

# **Отчет по лабораторной работе №2**

**Архитектура компьютеров и операционные системы**

Дмитрий Константинович Кобзев

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>15</b>
<b>6</b>	<b>Выводы</b>	<b>18</b>
	<b>Список литературы</b>	<b>19</b>

## Список иллюстраций

4.1	Установка git и gh . . . . .	8
4.2	Владелец репозитория . . . . .	8
4.3	Имя начальной ветки и параметры autocrlf и safecrlf . . . . .	8
4.4	Ключ ssh по алгоритму rsa . . . . .	9
4.5	Ключ ssh по алгоритму ed25519 . . . . .	9
4.6	Добавление PGP ключа в GitHub . . . . .	10
4.7	Новый PGP ключ . . . . .	11
4.8	Настройка автоматических подписей коммитов . . . . .	11
4.9	Настройка gh . . . . .	11
4.10	Настройка каталога курса . . . . .	13

# Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	7
-----	---	---

# 1 Цель работы

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

## 2 Задание

Создать базовую конфигурацию для работы с git.

Создать ключ SSH.

Создать ключ PGP.

Настроить подписи git.

Зарегистрироваться на Github.

Создать локальный каталог для выполнения заданий по предмету.

### 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

## 4 Выполнение лабораторной работы

Устанавливаем git и gh (рис. 4.1).

```
[root@dkkobzev ~]# dnf install git
Fedora 39 - x86_64 - Updates
Fedora 39 - x86_64 - Updates
Последняя проверка окончания срока действия метаданных: 0:00:07 назад, Вт 20 фев 2024 13:11:06.
Пакет git-2.43.2-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
[root@dkkobzev ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:00:24 назад, Вт 20 фев 2024 13:11:06.
Пакет gh-2.43.1-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 4.1: Установка git и gh

Задаем имя и email владельца репозитория и настраиваем utf-8 в выводе сообщений git (рис. 4.2)

```
[root@dkkobzev ~]# git config --global user.name "Dmitriy Kobzev"
[root@dkkobzev ~]# git config --global user.email "1132231936@pfur.ru"
[root@dkkobzev ~]# git config --global core.quotepath false
```

Рис. 4.2: Владелец репозитория

Задаем имя начальной ветки, задаем параметры autocrlf и safecrlf (рис. 4.3).

```
[root@dkkobzev ~]# git config --global init.defaultBranch master
[root@dkkobzev ~]# git config --global core.autocrlf input
[root@dkkobzev ~]# git config --global core.safecrlf warn
```

Рис. 4.3: Имя начальной ветки и параметры autocrlf и safecrlf

Создаем ssh ключ по алгоритму rsa с ключом размером 4096 бит (рис. 4.4)



```
[root@dkkobzev ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:MpylJBhiMTDAqoHWS56q4lMhvLxhuxW2H7qLHuAm17Q root@dkkobzev
The key's randomart image is:
+---[RSA 4096]-----+
|O+o                |
|.+.o               |
|+ o . . .         |
|++ + + +         |
|=.o= * S          |
|+=.Bo. o         |
|o+BoE.           |
|++ o .           |
|*+o+o.           |
+----[SHA256]-----+
```

Рис. 4.4: Ключ ssh по алгоритму rsa

Создаем ssh ключ по алгоритму ed25519 (рис. 4.5)

```
[root@dkkobzev ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
/root/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:hmTe8WsA15BHc/hrEoqzLc3XDNLWB65f0avvry04gEA root@dkkobzev
The key's randomart image is:
+--[ED25519 256]--+
|      .oo..       |
|      E .ooo      |
|      .+ o...     |
|      +.= o. o .   |
|      ooS+. + o. . |
|      o.+.*.= ...  |
|      * ooO o..   |
|      o +.o =.o.   |
|      . . .o+=+   |
+----[SHA256]-----+
```

Рис. 4.5: Ключ ssh по алгоритму ed25519

Генерируем ключ pgr (рис. ??).

```
[dkkobzev@dkkobzev ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.3; Copyright (C) 2023 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.

