

Лабораторная работа 3

Отчет по лабораторной работе 3

Куркина Евгения Вячеславовна

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
Список литературы	20

Список иллюстраций

3.1 Регистрация на Github	7
3.2 Базовая настройка git	8
3.3 Создание 1-го ключа SSH	8
3.4 Создание 2-го ключа SSH	8
3.5 Создание ключа PGP	9
3.6 Создание репозитория	10
3.7 Копирование ссылки репозитория	10
3.8 Вызов ключа	10
3.9 Копирование его в нужную ячейку	11
3.10 След приватного ключа	12
3.11 Созданный ключ	12
3.12 Создание основного дерева репозитория и получения обновлений	13
3.13 Отправление изменений в репозиторий	13
3.14 Настройка автоматических подписей	13
3.15 Создание шаблона рабочего пространства	14
3.16 Копирование репозитория	14
3.17 Вставка ссылки в мой github	15
3.18 Появившаяся папка	15
3.19 Соединение каталогов	16
3.20 Отправка на сервер	16

Список таблиц

1 Цель работы

Здесь приводится формулировка цели лабораторной работы. Формулировки цели для каждой лабораторной работы приведены в методических указаниях.

Цель лабораторной работы —Научиться оформлять отчёты с помощью легковесного языка разметки Markdown .

2 Задание

1.Сделать отчет по второй лабораторной работе в формате Markdown . 2.В качестве отчета предоставить отчеты в форматах: pdf, docx, md.

3 Выполнение лабораторной работы

1). Зарегистрировалась на Github (рис.3.1).

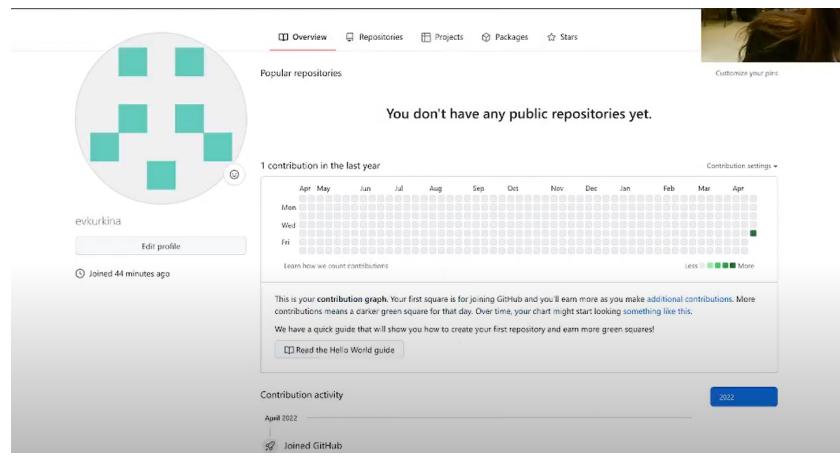


Рис. 3.1: Регистрация на Github

2). Провела Базовую настройку git с помощью консоли и команд:

- 2.1) git config –global user.name “Name Surname”- задаем имя владельца репозитория(Evgenya Kurkina)
- 2.2)git config–global user.email “work@mail”- задаем email (evgeshakurkina@yandex.ru)
- 2.3) git config –global core.quotepath false -Настроила utf-8 в воде сообщений git.
- 2.4) git config –global init.defaultBranch master- задала имя начальной ветки(master)
- 2.5) git config –global core.autocrlf input- задала параметр autocrlf
- 2.6) git config –global core.safecrlf warn-задала параметр safecrlf

(рис.3.2)

```

Файл Правка Вид Закладки Настройка Справка
Новая вкладка Разделить окно по вертикали Разделить окно по горизонтали Новая
evkirkina@dk6n54 ~ $ git config --global user.name EvgenyaKurkina
evkirkina@dk6n54 ~ $ git config --global user.email evgeshakurkina@yandex.ru
evkirkina@dk6n54 ~ $ git config --global core.quotePath false
evkirkina@dk6n54 ~ $ git config --global init.defaultBranch master
evkirkina@dk6n54 ~ $ git config --global core.autocrlf input
evkirkina@dk6n54 ~ $ git config --global core.safecrlf warn
evkirkina@dk6n54 ~ $

