

# **Отчёта по лабораторной работе №2:**

**Первоначальная настройка git**

Кононов Алексей Сергеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>14</b>
<b>6</b>	<b>Выводы</b>	<b>19</b>
	<b>Список литературы</b>	<b>20</b>

## Список иллюстраций

4.1	Установка git . . . . .	7
4.2	Настройка git . . . . .	7
4.3	ssh . . . . .	8
4.4	pgp . . . . .	8
4.5	GitHub . . . . .	9
4.6	ключи git . . . . .	10
4.7	Авторизация . . . . .	10
4.8	Авторизация (сайт) . . . . .	11
4.9	Создание репозитория . . . . .	12
4.10	Настройка каталога часть 1 . . . . .	12
4.11	Настройка каталога часть 2 . . . . .	13
4.12	Настройка каталога часть 3 . . . . .	13

# 1 Цель работы

Целью данной работы заключается в изучении идеологии и применении средств контроля версий, а также освоить умения по работе с git.

## 2 Задание

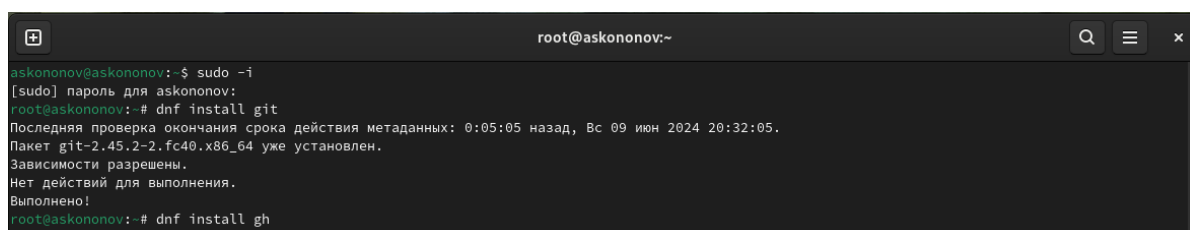
1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
4. Настроить подписи git.
5. Зарегистрироваться на Github.
6. Создать локальный каталог для выполнения заданий по предмету.

## 3 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.

## 4 Выполнение лабораторной работы

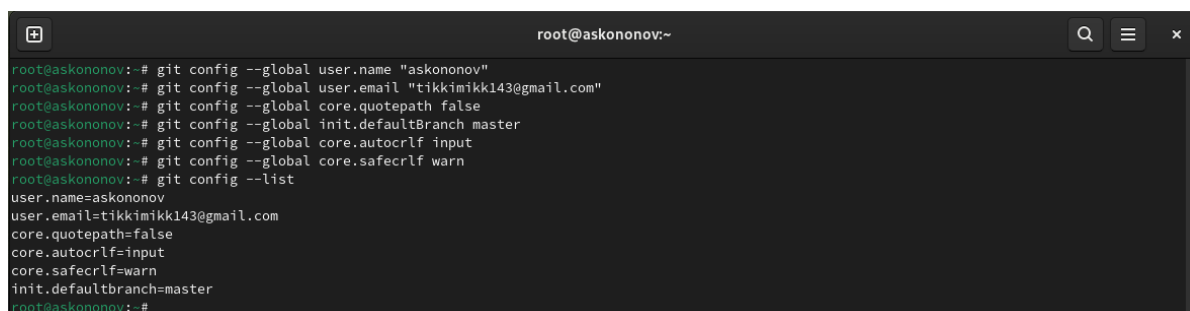
Установим git `dnf install git` и gh `dnf install gh` (рис. 4.1).



```
root@askononov:~  
askononov@askononov:~$ sudo -i  
[sudo] пароль для askononov:  
root@askononov:~# dnf install git  
Последняя проверка окончания срока действия метаданных: 0:05:05 назад, Вс 09 июн 2024 20:32:05.  
Пакет git-2.45.2-2.fc40.x86_64 уже установлен.  
Зависимости разрешены.  
Нет действий для выполнения.  
Выполнено!  
root@askononov:~# dnf install gh
```

