

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
по лабораторной работе № 3

дисциплина: Архитектура компьютера

Студент: Акопян Сатеник
Группа:НКАбд-01-22

МОСКВА
2022г.

Цель работы

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

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. 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.

Выполнение работы.

1. **Настройка github.** Создаем учётную запись на сайте <https://github.com/> и заполняем основные данные. (рис.1)

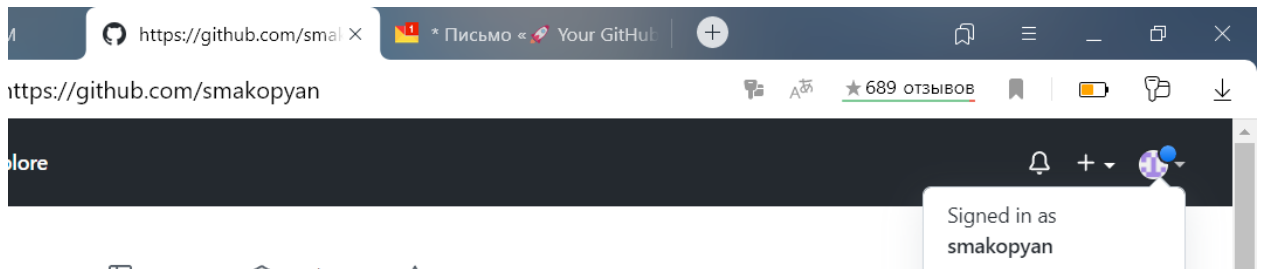


Рисунок 1

2. **Базовая настройка git**

Сделаем предварительную конфигурацию git, а так же настроим utf-8 в выводе сообщений git и зададим имя начальной ветки (будем называть её master). (рис.2)

