

# **Отчёт по лабораторной работе №2**

**Простейший вариант**

Тарутина Кристина Олеговна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>17</b>
	<b>Список литературы</b>	<b>20</b>

## Список иллюстраций

2.1	Предварительная конфигурация . . . . .	6
2.2	Создание SSH-ключа . . . . .	7
2.3	Копирование ключа в буфер обмена . . . . .	7
2.4	Создание каталога . . . . .	8
2.5	Создание репозитория . . . . .	8
2.6	Клонирование репозитория . . . . .	9
2.7	Настройка . . . . .	9
2.8	Отправка файлов на сервер . . . . .	9
2.9	Проверка . . . . .	10
2.10	Проверка . . . . .	10
2.11	Создание отчёта . . . . .	11
2.12	отчёт № 2 в каталог . . . . .	11
2.13	отчёт № 1 в каталоге . . . . .	12
2.14	Загрузка на github . . . . .	12
2.15	Установка gh . . . . .	13
2.16	Создание ключа pgr . . . . .	13
2.17	Копирование отпечатка ключа . . . . .	14
2.18	Настройка . . . . .	14
2.19	Авторизация . . . . .	15
2.20	Создание шаблона рабочего пространства . . . . .	15
2.21	Создание шаблона рабочего пространства . . . . .	15
2.22	Настройка . . . . .	16
2.23	Отправка . . . . .	16

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

#Внимание:

Так как в данной лабораторной работе часть заданий совпадает с заданиями в лабораторной работе №2 прошлого семестра, то на место этих заданий будет установлен старый отчёт. Потом он будет продолжаться новым

Шаг 1: Настройка github

Так как у меня уже была учётная запись на github, то этот шаг я пропустила

Шаг 2: Базовая настройка git

Делаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав своё имя и email Настраиваю utf-8 в выводе сообщений git Задаю имя начальной ветки, Параметр autocrlf и параметр safecrlf (рис. 2.1).

```
[kotarutina@fedora ~]$ git config --global user.name "<TiRisik>"
[kotarutina@fedora ~]$ git config --global user.email "<valentinrusso@mail>"
[kotarutina@fedora ~]$ git config --global core.quotepath false
[kotarutina@fedora ~]$ git config --global init.defaultBranch master
[kotarutina@fedora ~]$ git config --global core.autocrlf input
[kotarutina@fedora ~]$ git config --global core.safecrlf warn
[kotarutina@fedora ~]$ ssh-keygen -C "TiRisik <valentinrusso@mail>"
```

Рис. 2.1: Предварительная конфигурация

Шаг 3: Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория генерирую пару ключей (приватный и открытый)(рис. 2.2).

```
[kotarutina@fedora ~]$ ssh-keygen -C "TiRisik <valentinrusso@mail>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kotarutina/.ssh/id_rsa):
/home/kotarutina/.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 /home/kotarutina/.ssh/id_rsa
Your public key has been saved in /home/kotarutina/.ssh/id_rsa.pub
The key fingerprint is:
```

Рис. 2.2: Создание SSH-ключа

Загружаю сгенерированный открытый ключ. Для этого захожу на сайт <http://github.org/> под своей учётной записью и перехожу в меню Setting . После этого выбираю в боковом меню SSH and GPG keys и нажимаю кнопку New SSH key . Устанавливаю пакет xclip и копирую из локальной консоли ключ в буфер обмена (рис 3) (рис. 2.3)

```
[kotarutina@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
bash: xclip: команда не найдена...
Установить пакет «xclip», предоставляющий команду «xclip»? [N/y] y

* Ожидание в очереди...
Следующие пакеты должны быть установлены:
xclip-0.13-16.git11c6a61.fc36.x86_64 Command line clipboard grabber
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

[kotarutina@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[kotarutina@fedora ~]$
```

Рис. 2.3: Копирование ключа в буфер обмена

Вставляю ключ в появившееся на сайте поле и указываем для ключа имя (здесь я, к сожалению, забыла сделать скриншот)

Шаг 4: Создание рабочего пространства и репозитория курса на основе шаблона  
Создаю каталог для предмета «Архитектура компьютера» (рис. 2.4)

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

Рис. 2.4: Создание каталога

Перехожу на страницу репозитория с шаблоном курса, выбираю Use this template. В открывшемся окне задаю имя репозитория study\_2022-2023\_arh- pc и создаю репозиторий (рис. 2.5)

Owner <sup>\*</sup> Repository name <sup>\*</sup>

TiRisik / study\_2022-2023\_arh-pc ✓

Great repository name Your new repository will be created as study\_2022-2023\_arh-pc. laughing-sniffle?

