

# **отчёт по лабораторной работе**

Исаев Рамазан Курбанович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
4.1	Техническое обеспечение . . . . .	9
4.2	Базовая настройка Git . . . . .	9
4.3	Настройка параметров utf-8 . . . . .	10
4.4	Создание ключей . . . . .	11
4.5	Создание репозитория . . . . .	11
<b>5</b>	<b>Выводы</b>	<b>14</b>
<b>6</b>	<b>Список литературы</b>	<b>15</b>

# Список иллюстраций

4.1	1 Предварительная конфигурация Git . . . . .	9
4.2	2 Настройка параметров Git . . . . .	10
4.3	3 Создание пары ssh ключей. . . . .	11
4.4	4 активирован . . . . .	11
4.5	5 репозиторий . . . . .	12
4.6	6 копирование . . . . .	12
4.7	7 отправка изменений . . . . .	13
4.8	каталог курса . . . . .	13

## **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. 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.



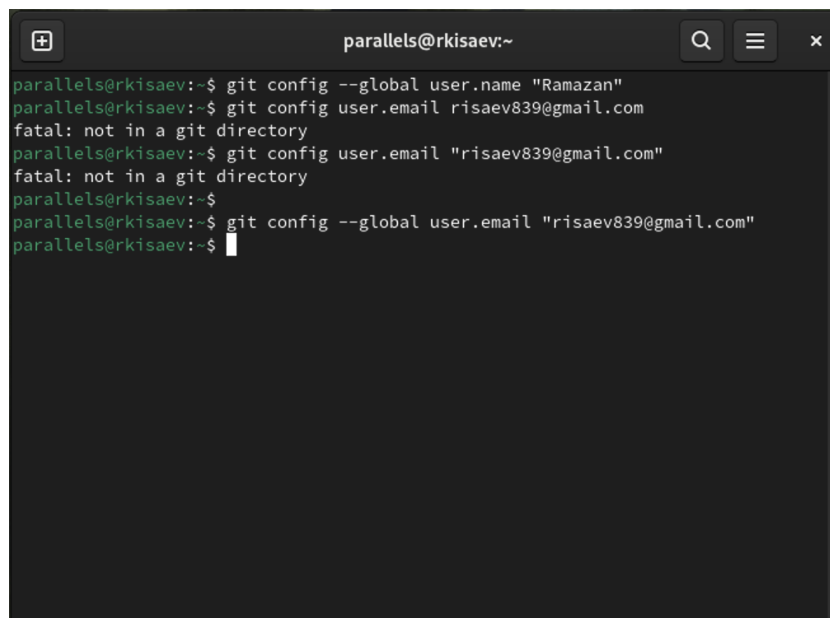
## 4 Выполнение лабораторной работы

### 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 1)

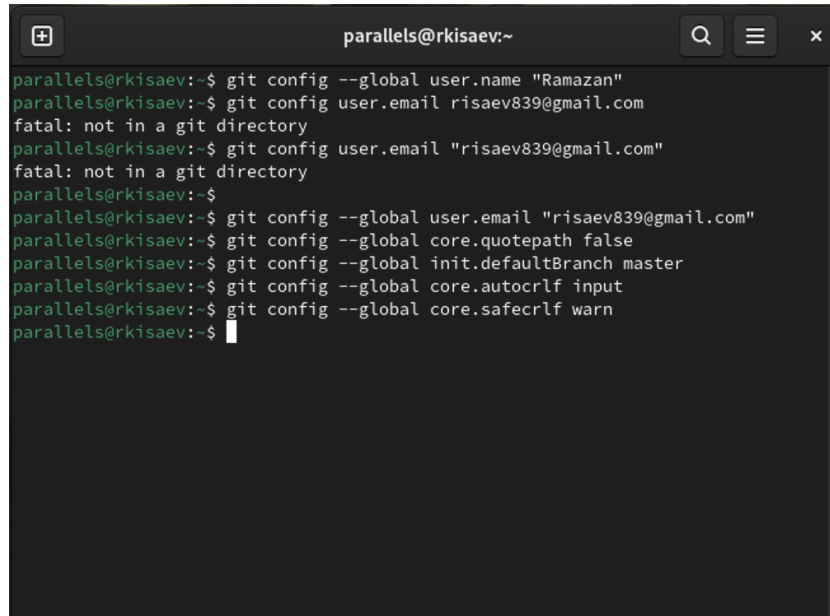
A screenshot of a terminal window titled "parallels@rkisaev:~". The terminal shows a series of Git configuration commands and their outputs. The commands are: "git config --global user.name 'Ramazan'", "git config user.email risaev839@gmail.com", "git config user.email 'risaev839@gmail.com'", and "git config --global user.email 'risaev839@gmail.com'". The outputs for the first two commands are "fatal: not in a git directory". The terminal has a dark background with light green text. The window has standard Linux window controls (minimize, maximize, close) and a search icon in the top right corner.