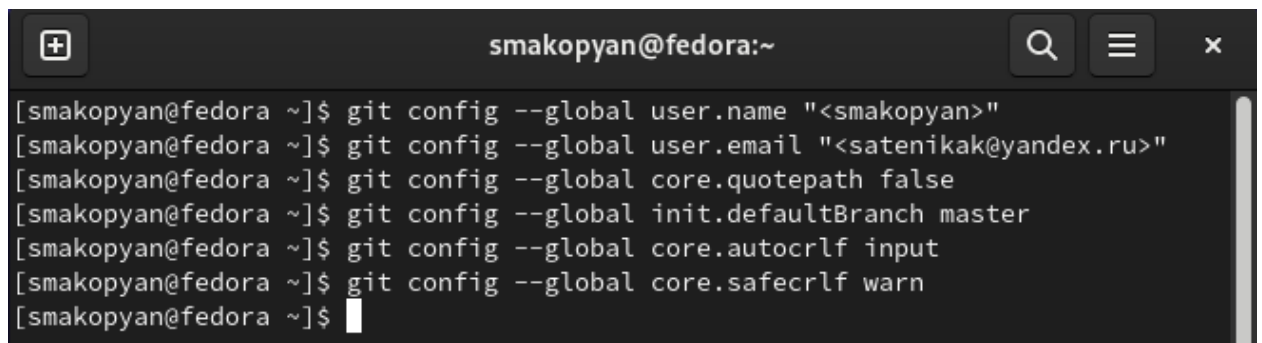


Рисунок 2

3. **Создание SSH ключа**

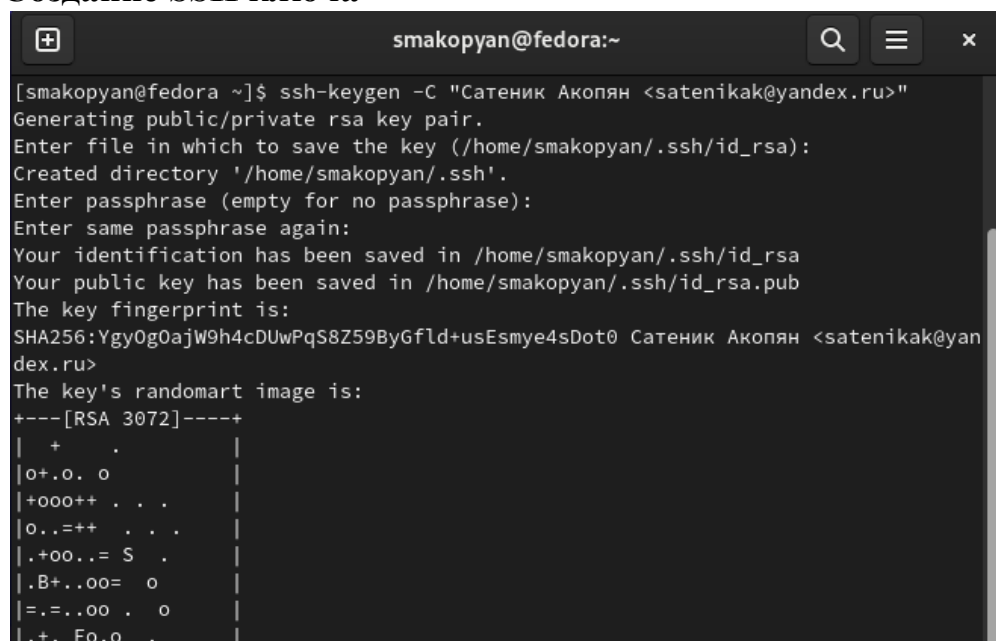
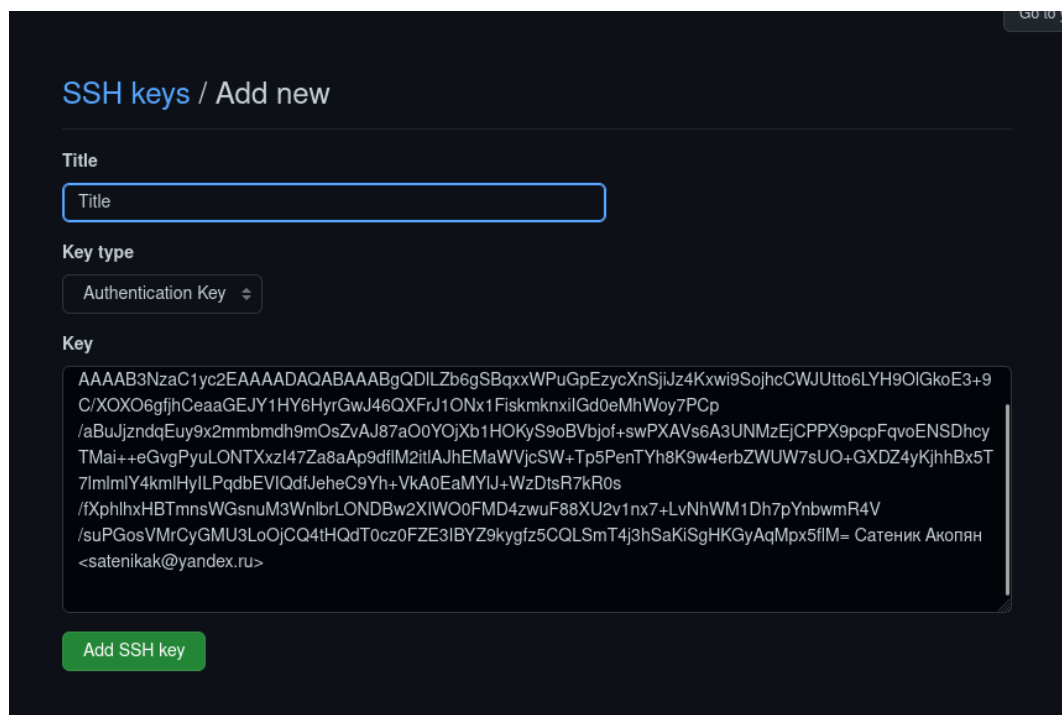


Рисунок 3

3.1 Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый (рис.3))

3.2 Далее необходимо загрузить на сайт сгенерённый открытый ключ(рис.4)



