|  |  |
| --- | --- |
| Изображение выглядит как текст, Шрифт, Графика, логотип  Автоматически созданное описание | Министерство образования и науки Российской Федерации  Федеральное государственное автономное образовательное учреждение высшего образования  «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» |

Институт радиоэлектроники и информационных технологий – РтФ

Департамент информационных технологий и автоматики

Школа бакалавриата

Отчёт по лабораторной работе №2 «Основы работы в Linux»

Выполнила:

Гордеева Ольга РИЗ-310916у

г. Екатеринбург   
 2024 г.

# Введение

**Цель работы:**

Целями лабораторной работы являются:

1. Научиться работать с основными командами Linux.
2. Научиться писать простые консольные утилиты на Python.
3. Научиться обрабатывать переданные в URL параметры разных типов.
4. Научиться возвращать пользователю форматированный текст с помощью HTML-тегов.

**Задачи:**

1. Задача 1. Список процессов.
2. Задача 2. Средний размер файла.
3. Задача 3. Дешифратор.
4. Задача 4. Хорошего дня!
5. Задача 5. Максимальное число.
6. Задача 6. Превью файла.
7. Задача 7. Учёт финансов.

# Ход работы

## 1 Установка Ubuntu

В процессе выполнения лабораторной работы мною был скачен Ubuntu dekstop 22.04, а также Oracle VM VirtualBox. После установки Oracle VM VirtualBox я приступила к созданию OS под названием WorkMachine (Рис.1).

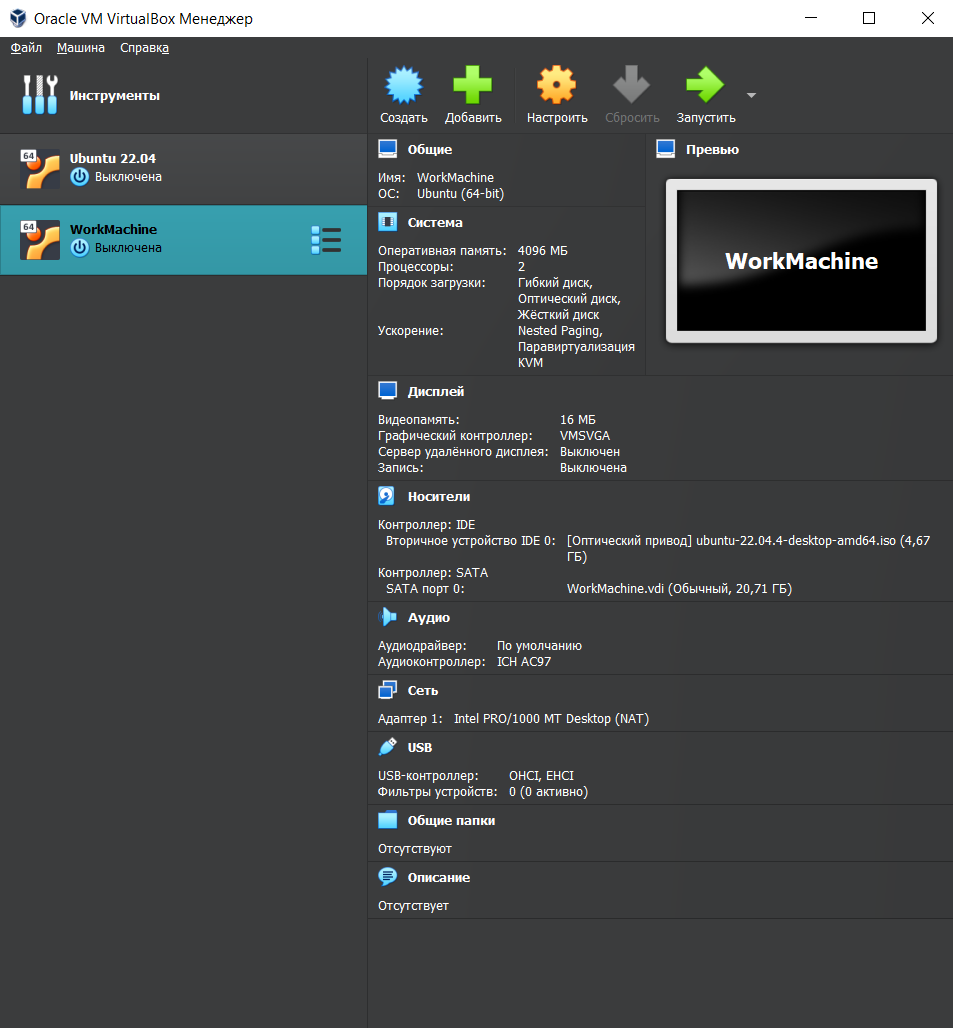


Рисунок 1 – данные о системе

После настройки виртуальной машины была произведена установки ubuntu 22.04. (Рис.2)