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: Dmitriy Kobzev
Адрес электронной почты: 1132231936@rudn.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Dmitriy Kobzev <1132231936@rudn.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  
 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/dkkobzev/.gnupg/trustdb.gpg: создана таблица доверия  
 gpg: создан каталог '/home/dkkobzev/.gnupg/openpgp-revocs.d'  
 gpg: сертификат отзыва записан в '/home/dkkobzev/.gnupg/openpgp-revocs.d/C18048985AC12059269D640736A915E946E64F53'  
 открытый и секретный ключи созданы и подписаны.

```
pub  rsa4096 2024-02-25 [SC]
     C18048985AC12059269D640736A915E946E64F53
uid          [ абсолютно ] Dmitriy Kobzev <1132231936@rudn.ru>
sub  rsa4096 2024-02-25 [E]
```

Выводим список ключей и копируем отпечаток приватного ключа. Копируем наш сгенерированный PGP ключ в буфер обмена (рис. 4.6).

```
[dkkobzev@dkkobzev ~]$ 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  rsa4096/36A915E946E64F53 2024-02-25 [SC]
     C18048985AC12059269D640736A915E946E64F53
uid          [ абсолютно ] Dmitriy Kobzev <1132231936@rudn.ru>
ssb  rsa4096/EF9B095256ED781F 2024-02-25 [E]

[dkkobzev@dkkobzev ~]$ gpg --armor --export 1132231936@rudn.ru | xclip -sel clip
```

Рис. 4.6: Добавление PGP ключа в GitHub

Переходим в настройки GitHub, нажимаем на кнопку New PGP key и вставляем полученный ключ в поле ввода (рис. 4.7).

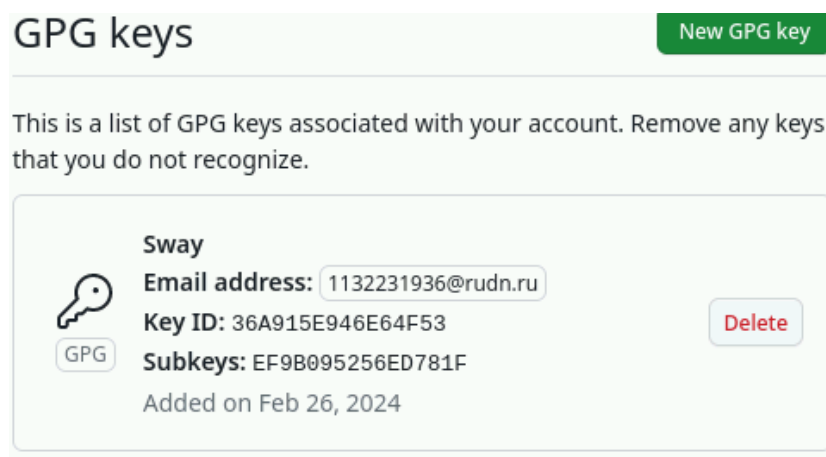


Рис. 4.7: Новый PGP ключ

Используя введенный email, указываем Git применять его при подписи коммитов (рис. 4.8).

```
[dkkobzev@dkkobzev ~]$ git config --global user.signingkey 1132231936@rudn.ru
[dkkobzev@dkkobzev ~]$ git config --global commit.gpgsign true
[dkkobzev@dkkobzev ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 4.8: Настройка автоматических подписей коммитов

Авторизовываемся через gh (рис. 4.9).

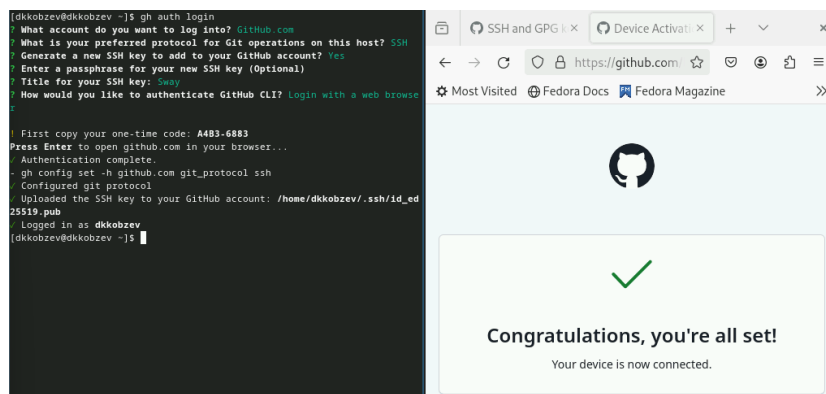


Рис. 4.9: Настройка gh

Создаем репозиторий курса на основе шаблона (рис. ??), (рис. ??), (рис. ??).

