

Лабораторная работа № 2

Операционные системы

Казазаев Д. М.

Российский университет дружбы народов, Москва, Россия

Информация

- Казазаев Даниил Михайлович
- Студент бакалавриата
- Российский университет дружбы народов
- [1132231427@rudn.ru]
- https://github.com/KazazaevDaniil/study_2022-2023_os-intro


Вводная часть

Освоить умения по работе с git, изучить идеологию и применение средств контроля версий.

1. Установка необходимого ПО.
2. Базовая настройка Git.
3. Создание ssh-ключа и его добавление на GitHub.
4. Настройка gh.
5. Создание и настройка рабочего пространства.

Основания часть

После запуска виртуальной машины, открываю терминал и пишу команду `sudo -i`, чтобы получить права суперпользователя. (рис. 1)

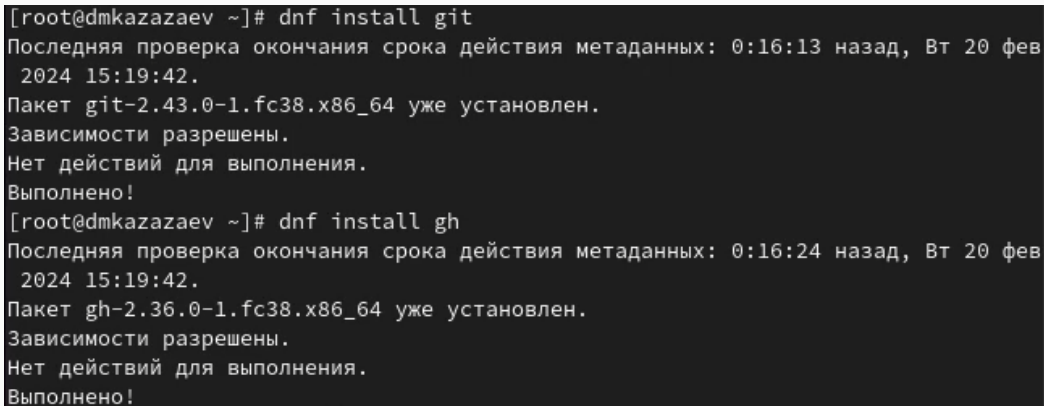


```
[dmkazazae@dmkazazae ~]$ sudo -i  
[sudo] пароль для dmkazazae:  
[root@dmkazazae ~]#
```

Рис. 1: Получение прав суперпользователя

Выполнение лабораторной работы. Установка необходимого ПО.

Устанавливаю git и gh командами `dnf install git` и `dnf install gh`. (рис. 2)



```
[root@dmkazazaev ~]# dnf install git
Последняя проверка окончания срока действия метаданных: 0:16:13 назад, Вт 20 фев 2024 15:19:42.
Пакет git-2.43.0-1.fc38.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!

[root@dmkazazaev ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:16:24 назад, Вт 20 фев 2024 15:19:42.
Пакет gh-2.36.0-1.fc38.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

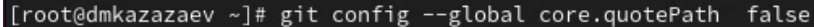
Рис. 2: Установка git и gh

Задаю имя и почту владельца репозитория. (рис. 3)

```
[root@dmkazazaev ~]# git config --global user.name "Daniil Kazazaev"  
[root@dmkazazaev ~]# git config --global user.email "kazazaevdaniil0@gmail.com"
```

Рис. 3: Установка имени и почты владельца репозитория


Настроиваю utf-8 в выводе сообщений git. (рис. 4)

A terminal window with a dark background and light green text. The prompt is [root@dmkazazaev ~]# and the command entered is git config --global core.quotePath false.

```
[root@dmkazazaev ~]# git config --global core.quotePath false
```

Рис. 4: Настройка utf-8

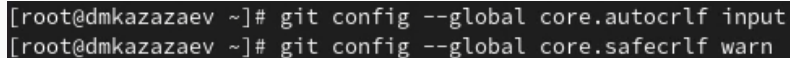
Задаю имя начальной ветки. (рис. 5)

A terminal window with a dark background. The prompt is [root@dmkazazaev ~]#. The command entered is git config --global init.defaultBranch master. The command is highlighted in white text on a dark background.

