

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Система контроля версий Git

дисциплина: Архитектура компьютера

Никулина Ксения НММбд-02-22

Содержание

1. Цель работы

2. Задание

3. Теоретическое введение

4. Выполнение лабораторной работы

5. Самостоятельная работа

6. Вывод

1. Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git

2. Задание

Изучить идеологию и применение средств контроля версий.

Опробовать на практике навыки по работе с системой git.

3. Теоретическое введение

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.

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.

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.

Стандартные процедуры работы при наличии центрального репозитория

Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений):

`git checkout master` `git pull` `git checkout -b имя_ветки` Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральной репозитории. Для этого необходимо проверить, какие файлы изменились к текущему моменту: `git status` и при необходимости удаляем лишние файлы, которые не хотим отправлять в центральный репозиторий. Затем полезно просмотреть текст изменений на предмет соответствия правилам ведения чистых коммитов: `git diff` Если какие-либо файлы не должны попасть в коммит, то помечаем только те файлы, изменения которых нужно сохранить. Для этого используем команды добавления и/или удаления с нужными опциями: `git add имена_файлов` `git rm имена_файлов` Если нужно сохранить все изменения в текущем каталоге, то используем: `git add .` Затем сохраняем изменения, поясняя, что было сделано: `git commit -am "Some commit message"` и отправляем в центральный репозиторий: `git push origin имя_ветки` или `git push`

4. Выполнение лабораторной работы

1.1) Я Создала учётную запись на сайте <https://github.com/> и заполнила основные данные.

