

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

ФАКУЛЬТЕТ ИНФОРМАТИКИ
КАФЕДРА ПРОГРАММНЫХ СИСТЕМ

Отчет по дополнительному заданию

Разработка метода удаления пользователей и их данных в TLJH

Выполнил:
Киселев Е. Д.
Кашапов Д.Р.
гр. 6301-090301D

Проверил:
Востокин С. В.

Самара 2023

ВВЕДЕНИЕ

Для выполнения этой работы:

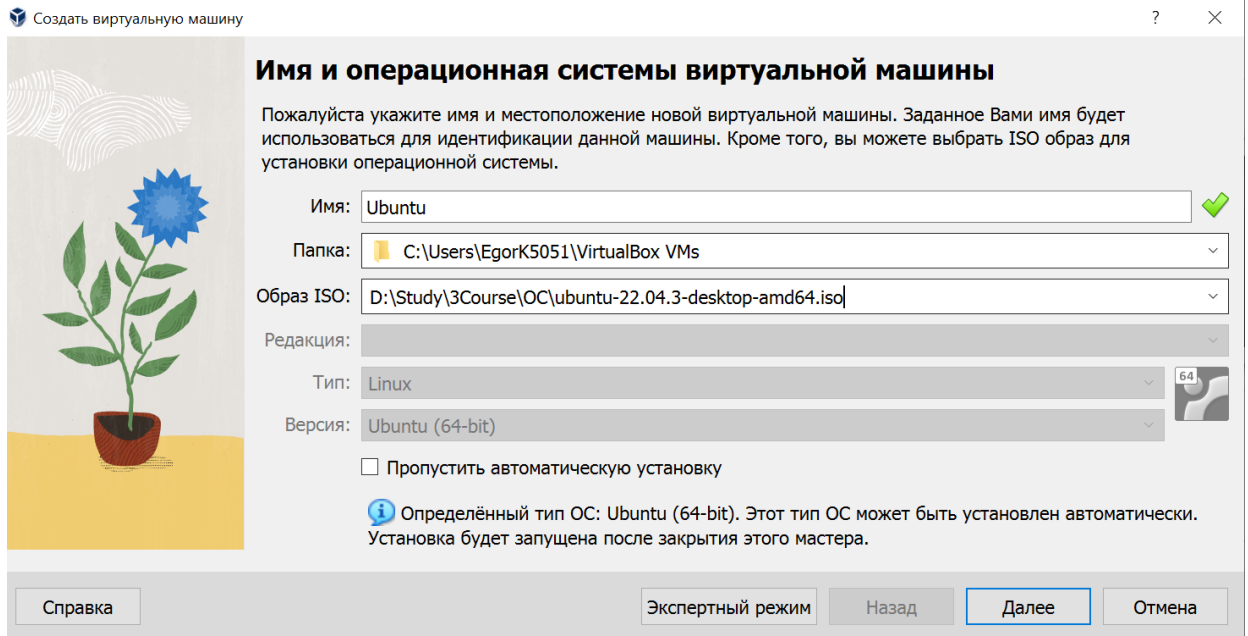
- Был взят образ операционной системы Ubuntu 22.04.3 LTS с официального сайта (<https://ubuntu.com/download/desktop>), который был установлен на виртуальную машину в VirtualBox;
- Клиент TLJH (<https://tljh.jupyter.org/en/latest>).

Подробнее об установке будет написано ниже в отдельном пункте.

Для работы с TLJH имеется свой отдельный REST API, благодаря которому можно реализовать наше задание (<https://jupyterhub.readthedocs.io/en/stable/reference/rest-api.html#/>). Подробный функционал написан на сайте.

