|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по блоку практических работ №1**

по дисциплине «Системное программное обеспечение»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИВБО-08-22 | Стецюк Вячеслав Викторович |
| **Проверил:** | Ст. преп. Воронцов Ю.А. |

МОСКВА 2024 г.

**Содержание**

[Практическая работа №1 2](#_Toc161345384)

[Практическая работа №2 4](#_Toc161345385)

[Практическая работа №3 8](#_Toc161345386)

[Практическая работа №4 13](#_Toc161345387)

[ЗАКЛЮЧЕНИЕ 18](#_Toc161345388)

# **Практическая работа №1**

**Цель работы:**

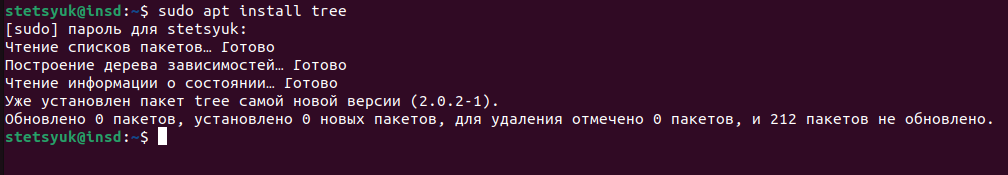
Получить навыки по работе с командной строкой операционной системы Linux при помощи дистрибутива Ubuntu Desktop версии 22.04.4 LTS (Jammy Jellyfish).

**Ход работы:**

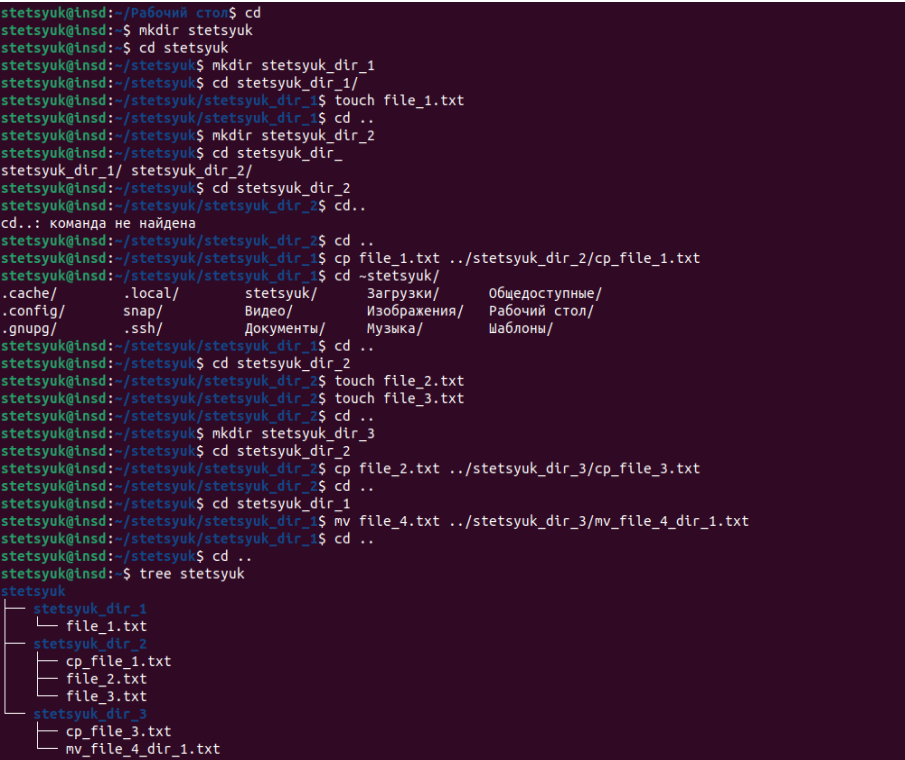
Знакомство с дистрибутивов мы начнем с создания дерева директорий в Linux. Для этого необходимо скачать пакет tree для работы с деревьями с помощью команды *sudo apt install tree*.

