

# **Отчёт по лабораторной работе №2**

**Дисциплина: компьютерные науки и технологии программирования**

Мирзоян Ян Игоревич

# 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	Настройка GitHub . . . . .	9
4.2	Базовая настройка Git . . . . .	9
4.3	Создание SSH-ключа . . . . .	10
4.4	Создание рабочего пространства и репозитория курса на основе шаблона . . . . .	11
4.5	Создание репозитория курса на основе шаблона . . . . .	12
4.6	Настройка каталога курса . . . . .	12
4.7	Задания для самостоятельной работы . . . . .	13
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

4.1	Рисунок 1. Создал учетную запись git . . . . .	9
4.2	Рисунок 2. Указал имя и email-адрес аккаунта с репозиторием . .	9
4.3	Рисунок 3. Настроил utf-8 в выводе сообщений git . . . . .	10
4.4	Рисунок 4. Задал имя мастер для начальной ветки и настроила параметры safecrlf и autocrlf . . . . .	10
4.5	Рисунок 5. Сгенерировал открытый и приватный ключи . . . . .	10
4.6	Рисунок 6. Скопировал ключ из консоли в буфер обмена . . . . .	10
4.7	Рисунок 7. Вставил новый ключ . . . . .	11
4.8	Рисунок 8. Создал ключ . . . . .	11
4.9	Рисунок 9. Создал каталог для дисциплины . . . . .	11
4.10	Рисунок 10. Создал репозиторий курса . . . . .	12
4.11	Рисунок 11. Перешел в каталог курса и клонировал репозиторий .	12
4.12	Рисунок 12. Перешел в каталог, удалил лишние файлы и создал каталоги . . . . .	12
4.13	Рисунок 13. Ввел данные команды . . . . .	12
4.14	Рисунок 14. Отправил файлы на сервер . . . . .	12
4.15	Рисунок 15. Проверил корректность создания файлов иерархии рабочего пространства . . . . .	13
4.16	Рисунок 16. Создал отчет по выполнению первой лабораторной работы в соответствующем каталоге . . . . .	13
4.17	Рисунок 17. Создал отчет по выполнению второй лабораторной работы в соответствующем каталоге . . . . .	14
4.18	Рисунок 18. Загрузил файлы на github . . . . .	14

## Список таблиц

# 1 Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе с системой git.

## 2 Задание

1. Настройка GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства и репозитория курса на основе шаблона.
5. Создание репозитория курса на основе шаблона.
6. Настройка каталога курса.
7. Выполнение заданий для самостоятельной работы.

### 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 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 локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.



## 4 Выполнение лабораторной работы

### 4.1 Настройка GitHub

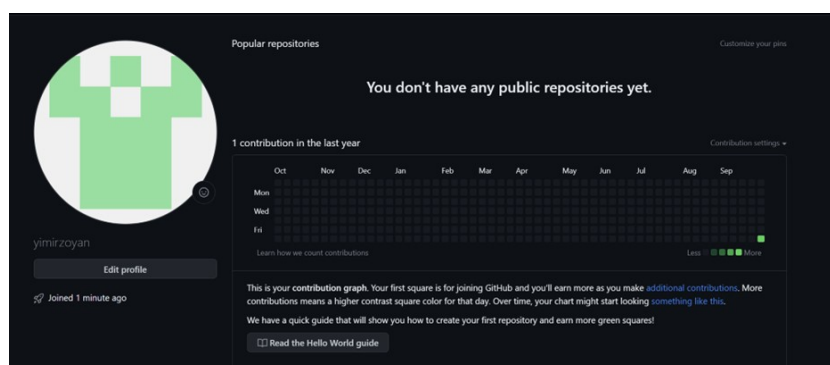


Рис. 4.1: Рисунок 1. Создал учетную запись git

### 4.2 Базовая настройка Git

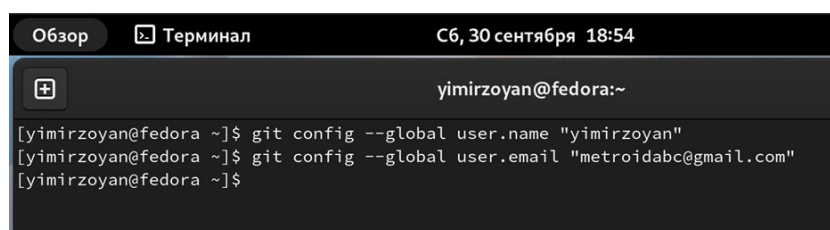


Рис. 4.2: Рисунок 2. Указал имя и email-адрес аккаунта с репозиторием

```
[yimirzoyan@fedora ~]$ git config --global core.quotepath false
[yimirzoyan@fedora ~]$
```