```
parallels@rkisaev:~  
parallels@rkisaev:~$ git config --global user.name "Ramazan"  
parallels@rkisaev:~$ git config user.email risaev839@gmail.com  
fatal: not in a git directory  
parallels@rkisaev:~$ git config user.email "risaev839@gmail.com"  
fatal: not in a git directory  
parallels@rkisaev:~$  
parallels@rkisaev:~$ git config --global user.email "risaev839@gmail.com"  
parallels@rkisaev:~$
```

Рис. 4.1: 1 Предварительная конфигурация Git

Далее настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf (рис. 2)

### 4.3 Настройка параметров utf-8

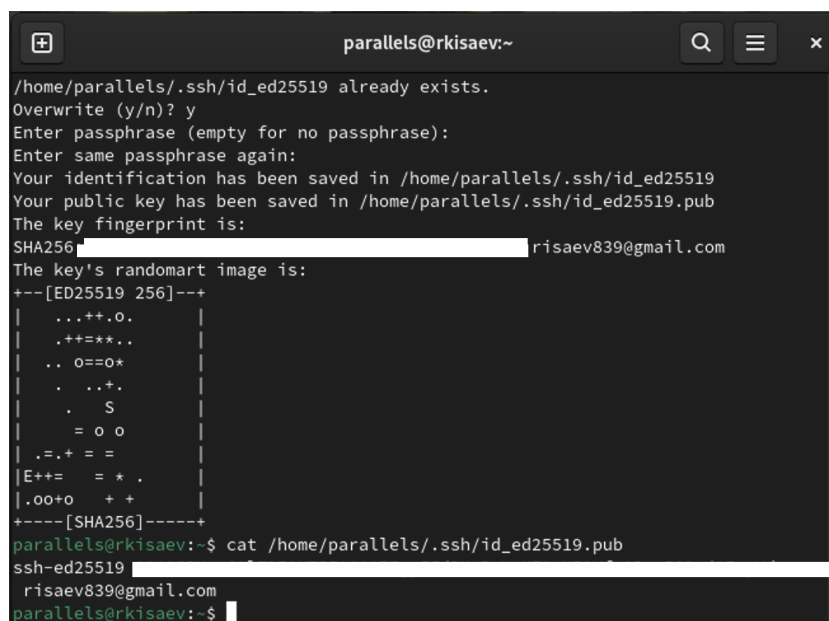
A terminal window titled 'parallels@rkisaev:~' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
parallels@rkisaev:~$ git config --global user.name "Ramazan"
parallels@rkisaev:~$ git config user.email risaev839@gmail.com
fatal: not in a git directory
parallels@rkisaev:~$ git config user.email "risaev839@gmail.com"
fatal: not in a git directory
parallels@rkisaev:~$
parallels@rkisaev:~$ git config --global user.email "risaev839@gmail.com"
parallels@rkisaev:~$ git config --global core.quotepath false
parallels@rkisaev:~$ git config --global init.defaultBranch master
parallels@rkisaev:~$ git config --global core.autocrlf input
parallels@rkisaev:~$ git config --global core.safecrlf warn
parallels@rkisaev:~$
```

Рис. 4.2: 2 Настройка параметров Git

Далее создаю пару ssh ключей для интеграции с платформой GitHub (рис. 3)

## 4.4 Создание ключей



