РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина: Основы информационной безопасности

Студент: ГЕОРГЕС ГедеонГруппа: НПМбд-02-21Студ. бил.:№ 1032204593

МОСКВА 2024 г.

Цель работы :

Целью данной работы является приобретение практических навыковустановки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

Задание :

Часть 11.Установка и настройка виртуальной машины VirtualBox (ОС Linux)

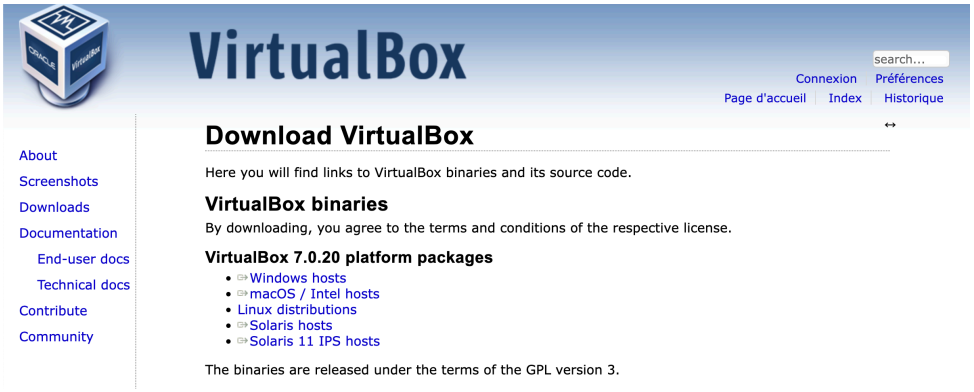
Часть 21.Создаем базовую конфигурацию для работы в git.2.Создаем ключ SSH.3.Создаем ключ PGP.4.Настройм установку git.5.Зарегистрировался на GitHub.

Часть 31.Подготовлено лабораторными работами в формате Markdown. 2. Файл предзагружен в 3-х форматах: pdf, docx и md.

