

Отчёт по лабораторной работе № 2

Операционные системы

Миронова Мария Вадимовна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Установка программного обеспечения	6
3.2	Базовая настройка git	7
3.3	Создали ключи ssh	8
3.4	Создали ключи pgr	9
3.5	Настройка github	11
3.6	Добавление PGP ключа в GitHub	12
3.7	Настройка автоматических подписей коммитов git	13
3.8	Настройка gh	14
3.9	Создание репозитория курса на основе шаблона	14
4	Выводы	15
5	Ответы на контрольные вопросы	16

Список иллюстраций

3.1	6
3.2	“gh”	7
3.3	7
3.4	7
3.5	8
3.6	8
3.7	8
3.8	9
3.9	9
3.10	10
3.11	11
3.12	11
3.13	12
3.14	12
3.15	12
3.16	13
3.17	13
3.18	13
3.19	14
3.20	14
3.21	14
3.22	14

1 Цель работы

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

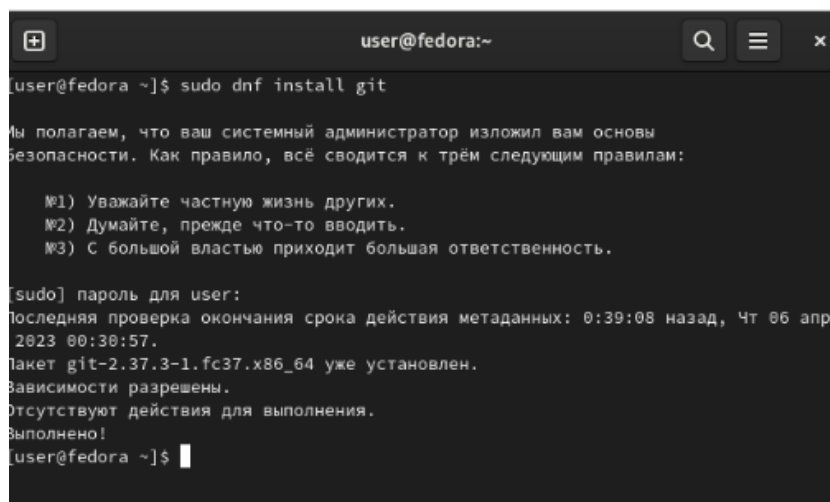
2 Задание

- Установить и настроить ПО для работы с git.

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Установили git:(рис. [3.1])



```
user@fedora:~  
[user@fedora ~]$ sudo dnf install git  
Мы полагаем, что ваш системный администратор изложил вам основы  
безопасности. Как правило, всё сводится к трём следующим правилам:  
  
№1) Уважайте частную жизнь других.  
№2) Думайте, прежде что-то вводить.  
№3) С большой властью приходит большая ответственность.  
  
[sudo] пароль для user:  
Последняя проверка окончания срока действия метаданных: 0:39:08 назад, Чт 06 апр  
2023 00:30:57.  
Пакет git-2.37.3-1.fc37.x86_64 уже установлен.  
Зависимости разрешены.  
Отсутствуют действия для выполнения.  
Выполнено!  
[user@fedora ~]$
```

Рис. 3.1: .

Установили gh:(рис. [3.2])

```
[user@fedora ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 0:42:50 назад, Чт 06 апр 2023 00:38:57.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh         x86_64       2.23.0-1.fc37  updates      8.3 М
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 8.3 М
Объем изменений: 42 М
Продолжить? [д/н]: д
Загрузка пакетов:
gh-2.23.0-1.fc37.x86_64.rpm          6.9 MB/s | 8.3 MB      00:01
=====
Общий размер          5.8 MB/s | 8.3 MB      00:01
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка           : 1/1
Установка            : gh-2.23.0-1.fc37.x86_64 1/1
Запуск скриплетов: gh-2.23.0-1.fc37.x86_64 1/1
Проверка             : gh-2.23.0-1.fc37.x86_64 1/1
Установлен:
gh-2.23.0-1.fc37.x86_64
Выполнено!
```

Рис. 3.2: “gh”

3.2 Базовая настройка git

Задали имя и email владельца репозитория: (рис. [3.3])

```
[user@fedora ~]$ git config --global user.name "Maria Mironova"
[user@fedora ~]$ git config --global user.email "unit.sb.disk@yandex.ru"
[user@fedora ~]$
```

Рис. 3.3: .

