

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: Югай Александр Витальевич

Группа: НПИбд-02-23

МОСКВА

2023 г.

Содержание

- 1. Цель работы**
- 2. Теоретическое введение**
- 3. Выполнение лабораторной работы**
- 4. Вывод**

1. Цель работы

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

2. 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 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.

3. Выполнение лабораторной работы.

3.1 Настройка github.

Создайте учётную запись на сайте <https://github.com/> и заполните основные данные

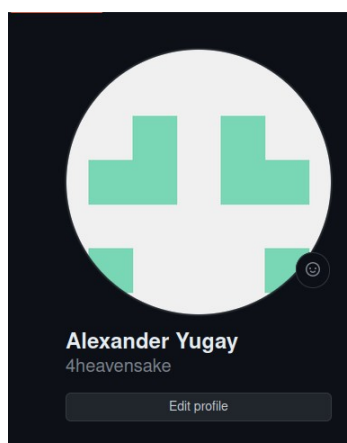


Рис 3.1.1. Профиль github

До выполнения данной лабораторной работы, я уже имел профиль в github и просто зашел в него.

3.2 Базовая настройка git.

Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория

```
avyugayj@ubuntu:~$ git config --global user.name "Alexander Yugay"
avyugayj@ubuntu:~$ git config --global user.email "knifecalledlust33@gmail.com"
```

Рис 3.2.1. Предварительная конфигурация git

Используя данные в лабораторной работе команды, я сделал предварительную конфигурацию на основе моего имени и моей почты

Настроим utf-8 в выводе сообщений git

```
avyugayj@ubuntu:~$ git config --global core.quotepath false
avyugayj@ubuntu:~$
```

Рис 3.2.2. Настройка utf-8

Зададим имя начальной ветки (будем называть её master)

```
avyugayj@ubuntu:~$ git config --global init.defaultBranch master
```

Рис 3.2.3. Дарим имя начальной ветки

Параметр autocrlf:

```
avyugayj@ubuntu:~$ git config --global core.autocrlf input
```

Рис 3.2.4. autocrlf

Параметр safecrlf:

```
avyugayj@ubuntu:~$ git config --global core.safecrlf warn
```

Рис 3.2.5. safecrlf

3.3 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозиториев необходимо генерировать пару ключей (приватный и открытый):

ssh-keygen -C "Имя Фамилия <work@mail>"

```
avyugayj@ubuntu:~$ ssh-keygen -C "Alexander Yugay k
```

Рис 3.3.1. Генерация ключей

С помощью данной в лабораторной работе команды, я сгенерировал пару ключей

Далее необходимо загрузить сгенерённый открытый ключ. Скопировав из локальной консоли ключ в буфер обмена.

```
avyugayj@ubuntu:~$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCuPV  
2r3fwKQCmppy8oKSjJzHHgXL7AL3injhjsw4MmRZ1h  
VpEBkrY2gS9XbElyKTZoZVK67jqop5NSca9ZuXptW8  
JEjxrna+v6sj/GPoQ+SPk65n84cdStvYHr/GrswHwB
```

Рис 3.3.2.

Вывод ключа

Используя команду cat, я вывел в консоль сгенерированный мною ключ

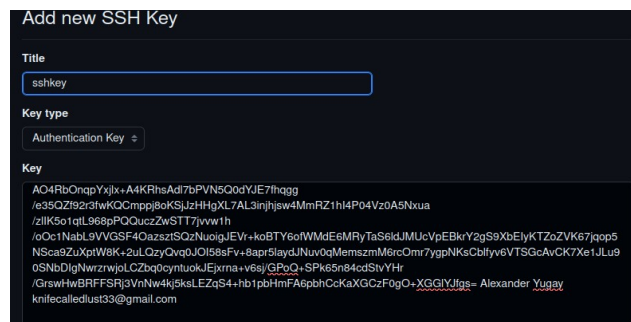


Рис 3.3.3.

Загрузка ключа в github

Следуя инструкциям в лабораторной работе, я загрузил ключ на github

3.4 Сознание рабочего пространства и репозитория курса на основе шаблона

Откройте терминал и создайте каталог для предмета «Архитектура компьютера»

```
avyugayj@ubuntu:~$ mkdir -p ~/work/study/2023-2024/"Архитектура Компьютера"  
avyugayj@ubuntu:~$
```

Рис 3.4.1. Создание ветки каталогов.

Используя команду mkdir с опцией -p, я создал цепочку каталогов.

