

Лабораторная работа №2

Markdown

РЫТОВ Алексей Константинович

Список иллюстраций

3.1	Рис. 1: Вход в учётную запись github	5
3.2	Рис. 2: Заполнение основных данных на github	6
3.3	Рис. 3: Установка git-flow	6
3.4	Рис. 4: Установка git-flow	7
3.5	Рис. 5: Настройка git	7
3.6	Рис. 6.2: Создание ssh ключа по алгоритму rsa размером 4096 бит	8
3.7	Рис. 7.2: Создание ключа gpg	9
3.8	Рис. 8.2: Добавление gpg ключа на guthub	10
3.9	Настроили автоматические подписи. (рис. 9)	10
3.10	Рис. 10.2: Подтверждение регистрации	11
3.11	Рис. 11.3: Создание репозитория по шаблону на github	12
3.12	Рис. 12.2: Вывод команд git commit и git push	12

1 Цель работы

– Изучить идеологию и применение средств контроля версий. – Освоить умения по работе с git.

2 Задание

- Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
 - В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)
-

3 Выполнение лабораторной работы

1. Создали учётную запись на <https://github.com>. (я же просто вошел в существующий аккаунт) (рис. 1)

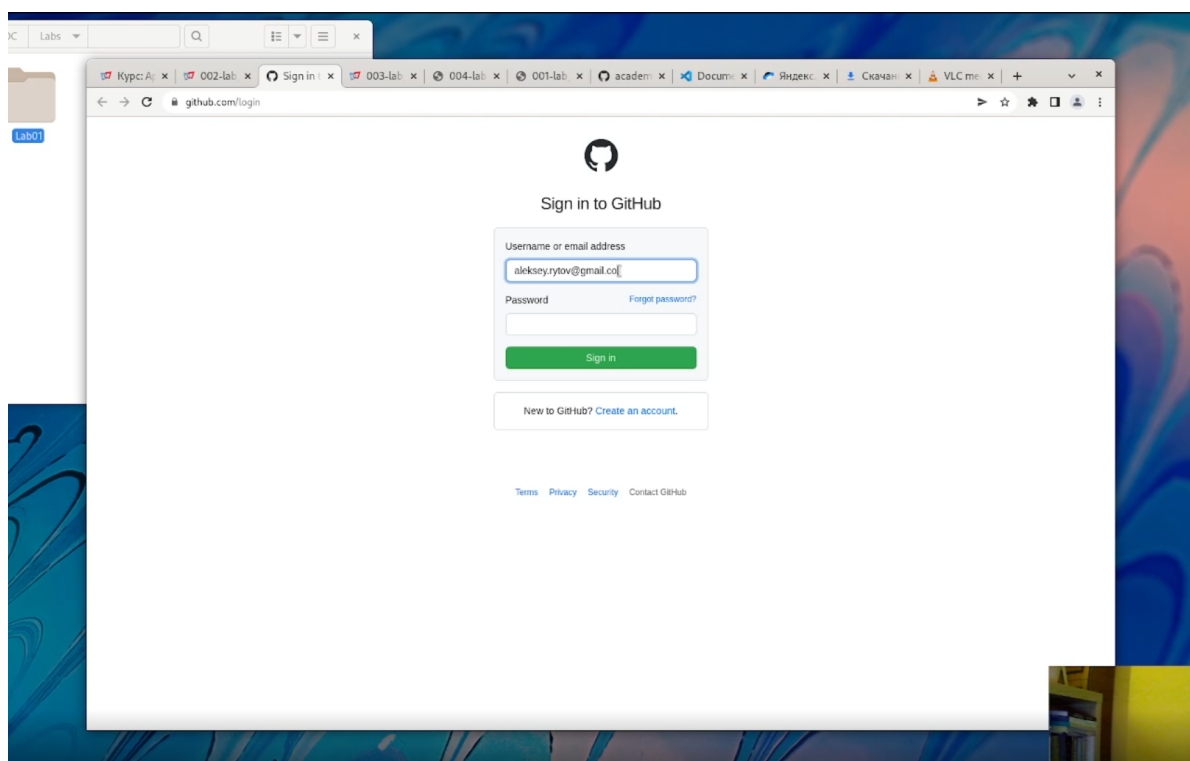


Рис. 3.1: Рис. 1: Вход в учётную запись github

-
2. Заполнили основные данные на <https://github.com>. (рис. 2)