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

Рис. 2.5: Создание репозитория

В терминале перехожу в каталог курса, клонирую созданный репозиторий, ссылку для клонирования копирую на странице созданного репозитория(рис. 2.6).



```
[kotarutina@fedora Архитектура компьютера]$ git clone --recursive git@github.com:TiRisik/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 | 2.34 МБ/с, 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/kotarutina/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 | 948.00 КиБ/с, готово.
Определение изменений: 100% (23/23), готово.
Клонирование в «/home/kotarutina/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), 202.27 КиБ | 1.73 МБ/с, готово.
Определение изменений: 100% (31/31), готово.
Submodule path 'template/presentation': checked out '2703b47423792d472694aaf7555a5626dce51a25'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'
```

Рис. 2.6: Клонирование репозитория

Шаг 5: Настройка каталога курса Перехожу в каталог курса, Удаляю лишние файлы, Создаю необходимые каталоги (рис. 2.7). Отправляю файлы на сервер (рис. 2.7 - рис. 2.8).

```
[kotarutina@fedora Архитектура компьютера]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc
[kotarutina@fedora arch-pc]$ rm package.json
[kotarutina@fedora arch-pc]$ echo arch-pc > COURSE
[kotarutina@fedora arch-pc]$ make
[kotarutina@fedora arch-pc]$ git add
ничего не указано, ничего не добавлено.
подсказка: Maybe you wanted to say 'git add .' ?
подсказка: Turn this message off by running
подсказка: "git config advice.addEmptyPathsSpec false"
[kotarutina@fedora arch-pc]$ git add .
[kotarutina@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 2f983de] 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/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
```

Рис. 2.7: Настройка

```
[kotarutina@fedora arch-pc]$ git push
Перечисление объектов: 22, готово.
Подсчет объектов: 100% (22/22), готово.
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (20/20), 310.94 КиБ | 1.87 МБ/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:TiRisik/study_2022-2023_arh-pc.git
 0e74abe..2f983de master -> master
```

Рис. 2.8: Отправка файлов на сервер

Проверяю правильность создания иерархии рабочего пространства в локальном репозитории и на странице github. (рис. 2.9 - 2.10).

config	Initial commit	2 days ago
labs	feat(main): make course structure	5 minutes ago
template	Initial commit	2 days ago
.gitattributes	Initial commit	2 days ago
.gitignore	Initial commit	2 days ago
.gitmodules	Initial commit	2 days ago
CHANGELOG.md	Initial commit	2 days ago
COURSE	feat(main): make course structure	5 minutes ago
LICENSE	Initial commit	2 days ago
Makefile	Initial commit	2 days ago
README.en.md	Initial commit	2 days ago
README.git-flow.md	Initial commit	2 days ago
README.md	Initial commit	2 days ago
prepare	feat(main): make course structure	5 minutes ago

Рис. 2.9: Проверка

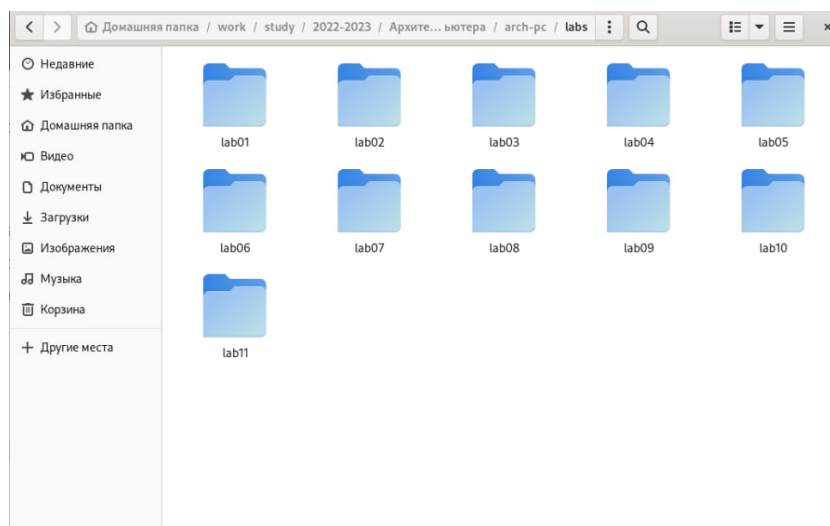


Рис. 2.10: Проверка

Далее идёт самостоятельная работп

- 1) Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab03>report). Создала отчёт в соответствующем каталоге (рис. 2.11).