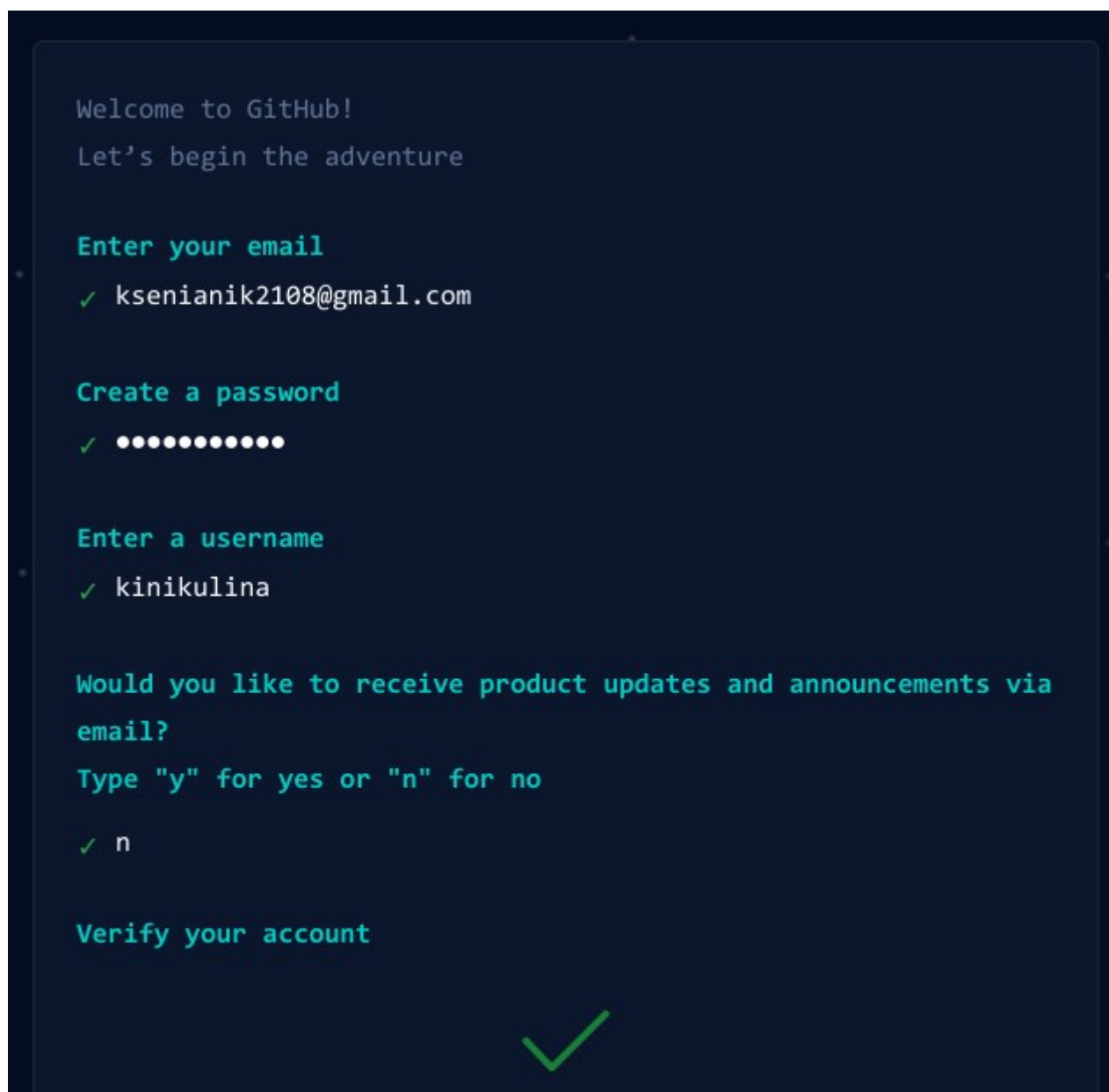


Рис.1.1. Регистрация на сайте

2.1) Открыла терминал и ввела следующие команды, указав свое имя и email

```
kinikulina@dk8n72 ~ $ git config --global user.name "<kinikulina>"
kinikulina@dk8n72 ~ $ git config --global user.email "<ksenianik2108@gmail.com>"
```

Рис.2.1. Имя и емэйл владельца репозитория

2.2) Настроила utf-8 в выводе сообщений git

```
kinikulina@dk8n72 ~ $ git config --global user.name "<kinikulina>"
kinikulina@dk8n72 ~ $ git config --global user.email "<ksenianik2108@gmail.com>"
kinikulina@dk8n72 ~ $ git config --global core.quotepath false
```

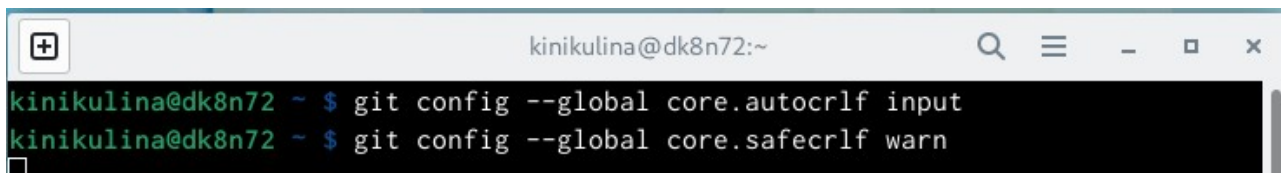
Рис.2.2. Настройка в выводе сообщений git

2.3) Зададала имя начальной ветки (назвала её master)

```
kinikulina@dk8n72 ~ $ git config --global user.name "<kinikulina>"
kinikulina@dk8n72 ~ $ git config --global user.email "<ksenianik2108@gmail.com>"
kinikulina@dk8n72 ~ $ git config --global core.quotepath false
kinikulina@dk8n72 ~ $ git congig --global init.defaultBranch master
```

Рис.2.3. Имя ветки

2.4) Ввела параметры autocrlf и safecrlf



```
kinikulina@dk8n72 ~ $ git config --global core.autocrlf input
kinikulina@dk8n72 ~ $ git config --global core.safecrlf warn
```

Рис.2.4. Параметры

3.1)Сгенерировала пару ключей (приватный и открытый)

```
kinikulina@dk8n72 ~ $ ssh-keygen -C "kinikulina <ksenianik2108@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/i/kinikulina/.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/i/kinikulina/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/i/kinikulina/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:aHNN0rIMIF821GHg0gxhKmJU/7+8//UbabU18eOtR08 kinikulina <ksenianik2108@gmail.com>
The key's randomart image is:
+---[RSA 3072]-----+
| .o.=o*oo.      |
| .  =.0 o..     |
|o.. o.= o o   . |
|o.  ..+ =    o |
|      +.S .   o.+|
|      . o..   ..oB|
|      .      .Bo|
|      . .   .ooo|
|      +o.....E|
+----[SHA256]-----+
kinikulina@dk8n72 ~ $ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис.3.1 Сгенерирование ключей

3.2)Скопировала из локальной консоли ключ в буфер обмена, вставила ключ в появившееся на сайте поле и указываем для ключа имя (domcat).

SSH keys / Add new

Title

domcat

Key type

Authentication Key

Key