```

Рис. 3.2: Базовая настройка git

3).Создала ключи SSH: 3.1)Командой - ssh-keygen -t rsa -b 4096 -по алгоритму rsa с ключем размером 4096 бит (рис.3.3)

```

evkirkina@dk6n54 ~ $ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/e/v/evkirkina/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/v/evkirkina/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/v/evkirkina/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:MP7NTRslHVh1KXLRedh8thhwYzFSe55vbZfvnDlE20I evkirkina@dk6n54
The key's randomart image is:
+---[RSA 4096]---+
| . . . |
| . . . oo |
| o . . =Eo. |
| S .ooBO.B |
| o***=0* |
| +0000 |
| ..0*** |
| *.* |
+---[SHA256]---+

```

Рис. 3.3: Создание 1-го ключа SSH

3.2) Командой-ssh-keygen -t ed25519- по алгоритму ed25519 (рис.3.4)

```

evkirkina@dk6n54 ~ $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/e/v/evkirkina/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/v/evkirkina/.ssh/id_ed25519
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/e/v/evkirkina/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:lsppaqI2k0pqyhNSXifvUJAc4XwJ/3jlTY786glcfQ evkirkina@dk6n54
The key's randomart image is:
+---[ED25519 256]---+
| .o+. |
| o+.. |
| oo+ |
| .+ + . |
| o . S E = |
| +.. . . = * o |
| |=o.. . . o = |
| |+B... . .o o = |
| |0.o . .o . . |
+---[SHA256]---+
evkirkina@dk6n54 ~ $

```

Рис. 3.4: Создание 2-го ключа SSH

4).Создание ключа PGP: 4.1) Командой-gpg –full-generate-key – сгенерировала ключ 4.2) Поочередно выбираем необходимые параметры: – тип RSA and RSA;

- размер 4096; – выберите срок действия; значение по умолчанию— 0 (срок действия не истекает никогда). – GPG запросит личную информацию, которая сохранится в ключе:
 - Имя (не менее 5 символов)(Evgenya)
 - Адрес электронной почты. (evgeshakurkina@yandex.ru)
- 4.3) Нажимаем “о” для принятия (рис.3.5)

```

evkurkina@dk6n54 ~ $ gpg --full-generate-key
bash: gpg: команда не найдена
evkurkina@dk6n54 ~ $ gpg --full-generate-key
gpg (GnuPG) 2.2.33; 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.

Выберите тип ключа:
(1) RSA и RSA (по умолчанию)
(2) DSA и Elgamal
(3) DSA (только для подписи)
(4) RSA (только для подписи)
(14) Иммигрирующийся на карте ключ
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) <n>y
недопустимое значение
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: Evgenya
Адрес электронной почты: evgeshakurkina@yandex.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Evgenya <evgeshakurkina@yandex.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0

```

Рис. 3.5: Создание ключа PGP

- 5).Создала репозиторий с названием “Lab2” (рис.6) и скопировала его ссылку (рис.3.7)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * evkurnina **Repository name *** Lab2

Great repository names are Lab2 is available. durable. Need inspiration? How about fuzzy-doodle?

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.

Initialize this repository with:

Рис. 3.6: Создание репозитория

Owner * evkurnina Repository name * / >/study_2021-2022_os-intro.g

Great repository names are Your new repository will be created as -study_2021-2022_os-intro.g sh-spork?

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 yamedharma/course-directory-student-template and not just master.

