

Лабораторная работа №2

дисциплина: Архитектура компьютера

Гаврилейко Алина Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Настройка GitHub	9
4.2	Создание SSH-ключа	10
4.3	Создание рабочего пространства и репозитория курса на основе шаблона	12
4.4	Настройка каталога курса	14
4.5	Задания для самостоятельной работы	15
4.6	Вывод	16
4.7	Список литературы	16
5	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Создание аккаунта на GitHub	9
4.2	Предварительная конфигурация Git	9
4.3	Предварительная конфигурация Git	10
4.4	Настройка кодировки	10
4.5	Создание имени для начальной ветки	10
4.6	Параметр autocrlf	10
4.7	Параметр safecrlf	10
4.8	Создание SSH-ключа	11
4.9	Установка команды xclip	11
4.10	Копирование содержимого файла	12
4.11	Вставка SSH-ключа	12
4.12	Создание рабочего пространства	12
4.13	Страница создания репозитория	13
4.14	Переход в каталог курса	13
4.15	Клонирование репозитория	13
4.16	Окно с ссылкой на копирование репозитория	14
4.17	Перемещение между директориями	14
4.18	Удаление файлов	14
4.19	Создание каталогов	14
4.20	Добавление и сохранение изменений на сервере	15
4.21	Выгрузка изменений на сервер	15
4.22	Создание файла	15
4.23	Добавление файла	15
4.24	Отправка изменений в центральный репозиторий	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 Настройка GitHub

Создаю учетную запись на сайте GitHub, заполняю данные учетной записи и создаю аккаунт (рис. 4.1).

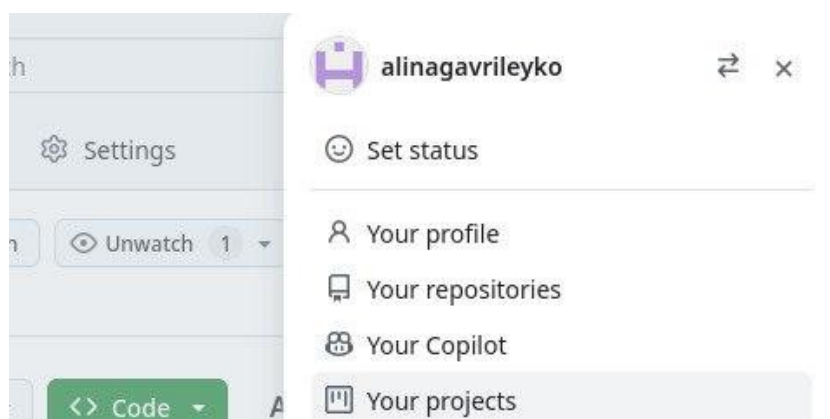


Рис. 4.1: Создание аккаунта на GitHub

Открываю виртуальную машину, затем терминал и делаю предварительную конфигурацию Git. Вбиваю команду `git config --global user.name "`, указывая свое имя и фамилию (рис. 4.2).

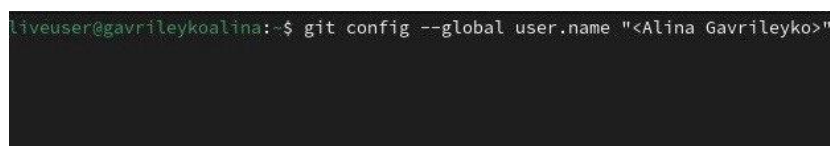


Рис. 4.2: Предварительная конфигурация Git

затем пишу команду `git config --global user.email "`, указывая почту владельца.

(рис. 4.3).

```
liveuser@gavrileykoalina:~$ git config --global user.email "<gavrileykoalina@gmail.com>"
```

Рис. 4.3: Предварительная конфигурация Git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 4.4).

```
liveuser@gavrileykoalina:~$ git config --global user.email "<gavrileykoalina@gmail.com>"
liveuser@gavrileykoalina:~$ git config --global core.quotepath false
```

Рис. 4.4: Настройка кодировки

Задаю имя master для начальной ветки (рис. 4.5).

```
liveuser@gavrileykoalina:~$ git config --global user.email "<gavrileykoalina@gmail.com>"
liveuser@gavrileykoalina:~$ git config --global core.quotepath false
liveuser@gavrileykoalina:~$ git config --global init.defaultBranch master
```

