

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: Бессонов Андрей Максимович

Группа: НКАбд - 01 - 25

МОСКВА

2025 г.

Содержание

1. Цель работы	4
2. Теоретическое введение	5
3. Выполнение лабораторной работы	7
4. Выполнение самостоятельной работы	13
5. Выводы	14

Список иллюстраций

3.1 Terminal1	7
3.2.1 Terminal21	8
3.2.2 Terminal22	8
3.2.3 Terminal23	8
3.3.1 Terminal31	9
3.3.2 Terminal32	10
3.4.1 Terminal41	11
3.4.2 Terminal42	12
4.1 swork1	13
4.2 swork2	13

1. Цель работы

Целью работы является изучение идеологии и применения средств контроля версий, приобретение практических навыков по работе с системой контроля версий git.

2. Теоретическое введение

2.1 Системы контроля версий. Общие понятия

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.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом.

Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая, таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди

распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

2.2 Система контроля версий Git

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.

Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией

3. Выполнение лабораторной работы

3.1 Создаем учётную запись на сайте <https://github.com/> и заполняем основные данные.

Сначала сделаем предварительную конфигурацию git. Откроем терминал и введем следующие команды, указав имя и e-mail владельца репозитория:

```
git config --global user.name ""
```

```
git config --global user.email ""
```

Настроим utf-8 в выводе сообщений git:

```
git config --global core.quotePath false
```

Зададим имя начальной ветки (будем называть её master):

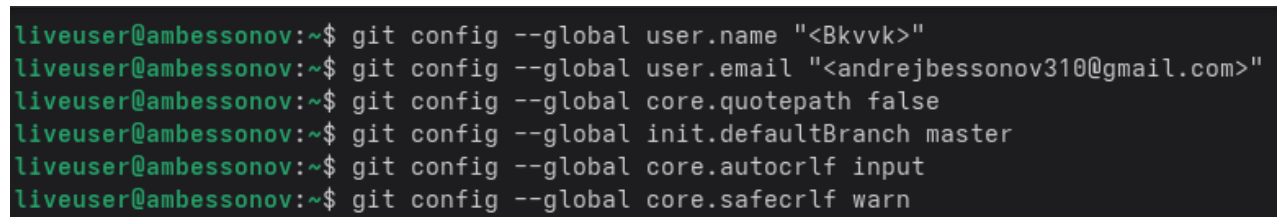
```
git config --global init.defaultBranch master
```

Параметр autocrlf:

```
git config --global core.autocrlf input
```

Параметр safecrlf:

```
git config --global core.safecrlf warn
```



```
liveuser@ambessonov:~$ git config --global user.name "<Bkvvk>"
liveuser@ambessonov:~$ git config --global user.email "<andrejbessonov310@gmail.com>"
liveuser@ambessonov:~$ git config --global core.quotePath false
liveuser@ambessonov:~$ git config --global init.defaultBranch master
liveuser@ambessonov:~$ git config --global core.autocrlf input
liveuser@ambessonov:~$ git config --global core.safecrlf warn
```

Рис 3.1: Terminal1

3.2 Сгенерируем пару ключей приватный и открытый

ssh-keygen -C "Имя Фамилия "

Далее необходимо загрузить сгенерированный открытый ключ. Для этого зайдём на сайт <http://github.org/> под своей учётной записью и перейдем в меню Setting . После этого выберем в боковом меню SSH and GPG keys и нажимаем кнопку New SSH key Копируем из локальной консоли ключ в буфер обмена

Вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title).

```
liveuser@ambessonov:~$ ssh-keygen -C "Bkvvk <andreyjbessonov310@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_ed25519):
Created directory '/home/liveuser/.ssh'.
Enter passphrase for "/home/liveuser/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_ed25519
Your public key has been saved in /home/liveuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:500gP0fAFj+aa14y+6QaKwVCXZIXvhQyW1iwwMV52Ys Bkvvk <andreyjbessonov310@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|...+B*+          |
| o.*0+..         |
| .ooo.o          |
|o  .E.+          |
|.. ...S          |
| .o + o .        |
|... B + .        |
|==.. B .         |
|OB+.o .          |
+----[SHA256]-----+
liveuser@ambessonov:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
bash: xclip: команда не найдена...
cat: /home/liveuser/.ssh/id_rsa.pub: Нет такого файла или каталога
liveuser@ambessonov:~$ cat /home/liveuser/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIN5NR0vw7ZqEbziJWi7+2CYbutMDYvPaC6TUuWbSffwt Bkvvk <andreyjbessonov310@gmail.com>
liveuser@ambessonov:~$
```