```

[dkkobzev@dkkobzev ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"
[dkkobzev@dkkobzev ~]$ cd ~/work/study/2023-2024/"Операционные системы"
[dkkobzev@dkkobzev Операционные системы]$ git clone --recursive git@github.com:dkkobzev/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:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
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 | 6.20 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/dkkobzev/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 | 237.00 КиБ/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/home/dkkobzev/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 | 535.00 КиБ/с, готово.
Определение изменений: 100% (52/52), готово.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca72c60a304f24c'
Submodule path 'template/report': checked out '7c31ab8e5dfa8cdb2d67cae8a19ef8028ced88e'

```

```

[dkkobzev@dkkobzev Операционные системы]$ gh
os-intro --template=yamadharma/course-directic
Created repository dkkobzev/study_2023-2024
https://github.com/dkkobzev/study_2023-2024

```

Переходим в каталог курса,удаляем лишние файлы и создаем необходимые каталоги (рис. 4.10).

```
[dkkobzev@dkkobzev study]$ cd ~/work/study/2023-2024/"Операционные системы"/os-intro
[dkkobzev@dkkobzev os-intro]$ rm package.json
[dkkobzev@dkkobzev os-intro]$ echo os-intro > COURSE
[dkkobzev@dkkobzev os-intro]$ make
Usage:
  make <target>

Targets:
  list                List of courses
  prepare             Generate directories structure
  submodule           Update submules

[dkkobzev@dkkobzev os-intro]$ make prepare
```

Рис. 4.10: Настройка каталога курса

Отправляем файлы на сервер (рис. ??), (рис. ??), (рис. ??), (рис. ??).

```
[dkk kobzev@dkk kobzev os-intro]$ git commit -am
ructure'
[master b02b8dd] feat(main): make course stru
361 files changed, 98413 insertions(+), 14 d
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/M
create mode 100644 labs/lab01/presentation/
create mode 100644 labs/lab01/presentation/p
create mode 100644 labs/lab01/report/Makefil
create mode 100644 labs/lab01/report/bib/cit
create mode 100644 labs/lab01/report/image/p
create mode 100644 labs/lab01/report/pandoc/
ic.csl
create mode 100755 labs/lab01/report/pandoc/
create mode 100755 labs/lab01/report/pandoc/
create mode 100755 labs/lab01/report/pandoc/
create mode 100755 labs/lab01/report/pandoc/
create mode 100644 labs/lab01/report/pandoc/
.py
create mode 100644 labs/lab01/report/pandoc/
create mode 100644 labs/lab01/report/pandoc/
create mode 100644 labs/lab01/report/pandoc/
tributes.py
create mode 100644 labs/lab01/report/report.
create mode 100644 labs/lab02/presentation/M
create mode 100644 labs/lab02/presentation/i
create mode 100644 labs/lab02/presentation/p
create mode 100644 labs/lab02/report/Makefil
create mode 100644 labs/lab02/report/bib/cit
create mode 100644 labs/lab02/report/image/p
create mode 100644 labs/lab02/report/pandoc/
ic.csl
create mode 100755 labs/lab02/report/pandoc/
create mode 100755 labs/lab02/report/pandoc/
create mode 100755 labs/lab02/report/pandoc/
create mode 100755 labs/lab02/report/pandoc/
create mode 100644 labs/lab02/report/pandoc/
.py
create mode 100644 labs/lab02/report/pandoc/
create mode 100644 labs/lab02/report/pandoc/
create mode 100644 labs/lab02/report/pandoc/
tributes.py
create mode 100644 labs/lab02/report/report.
```

```
[dkkobzev@dkkobzev os-intro]$ git add
```

```

_eqnos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc
_fignos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc
_secnos.py
create mode 100755 project-personal/stage5/report/pandoc/filters/pandoc
_tablenos.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandoc
xnos/__init__.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandoc
xnos/core.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandoc
xnos/main.py
create mode 100644 project-personal/stage5/report/pandoc/filters/pandoc
xnos/pandocattributes.py
create mode 100644 project-personal/stage5/report/report.md
create mode 100644 project-personal/stage6/presentation/Makefile
create mode 100644 project-personal/stage6/presentation/image/kulyabov.
jpg
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_60
0_tech.jpg
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0
-5-2008-numeric.csl
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc
_eqnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc
_fignos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc
_secnos.py
create mode 100755 project-personal/stage6/report/pandoc/filters/pandoc
_tablenos.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc
xnos/__init__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc
xnos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc
xnos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandoc
xnos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
[dkkobzev@dkkobzev os-intro]$

```