Create repository from template

Рис. 3.7: Копирование ссылки репозитория

6).Добавление созданных ключей (в пункте 3 и 4) 6.1) Добавление SSH ключа на Github: - Командой cat ~/.ssh/id_rsa.pub | xclip -sel clip – вызвали ключ, затем скопировали его и после вставили в нужное поле для ключа на github (рис.3.8)(рис.3.9)

```
evkurnina@dk6n54: ~ cat /afs/dk.sci.pfu.edu.ru/home/e/v/evkurnina/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQDQABAAQACQDSEQu0eTmHBae5wUzUrR4Qx6EQqaTQ00tAHttNAjIEPaeEEyseqXG8tw45Psu+oSNKivrs+9/Z
v5/04gevyPb0pw69LEmriy17R0+YExm8pAcSzSLWqdg1+uF7K0e7aJU3Bqva+R+Xmdg850hqt0vg41/CzSYLqamq2YJa+eBlrcBM/pg+zsb8nk20TMpuwJ
fHC675eVxBNOKgfMTHnMaADlxGBsxakke6NoaPhL4dfZl5Iy2wY8YkmEGsOV/RNbJqTca7RZhH4m1FKN9978hfHAP2wPYGtGKkJfnDkxbxtOzcMgbCT
swKzwqZVaZht7AK4C6joA/mxumeBeazSN1dcS9/13tvf3L1zyF6rauqiRbh0mExbh/udlf+e+4l099cbPdhM24LBQnC61/bzz677R52g#nsAkndM7o@t
gScFIalmdZBeSTIX47h690kg8+1xNmzbExxfKnKege@9wQI88CNo710BwtjTv1MfhHTfpAzVLJVLrvuN2Ctmxb2FE14TztcoeP/os1XH+lo2rvkdKE
TTGdUs/GkQOHNLVV+H1E5UlpzmcHoo/LT50TK2U0g77rSH77Ch/8uQ78381Aoqd5QWUacuXcZdBGPrgzvpastlhJsdzhh2XCFTSfpScihm8CuksDFK
cn7dbZtwcF+fffb20pHPw== evkurnina@dk6n54
```