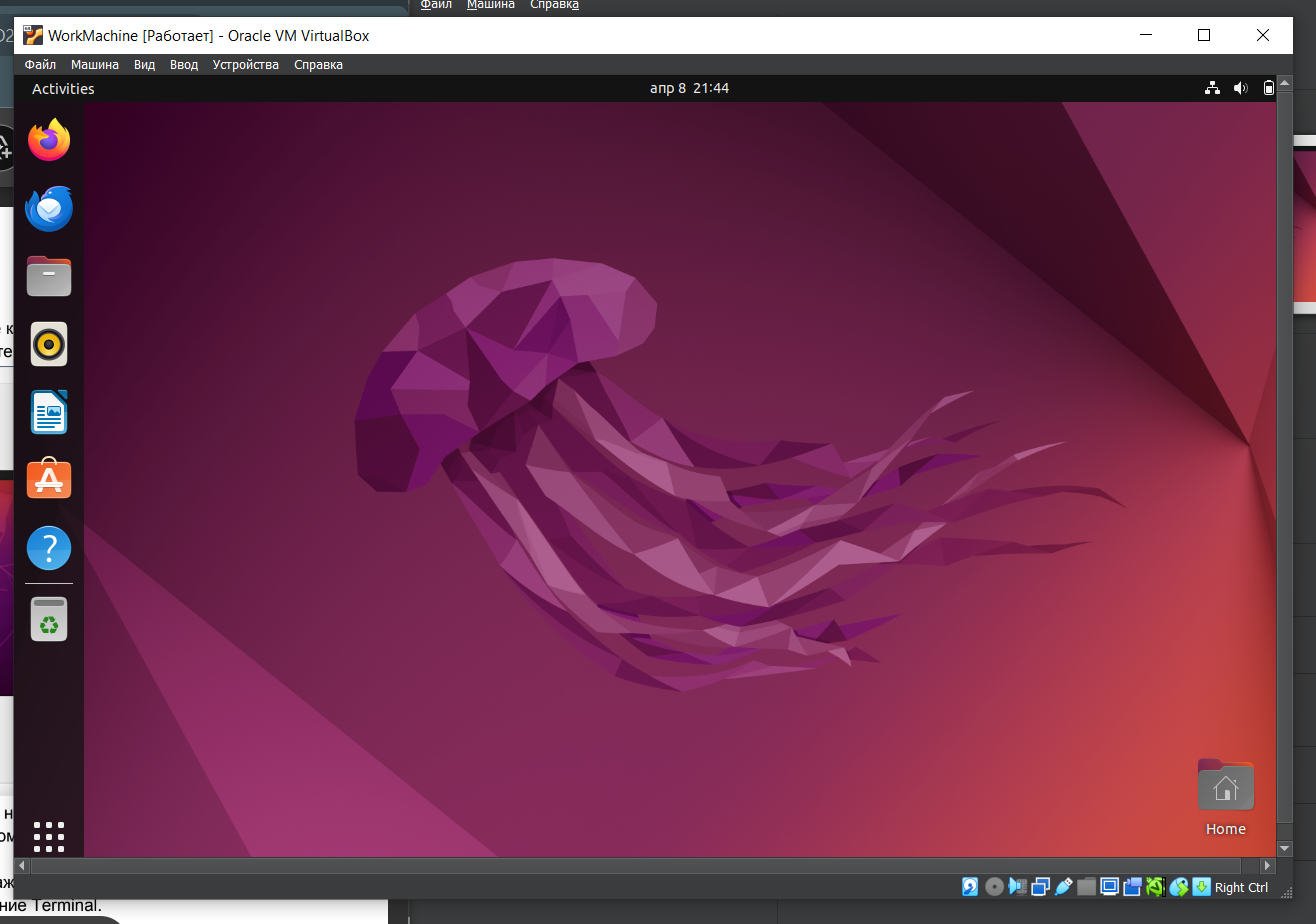


Рисунок 2 – рабочий стол ubuntu после установки

Далее было выполнено обновление и установка apt, что можно увидетьна рисунке 3 и 4.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 3 – обновление пакета apt

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Рисунок 4 – установка пакета apt

После этого была выполнена команда обновления пакета apt (Рис.5) и перезагрузка виртуальной машины с помощью команды:

*sudo reboot*

# 

Рисунок 6 – обновление пакета apt

Для последующего выполнения задач лабораторной работы потребуется установка python3, для этого в терминале была введена команда:

*sudo apt install python3-virtualenv*

А также требуется установить библиотеку Flask с помощью команды pip:

*pip install flask requests*

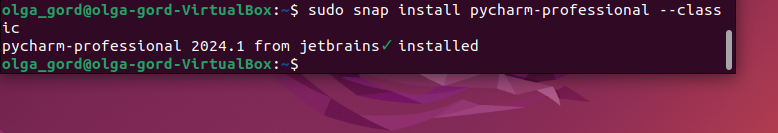
Была выполнена команда sudo snap install pycharm-professional –classic (Рис.7) и запущена IDE для pycharm.

