

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

ДИСЦИПЛИНА: Система контроля версий GIT

Студент: Джахангиров Илгар

Группа: НКАбд-05-22

№ ст. билета: 1032225689

МОСКВА

2022 г.

СОДЕРЖАНИЕ:

1. Цель работы.
2. 3.2. Теоретическое введение
3. 3.2.1. Системы контроля версий. Общие понятия
4. 3.2.2. Система контроля версий Git
5. 3.2.3. Основные команды git.
6. 3.2.4. Стандартные процедуры работы при наличии центрального репозитория
7. 3.3. Техническое обеспечение
8. 3.4. Порядок выполнения лабораторной работы
 - 3.4.1. Настройка github
 9. 3.4.2. Базовая настройка git
 10. 3.4.3. Создание SSH ключа
 11. 3.4.4. Создание рабочего пространства и репозитория курса на основе Шаблона.
 12. 3.4.5. Создание репозитория курса на основе шаблона
 13. 3.4.6. Настройка каталога курса
 14. Самостоятельная работа

.1. Цель работы

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

3.2. Теоретическое введение

3.2.1. 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.

3.2.2. 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.

3.2.3. Основные команды git.

Таблица 3.1. Основные команды git

Наиболее часто используемые команды git представлены ниже:

git init-создание основного дерева репозитория

git pull-олучение обновлений (изменений) текущего дерева из центрального репозитория.

git push-отправка всех произведённых изменений локального дерева в центральный репозиторий.

git status-просмотр списка изменённых файлов в текущей директории.

git diff-просмотр текущих изменения.

git add-добавить конкретные изменённые и/или созданные файлы и/или каталоги.

git rm имена_файлов-удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной.

it commit-am 'Описание коммита'-сохранить все добавленные изменения и все изменённые файлы.

git checkout-b имя_ветки-создание новой ветки, базирующейся на текущей.

git checkout имя_ветки-переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой).

git push origin-отправка изменений конкретной ветки в центральный репозиторий.

git merge--no-ff-слияние ветки с текущим деревом.

git branch -d имя_ветки-удаление локальной уже слитой с основным деревом ветки.

git branch -d имя_ветки-удаление локальной уже слитой с основным деревом ветки.

git branch -D имя_ветки-принудительное удаление локальной ветки.

git push origin-удаление ветки с центрального репозитория.

3.2.4. 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 центральном 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"

git push origin имя_ветки

или

git push

3.3. 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/>);

3.4. Порядок выполнения лабораторной работы

3.4.1. 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.

3.4.2. Базовая настройка git

Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория:

git config --global user.name "<Name Surname>"

git config --global user.email "<work@mail>"

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**:

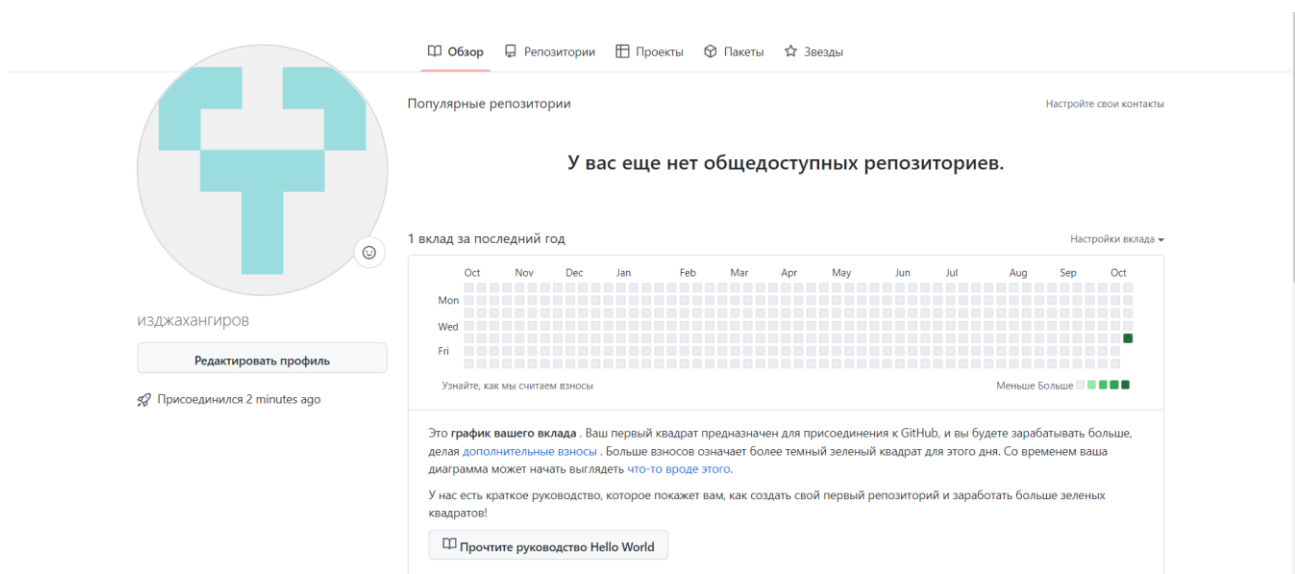
git config --global core.autocrlf input

```
izdzhakhangirov@fedora:~$ git config --global user.name "izdzhakhangirov"
[izdzhakhangirov@fedora ~]$ git config --global user.email "1032225689@pfur.ru"
[izdzhakhangirov@fedora ~]$ git config --global core.quotepath false
[izdzhakhangirov@fedora ~]$ git config --global init.defaultbranch master
[izdzhakhangirov@fedora ~]$ git config --global core.autocrlf input
[izdzhakhangirov@fedora ~]$ git config --global core.safecrlf warn
[izdzhakhangirov@fedora ~]$
```