Рис 3.2.1: Terminal21

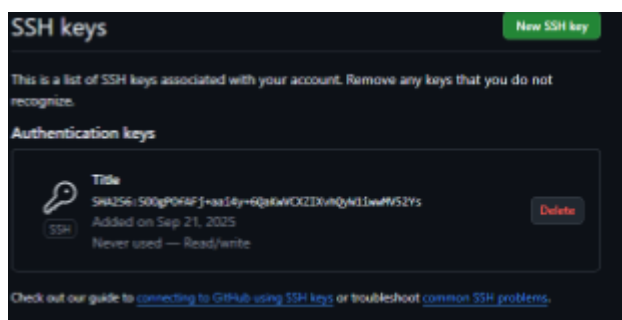


Рис 3.2.2: Terminal22

Добавьте новый SSH-ключ

Название

Тип ключа
 Authentication Key

Ключища

```
ssh-ed25519
AAAAAC3NzaC1lZDI1NTE5AAAAIN5NR0vw7ZqEbziJWi7+2CYbutMDYvPaC6TUuWbSffwt
Bkvvk <andreyjbessonov310@gmail.com>
```

Добавить SSH-ключ

Рис 3.2.3: Terminal23

3.3 Откроем терминал и создадим каталог для предмета «Архитектура компьютера». Перейдем на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Выберем Use this template и в открывшемся окне задаем имя репозитория (Repository name)

study_2025–2026_arh-pc и создаем репозиторий (кнопка Create repository from template).

Откроем терминал и перейдем в каталог курса:

```
cd ~/work/study/2025–2026/"Архитектура компьютера"
```

Далее клонируем созданный репозиторий командой:

```
git clone --recursive git@github.com:/study_2025–2026_arh-pc.git arch-pc
```

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.

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

General

Owner * **Repository name ***

andrejbessonov310-commits / study_2025–2026_arh-pd

Your new repository will be created as study_2025-2026_arh-pc.
✓ The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. How about **verbose-journey**?

Description

0 / 350 characters

Configuration

Choose visibility *
Choose who can see and commit to this repository

Public

Create repository

Рис 3.3.1: Terminal31

```

liveuser@ambessonov:~$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
liveuser@ambessonov:~$ cd ~/work/study/2025-2026/"Архитектура компьютера"
liveuser@ambessonov:~/work/study/2025-2026/Архитектура компьютера$ git clone --recursive git@github.com:an
drejbessonov310-commits/study_2025-2026_arh-pc.git arch-pc
Клонирование в «arch-pc»...
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)
Получение объектов: 100% (38/38), 23.45 КиБ | 245.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.g
it) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/liveuser/work/study/2025-2026/Архитектура компьютера/arch-pc/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)
Получение объектов: 100% (161/161), 2.65 МиБ | 578.00 КиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Клонирование в «/home/liveuser/work/study/2025-2026/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 221, done.
remote: Counting objects: 100% (221/221), done.
remote: Compressing objects: 100% (152/152), done.
remote: Total 221 (delta 98), reused 180 (delta 57), pack-reused 0 (from 0)
Получение объектов: 100% (221/221), 765.46 КиБ | 750.00 КиБ/с, готово.
Определение изменений: 100% (98/98), готово.
Submodule path 'template/presentation': checked out '6efd5c4ee78e4456caff3dc7062cfcad26058ca6'
Submodule path 'template/report': checked out '89a9622199b4df88227b9b3fa3d4714c85f68dd2'
liveuser@ambessonov:~/work/study/2025-2026/Архитектура компьютера$

```

Рис 3.3.2: Terminal32

3.4 Перейдем в каталог курса:

```
cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
```

Создадим необходимые каталоги:

```
echo arch-pc > COURSE
```

```
make prepare
```

Отправим файлы на сервер:

```
git add .
```

```
git commit -am 'feat(main): make course structure'
```

```
git push
```