```
ssh-rsa
AAAAAB3NzaC1yc2EAAAADAQABAAQgQDb73NnGvb2FOM6pcj87aZdARuJnHnZMq7p66aCPmq4UoQpLM56ZGh5DMX
EOiVFIH+WnsMatb8LmlxZcwXZy/LoEmJ9nNJGZDPSVPNPA3zTk4gE28i3hLmd6qliZfrY82CxLbXw2evdgHPPIV2+EyKE4OQi
6b4vrb9A6yEDa5/jZJuMoDMbv4cZZp6Fv6CzszkxUchBNPE4XddsLfcBULbB8ef
/2G7B3JzDTPH46smD+F6rNz4EDBaBLFwPDy8YPQJYApe5IA6yywt2g/7kpyp4Guk7rYskGuBrQngLn2IULNyQQXGlnz
/Sbulc4FhwGI2AoxzorLv9YVC3A/Q62/MGfveHYiHvtD/5VfeR2h79Yo81feXS
/YNosrcsK+aAHscGunHX3Gk8W6lWJ4ttPUplOelBtub7sE62N2i37TMcfJUw2jNNSabNC8S1JX7mG3Y81ORXc0dM5WvTCn4
bKuUm8J7Lxxo5HGnetdw14jLmCFHNmElcGqEBd+JL2CPNdc= kinikulina <ksenianik2108@gmail.com>
```

Add SSH key

Рис.3.2. Загрузка сгенерированного открытого ключа

3.3) Проверила, что ключ создан

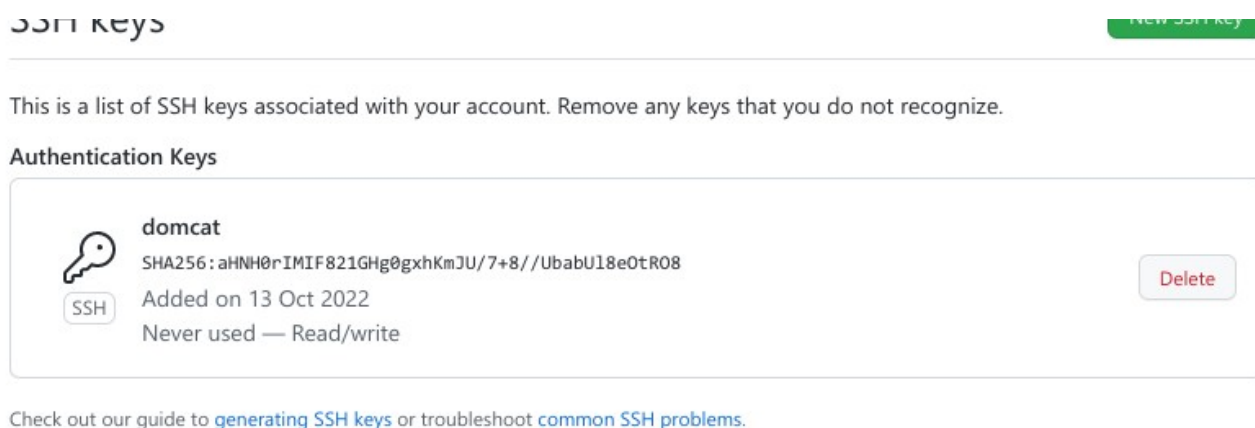


Рис.3.3. Ключ

3.4) Открыла терминал и создала каталог для предмета «Архитектура компьютера»

```
kinikulina@dk8n72 ~ $ mkdir -p ~/work/study/2022-2023/"Архитектура компьютера"
```

Рис.3.4. Создание каталога

4.1) Создала репозиторий

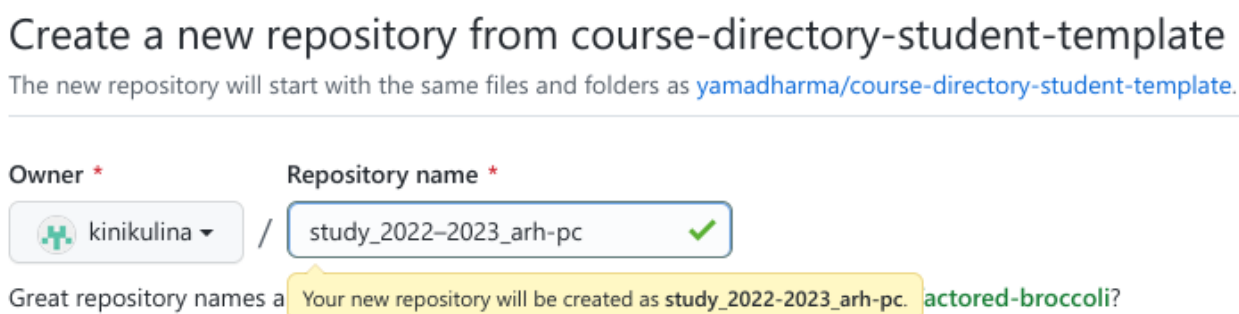


Рис.4.1. Созданный репозиторий

4.2) Клонировала созданный репозиторий