Параметр **safecrlf**:

git config --global core.safecrlf warn

Для этого зайти на сайт <http://github.org/> под своей учётной записью и перейти в меню Setting .



3.4.3. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):

ssh-keygen -C "Имя Фамилия <work@mail>"

Ключи сохраняются в каталоге **~/.ssh/**.

```
[izdzhakhangirov@fedora ~]$ ssh-keygen -C "caxanqirov ilqar 1032225689@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/izdzhakhangirov/.ssh/id_rsa):
Created directory '/home/izdzhakhangirov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/izdzhakhangirov/.ssh/id_rsa
Your public key has been saved in /home/izdzhakhangirov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:TQsCV/aylF/9ioIU1+gsWX/bCSEbuPrPsZFC0yMwP5M caxanqirov ilqar 1032225689@pfur.ru
The key's randomart image is:
+----[RSA 3072]-----+
|      . . .      |
|    o . o . o .  |
|   . * . * o o .  |
|  o # . + . + .  |
| o So* . . . . . |
| + . + o . o . + |
| E . = . + . o . |
|   = . +         |
|   . . +         |
+----[SHA256]-----+
[izdzhakhangirov@fedora ~]$
```

Далее необходимо загрузить сгенерённый открытый ключ.

izdzhakhangirov
Your personal account

Public profile
Account
Appearance
Accessibility
Notifications

Access
Billing and plans
Emails
Password and authentication
SSH and GPG keys
Organizations
Moderation

Code, planning, and automation
Repositories
Packages
GitHub Copilot
Pages

SSH keys / Add new

Title
ilqar

Key type
Authentication Key

Key
Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

3.4.4. Создание рабочего пространства и репозитория курса на основе шаблона

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства. Рабочее пространство по предмету располагается в следующей иерархии:

~/work/study/

└─ <учебный год>/

└─ <название предмета>/

└─ <код предмета>/

Каталог для лабораторных работ имеет вид labs.

- Каталоги для лабораторных работ имеют вид lab<номер>, например: lab01, lab02 и т.д.

Название проекта на хостинге git имеет вид: study_<учебный год>_<код предмета>

Откройте терминал и создайте каталог для предмета «Архитектура компью- тера»:

`mkdir -p ~/work/study/2022-2023/"Архитектура компьютера"`

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

3.4.5. Создание репозитория курса на основе шаблона

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

Перейдите на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>.

yamadharma / course-directory-student-template Public template

Watch 1 Fork 2 Star 0

<> Code Issues Pull requests Actions Projects Security Insights

73455291b3 2 branches 2 tags Go to file Code Use this template

File	Commit Message	Commit Date
config	feat(course): add sciprolog	last month
template	feat(course): add new course	last month
.gitattributes	Initial commit	6 months ago
.gitignore	Initial commit	6 months ago
.gitmodules	chore(main): add conventional changelog support	6 months ago
CHANGELOG.md	chore(main): update changelog	last month
COURSE	feat(script): add script for auto-determine name of course	2 months ago
LICENSE	Initial commit	6 months ago
Makefile	feat(script): add script for auto-determine name of course	2 months ago
README.en.md	chore(submodules): update submodules	6 months ago
README.git-flow.md	Initial commit	6 months ago
README.md	fix(doc): move documentation to README.md	2 months ago

About

Course Catalog Template for Students

Readme

CC-BY-4.0 license

0 stars

1 watching

2 forks

Releases 2

v1.0.1 Latest on 6 Sep

+ 1 release

Languages

Shell 75.0% Makefile 25.0%

Далее выберите Use this template.