```
[root@dmkazazaev ~]# git config --global init.defaultBranch master
```

Рис. 5: Установка имени начальной ветки

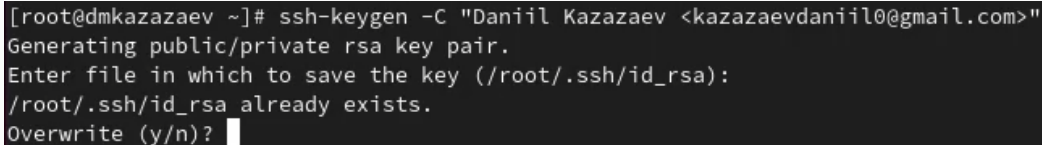
Задаю параметры autocrlf и safecrlf. (рис. 6)

A terminal window with a dark background and light gray text. It shows two lines of commands being executed. The first line sets the global autocrlf configuration to 'input'. The second line sets the global safecrlf configuration to 'warn'.

```
[root@dmkazazaev ~]# git config --global core.autocrlf input  
[root@dmkazazaev ~]# git config --global core.safecrlf warn
```

Рис. 6: Установка параметров

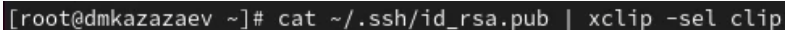
Генерирую ssh-ключ. (рис. 7)

A terminal window with a dark background and light gray text. The prompt is [root@dmkazazaev ~]#. The command executed is ssh-keygen -C "Daniil Kazazaev <kazazaevdaniil0@gmail.com>". The output shows the generation of a public/private RSA key pair, the file path /root/.ssh/id_rsa, and a confirmation that the file already exists. The prompt for overwriting is shown with a white cursor.

```
[root@dmkazazaev ~]# ssh-keygen -C "Daniil Kazazaev <kazazaevdaniil0@gmail.com>"  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
/root/.ssh/id_rsa already exists.  
Overwrite (y/n)?
```

Рис. 7: Генерация ssh-ключа

Копируем ssh-ключ командой `cat ~/.ssh/id_rsa.pub | xclip -sel clip`. (рис. 8)

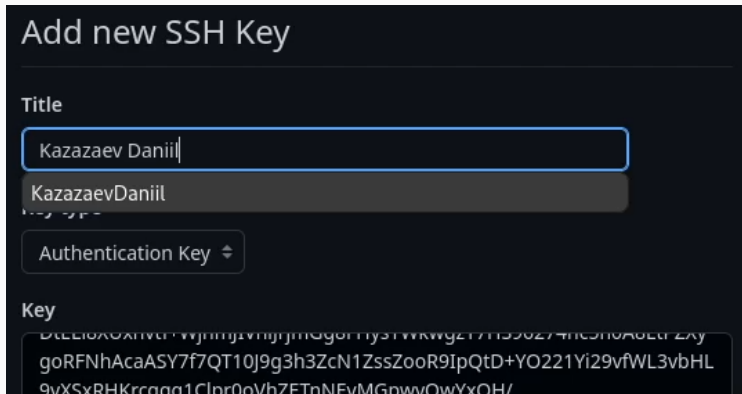
A terminal window with a dark background. The prompt is [root@dmkazazaev ~]#. The command entered is cat ~/.ssh/id_rsa.pub | xclip -sel clip. The command is highlighted in white text on the dark background.

```
[root@dmkazazaev ~]# cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 8: Копирование ssh-ключа

Выполнение лабораторной работы. Создание ssh-ключа и его добавление на GitHub.

Перехожу на GitHub и в настройках добавляю новый ssh-ключ. (рис. 9)



Add new SSH Key

Title

Kazazaev Daniil

KazazaevDaniil

Authentication Key +

Key

DCELE0X0XivCpVwjnmgjVnigjmgDg0fTysTvrWgZf7f1330274nc3noA02fZXy
goRFNhAcaASY7f7QT10J9g3h3ZcN1ZssZooR9IpQtD+YO221Yi29vfWL3vbHL
9vXSxRHKrcqaa1Clpr0oVh7FTnNEvMGpwwOwYxOH/

Рис. 9: Добавление ssh-ключа

Выполнение лабораторной работы. Настройка gh.