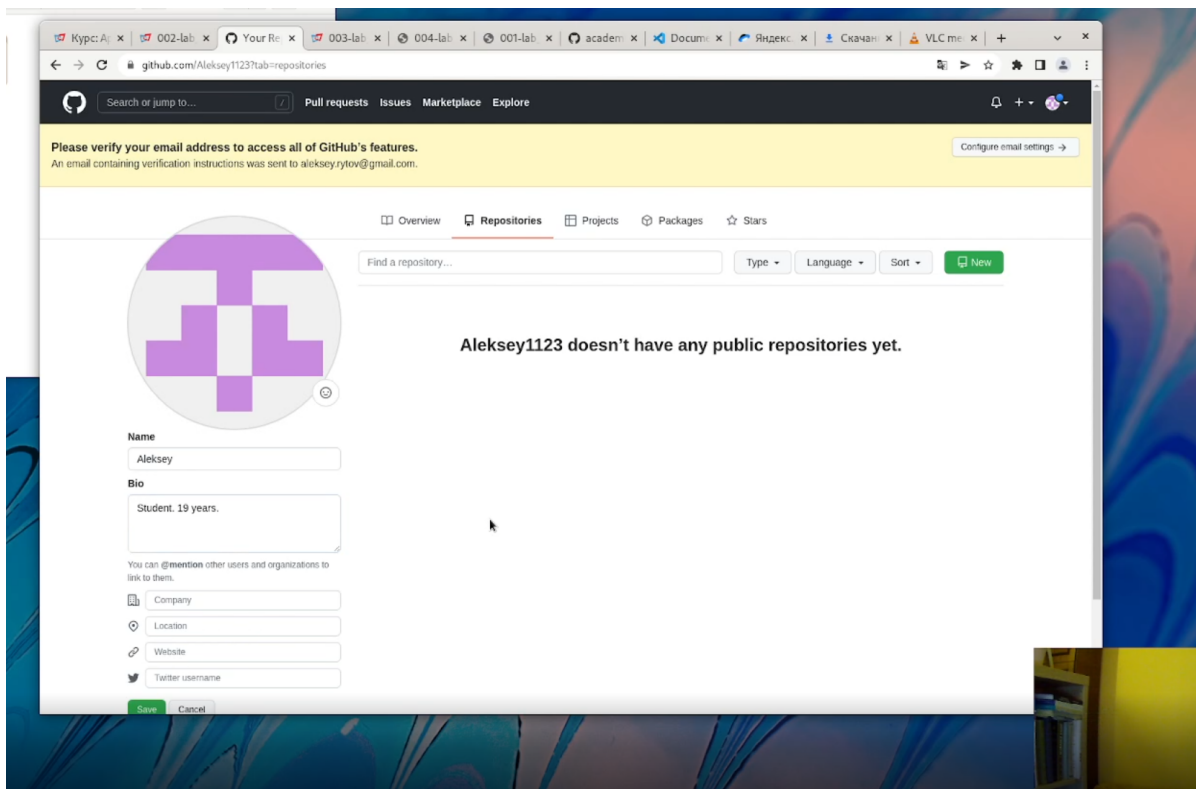


Рис. 3.2: Рис. 2: Заполнение основных данных на github

3. Установили git-flow в Fedora Linux. (рис. 3)



Рис. 3.3: Рис. 3: Установка git-flow

4. Установили gh в Fedora Linux. (рис. 4)

```
[akrihtov@fedora tmp]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 0:32:14 назад, Ср 20 апр 2022 20:41:03.
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий
=====
Установка:
gh                    x86_64                2.7.0-1.fc35          updates
Результат транзакции
=====
Установка 1 Пакет
Объем загрузки: 6.8 М
Объем изменений: 32 М
Продолжить? [д/н]:
```

