**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятности

**ОТЧЕТ  
по лабораторной работе №2**

темА «Тема лабораторной работы»  
по дисциплине «Операционные системы»

Выполнил:

Студент группы НПИбд-02-21

Хусаинова Динара Айратовна

« 21 » апрель 2022г.

Москва 2022

Цель работы: изучить идеологию и применение средств контроля версий и освоить умения по работе с git.

Ход работы

1. Создаем учетную запись на GitHub.

Устанавливаем git-flow в Fedora Linux (рис. 1.1).

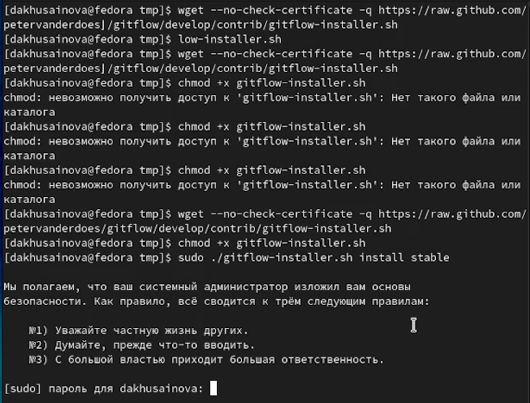


Рис. 1.1

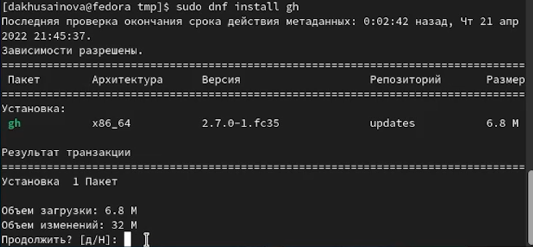
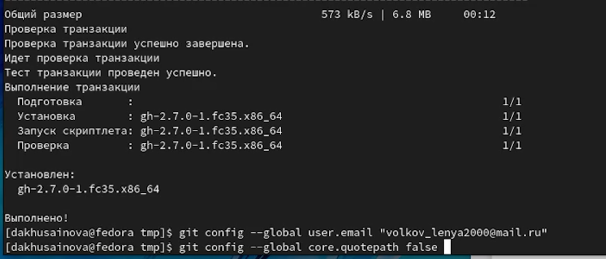
Устанавливаем gh в Fedora Linux с помощью команды sudo( рис. 1.2).

Рис. 1.2

Далее следует базовая настройка git. Используем имя, которое мы вводили при регистрации на GitHub и ту же электронную почту(рис. 1.3).

Рис. 1.3

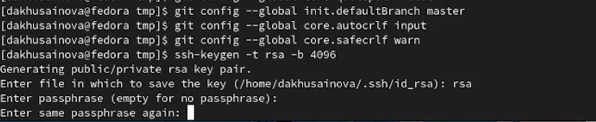
Настраиваем utf-8 в выводе сообщений git (рис. 1.4).

Рис. 1.4

При генерации ключа выбираем необходимые опции (рис. 1.5,1.6).

– 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).

– Адрес электронной почты.

– При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.

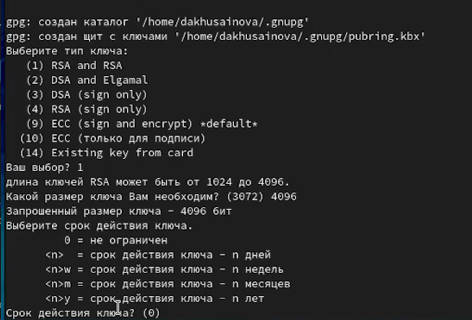
– Комментарий.