SSH keys / Add new

Title

Key type

Authentication Key

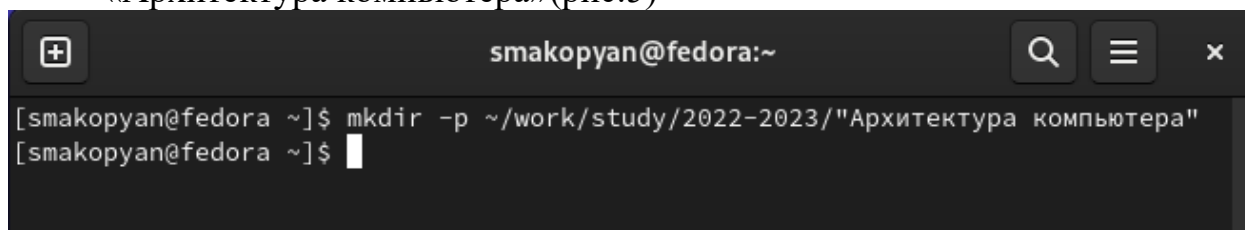
Key

AAAAB3NzaC1yc2EAAAADAQABAAQGDILZb6gSBqxxWPuGpEzycXnSjiJz4Kxwi9SojhcCWJUtto6LYH9OIGkoE3+9C/XOXO6gfjhcEaaGEJY1HY6HyrGwJ46QXFrJ1ONx1FiskmknxiGd0eMhWoy7PCp/aBuJjzndqEuy9x2mmbmdh9mOsZvAJ87aO0YOjXb1HOKyS9oBVbjof+swPXAvs6A3UNMzEjCPPX9pcpFqvoENSdhcyTMai++eGvgPyuLONTXxzI47Za8aAp9dfIM2itlAJhEMaWVjcSW+Tp5PenTYh8K9w4erbZWUW7sUO+GXDZ4yKjhhBx5T7lmimlY4kmlHyILPqdbEVIQdfJeheC9Yh+VKA0EaMYIJ+WzDtsR7kR0s/iXphlhxHBTmnsWGsnuM3WnlbrLONDBw2XIWO0FMD4zwwF88XU2v1nx7+LvNhWM1Dh7pYnbwmR4V/suPGosVMrCyGMU3LoOjCQ4tIHQdT0cz0FZE3lBYZ9kygfz5CQLSmT4j3hSaKiSgHKGAqMpx5fIM= Сатеник Аюлян<satenikak@yandex.ru>

Add SSH key

Рисунок 4

4. Создание рабочего пространства и репозитория курса на основе шаблона. Открываем терминал и создаем каталог для предмета «Архитектура компьютера»(рис.5)



```
smakopyan@fedora:~  
[smakopyan@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Архитектура компьютера"  
[smakopyan@fedora ~]$
```

Рисунок 5

5. Создание репозитория курса на основе шаблона

5.1 Репозиторий на основе шаблона можно создать через web-интерфейс github, для этого переходим на страницу репозитория с шаблоном курса, задаем имя репозитория study_2022–2023_arh-pc и создаем его. (рис.6)

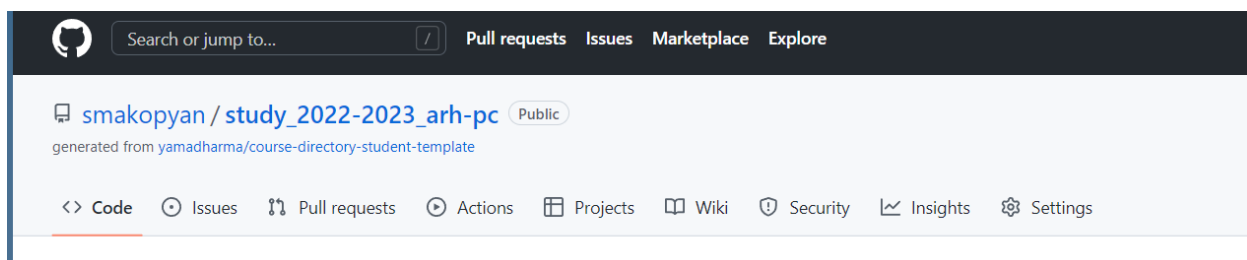


Рисунок 6

5.2 Далее открываем терминал и переходим в каталог курса “Архитектура компьютера”.(рис.7)

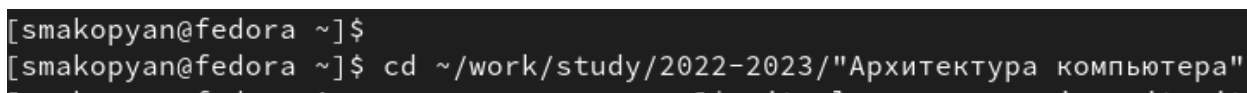


Рисунок 7

5.3 Клонировем созданный репозиторий(рис.8)

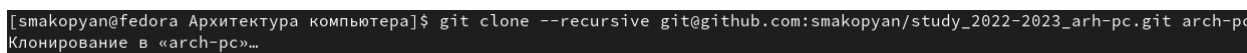


Рисунок 8

6. Настройка каталога курса.