После успешной установки приступим к созданию дерева, описанного в задании:

В домашней директории создадим директорию stetsyuk, в ней – еще три директории: stetsyuk\_dir\_1, stetsyuk\_dir\_2, stetsyuk\_dir\_3. В первой директории создадим файл file1.txt, во второй – file2.txt, file3.txt, и скопируем file1.txt из первой директории, переименовав его в cp\_file1.txt. В третью директорию скопируем file3.txt, так же переименовав его, и, предварительно создав в первой директории файл file4.txt, перенесем его с помощью команды mv в третью директорию, переименовав его в mv\_file4\_dir\_1.txt, и, наконец, выведем дерево файлов на экран. Весь процесс описан на рисунках 1.1, 1.2:



**Рисунок 1.1 – Установка пакета tree**



**Рисунок 1.2 – Создание дерева файлов**

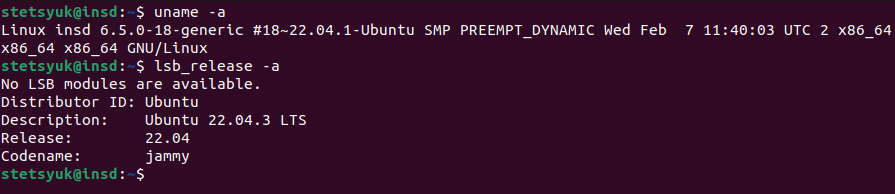
# **Практическая работа №2**

**Цель работы:**

Изучить структуру операционной системы при помощи специальных команд поиска.

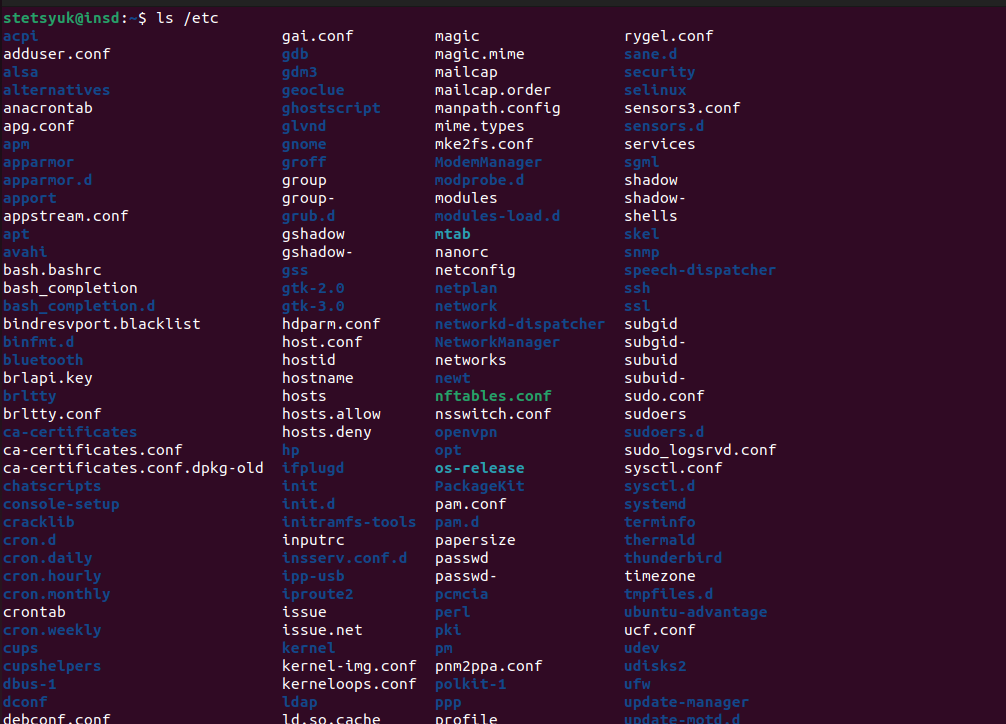
**Ход работы:**

Для начала выведем полную информацию о ядре Linux и дистрибутиве с помощью команды uname -a и lsb\_release -a (рисунок 2.1)