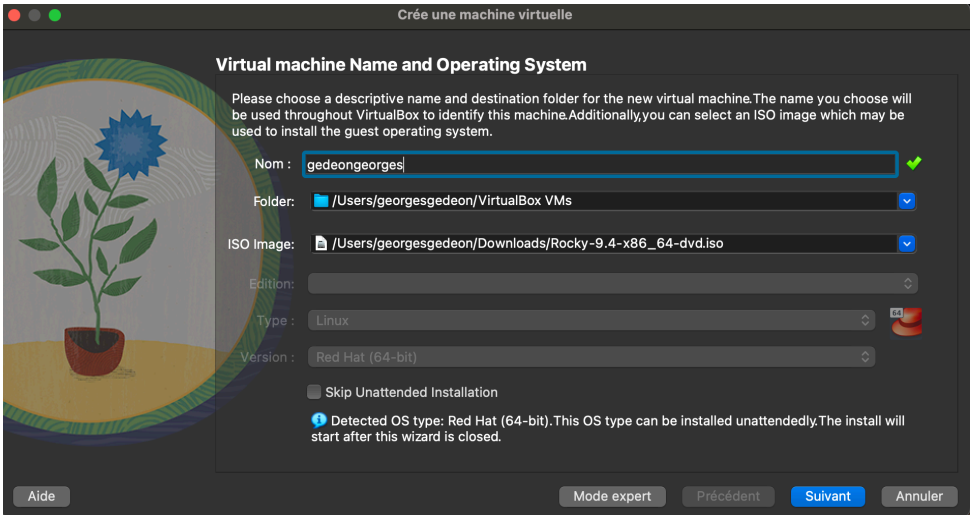
Выполнение лабораторной работы

Часть 1. Установка и настройка Виртуальной машины VirtualBox

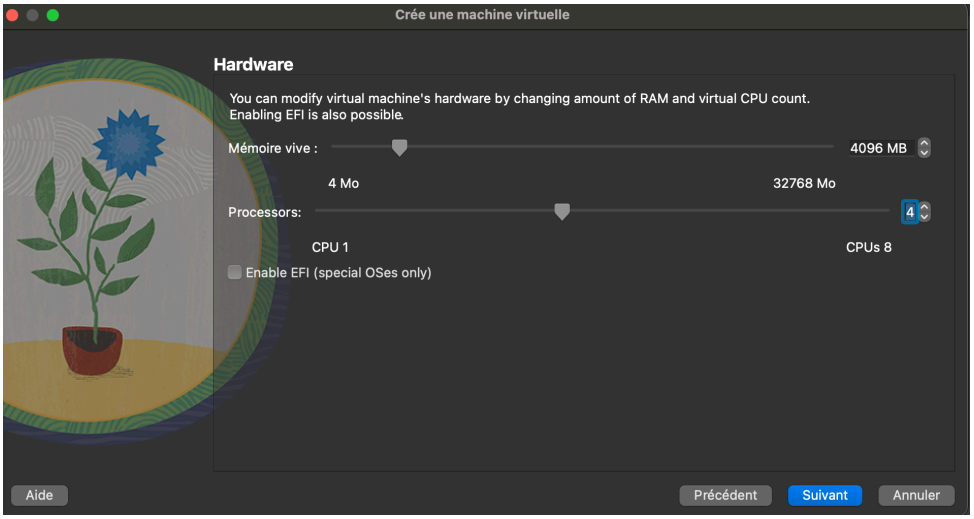
1)Зайдите на официальный сайт VirtualBox и скачайте файл. Установите VirtualBox на свой компьютер. Я часто использую Linux.



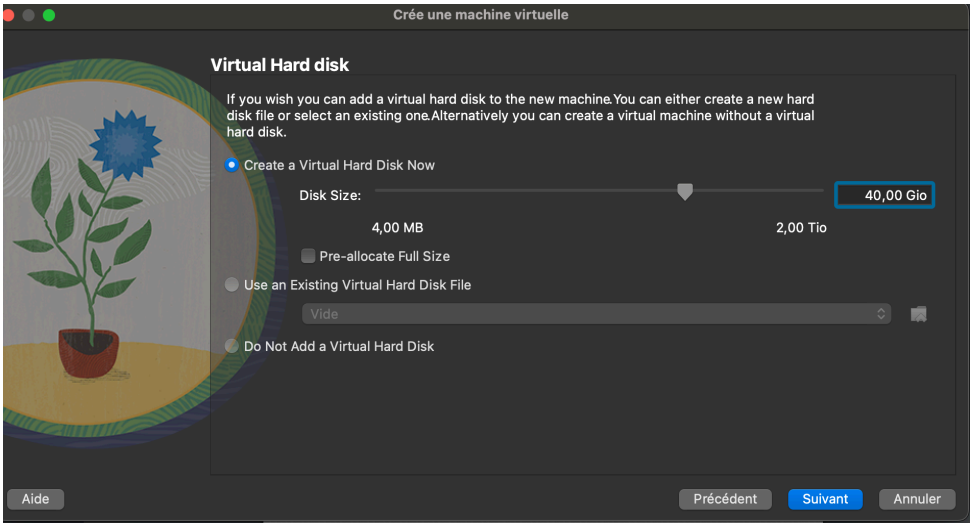
2)Создайте виртуальную машину. Для этого нажмите «Создать». Имя нашей виртуальной машины — GedeonGeorges. Выберите наш образ диска Rocky Linux. Чтобы он распознал его как образ, сначала скачиваем UltraISO.



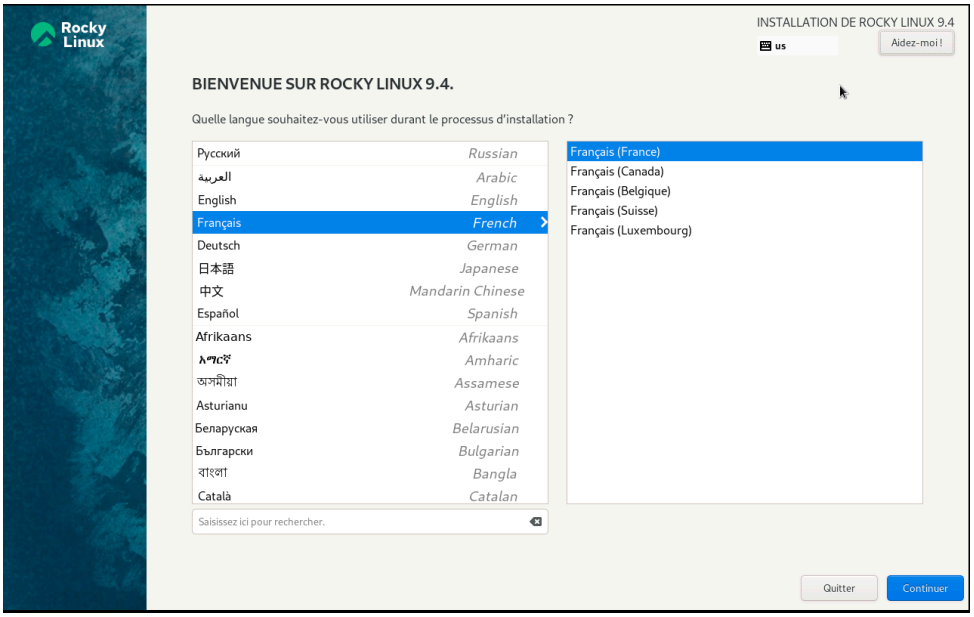
3)Виртуальной машине требуется оперативная память. Выделяем 4096 МБ, это половина основной оперативной памяти. Мы также даем ему 4 процессора.