Настроили utf-8 в выводе сообщений git:(рис. [3.4])

```
[user@fedora ~]$ git config --global core.quotePath false
[user@fedora ~]$
```

Рис. 3.4: .

Настроили верификацию и подписание коммитов git. Задали имя начальной ветки (будем называть её master).(рис. [3.5])

```
[user@fedora ~]$ git config --global init.defaultBranch master  
[user@fedora ~]$
```

Рис. 3.5: .

Параметр autocrlf:(рис. [3.6])

```
[user@fedora ~]$ git config --global core.autocrlf input  
[user@fedora ~]$
```

Рис. 3.6: .

Параметр safecrlf: (рис. [3.7])

```
[user@fedora ~]$ git config --global core.safecrlf warn  
[user@fedora ~]$
```

Рис. 3.7: .

3.3 Создали ключи ssh

по алгоритму rsa с ключём размером 4096 бит: (рис. [3.8])


```

[user@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa
Your public key has been saved in /home/user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:S4LOYQi9Mp6XmrpLexztxVk6+40f6g4gy2heHZi32Dk user@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|
| .
| . .
| . oo. .
| o.o+=oo S
| +o+=Bo+O .
| .B.==E. =
| + *O .oo ..
| =B. o+++.
+----[SHA256]-----+
[user@fedora ~]$

```

Рис. 3.8: .

по алгоритму ed25519: (рис. [3.9])

```

[user@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/user/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_ed25519
Your public key has been saved in /home/user/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:f0m/ycmPkSgovORHSrSXuQifAiIYeShvntJhj2pffpY user@fedora
The key's randomart image is:
+--[ED25519 256]--+
|
| o
| = . o
| o+ E .So .
| + * o.+ =o . + .
| . * * +*..o + +
| .. = o=oo o o *
| oo.. .o. B..
+----[SHA256]-----+
[user@fedora ~]$

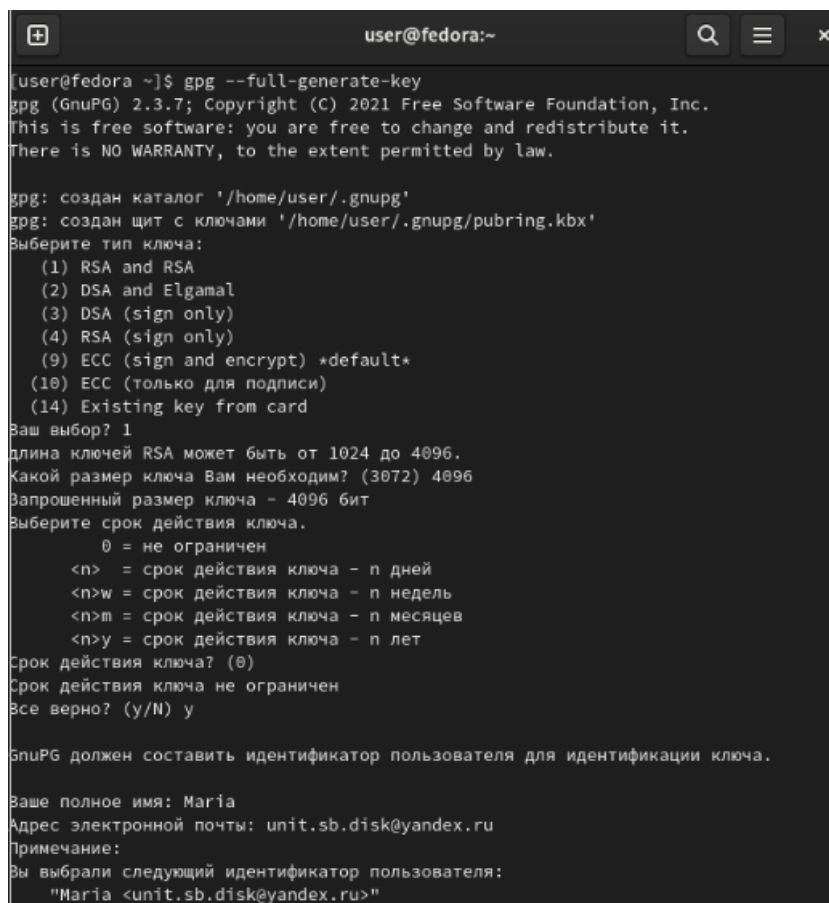
```

Рис. 3.9: .

