

# **Отчет по лабораторной работе №2**

**Дисциплина: Архитектура компьютеров**

Царёв Максим Александрович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>4</b>	<b>Выводы</b>	<b>15</b>

## **Список иллюстраций**

## **Список таблиц**

# 1 Цель работы

Изучить идеологию и применение средств контроля версий.

Освоить умения по работе с `git`.

## 2 Теоретическое введение

Системы контроля версий. Общие понятия

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.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными

участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

### 3 Выполнение лабораторной работы

устанавливаю Гит и gh

```
Терминал - tsarevmax@TsarevMax:~  
Файл  Правка  Вид  Терминал  Вкладки  Справка  
tsarevmax@TsarevMax:~$ sudo dnf -y install git  
[sudo] пароль для tsarevmax:  
Обновление и загрузка репозитория:  
Репозитории загружены.  
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.
```

делаю базовую настройку гита

```
Терминал - tsarevmax@TsarevMax:~  
Файл  Правка  Вид  Терминал  Вкладки  Справка  
tsarevmax@TsarevMax:~$ git config --global user.name "Tsarev Max"  
tsarevmax@TsarevMax:~$ git config --global user.email "floozy06@mail.ru"  
tsarevmax@TsarevMax:~$ git config --global core.quotepath false  
tsarevmax@TsarevMax:~$ git config --global init.defaultBranch master  
tsarevmax@TsarevMax:~$ git config --global core.autocrlf input  
tsarevmax@TsarevMax:~$ git config --global core.safecrlf warn  
tsarevmax@TsarevMax:~$
```

создаю ключи по алгоритму rsa с ключом размером 4096 бит



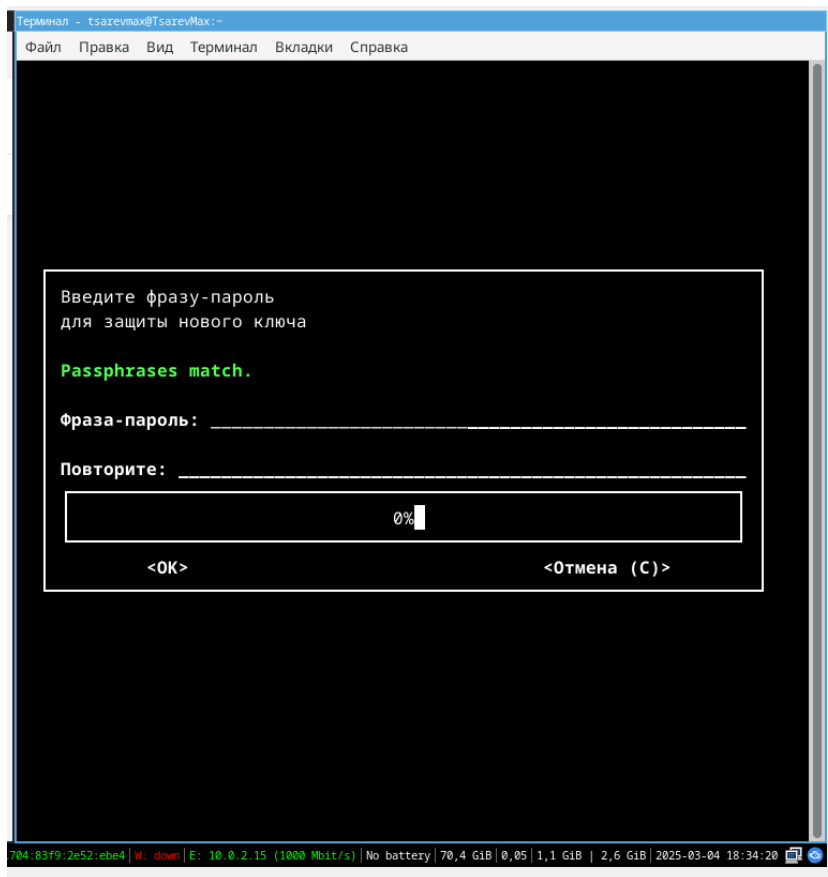
```
Терминал - tsarevmax@TsarevMax:~
Файл  Правка  Вид  Терминал  Вкладки  Справка
tsarevmax@TsarevMax:~$ gpg --full-generate-key
gpg (GnuPG) 2.4.5; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

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: TsarevMax
```