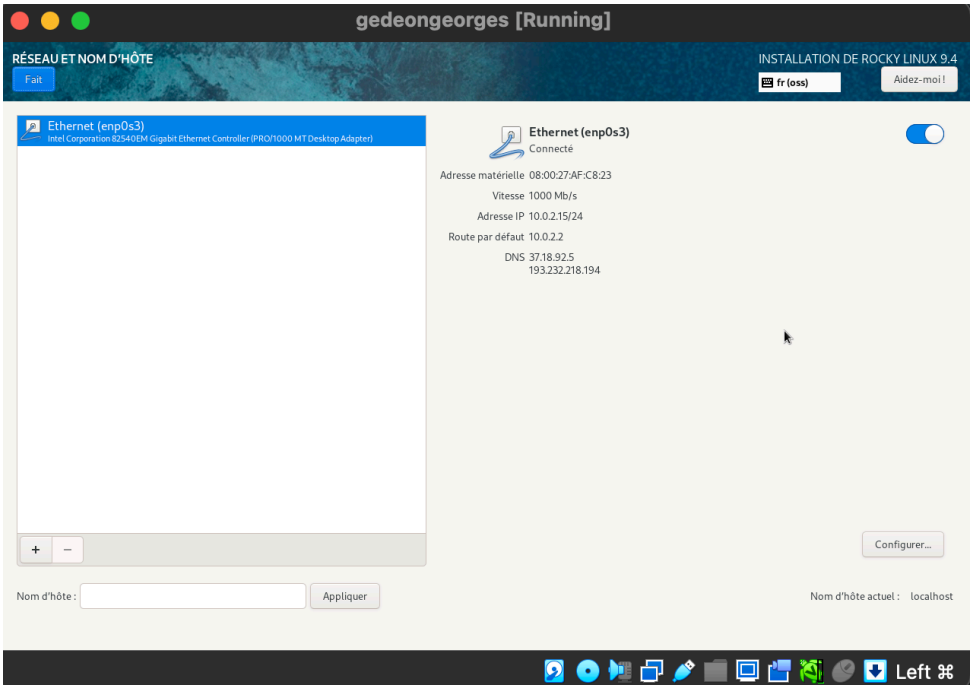
4)Выделяем под виртуальную машину 40 ГБ памяти.



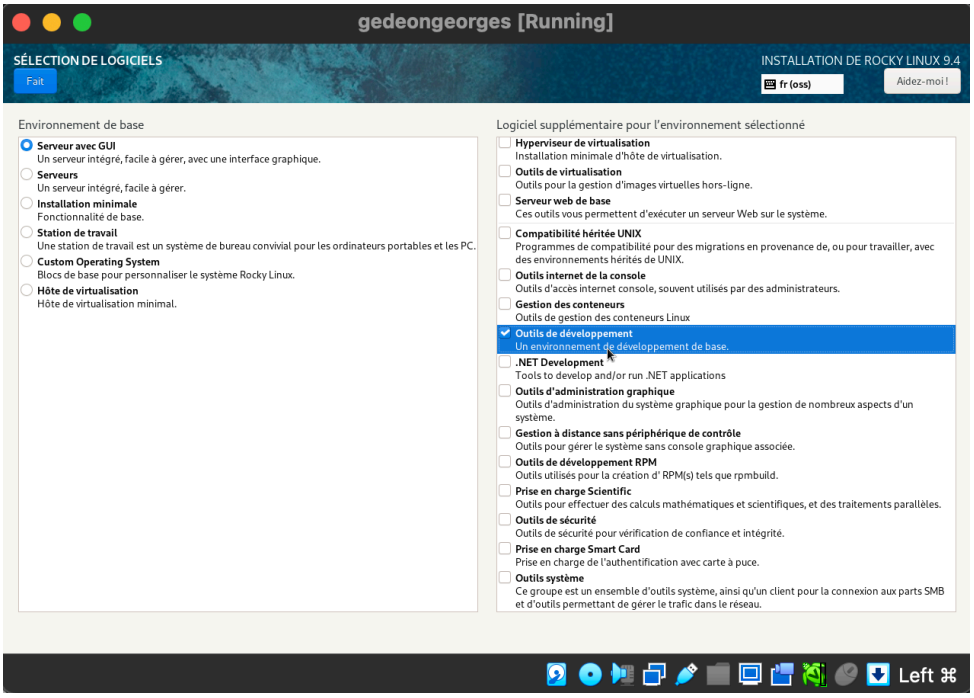
5)Включаем виртуальную машину, выбор языка.



