

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: Пономарева В.А.

Группа: НПИбд-02-25

Студ. билет: №1032253634

МОСКВА

2025 г.

1 Цель работы

Изучение идеологии и применения средств контроля версий, приобретение практических навыков по работе с системой контроля версий git.

2 Порядок выполнения работы

1. Базовая настройка git.

Сначала делаем предварительную конфигурацию git. На [рисунке 1.1](#) показано то, как задается имя и почта пользователя репозитория.

```
vaponomareva@localhost-live:~$ git config --global user.name "PonomarevaVA"  
vaponomareva@localhost-live:~$ git config --global user.mail "1032253635@pfur.ru"
```

Рисунок 1.1 - Задаем имя и email репозитория

Настроим utf-8 в выводе сообщений git. На [рисунке 1.2](#) изображен процесс настройки utf-8.

```
vaponomareva@localhost-live:~$ git config --global core.quotepath false
```

Рисунок 1.2 - Используем нужную команду для настройки utf-8

Зададим имя начальной ветки (будем называть её master). На [рисунке 1.3](#) изображен процесс задания имени для начальной ветки.

```
vaponomareva@localhost-live:~$ git config --global init.defaultBranch master  
vaponomareva@localhost-live:~$
```

Рисунок 1.3 - Начальная ветка master

Параметр autocrlf определяет, как Git будет обрабатывать концы строк при добавлении файлов в индекс и при выгрузке файлов из индекса. Значение input означает, что Git будет преобразовывать концы строк в формат LF (Unix-style) при добавлении файлов в индекс, но не будет выполнять никаких преобразований при выгрузке файлов из индекса. На [рисунке 1.4](#) показана настройка параметра autocrlf.

```
vaponomareva@localhost-live:~$ git config --global core.autocrlf input  
vaponomareva@localhost-live:~$
```

Рисунок 1.4 - Настройка autocrlf

Настройка core.safecrlf warn определяет, как Git будет реагировать на различные форматы концов строк в файлах. Значение warn означает, что Git будет выдавать предупреждение, если в файле обнаружены смешанные концы строк. На [рисунке 1.5](#) изображена настройка параметра safecrlf.

```
vaponomareva@localhost-live:~$ git config --global core.safecrlf warn  
vaponomareva@localhost-live:~$
```

Рисунок 1.5 - Устанавливаем параметр safecrlf

2. Создание SSH ключа.

Для последующей идентификации пользователя на сервере репозитория

сгенерируем пару ключей (приватный и открытый). На [рисунке 2.1](#) изображен процесс создания ключа с помощью команды `ssh-keygen`. На [рисунке 2.2](#) показано копирование сгенерированного ключа в буфер обмена с помощью команды `xclip` (согласно источнику информации Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science)).

```
vaponomareva@localhost-live:~$ ssh-keygen -C "PonomarevaVA <1032253635@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/vaponomareva/.ssh/id_ed25519):
Created directory '/home/vaponomareva/.ssh'.
Enter passphrase for "/home/vaponomareva/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vaponomareva/.ssh/id_ed25519
Your public key has been saved in /home/vaponomareva/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:r95biYyLfC66W26pbLX0Vu4hJyskf7FUp6jwrqUC4IY PonomarevaVA <1032253635@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|      . . .      |
|      o. o      |
|      . . o.     |
|      . o .o     |
|+   o   .S      |
|Eo .. = +.. .   |
|o   o*.++o* o   |
|.   =.++.+0.o   |
| .o.**+B*.+.    |
+----[SHA256]-----+
vaponomareva@localhost-live:~$
```

Рисунок 2.1 - Генерация SSH ключа

```
Завершено!
vaponomareva@localhost-live:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
vaponomareva@localhost-live:~$
```

Рисунок 2.2 - Копирование ключа в буфер обмена

Заходим в аккаунт на github, переходим в настройки и добавляем новый SSH-ключ. На [рисунке 2.3](#) изображена страница с созданным ключом.

