

# **Лабораторная работа №2.**

**Система контроля версий Git**

Сорокин Кирилл Васильевич

## **Содержание**

1 Цель работы 4

2 Задание 5

3 Теоретическое введение 6

4 Выполнение лабораторной работы 7-11

5 Выводы 12

6 Список литературы 13

## **Список иллюстраций**

4.1. - Профиль Github	- 7
4.2. - Команды для настройки git	- 7
4.3. - Генерация ключей	- 8
4.4. - Копирование ключа	- 8
4.5. - Ключ на Github	- 9
4.6. - Создания папок	- 9
4.7. - Копирование репозитория	- 10
4.8. – Удаление	- 10
4.9. - Добавление изменений	- 10
4.10. - Выгрузка изменений	- 11

## **1 Цель работы**

Понять суть средств контроля версии и с полученными знаниями приобрести навыки работы с системой git.

## **2 Задание**

Ознакомиться с идеей контроля версии и научиться работать с git.

### **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.

## 4 Выполнение лабораторной работы

Создаём аккаунт на сайте <https://github.com/> (рис.4.1.)



Рис.4.1 – Профиль Github

Сделаем предварительную конфигурацию git, настроим UTF-8, зададим начальное имя ветки параметры autocrlf и safecrlf (рис.4.2)

```
kvsorokin@dk6n52 ~ $ git config --global user.name "Sorokin Kirill"
kvsorokin@dk6n52 ~ $ git config --global user.email "1132236060@pfur.ru"
kvsorokin@dk6n52 ~ $ git config --global core.quotepath false
kvsorokin@dk6n52 ~ $ git config --global init.defaultBranch master
kvsorokin@dk6n52 ~ $ git config --global core.autocrlf input
kvsorokin@dk6n52 ~ $ git config --global core.safecrlf warn
kvsorokin@dk6n52 ~ $
```

Рис.4.2 – Команды для настройки git

Для работы с репозиторием сгенерируем SSH ключ (рис.4.3.)

```
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера $ ssh-keygen -C "Sorokin Kirill 1132236060@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:CDKWqHJzHUF8aRVLkYnqnHfkdwxvSP8YP5lDI+Z0pdc Sorokin Kirill 1132236060@pfur.ru
The key's randomart image is:
+---[RSA 3072]-----+
|      oo  +==      |
|  . .  ..=.o.      |
| . = . .+ .        |
|.. o o.o . o .    |
|o o .oo.So . * .o|
|.. o  + . o o+0+E|
|      . . .+o+*+|
|      ..=o|
|      o|
+-----[SHA256]-----+
```

Рис.4.3. – Генерация ключей

Скопируем ключ командой cat (рис.4.4.)

```
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера $ cat ~/.ssh/id_rsa.pub | xclip -se
l clip
```

Рис.4.4. – Копирование ключа




Введём SSH ключ в Github (рис.4.5.)

## 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



DK

SHA256:CDKwqHJzHUF8aRVLkYnqnHfkdwxvSP8YP51DI+Z0pdc

Added on Sep 27, 2023

Never used — Read/write

SSH

Delete

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

Рис.4.5. – Ключ на Github

Создадим папки для копирования репозитория (рис.4.6.)

```
kvsorokin@dk6n52 ~ $ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"  
kvsorokin@dk6n52 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис.4.6. – Создания папок

Скопируем репозиторий командой clone (рис.4.7.)

```
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:sorokinkirill1132236060/-study_2023-2024_arh--pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.94 КиБ | 826.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 1.16 МиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report»...
```

Рис.4.7. – Копирование репозитория

Удалим ненужный файл командой rm(рис.4.8.)

```
kvsorokin@dk6n51 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ ls
CHANGELOG.md  COURSE  Makefile  README.git-flow.md  template
config        LICENSE  README.en.md  README.md
```

Рис.4.8. – Удаление

Добавим изменения в commit (рис.4.9.)

```
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add .
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am 'feat(main): make course structure'
[master 7bfb4b0] 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
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
```

Рис.4.9. – Добавление изменений

Выгрузим все изменения в наш проект командой push(рис.4.10.)

```
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.63 МиБ/с, готово.
Всего 35 (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:sorokinkirill1132236060/-study_2023-2024_arh--pc.git
   bb50821..7bfb4b0  master -> master
kvsorokin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $
```

Рис.4.10. – Выгрузка изменений

В качестве самостоятельного задания выгрузим отчёты лабораторных работ в папки нашего проекта.

## **5 Вывод**

В ходе лабораторной работы мы изучили идеологию и применение средств контроля версий и научились работать с git.

## 6 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, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. 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).
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).