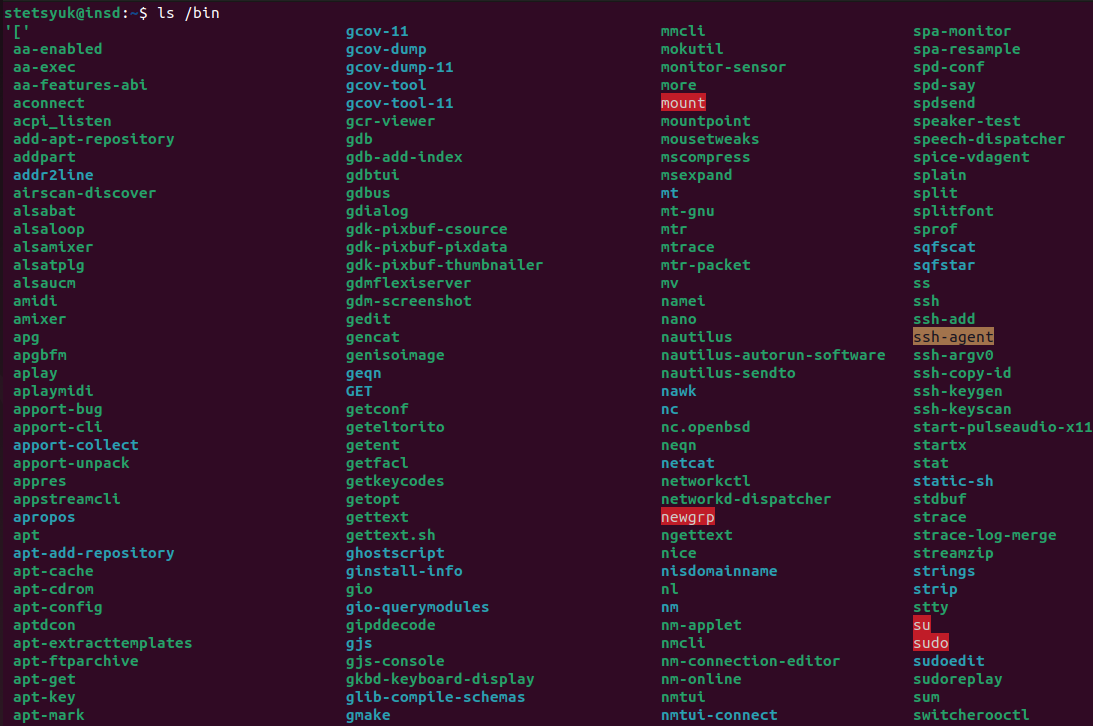
**Рисунок 2.1 – Информация о ядре**

Далее исследуем базовые команды терминала: ls, less и file.

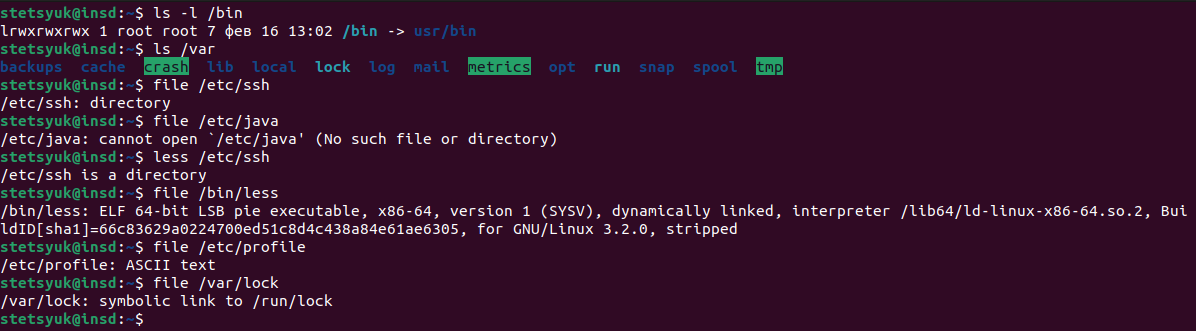
Исследуем директории bin, etc и var, в каждой из них узнаем информацию о нескольких файлах, выбранных случайно и прочтем содержимое файлов с помощью команды less (Рисунки 2.2-2.6):