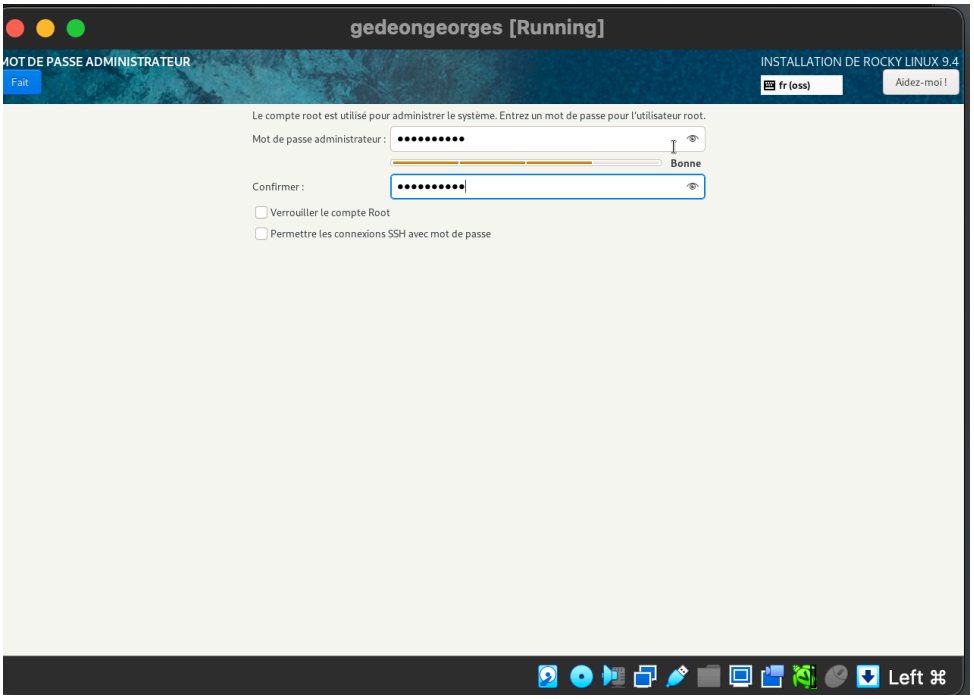
6)Загрузите сеть и следуйте за ней.



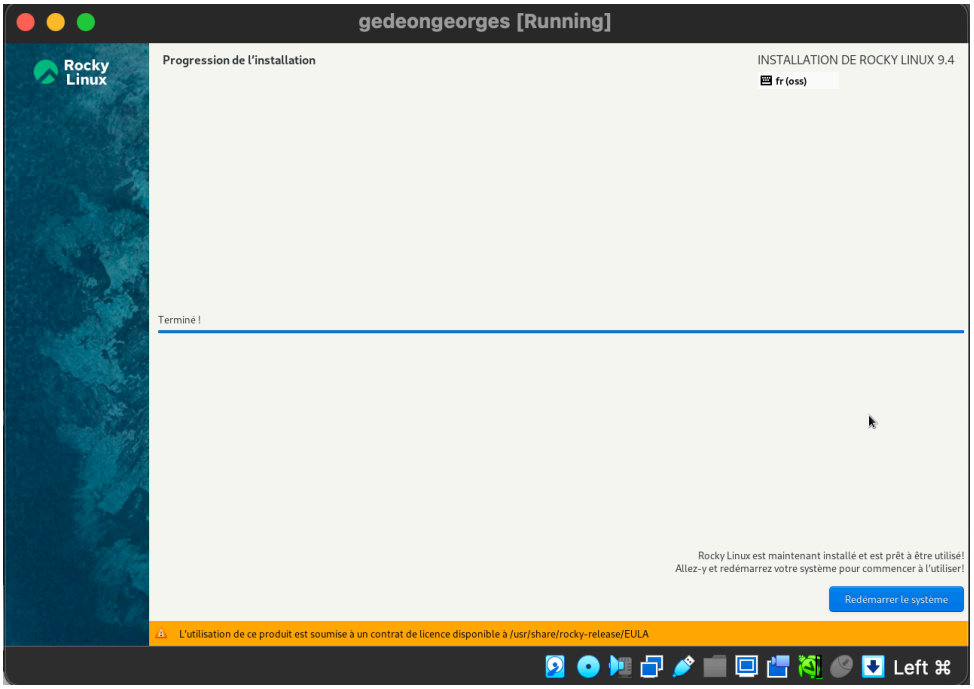
7)Базовое окружение выбираем Сервер с GUI.