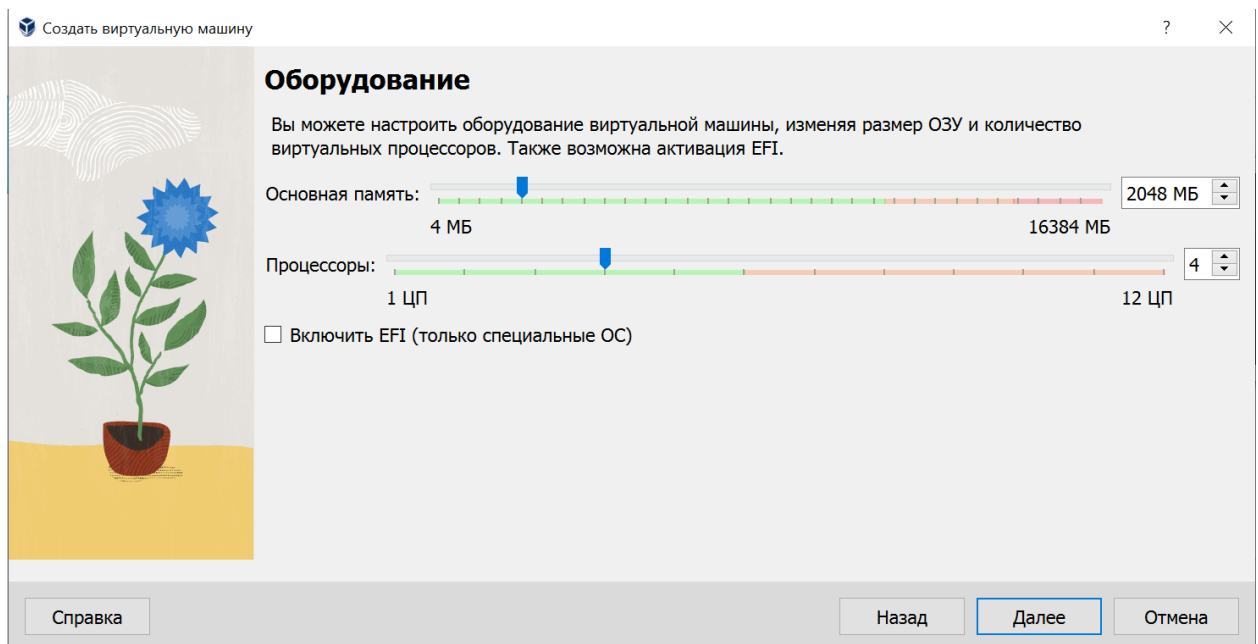
УСТАНОВКА UBUNTU И TLJH

1. Создать виртуальную машину с указанием необходимых параметров:
 - **Имя: Ubuntu** (требуется для автоматического определения VirtualBox-ом типа и версии операционной системы)
 - **Образ ISO: Указать свой путь к файлу ISO**