Рис. 3.8: Вызов ключа

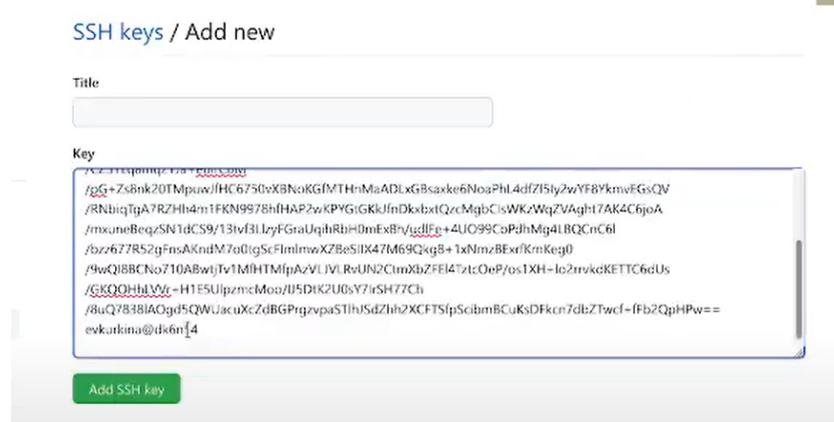


Рис. 3.9: Копирование его в нужную ячейку

6.2) Добавление PGP ключа в GitHub: -Командой-gpg –list-secret-keys–keyid-format LONG-вызывала и копировала отпечаток приватного ключа, а затем скопировала в поле на github (рис.3.10). Результат добавление приватного ключа (рис.[-fig:11]).

```

mQINBGJhPSUDBBzD7qjgnX9XLnhqApRd6tEq8jqjJpLwRFVU7Yt1N5F30na0
mdyyzxQSU71mTyZ6r1ug3/s9XLGqoyz6xQcsz2v1BMy3/IcpE70KtkR1snzfY1t
+uy1CNA82t0nkNclwyR0/gmnEmvI20jWfwooQ9j9qJAnhK8XAKfSfSt7BDfcKtDb
@Cn97eC9rk0H0sayaJo5mZef+93srkVbK27pYhrfZKrVbbhb*4xZ1SMVV1coPnx
y3qgmaBDMhbqGVn21VhegfpW/qX3ojIaSwqMUBSzghmenF/346WayelqBExcpr5q
KnHSayvhXaX/7wOZ30wDfEEeArnD16jAuK4/H2w4pShvvqj3HBdfmzFqClcdcnA
dUy/8SKXW4Gz4C8ZveDdton9xFzb1VfKUqnH0YLMRIGRUppqiy8tHRPN5uVA
INHwQTeYqDgb+bJvF/BuJ/KJxTPbbIEpn+exONTYuPMUZO4G4uwY7FEGe2ctkI
3jGS1U+qFL9nRdlxnRZZ/eDo7Zhucy2l3nH2ImeLwAcac/i4n46p60mwH5eisbhG
fTPMNKG1P/yex6XV2cTw0vUgLvoqDPJw0beHjhBy2smV2BxW79nncwqxmZJymF
zuBhbgKJY0jGwzGuuyla++fveRiWkt7EdhCw1CeC/CvaHx15CBUx8WARQAB
tcJFdnd1bn1hIDxlndl2ha3ya2luVUB5YW5KXgucnU+IQJO6BMBCAA4F1EE
XPVb1iTsvPcfNUlUe4stByJhQQgFAmJhPSUCGwHFCwk1Bw1GfQoJCAsCBBYCAwEC
HgECFAACgkQE4stByJhQQjY8A//TlAMbTSuMfxFYosZ9C7W+Xctj7YFC/24FVrd
RNgsSwDveMdfLs+xGyNwL+p8fJzuwR+u43EUcgnfsz57FK10R6632mKVSYpProt
szia2A4CoJ+TxacQEMHYAB1d2R3dZ/UKRDSWJooOlfc0qQcIA6qj2LAmHGFYOG
DEDtgeEe1Hh2AT8dK50aEl+MhwM5drkfWJKduve/GIEgYMfFplkRRxgNgx8MHJ
9FJRCw0/G38p4MrMHeKjN6s+zVFadHOA7XetRU3GQfLM6B6uvH032VrDG0ZFyec
teHy7aaJT3pHi0eedVveYssmabg0YJ/6B130phNf031mJSA5euqxV*DmmsLmfb
ND+0bzaPz1W17V95rJw6DFEYlIpTiicLWkGnH0gukK1hGOrfe8sArQqW5WolYkiz
UmChIS8bF0tpBGsa1PNe0lORSAyPdhRPQzzNhN24Iais7Hkc3HcwBhscIBzuca0
ceHLBd1CpTiovIsqWRAbmKt7JJAUaM+zoxu5/Y06HafWe#Mnd#0sv#4MWnPPeWB
nJD0AVJHOUm51hPsafN2ifvXqJEvxRpDZJDsDU343kUL9Tpfi0apDVsqZC/Izssoy
VhJannZVNm1qeIILCKTabXur+K70hB9T3fuWIQMh6uWcVRxH2dSNffAJHpgq+Jc2
14yx*by5azg0EYmE9JQEQAMCS0p04mpigGki8skwq96fA7G92AcntdHN3yBekn7Q
EcPSw0CjL0Q0uIxSq365ETDCP45KnmLSG4yhgi81ld5Ai6p+PGc0qrBzaNQjY1D
UQRSnMoyv552FotgMK-Yu5fBl489tjNrnrby1VaQRC73e8bztx2MFSC6BP0Z5z+
Ui+hb1l/p111Huze0zuTixIVT+it7d03iP7i1Lpn#Ebkd4eLrNqtLem1cgTPBo
siv4LHeo5f5JboX7roxLyw+067JgwUebHEW7Boiijk+k+pHQRGiubD#Ef1naslds
rewZP516EG1ru1SPxeIMNmzAmcvqFFtCpR6RRwdtk20uurAS+5ThxldsKmXjk
qCmH9p915doBA03k10mbrrn0o10ye8iYuYG8044v8Qbxu07p5552JzgsGdI3Vhx
601zrRg4GqNb8x03mCTjxn99Fin2BHUTFMLR3C/ZoKi6bsRBtJu/g+Mb7w7hz
E8K/VHQIGNsFeZpX0TKKmcuSSASr5Q00gSL44ipf6E)47ZH/F98BZJbGjMnyJkg
OctUaxoKmxTH3LK43yzJH9WP7p5y4dQRGPHdeqXwKdkZDlkghM19tVNW0a3hE/
Nk0wxBQAA81iHs5aY5/lh02UswiK3IRGpfFB3AREQIZycDlDEDRkEhR0FMB0mGPf
ABEBAAAGTAjUEGAEIAcAWiRC9VW70xU9wU1SW4Tiy0HimFBCAUyNE9J0f1bDAK
CRATiyoHImfBCNk+D/YxQrj7Z9c3XoHXFMW28ARzdCxU32SU7k0ng03gQeyd0Yg4
mSPo/vv70IxcpoP9Av53j0VKZZnHk1h91rd4t7YHRzx2BByRXxdgDV0vFNwX8R
3Bb4xyfBKH51kZ2MKA6R8qHNS7z#fRCP69Waak9rsRkan/1/abw96JfSR2kFHyqx
nmZpIb3f071BmHorBbAYivEPQoprvtnta35mPdw1QD3sfejWysSsMbdG0yztRtBb1
mommaPfNz/E2gDoQ4JSL4mOnTn1/PERfYS6Inn4Q8uu8XSZKD3By0YCMclB1w0
H8C10A+Dx5+x0Yfp8SLmiqb9@lWNHW4e/5lp+Ehwlv32Ee5eocmRic9UbnVLs/0
ZTQ03a0vxr8sp2xkdtTE*B9if76HzNuGdlnbemaxU*pONKicr*HQiq8uDksVugvy
BLACSzXeqYYGqwhgCctxNcSS0MxCImwbUpjszxqylA@cbAnihGEAiQmHyId3vv5v
IPew300j1ab8fJtHSf/Uof3LDxPTC1Gqhn7lG3QebCO4domMZ350W1PFpooX9Lu
cnfHqhNCHYoBZMajh2h5a5rAZDyD3AalQcT4fEgY7VR51H17gWDoAy4cdTUWfz
P0cUYxctUEp4s7Q4HanUimA2DN9IKm9yJt3zCjeJgB7Qf177651IBPqdcd
=rHtd
-----END PGP PUBLIC KEY BLOCK-----