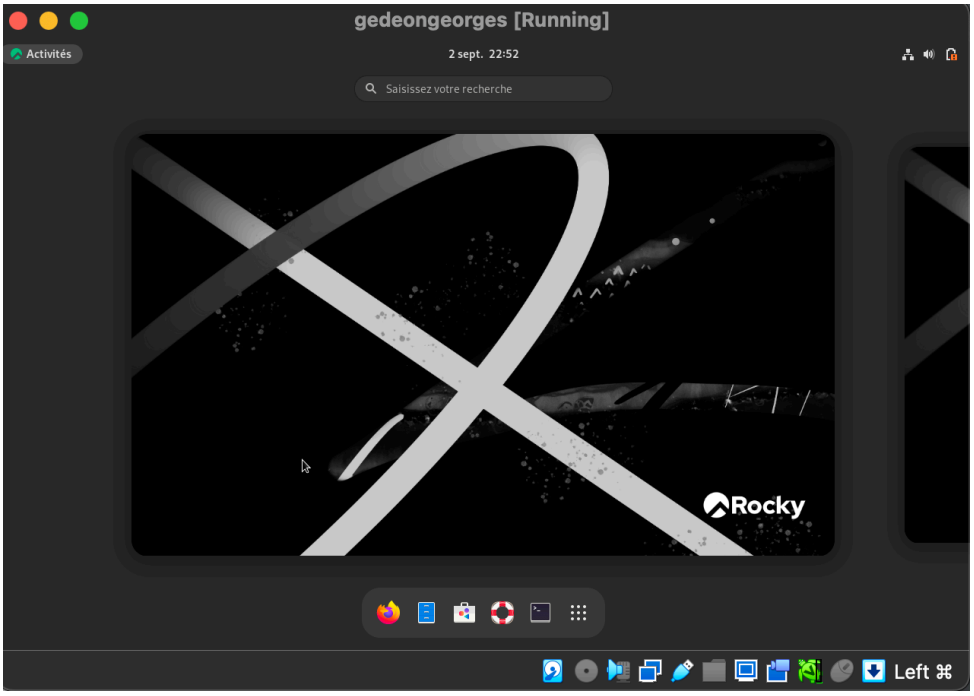
8)Установите пароль для пользователя root.



9)Установка завершена. Перезагрузить.

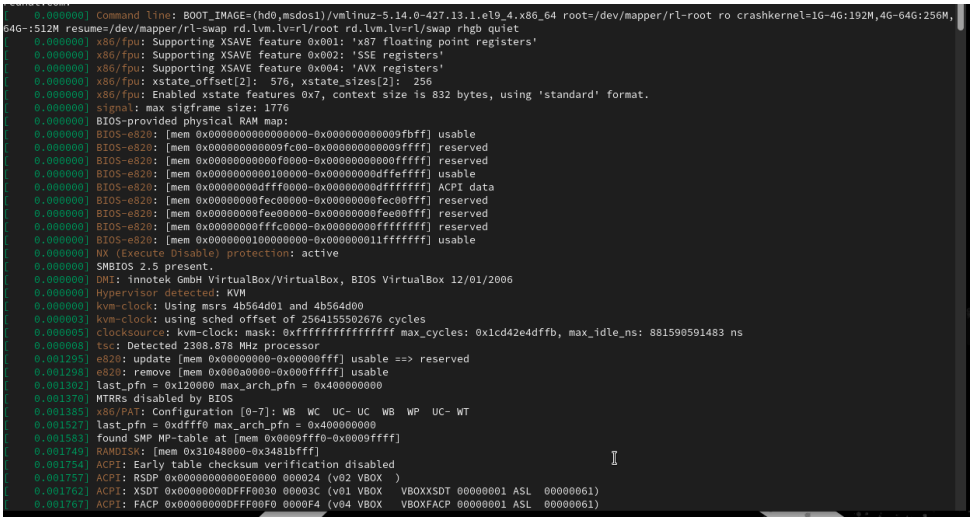


10)Создайте систему Rocky Linux. Установка завершена!



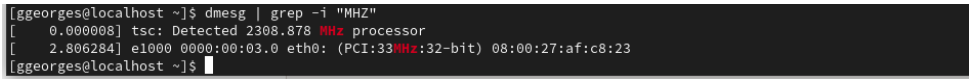
ДОМАШНЕЕ ЗАДАНИЕ №1:

1)Окно терминала подключается к системе, запустивкоманда dmesg.