Рис. 4.5: Создание имени для начальной ветки

Задаю параметр autocrlf со значением input (рис. 4.6).

```
liveuser@gavrileykoalina:~$ git config --global user.email "<gavrileykoalina@gmail.com>"
liveuser@gavrileykoalina:~$ git config --global core.quotepath false
liveuser@gavrileykoalina:~$ git config --global init.defaultBranch master
liveuser@gavrileykoalina:~$ git config --global core.autocrlf input
```

Рис. 4.6: Параметр autocrlf

Задаю параметр safecrlf (рис. 4.7).

```
liveuser@gavrileykoalina:~$ git config --global user.email "<gavrileykoalina@gmail.com>"
liveuser@gavrileykoalina:~$ git config --global core.quotepath false
liveuser@gavrileykoalina:~$ git config --global init.defaultBranch master
liveuser@gavrileykoalina:~$ git config --global core.autocrlf input
liveuser@gavrileykoalina:~$ git config --global core.safecrlf warn
```

Рис. 4.7: Параметр safecrlf

4.2 Создание SSH-ключа

Для идентификации пользователя на сервере репозитория нужно сгенерировать пару ключей – приватный и открытый. Для этого в терминал ввожу команду

ssh-keygen -C 'Имя Фамилия, work@email' и ввожу туда имя, фамилию, почту.
Ключ автоматически сохранится в каталоге ~/.ssh/. (рис. 4.8).

```
liveuser@gavrileykoalina:~$ ssh-keygen -C "Alina Gavrileyko <gavrileykoalina@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_ed25519):
Created directory '/home/liveuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_ed25519
Your public key has been saved in /home/liveuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:ja8qS5/ZruivIqjTKslh9odrSxxzBo06tyBfgaliGKk Alina Gavrileyko <gavrileykoalina@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|
| . oo
|o .oo.
|o..... o
|E=..+o S .
|o*=. *
|=-o+=o .
|*..+=.+ + .
|+=..=B++
+----[SHA256]-----
```

Рис. 4.8: Создание SSH-ключа

Далее на дистрибутив fedora необходимо скачать команду xclip. Устанавливаю ее с помощью команды sudo dnf install xclip (рис. 4.9).

```
liveuser@gavrileykoalina:~$ sudo dnf install xclip
Fedora 40 - x86_64                               250 kB/s | 20 MB   01:21
Fedora 40 openh264 (From Cisco) - x86_64        224 B/s | 1.4 kB  00:06
Fedora 40 - x86_64 - Updates                     290 kB/s | 10 MB  00:36
Dependencies resolved.
=====
Package            Architecture Version                      Repository      Size
=====
Installing:
xclip              x86_64      0.13-21.git11cba61.fc40    fedora          37 k
=====
Transaction Summary
=====
Install 1 Package

Total download size: 37 k
Installed size: 62 k
Is this ok [y/N]: y
Downloading Packages:
xclip-0.13-21.git11cba61.fc40.x86_64.rpm         33 kB/s | 37 kB   00:01
-----
Total                                           10 kB/s | 37 kB   00:03
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : xclip-0.13-21.git11cba61.fc40.x86_64 1/1
  Running scriptlet: xclip-0.13-21.git11cba61.fc40.x86_64 1/1

Installed:
  xclip-0.13-21.git11cba61.fc40.x86_64

Complete!
```

Рис. 4.9: Установка команды xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью команды xclip (рис. 4.10).

```
liveuser@gavrileykoalina: ~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
```

Рис. 4.10: Копирование содержимого файла

Далее захожу на сайт GitHub, открываю свой профиль и выбираю страницу “SSH and GPG keys”. Нажимаю на кнопку “New SSH key”, вставляю в поле скопированный ключ, в поле ‘Title’ указываю имя ключа, затем нажимаю “Add SSH- key”. (рис. 4.11).