Рис. 3.4: Рис. 4: Установка git-flow

5. Произвели базовую настройку git. (рис. 5)

```
[akrihtov@fedora tmp]$ git config --global user.name "Aleksey Rytov"
[akrihtov@fedora tmp]$ git config --global user.email "aleksey.rytov@gmail.com"
[akrihtov@fedora tmp]$ git config --global core.quotepath false
[akrihtov@fedora tmp]$ git config --global init.defaultBranch master
[akrihtov@fedora tmp]$ git config --global core.autocrlf input
[akrihtov@fedora tmp]$ git config --global core.safecrlf warn
```

Рис. 3.5: Рис. 5: Настройка git

6. Создали ключи ssh. (рис. 6.1 – 6.2)

```
[akrihtov@fedora tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/akrihtov/.ssh/id_rsa):
Created directory '/home/akrihtov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/akrihtov/.ssh/id_rsa
Your public key has been saved in /home/akrihtov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Nf6S2mXu/10PxEr4ABwXHXznVX+Fxe8SjicqAq0RUJw akrihtov@fedora
The key's randomart image is:
+----[RSA 4096]-----+
|  o.. . 0+.. +=|
| . E . o o o *|
| . o o . o=|
| . + o .. +|
| o S + .ooo |
| o . *oo+ .|
| + o.*o...|
| .. .o.= .+|
| .....o...+|
+----[SHA256]-----+
```

```
[akrihtov@fedora tmp]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/akrihtov/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/akrihtov/.ssh/id_ed25519
Your public key has been saved in /home/akrihtov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:EoFeVc7ruxfRAR+QB8CqJ4Y4C1WQ4UG6EJFTqtLV0wA akrihtov@fedora
The key's randomart image is:
+--[ED25519 256]--+
|o=E+ ....oo.+=. |
|o=.oo... o. ..o. |
|+..oo.o .o o.. |
|oo.... o. .. . |
|+... .+.S . . |
|o o . +o.. . |
|. o . o . . |
|. . . . . |
| . . . . . |
| . . . . . |
+-----[SHA256]-----+
[akrihtov@fedora tmp]$
```

Рис. 3.6: Рис. 6.2: Создание ssh ключа по алгоритму rsa размером 4096 бит

7. Создали ключ pgr. (рис. 7.1 – 7.2)

```
[akrihtov@fedora tmp]$ gpg --full-generate-key
gpg (GnuPG) 2.3.4; Copyright (C) 2021 Free Software Foundation, Inc.
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/akrihtov/.gnupg'
gpg: создан щит с ключами '/home/akrihtov/.gnupg/pubring.kbx'
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)
```



```

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: Aleksey
Адрес электронной почты: aleksey.rytov@gmail.com
Примечание: z
Вы выбрали следующий идентификатор пользователя:
"Aleksey (z) <aleksey.rytov@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 (печатать
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 (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/akrihtov/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/akrihtov/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/akrihtov/.gnupg/openpgp-revocs.d/4C1ED3661EC3E9B918CA43E8D78953DEA3F20495.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2022-04-20 [SC]
    4C1ED3661EC3E9B918CA43E8D78953DEA3F20495
uid                          Aleksey (z) <aleksey.rytov@gmail.com>
sub  rsa4096 2022-04-20 [E]

```

Рис. 3.7: Рис. 7.2: Создание ключа gpg

8. Добавили pgr ключ в github. (рис. 7.1 – 7.2)

```

[akrihtov@fedora tmp]$ gpg --list-secret-keys --keyid-format LONG
/home/akrihtov/.gnupg/pubring.kbx
-----
sec  rsa4096/D78953DEA3F20495 2022-04-20 [SC]
    4C1ED3661EC3E9B918CA43E8D78953DEA3F20495
uid          [ абсолютно ] Aleksey (z) <aleksey.rytov@gmail.com>
ssb  rsa4096/2DFEC30BCCF442D8 2022-04-20 [E]

[akrihtov@fedora tmp]$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[akrihtov@fedora tmp]$ gpg --armor --export <PGP Fingerprint> > xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «>»
[akrihtov@fedora tmp]$ gpg --armor --export <PGP Fingerprint> > mekey.asc
bash: синтаксическая ошибка рядом с неожиданным маркером «>»
[akrihtov@fedora tmp]$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[akrihtov@fedora tmp]$ gpg --armor --export <D78953DEA3F20495> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[akrihtov@fedora tmp]$ gpg --armor --export <D78953DEA3F20495> | xclip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[akrihtov@fedora tmp]$ gpg --armor --export D78953DEA3F20495 | xclip -sel clip
bash: xclip: command not found...
Install package 'xclip' to provide command 'xclip'? [N/y] y

* Waiting in queue...
* Loading list of packages....