Рис. 4.1: Установка git

Сделаем базовые настройки git. Для этого зададим имя и почту владельца репозитория (`git config --global user.name "Name Surname"` и `git config --global user.email "work@mail)`, настроим utf-8 в выводе сообщений git (`git config --global core.quotepath false`), зададим имя начальной ветки, которую будем называть ее master (`git config --global init.defaultBranch master`) и установим пару параметров (`git config --global core.autocrlf input` и `git config --global core.safecrlf warn`). И проверим изменения с помощью команды `git config --list` (рис. 4.3).



```
root@askononov:~  
root@askononov:~# git config --global user.name "askononov"  
root@askononov:~# git config --global user.email "tikkimikk143@gmail.com"  
root@askononov:~# git config --global core.quotepath false  
root@askononov:~# git config --global init.defaultBranch master  
root@askononov:~# git config --global core.autocrlf input  
root@askononov:~# git config --global core.safecrlf warn  
root@askononov:~# git config --list  
user.name=askononov  
user.email=tikkimikk143@gmail.com  
core.quotepath=false  
core.autocrlf=input  
core.safecrlf=warn  
init.defaultbranch=master  
root@askononov:~#
```

Рис. 4.2: Настройка git

Создадим ключ **ssh** **ssh-keygen -t rsa -b 4096** (рис. 4.3).

```
askononov@askononov:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/askononov/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/askononov/.ssh/id_rsa
Your public key has been saved in /home/askononov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:d+SJXSjZ53zp1kKnmRUP3E4TuqBH75vQ7Mo801fITc askononov@askononov
The key's randomart image is:
+----[RSA 4096]-----+
|
| o o...|
| oo+.oo|
| ++o+.+=|
| Soo+=E+.*|
| . ...+o.+B.|
| o ..o o.=o.|
| o o.+..|
| oo+o..|
+----[SHA256]-----+
askononov@askononov:~$
```

Рис. 4.3: ssh

Сгенерируем ключ **pgp** с помощью команды **gpg --full-generate-key**. (рис. 4.4).

```
askononov@askononov:~$ gpg --full-generate-key
gpg (GnuPG) 2.4.4; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/askononov/.gnupg'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.
Ваше полное имя: askononov
Адрес электронной почты: tikkimikk143@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "askononov <tikkimikk143@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? o
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/askononov/.gnupg/trustdb.gpg: создана таблица доверия
```

Рис. 4.4: pgp

Буду использовать уже существующую учетную запись (рис. 4.5).



```
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 344b374] feat(main): make course structure
361 files changed, 98413 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.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/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 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.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/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
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
```

Рис. 4.5: GitHub

Выводим список ключей **gpg --list-secret-keys --keyid-format LONG** и копируем опечаток приватного ключа **gpg --armor --export | xclip -sel clip**. (рис. 4.6) Ключи добавляем на GitHub (**settings -> SSH and GPG keys -> New GPG key**), а также настроим автоматические подписи коммитов git с помощью команд: **git config --global user.signingkey** , **git config --global commit.gpgsign true**, **git config --global gpg.program \$(which gpg2)**.

```
askononov@askononov:~$ gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/12690C93E64A795A 2024-06-09 [SC]
      BB11C5B75D5FE377BCB3EDF312690C93E64A795A
uid   [ абсолютно ] askononov <tikkimikk143@gmail.com>
ssb   rsa4096/9E35FE98F083A851 2024-06-09 [E]

askononov@askononov:~$ gpg --armor --export 2690C93E64A795A | xclip -sel clip
gpg: Внимание: нечего экспортировать
askononov@askononov:~$ gpg --armor --export 12690C93E64A795A | xclip -sel clip
askononov@askononov:~$ git config --global user.signingkey 12690C93E64A795A
askononov@askononov:~$ git config --global commit.gpgsign true
askononov@askononov:~$ git config --global gpg.program $(which gpg2)
askononov@askononov:~$
```