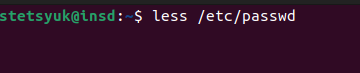
**Рисунок 2.2 – Исследование команды ls**



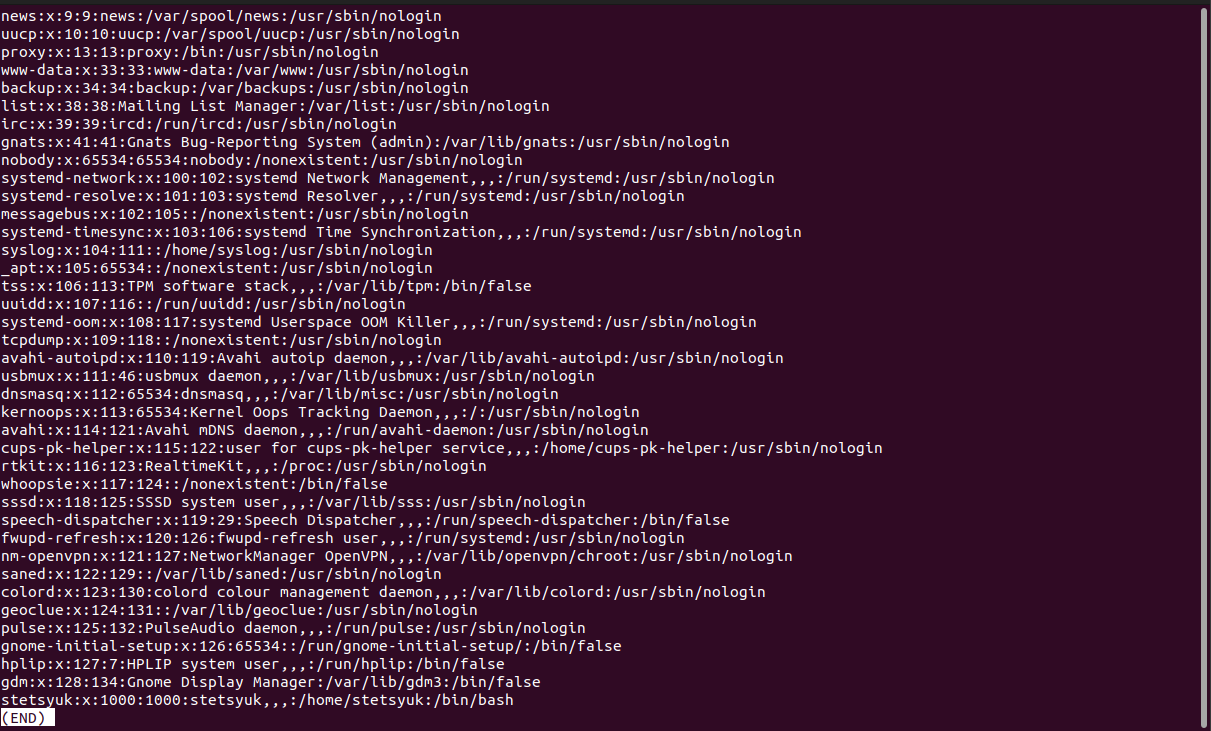
**Рисунок 2.3 – Исследование команды ls**