```
parallels@rkisaev:~  
/home/parallels/.ssh/id_ed25519 already exists.  
Overwrite (y/n)? y  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/parallels/.ssh/id_ed25519  
Your public key has been saved in /home/parallels/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256: [redacted] risaev839@gmail.com  
The key's randomart image is:  
+--[ED25519 256]--+  
|    . . . . .  
|    . . . . .  
|    . . . . .  
|    . . . . .  
|    . . . . .  
|    . . . . .  
|    . . . . .  
|    . . . . .  
|    . . . . .  
|    . . . . .  
+-----+  
parallels@rkisaev:~$ cat /home/parallels/.ssh/id_ed25519.pub  
ssh-ed25519 [redacted]  
[redacted] risaev839@gmail.com  
parallels@rkisaev:~$
```

Рис. 4.3: 3 Создание пары ssh ключей.

Ключ добавлен (рис. 4)



Рис. 4.4: 4 активирован

## 4.5 Создание репозитория

Репозиторий у меня был создан (рис 5)

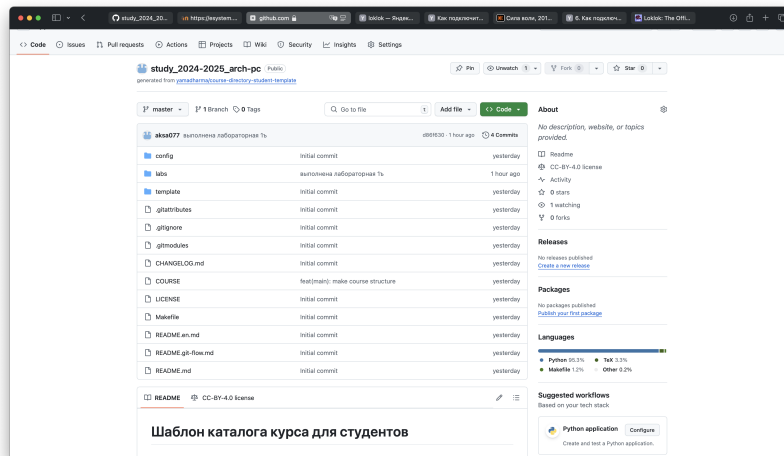


Рис. 4.5: 5 репозиторий

## 4.6

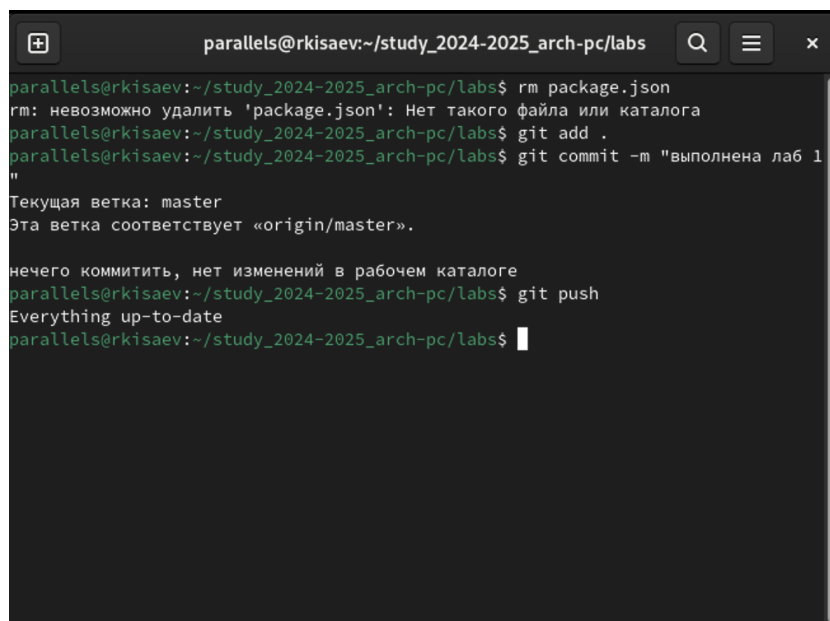
Копирование репозитория на рабочую среду (рис 6)

