

Лабораторная работа №2

дисциплина: Архитектура компьютера

Мазурский Александр Дмитриевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Техническое обеспечение	9
4.2	Базовая настройка Git	9
4.3	Создание рабочего пространства и репозитория курса на основе шаблона	11
4.4	Создание репозитория курса на основе шаблона	12
4.5	Настройка каталога курса	14
4.6	Задания для самостоятельной работы	15
5	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Предварительная конфигурация Git	9
4.2	Настройка параметров Git	10
4.3	Создание пары ssh ключей	10
4.4	Установка пары ключей	11
4.5	Создание рабочей директории	12
4.6	Шаблон для клонирования на платформе GitHub	12
4.7	Процесс клонирования репозитория через интерфейс GitHub . . .	13
4.8	Копирование ссылки для последующей вставки в терминал	14
4.9	Копирование репозитория на рабочий компьютер	14
4.10	Настройка каталога курса	15
4.11	Отправка изменений на удаленный репозиторий	15
4.12	Отправка изменений на удаленный репозиторий	16
4.13	Проверка выполненных изменений на GitHub.	16

Список таблиц

1 Цель работы

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

2 Задание

На основе методических указаний провести 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 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 зави-

симости 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.

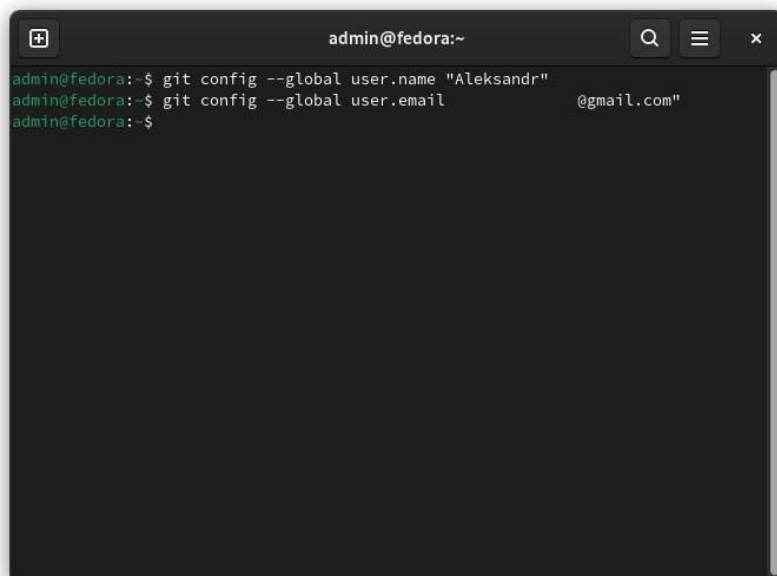
4 Выполнение лабораторной работы

4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Fedora Workstation 40.

4.2 Базовая настройка Git

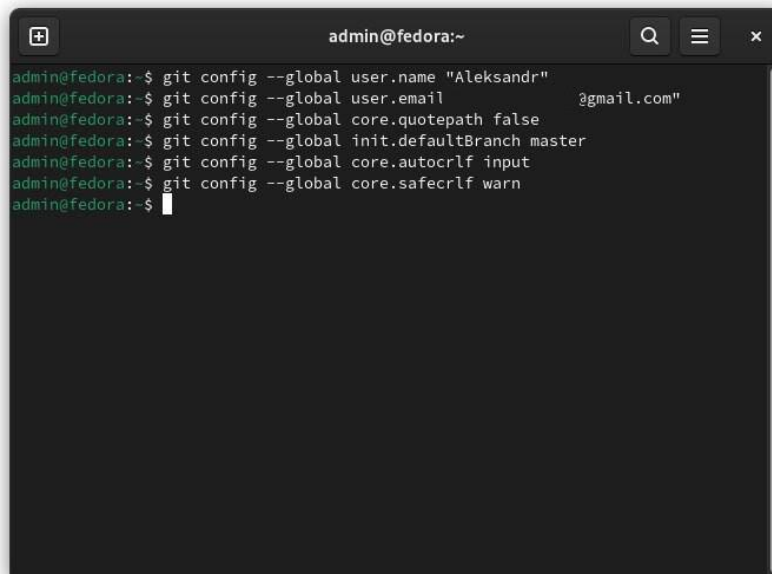
Для начала я проведу предварительную конфигурацию Git, для этого открываю терминал и ввожу команды на (рис. 4.1).