В открывшемся окне задайте имя репозитория (Repository name)

study_2022–2023_arh-pc и создайте репозиторий (кнопка **Create repository from template**).

Create a new repository from course-directory-student-template

The new repository will start with the same files and folders as [yamadharm/course-directory-student-template](#).

Owner *



izdzhakhangirov ▾

/

Repository name *

study_2022-2023_arh-pc



Great repository names are short and memorable. Need inspiration? How about [laughing-telegram](#)?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

☐ Include all branches

Copy all branches from yamadharm/course-directory-student-template and not just master.

You are creating a public repository in your personal account.

Create repository from template

Откройте терминал и перейдите в каталог курса:

```
cd ~/work/study/2022-2023/"Архитектура компьютера"
```

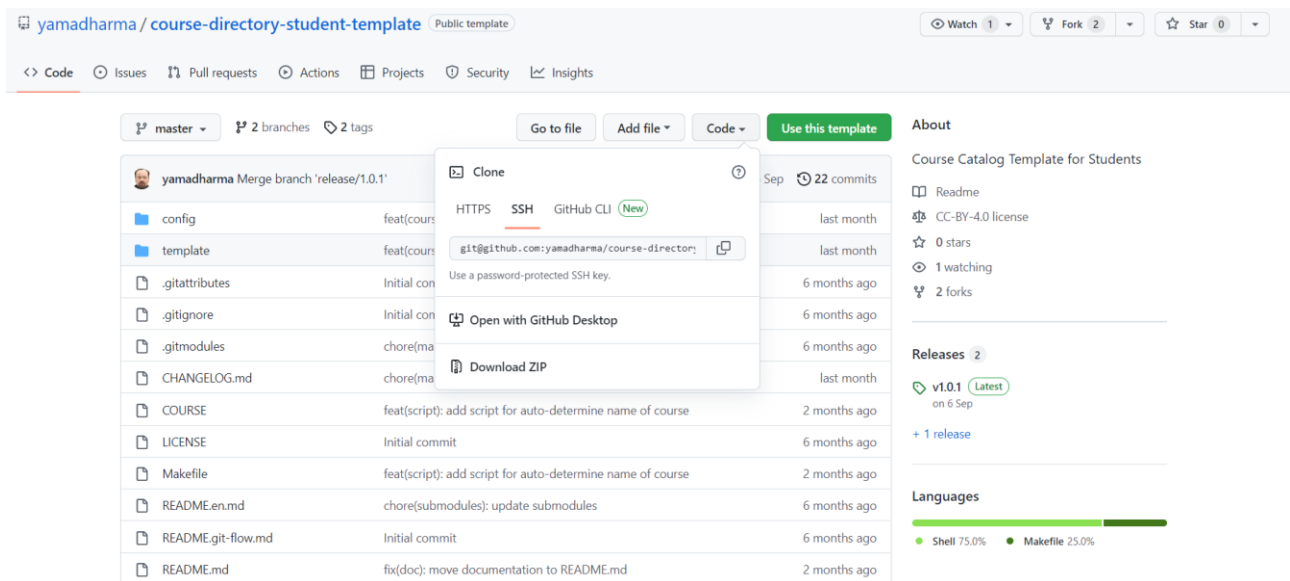
Клонируйте созданный репозиторий:

git clone --recursive

↪ **git@github.com:<user_name>/study_2022-2023_arh-pc.git arch-pc**

```
[izdzhakhangirov@fedora Архитектура компьютера]$ git clone --recursive git@github.com:izdzhakhangirov/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 | 3.28 МиБ/с, готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/izdzhakhangirov/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 | 901.00 КиБ/с, готово.
Определение изменений: 100% (23/23), готово.
Клонирование в «/home/izdzhakhangirov/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 | 1.73 МиБ/с, готово.
Определение изменений: 100% (31/31), готово.
Submodule path 'template/presentation': checked out '2703b47423792d472694aaf7555a562dce51a25'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'
[izdzhakhangirov@fedora Архитектура компьютера]$
```

Ссылку для клонирования можно скопировать на странице созданного репозитория Code -> SSH:



3.4.6. Настройка каталога курса

Перейдите в каталог курса:

```
cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc
```

Удалите лишние файлы:

```
rm package.json
```

Создайте необходимые каталоги:

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

```
make
```

```
[izdzhakhangirov@fedora Архитектура компьютера]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc
[izdzhakhangirov@fedora arch-pc]$ rm package.json
[izdzhakhangirov@fedora arch-pc]$ echo arch-pc > COURSE
[izdzhakhangirov@fedora arch-pc]$ make
[izdzhakhangirov@fedora arch-pc]$ git add .
[izdzhakhangirov@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 703bf7e] 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
```

Отправьте файлы на сервер:

```
git add .
```

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

```
git push
```

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

```

[izdzhakhangirov@fedora Архитектура компьютера]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc
[izdzhakhangirov@fedora arch-pc]$ rm package.json
[izdzhakhangirov@fedora arch-pc]$ echo arch-pc > COURSE
[izdzhakhangirov@fedora arch-pc]$ make
[izdzhakhangirov@fedora arch-pc]$ git add .
[izdzhakhangirov@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 703bf7e] 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/lab09/presentation/image/kulyabov.jpg
create mode 100644 labs/lab09/presentation/presentation.md
create mode 100644 labs/lab09/report/Makefile
create mode 100644 labs/lab09/report/bib/cite.bib
create mode 100644 labs/lab09/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab09/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab09/report/report.md
create mode 100644 labs/lab10/presentation/Makefile
create mode 100644 labs/lab10/presentation/image/kulyabov.jpg
create mode 100644 labs/lab10/presentation/presentation.md
create mode 100644 labs/lab10/report/Makefile
create mode 100644 labs/lab10/report/bib/cite.bib
create mode 100644 labs/lab10/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab10/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab10/report/report.md
create mode 100644 labs/lab11/presentation/Makefile
create mode 100644 labs/lab11/presentation/image/kulyabov.jpg
create mode 100644 labs/lab11/presentation/presentation.md
create mode 100644 labs/lab11/report/Makefile
create mode 100644 labs/lab11/report/bib/cite.bib
create mode 100644 labs/lab11/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab11/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab11/report/report.md
delete mode 100644 package.json
create mode 100644 prepare
[izdzhakhangirov@fedora arch-pc]$ git push
Перечисление объектов: 22, готово.
Подсчет объектов: 100% (22/22), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (20/20), 310.95 КиБ | 2.25 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:izdzhakhangirov/study_2022-2023_arh-pc.git
   1e0896f..703bf7e master -> master
[izdzhakhangirov@fedora arch-pc]$

```

Самостоятельная работа

зашел на **github** в свой репозиторий, потом перешел в раздел labs, убедился что 11 папок на для наших лабораторных работ созданы.

izdzhakhangirov / study_2022-2023_arh-pc Public
generated from yamadharma/course-directory-student-template

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master study_2022-2023_arh-pc / labs / Go to file Add file ...

izdzhakhangirov feat(main): make course structure 703bf7e 1 hour ago History

..		
lab01	feat(main): make course structure	1 hour ago
lab02	feat(main): make course structure	1 hour ago
lab03	feat(main): make course structure	1 hour ago
lab04	feat(main): make course structure	1 hour ago
lab05	feat(main): make course structure	1 hour ago
lab06	feat(main): make course structure	1 hour ago
lab07	feat(main): make course structure	1 hour ago
lab08	feat(main): make course structure	1 hour ago
lab09	feat(main): make course structure	1 hour ago
lab10	feat(main): make course structure	1 hour ago
lab11	feat(main): make course structure	1 hour ago

izdzhakhangirov / study_2022-2023_arh-pc Public
generated from yamadharma/course-directory-student-template

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master study_2022-2023_arh-pc / labs / lab01 / report / Go to file Add file ...

izdzhakhangirov Add files via upload 03f489a now History

..		
bib	feat(main): make course structure	1 hour ago
image	feat(main): make course structure	1 hour ago
pandoc/csl	feat(main): make course structure	1 hour ago
Makefile	feat(main): make course structure	1 hour ago
report.md	feat(main): make course structure	1 hour ago
ЛО1-Джахангиров-ОТЧЕТ 1.pdf	Add files via upload	now

перешел по нужному пути до папки **report** и загрузил туда первую лабораторную работу, аналогично загрузил вторую и третью.

Вывод: Практически изучил идеологию средств контроля версий, приобрел практические навыки по работе с системой git.

