

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: Нечипоренко Дмитрий Олегович

Группа: НКАбд-03-24

МОСКВА

2024 г.

Содержание

1. Цель работы.....	3
2. Выполнение лабораторной работы	3
3. Выполнение самостоятельной работы.....	9
4. Выводы.....	10

1. Цель работы

Получить навыки работы с хранилищем репозитория, а также базовые навыки работы с git.

2. Выполнение лабораторной работы

Создадим аккаунт в сервисе - хранилище репозитория GitHub.

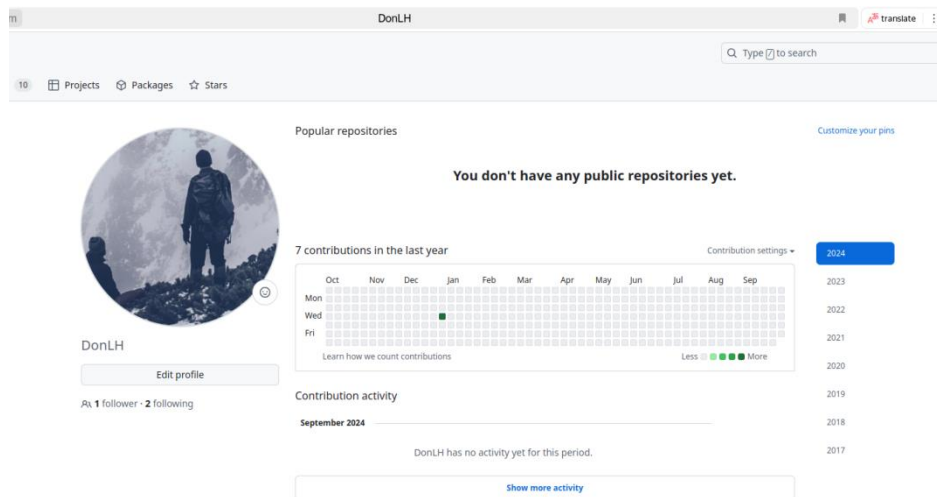


Рис. 2.1 Страница профиля GitHub

Настроим имя и email пользователя, ветку по умолчанию, а также некоторые настройки правильного отображения символов.

```
donechiporenko@donechiporenko:~$ git config --global user.name "Dmitry Nechiporenko"
donechiporenko@donechiporenko:~$ git config --global user.email "donlh@yandex.ru"
donechiporenko@donechiporenko:~$ git config --global core.quotepath false
donechiporenko@donechiporenko:~$ git config --global init.defaultBranch master
donechiporenko@donechiporenko:~$ git config --global core.autocrlf input
donechiporenko@donechiporenko:~$ git config --global core.safecrlf warn
donechiporenko@donechiporenko:~$
```

Рис. 2.2 Настройка git

Сгенерируем ключ для подключения к GitHub при помощи асимметричного шифрования.

```
donechiporenko@donechiporenko:~$ ssh-keygen -C "Dmitry Nechiporenko donlh@yandex.ru"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/donechiporenko/.ssh/id_ed25519):
Created directory '/home/donechiporenko/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/donechiporenko/.ssh/id_ed25519
Your public key has been saved in /home/donechiporenko/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:IyG9WOnRWFhKhRseLSolG5/rOR7U/nJfjDJy6CQKe68 Dmitry Nechiporenko donlh@yandex.ru
The key's randomart image is:
+--[ED25519 256]--+
|      .*+      |
|    +o**      |
|   ..BO*      |
|  . o===      |
|   .. =.S      |
|    ..o o  o   |
|   . .o.= + . o |
|  o..+=..+.o . |
|   ..E+o..o... |
+-----[SHA256]-----+
donechiporenko@donechiporenko:~$
```

Рис. 2.3 Генерация ключей

В директории `.ssh` автоматически сгенерировались два ключа: один открытый - его необходимо загрузить на GitHub, и один закрытый. Скопируем открытый ключ с помощью `xclip`.

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

Рис. 2.4 Копирование открытого ключа

Добавляем скопированный ключ в соответствующий раздел настроек GitHub (SSH and GPG keys). Теперь мы можем обновлять содержимое личных репозиториев.