```

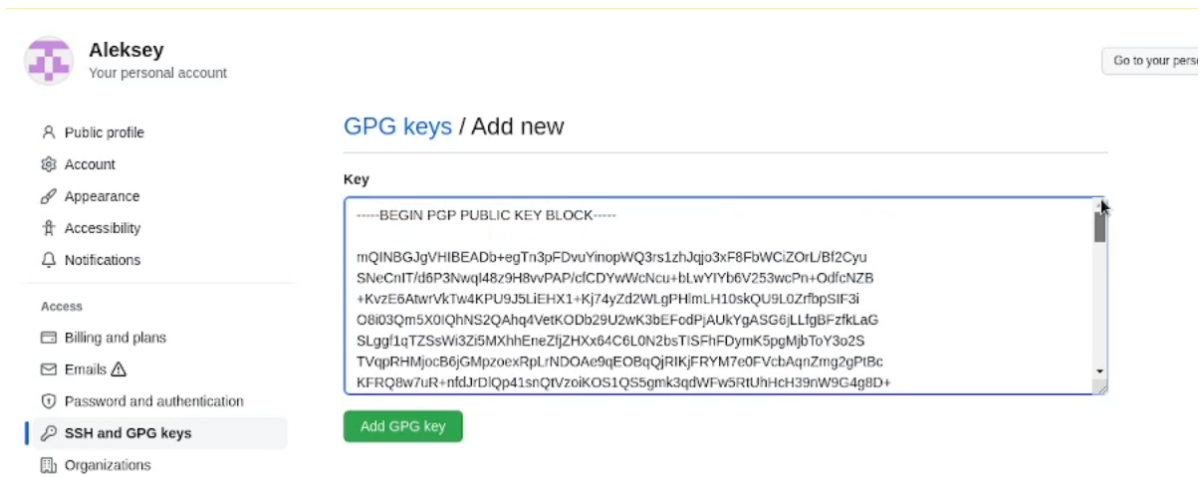


Рис. 3.8: Рис. 8.2: Добавление gpg ключа на guthub

9. Настроили автоматические подписи. (рис. 9)

```
[akrihtov@fedora tmp]$ git config --global user.signingkey D78953DEA3F20495
[akrihtov@fedora tmp]$ git config --global commit.gpgsign true
[akrihtov@fedora tmp]$ git config --global gpg.program $(which gpg2)
```

Рис. 3.9: Настроили автоматические подписи. (рис. 9)

10. Настроили gh. (рис. 10.1 - 10.2)

```
[akrihtov@fedora tmp]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? 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: 43E8-D906
Press Enter to open github.com in your browser...
```