Рис. 4.6: ключи git

Настраиваем `gh` **gh auth login** и авторизуемся через браузер (рис. 4.7), (рис. 4.8).

```
askononov@askononov:~$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: F69C-652D
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as bezponiki143
askononov@askononov:~$
```

Рис. 4.7: Авторизация

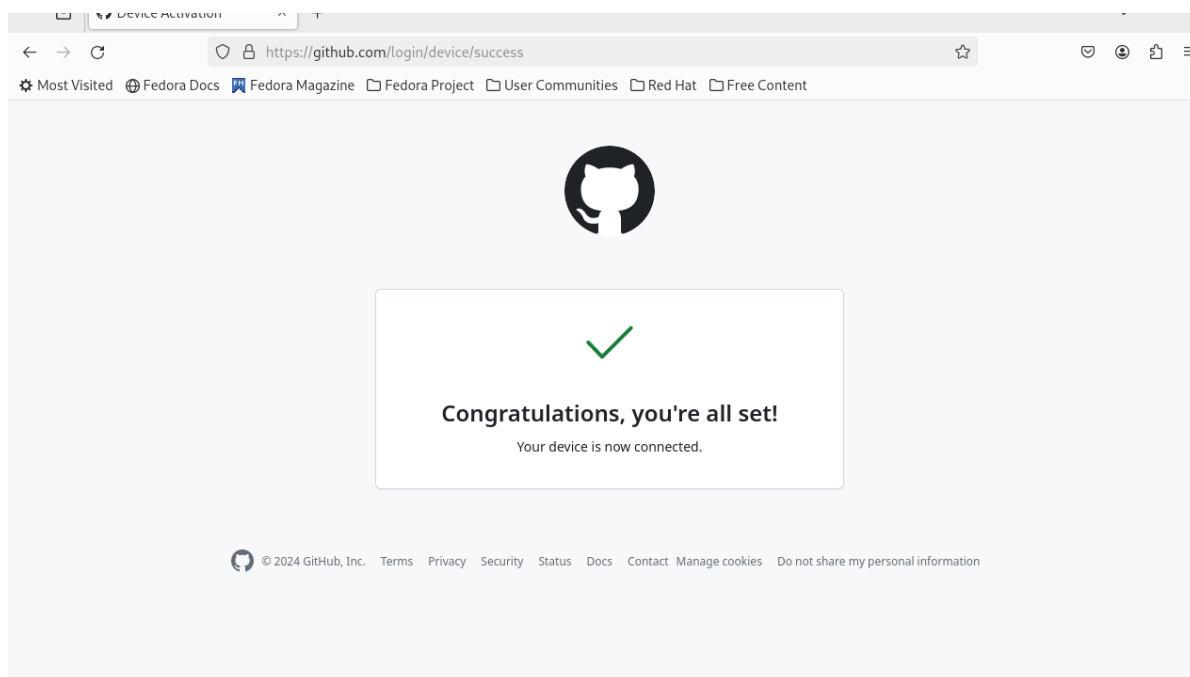


Рис. 4.8: Авторизация (сайт)

Создадим репозиторий для 2023–2024 учебного года:

```
mkdir -p ~/work/study/2023-2024/“Операционные системы”
```

```
cd ~/work/study/2023-2024/“Операционные системы”
```

```
gh repo create study_2023-2024_os-intro --template=yamadharma/course-  
directory-student-template --public
```

```
git clone --recursive git@github.com:/study_2023-2024_os-intro.git os-intro
```