```
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера $ git clone --recursive git@github.com:kinikulina/study_2022-2023_arh-pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 26 (delta 0), reused 17 (delta 0), pack-reused 0
Получение объектов: 100% (26/26), 16.39 КиБ | 16.39 МиБ/с, готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/k/i/kinikulina/work/study/2022-2023/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0
Получение объектов: 100% (71/71), 88.89 КиБ | 1.01 МиБ/с, готово.
Определение изменений: 100% (23/23), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/k/i/kinikulina/work/study/2022-2023/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Получение объектов: 100% (78/78), 292.27 КиБ | 1.95 МиБ/с, готово.
Определение изменений: 100% (31/31), готово.
Submodule path 'template/presentation': checked out '2703b47423792d472694aaf7555a5626dce51a25'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера $
```

Рис.4.2. Клонирование репозитория

5.1) Перешла в каталог курса и удалила лишние файлы

```
kinikulina@dk3n62 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ rm package.json
```

Рис.5.1. Удаление лишних файлов

5.2) Создала необходимые каталоги

```
kinikulina@dk3n62 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ rm package.json
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ echo arch-pc > COURSE
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ make
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $
```

Рис.5.2. Создание каталогов

5.3) Отправила файлы на сервер

```
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ git add .
kinikulina@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ git commit -am 'feat(main): make course structure'
[master 4f5505a] feat(main): make course structure
 91 files changed, 8229 insertions(+), 14 deletions(-)
 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 100644 labs/lab01/report/report.md
 create mode 100644 labs/lab02/presentation/Makefile
 create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
 create mode 100644 labs/lab02/presentation/presentation.md
 create mode 100644 labs/lab02/report/Makefile
 create mode 100644 labs/lab02/report/bib/cite.bib
kinikulina@dk6n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 22, готово.
Подсчет объектов: 100% (22/22), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (20/20), 310.94 КиБ | 7.58 МиБ/с, готово.
Всего 20 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:kinikulina/study_2022-2023_arh-pc.git
 c3e6ffb..4f5505a master -> master
kinikulina@dk6n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $
```

Рис.5.3. Отправление файлов на сервер

5. Самостоятельная работа

1. создала отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab03>report): открыла файл в libreoffice и написала отчет