Пишу фразу-пароль



Выводим список ключей и копируем приватный ключ


```
Терминал - tsarevmax@TsarevMax:~  
Файл Правка Вид Терминал Вкладки Справка  
.  
Ваше полное имя: TsarevMax  
Адрес электронной почты: floozy06@mail.ru  
Примечание:  
Вы выбрали следующий идентификатор пользователя:  
"TsarevMax <floozy06@mail.ru>"  
  
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 энтро-  
пии.  
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 энтро-  
пии.  
gpg: /home/tsarevmax/.gnupg/trustdb.gpg: создана таблица доверия  
gpg: создан каталог '/home/tsarevmax/.gnupg/openpgp-revocs.d'  
gpg: сертификат отзыва записан в '/home/tsarevmax/.gnupg/openpgp-revocs.d  
/72B41B15CFDED76B22936275C5CA56F395695F91.rev'.  
открытый и секретный ключи созданы и подписаны.  
  
pub   rsa4096 2025-03-04 [SC]  
       72B41B15CFDED76B22936275C5CA56F395695F91  
uid           TsarevMax <floozy06@mail.ru>  
sub   rsa4096 2025-03-04 [E]  
  
tsarevmax@TsarevMax:~$
```

Захожу на ГитХаб и в настройках добавляю ключ

## GPG keys

New GPG key

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.