(рис. 4.9).

```
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro
askononov@askononov:~$ mkdir -p ~/work/study/2023-2024/"Операционные системы"
askononov@askononov:~$ cd ~/work/study/2023-2024/"Операционные системы"
askononov@askononov:~/work/study/2023-2024/Операционные системы$ gh repo create study_2023-2024_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository bezponiki143/study_2023-2024_os-intro on GitHub
https://github.com/bezponiki143/study_2023-2024_os-intro
askononov@askononov:~/work/study/2023-2024/Операционные системы$ git clone --recursive git@github.com:study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvvV6TuJJhbpZisF/zLDA0zPMSvHdKr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
ERROR: Repository not found.
fatal: Не удалось прочесть из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
askononov@askononov:~/work/study/2023-2024/Операционные системы$ git clone --recursive git@github.com:bezponiki143/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (32/32), 18.60 КиБ | 405.00 КиБ/с, готово.
Определение изменений: 100% (1/1), 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 «/home/askononov/work/study/2023-2024/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Получение объектов: 100% (95/95), 96.99 КиБ | 1013.00 КиБ/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/home/askononov/work/study/2023-2024/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Получение объектов: 100% (126/126), 335.80 КиБ | 1.70 МБ/с, готово.
Определение изменений: 100% (52/52), готово.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8843ff1ca72c60a304f24c'
Submodule path 'template/report': checked out '7c31ab8e5dfa8cdb2d67caeb8a19ef8028ced88e'
askononov@askononov:~/work/study/2023-2024/Операционные системы$ cd os-intro/
```

Рис. 4.9: Создание репозитория

Настроим каталог, удалим лишние файлы **rm package.json**, также создадим еще необходимые каталоги **echo os-intro > COURSE** и **make prepare**, и отправим на сервер **git add .**, **git commit -am 'feat(main): make course structure'**, **git push** (рис. 4.10), (рис. 4.11), (рис. 4.12).

```
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro
askononov@askononov:~/work/study/2023-2024/Операционные системы$ cd os-intro/
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ rm package.json
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ echo os-intro > COURSE
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ make prepare
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ ls
CHANGELOG.md  config  COURSE  labs  LICENSE  Makefile  prepare  presentation  project-personal  README.en.md  README.git-flow.md  README.md  template
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ git add
Ничего не проиндексировано.
hint: Maybe you wanted to say 'git add .'?
hint: Disable this message with "git config advice.addEmptyPathsSpec false"
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ git add .
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
```

Рис. 4.10: Настройка каталога часть 1

```
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 344b374] feat(main): make course structure
361 files changed, 98413 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.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/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 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.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/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
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
```

Рис. 4.11: Настройка каталога часть 2

```
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.23 Киб | 2.39 Миб/с, готово.
Total 38 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:bezponiki143/study_2023-2024_os-intro.git
6853bb0..344b374 master -> master
askononov@askononov:~/work/study/2023-2024/Операционные системы/os-intro$
```

Рис. 4.12: Настройка каталога часть 3

## 5 Контрольные вопросы

1. **Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?**

Системы контроля версий (Version Control System, VCS) — это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени. В свете усложнения сред разработки они помогают командам разработчиков работать быстрее и эффективнее.

2. **Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.**

**Хранилище (репозиторий)** — это централизованное место, где хранятся файлы и история изменений проекта. Оно содержит все версии файлов, метаданные и историю коммитов.

**Commit (фиксация)** — это действие по сохранению изменений в системе контроля версий. При коммите разработчик предоставляет описание внесенных изменений, и эти изменения фиксируются в репозитории.

**История (history)** — это список всех коммитов и изменений, связанных с проектом. История содержит информацию о том, кто, когда и какие изменения вносил, и позволяет отслеживать всю историю проекта.

**Рабочая копия (working copy)** — это локальная копия файлов из репозитория, с которой работает разработчик. Рабочая копия содержит текущую версию проекта, и разработчик вносит изменения в нее перед их фиксацией (коммитом) в репозиторий.

**3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.**

**Централизованная система контроля версий (Centralized Version Control System, CVS)** история версий хранится на удалённом сервере, а рабочий код на нескольких компьютерах. Компьютеры связаны с одним сервером. Примером может послужить SVN (Subversion).

**Децентрализованные системы контроля версий (Distributed Version Control System, DVCS)** рабочий код хранится на нескольких компьютерах, а история всех версий хранится как на удалённом сервере, так и на каждом из этих компьютеров. Все компьютеры связаны с сервером, но ещё дополнительно связаны между собой. Пример Git.