Add new SSH Key

Title

Key type

Authentication Key ▾

Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIoGKf1hBmagsImWBILga/aQ6jIQ5srxMDqure8ffgMe Varvara Ponomareva
<1032253635@pfur.ru>
```

Add SSH key

Рисунок 2.3 - SSH-ключ Title

Проверим наличие созданного ключа в профиле. На рисунке 2.4 показан созданный ключ.

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



Title

SHA256:ua9eMn7+ZNYy0XnMjbqFvnaDRXE1ffrTWb/hfUk/cFg

Added on Sep 21, 2025

Never used — Read/write

Delete

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

Рисунок 2.4 - Проверяем наличие ключа

3. Создание рабочего пространства и репозитория курса на основе шаблона.

Создадим каталог для предмета «Архитектура компьютера» с нужным названием. На [рисунке 3.1](#) показан процесс создания каталога.

```
varonomareva@localhost-live:~$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
varonomareva@localhost-live:~$
```

Рисунок 3.1 - Создаем каталог "Архитектура компьютера"

4. Создание репозитория курса на основе шаблона.

Переходим на страницу репозитория с шаблоном курса и с помощью use this template создаем репозиторий. На [рисунке 4.1](#) показана страница созданного репозитория.

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

Start with a template

Templates pre-configure your repository with files.

yamadharma/course-directory-student-template

Include all branches

If enabled, all branches from the template repository will be included.

Off ☐

1 General

Owner *

PonomarevaVA0

Repository name *

study_2025-2026_arh-pc

study_2025-2026_arh-pc is available.

Great repository names are short and memorable. How about cautious-octo-system?

Description

0 / 350 characters

2 Configuration

Choose visibility *

Private

Create repository

Рисунок 4.1 - Создаем репозиторий с названием study_2025–2026_arh-pc

Далее переходим в каталог курса и клонируем созданный репозиторий с помощью команды git clone (согласно источнику информации 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>). На [рисунке 4.2](#) изображено использование команды `cd` для перехода в нужный каталог. На [рисунке 4.3](#) показан процесс клонирования репозитория.

```
vaponomareva@localhost-live:~$ cd ~/work/study/2025-2026/"Архитектура компьютера"
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера$
```

Рисунок 4.2 - Переходим в каталог

```
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера$ git clone --recursive g
it@github.com:PonomarevaVA0/study_2025-2026_arh-pc.git
Cloning into 'study_2025-2026_arh-pc'...
Enter passphrase for key '/home/vaponomareva/.ssh/id_ed25519':
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 38 (delta 1), reused 27 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (38/38), 23.45 KiB | 7.82 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-te
mplate.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git
) registered for path 'template/report'
Cloning into '/home/vaponomareva/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-p
c/template/presentation'...
remote: Enumerating objects: 161, done.
remote: Counting objects: 100% (161/161), done.
remote: Compressing objects: 100% (111/111), done.
remote: Total 161 (delta 60), reused 142 (delta 41), pack-reused 0 (from 0)
Receiving objects: 100% (161/161), 2.65 MiB | 3.14 MiB/s, done.
```

Рисунок 4.3 - Клонировем репозиторий

5. Настройка каталога курса.

Перейдем в каталог курса и создадим там каталоги с помощью команд `echo arch-pc > COURSE` и `make prepare`. [Рисунок 5.1](#) иллюстрирует переход в каталог курса. На [рисунке 5.2](#) показан процесс создания каталогов

```
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера$ cd ~/work/study/2025-20
26/"Архитектура компьютера"/arch-pc
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc$
```

Рисунок 5.1 - Каталог курса

```
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ make prepare
```

Рисунок 5.2 - Создаем нужные каталоги

Добавляем все измененные файлы с помощью команды `git add` для последующего коммита и записываем изменения в репозиторий. [Рисунок 5.3](#) иллюстрирует добавление измененных файлов.

