/report/_quarto.yml
create mode 100644 labs/lab11/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab11/report/arch-pc--lab11--report.qmd
create mode 100644 labs/lab11/report/bib/cite.bib
create mode 100644 labs/lab11/report/image/solvay.jpg
create mode 100644 prepare
create mode 100644 presentation/README.md
create mode 100644 presentation/README.ru.md
create mode 100644 presentation/presentation/.gitignore
create mode 100644 presentation/presentation/.marksman.toml
create mode 100644 presentation/presentation/.projectile
create mode 100644 presentation/presentation/Makefile
create mode 100644 presentation/presentation/_quarto.yml
create mode 100644 presentation/presentation/_resources/image/logo_rudn.png
create mode 100644 presentation/presentation/arch-pc--presentation--presentation.qmd
create mode 100644 presentation/presentation/image/kulyabov.jpg
create mode 100644 presentation/report/.gitignore
create mode 100644 presentation/report/.marksman.toml
create mode 100644 presentation/report/.projectile
create mode 100644 presentation/report/Makefile
create mode 100644 presentation/report/_quarto.yml
create mode 100644 presentation/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 presentation/report/arch-pc--presentation--report.qmd
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/solvay.jpg
iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
ssh: connect to host github.com port 22: Connection refused
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 69, готово.
Подсчет объектов: 100% (69/69), готово.
При сжатии изменений используется до 4 потоков
```

Рис. 4.5.2. Отправление файлов на github

5. Задания для самостоятельной работы.

Через терминал отправляю предыдущий отчёт по лабораторной работе на свой удалённый репозиторий в GitHub (рис. 5.1.), затем проверяем изменения на самом GitHub (рис.5.2.)

```
iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit -am "feat(mail): upload reports"
iekim@LINUXUBUNTU:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 69, готово.
Подсчет объектов: 100% (69/69), готово.
При сжатии изменений используется до 4 потоков
```

Рис. 5.1. Предыдущий отчёт по лабораторной работе

















 Iluhamusic first commit		674e051 · 49 minutes ago	 3 Commits
 labs	first commit	49 minutes ago	
 presentation	first commit	49 minutes ago	
 template	Initial commit	2 hours ago	
 .gitattributes	Initial commit	2 hours ago	
 .gitignore	Initial commit	2 hours ago	
 .gitmodules	Initial commit	2 hours ago	
 COURSE	feat(main): make course structure	1 hour ago	
 LICENSE	Initial commit	2 hours ago	
 Makefile	Initial commit	2 hours ago	
 README.en.md	Initial commit	2 hours ago	
 README.git-flow.md	Initial commit	2 hours ago	
 README.md	Initial commit	2 hours ago	
 package.json	first commit	49 minutes ago	
 prepare	first commit	49 minutes ago	

Рис. 5.2. Изменения на GitHub

6. Выводы

При выполнении данной работы мы изучили идеологию и применение средств контроля версий, а также приобрели практические навыки по работе с реализацией VSC git.

Список литературы

1. https://esystem.rudn.ru/pluginfile.php/2089082/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%962.%20%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9%20Git.pdf
2. <https://esystem.rudn.ru/mod/page/view.php?id=1030492>
3. <https://esystem.rudn.ru/mod/resource/view.php?id=1030495>
4. <https://esystem.rudn.ru/mod/resource/view.php?id=1030496>
5. https://github.com/evdvorkina/study_2022-2023_arh-pc/tree/master