РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Операционные системы

Студент: Репина Ангелина Олеговна

Группа: НПМбд-03-21

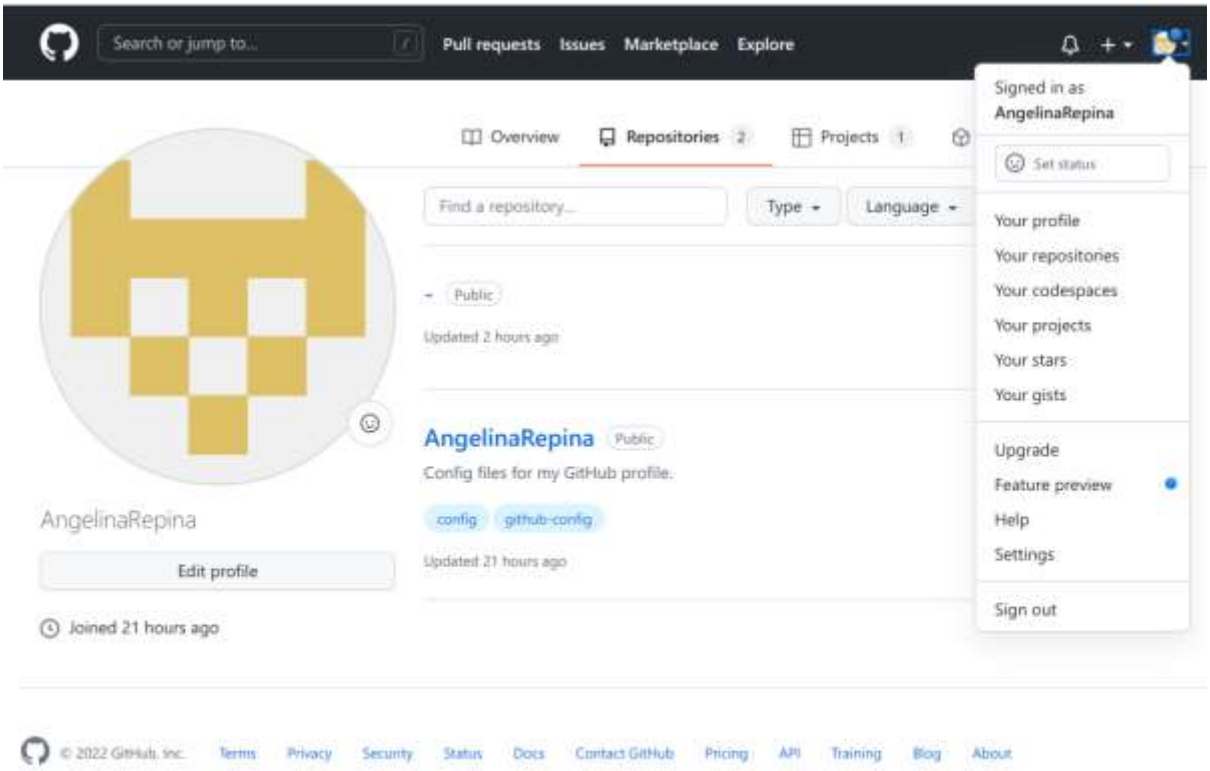
МОСКВА

2022 г.

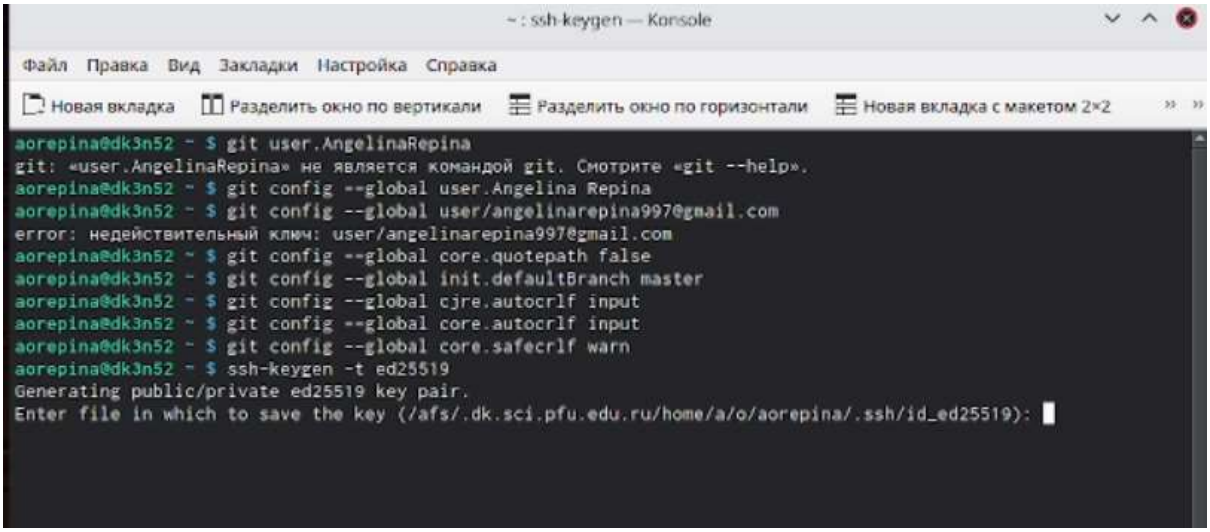
Цель: изучение идеологии и применение контроля версий, освоение умений по работе с git.