6.1 Переходим в каталог курса, удаляем лишние файлы и создаем необходимые каталоги.(рис.9)

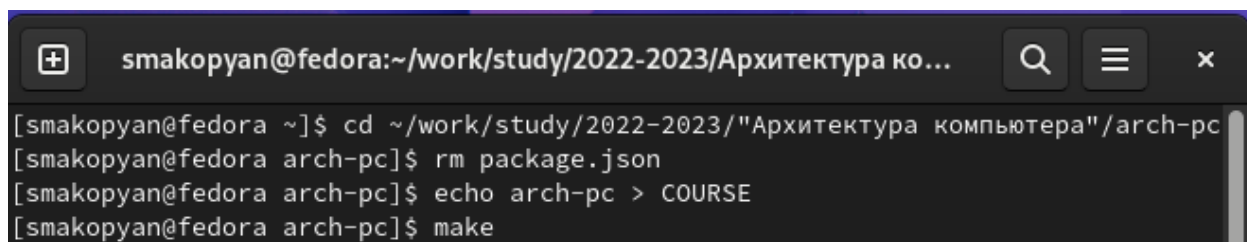


Рисунок 9

6.2 Отправляем файлы на сервер. (рис.10, рис.11)

```
[smakopyan@fedora arch-pc]$ git push
Перечисление объектов: 22, готово.
Подсчет объектов: 100% (22/22), готово.
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (20/20), 310.94 КиБ | 2.07 МиБ/с, готово.
Всего 20 (изменений 1), повторно использовано 0 (изменений 0), повторно использо
вано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:smakopyan/study_2022-2023_arh-pc.git
   7820837..76e3dbe  master -> master
[smakopyan@fedora arch-pc]$
```

Рисунок 10

```
[smakopyan@fedora arch-pc]$ git add .
[smakopyan@fedora arch-pc]$ git commit -am 'feat(main):make course structure'
[master 76e3dbe] 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
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
```

Рисунок 11

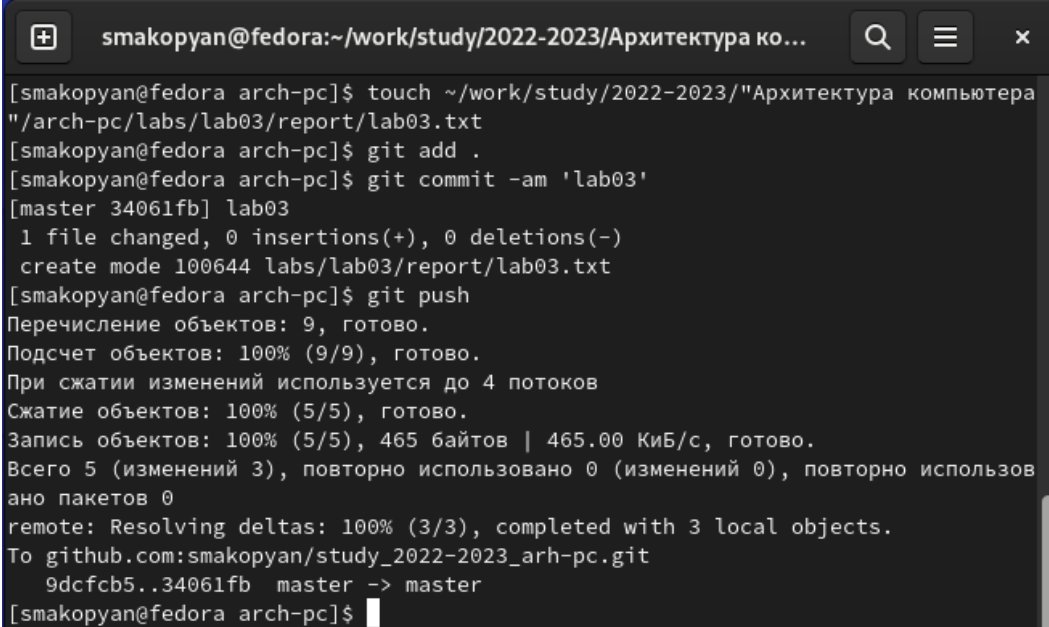
6.3 Проверяем правильность создания иерархии рабочего пространства (рис.11)

smakopyan feat(main): make course structure		d5cec29 11 minutes ago	4 commits
config	Initial commit	10 days ago	
labs	feat(main):make course structure	10 days ago	
template	Initial commit	10 days ago	
.gitattributes	Initial commit	10 days ago	
.gitignore	Initial commit	10 days ago	
.gitmodules	Initial commit	10 days ago	
CHANGELOG.md	Initial commit	10 days ago	
COURSE	feat(main): make course structure	11 minutes ago	
LICENSE	Initial commit	10 days ago	
Makefile	Initial commit	10 days ago	
README.en.md	Initial commit	10 days ago	
README.git-flow.md	Initial commit	10 days ago	
README.md	Initial commit	10 days ago	