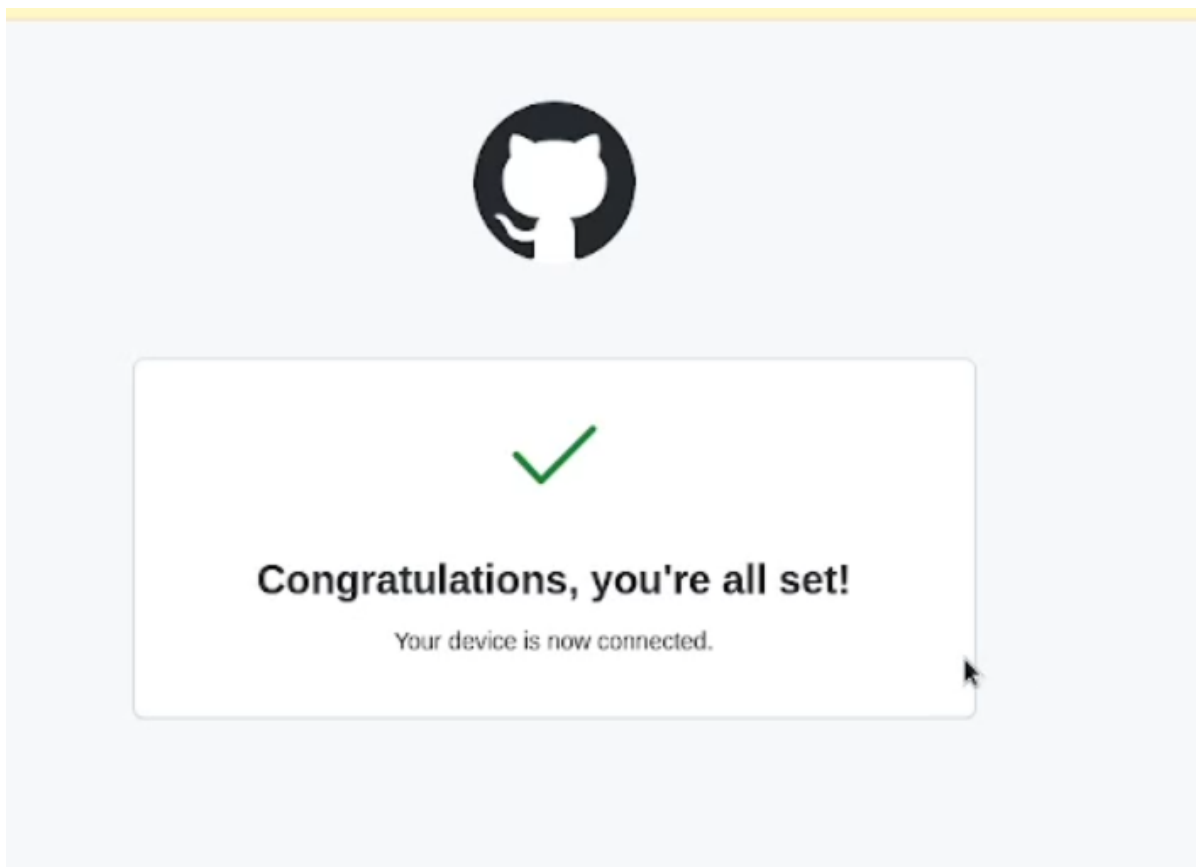


Рис. 3.10: Рис. 10.2: Подтверждение регистрации

11. Создали репозиторий курса по шаблону. (рис. 11.1 – 11.3)

```
[akrihtov@fedora ~]$ mkdir "Операционные системы"
[akrihtov@fedora ~]$ cd "Операционные системы"
[akrihtov@fedora Операционные системы]$ gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository Aleksey1123/study_2021-2022_os-intro on GitHub
[akrihtov@fedora Операционные системы]$
```

```
[akrihtov@fedora Операционные системы]$ git clone --recursive https://github.com/Aleksey1123/study_2021-2022_os-intro os-intro
fatal: целевой путь «os-intro» уже существует и не является пустым каталогом.
[akrihtov@fedora Операционные системы]$ git clone --recursive https://github.com/Aleksey1123/study_2021-2022_os-intro os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 КиБ | 297.00 КиБ/с, готово.
Определение изменений: 100% (2/2), 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/akrihtov/study/2021-2022/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
```

Рис. 3.11: Рис. 11.3: Создание репозитория по шаблону на github

12. Настроили каталога курса. (рис. 12.1 – 12.)

```
[akrihtov@fedora Операционные системы]$ cd os-intro
[akrihtov@fedora os-intro]$ rm package.json
[akrihtov@fedora os-intro]$ make COURSE=os-intro
[akrihtov@fedora os-intro]$ git add .
[akrihtov@fedora os-intro]$ git commit -am 'feat(main): make course structure'
```

```
create mode 100644 project-personal/stage4/report/bib/cite.bib
create mode 100644 project-personal/stage4/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage4/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage4/report/report.md
create mode 100644 project-personal/stage5/presentation/Makefile
create mode 100644 project-personal/stage5/presentation/presentation.md
create mode 100644 project-personal/stage5/report/Makefile
create mode 100644 project-personal/stage5/report/bib/cite.bib
create mode 100644 project-personal/stage5/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage5/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage5/report/report.md
create mode 100644 project-personal/stage6/presentation/Makefile
create mode 100644 project-personal/stage6/presentation/presentation.md
create mode 100644 project-personal/stage6/report/Makefile
create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage6/report/report.md
create mode 100644 structure
[akrihtov@fedora os-intro]$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (19/19), 266.53 КиБ | 14.81 МиБ/с, готово.
Всего 19 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Aleksey1123/study_2021-2022_os-intro
f2d6fe9..4067e67 master -> master
[akrihtov@fedora os-intro]$
```

Рис. 3.12: Рис. 12.2: Вывод команд git commit и git push

4 Выводы

Мы изучили идеологию и применение средств контроля версий, а также освоили умения по работе с git.

