

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: Четвергова Мария Викторовна

Студенческий билет № 1132232886

Группа: НПИбд-02-23

Москва

2023 г.

Содержание

1. Цель работы	4
2. Задание	5
3. Теоретическое введение	6
4. Выполнение лабораторной работы.....	8
5. Выводы	20
Список литературы.....	21

Иллюстрации:

- 1) Рисунок 1 «Учетная запись git.hub » -----стр.8
- 2) Рисунок 2 «Рис.2 предварительная конфигурация git» -----стр.9
- 3) Рисунок 3 «СОЗДАНИЕ SSH ключа» -----стр.9
- 4) Рисунок 4 «создание каталога для предмета «Архитектура компьютера»:»---стр.10
- 5) Рисунок 5 «репозиторий курса на основе шаблона» -----стр.10
- 6) Рисунок 6 «клонировем созданный репозиторий» -----стр.11
- 7) Рисунок 7-8 «настройка каталога курса и удаление лишних файлов» -----стр.11
- 8) Рисунок 9 «отправка файлов на сервер» -----стр.12
- 9) Рисунок 10-11 «создание отчёта в каталоге lab2»-----стр.13
- 10)Рисунок 12 «копирование отчётов по выполненной работе»-----стр.14
- 11)Рисунок 13 «загрузка отчётов на git.hub»-----стр.15

1. Цель работы

Целью работы является изучение идеологии и применение средств контроля версий. Важной составляющей лабораторной работы № 2 является приобретение практических навыков по работе с системой git.

2. Задание

Приобрести теоретические навыки работы с системой git. Практически закрепить знания работы и применить на практике полученные знания об идеологии и применении средств контроля версий. Усвоить знания, приобретённые в ходе выполнения лабораторной работы № 2. И получить необходимые навыки для работы с системой git.

3. Теоретическое введение

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. 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. Техническое обеспечение Лабораторная работа подразумевает выполнение настройки и работы с системой контроля версий Git (<https://git-scm.com/>). Выполнение работы возможно как в дисплейном классе факультета физико-математических и естественных наук РУДН, так и дома. Описание выполнения работы приведено для дисплейного класса со следующими характеристиками техники:

- Intel Core i3-550 3.2 GHz, 4 GB оперативной памяти, 8 GB свободного места на жёстком диске;
- ОС Linux Gentoo (<http://www.gentoo.ru/>)

4. Выполнение лабораторной работы

4.1 Настройка github

Существует несколько доступных серверов репозитория с возможностью бесплатного размещения данных. Например, <http://bitbucket.org/>, <https://github.com/> и <https://gitflic.ru>. Для выполнения лабораторных работ предлагается использовать Github. Создадим учётную запись на сайте <https://github.com/> и заполним основные данные:

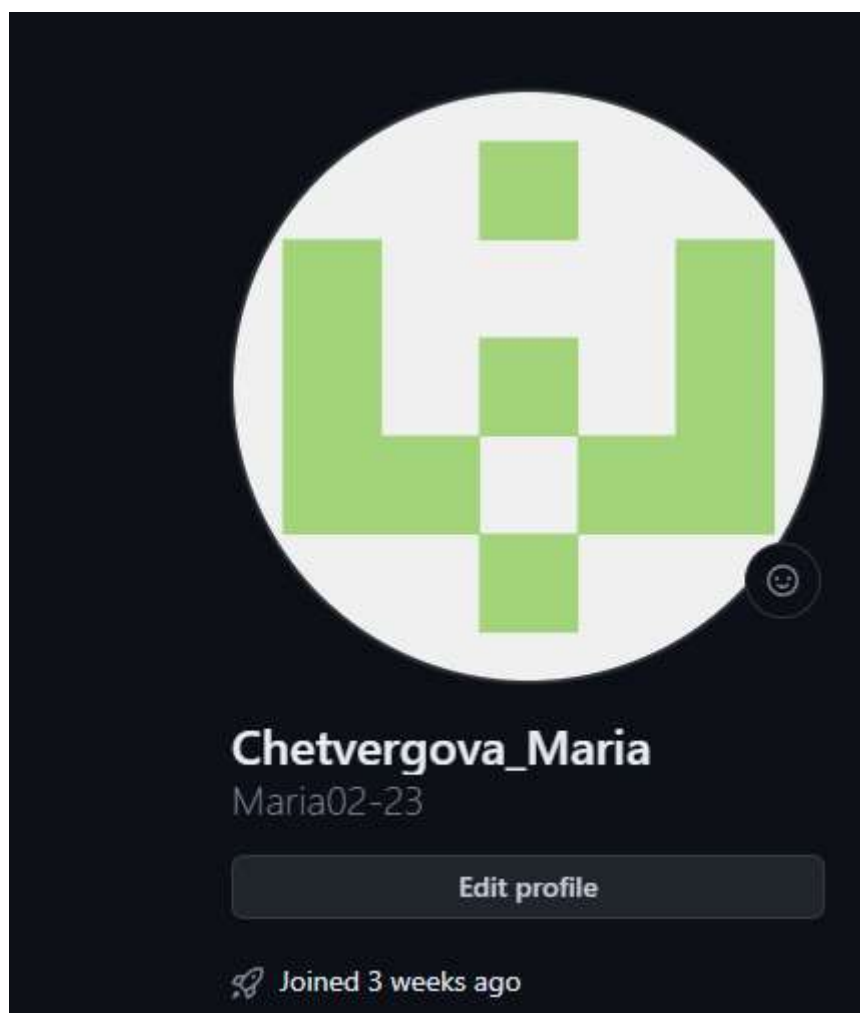


рис.1 Учётная запись git.hub

4.2. Базовая настройка git


```
mvchetvergova@dk8n59 - $ git config --global user.name "Maria82-23"
mvchetvergova@dk8n59 - $ git config --global user.email "1132232886@pfr.ru"
mvchetvergova@dk8n59 - $ git config --global core.quotepath false
mvchetvergova@dk8n59 - $ git config --global init.defaultBranch master

mvchetvergova@dk8n59 - $ git config --global core.quotepath false
mvchetvergova@dk8n59 - $ git config --global init.defaultBranch master

mvchetvergova@dk8n59 - $
mvchetvergova@dk8n59 - $ git config --global core.autocrlf input

fatal: not in a git directory
mvchetvergova@dk8n59 - $
mvchetvergova@dk8n59 - $ git config --global core.autocrlf input
mvchetvergova@dk8n59 - $ git config --global core.safecrlf warn
```

4.3. Создание SSH ключа

Далее необходимо загрузить сгенерённый открытый ключ. Для этого зайти на сайт [github](#) под своей учётной записью и перейти в меню **Setting** . После этого выбрать в боковом меню **SSH and GPG keys** и нажать кнопку **New SSH key** . Скопировав из локальной консоли ключ в буфер вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title):

```
Owens@lisa [~/03]$ mvchvetvergova@dk6n59 ~$ ssh-keygen -t rsa -C "Haria02-23 1132232886@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/dk.sci.pfu.edu.ru/home/m/v/mvchetvergova/.ssh/id_rsa): /afs/dk.sci.pfu.edu.ru/home/m/v/mvchetvergova/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase): 
Enter same passphrase again: 
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase): 
Enter same passphrase again: 
Your identification has been saved in /afs/dk.sci.pfu.edu.ru/home/m/v/mvchetvergova/.ssh/id_rsa
Your public key has been saved in /afs/dk.sci.pfu.edu.ru/home/m/v/mvchetvergova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:y1DQmKJpgzZicd9j1RYsDJ4VNH/TyNcR/geYU5j1/I Haria02-23 1132232886@pfur.ru
The key's randomart image is:
[ RSA 3072 ]
+-----+
|      .   +      |
|    o + . + . +   |
|  + + + . + S . + |
| = O .   o i o    |
| @ o .   L *      |
| 3ooo          |
| m i . o         |
|-----+ SHA256 |-----+
mvchvetvergova@dk6n59 ~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

9

4.4. Создание рабочего пространства и репозитория курса на основе шаблона

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства. Название проекта на хостинге git имеет вид:

study_<учебный год>_<код предмета>

Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) название проекта примет следующий вид:

study_2023–2024_arch-pc

Откроем терминал и создадим каталог для предмета «Архитектура компьютера»:

```
muchetvergova@dk2n59 ~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
muchetvergova@dk2n59 ~$ mkdir -p ~/work/study/2023-2024/«Архитектура компьютера»
muchetvergova@dk2n59 ~$ cd ~/work/study/2023-2024/«Архитектура компьютера»
muchetvergova@dk2n59 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:Maria07-23/st
Клонирование в «arch pc»...
ssh: Could not resolve hostname github.com: Device or resource busy
fatal: Не удалось прочитать из внешнего репозитория.
```

Рис.4 создание каталога для предмета «Архитектура компьютера»:

4.5. Создание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github.

Перейдём на страницу репозитория с шаблоном курса. Далее выбираем *Use this template*:

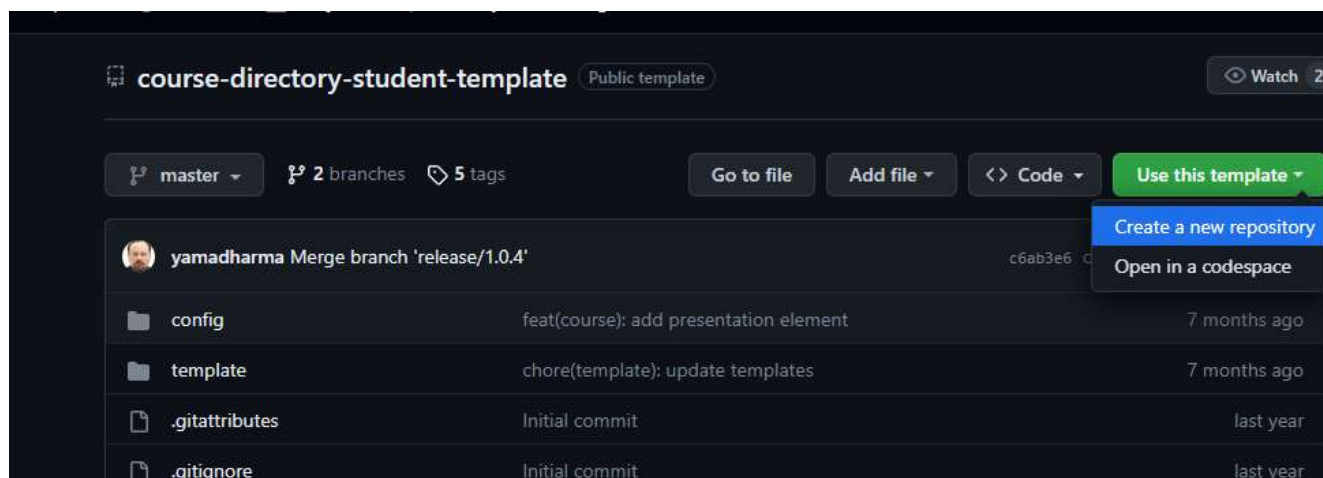


Рис.5 репозиторий курса на основе шаблона

В открывшемся окне задайте имя репозитория (Repository name) study_2023–2024_arhps и создайте репозиторий (кнопка Create repository from template).

Откроем терминал и перейдём в каталог курса, клонируем созданный репозиторий:

```
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:Harib02-23/study_2023-2024
Клонирование в «arch-pc»...
The authenticity of host 'github.com [140.82.121.3]' can't be established.
ED25519 key fingerprint is: SHA256:+DiY3wvV6lujhBpzisf/2L0A8dPMSvHdk4UvC0q0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 6
```

Рис. 6 клонируем созданный репозиторий

4.6. Настройка каталога курса

Перейдите в каталог курса. Удалите лишние файлы, создайте необходимые каталоги и отправим файлы на сервер:

```
Submodule path 'template/report': checked out '1d1b61dcac9c28/a8391fb82c3ac711a33b1c3b2'
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
bash: cd: /afs/dk.sci.pfu.edu.ru/home/m/v/mvchetvergova/work/study/2023-2024/Архитектура компьютера/arch-pc: Нет такого файла или каталога
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера $ cd
mvchetvergova@dkn59: ~ $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ echo -e "\nmake" > CMakeLists.txt
```

```
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
bash: cd: /afs/dk.sci.pfu.edu.ru/home/m/v/mvchetvergova/work/study/2023-2024/Архитектура компьютера/arch-pc: Нет такого файла или каталога
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера $ cd
mvchetvergova@dkn59: ~ $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ echo -e "\nmake" > CMakeLists.txt
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add
mvchetvergova@dkn59: ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -m "first commit: make course structure"
[master 1ab158e] test(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/PKADPP.md
create mode 100644 labs/PKADPP.ru.md
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/chih/cite.bib
create mode 100644 labs/lab01/report/image/placing_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-1000-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_argon.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_ignores.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnoa.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnoa_init.py
create mode 100644 labs/lab01/report/pandoc/filters/pandoc_secnoa_core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandoc_secnoa_main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandoc_secnoa/pandoc_attributes.py
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/chih/cite.bib
create mode 100644 labs/lab02/report/image/placing_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-1000-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_argon.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_ignores.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnoa.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnoa_init.py
create mode 100644 labs/lab02/report/pandoc/filters/pandoc_secnoa_core.py
create mode 100644 labs/lab02/report/pandoc/filters/pandoc_secnoa_main.py
create mode 100644 labs/lab02/report/pandoc/filters/pandoc_secnoa/pandoc_attributes.py
```

Рис. 7-8 настройка каталога курса и удаление лишних файлов