```

Рис. 3.10: След приватного ключа



Рис. 3.11: Созданный ключ

7).Выполнение основные команды git 7.1) Git init- создала основное дерево
репозитория 7.2) git pull-получила обновления текущего дерева из центрального

репозитория 7.3) git push- отправила все произведённые изменения локального дерева в центральный репозиторий (рис.3.12) (рис.3.13)

```
evkirkina@dk6n54 ~ $ git init
Инициализирован пустой репозиторий Git в /afs/.dk.sci.pfu.edu.ru/home/e/v/evkirkina/.git/
evkirkina@dk6n54 ~ $ git pull
У текущей ветки нет информации об отслеживании.
Пожалуйста, укажите с какой веткой вы хотите слить изменения.
Для дополнительной информации, смотрите git-pull(1).

git pull <внешний-репозиторий> <ветка>

Если вы хотите указать информацию о отслеживаемой ветке, выполните:

git branch --set-upstream-to=<внешний-репозиторий>/<ветка> master

evkirkina@dk6n54 ~ $ git push
fatal: Не настроена точка назначения для отправки.
Либо укажите URL с помощью командной строки, либо настройте внешний репозиторий с помощью

git remote add <имя> <адрес>
а затем отправьте изменения с помощью имени внешнего репозитория

git push <имя>

evkirkina@dk6n54 ~ $ git
```