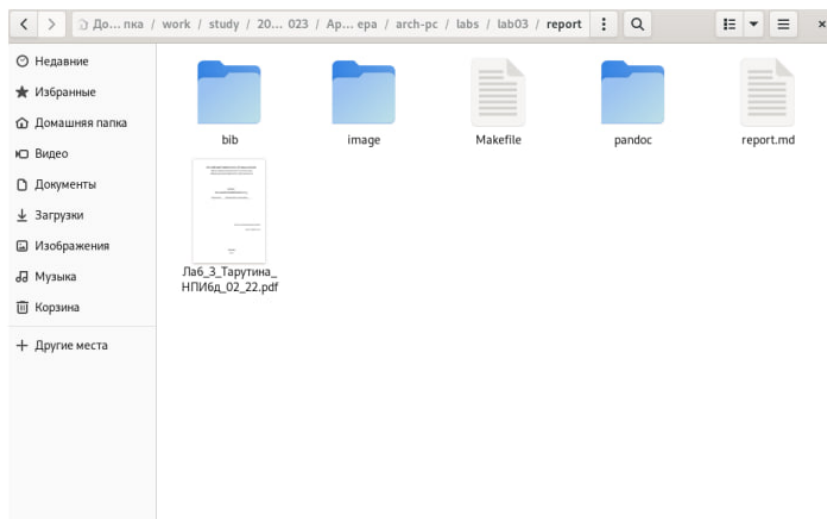


Рис. 2.11: Создание отчёта

- 2) Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства. Скопировала отчёты в соответствующие им каталоги (рис. 2.12 - 2.13).