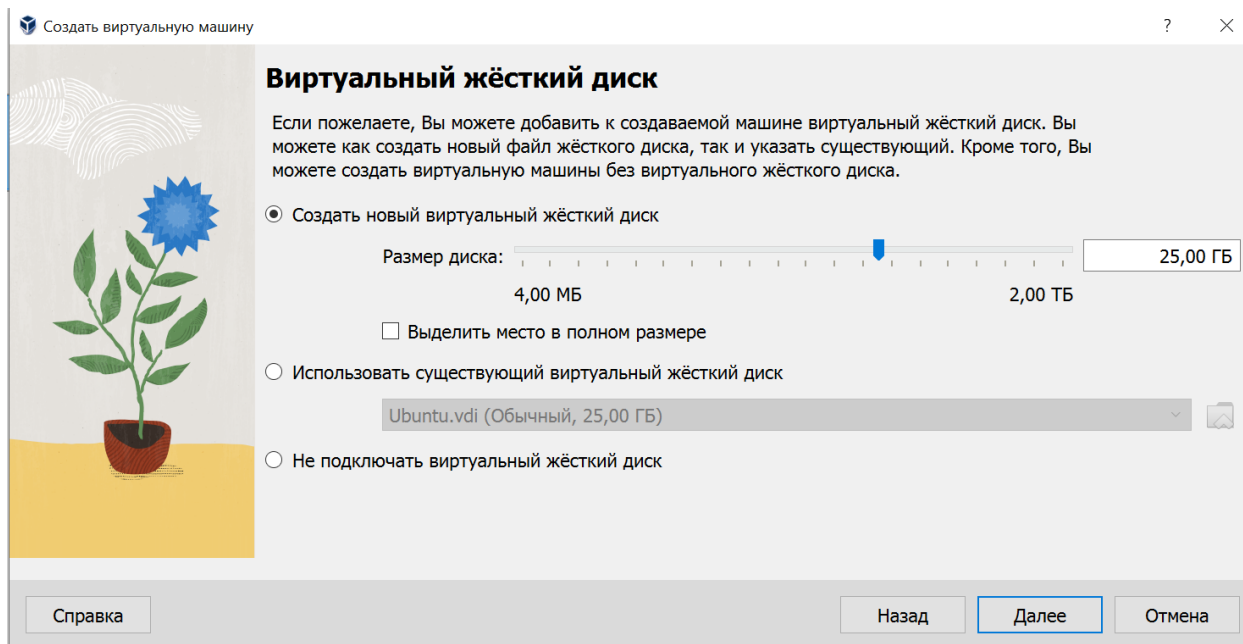


В следующем окне указать имя пользователя и пароль на свое усмотрение.

В окне «Оборудование» указываем **2048 мб** и **4 ЦП**, для комфортной работы с Ubuntu



На виртуальный жесткий диск рекомендую выделить **не менее 25 Гб**, так как Ubuntu в установленном виде занимает 12 Гб.



После окна «Итог» нажать кнопку «Далее». У нас создастся виртуальная машина с именем Ubuntu, которая автоматически запустится VirtualBox-ом. Установка ОС произойдёт автоматически.

2. Для установки python3, python3-dev, curl, git, введите строку ниже в терминал и нажмите enter.

```
sudo apt install python3 python3-dev git curl
```

3. Скопируйте

```
curl -L https://tljh.jupyter.org/bootstrap.py | sudo -E python3 - --admin <NAME>
```

и вставьте его в терминал, предварительно заменив <NAME> на имя администратора, и нажмите enter. Имя администратора выберите любое, которое вам нравится. Администратор может войти в систему после установки tljh. Процесс установки займет 5-10 минут.

4. Скопируйте общедоступный IP-адрес вашего сервера (у VirtualBox он всегда **10.0.2.15**), вставьте его в адресную строку браузера и нажмите enter.

Для того чтобы узнать ваш ip-адрес введите в терминал `ifconfig`

5. Войдите в систему, используя имя администратора, которое вы использовали на шаге 2. Пароль можете выбрать любой, который вы хотите. Главное не забудьте его.

6. Поздравляю! TLJH установлен на linux Ubuntu.

НАПИСАНИЕ ПРОГРАММЫ

Напишем программу для удаления пользователей с сервера и их данных, которые:

- Не имеют прав администратора;
- Заходили через свой логин хотя бы один раз;
- Не проявляли никакой активности на сервере в течение 10 минут.

Важно! Перед запуском программы удостоверьтесь, что в поле `api_url` указана ваша ссылка на сервер вида `http://<ВАШ-IP>/hub/api` и ваш собственный токен, который можно получить пользователю с правами администратора во вкладке **Token**