Рис. 3.12: Создание основного дерева репозитория и получения обновлений

```
create mode 100644 .gitignore
evkirkina@dk6n54 ~$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (19/19), 265.87 КиБ | 2.20 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 github.com:evkirkina/study_2021-2022_os-intro.git
  61fcf71..2e7e8bc master -> master
```

Рис. 3.13: Отправление изменений в репозиторий

8).Настроила автоматические подписи коммитов git (рис.3.14)

```
evkirkina@dk6n54 ~$ git config --global user.signingkey 138B2D0722614108
evkirkina@dk6n54 ~$ git config --global commit.gpgsign true
evkirkina@dk6n54 ~$ git config --global gpg.program $(which gpg2)
```

Рис. 3.14: Настройка автоматических подписей

9).Создание репозитория курса на основе шаблона: 9.1) Командами -mkdir -p ~/work/study/2021-2022/“Операционные системы” cd ~/work/study/2021-2022/“Операционные системы” gh repo create study_2021-2022_os-intro-template=yamadharma/c directory-student-template –public 4, git clone –recursive git@github.com:/study_2021-2022_os-intro.git os-intr - создала шаблон рабочего пространства. –для 2021–2022 учебного года и предмета «Операционные системы» (код предмета os-intro). (рис.3.15)

```
evkurkina@dk6n54 ~ $ mkdir -p ~/work/study/2021-2022/"Операционные системы"
evkurkina@dk6n54 ~ $ cd ~/work/study/2021-2022/"Операционные системы"
evkurkina@dk6n54 ~/work/study/2021-2022/Операционные системы $ 
evkurkina@dk6n54 ~/work/study/2021-2022/Операционные системы $ git clone --recursive
git clone --recursive
```

Рис. 3.15: Создание шаблона рабочего пространства

9.2) Показан процесс копирования репозитория в свой github (рис.3.16) (рис.3.17).Появившаяся папка Labs(рис.3.18)