3.4 Создали ключи рдр

Сгенерировали ключ (рис. [3.10])

Из предложенных опций выбирали: тип RSA and RSA; размер 4096; выбрали срок действия; значение по умолчанию — 0 (срок действия не истекает никогда). GPG запросил личную информацию, которая сохранится в ключе: Имя. Адрес электронной почты. При вводе email убедились, что он соответствует адресу, используемому на GitHub. (рис. [3.11])



```
user@fedora:~  
[user@fedora ~]$ gpg --full-generate-key  
gpg (GnuPG) 2.3.7; 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/user/.gnupg'  
gpg: создан щит с ключами '/home/user/.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)  
Срок действия ключа не ограничен  
Все верно? (y/N) y  
  
GnuPG должен составить идентификатор пользователя для идентификации ключа.  
  
Ваше полное имя: Maria  
Адрес электронной почты: unit.sb.disk@yandex.ru  
Примечание:  
Вы выбрали следующий идентификатор пользователя:  
"Maria <unit.sb.disk@yandex.ru>"
```

Рис. 3.10: .

```

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 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/user/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/user/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/user/.gnupg/openpgp-revocs.d/3696EBAC4DC
CB31CE23A88708526B517203B472.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2023-04-05 [SC]
       3696EBAC4DCCB31CE23A88708526B517203B472
uid           Maria <unit.sb.disk@yandex.ru>
sub   rsa4096 2023-04-05 [E]

[user@fedora ~]$

```

Рис. 3.11: .

3.5 Настройка github

Создайте учётную запись на github.com. (рис. [3.12])

Рис. 3.12: .

Заполните основные данные на github.com. (рис. [3.13])

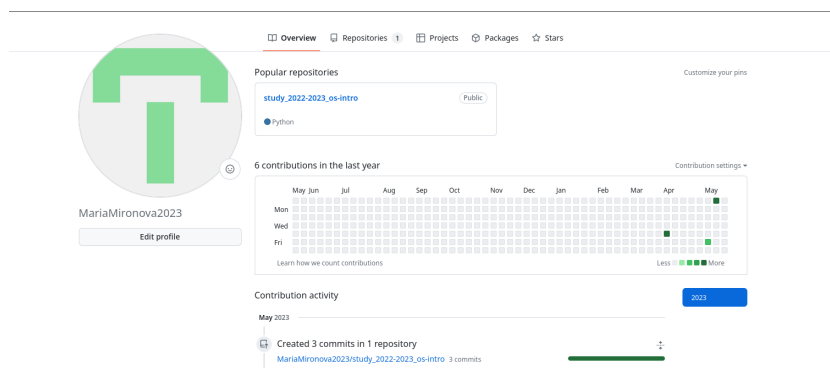


Рис. 3.13: .

3.6 Добавление PGP ключа в GitHub

Вывели список ключей и копировали отпечаток приватного ключа: (рис. [3.14])
Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

```
[user@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
/home/user/.gnupg/pubring.kbx
-----
sec   rsa4096/08526B517203B472 2023-04-05 [SC]
      3696EBAC4DCCBB31CE23A88708526B517203B472
uid   [ а6солютно ] Maria <unit.sb.disk@yandex.ru>
ssb   rsa4096/496FA6AE6DA6E6E7 2023-04-05 [E]
```

Рис. 3.14: .

Скопировали сгенерированный PGP ключ в буфер обмена: (рис. [3.15])

```
[user@fedora ~]$ gpg --armor --export 08526B517203B472 | xclip -sel clip
[user@fedora ~]$
```

Рис. 3.15: .


Перешли в настройки GitHub, нажали на кнопку New GPG key и вставили полученный ключ в поле ввода. (рис. [3.16], [3.17])

YkPxOGduKlIdJJRxPgvSLjt7BtOFTHPD1A6PTclqoty44p/UxspzKn8EuyBIRYU
iVruXw0xWZiHQ85Sa+P+9N8YLRQlbwouKHhDmGiklu6juGF1172FS5q4w05IOZYZZ
lIT2jxYY03PHFydCaFD5HRfklIKz5P1GzTC1g2FIFKz7XiVDRJsc36yIDQ/InU5KDK
VtrjrVmnS94ylTwhz8lKfdpqB75XqWxurFSlnP:BW+HvPC01zgMeGo7AJlC1b8z1
GkCMdpdBlk27lMaMoPvAG6F+vaDPuqhi6CqaWpo51Jv9PKEDLam25EturD+RHxPs
JRWW6EerHZq/u+yTfe64Be9QT4reEpiMOTbAlifbz/ChDEe01tYYGksTrkUD
=J5wN
-----END PGP PUBLIC KEY BLOCK-----