5 Ответы на контрольные вопросы:

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

Система контроля версий(VCS) – это хранилище, где находится код. Она предназначена для хранения нескольких версий файлов, позволяет совмещать изменения, произведённые разными участниками проекта, а также производить откат к любой более ранней версии проекта.

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

Хранилище – место, где хранится вся служебная информация, история изменения документов. Commit – процесс создания новой версии. Рабочая копия – последняя рабочая версия проекта, загруженная из хранилища.

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

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы является совокупностью решений отдельных узлов.

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

- 1) Произвожу блокировку(Lock) последней версии проекта.
- 2) Выгружаю проект с VCS. (Check-out)
- 3) Произвожу работу над проектом.
- 4) Сохраняю и выгружаю проект на VCS. (Check-in)
- 5) Произвожу разблокировку(Unlock) проекта.

5. Опишите порядок работы с общим хранилищем VCS.

- 1) Произвожу блокировку(Lock) последней версии проекта.
- 2) Выгружаю проект с VCS. (Check-out)
- 3) Произвожу работу над проектом.
- 4) Сохраняю и выгружаю проект на VCS. (Check-in)
- 5) Произвожу разблокировку(Unlock) проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

Git предназначен для хранения нескольких версий проекта, позволяет совмещать изменения, произведённые разными участниками проекта.

7. Назовите и дайте краткую характеристику командам git.

- создание основного дерева репозитория: `git init`
- получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull`
- отправка всех произведённых изменений локального дерева в центральный репозиторий: `git push`
- просмотр списка изменённых файлов в текущей директории: `git status` – просмотр текущих изменения:
`git diff`

- сохранение текущих изменений:
- добавить все изменённые и/или созданные файлы и/или каталоги: `git add`
- добавить конкретные изменённые и/или созданные файлы и/или каталоги:
`git add имена_файлов`
- удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): `git rm имена_файлов`
- сохранение добавленных изменений: – сохранить все добавленные изменения и все изменённые файлы: `1 git commit -am 'Описание коммита'`
- сохранить добавленные изменения с внесением комментария через встроенный редактор: Кулябов Д. С. и др. Операционные системы 23 `git commit`
- создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки`
- переключение на некоторую ветку: `git checkout имя_ветки` (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
- отправка изменений конкретной ветки в центральный репозиторий: `git push origin имя_ветки`
- слияние ветки с текущим деревом: `git merge --no-ff имя_ветки`
- удаление ветки:
- удаление локальной уже слитой с основным деревом ветки: `git branch -d имя_ветки`
- принудительное удаление локальной ветки: `git branch -D имя_ветки`
- удаление ветки с центрального репозитория: `git push origin :имя_ветки`

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

Для инициализации локального репозитория, расположенного, например, в каталоге `~/tutorial`, необходимо ввести в командной строке:

1 `cd`

2 `mkdir tutorial`

3 cd tutorial

4 git init

команда status для просмотра изменений в рабочем каталоге, сделанных с момента последней ревизии: git status

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

Ветка – это просто «скользящий» указатель на один из коммитов. Когда мы создаём новые коммиты, указатель ветки автоматически сдвигается вперёд, к вновь созданному коммиту.

Ветки используются для разработки одной части функционала изолированно от других. Каждая ветка представляет собой отдельную копию кода проекта. Ветки позволяют одновременно работать над разными версиями проекта.

Ветвление («ветка», branch) — один из параллельных участков истории в одном хранилище, исходящих из одной версии (точки ветвления). Ветки нужны для того, чтобы программисты могли вести совместную работу над проектом и не мешать друг другу при этом.

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

Игнорируемые файлы обычно представляют собой файлы, специфичные для платформы, или автоматически созданные из сборочных систем. Временно игнорировать изменения в файле можно командой: git update-index —assume-unchanged