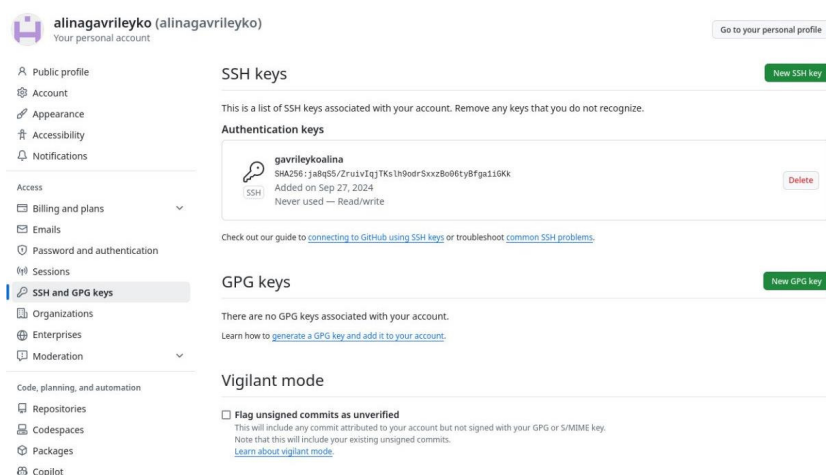


Рис. 4.11: Вставка SSH-ключа

4.3 Создание рабочего пространства и репозитория курса на основе шаблона

Открываю терминал, создаю директорию, рабочее пространство с помощью утилиты mkdir, с помощью ключа -p создаю рекурсивно все директории после домашней ~/work/study/2024-2025/”Архитектура компьютера”. (рис. 4.12).

```
liveuser@gavrileykoalina: ~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютеров"
```

Рис. 4.12: Создание рабочего пространства

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>, выбираю use

this template, чтобы использовать данный шаблон для своего репозитория. В открывшемся окне задаю название репозитория: study_2024-2025_arch-pc и создаю репозиторий, нажимая на кнопку “Create repository” (рис. 4.13).

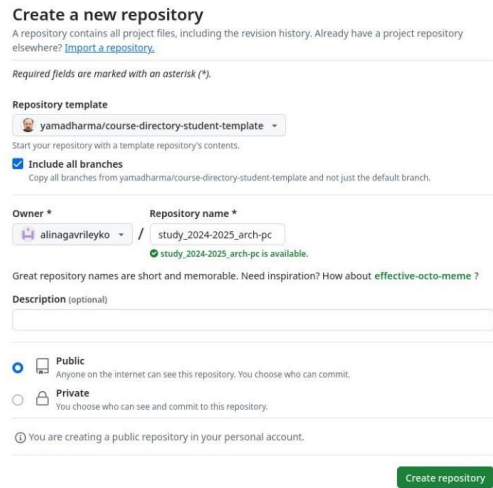


Рис. 4.13: Страница создания репозитория

Через терминал перехожу в созданный каталог курса с помощью утилиты cd (рис. 4.14).

```
liveuser@gavrileykoalina:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютеров"
```

Рис. 4.14: Переход в каталог курса

Далее клонирую созданный репозиторий (рис. 4.15).

```
$ git clone --recursive git@github.com:alinagavrileyko/study_2024-2025_arch-pc.git arch-pc
Cloning into 'arch-pc'...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:0D1V2wvV6T03hptz1sf/zLOdZrH5vHdKr4UvCoQu.
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: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 34 (delta 11, reused 18 (delta 0), pack-reused 0 (from 0))
Receiving objects: 100% (34/34), 19.58 KiB | 1.78 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharm/academic-presentation-random-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharm/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/home/liveuser/work/study/2024-2025/Архитектура компьютеров/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 | 1.00 MiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/liveuser/work/study/2024-2025/Архитектура компьютеров/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 | 997.00 KiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22ef67b3e0495707d02ef561ab185f5c748'
```

Рис. 4.15: Клонирование репозитория

Ссылку для клонирования копирую на странице созданного репозитория, перейдя сначала в окно 'code', выбрав затем вкладку 'SSH' (рис. 4.16).