```
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit -am
"feat(main): make course structure"
[master 5fb738b] feat(main): make course structure
212 files changed, 8074 insertions(+), 207 deletions(-)
delete mode 100644 CHANGELOG.md
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.gitignore
create mode 100644 labs/lab01/presentation/.marksman.toml
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/_quarto.yml
create mode 100644 labs/lab01/presentation/_resources/image/logo_rudn.png
create mode 100644 labs/lab01/presentation/arch-pc--lab01--presentation.qmd
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/report/.gitignore
create mode 100644 labs/lab01/report/.marksman.toml
create mode 100644 labs/lab01/report/.projectile
create mode 100644 labs/lab01/report/Makefile
```

Рисунок 5.3 - Добавляем измененные файлы

Отправляем данные в репозиторий с помощью команды `git push`. [Рисунок 5.4](#) показывает отправку данных с помощью команды `git push` (согласно источнику информации GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>).

```
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Enter passphrase for key '/home/vaponomareva/.ssh/id_ed25519':
Enumerating objects: 67, done.
Counting objects: 100% (67/67), done.
Delta compression using up to 2 threads
Compressing objects: 100% (52/52), done.
Writing objects: 100% (64/64), 700.61 KiB | 4.25 MiB/s, done.
Total 64 (delta 22), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (22/22), completed with 1 local object.
To github.com:PonomarevaVA0/study_2025-2026_arh-pc.git
5ca949e..5fb738b master -> master
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc$
```

Рисунок 5.4 - Отправляем данные

Проверим выполнение команд с помощью команды `ls` и посмотрим файлы. [Рисунок 5.5](#) показывает проверку выполнения команды с помощью `ls`.

```
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs$ ls
lab01 lab03 lab05 lab07 lab09 lab11 README.ru.md
lab02 lab04 lab06 lab08 lab10 README.md
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs$
```

Рисунок 5.5 - Проверка выполнения

Самостоятельная работа.

1. Создадим отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (`labs/lab02/report`) с помощью `touch`. Проверим, что документ с отчетом находится в папке (с помощью команды `ls`). На [рисунке 6.1](#) изображено применение команды `ls` и проверка наличия файла.

```
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab02/report$ touch Л02_Пономарева_отчет.docx
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab02/report$ ls
arch-pc--lab02--report.qmd bib image Makefile _quarto.yml _resources Л02_Пономарева_отчет.docx
vaponomareva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рисунок 6.1 - Проверяем, что файл в нужном каталоге.

2. Скопируем отчет по выполнению предыдущей лабораторной работы в соответствующий каталог созданного рабочего пространства. Используем команду `mv`, чтобы переместить файл и команду `ls`, чтобы проверить его наличие в нужном

каталоге. На [рисунке 6.2](#) изображено применение команды mv и проверка наличия файла.

```
varonomaeva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab01/report$ sudo mv /home/liveuser/Downloads/Л01_Пономарева_отчет.pdf .
[sudo] password for varonomaeva:
varonomaeva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
arch-pc--lab01--report.qmd  hib  image  Makefile  _quarto.yml  resources  Л01_Пономарева_отчет.pdf
```

Рисунок 6.2 - Перемещаем файл и проверяем его

3. Отправим файлы на github с помощью команды git add и добавим описание выполненного действия с помощью команды commit -m, а также подтвердим все с помощью git push. На [рисунке 6.3](#) показано добавление файлов и сообщения к ним, а на [рисунке 6.4](#) синхронизация с github.

```
varonomaeva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab01/report$ git add Л01_Пономарева_отчет.pdf
varonomaeva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab01/report$ git commit -m "Добавлен отчет по лабораторной работе 1"
[master 726cb01] Добавлен отчет по лабораторной работе 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Пономарева_отчет.pdf
```

Рисунок 6.3 - Добавляем нужные файлы на github

```
varonomaeva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab01/report$ git push
Enter passphrase for key '/home/varonomaeva/.ssh/id_ed25519':
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 136.61 KiB | 1.28 MiB/s, done.
Total 8 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To github.com:PonomarevaVA0/study_2025-2026_arh-pc.git
 feafe0b..726cb01 master -> master
varonomaeva@localhost-live:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рисунок 6.4 - Отправляем файлы

Вывод:

Таким образом, мы приобрели практические навыки работы с операционной системой Linux на уровне командной строки, а именно организации файловой системы, навигации по файловой системе, создании и удалении файлов и директорий.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>
2. 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>
3. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).