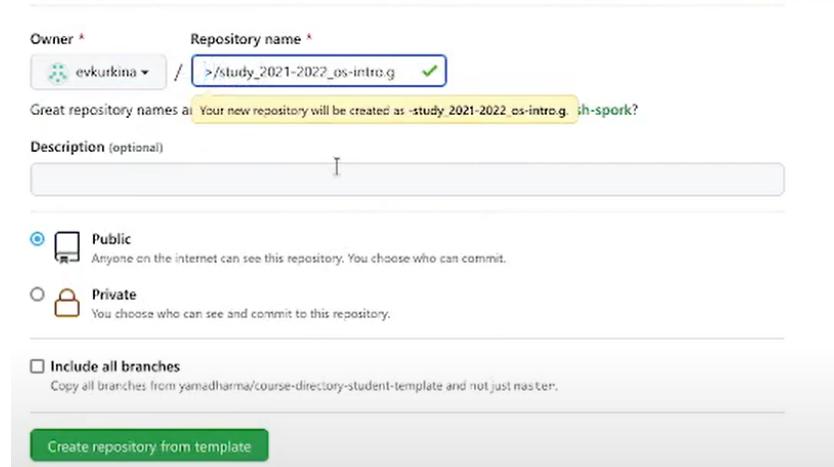


Рис. 3.16: Копирование репозитория

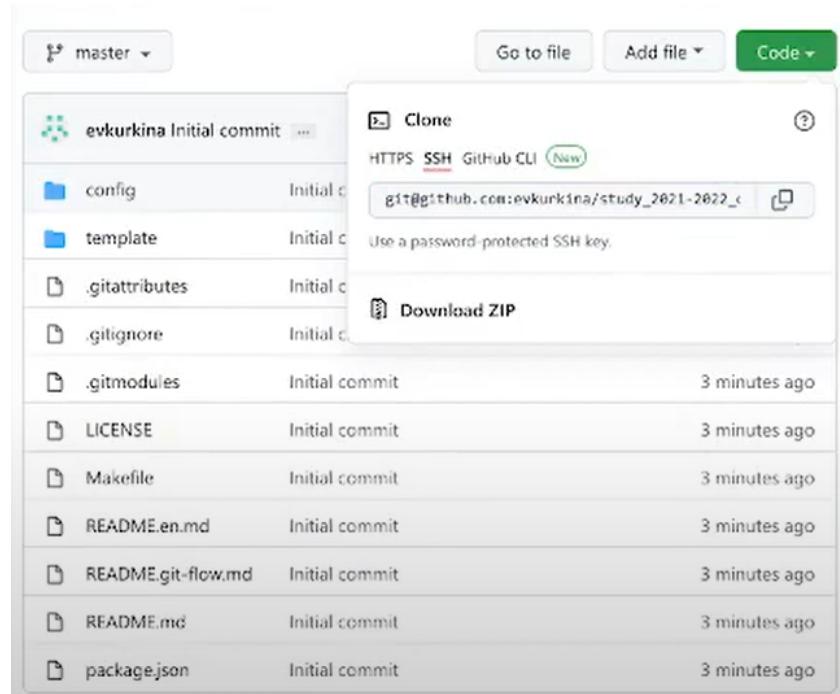


Рис. 3.17: Вставка ссылки в мой github

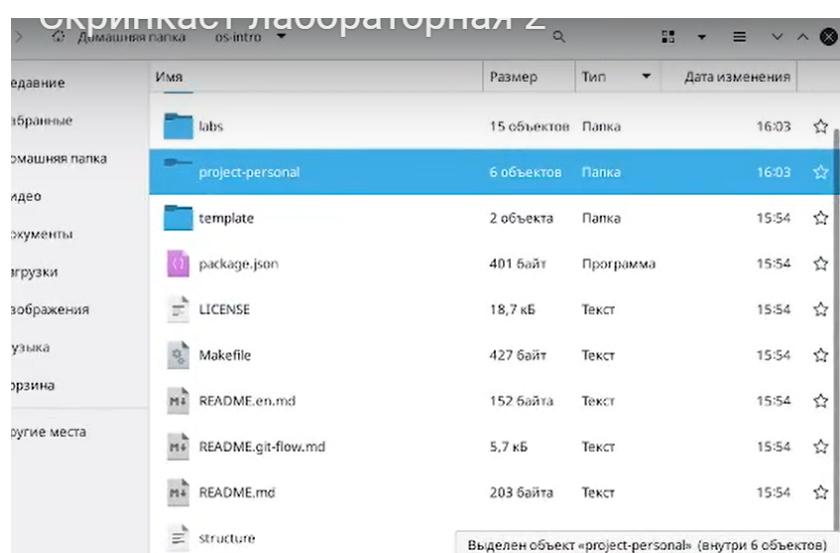
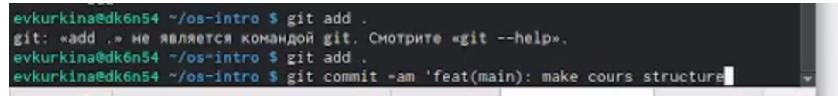