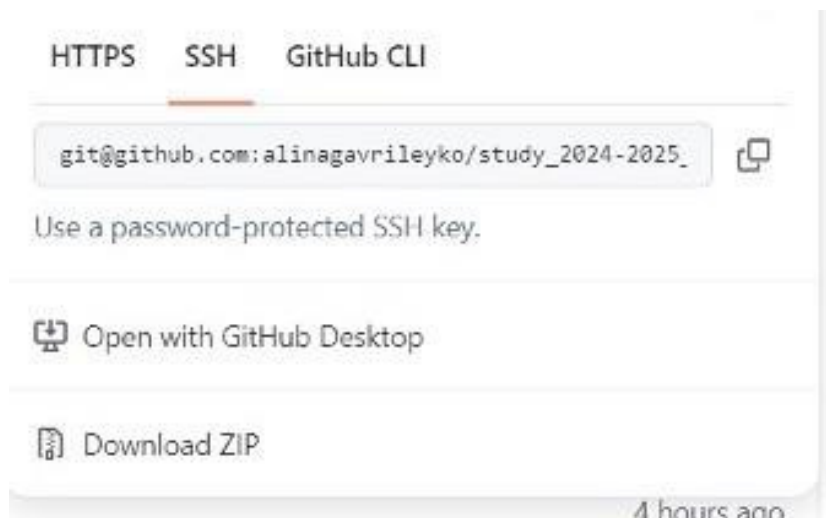


Рис. 4.16: Окно с ссылкой на копирование репозитория

4.4 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd (рис. 4.17).

```
l1veuser@gavrileykoalina: /work/study/2024-2025/Архитектура компьютеров$ cd ~/work/study/2024-2025/"Архитектура компьютеров"/arch-pc
```

Рис. 4.17: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты rm (рис. 4.18).

```
l1veuser@gavrileykoalina: /work/study/2024-2025/Архитектура компьютеров/arch-pc$ rm package.json
```

Рис. 4.18: Удаление файлов

Создаю необходимые каталоги (рис. 4.19).

```
l1veuser@gavrileykoalina: /work/study/2024-2025/Архитектура компьютеров/arch-pc$ echo arch-pc > COURSE
l1veuser@gavrileykoalina: /work/study/2024-2025/Архитектура компьютеров/arch-pc$ make prepare
```

Рис. 4.19: Создание каталогов

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью git add, комментирую и сохраняю изменения на сервере как добавление курса с помощью git commit (рис. 4.20).

```
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ git add .
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ git commit -m 'feat(main): make course structure'
```

Рис. 4.20: Добавление и сохранение изменений на сервере

Затем все отправляю на сервер с помощью команды git push (рис. 4.21).

```
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 341.41 KiB | 598.00 KiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
to github.com:alinagavrileyko/study_2024-2025_arch-pc.git
99f8fbb..ba75952 master -> master
```

Рис. 4.21: Выгрузка изменений на сервер

4.5 Задания для самостоятельной работы

Перехожу в директорию labs/lab02/report, создаю в каталоге файл для отчета по второй лабораторной работе с помощью команды touch (рис. 4.22).

```
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ cd labs/lab02/report
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab02/report$ touch Л02_Гаврилейко_отчёт
```

Рис. 4.22: Создание файла

Перехожу в подкаталог lab01/репорт, затем копирую первую лабораторную работу с помощью команды cp, а далее с помощью ls удостоверяюсь, что скопировано все верно (рис. 4.23).

```
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ cd ~/work/study/2024-2025/Архитектура компьютеров/arch-pc
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc$ cd labs/lab01/report
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ ls -la
total 12
-rw-r--r-- 1 liveuser liveuser 4096 Nov 15 14:01 Л01_Гаврилейко_отчет.pdf
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ cp ~/Downloads/Л01_Гаврилейко_отчет.pdf ./
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ ls
total 12
-rw-r--r-- 1 liveuser liveuser 4096 Nov 15 14:01 Л01_Гаврилейко_отчет.pdf
-rw-r--r-- 1 liveuser liveuser 4096 Nov 15 14:01 Л01_Гаврилейко_отчет.pdf
```

Рис. 4.23: Добавление файла

С помощью команды git add добавляю в коммит файл «Л01_Гаврилейко_отчет», перехожу в директорию, в которой находится отчет, добавляю нужный файл и сохраняю изменения на сервере с помощью команды git commit -m " " и далее с помощью git push -f origin master отправляю изменения в главный репозиторий (рис. 4.24).

```
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ git add 001_faapneeko_orer.pdf
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ git commit -m "Add existing file"
[master 933b97e] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/001_faapneeko_orer.pdf
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$ git push -f origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (0/0), done.
Writing objects: 100% (6/6), 1.03 MiB | 1.76 MiB/s, done.
Total 0 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:conalagavrileyko/study_2024-2025_arch-pc.git
ba75952..933b97e master -> master
liveuser@gavrileykoalina:~/work/study/2024-2025/Архитектура компьютеров/arch-pc/labs/lab01/report$
```

Рис. 4.24: Отправка изменений в центральный репозиторий

4.6 Вывод

При выполнении данной лабораторной работы я изучила практические навыки по работе с системой Git, изучила применение и идеологию средств контроля версий.

4.7 Список литературы

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%20E2%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

5 Выводы

Здесь кратко описываются итоги проделанной работы.

Список литературы