Рис. 1.5

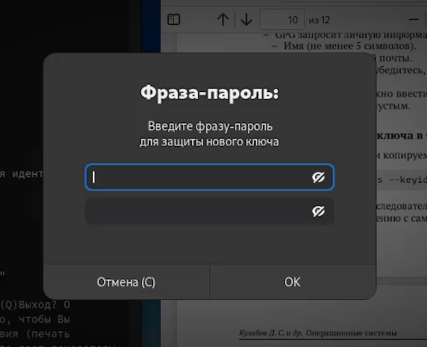


Рис. 1.6

Выводим список ключей и копируем отпечаток приватного ключа, устанавливаем необходимый пакет для дальнейших действий с ключом (рис. 1.7 -1.8).



Рис. 1.7

Рис. 1.8

Заходим на сам сайт GitHub, копируем сгенерированный PGP ключ в буфер обмена и вставляем полученный ключ в поле ввода (рис.1.9,1.10,1.11).



Рис. 1.9

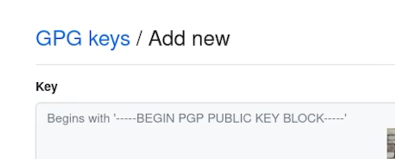


Рис. 1.10

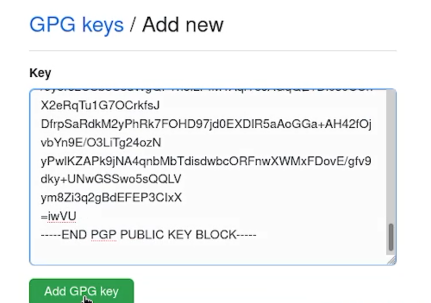
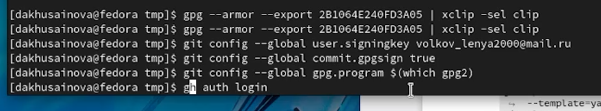


Рис. 1.11

Настраиваем автоматические подписи коммитов git, отвечаем на вопросы, выбирая GitHub,SSH,yes,вводим пароль, «залогиниться через браузер»(рис.1.12-1.13).

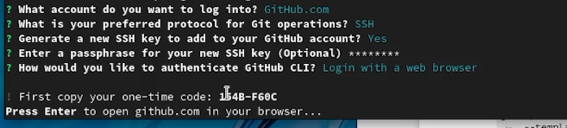
Рис.1.12.

Рис. 1.13

Получаем код в терминале и вводим его на сайте( рис.1.14).

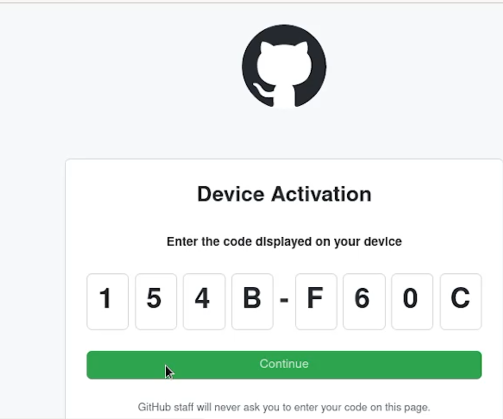
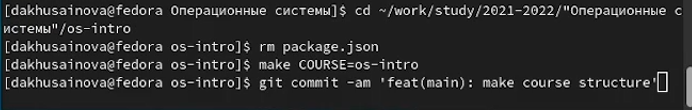


Рис.1.14

Переходим в каталог курса, удаляем ненужные файлы, создаем, создаем каталог os-intro и отправляем файлы на сервер(рис. 1.15,1.16).

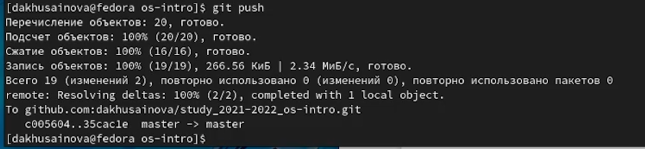
Рис.1.15

Рис.1.16

Контрольные вопросы

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

Система контроля версий (Version Control System, VCS) — программное обеспечение для облегчения работы с изменяющейся информацией. u VCS 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.