A screenshot of a terminal window titled 'admin@fedora:~'. The window has a dark background and a light-colored text. The terminal shows three lines of text: the first line is 'admin@fedora:~\$ git config --global user.name "Aleksandr"', the second line is 'admin@fedora:~\$ git config --global user.email @gmail.com"', and the third line is 'admin@fedora:~\$'. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

```
admin@fedora:~$ git config --global user.name "Aleksandr"
admin@fedora:~$ git config --global user.email @gmail.com"
admin@fedora:~$
```

Рис. 4.1: Предварительная конфигурация Git

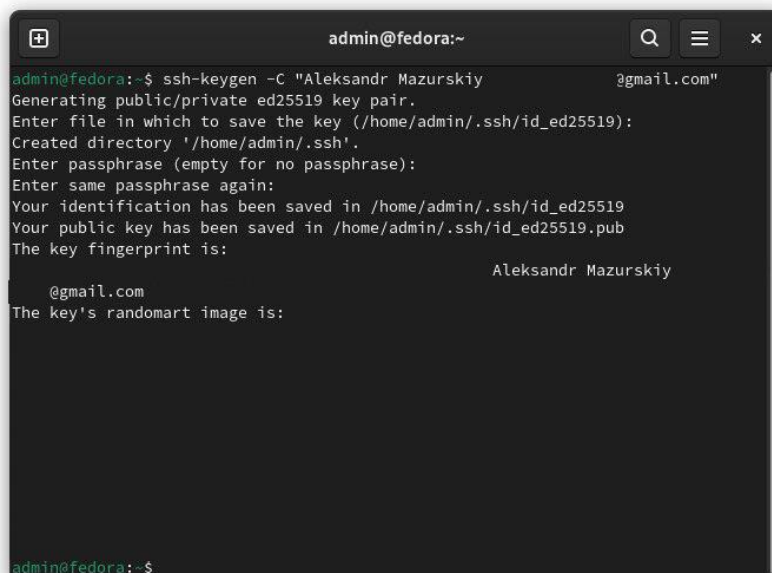
Далее настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf.
(рис. 4.2)

A terminal window titled 'admin@fedora:~' with search, menu, and close icons. It shows a series of 'git config' commands being entered at the 'admin@fedora:~\$' prompt. The commands set global user information and core settings for Git.

```
admin@fedora:~$ git config --global user.name "Aleksandr"
admin@fedora:~$ git config --global user.email @gmail.com"
admin@fedora:~$ git config --global core.quotepath false
admin@fedora:~$ git config --global init.defaultBranch master
admin@fedora:~$ git config --global core.autocrlf input
admin@fedora:~$ git config --global core.safecrlf warn
admin@fedora:~$
```

Рис. 4.2: Настройка параметров Git

Далее создаю пару ssh ключей для интеграции с платформой GitHub. (рис. 4.3)

A terminal window titled 'admin@fedora:~' with search, menu, and close icons. It shows the execution of the 'ssh-keygen' command with specific options. The output shows the generation of an ed25519 key pair, the creation of a directory, and the saving of the key and its fingerprint.

```
admin@fedora:~$ ssh-keygen -C "Aleksandr Mazurskiy" @gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/admin/.ssh/id_ed25519):
Created directory '/home/admin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/admin/.ssh/id_ed25519
Your public key has been saved in /home/admin/.ssh/id_ed25519.pub
The key fingerprint is:
                                Aleksandr Mazurskiy
                                @gmail.com
The key's randomart image is:

admin@fedora:~$
```

Рис. 4.3: Создание пары ssh ключей

Далее я перехожу на сайт GitHub, авторизуюсь, перехожу в настройки аккаунта, вставляю публичный ключ в предназначенном для этого поле. (рис. 4.4)