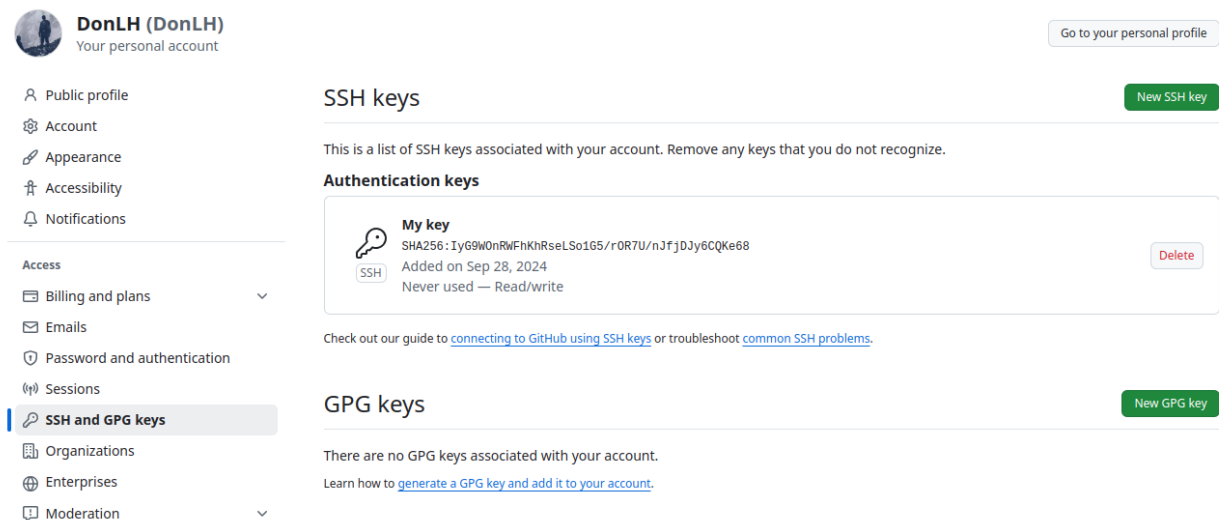


Рис. 2.5 Добавления ключа для доступа по SSH

Создадим рабочее пространство в соответствии с указаниями в лабораторной работе.

```
donechiporenko@donechiporenko:~$ mkdir -p work/study/2024-2025/"Архитектура компьютера"
donechiporenko@donechiporenko:~$
```

Рис. 2.6 Создание структуры рабочего пространства

Используем репозиторий `course-directory-student-template` в качестве шаблона (Кнопка Use this template).

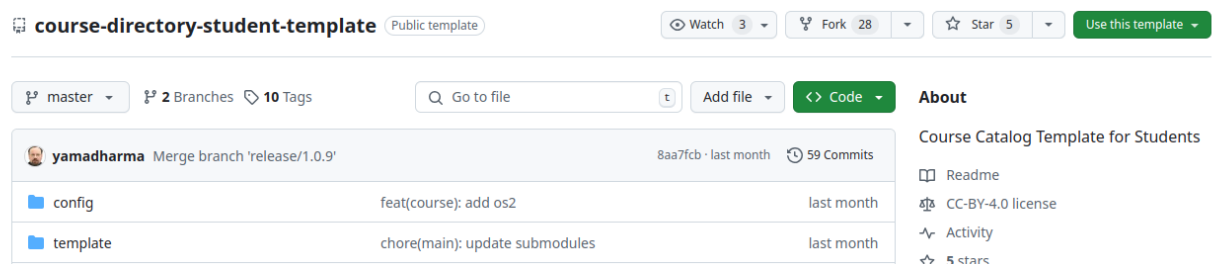


Рис. 2.7 Шаблон репозитория


При создании репозитория указываем название study_2024-2025_arch-pc.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk ().*

Repository template


 yamadharm/course-directory-student-template ▾

Start your repository with a template repository's contents.

☐ **Include all branches**

Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

Owner *

 DonLH ▾

Repository name *

/ study_2024-2025_arch-pc

✔ study_2024-2025_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about **jubilant-computing-machine** ?

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. 2.8 Создание копии репозитория

Скопируем ссылку на только что созданный репозиторий для доступа по SSH.

 **Clone** 

HTTPS **SSH** **GitHub CLI**



Use a password-protected SSH key.