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

Commit (Коммит) – сохранение изменений в репозиторий.  
Хранилище (repository), или репозитарий, — место хранения всех версий и служебной информации.

История VCS – история изменений файла или папки.

Рабочая копия (working copy) — текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, измененных.

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

Централизованные системы — это системы, которые используют архитектуру клиент / сервер, где один или несколько клиентских узлов напрямую подключены к центральному серверу. Это наиболее часто используемый тип систем во многих организациях, где клиент отправляет запрос на сервер компании и получает ответ.

Wikipedia. Рассмотрим огромный сервер, на который мы отправляем наши запросы, и сервер отвечает запрашиваемой статьей. Предположим, мы ввели поисковый запрос «нездоровая пища» в строке поиска Википедии. Этот поисковый запрос отправляется как запрос на серверы Википедии (в основном, расположенные в штате Вирджиния, США), которые затем возвращают статьи, основанные на релевантности. В этой ситуации мы являемся клиентским узлом, серверы Википедии являются центральным сервером.

В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы является совокупностью решений отдельных узлов. Обратите внимание, что нет единого объекта, который получает и отвечает на запрос.

Bitcoin. Давайте возьмем биткойны, например, потому что это самый популярный пример использования децентрализованных систем. Ни одна организация / организация не владеет сетью биткойнов. Сеть представляет собой сумму всех узлов, которые общаются друг с другом для поддержания количества биткойнов, которое есть у каждого владельца счета.

1. . Опишите действия с VCS при единоличной работе с хранилищем.

Вы можете скачать себе файл и отключить интернет. Файл останется на компьютере и с ним можно будет работать. А когда интернет появится, он перенесется на сервер. И если кто-то еще трогал тот же файл, появится конфликт, который придется разрешить человеку.

Суть системы такова, что все файлы хранятся на сервере, а клиенты подключаются к серверу, скачивают актуальное состояние файлов, вносят правки и отсылают изменения обратно на сервер, чтобы другие клиенты могли работать уже и с их правками.

5. Каковы основные задачи, решаемые инструментальным средством git?

Git — это система управления версиями. 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.

6. Назовите и дайте краткую характеристику командам git.

Проверьте версию Git с помощью следующей команды (это также поможет убедиться, что Git установлен).

git --version

Вы можете инициализировать ваш текущий рабочий каталог как Git-репозиторий с помощью команды init.

git init

Чтобы скопировать существующий удаленный Git-репозиторий, используйте команду git clone с URL-адресом репозитория или расположением сервера (в последнем случае нужно использовать ssh).

git clone https://www.github.com/username/repo-name

Просмотреть список текущих удалённых репозиториев Git можно с помощью команды:

git remote

Для более подробного вывода используйте флаг -v.

git remote -v

Добавьте Git upstream: можно указать URL-адрес, а если он размещен на сервере, нужно подключиться по ssh.

git remote add upstream https://www.github.com/username/repo-name

7. Приведите примеры использования при работе с локальным и удалённым репозиториями.

Для того, чтобы внести вклад в какой-либо Git-проект, вам необходимо уметь работать с удалёнными репозиториями. Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети. У вас может быть несколько удалённых репозиториев, каждый из которых может быть доступен для чтения или для чтения-записи. Взаимодействие с другими пользователями предполагает управление удалёнными репозиториями, а также отправку и получение данных из них. Управление репозиториями включает в себя как умение добавлять новые, так и умение удалять устаревшие репозитории, а также умение управлять различными удалёнными ветками, объявлять их отслеживаемыми или нет и так далее. В данном разделе мы рассмотрим некоторые из этих навыков.

8. 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 репозитарии git branch. Создать ветку git branch имя.

9. 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.

Вывод: изучили идеологию и применение средств контроля версий и освоили умения по работе с git.