Рис. 4.3: Рисунок 3. Настроил utf-8 в выводе сообщений git

```
[yimirzoyan@fedora ~]$ git config --global init.defaultBranch master
[yimirzoyan@fedora ~]$ git config --global core.autocrlf
[yimirzoyan@fedora ~]$ git config --global core.autocrlf input
[yimirzoyan@fedora ~]$ git config --global core.safecrlf warn
[yimirzoyan@fedora ~]$
```

Рис. 4.4: Рисунок 4. Задал имя мастер для начальной ветки и настроила параметры safecrlf и autocrlf

## 4.3 Создание SSH-ключа

```
[yimirzoyan@fedora ~]$ ssh-keygen -C "Ян Мирзоян metroidabc@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/yimirzoyan/.ssh/id_rsa):
Created directory '/home/yimirzoyan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/yimirzoyan/.ssh/id_rsa
Your public key has been saved in /home/yimirzoyan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:GiSDI5gRk3HQmU78FELqVSDvTmoCZSMLvZF3Ne6gCUs Ян Мирзоян metroidabc@gmail.com
The key's randomart image is:
+---[RSA 3072]-----+
|B@==oo .o          |
|o0%o+ .. .         |
|*EOo.o .           |
|.==.* o            |
|...oo . S          |
|. + o              |
|.o . .             |
|o                  |
|                   |
+---[SHA256]-----+
[yimirzoyan@fedora ~]$
```

Рис. 4.5: Рисунок 5. Сгенерировал открытый и приватный ключи

```
[yimirzoyan@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[yimirzoyan@fedora ~]$ S
```

Рис. 4.6: Рисунок 6. Скопировал ключ из консоли в буфер обмена

## Add new SSH Key

Title

Laptop

Laptop

Authentication Key

Key

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC+WaEpmYVWxZKl8mPwUNIECG/rBjtwg30qVB89ZP3bbRAALzPW
/P/Ryn6eReFrKyzfiQBuaKMM6MVqrSojWA4mD3mhpEXcmpqF1vBg9lGzjcCRxL+c5SEob0wE2NQ2qoLFWiywI53Hsu
F7zZxI7lKPdLmG/FcuZt/GAcRSih9XVUgm1AKuJG0EV8pVS5UKn26UTqj/AKxHsWtMHisWVpsqI3tEK
/ISL/aPOVYNUG4Q854Wnos7
/yIXJeKr6i0+2Nz4dBzQsIPJajqg2Le9qaWLEFWXmS2A5ubEuGEiAET2hXTsWedrQHUIOIOYKAHD
/EZLdoFuOtkRXqmd9/HdIo/FOsx
/W2fme4KI7PM7jIZY80iMQZqoxUuKKMXij447YHsOVanAnAZgmIGkdP1UXAv1raNh7cSfQZ
/cSpVU8XNu3vRn22Z2ZWIW9wHcwpFAxNAIqGcaS6ueWRVwkCLTfw3+CVPBbMLyDWnmSZEF5SgxZWwsANCFLZd2
sMqg7/F+k= Ян Мирзоян metroidabc@gmail.com
```

Add SSH key

Рис. 4.7: Рисунок 7. Вставил новый ключ

## SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

### Authentication Keys

 **Laptop**  
SHA256: I1JsIFLZXVPYt1889te69wxjvqJYApI3XWZw0zDJU1U  
Added on Sep 30, 2023  
Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Рис. 4.8: Рисунок 8. Создал ключ

## 4.4 Создание рабочего пространства и репозитория курса на основе шаблона

```
[yimirzoyan@fedora ~]$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
[yimirzoyan@fedora ~]$
```

Рис. 4.9: Рисунок 9. Создал каталог для дисциплины

## 4.5 Создание репозитория курса на основе шаблона

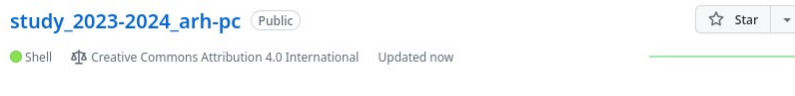


Рис. 4.10: Рисунок 10. Создал репозиторий курса

## 4.6 Настройка каталога курса

```
[yimirzoyan@fedora Архитектура компьютера]$ cd ~/work/study/2023-2024/"Архитектура компь  
ютера"  
[yimirzoyan@fedora Архитектура компьютера]$ git clone --recursive git@github.com:yimirzo  
yan/study_2023-2024_arh-pc arh-pc
```

Рис. 4.11: Рисунок 11. Перешел в каталог курса и клонировал репозиторий

```
[yimirzoyan@fedora arh-pc]$ rm package.json  
[yimirzoyan@fedora arh-pc]$ echo arh-pc > COURSE  
[yimirzoyan@fedora arh-pc]$ make
```

Рис. 4.12: Рисунок 12. Перешел в каталог, удалил лишние файлы и создал каталоги