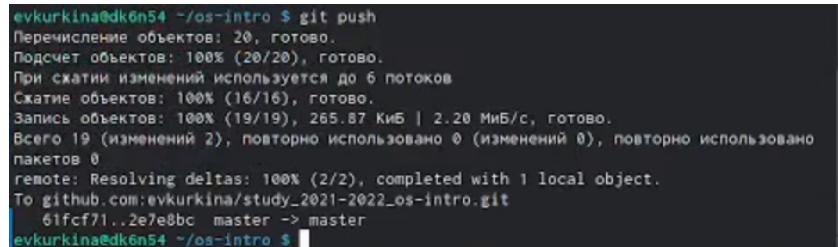
Рис. 3.18: Появившаяся папка

10).Настроила каталог курса:Перешла в ранее созданный каталог курса, соединила необходимые каталоги, а затем отправила все на сервер (рис.3.19) (рис.3.20)



```
evkurkina@dk6n54 ~/os-intro $ git add .
git: «add» не является командой git. Смотрите «git --help».
evkurkina@dk6n54 ~/os-intro $ git add .
evkurkina@dk6n54 ~/os-intro $ git commit -am 'feat(main): make cours structure'
```

Рис. 3.19: Соединение каталогов



```
evkurkina@dk6n54 ~/os-intro $ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (19/19), 265.87 КиБ | 2.20 МиБ/с, готово.
Всего 19 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано
пакетов 0
репом: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:evkurkina/study_2021-2022_os-intro.git
  61fcf71..2e7e8bc master -> master
evkurkina@dk6n54 ~/os-intro $
```

Рис. 3.20: Отправка на сервер

11).Ответы на вопросы:

1).VCS-программное обеспечение для работы с изменяющейся информацией, позволяет хранить несколько версий одного и того же документа и возвращаться к более ранним версиям.

2). 2.1Хранилище-репозиторий файлов и папок, изменения которых отслеживаются2.2.commit-объект, в котором содержится описание изменений, которые были внесены в код Git.

2.3) История- VCS-хранит все изменения и копии

2.4)Рабочая копия- каталог на компьютере пользователя, в котором происходит работа

3). 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 качестве примеров таких программных продуктов можно привести CVS, Subversion.Распределенные системы контроля версий (Distributed Version Control System,DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории

будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”. Большое преимущество такого подхода заключается в автономии разработчика при работе над проектом, гибкости общей системы и повышение надежности, благодаря тому, что каждый разработчик имеет локальную копию центрального репозитория. Две наиболее известные DVCS – это Git и Mercurial. 4). Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория: git config –global user.name”Имя Фамилия” git config –global user.email”work@mail” и настроив utf-8 в выводе сообщенийgit: git config –global quotepath false Для инициализации локального репозитория, расположенного, например, в каталоге ~/tutorial, необходимо ввести в командной строке: cd mkdir tutorial cd tutorial git init 5). Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый): ssh-keygen -C”Имя Фамилия work@mail” Ключи сохраняются в каталоге~/.ssh/. Скопировав из локальной консоли ключ в буфер обмена cat ~/.ssh/id_rsa.pub | xclip -sel clip вставляем ключ в появившееся на сайте поле. 6). У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом. 7). Основные команды git: Наиболее часто используемые команды 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: git commit -am ‘Описание коммита’ – сохранить добавленные изменения с внесением комментария через встроенный редактор: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 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). Использования git при работе с локальными репозиториями (добавления текстового документа в локальный репозиторий): git add hello.txt git commit -am‘Новый файл 9). Проблемы, которые решают ветки git: · нужно постоянно создавать архивы с рабочим кодом · сложно “переключаться” между архивами · сложно перетаскивать изменения между архивами · легко что-то напутать или потерять 10). Во время работы над проектом так или иначе могут создаваться файлы, которые нетребуется добавлять в последствии в репозиторий. Например, временные файлы, создаваемые редакторами, или объектные файлы, создаваемые компиляторами. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл.gitignore с помощью curl https://www.gitignore.io/api/list Затем скачать шаблон, например, для C и C++ curl -L -s https://www.gitignore.io/api/c ».gitignore curl -L -s https://www.gitignore.io/api/c++ ».gitigno # Выводы

Во время выполнения лабораторной работы, я научилась оформлять отчеты с

помощью легковесного языка разметки Markdown.

Список литературы