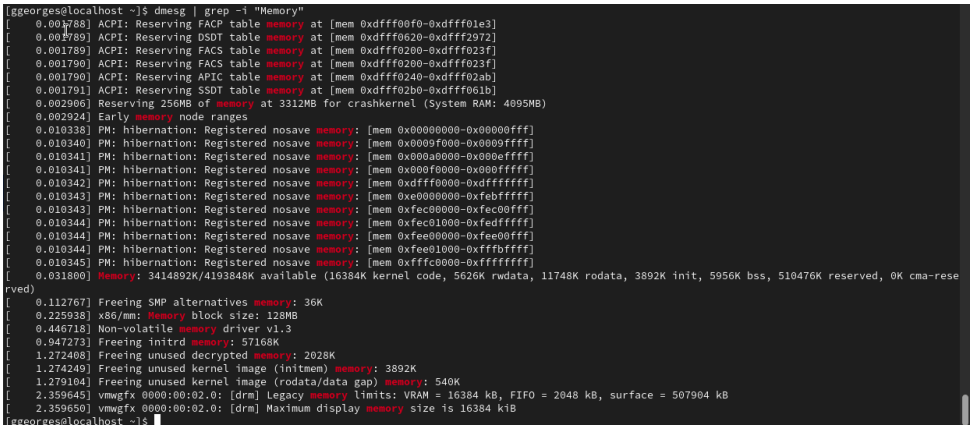
Вы можете добавить ссылку на grep:dmesg | grep -i "то, что мы ищем"Команда grep используется для поиска данных.a. Версия ядра Linux (версия Linux).

b. Частота процессора (Detected MHz processor).



c. Модель процессора (CPU0)

d. Объем доступной оперативной памяти (Memory available)



e. Тип обнаруженного гипервизора (Hypervisor detected).

```
[ggeorges@localhost ~]$ dmesg | grep -i "Hypervisor detected"
[    0.000000] Hypervisor detected: KVM
[ggeorges@localhost ~]$
```

f. Тип файловой системы корневого раздела. Последовательность монтированияфайловых систем

```
[ggeorges@localhost ~]$ dmesg | grep -i "Mount"
[    0.099770] Mount-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[    0.099778] Mountpoint-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
[    4.285938] XFS (dm-0): Mounting V5 Filesystem 2020d098-018e-4bff-a410-aa13d910fbcb
[    4.318355] XFS (dm-0): Ending clean mount
[    5.487723] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[    5.513005] systemd[1]: Mounting Huge Pages File System...
[    5.514887] systemd[1]: Mounting POSIX Message Queue File System...
[    5.517547] systemd[1]: Mounting Kernel Debug File System...
[    5.520576] systemd[1]: Mounting Kernel Trace File System...
[    5.580535] systemd[1]: Starting Remount Root and Kernel File Systems...
[    5.593904] systemd[1]: Mounted Huge Pages File System.
[    5.594364] systemd[1]: Mounted POSIX Message Queue File System.
[    5.594619] systemd[1]: Mounted Kernel Debug File System.
[    5.595146] systemd[1]: Mounted Kernel Trace File System.
[    5.619679] systemd[1]: Mounting FUSE Control File System...
[    5.624511] systemd[1]: Mounting Kernel Configuration File System...
[    7.320167] XFS (sda1): Mounting V5 Filesystem b173229b-baba-4e2e-bee2-2bb8cd7ea27c
[    7.799743] XFS (sda1): Ending clean mount
[ggeorges@localhost ~]$
```

КОНТРОЛЬНЫЕ ВОПРОСЫ №1

1)Учётная запись пользователя содержит сведения, необходимые для идентификации пользователя при подключении к системе, такие как имя пользователя, имя хоста и пароль.

2)Команды терминала:a.Для получения справки используется ключ –help или команда man. Например, ls –help или man ls.b.Для перемещения по файловой системе используется команда cd. Например cd ~.c.Для просмотра содержимого каталога используется команда ls. Например ls ~/work.d.Для определения объёма каталога используется команда du.e.Для создания каталогов используется mkdir, для удаления пустых каталогов используется rmdir. Для создания файлов используется touch, для удаления файлов и каталог используется rm.f.Для задания прав используется команда chmod. Например, chmod u-w test.txt.g.Для просмотра истории команд используется команда history.