Ход работы:

1) Знакомимся с пунктами 2.1-2.4
2) Начинаем делать задание к теории. Первым делом создаём учётную запись на гитхабе и заполняем основные данные.



3) Делаем базовую настройку git. Задаём имя и емайл владельца репозитория, настраиваем utf-8 в выводе сообщений git, настраиваем верификацию и подписание коммитов git и задаём имя начальной ветки (назовём её master), параметр autocrlf, safecrlf.



4) Создаём ключи ssh по алгоритму ed25519.

```
aorepina@dk3n52 ~ $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/a/o/aorepina/.ssh/id_ed25519): public
public already exists.
Overwrite (y/n)? n
aorepina@dk3n52 ~ $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/a/o/aorepina/.ssh/id_ed25519): keygen
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in keygen
Your public key has been saved in keygen.pub
The key fingerprint is:
SHA256:75WczPyoS9pmJp6m2p5v/a6o2ZBDWgUg6y/Vh4IgM1E aorepina@dk3n52
The key's randomart image is:
+--[ED25519 256]--+
|..E ...          |
| . o   .         |
|= .     .        |
|.= . . o         |
|  o o = S        |
|   o = o . = o   |
|  . o *  .o 0    |
|   . . ++B=. o   |
|    .oBBB=*=* .  |
+----[SHA256]-----+
aorepina@dk3n52 ~ $ gpg --f
```

5) Создаём ключ pgp. Генерируем ключ, опции выбираем по лабораторной работе 2.

```
gpg (GnuPG) 2.2.33; 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.

Выберите тип ключа:
   (1) RSA и RSA (по умолчанию)
   (2) DSA и Elgamal
   (3) DSA (только для подписи)
   (4) RSA (только для подписи)
  (14) Имеющийся на карте ключ
Ваш выбор?
```

```
Выберите тип ключа:
   (1) RSA и RSA (по умолчанию)
   (2) DSA и Elgamal
   (3) DSA (только для подписи)
   (4) RSA (только для подписи)
  (14) Имеющийся на карте ключ
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: AngelinaRepina
Адрес электронной почты: angelinarepina997@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
    "AngelinaRepina <angelinarepina997@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 
```

6) Добавляем pgp ключ в ГитХаб. Выводим список ключей и копируем отпечаток приватного ключа. Копируем сгенерированный ключ в буфер обмена, вставляем его в ГитХаб.

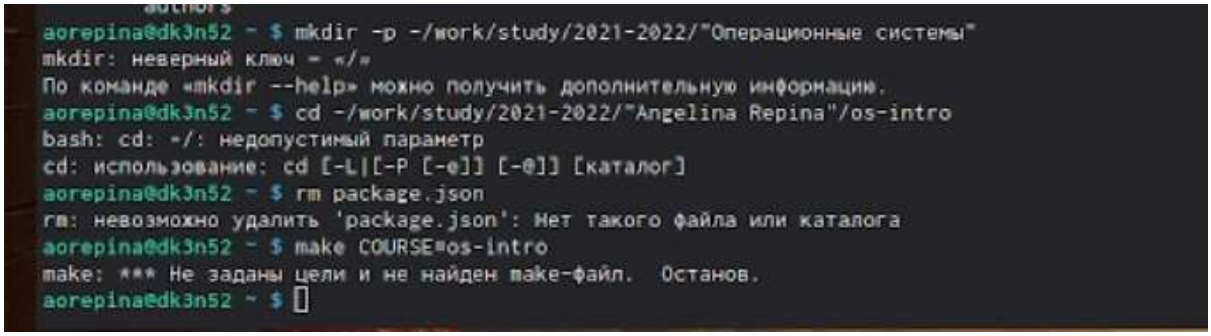
```
aorepina@dk3n52 ~ $ gpg --list-secret-keys --keyvid-format LONG
недопустимый параметр "--keyvid-format"
aorepina@dk3n52 ~ $ gpg --list-secret-keys A391363E330B9D43
gpg: проверка таблицы доверия
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: глубина: 0  достоверных:   1  подписанных:   0  доверие: 0-, 0q, 0n, 0m, 0f, 1u
sec   rsa4096 2022-04-21 [SC]
      596C15FA8FA1BD1EDBA488BCA391363E330B9D43
uid         [  абсолютно ] AngelinaRepina <angelinarepina997@gmail.com>
ssb   rsa4096 2022-04-21 [E]

aorepina@dk3n52 ~ $ 
```

```
aorepina@dk3n52 ~ $ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
aorepina@dk3n52 ~ $ gpg --armor --export A391363E330B9D43 | xclip -sel clip
aorepina@dk3n52 ~ $ 
```