**4. Опишите действия с VCS при единоличной работе с хранилищем.**

При единоличной работе с хранилищем VCS основными действиями будут:

- **Инициализация репозитория:** создание нового проекта или клонирование существующего репозитория из удаленного источника (например, GitHub).
- **Добавление файлов:** добавление новых файлов в репозиторий или изменение уже существующих файлов.
- **Фиксация изменений:** коммит изменений в репозиторий с указанием описания изменений.
- **Просмотр истории изменений:** просмотр и анализ всех предыдущих коммитов, внесенных в репозиторий.
- **Ветвление:** создание отдельных веток для разработки новых функций или исправлений багов.
- **Объединение изменений:** слияние веток и консолидация изменений в основной ветке разработки.

- **Удаление файлов:** удаление ненужных файлов из репозитория.
- **Удаленная работа:** отправка изменений на удаленный сервер и получение изменений из удаленного репозитория.

## 5. Опишите порядок работы с общим хранилищем VCS.

- **Создание репозитория:** сначала необходимо создать репозиторий на сервере или в облаке, где будет храниться общее хранилище файлов.
- **Клонирование репозитория:** разработчики должны клонировать репозиторий себе на локальную машину, чтобы иметь доступ к файлам и иметь возможность вносить изменения.
- **Работа с файлами:** разработчики могут вносить изменения в файлы на локальной машине, создавать новые файлы, удалять или редактировать существующие.
- **Подготовка к коммиту:** перед сохранением изменений в репозиторий, необходимо подготовить их к коммиту, добавив их в “индекс” при помощи команды `git add`.
- **Коммит изменений:** после подготовки изменений, разработчики должны сделать коммит, сохраняя все внесенные изменения в историю репозитория при помощи команды `git commit`.
- **Пуш изменений:** после коммита, изменения могут быть отправлены в общее хранилище с помощью команды `git push`, что позволит другим разработчикам видеть и получать эти изменения.
- **Обновление локального репозитория:** разработчики могут получить последние изменения из общего хранилища с помощью команды `git pull`, чтобы обновить свою локальную версию репозитория.



**6. Каковы основные задачи, решаемые инструментальным средством git?**

- Хранить информацию и всех изменениях в коде, с возможностью в любой момент перейти к любому из них.
- Обеспечение удобства командной работы над кодом.

**7. Назовите и дайте краткую характеристику командам git.**

- **git init**: инициализация нового репозитория
- **git add**: добавление изменений в индекс
- **git commit**: сохранение изменений в репозитории
- **git push**: отправка изменений в удаленный репозиторий
- **git pull**: получение изменений из удаленного репозитория
- **git branch**: создание, удаление и просмотр веток
- **git merge**: объединение изменений из другой ветки
- **git checkout**: переключение между ветками

**8. Приведите примеры использования при работе с локальным и удалённым репозиториями.**

**Локальный репозиторий:** создание нового проекта с помощью `git init`, добавление новых файлов с помощью `git add`, сохранение изменений в репозитории с помощью `git commit`.

**Удаленный репозиторий:** отправка изменений из локального репозитория на удаленный с помощью `git push`, получение изменений из удаленного репозитория с помощью `git pull`.

**9. Что такое и зачем могут быть нужны ветви (branches)?**

Ветка - параллельный участок истории в одном хранилище, между ветками возможно слияние. Обычно используются для создания новых функций или новых версий приложения.

## 10. Как и зачем можно игнорировать некоторые файлы при commit?

Можно просто не добавлять их в индекс командой `git add` или создать файл `.gitignore`, в котором перечислить все файлы и папки которые требуется игнорировать. Может понадобиться игнорировать настройки IDE, или бинарные файлы. Так как они зависят от конкретного разработчика и платформы, и в репозитории они могут быть лишними.

## 6 Выводы

В этой лабораторной работе мы научились настраивать git генерировать ключи для ssh и gpg, а также взаимодействовать с удаленными репозиториями, создавая свои из шаблонов и загружая изменения в локальном репозитории на удаленный.

## **Список литературы**