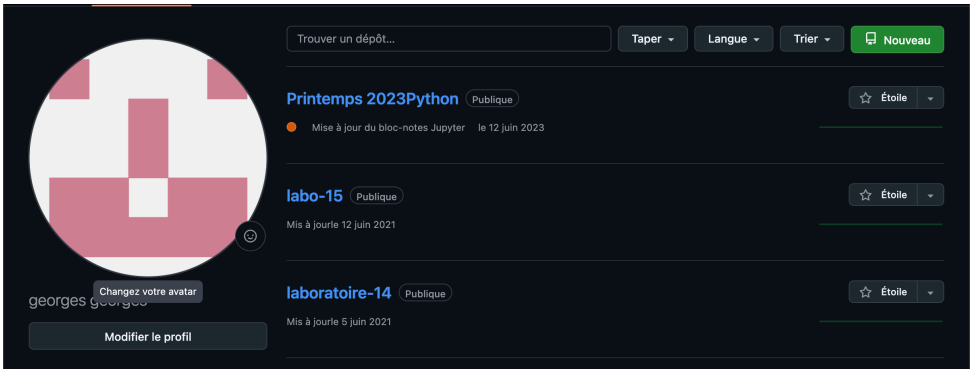
3)Файловая система — часть ОС, которая обеспечивает чтение и запись файлов на дисковых носителях информации.a.Ext2 — расширенная файловая система. Данные сначала кэшируются и только потом записываются на диск.b.Ext3 и Ext4 — журналируемые файловые системы. Осуществляется хранение в виде журнала со списком изменений, что помогает сохранить целостность при сбоях.c.XFS — высокопроизводительная журналируемая файловая система, рассчитанная для работы на дисках большого объёма.

4)Для просмотра подмонтированных в ОС файловых систем необходимо использовать команду findmht.

5)Для удаления зависшего процесса используется команда kill PID или killall название.

Часть 2. Работа с GitHub

1)Первым делом, мы регистрируемся на сайте github.com. Так как мы не в первый раз имеем дело с github, то аккаунт уже готов.



2)Первоначальная настройка для работы с github.

```
[ggeorges@localhost ~]$ git config --global user.name "Gedeon Georges"
[ggeorges@localhost ~]$ git config --global user.email "gedeongeorges293@gmail.com
```

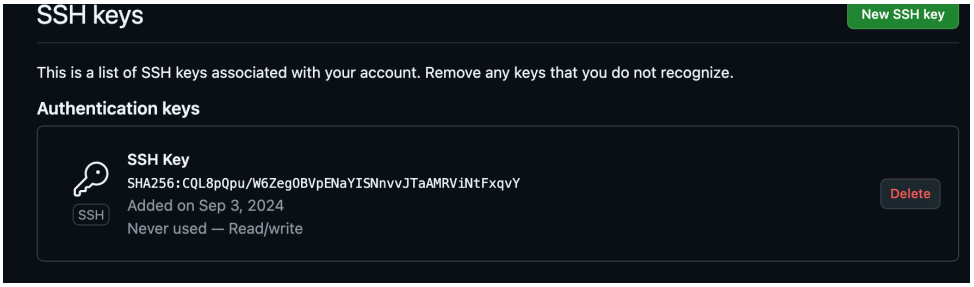
3)Генерация SSH ключа.

```
[ggeorges@localhost ~]$ ssh-keygen -C "Gedeon Georges <gedeongeorges293@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ggeorges/.ssh/id_rsa):
Created directory '/home/ggeorges/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ggeorges/.ssh/id_rsa
Your public key has been saved in /home/ggeorges/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:kFSMikxx58SRiTNR6y53itysWLt5HLOmjtvMzP4Vbvg Gedeon Georges <gedeongeorges293@gmail.com>
The key's randomart image is:
+---[RSA 3072]----+
|  ..o=*B.        |
|  ..+==o.        |
| o . +=          |
|  o .. .         |
|      . S        |
|     .oo .       |
|    o.++=        |
|   @ O=*         |
|  ++^B= E        |
+----[SHA256]-----+
```

4)Переходим на сайте в настройки и в SSH Keys. Копируем наш SSH ключ, предварительно выведя его в консоль с помощью cat ~/.ssh/id_rsa.pub.

```
[ggeorges@localhost ~]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCtNqMVdPOe5gNAIOfc2OrcgyAunFxS8pb+IrNFRMHQ31vm3Uh2lnWe7d6mWgtiTF0O+r6X3ZmXDETvC9jOuJ5jg1LtWQOqM/qd4shXQo0mJXUw5pA/l
3xMTdzh8pTz8/FWo0dluEmo3EN3jYbZqYhDOLxZS7o8kTffIjv3He4+dhhTmmpqPIz/iWEzEO18fi+h3hZlFUvMzG6/2qj6l6B/Su3m7i5067f09jV8fm9vqrzPixveHDAEbWmOchfHdJEe9a4tk5YDN0
nxVwjo2L/A3mMPNVe122Ji8tmC977kmzZ0FyYDiPSwHAHhUXXQWvpModVRbG8kLeJbOtGov/X8VEZQx8KW6hC10kGtTHAVSSpU2IikgDNNCRx6Z8XoENoxRLWBYtkUq5SJixbnD+3lmAnIJwx6pApABKm
tJ5F4vUrSZElI85YnCNV0Qx5is2kw2KtXV4Mr8k5E2VzH8GIAr04ldxDmcRuEg1DKYXRNadf2FZz/7Hquu1thd2zNEU= Gedeon Georges <gedeongeorges293@gmail.com>
```