```

create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab02/report/report.md
create mode 100644 labs/lab03/presentation/Makefile
create mode 100644 labs/lab03/presentation/image/kulyabov.jpg
create mode 100644 labs/lab03/presentation/presentation.md
create mode 100644 labs/lab03/report/Makefile
create mode 100644 labs/lab03/report/bib/cite.bib
create mode 100644 labs/lab03/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab03/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab03/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab03/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab03/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab03/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab03/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab03/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab03/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab03/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab03/report/report.md
create mode 100644 labs/lab04/presentation/Makefile
create mode 100644 labs/lab04/presentation/image/kulyabov.jpg
create mode 100644 labs/lab04/presentation/presentation.md
create mode 100644 labs/lab04/report/Makefile
create mode 100644 labs/lab04/report/bib/cite.bib
create mode 100644 labs/lab04/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab04/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab04/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab04/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab04/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab04/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab04/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab04/report/pandoc/filters/pandocxnos/core.py

```

Рис. 9 отправка файлов на сервер

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

Пояснение: командой «`cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc`» перейдём в каталог курса и с помощью команды «`rm package.json`» удалим лишние файлы. Затем создадим новые необходимые файлы и добавим файлы на сервер. Затем переместимся на страницу github и проверим правильность создания репозитория.

5. Самостоятельная работа

5.1 Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report):

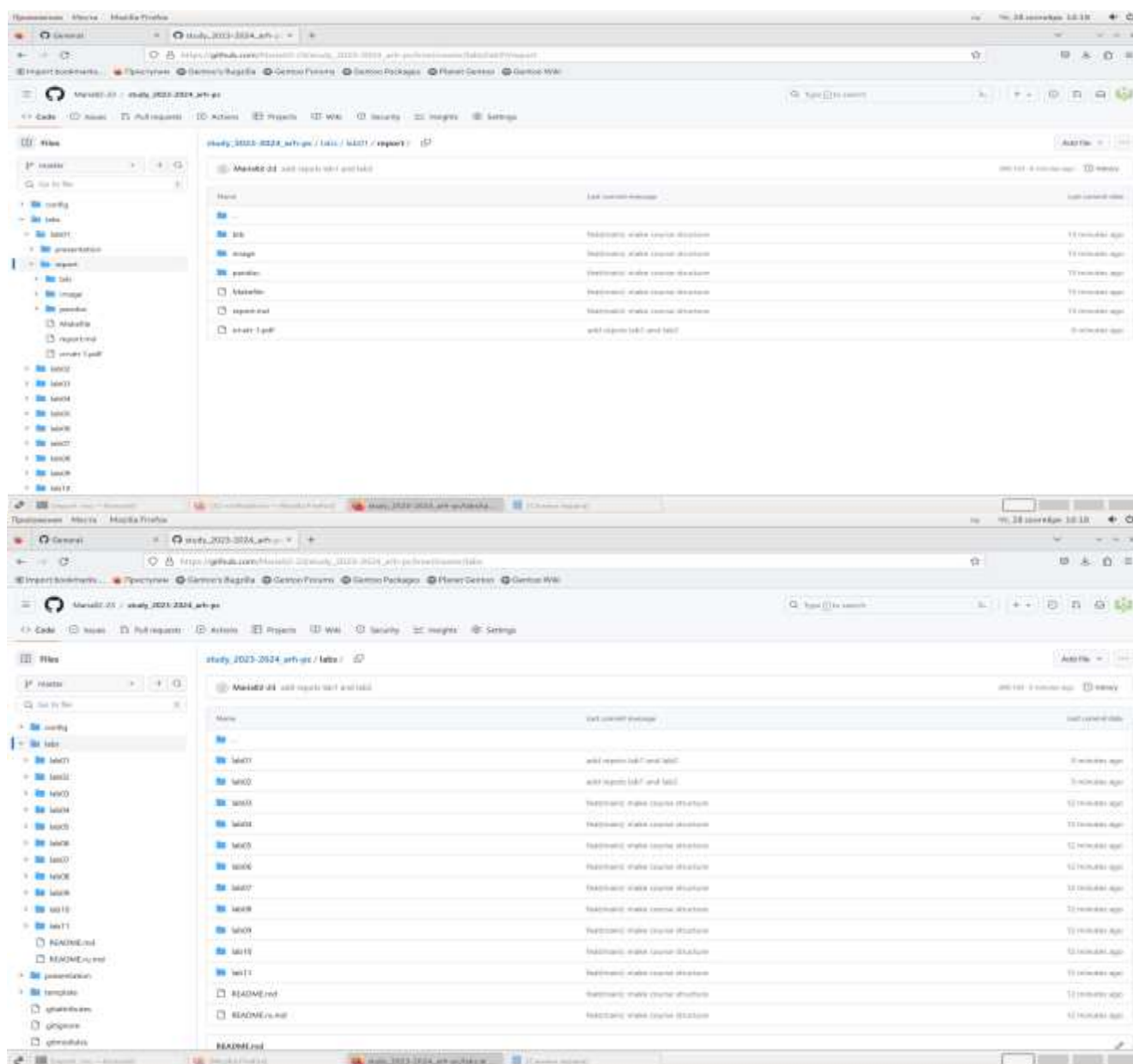


Рис.10-11 создание отчёта в каталоге lab2

Пояснение: В подкаталоге lab02 каталога labs создаём файл report. Данный файл содержит отчёт по выполнению лабораторной работы №2. Таким образом отчёт по выполнению лабораторной работы хранится в подкаталоге lab02 каталога labs

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

[illegible]

Рис.12 копирование отчётов по выполненной работе

5.3 Загрузите файлы на github

[illegible]

Рис.13 загрузка файлов на git.hub

Вывод

В ходе лабораторной работы №2 мы получили необходимые теоретические знания для работы с системой git, а также закрепили полученные знания на практике. Изучены базовые команды для работы с системой git, изучили идеологию и применение средств контроля версий. Работа выполнена с помощью операционной системы Linux и сайта git hub. Таким образом получены базовые знания, позволяющие работать с системой git и средствами контроля версий.

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/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
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. — БХВПетербург, 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).