**Рисунок 2.4 – Исследование команды ls и file**



**Рисунок 2.5 – Исследование команды less**



**Рисунок 2.6 – Исследование команды less**

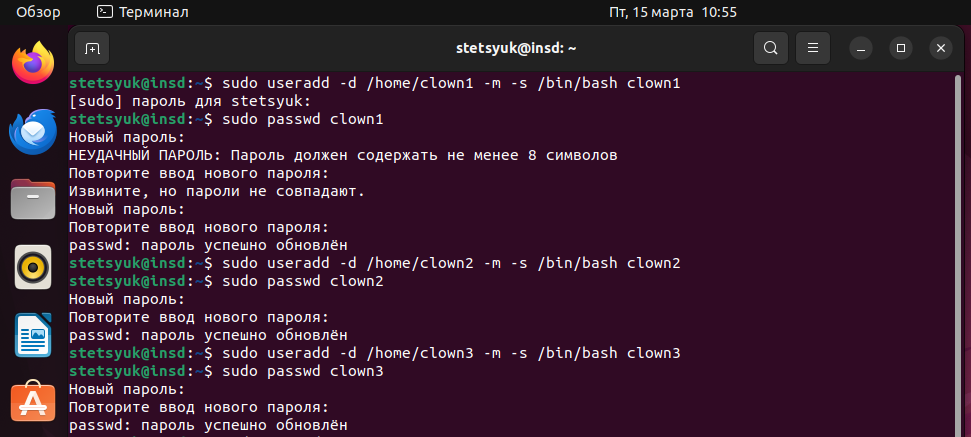
# **Практическая работа №3**

**Цель работы:**

Изучить возможности работы с разграничениями прав пользователей, возможностями суперпользователя в операционных системах семейства Linux.

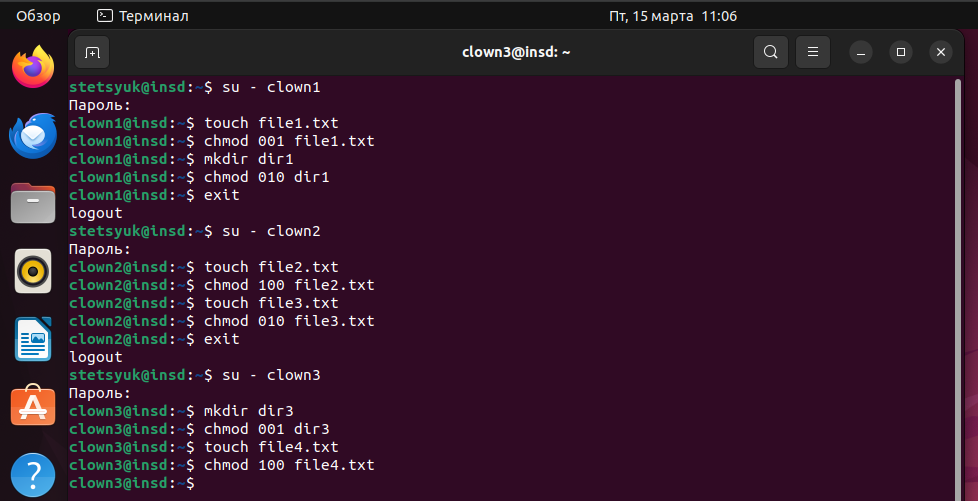
**Ход работы:**

Создадим трёх пользователей с обычными правами: user1, user2, user3 (Рисунок 3.1).