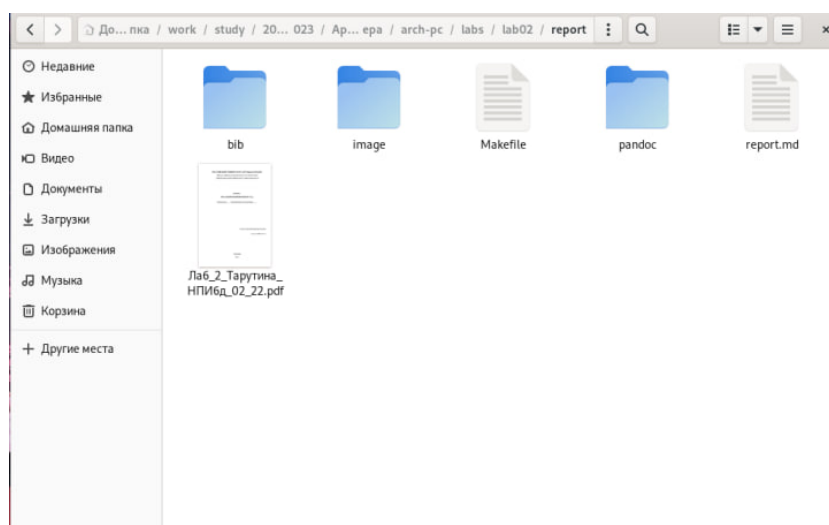


Рис. 2.12: отчёт № 2 в каталогк

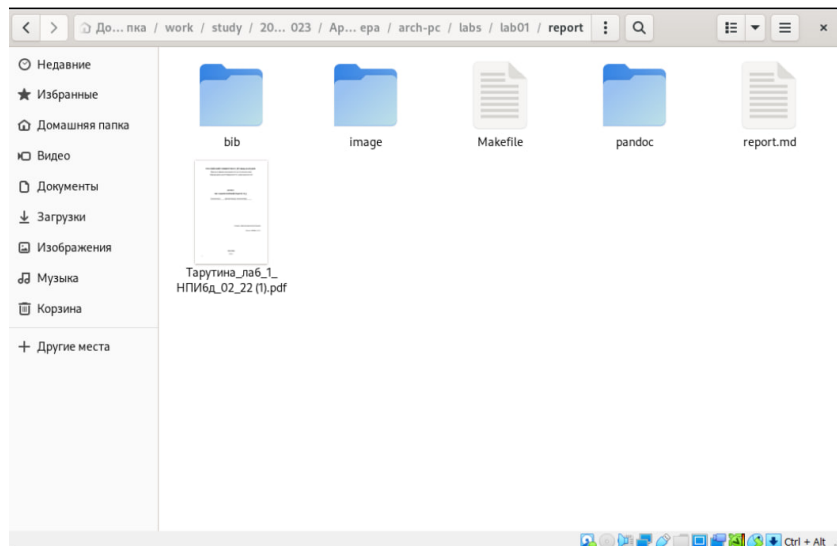


Рис. 2.13: отчёт № 1 в каталоге

3) Загрузите файлы на github.

Загрузила файлы с помощью изображённых ниже команд(рис. 2.14)

```
[kotarutina@fedora ~]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc
[kotarutina@fedora arch-pc]$ git add .
[kotarutina@fedora arch-pc]$ git commit -am "Добавила лабораторные работы 1 и 2"
[master e85ed43] Добавила лабораторные работы 1 и 2
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Тарутина_лаб_1_НПИбд_02_22 (1).pdf
create mode 100644 labs/lab02/report/Лаб_2_Тарутина_НПИбд_02_22.pdf
[kotarutina@fedora arch-pc]$ git push
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (13/13), готово.
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (9/9), 2.60 МиБ | 1.29 МиБ/с, готово.
Всего 9 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:TiRisik/study_2022-2023_arh-pc.git
2f983de..e85ed43 master -> master
[kotarutina@fedora arch-pc]$
```

Рис. 2.14: Загрузка на github

На этом перенос старого отчёта закончен

Так как роль суперпользователя убирает из терминала надпись kotarutina (что мешает соглашению о наименовании), то я делаю снимок практически всего терминала, захватывая при этом kotarutina в самом верху окна. Иногда во избежании

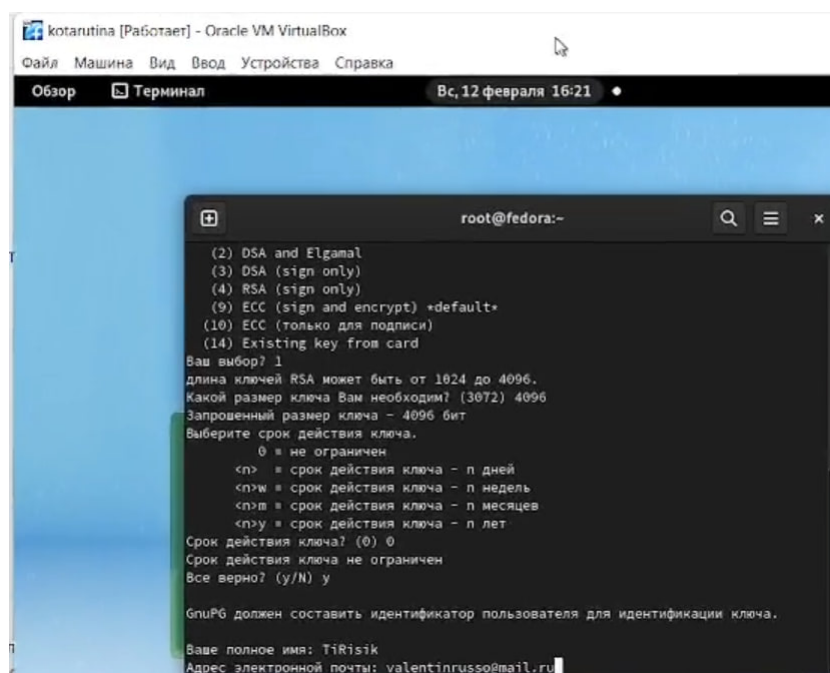
путаницы с командами я замазываю ненужные чёрным.