Листинг программы:

```
import requests
import shutil
import os
from datetime import datetime
from time import sleep

#insert your own admin-permission token and url in format
"http://<your_own_url>/hub/api"
token = "3223bbec00f640fa916213854290375d"
api_url = 'http://10.0.2.15/hub/api'

#in this object we keep name, permissions and latest time online of every user we
found
class user:
    def __init__(self, user_name, user_user_is_admin, user_latest_time_online):
        self.user_name = user_name
        self.user_is_admin = user_user_is_admin
        self.user_latest_time_online = user_latest_time_online

    def getName(self):
        return self.user_name

    def getUserIsAdmin(self):
        return self.user_is_admin

    def getUserLatestTimeOnline(self):
        return self.user_latest_time_online

#returns full list of info about users
def getInfoList():
    users_list_req = requests.get(api_url + '/users', headers={'Authorization':
f'token {token}'})
    users_list_req.raise_for_status()
```

```

usersInfoList = users_list_req.json()
return usersInfoList

#returns list of all users
def getUsersList(usersInfoList):
    users_list = []
    for i in range (len(usersInfoList)):
        user_name = usersInfoList[i]['name']
        user_latest_time_online = usersInfoList[i]['last_activity']
        if ('admin' in usersInfoList[i]['roles']):
            user_is_admin = 1
        else:
            user_is_admin = 0

        users_list.append(user(user_name, user_is_admin,
user_latest_time_online))
    return users_list

#deletes unactive users
def deleteUnactiveUsers(usersList):
    curr_datetime = datetime.utcnow()
    for i in range (len(usersList)):
        if (usersList[i].getUserIsAdmin() != 1) and
(str(usersList[i].getUserLatestTimeOnline()) != "None"):
            user_time =
datetime.strptime(str(usersList[i].getUserLatestTimeOnline()), '%Y-%m-
%dT%H:%M:%S.%fz' )
            if (abs(user_time.timestamp() - curr_datetime.timestamp()) > 600):
                name = str(usersList[i].getName())
                delete_user_req = requests.delete(api_url + '/users/' + name,
headers= {'Authorization': f'token {token}',})
                delete_user_req.raise_for_status()

                if os.path.isdir('home/jupiter-' + name):
                    shutil.rmtree('home/jupiter-' + name)
                    print("deleted ", name, "!")
    print("-----")

#prints all active users
def printUsers(usersList):
    for i in range (len(usersList)):
        print(f"Name: {usersList[i].getName()}, Has Admin:
{usersList[i].getUserIsAdmin()}, Latest Activity:
{usersList[i].getUserLatestTimeOnline()}")

#main code
users_list = getUsersList(getInfoList())

printUsers(users_list)

```

```
while True:  
    users_list = getUsersList(getInfoList())  
    deleteInactiveUsers(users_list)  
    sleep(5)
```

Результат работы программы представлен на изображениях 1-5:

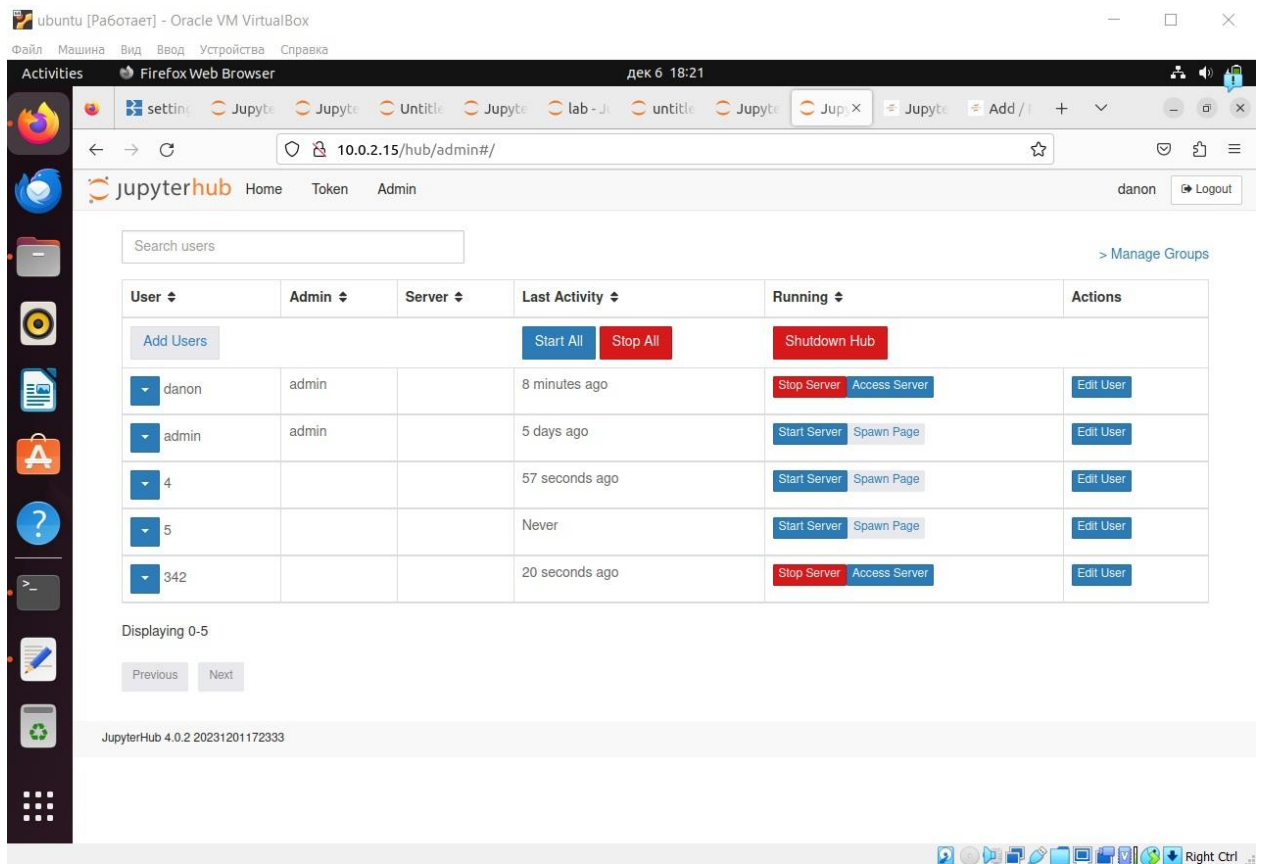


Рис. 1 – Список пользователей до начала выполнения программы

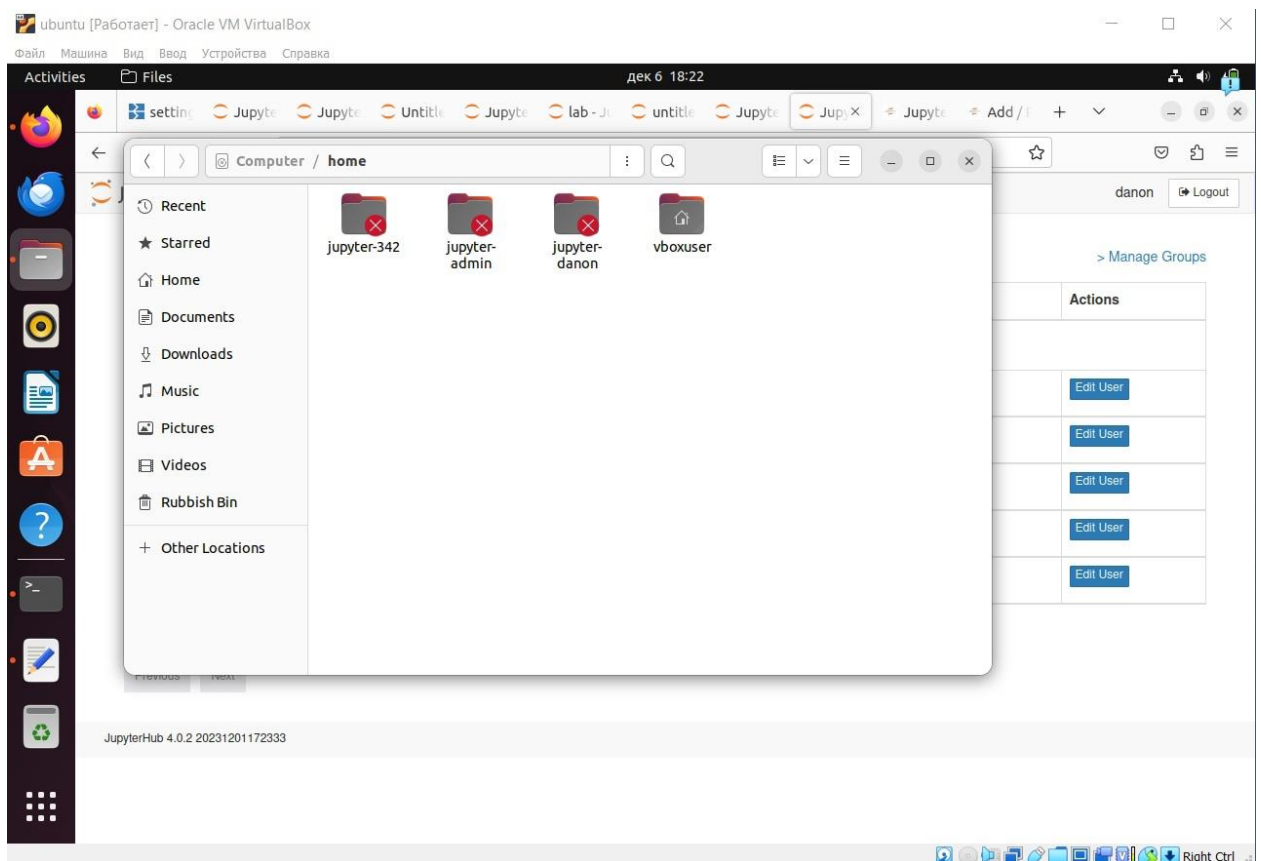