Для настройки gh необходимо авторизоваться. После ввода команды отвечаю на несколько наводящих вопросов и захожу под свой аккаунт GitHub. (рис. 10)

```
[root@dmkazazaev ~]# gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? [Use arrows to move, type to filter]
> /root/.ssh/id_rsa.pub
Skip
```

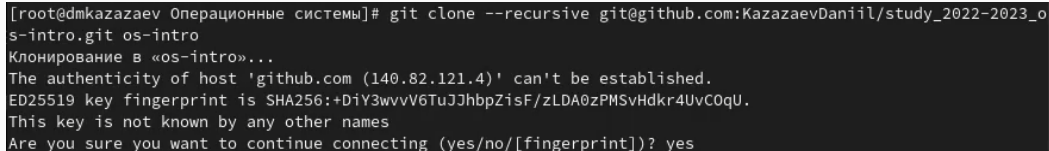
Рис. 10: Настройка gh

Создаю директорию, после чего перехожу в нее и создаю репозиторий. (рис. 11)

```
[root@dmkazazaev ~]# mkdir -p ~/work/study/2022-2023/"Операционные системы"  
[root@dmkazazaev ~]# cd ~/work/study/2022-2023/"Операционные системы"  
[root@dmkazazaev Операционные системы]# gh repo create study_2022-2023_os-intro --template=yamadharma/course-  
directory-student-template --public  
✓ Created repository KazazaevDaniil/study_2022-2023_os-intro on GitHub
```

Рис. 11: Создание директории и репозитория

Клонирую репозиторий в только что созданную директорию. (рис. 12)



```
[root@dmkazazaev Операционные системы]# git clone --recursive git@github.com:KazazaevDaniil/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```


Рис. 12: Клонирование репозитория

Перехожу в сам репозиторий и удаляю лишние файлы. (рис. 13)

```
[root@dmkazazaev Операционные системы]# cd os-intro/  
[root@dmkazazaev os-intro]# rm package.json
```

Рис. 13: Переход в репозиторий и удаление лишние файлов

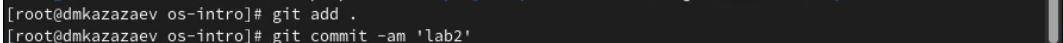
После удаления файлов создаю необходимые каталоги командой make. (рис. 14)

A terminal window with a dark background and light-colored text. The prompt is [root@dmkazazaev os-intro]#. The first command is echo os-intro > COURSE. The second command is make.

```
[root@dmkazazaev os-intro]# echo os-intro > COURSE  
[root@dmkazazaev os-intro]# make
```

Рис. 14: Создание необходимых каталогов

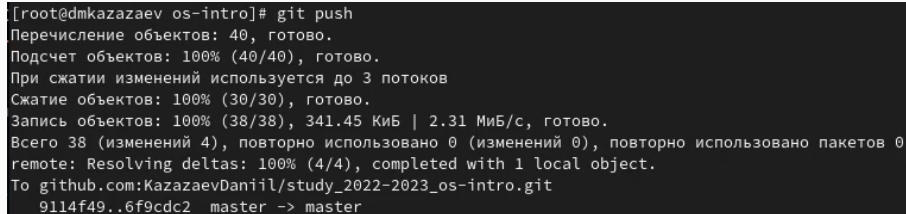
Добавляю файлы на сервер и подписываю их командой `git commit`. (рис. 15)

A terminal window with a dark background and light-colored text. It shows two lines of commands being executed in a shell. The first line is '[root@dmkazazaev os-intro]# git add .' and the second line is '[root@dmkazazaev os-intro]# git commit -am 'lab2''.

```
[root@dmkazazaev os-intro]# git add .  
[root@dmkazazaev os-intro]# git commit -am 'lab2'
```

Рис. 15: Добавление файлов с их подписью

Отправляю файлы на сервер командой `git push`. (рис. 16)

A terminal window with a dark background and light-colored text. The text shows the execution of 'git push' and its progress: counting objects, compressing them, and pushing them to a remote repository. The final line shows the commit hash and the branch name 'master' being pushed to 'master'.

```
[root@dmkazazaev os-intro]# git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 341.45 КиБ | 2.31 МиБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:KazazaevDaniil/study_2022-2023_os-intro.git
  9114f49..6f9cdc2  master -> master
```

Рис. 16: Отправка файлов на сервер

- Успешная установка всего необходимого ПО.
- Базовые настройки Git успешно установлены.
- Создан ssh-ключ и добавлен на аккаунт GitHub.
- Успешно создано и настроено рабочее пространство.

В результате выполнения данной лабораторной работы я освоил умения по работе с git, изучил идеологию и применение средств контроля версий.