Устанавливаю gh(рис. 2.15)

```
[kotarutina@fedora ~]$ dnf install gh
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на больш
стве систем - под именем пользователя root).
[kotarutina@fedora ~]$ sudo -i
[sudo] пароль для kotarutina:
Попробуйте ещё раз.
[sudo] пароль для kotarutina:
Попробуйте ещё раз.
[sudo] пароль для kotarutina:
[root@fedora ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:34:49 назад, Вс 12 Ф
2023 15:43:30.
Пакет gh-2.22.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
```

Рис. 2.15: Установка gh

Генерирую ключ Из предложенных опций выбираю: тип RSA and RSA; размер 4096; выбираю срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). Ввожу личную информацию по запросу(рис. 2.16)



```
kotarutina [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Обзор Терминал Вс, 12 февраля 16:21
root@fedora:~
(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: TiRisik
Адрес электронной почты: valentinrusso@mail.ru
```

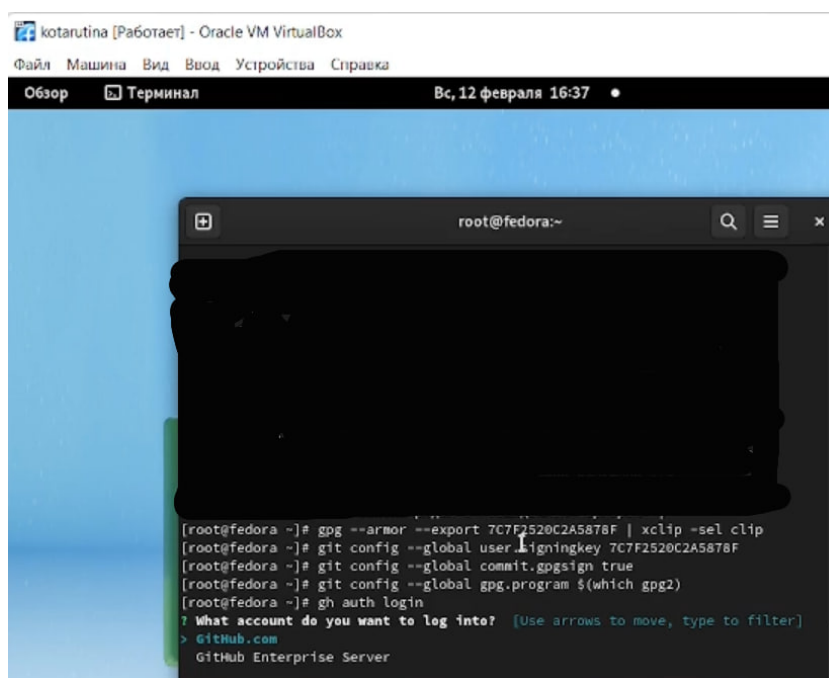
Рис. 2.16: Создание ключа gpg

Вывожу список ключей и копирую отпечаток приватного ключа (рис. 2.17)

```
[root@fedora ~]# 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
-----
/root/.gnupg/pubring.kbx
-----
sec   rsa4096/7C7F2520C2A5878F 2023-02-12 [SC]
      F8F1ED3ECE07853A4347B4367C7F2520C2A5878F
uid    [ абсолютно ] TiRisik <valentinrusso@mail.ru>
ssb    rsa4096/8C1D3D10E4693F19 2023-02-12 [E]
```

Рис. 2.17: Копирование отпечатка ключа

Копирую сгенерированный PGP ключ в буфер обмена, используя введённый email, указываю Git применять его при подписи коммитов(рис. 2.18)



The screenshot shows a terminal window titled "kotarutina [Работает] - Oracle VM VirtualBox" with a menu bar (Файл, Машина, Вид, Ввод, Устройства, Справка) and a toolbar (Обзор, Терминал). The terminal output shows the following commands and their results:

```
root@fedora:~
[root@fedora ~]# gpg --armor --export 7C7F2520C2A5878F | xclip -sel clip
[root@fedora ~]# git config --global user.signingkey 7C7F2520C2A5878F
[root@fedora ~]# git config --global commit.gpgsign true
[root@fedora ~]# git config --global gpg.program $(which gpg2)
[root@fedora ~]# gh auth login
? What account do you want to log into? [Use arrows to move, type to filter]
> GitHub.com
  GitHub Enterprise Server
```

Рис. 2.18: Настройка

Авторизовываюсь (рис. 2.19)

```
[kotarutina@fedora ~]$ sudo -i
[sudo] пароль для kotarutina:
[root@fedora ~]# 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? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol https
/ Configured git protocol
/ Logged in as TiRisiK
[root@fedora ~]#
```

Рис. 2.19: Авторизация

Создаю шаблон рабочего пространства(рис. 2.20 - 2.21)

```
[kotarutina@fedora Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
GraphQL: Your token has not been granted the required scopes to execute this query. The 'cloneTemplateRepository' field requires one of the following scopes: ['public_repo'], but your token has only been granted the: [] scopes. Please modify your token's scopes at: https://github.com/settings/tokens.
```