Проверим правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.

```

liveuser@ambessonov: ~/work/study/2025-2026/Архитектура компьютера$ cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc
liveuser@ambessonov: ~/work/study/2025-2026/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
liveuser@ambessonov: ~/work/study/2025-2026/Архитектура компьютера/arch-pc$ make prepare
liveuser@ambessonov: ~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
liveuser@ambessonov: ~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master ed6fad0] 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
create mode 100644 labs/lab01/report/_quarto.yml
create mode 100644 labs/lab01/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab01/report/arch-pc--lab01--report.qmd
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/solvay.jpg
create mode 100644 labs/lab02/presentation/.gitignore
create mode 100644 labs/lab02/presentation/.marksman.toml
create mode 100644 labs/lab02/presentation/.projectile
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/_quarto.yml
create mode 100644 labs/lab02/presentation/_resources/image/logo_rudn.png
create mode 100644 labs/lab02/presentation/arch-pc--lab02--presentation.qmd
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/report/.gitignore
create mode 100644 labs/lab02/report/.marksman.toml
create mode 100644 labs/lab02/report/.projectile
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/_quarto.yml
create mode 100644 labs/lab02/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab02/report/arch-pc--lab02--report.qmd
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/solvay.jpg
create mode 100644 labs/lab03/presentation/.gitignore

```

Рис 3.4.1: Terminal41

```

create mode 100644 labs/lab11/presentation/_quarto.yml
create mode 100644 labs/lab11/presentation/_resources/image/logo_rudn.png
create mode 100644 labs/lab11/presentation/arch-pc--lab11--presentation.qmd
create mode 100644 labs/lab11/presentation/image/kulyabov.jpg
create mode 100644 labs/lab11/report/.gitignore
create mode 100644 labs/lab11/report/.marksman.toml
create mode 100644 labs/lab11/report/.projectile
create mode 100644 labs/lab11/report/Makefile
create mode 100644 labs/lab11/report/_quarto.yml
create mode 100644 labs/lab11/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab11/report/arch-pc--lab11--report.qmd
create mode 100644 labs/lab11/report/bib/cite.bib
create mode 100644 labs/lab11/report/image/solvay.jpg
create mode 100644 presentation/prepare
create mode 100644 presentation/README.md
create mode 100644 presentation/README.ru.md
create mode 100644 presentation/presentation/.gitignore
create mode 100644 presentation/presentation/.marksman.toml
create mode 100644 presentation/presentation/.projectile
create mode 100644 presentation/presentation/Makefile
create mode 100644 presentation/presentation/_quarto.yml
create mode 100644 presentation/presentation/_resources/image/logo_rudn.png
create mode 100644 presentation/presentation/arch-pc--presentation--presentation.qmd
create mode 100644 presentation/presentation/image/kulyabov.jpg
create mode 100644 presentation/report/.gitignore
create mode 100644 presentation/report/.marksman.toml
create mode 100644 presentation/report/.projectile
create mode 100644 presentation/report/Makefile
create mode 100644 presentation/report/_quarto.yml
create mode 100644 presentation/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 presentation/report/arch-pc--presentation--report.qmd
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/solvay.jpg
liveuser@ambessonov:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 67, готово.
Подсчет объектов: 100% (67/67), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (52/52), готово.
Запись объектов: 100% (64/64), 700.60 КиБ | 3.49 МиБ/с, готово.
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:andrejbessonov310-commits/study_2025-2026_arh-pc.git
   9052fe8..ed6fad0  master -> master
liveuser@ambessonov:~/work/study/2025-2026/Архитектура компьютера/arch-pc$

```

Рис 3.4.1: Terminal42

Вывод: выполнили задания лабораторной работы, и научились работать с новыми командами и VCS Git.

4. Выполнение самостоятельной работы

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

Решение:

```
ambessonov@fedora:~/Загрузки$ mv ~/Загрузки/Л02_БессоновАндрей_отчет.pdf ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab02/report
ambessonov@fedora:~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab02/report$ ls
arch-pc--lab02--report.qmd  bib  image  Makefile  _quarto.yml  _resources  Л02_БессоновАндрей_отчет.pdf
```

Рис 4.1: swork1

Ответ: Создали и скопировали отчет в соответствующий каталог.

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

Решение:

```
ambessonov@fedora:~$ mv ~/Загрузки/Л01_БессоновАндрей_отчет.pdf ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab01/report
ambessonov@fedora:~$ ls ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab01/report
arch-pc--lab01--report.qmd  bib  image  Makefile  _quarto.yml  _resources  Л01_БессоновАндрей_отчет.pdf
```

Рис 4.2: swork2

Ответ: Скопировали отчет в соответствующий каталог.

3. Загрузите файлы на github.

Решение: Файлы на github загрузим с помощью функций

git add .

git commit -am 'feat(main): make course structure'

git push

Вывод: выполнили задания самостоятельной работы, отработали навыки полученные в ходе лабораторной работы.

5. Выводы

Изучили идеологии и научились применять средства контроля версий, приобрели практические навыки по работе с системой контроля версий git.