7) Настраиваем автоматические подписи коммитов.

```
aorepina@dk3n52 ~ $ git config --global user.singinkey A391363E330B9D43
aorepina@dk3n52 ~ $ git config --global commit.gpgsign true
aorepina@dk3n52 ~ $ git config --global gpg.program $(which gpg2)
```

8) Создала репозиторий на основе шаблона, настроила каталог курса, сделала необходимые операции.

9) 

Ответы на контрольные вопросы:

1) Система контроля версий представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Системы контроля версий применяются при работе нескольких человек над проектом.
2) 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 дельтакомпрессию — 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.
3) Централизованные системы - системы, которые используют архитектуру клиент сервер, где один или несколько клиентских узлов напрямую подключены к центральному серверу (Википедия).

В децентрализованных системах каждый узел принимает своё собственное решение. Конечное поведение системы является совокупностью решений отдельных узлов (Биткоин).

4) Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию. указав имя и емайл владельца репозиториев. Для инициализации локального репозитория необходимо ввести команду с командной строке.
5) Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей, сохранить их, и , скопировав из локальной консоли ключ в буфер обмена вставляем ключ в поле на сайте.
6) Две основные задачи - хранить иенформацию о всех изменениях в вашем коде и обеспечение удобства командной работы над кодом.
7) Наиболее часто используемые команды git: – создание основного дерева репозитория: 1 git init – получение обновлений (изменений) текущего дерева из центрального репозитория: 1 git pull – отправка всех произведённых изменений локального дерева в центральный репозиторий: 1 git push – просмотр списка изменённых файлов в текущей директории: 1 git status – просмотр текущих изменения: 1 git diff – сохранение текущих изменений: – добавить все изменённые и/или созданные файлы и/или каталоги: 1 git add . – добавить конкретные изменённые и/или созданные файлы и/или каталоги: 1 git add имена_файлов – удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): 1 git rm имена_файлов – сохранение добавленных изменений: – сохранить все добавленные изменения и все изменённые файлы: 1 git commit -am 'Описание коммита' – сохранить добавленные изменения с внесением комментария через встроенный
git commit – создание новой ветки, базирующейся на текущей: 1 git checkout -b имя_ветки – переключение на некоторую ветку: 1 git checkout имя_ветки (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) – отправка изменений конкретной ветки в центральный репозиторий: 1 git push origin имя_ветки – слияние ветки с текущим деревом: 1 git merge --no-ff имя_ветки – удаление ветки: – удаление локальной уже слитой с основным деревом ветки: 1 git branch -d имя_ветки – принудительное удаление локальной ветки: 1 git branch -D имя_ветки – удаление ветки с центрального репозитория: 1 git push origin :имя_ветки
8) Использования гит при работе с локальными репозиториями (добавления текстового документа в локальный репозиторий):
git add hello.txt
git commit -am
9) Проблемы, которые решают ветки гит: нужно постоянно создавать архивы с рабочим кодом, сложно переключаться между архивами, сложно перетаскивать изменения между архивами, легко что-то напутать или потерять
10) Во время работы над проектом так или иначе могут создаваться файлы, которые не требуется добавлять в последствии в репозиторий.

Например, временные файлы, создаваемые редакторами, или объектные файлы, создаваемые компиляторами. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл .gitignore с помощью сервисов. Для этого сначала нужно получить список имеющихся шаблонов: 1 curl -L -s https://www.gitignore.io/api/list Затем скачать шаблон, например, для C и C++ 1 curl -L -s https://www.gitignore.io/api/c >> .gitignore 2 curl -L -s https://www.gitignore.io/api/c++ >> .gitignore

Вывод: в ходе лабораторной работы я изучила идеологию и применение средств контроля версий и освоила умения по работе с гит.