Рисунок 10

Задание для самостоятельной работы

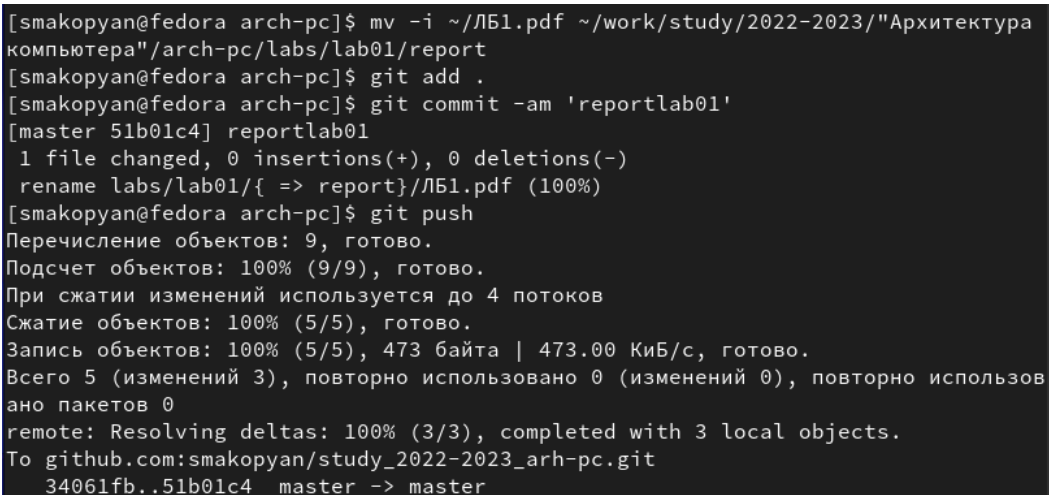
1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab03>report).(рис.13)



```
smakopyan@fedora:~/work/study/2022-2023/Архитектура ко...
[smakopyan@fedora arch-pc]$ touch ~/work/study/2022-2023/"Архитектура компьютера"
/arch-pc/labs/lab03/report/lab03.txt
[smakopyan@fedora arch-pc]$ git add .
[smakopyan@fedora arch-pc]$ git commit -am 'lab03'
[master 34061fb] lab03
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab03/report/lab03.txt
[smakopyan@fedora arch-pc]$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 465 байтов | 465.00 КиБ/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:smakopyan/study_2022-2023_arh-pc.git
9dcfcb5..34061fb master -> master
[smakopyan@fedora arch-pc]$
```

Рисунок 13

2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства. (рис.14, рис.15)



```
[smakopyan@fedora arch-pc]$ mv -i ~/ЛБ1.pdf ~/work/study/2022-2023/"Архитектура
компьютера"/arch-pc/labs/lab01/report
[smakopyan@fedora arch-pc]$ git add .
[smakopyan@fedora arch-pc]$ git commit -am 'reportlab01'
[master 51b01c4] reportlab01
1 file changed, 0 insertions(+), 0 deletions(-)
rename labs/lab01/{ => report}/ЛБ1.pdf (100%)
[smakopyan@fedora arch-pc]$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 473 байта | 473.00 КиБ/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:smakopyan/study_2022-2023_arh-pc.git
34061fb..51b01c4 master -> master
```

Рисунок 14


```

[smakopyan@fedora arch-pc]$ mv -i ~/ЛБ2.pdf ~/work/study/2022-2023/"Архитектура
компьютера"/arch-pc/labs/lab02/report
[smakopyan@fedora arch-pc]$ git add .
[smakopyan@fedora arch-pc]$ git commit -am 'lab02report'
[master 70ae0e6] lab02report
1 file changed, 0 insertions(+), 0 deletions(-)
rename labs/lab02/report/{Лабораторная работа №2.pdf => ЛБ2.pdf} (100%)
[smakopyan@fedora arch-pc]$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 451 байт | 451.00 КиБ/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:smakopyan/study_2022-2023_arh-pc.git
51b01c4..70ae0e6 master -> master

```

Рисунок 15

3. Загрузите файлы на github. (рис.14, рис.15)

Вывод: в результате выполнения данной лабораторной работы я изучила идеологию и применение средств контроля версий, приобрела практические навыки по работе с системой git.