```
kinikulina@dk6n51 ~ $ cd work/study/2022-2023/Архитектура\ компьютера/arch-pc/labs/lab03/report
kinikulina@dk6n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab03/report $ Л03_Никулина_отчет.docx
bash: Л03_Никулина_отчет.docx: команда не найдена
kinikulina@dk6n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab03/report $ touch Л03_Никулина_отчет.docx
kinikulina@dk6n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab03/report $ libreoffice Л03_Никулина_отчет.docx
```



Рис.6.1. Создание отчета в нужной папке

2. Скопировала отчеты по выполнению предыдущих лабораторных работ из папки лабы в соответствующие каталоги созданного рабочего пространства(было сделано через терминал, но скриншоты не сделаны)

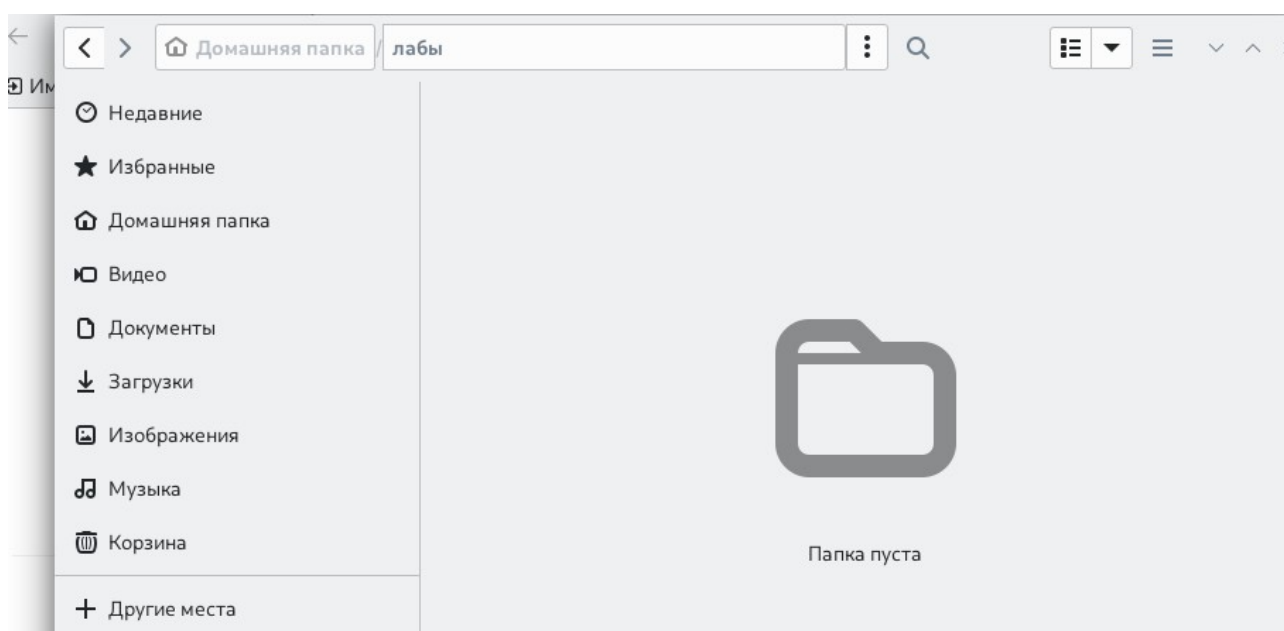


Рис.6.2. Наличие файлов в папке лабы после выполнения задания 2

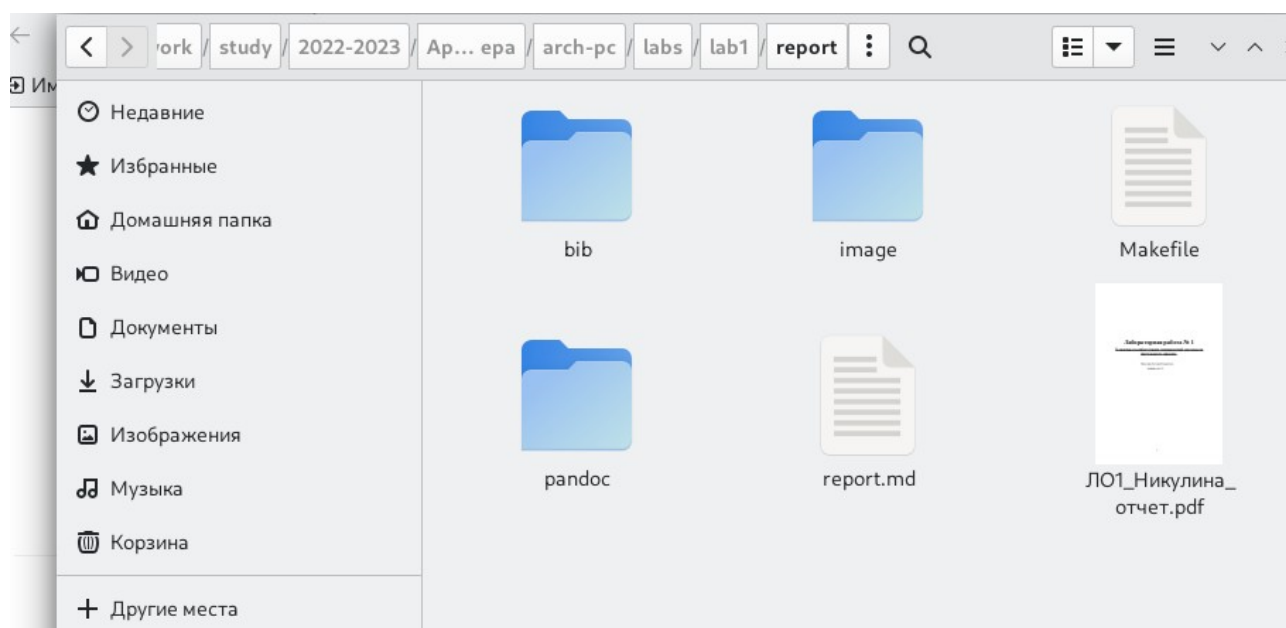


Рис.6.3. Отчет по 1 лабораторной в нужном каталоге созданного рабочего пространства

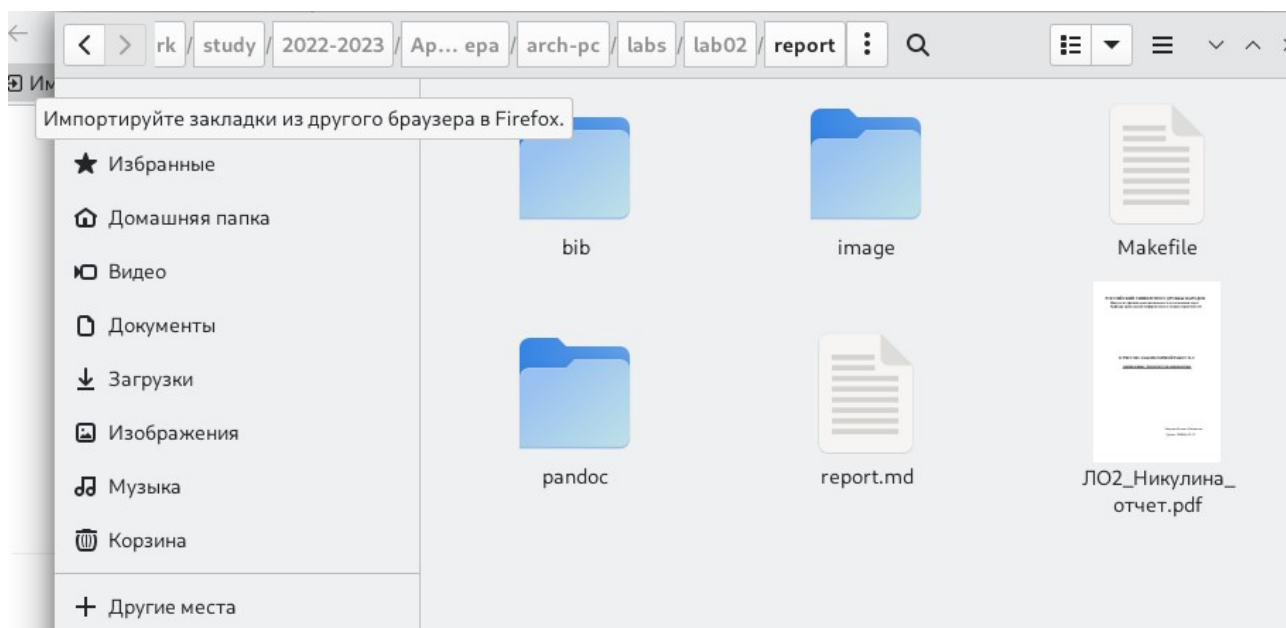


Рис.6.4. Отчет по 2 лабораторной в каталоге созданного рабочего пространства

3. Загрузила файлы на github и проверила их наличие