3.5. Сознание репозитория курса на основе шаблон

В открывшемся окне задайте имя репозитория (Repository name) study_2023–

2024_arh-pc и создайте репозиторий (кнопка Create repository from template)

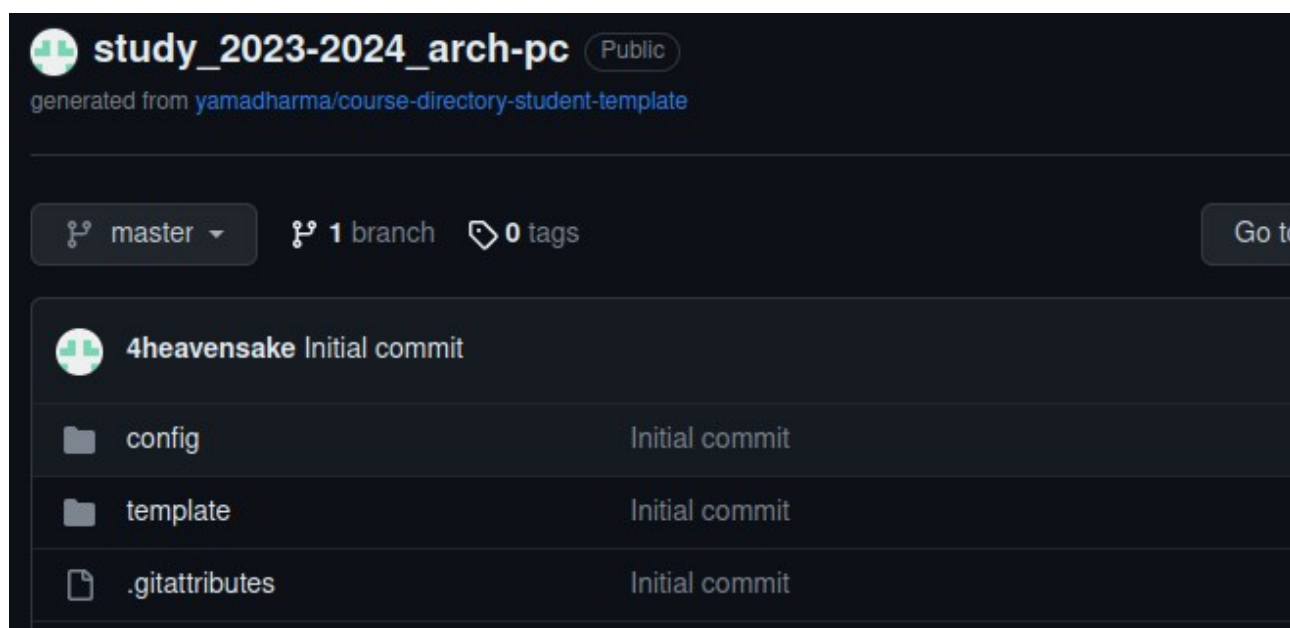


Рис 3.5.1. Создание репозитория по шаблону.

Следуя инструкциям в лабораторной работе, я создал репозиторий по данному мне шаблону

Откройте терминал и перейдите в каталог курса:

```
avyugayj@ubuntu:~$ cd work/study/2023-2024/Архитектура\ Компьютера/
```

Рис 3.5.2. Переход в каталог курса

Клонируйте созданный репозиторий:

```
avyugayj@ubuntu:~/work/study/2023-2024/Архитектура Компьютера$ git clone --recursive
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
```

Рис 3.5.3. Клонирование репозитория

Благодаря команде git clone, я клонировал репозиторий из github в мою систему

3.6 Настройка каталога курса

Перейдите в каталог курса:

```
avyugayj@ubuntu:~/work/study/2023-2024/Архитектура Компьютера$ cd arch-pc/
```

Рис 3.6.1. Переход в каталог курса

Удалите лишние файлы:

```
rm package.json
```

```
avyugayj@ubuntu:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ rm package.json
```

Рис 3.6.2. Удаление лишнего файла

Создайте необходимые каталоги:

```
echo arch-pc > COURSE
```

```
make
```

```
avyugayj@ubuntu:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ echo arch-pc > COURSE
avyugayj@ubuntu:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ make
```

Рис 3.6.3. Создание необходимых каталогов

Благодаря команде echo я создал файл в моем древе каталогов

Отправьте файлы на сервер:

```
git add .
```

```
git commit -am 'feat(main): make course structure'
```

```
git push
```

```
avyugayj@ubuntu:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git add .
avyugayj@ubuntu:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git commit -am
[master cdaa704] feat(main) make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
```

Рис 3.6.4. Отправка файлов в github

Благодаря данным командам из лабораторной работы, я синхронизировал файлы со средой github

Проверьте правильность создания иерархии рабочего пространства в локальном репозитории и на странице github


