

Отчет по лабораторной работе №2

Дисциплина: архитектура компьютера

Комягин Андрей Николаевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Описание выполнения заданий для самостоятельной работы	12
3.1	Задание 1	12
3.2	Задание 2	12
3.3	Задание 3	13
4	Вывод	14

Список иллюстраций

2.1	Аккаунт на GitHub	6
2.2	Настройка git, имя пользователя и email	6
2.3	Настройка UTF-8	7
2.4	Инициализация начальной ветки	7
2.5	Параметры autocrlf и safecrlf	7
2.6	Генерация ключей	7
2.7	Копирование ключа	7
2.8	Загрузка ключа на GitHub	8
2.9	Создание каталога	8
2.10	Создание репозитория	9
2.11	Клонирование данных	9
2.12	Удаление лишних файлов	9
2.13	Создание необходимых каталогов	10
2.14	Отправка файлов на сервер	10
2.15	Локальный репозиторий (в пути видна правильность иерархии)	10
2.16	Изменившийся репозиторий на GitHub	11
3.1	Перемещение файла 1	12
3.2	Перемещение файла 2	12
3.3	Загрузка файлов на GitHub)	13
3.4	Обновленный репозиторий GitHub	13

Список таблиц

1 Цель работы

Целью работы является знакомство с системой управления версий git, изучение её идеологии и приобретение практических навыков по её использованию.

2 Выполнение лабораторной работы

Создаём учетную запись на сайте **GitHub** (рис. 2.1).

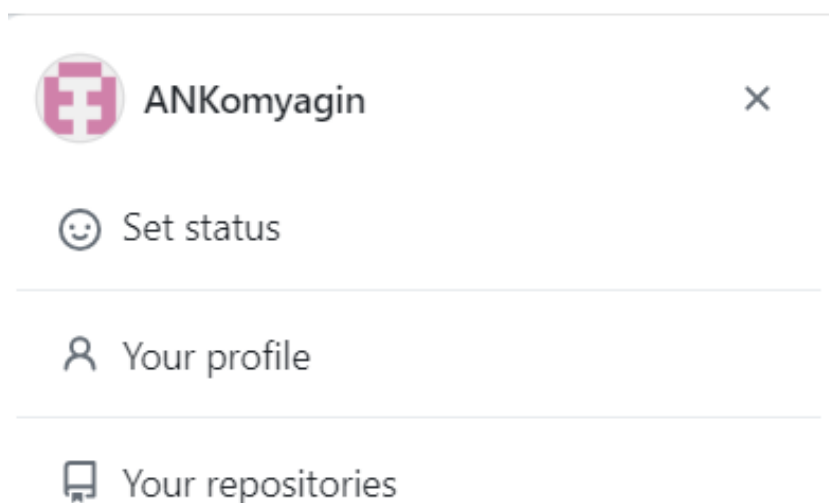


Рис. 2.1: Аккаунт на GitHub

Затем сделаем предварительную конфигурацию **git**. Откроем терминал и выполним некоторые команды. Укажем имя и email владельца репозитория (рис. 2.2).

```
Файл  Правка  Вид  Поиск  Терминал  Справка
[ankomyagin@localhost ~]$ git config --global user.name "ANKomyagin"
[ankomyagin@localhost ~]$ git config --global user.email "<Komyagin12345@mail.ru>"
```

Рис. 2.2: Настройка git, имя пользователя и email

Также настроим UTF-8 в выводе сообщений git (рис. 2.3).

```
[ankomyagin@localhost ~]$ git config --global core.quotePath false
```

Рис. 2.3: Настройка UTF-8

Зададим имя начальной ветки (master) (рис. 2.4).

```
[ankomyagin@localhost ~]$ git config --global init.defaultBranch master
```

Рис. 2.4: Инициализация начальной ветки

Настроим параметры ядра (рис. 2.5).

```
[ankomyagin@localhost ~]$ git config --global core.autocrlf input  
[ankomyagin@localhost ~]$ git config --global core.safecrlf warn
```

Рис. 2.5: Параметры autocrlf и safecrlf

Для идентификации пользователя на сервере сгенерируем пару SSH ключей (рис. 2.6).

```
[ankomyagin@localhost ~]$ ssh-keygen -C "Андрей Комягин <Komyagin12345@mail.ru>"
```