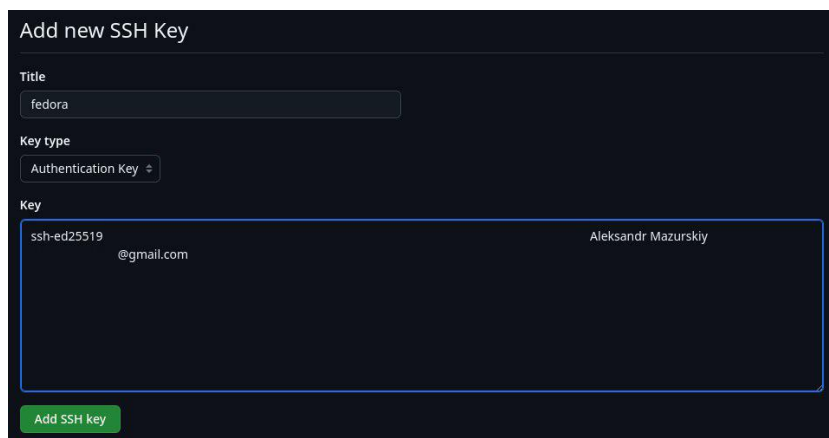


Рис. 4.4: Установка пары ключей

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 своем рабочем компьютере. (рис. 4.5)

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 ().*

Repository template

yamadharm/course-directory-student-template ▾

Start your repository with a template repository's contents.

☐ **Include all branches**
Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

Owner * **Repository name ***

nowherewashere / study_2024_2025_arch-pc

✔ study_2024_2025_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about **solid-octo-spork** ?

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.

Create repository

Рис. 4.7: Процесс клонирования репозитория через интерфейс GitHub

Сгенерированный репозиторий на основе шаблона клонирую на свой рабочий компьютер, для этого беру ссылку для клонирования через интерфейс GitHub (рис. 4.8) и затем ввожу в терминале `git clone`. (рис. 4.9)

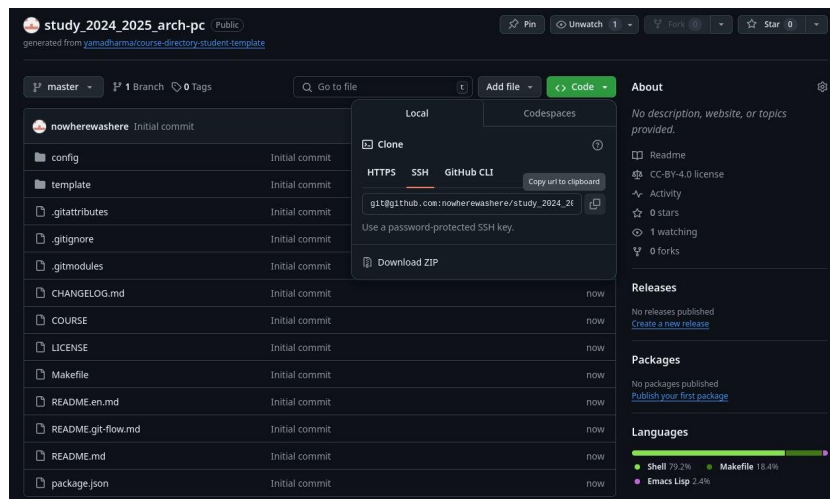


Рис. 4.8: Копирование ссылки для последующей вставки в терминал

```
admin@fedora: ~$ cd /work/study/2024-2025/Архитектура/компьютера/
admin@fedora: /work/study/2024-2025/Архитектура/компьютера$ git clone --recursive git@github.com:nowherewashere/study_2024_2025_arch-pc.git arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (33/33), 18.83 KiB | 18.83 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharmacourse-directory-student-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharmacourse-directory-student-template.git) registered for path 'template/report'
Cloning into '/home/admin/work/study/2024-2025/Архитектура/компьютера/arch-pc/template/presentation'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 1.20 MiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/admin/work/study/2024-2025/Архитектура/компьютера/arch-pc/template/report'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Receiving objects: 100% (142/142), 341.09 KiB | 2.66 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c992712b4b2d431ad5086c9c72a92bd2fcald4a6'
Submodule path 'template/report': checked out 'c28622eff7b3e0495707892ef561ab185f5c748'
admin@fedora: /work/study/2024-2025/Архитектура/компьютера$
```

Рис. 4.9: Копирование репозитория на рабочий компьютер

4.5 Настройка каталога курса

В каталоге курса удаляю лишние файлы и формирую необходимые каталоги.
(рис. 4.10)