```

[dkkobzev@dkkobzev os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.10 КиБ | 2.33
Всего 38 (изменений 4), повторно использовано 0
использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed
To github.com:dkkobzev/study_2023-2024_os-intro.
   ed5d420..b02b8dd master -> master

```

## 5 Контрольные вопросы

1. Системы контроля версий (VCS) предназначены для управления изменениями в исходном коде и других файлов проекта, позволяя отслеживать изменения, управлять версиями, совместно работать, создавать резервные копии, отслеживать авторство, управлять конфликтами, а также экспериментировать и ветвить проекты.
2. Хранилище - место, где хранятся файлы проекта и история их изменений. Commit - сохранение текущего состояния файлов проекта с описанием изменений. История - записи о всех сделанных изменениях в проекте. Рабочая копия - локальная копия файлов проекта, позволяющая вносить изменения и синхронизировать их с хранилищем.
3. Централизованные VCS: В таких системах основной репозиторий находится на центральном сервере. Разработчики обычно работают с локальными копиями файлов, синхронизируя их с центральным сервером при необходимости. Все действия происходят через этот центральный репозиторий. Примеры централизованных VCS: Subversion (SVN) CVS (Concurrent Versions System) Децентрализованные VCS: В децентрализованных системах каждый разработчик имеет локальную копию полного репозитория, включая всю его историю и метаданные. Это означает, что разработчики могут работать независимо друг от друга и без постоянного подключения к центральному серверу. Синхронизация изменений происходит напрямую между локальными репозиториями. Примеры децентрализованных VCS: Git Mercurial

4. Создание репозитория, добавление файлов для работы, фиксация изменений, отправка файлов на сервер.
5. Клонирование репозитория, работа с файлами, добавление файлов, фиксация изменений, получение изменений, отправка изменений на сервер.
6. Управление версиями файлов проекта. Совместная разработка между несколькими разработчиками. Отслеживание изменений и истории проекта. Ветвление и слияние изменений для разработки новых функций. Резервное копирование и восстановление данных. Работа с удаленными репозиториями для совместной работы и обмена изменениями.
7. Команда `git add` добавляет содержимое рабочего каталога в индекс для последующего коммита. Команда `git commit` берёт все данные, добавленные в индекс с помощью `git add`, и сохраняет их. Команда `git clone` клонирует существующий репозиторий Git, создавая локальную копию. Команда `git init` создает новый репозиторий Git в текущем каталоге. Команда `git status` показывает состояния файлов в рабочем каталоге и индексе. Команда `git push` отправляет локальные коммиты в удаленный репозиторий. Команда `git pull` получает изменения из удаленного репозитория и объединяет их с локальными изменениями. Команда `git diff` используется для вычисления разницы между любыми двумя Git деревьями. Команда `git branch` показывает список веток, а также создает, удаляет или переключается между ними.
8. Локальный: `git commit` - фиксирует изменения Удаленный: `git push` - отправляет изменения на сервер
9. Ветви (branches) в системах контроля версий, представляют собой механизм, позволяющий разработчикам работать параллельно над разными версиями проекта, изолируя изменения и эксперименты от основной линии разработки.
10. Игнорировать файлы при commit можно с помощью файла `‘.gitignore’`. Этот



файл содержит шаблоны путей к файлам и каталогам, которые не должны быть добавлены в репозиторий Git. Как использовать: Создаем файл `‘.gitignore’` в корневом каталоге проекта и добавляем в него шаблоны путей к файлам и каталогам, которые хотим проигнорировать. Добавляем файл в репозиторий Git и выполняем коммит. Зачем: Игнорировать временные файлы, не являющиеся частью исходного кода. Предотвращение попадания конфиденциальных данных в репозиторий. Улучшение производительности Git, исключив большие или ненужные файлы из репозитория.

## 6 Выводы

Мною были изучены идеология и применение средств контроля версий и освоены умения по работе с git.

## Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.