Рис. 2 – Список папок пользователей до начала выполнения программы

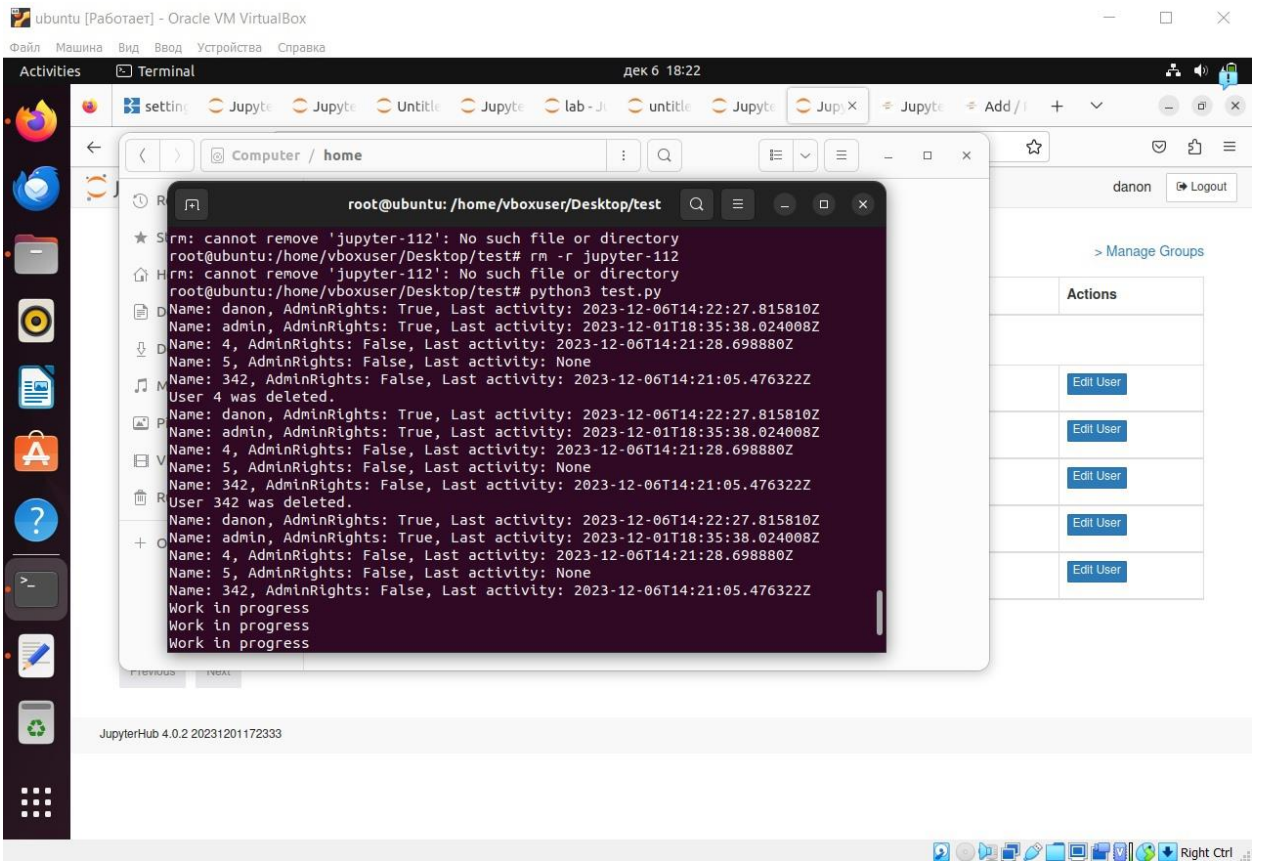