5)Ключ отобразился на сайте.



КОНТРОЛЬНЫЕ ВОПРОСЫ №2:

1)Система контроля версий Git представляетсобой набор программ командной строки.Доступ к ним можно получить изтерминала посредством ввода командыgitc различ-ными опциями. Системы контроля версий (Version Control System,VCS)применяются при работе нескольких человек над одним 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 определённыхкоманд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляютсяиз центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять неполную версию изменённых файлов, а производить так называемую дельта-компрессию—сохранять только изменения между последовательными версиями,чтопозволяет уменьшить объём хранимых данных.Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например,они могут поддерживать работу с несколь-ми версиями одного файла,сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Крометого, 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 — это системы, которые используют архитектуру клиент / сервер, где один или несколько клиентских узлов напрямую подключены к центральному серверу. Пример - Wikipedia.В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы является

совокупностью решений отдельных узлов. Пример — Bitcoin.В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов.Выполнение большинства функций по управлению версиями осуществляется специальным сервером.

4)Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория:git config --global user.name"Имя Фамилия"git config --global user.email"work@mail"и настроив utf-8 в выводе сообщенийgit:git config --global quotepath falseДля инициализации локального репозитория, расположенного, например, в каталоге ~/tutorial, необходимо ввести в командной строке:cdmkdir tutorialcd tutorialgit init

5)Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый):ssh-keygen -C"Имя Фамилия work@mail"Ключи сохраняться в каталоге~/.ssh/.Скопировав из локальной консоли ключ в буфер обменаcat ~/.ssh/id_rsa.pub | xclip -sel clipвставляем ключ в появившееся на сайте поле.

6)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)Основные команды git:Наиболее часто используемые команды git: – создание основного дерева репозитория:git init–получение обновлений (изменений)текущего дерева из центрального репозитория:git pull–отправка всех произведённых изменений локального дерева в центральный репози-торий:git push–просмотр списка изменённых файлов втекущей директории:git status–просмотртекущих изменения:git diff–сохранениетекущих изменений:–добавить все изменённые и/или созданные файлы и/или каталоги:git add .–добавить конкретные изменённые и/или созданные файлы и/или каталоги:git add имена_файлов – удалить файл и/или каталог из индекса репозитория (приэтомфайл и/илик аталог остаётся в локальной директории): git rm имена_файлов – сохранение добавленных изменений: – сохранить все добавленные изменения и все изменённые файлы: git commit -am 'Описание коммита'–сохранить добавленные изменения с внесением комментария через встроенный редактор:git commit–создание новой ветки, базирующейся натекущей: git checkout -b имя_ветки–переключение

на некоторую ветку: 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: git push origin имя_ветки–слияние ветки стекущим деревом:git merge --no-ff имя_ветки–удаление ветки: – удаление локальной уже слитой с основным деревом ветки:git branch -d имя_ветки–принудительное удаление локальной ветки:git branch -D имя_ветки–удаление ветки с центрального репозитория: git push origin :имя_ветки

8)Использования git при работе с локальными репозиториями (добавления текстового документа в локальный репозиторий):git add hello.txtgit commit -am'Новый файл

9)Проблемы, которые решают ветки git:• 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 проектомтак или иначе могутсоздаваться файлы,которые нетребуется добавлять в последствии в репозиторий. Например, временные файлы, со-здаваемые редакторами,или объектные файлы, создаваемые компиляторами.Можно прописать шаблоны игнорируемых при добавлении в репозиторийтипов файлов в файл.gitignore с помощьюс ервисов. Для этого сначала нужно получить списоки меющихся шаблонов: curl -L -s https://www.gitignore.io/api/listЗатем скачать шаблон,например, для C и C++curl -L -s https://www.gitignore.io/api/c >> .gitignorecurl -L -s https://www.gitignore.io/api/c++ >> .gitignore

Выводы

Я установил VirtualBox, поставил на виртуальную машину Rocky Linux. Скомпилировал файлы из формата md в pdf и docx.