```
[yimirzoyan@fedora arh-pc]$ git add .  
[yimirzoyan@fedora arh-pc]$ git commit -am 'feat(main): make course structure'
```

Рис. 4.13: Рисунок 13. Ввел данные команды

```
[yimirzoyan@fedora arh-pc]$ git push  
Перечисление объектов: 5, готово.  
Подсчет объектов: 100% (5/5), готово.  
При сжатии изменений используется до 8 потоков  
Сжатие объектов: 100% (2/2), готово.  
Запись объектов: 100% (3/3), 286 байтов | 286.00 КиБ/с, готово.  
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано паке  
тов 0  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To github.com:yimirzoyan/study_2023-2024_arh-pc  
bedac0d..eab8a9f master -> master
```

Рис. 4.14: Рисунок 14. Отправил файлы на сервер


 yimirzoyan feat(main): make course structure	eab8a9f 7 minutes ago	🕒 2 commits
📁 config	Initial commit	34 minutes ago
📁 template	Initial commit	34 minutes ago
📄 .gitattributes	Initial commit	34 minutes ago
📄 .gitignore	Initial commit	34 minutes ago
📄 .gitmodules	Initial commit	34 minutes ago
📄 CHANGELOG.md	Initial commit	34 minutes ago
📄 COURSE	feat(main): make course structure	7 minutes ago
📄 LICENSE	Initial commit	34 minutes ago
📄 Makefile	Initial commit	34 minutes ago
📄 README.en.md	Initial commit	34 minutes ago
📄 README.git-flow.md	Initial commit	34 minutes ago
📄 README.md	Initial commit	34 minutes ago

Рис. 4.15: Рисунок 15. Проверил корректность создания файлов иерархии рабочего пространства

## 4.7 Задания для самостоятельной работы

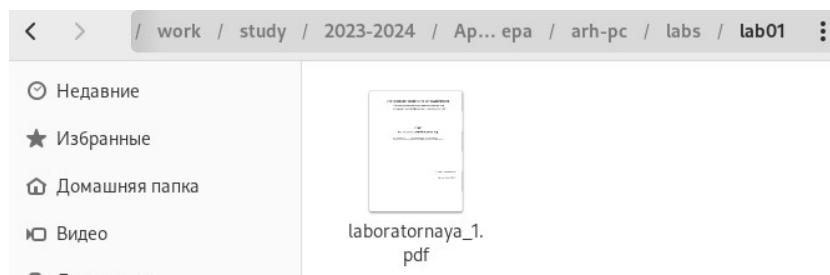


Рис. 4.16: Рисунок 16. Создал отчет по выполнению первой лабораторной работы в соответствующем каталоге

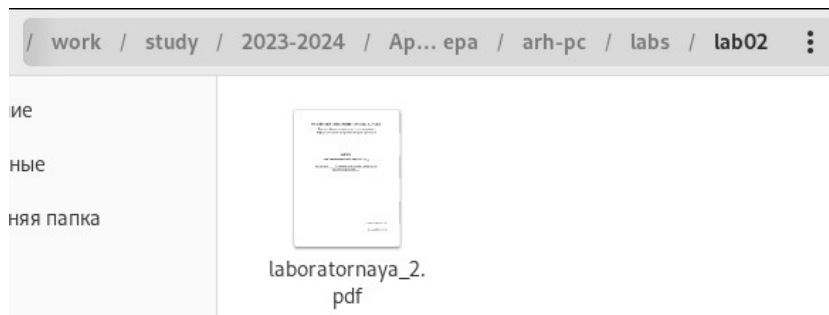


Рис. 4.17: Рисунок 17. Создал отчет по выполнению второй лабораторной работы в соответствующем каталоге

```
[yimirzoyan@fedora arh-pc]$ git add .
[yimirzoyan@fedora arh-pc]$ git commit -am 'feat(main): make course structure'
[master eab8a9f] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
[yimirzoyan@fedora arh-pc]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 286 байтов | 286.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пак
етов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:yimirzoyan/study_2023-2024_arh-pc
bedac0d..eab8a9f master -> master
[yimirzoyan@fedora arh-pc]$ git add .
[yimirzoyan@fedora arh-pc]$ git commit -am 'feat(main): make reports'
[master fe981ef] feat(main): make reports
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/laboratornaya_1.pdf
create mode 100644 labs/lab02/laboratornaya_2.pdf
[yimirzoyan@fedora arh-pc]$ git push
Перечисление объектов: 8, готово.
Подсчет объектов: 100% (8/8), готово.
```

Рис. 4.18: Рисунок 18. Загрузил файлы на github

## 5 Выводы

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

## Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. 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/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
15. 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).
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).