Рисунок 7 – установка IDE для pycharm

## 2 Выполнение практических задач

2.1 Список процессов

Что нужно сделать:

С помощью команды ps можно посмотреть список запущенных процессов. С флагами aux эта команда выведет информацию обо всех процессах, запущенных в системе. Запустите эту команду и сохраните выданный результат в файл:

$ ps aux > output\_file.txt

Столбец RSS показывает информацию о потребляемой памяти в байтах. Напишите функцию get\_summary\_rss, которая на вход принимает путь до файла с результатом выполнения команды ps aux, а возвращает суммарный объём потребляемой памяти в человекочитаемом формате. Это означает, что ответ надо перевести в байты, килобайты, мегабайты и так далее.

Для разбиения строки по столбцам можно воспользоваться функцией split без аргументов:  
columns = line.split()

Так как первой строкой является заголовок, её можно отбросить при расчётах суммарного объёма потребляемой памяти. Это можно сделать, например, с помощью среза:  
lines = output\_file.readlines()[1:]

Что сделано:  
**Код реализован и работает, вывод в консоли – 1718 гигабайт (Рис.8)**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 8 – вывод кода

2.2 Задача 2. Средний размер файла

**Что нужно сделать**

Удобно направлять результат выполнения команды напрямую в программу с помощью конвейера (pipe):

$ ls -l | python3 get\_mean\_size.py

Напишите функцию get\_mean\_size, которая на вход принимает результат выполнения команды ls -l, а возвращает средний размер файла в каталоге.

**Советы и рекомендации**

* **Конвейер (pipe)** — это механизм передачи данных со стандартного потока вывода одной программы на стандартный поток ввода другой программы. Пример запуска конвейера:  
    
  $ ls -R | grep “\.txt” | wc -w   
  1)      2)             3)  
    
  1) Получаем рекурсивно все файлы в текущем каталоге.  
  2) Получаем из них файлы с расширением .txt.  
  3) Получаем общее количество слов в .txt-файлах.
* Получить входные данные можно следующим образом:  
    
  import sys   
     
  data = sys.stdin.read()

Вывод можно делать с помощью того же print.

* Входные данные можно получить сразу в виде списка строк:  
    
  lines = sys.stdin.readlines()
* Первая строка не является информацией о файле, поэтому её можно отбросить при прочтении входных данных:  
    
  lines = sys.stdin.readlines()[1:]
* Кстати, программа также может получать информацию из файла и из результата выполнения другой команды или программы.  
    
  $ ls -l > ls.txt   
  $ python3 get\_mean\_size.py < ls.txt   
  $ cat ls.txt | python3 get\_mean\_size.py   
     
  cat <filename> выводит содержимое файла.

**Что оценивается**

* Программа поддерживает обработку входных данных через конвейер.
* Программа обрабатывает случай, когда файлов нет или не удаётся получить их размер.
* Получение входных данных и вывод результата происходит в блоке \_\_main\_\_.

Изображение выглядит как текст, снимок экрана, Шрифт, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 9 – результат команды ls -l | python3 PycharmProjects/pythonProject/.venv/get\_mean\_size.py

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 10 – код

2.3 Задача 3. Дешифратор

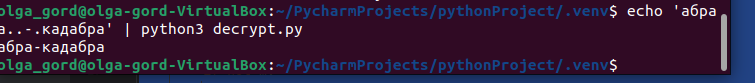


Рисунок 11 – результат работы дешифратора

2.4 Задача 4. Хорошего дня!

На рисунке 12 изображен результат работы кода.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 12 - результат

2.5 Задача 5. Максимальное число

Изображение выглядит как снимок экрана, текст, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 13 – максимальное число

2.6 Задача 6. Превью файла

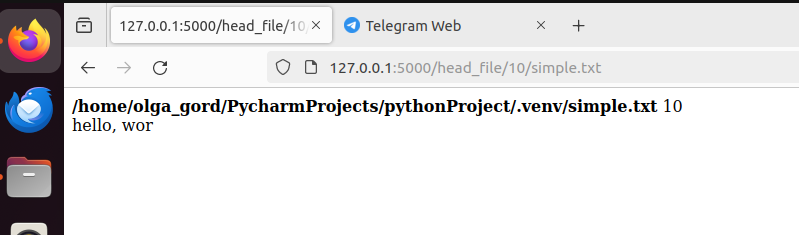


Рисунок 14 – превью файла – первые 8 символов

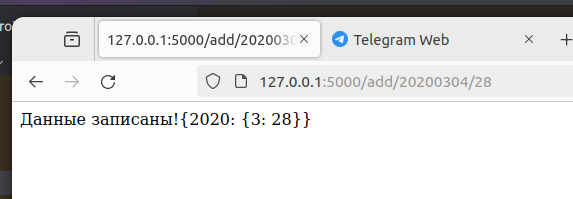
2.7. Задача 7. Учет финансов

Рисунок 14 – добавление записи за март 2020 года

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 15 – итог годовой

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 16 – итог по месяцам