```
parallels@rkisaev:~$ cat /home/parallels/.ssh/id_ed25519.pub
ssh-ed25519 [redacted]
rkisaev839@gmail.com
parallels@rkisaev:~$ ssh -T git@github.com
The authenticity of host 'github.com (4.225.11.194)' can't be established.
ED25519 key fingerprint is SHA256:[redacted]
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.
Hi aksa077! You've successfully authenticated, but GitHub does not provide shell
access.
parallels@rkisaev:~$ git clone git@github.com:aksa077/study_2024-2025_arch-pc.git
Клонирование в «study_2024-2025_arch-pc»...
remote: Enumerating objects: 103, done.
remote: Counting objects: 100% (103/103), done.
remote: Compressing objects: 100% (80/80), done.
remote: Total 103 (delta 19), reused 85 (delta 17), pack-reused 0 (from 0)
Получение объектов: 100% (103/103), 1.28 МиБ | 1.05 МиБ/с, готово.
Определение изменений: 100% (19/19), готово.
parallels@rkisaev:~$
```

Рис. 4.6: 6 копирование

Далее я удалил файлы package.json, настроил каталог курса и отправил изменения на удаленный репозиторий (рис 7)

## 4.7



```
parallels@rkisaev:~/study_2024-2025_arch-pc/labs$ rm package.json
rm: невозможно удалить 'package.json': Нет такого файла или каталога
parallels@rkisaev:~/study_2024-2025_arch-pc/labs$ git add .
parallels@rkisaev:~/study_2024-2025_arch-pc/labs$ git commit -m "выполнена лаб 1"
Текущая ветка: master
Эта ветка соответствует «origin/master».

ничего коммитить, нет изменений в рабочем каталоге
parallels@rkisaev:~/study_2024-2025_arch-pc/labs$ git push
Everything up-to-date
parallels@rkisaev:~/study_2024-2025_arch-pc/labs$
```

Рис. 4.7: 7 отправка изменений

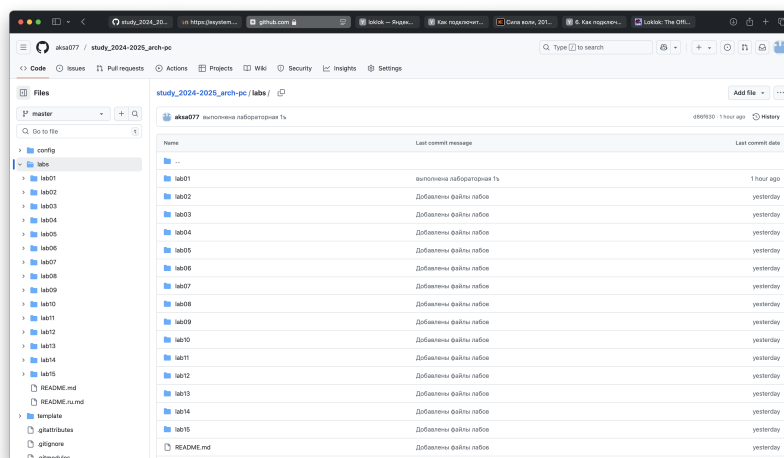


Рис. 4.8: каталог курса

## 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.

## 6 Список литературы

GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix). Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science). Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).