 **Download ZIP**

Рис. 2.8 Ссылка на репозиторий

Загружаем репозиторий при помощи команды `git clone` и называем его `arch-pc`.

```
donechiporenko@donechiporenko:~$ cd work/study/2024-2025/"Архитектура компьютера"
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:DonLH/study_2024-2025_arch-pc.git arch-pc
```

Рис. 2.9 Клонирование репозитория

Выполним несколько изменений внутри репозитория. Команда `make prepare` использует инструкции в `Makefile`, чтобы создать первичную структуру рабочего пространства в соответствии с заданием.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера$ cd arch-pc
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make
Usage:
  make <target>

Targets:
  list              List of courses
  prepare           Generate directories structure
  submodule         Update submodules

donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 2.10 Изменения в репозитории

Выполнив команду `ls` убеждаемся, что команда `make prepare` отработала правильно.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
config        labs   Makefile  presentation  README.git-flow.md  template
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ ls labs
lab01 lab02 lab03 lab04 lab05 lab06 lab07 lab08 lab09 lab10 lab11 README.md README.ru.md
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 2.11 Проверка изменений

Добавляем внесённые в репозиторий изменения для коммита. На скриншоте показана часть вывода команды `git status`.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add *
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   COURSE
    new file:   labs/README.md
    new file:   labs/README.ru.md
    new file:   labs/lab01/presentation/.projectile
    new file:   labs/lab01/presentation/.texlabroot
    new file:   labs/lab01/presentation/Makefile
```

Рис. 2.12 Добавление изменений

Так как гит автоматически добавляет все изменения кроме тех, которые связаны с удалением файлов, выполняем специальную команду (`git rm`), чтобы добавить соответствующее изменение. Теперь можно создать коммит с внесёнными изменениями командой `git commit`.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git rm package.json
rm 'package.json'
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -m "Prepare structure"
[master 90f1a63] Prepare structure
223 files changed, 53681 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
```

Рис. 2.13 Удаление и коммит

Загружаем этот коммит в удалённый репозиторий. При этом авторизация происходит автоматически благодаря настроенной аутентификации `.ssh`.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push -u origin master
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 2 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 341.27 KiB | 2.35 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:DonLH/study_2024-2025_arch-pc.git
 91ff73d..90f1a63 master -> master
branch 'master' set up to track 'origin/master'.
```

Рис. 2.14 Загрузка на удалённые репозиторий

Убеждаемся, что все файлы действительно загружены на удалённый репозиторий.

The screenshot shows a GitHub repository page for 'study_2024-2025_arch-pc'. The repository is public and was generated from 'yamadharm/course-directory-student-template'. It has 1 branch (master) and 0 tags. The 'About' section indicates no description, website, or topics are provided. The 'Releases' section shows no releases published. The 'Files' section lists the following files and their commit history:

File	Commit	Time
config	Initial commit	11 minutes ago
labs	Prepare structure	2 minutes ago
presentation	Prepare structure	2 minutes ago
template	Initial commit	11 minutes ago
.gitattributes	Initial commit	11 minutes ago
.gitignore	Initial commit	11 minutes ago
.gitmodules	Initial commit	11 minutes ago

Рис. 2.15 Проверка загрузки на удалённый репозиторий

3. Выполнение самостоятельной работы

3.1 Задание 1

Создадим отчёт к данной лабораторной работе.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cp ~/Л02_Нечипоренко_отчёт.pdf labs/lab02/report
```

Рис. 3.1 Создание отчёта

3.2 Задание 2

Скопируем отчёты к предыдущим лабораторным работам.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cp ~/Л01_Нечипоренко_отчёт.pdf labs/lab01/report
```

Рис. 3.2 Копирование отчётов предыдущих лабораторных работ

3.3 Задание 3

Добавим изменения в репозитории для коммита и загрузим его на удалённый репозиторий.

```
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add *
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   labs/lab01/report/Л01_Нечипоренко_отчёт.pdf
        new file:   labs/lab02/report/Л02_Нечипоренко_отчёт.pdf

donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -m "Add lab01, lab02 reports"
[master c30a4ae] Add lab01, lab02 reports
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Нечипоренко_отчёт.pdf
create mode 100644 labs/lab02/report/Л02_Нечипоренко_отчёт.pdf
donechiporenko@donechiporenko:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push -u origin master
Enumerating objects: 14, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 861.24 KiB | 6.29 MiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:DonLH/study_2024-2025_arch-pc.git
   90f1a63..c30a4ae  master -> master
branch 'master' set up to track 'origin/master'.
```

Рис. 3.3 Добавление изменений в репозиторий GitHub

4. Выводы

В результате выполнения лабораторной работы были изучены принципы работы с удалёнными репозиториями git, авторизации с применением асимметричного шифрования, а также базовые команды и особенности git.

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/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 9781491941591.
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. — БХВ-Петербург, 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.
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).