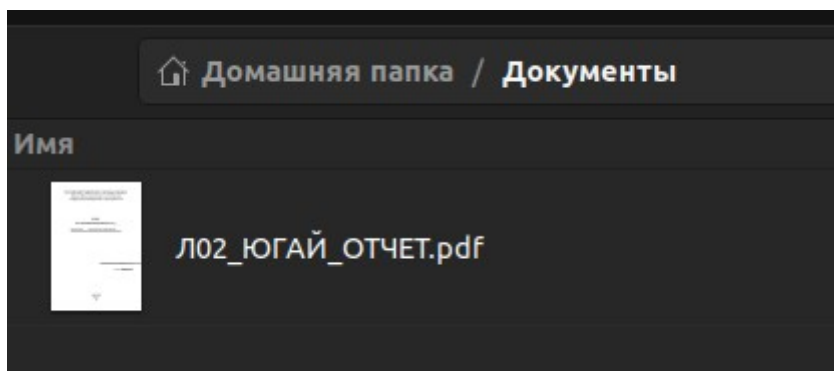
 4heavensake feat(main) make course structure	cdaa704 now	🕒 2 commits
📁 config	Initial commit	4 minutes ago
📁 labs	feat(main) make course structure	now
📁 presentation	feat(main) make course structure	now

Рис 3.6.5. Проверка правильности задания на github

3.6 Задания для самостоятельной работы.

1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report)



Рис

3.6.1.1. Создание отчета в папке “Документы”

```
Л02_ЮГАЙ_ОТЧЕТ.pdf
avyugayj@ubuntu:~/Документы$ cp Л02_ЮГАЙ_ОТЧЕТ.pdf ~/work/study/2023-
avyugayj@ubuntu:~/Документы$
```

Рис 3.6.1.2. Копирование отчета из “Документы” в нужный каталог

```
avyugayj@ubuntu:~/Документы$ ls ~/work/study/2023-2024/Архитектура\
bib image Makefile pandoc report.md Л02_ЮГАЙ_ОТЧЕТ.pdf
```

Рис 3.6.1.3 Проверка выполненных действий.

2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

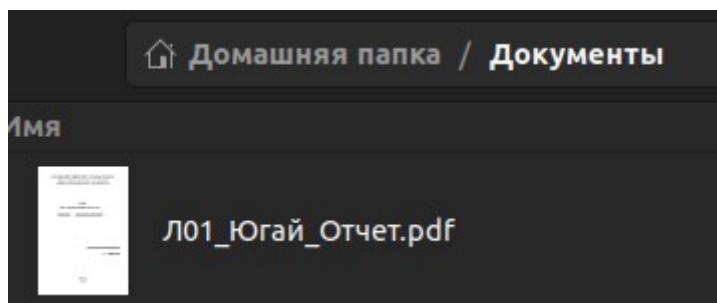


Рис 3.6.2.1 Перенос первого отчета в каталог “Документы”

```
avyugayj@ubuntu: ~/Документы$ cp Л01_Югай_Отчет.pdf ~/work/study/
avyugayj@ubuntu: ~/Документы$
```

Рис 3.6.2.2 Копирование отчета в нужный каталог

```
avyugayj@ubuntu: ~/Документы$ ls ~/work/study/2023-2024/Архитекту
bib image Makefile pandoc report.md Л01_Югай_Отчет.pdf
```

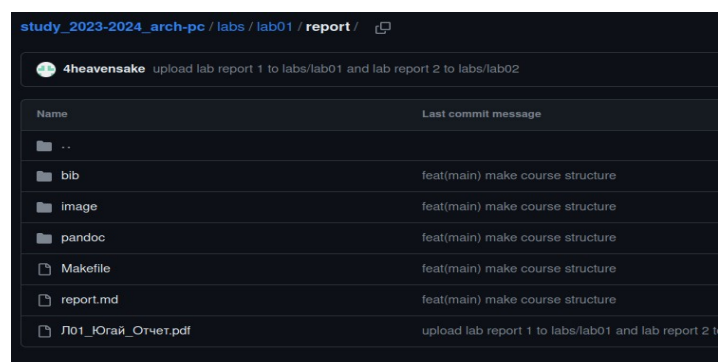
Рис 3.6.2.3. Проверка выполненных действий.

3. Загрузите файлы на github

```
avyugayj@ubuntu: ~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git add .
avyugayj@ubuntu: ~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git commit -m '2'
[master e3a7940] upload lab report 1 to labs/lab01 and lab report 2 to labs/lab02
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Югай_Отчет.pdf
create mode 100644 labs/lab02/report/Л02_ЮГАЙ_ОТЧЕТ.pdf
avyugayj@ubuntu: ~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git push
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (13/13), готово.
```

Рис 3.6.3.1 Загрузка файлов в github

Благодаря уже известным мне командам, я синхронизировал все изменения в системе со средой github



Name	Last commit message
..	
bib	feat(main) make course structure
image	feat(main) make course structure
pandoc	feat(main) make course structure
Makefile	feat(main) make course structure
report.md	feat(main) make course structure
Л01_Югай_Отчет.pdf	upload lab report 1 to labs/lab01 and lab report 2 to labs/lab02

Рис 3.6.3.2

Проверка правильности загрузки.

4. Вывод:

В ходе выполнения лабораторной работы были изучены идеологии и применение средств контроля версий. Также были приобретены практические навыки по работе с системой git