Рис. 3 – Запуск и результат работы программы

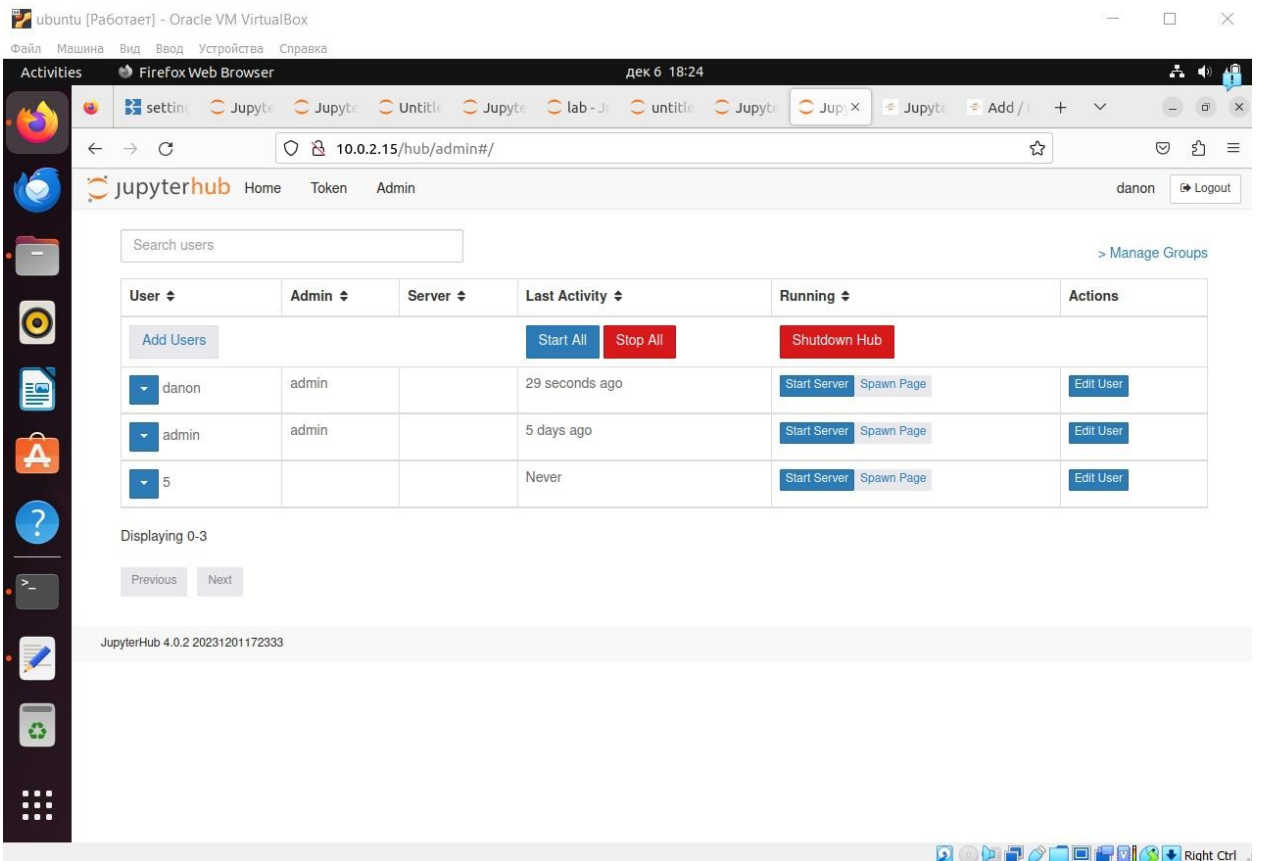


Рис. 4 – Список пользователей после