GPG keys

New GPG key

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



MariaMironova2023

Email address: uni1sb.disk@yandex.ru

Key ID: 08526B517203B472

GPG

Subkeys: 496FAGAE6DAGEE7

Added on Apr 6, 2023

Delete

[Learn how to generate a GPG key and add it to your account.](#)

3.7 Настройка автоматических подписей коммитов git

```
[user@fedora ~]$ git config --global user.signingkey 08526B517203B472
[user@fedora ~]$ git config --global commit.gpgsign true
[user@fedora ~]$ git config --global gpg.program $(which gpg2)
[user@fedora ~]$
```

3.8 Настройка gh

Авторизовались в gh. (рис. [3.19]) Утилита задали несколько наводящих вопросов.

```
[user@fedora ~]$ gh auth login
? You're already logged into github.com. Do you want to re-authenticate? Yes
? What is your preferred protocol for Git operations? HTTPS
? How would you like to authenticate GitHub CLI? Login with a web browser
```

Рис. 3.19: .

3.9 Создание репозитория курса на основе шаблона

Создали шаблон рабочего пространства. (рис. [3.20], [3.21], [3.22])

```
[user@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[user@fedora ~]$ cd ~/work/study/2022-2023/"Операционные системы"
```

Рис. 3.20: .

```
[user@fedora Операционные системы]$ gh repo create study_2022-2023_os-intro --te
mplate=yamadharma/course-directory-student-template --public
✓ Created repository MariaMironova2023/study_2022-2023_os-intro on GitHub
```

Рис. 3.21: .

```
[user@fedora Операционные системы]$ git clone --recursive git@github.com:Mar
ronova2023/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
```

Рис. 3.22: .

4 Выводы

В ходе выполнения данной лабораторной работы была изучена идеология и применение средств контроля версий и освоены умения по работе с git.

5 Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Система управления версиями (также используется определение «система контроля версий», от англ. Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. 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.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище (repository), или репозиторий, — место хранения файлов и их версий, служебной информации. Версия (revision), или ревизия, — состояние всего хранилища или отдельных файлов в момент времени («пункт истории»). Commit («трудовой вклад», не переводится) — процесс создания новой версии; иногда синоним версии. Рабочая копия (working copy) — текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Децентрализованные VCS: У каждого пользователя свой вариант (возможно не один) репозитория. Присутствует возможность добавлять и забирать изменения из любого репозитория (Git, Mercurial, Bazaar).

Централизованные VCS : Одно основное хранилище всего проекта Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно (Subversion, CVS, TFS, VAULT, AccuRev)

4. Опишите действия с VCS при единоличной работе с хранилищем.
5. Опишите порядок работы с общим хранилищем VCS.
6. 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.
7. Назовите и дайте краткую характеристику командам git. git init - создание репозитория git add (имена файлов) - Добавляет файлы в индекс git commit – выполняет коммит проиндексированных файлов в репозиторий git status – показывает какие файлы изменились между текущей стадией и HEAD. Файлы разделяются на 3 категории: новые файлы, измененные файлы, добавленные новые файлы git checkout (sha1 или метка) - получение указанной версии файла git push – отправка изменений в удаленный репозиторий git fetch – получение изменений из удаленного репозитория git clone (remote url) - клонирование удаленного репозитория себе
8. Приведите примеры использования при работе с локальным и удалённым репозиториями.
9. 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. Показать все ветки, существующие в репозитории git branch. Создать ветку git branch имя.

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.

10. Как и зачем можно игнорировать некоторые файлы при commit? Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты. Вот некоторые распространенные примеры таких файлов:

кэши зависимостей, например содержимое `node_modules` или `packages`; скомпилированный код, например файлы `.o`, `.рус` и `.class` ; каталоги для выходных данных сборки, например `bin`, `out` или `target`; файлы, сгенерированные во время выполнения, например `.log`, `.lock` или `.tmp`; скрытые системные файлы, например `.DS_Store` или `Thumbs.db`; личные файлы конфигурации IDE, например `.idea.workspace.xml`.