```
kinikulina@dk6n51 ~ $ cd work/study/2022-2023/Архитектура\ компьютера/arch-pc/labs
kinikulina@dk6n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs $ git add .
kinikulina@dk6n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs $ git status
На ветке master
Ваша ветка обновлена в соответствии с «origin/master».

Изменения, которые будут включены в коммит:
(используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    новый файл:   lab02/report/Л02_Никулина_отчет.pdf
    новый файл:   lab03/report/Л03_Никулина_отчет.docx
    переименовано: lab01/presentation/Makefile -> lab1/presentation/Makefile
    переименовано: lab01/presentation/image/kulyabov.jpg -> lab1/presentation/image/kulyabov.jpg
    переименовано: lab01/presentation/presentation.md -> lab1/presentation/presentation.md
    переименовано: lab01/report/Makefile -> lab1/report/Makefile
    переименовано: lab01/report/bib/cite.bib -> lab1/report/bib/cite.bib
    переименовано: lab01/report/image/placeimg_800_600_tech.jpg -> lab1/report/image/placeimg_800_600_tech.jpg
    переименовано: lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl -> lab1/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
    переименовано: lab01/report/report.md -> lab1/report/report.md
    новый файл:   lab1/report/Л01_Никулина_отчет.pdf
```

6. Вывод

В ходе лабораторной работы мы ознакомились с системой контроля версий GIT, узнали разницу между централизованным VCS и распределенными (такими как GIT), Создали репозиторий из шаблона на сервисе GitHub и внесли в него некоторые изменения. Мы освоили базовые команды утилиты git и создали рабочее пространство для следующих лабораторных работ.