Рис. 2.20: Создание шаблона рабочего пространства

```
kotarutina@fedora:~/work/study/2022-2023/Операционные с...
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 КиБ | 8.46 МиБ/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/kotarutina/work/study/2022-2023/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 779.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/kotarutina/work/study/2022-2023/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 КиБ | 1.74 МиБ/с, готово.
Определение изменений: 100% (40/40), готово.
```

Рис. 2.21: Создание шаблона рабочего пространства

Перехожу в каталог курса, удаляю лишние файлы, создаю необходимые ката-



логи(рис. 2.22)

```
[kotarutina@fedora Операционные системы]$ cd os-intro
[kotarutina@fedora os-intro]$ rm package.json
[kotarutina@fedora os-intro]$ echo os-intro > COURSE
[kotarutina@fedora os-intro]$ make
```

Рис. 2.22: Настройка

Отправляю файлы на сервер(рис. 2.23)

```
[kotarutina@fedora os-intro]$ git add .
bash: git: команда не найдена...
[kotarutina@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master fc9a537] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
[kotarutina@fedora os-intro]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 283 байта | 94.08 КиБ/с, готово.
Всего 3 (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:TiRisik/study_2022-2023_os-intro.git
 be66ce6..fc9a537 master -> master
[kotarutina@fedora os-intro]$
```

Рис. 2.23: Отправка



## 3 Выводы

Изучение идеологии и применение средств контроля версий; освоение умения по работе с git прошло успешно

#Контрольные вопросы

Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы.

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

Хранилище или репозиторий, — место хранения всех версий и служебной информации. Рабочая копия — текущее состояние файлов проекта, основанное на версии из хранилища (обычно на последней). Коммит - создание новой версии кода

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

Централизованные - Клиент-серверная модель: один центральный репозиторий, с которым разработчики взаимодействуют по сети. Централизованные VCS: Примеры • CVS- одна из первых систем второго поколения (1986г.). Обладает множеством недостатков и считается устаревшей. • Subversion (SVN) – система

второго поколения, созданная для замены CVS. Одна из самых распространенных систем контроля версий.

Распределенные VCS - В отличие от централизованной модели, может существовать несколько экземпляров репозитория, которые время от времени синхронизируются между собой. Распределенные VCS: примеры • Git- распределенная система управления версиями, созданная Л. Торвальдсом для управления разработкой ядра Linux. • Mercurial- другая распределенная VCS. Создана в 2005 году М. Макалом с практически одновременно с началом разработке git'a и с аналогичными целями.

Каковы основные задачи, решаемые инструментальным средством git?

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.

Назовите и дайте краткую характеристику командам git.

Команда `git add` добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита. Команда `git status` показывает состояния файлов в рабочем каталоге и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов. Команда `git diff` используется для вычисления разницы между любыми двумя Git деревьями. Команда `git commit` берёт все данные, добавленные в индекс с помощью `git add`, и сохраняет их коммит во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот коммит. Команда `git reset`, как можно догадаться из названия, используется в основном для отмены изменений. Команда `git rm` используется в Git для удаления файлов из индекса и рабочей копии. Команда `git mv` — это всего лишь удобный способ переместить файл, а затем выполнить `git add` для нового файла и `git rm` для старого. Команда `git clean` используется для удаления мусора из рабочего каталога. Это могут быть результаты сборки проекта или файлы конфликтов слияний.

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

Ветка (англ. branch) — это последовательность коммитов, в которой ведётся параллельная разработка какого-либо функционала. Основная ветка – master Ветки в GIT. Ветки нужны, чтобы несколько программистов могли вести работу над одним и тем же проектом или даже файлом одновременно, при этом не мешая друг другу. Кроме того, ветки используются для тестирования экспериментальных функций: чтобы не повредить основному проекту, создается новая ветка специально для экспериментов.

Как и зачем можно игнорировать некоторые файлы при commit?

Игнорируемые файлы отслеживаются в специальном файле .gitignore, который регистрируется в корневом каталоге репозитория. В Git нет специальной команды для указания игнорируемых файлов: вместо этого необходимо вручную отредактировать файл .gitignore, чтобы указать в нем новые файлы, которые должны быть проигнорированы

Игнорировать файлы можно при их конфликте с чем-либо

## **Список литературы**