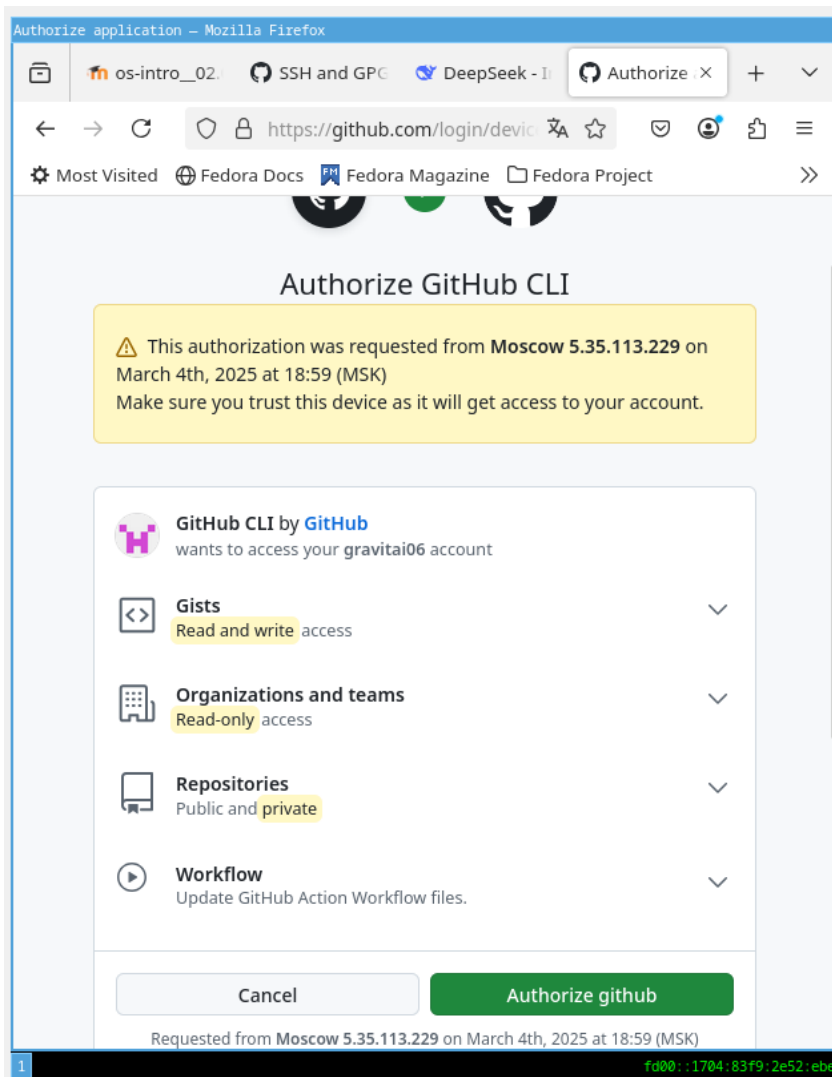


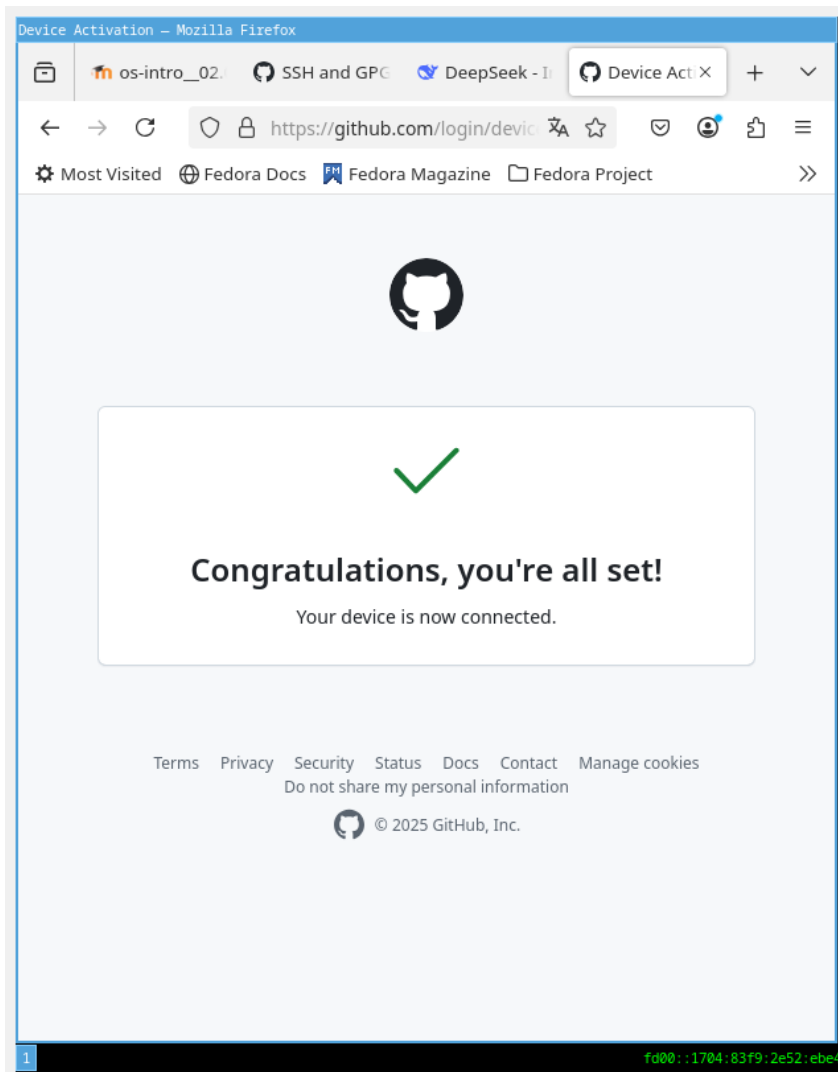
**fedora-lab**  
Email address: floozy06@mail.ru  
Key ID: C5CA56F395695F91  
Subkeys: 67F65A09E1579B10  
Added on Mar 4, 2025

Delete

Начинаю авторизацию через gh и в конце выбираю авторизоваться через браузер

```
Терминал - tsarevmax@TsarevMax:~  
Файл Правка Вид Терминал Вкладки Справка  
tsarevmax@TsarevMax:~$ git config --global user.signingkey C5CA56F395695F91  
tsarevmax@TsarevMax:~$ git config --global commit.gpgsign true  
tsarevmax@TsarevMax:~$ git config --global gpg.program $(which gpg2)  
tsarevmax@TsarevMax:~$
```





вижу сообщение об авторизации

```
терминал - tsarevmax@tsarevMax:~$  
Файл Правка Вид Терминал Вкладки Справка  
? How would you like to authenticate GitHub CLI? Login with a web browser  
! First copy your one-time code: D2CF-3517  
Press Enter to open https://github.com/login/device in your browser...  
✓ Authentication complete. - gh con  
fig set -h github.com git_protocol https  
✓ Configured git protocol  
! Authentication credentials saved in plain text  
✓ Logged in as gravitai06  
tsarevmax@TsarevMax:~$
```

Создаю репозиторий на основе шаблона

```
Терминал - tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2024-2025_os-intro
Файл  Правка  Вид  Терминал  Вкладки  Справка
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы$ ls
study_2024-2025_os-intro
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы$ cd stu
dy_2024-2025_os-intro/
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ ls
CHANGELOG.md  COURSE  Makefile      README.en.md  README.md
config        LICENSE  package.json  README.git-flow.md  template
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$
```

После созданию репозитория настраиваю файлы и в конце пушу в Гит

```
Терминал - tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2024-2025_os-intro
Файл  Правка  Вид  Терминал  Вкладки  Справка
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ rm package.json
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ echo os-intro > COURSE
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ make
Usage:
  make <target>

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

tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ git add .
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ git commit -am 'feat(main): make course structure'
>
>
> ^C
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ git commit -am 'feat(main): make course structure'

[master 2709763] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
tsarevmax@TsarevMax:~/work/study/2024-2025/Операционные системы/study_2
024-2025_os-intro$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
83f9:2e52:ebef | W: down | E: 10.0.2.15 (1000 Mbit/s) | No battery | 70,3 GiB | 0,05 | 1,3 GiB | 2,3 GiB | 2025-03-04 19:12:30
```

## 4 Выводы

Я изучил идеологию и применение средств контроля версий. Освоил умения по работе с git.