выполнения программы

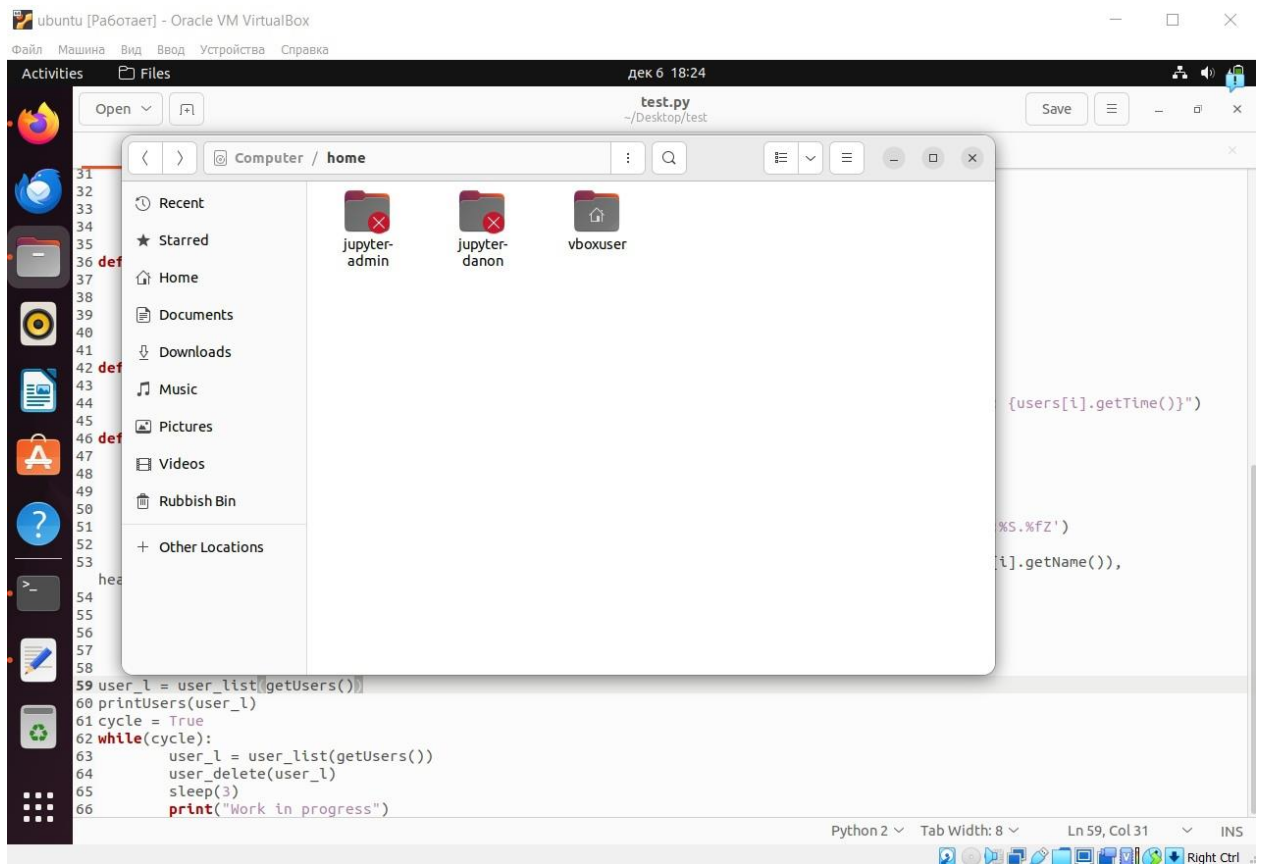


Рис. 5 – Список папок пользователей после выполнения программы