Рис. 2.6: Генерация ключей

Скопируем ключ из локальной консоли в буфер обмена (рис. 2.7) и загружаем его на GitHub (рис. 2.8).

```
[ankomyagin@localhost ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 2.7: Копирование ключа

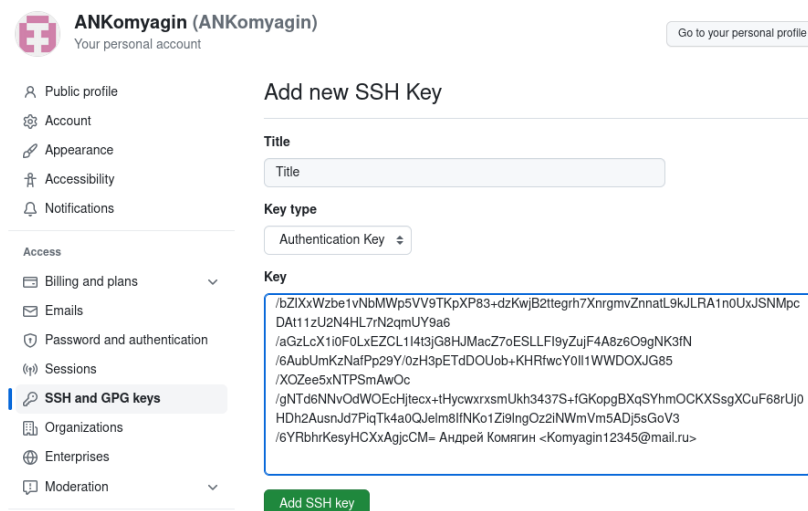


Рис. 2.8: Загрузка ключа на GitHub


Создадим каталог для предмета «Архитектура компьютера» (рис. 2.9)

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

Рис. 2.9: Создание каталога

Создадим репозиторий курса на основе шаблона. Для этого перейдем на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Создаём репозиторий, назвав его «study_2023–2024_arhpc» (рис. 2.10).


Repository template

 yamadharma/course-directory-student-template ▾

Start your repository with a template repository's contents.

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


Owner * **Repository name ***


 ANKomyagin ▾ /


⚠ **Your new repository will be created as study_2023-2024_arh-pc-**
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about **expert-doodle** ?

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. 2.10: Создание репозитория

Откроем терминал, перейдём в каталог курса и клонируем только что созданный репозиторий (рис. 2.11).

```
ankomyagin@localhost:~/work/study/2023-2024/Архитектура компьютера
Файл  Правка  Вид  Поиск  Терминал  Справка
[ankomyagin@localhost Архитектура компьютера]$ git clone --recursive git@github.com:ANKomyagin/study_2023-2024_arh-pc.git arch-pc
Клонирование в «arch-pc»...
```

Рис. 2.11: Клонирование данных

Настроим каталог курса. Сначала перейдем в каталог курса и удалим лишние файлы (package.json) (рис. 2.12).

```
[ankomyagin@localhost ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[ankomyagin@localhost arch-pc]$ rm package.json
```

Рис. 2.12: Удаление лишних файлов

Создадим необходимые каталоги (рис. 2.13).

```
[ankomyagin@localhost arch-pc]$ echo arch-pc > COURSE
[ankomyagin@localhost arch-pc]$ make
```

Рис. 2.13: Создание необходимых каталогов

Сохраним изменения и отправим файлы на сервер (рис. 2.14).

```
[ankomyagin@localhost arch-pc]$ git add .
[ankomyagin@localhost arch-pc]$ git commit -am 'feat(main): make course structure'
[master 72acc49] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
```

Рис. 2.14: Отправка файлов на сервер

Осталось убедиться в правильном создании иерархии рабочего пространства в локальном и серверном репозиториях (рис. 2.15) и (рис. 2.16).



Рис. 2.15: Локальный репозиторий (в пути видна правильность иерархии)

3 Описание выполнения заданий для самостоятельной работы

3.1 Задание 1

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

Для этого переместим файл с лабораторной работой в необходимую папку (рис. 3.1).

```
[ankomyagin@localhost Архитектура компьютера]$ mv ~/Загрузки/Лабораторная работа 2.doc arch-pc/labs/lab02/report
```

Рис. 3.1: Перемещение файла 1

3.2 Задание 2

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

Для этого переместим файл с лабораторной работой в необходимую папку (рис. 3.2).

```
[ankomyagin@localhost Архитектура компьютера]$ mv ~/Загрузки/Лабораторная работа 1.pdf arch-pc/labs/lab01/report
```

Рис. 3.2: Перемещение файла 2

3.3 Задание 3

Загрузите файлы на **GitHub**

Для загрузки файлов на **GitHub** воспользуемся последовательностью команд (рис. 3.3).

```
[ankomyagin@localhost arch-pc]$ git add .
[ankomyagin@localhost arch-pc]$ git commit -am 'new labs'
[master 5e1ad2c] new labs
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Лабораторная_работа_1.pdf
create mode 100644 labs/lab02/report/Лабораторная_работа_2.doc
[ankomyagin@localhost arch-pc]$ git push
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (13/13), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (9/9), 5.21 МБ | 2.41 МБ/с, готово.
Всего 9 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:ANKomyagin/study_2023-2024_arch-pc.git
72acc49..5e1ad2c master -> master
[ankomyagin@localhost arch-pc]$
```

Рис. 3.3: Загрузка файлов на GitHub)

Проверим файлы на сервере (рис. 3.4).

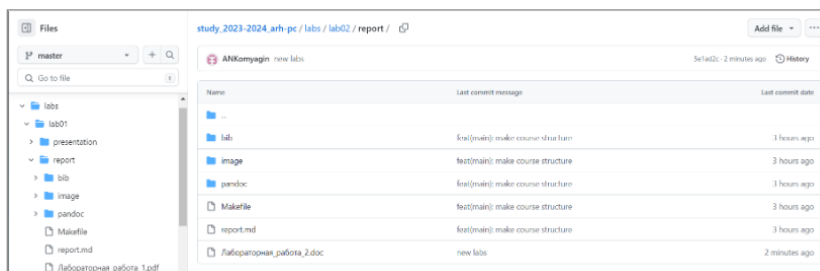


Рис. 3.4: Обновленный репозиторий GitHub

4 Вывод

В ходе работы я узнал о системе обновления версий, научился взаимодействовать с GitHub (изменять, удалять и добавлять новые файлы) при помощи командной строки Linux. Узнал о безопасном способе соединения с сервером, при помощи SSH ключей