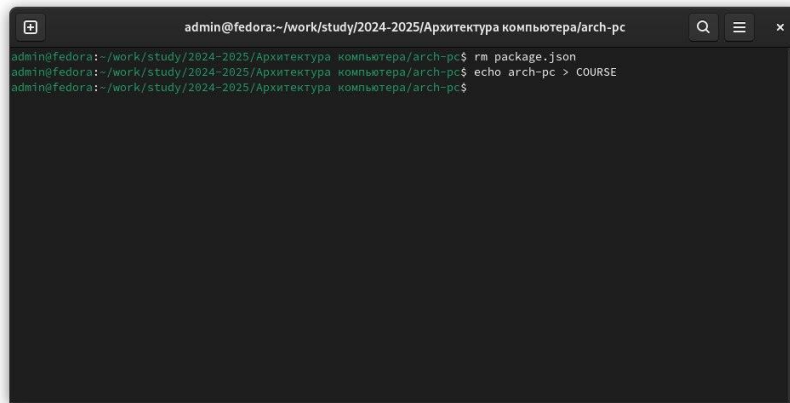
A terminal window with a dark background. The title bar shows 'admin@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc'. The terminal contains three lines of commands and their outputs: 'rm package.json', 'echo arch-pc > COURSE', and a prompt 'admin@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc\$'.

Рис. 4.10: Настройка каталога курса

Делаю снимок сделанных изменений и push'у их на свой репозиторий в GitHub.
(рис. 4.11)

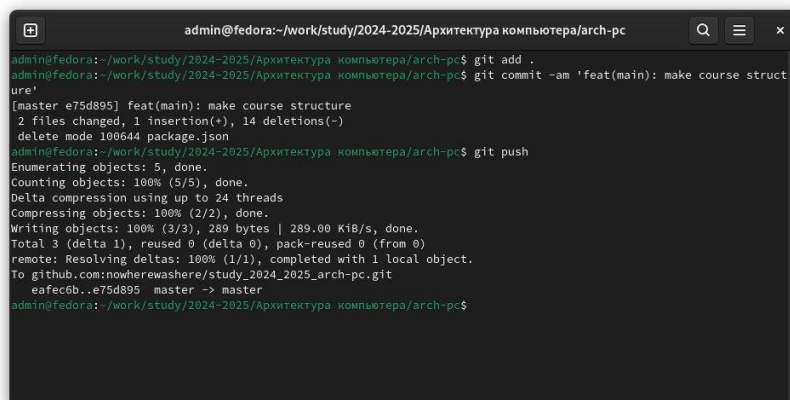
A terminal window with a dark background. The title bar shows 'admin@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc'. The terminal shows the execution of 'git add .', 'git commit -am "feat(main): make course structure"', and 'git push'. It displays the commit hash 'e75d895', file changes ('2 files changed, 1 insertion(+), 14 deletions(-)'), and the push progress to the remote repository 'github.com:nowherewasere/study_2024_2025_arch-pc.git'.

Рис. 4.11: Отправка изменений на удаленный репозиторий

4.6 Задания для самостоятельной работы

Через терминал отправляю предыдущий отчет по лабораторной работе на свой удаленный репозиторий в GitHub (рис. 4.12), затем проверяю изменения на самом GitHub. (рис. 4.13)

```
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am "feat(main): upload reports"
[master ffabf0b] feat(main): upload reports
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report.pdf
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 24 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 657 bytes | 657.00 KiB/s, done.
Total 8 (delta 2), reused 1 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:nowherewashere/study_2024_2025_arch-pc.git
 e75d895..ffabf0b master -> master
admin@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.12: Отправка изменений на удаленный репозиторий

master	1 Branch	0 Tags	Go to file	Add file	Code
nowherewashere	feat(main): make course structure	e75d895 · 2 minutes ago	2 Commits		
config	Initial commit	15 minutes ago			
template	Initial commit	15 minutes ago			
.gitattributes	Initial commit	15 minutes ago			
.gitignore	Initial commit	15 minutes ago			
.gitmodules	Initial commit	15 minutes ago			
CHANGELOG.md	Initial commit	15 minutes ago			
COURSE	feat(main): make course structure	2 minutes ago			
LICENSE	Initial commit	15 minutes ago			
Makefile	Initial commit	15 minutes ago			
README.en.md	Initial commit	15 minutes ago			
README.git-flow.md	Initial commit	15 minutes ago			
README.md	Initial commit	15 minutes ago			

Рис. 4.13: Проверка выполненных изменений на GitHub.

5 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с реализацией VSC git.

Список литературы

1. Лабораторная работа
2. Курс на ТУИС
3. Методические указания
4. Шаблон выполнения лабораторной работы