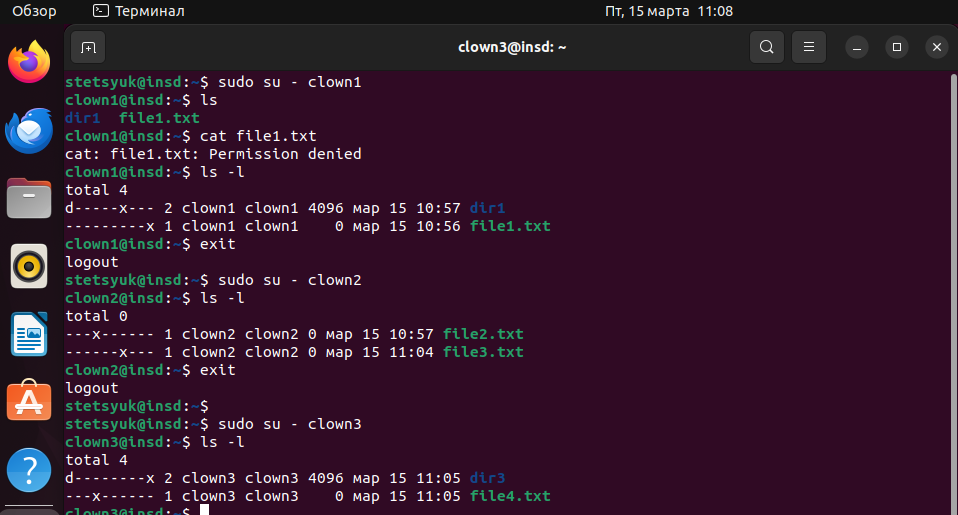
**Рисунок 3.1 – Создание пользователей**

Далее создадим набор различных файлов и директорий в домашних директориях всех пользователей и изменим случайным образом права у этих файлов (Рисунок 3.2).

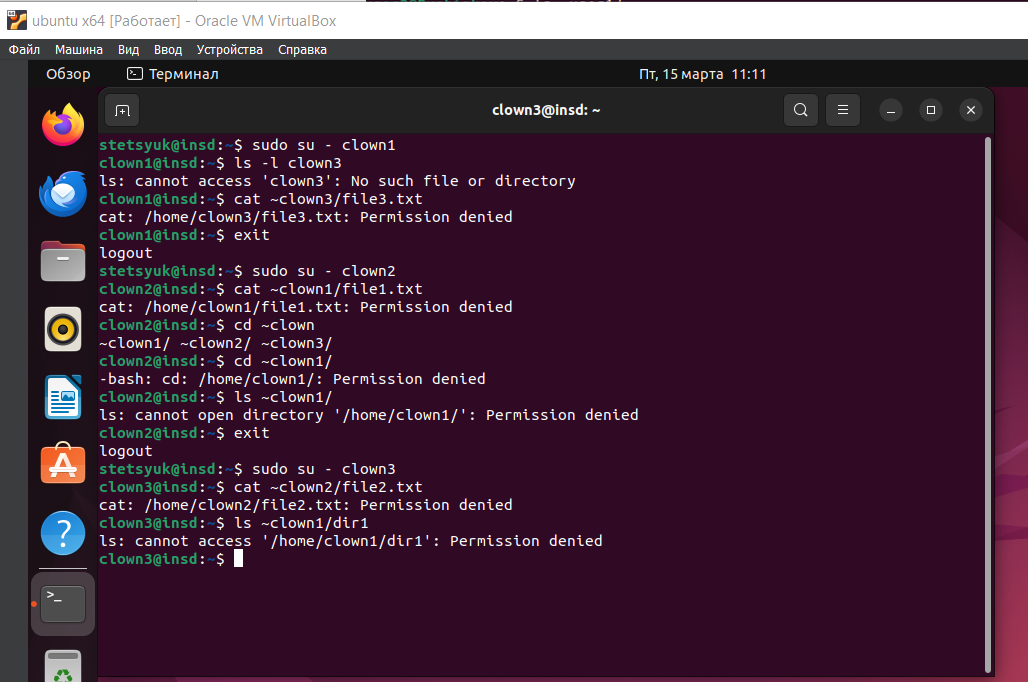


**Рисунок 3.2 – Создание файлов и модификация доступа**

Далее попытаемся повзаимодействовать с файлами и дерикториями, имеющие различные модификации доступа (Рисунок 3.3).

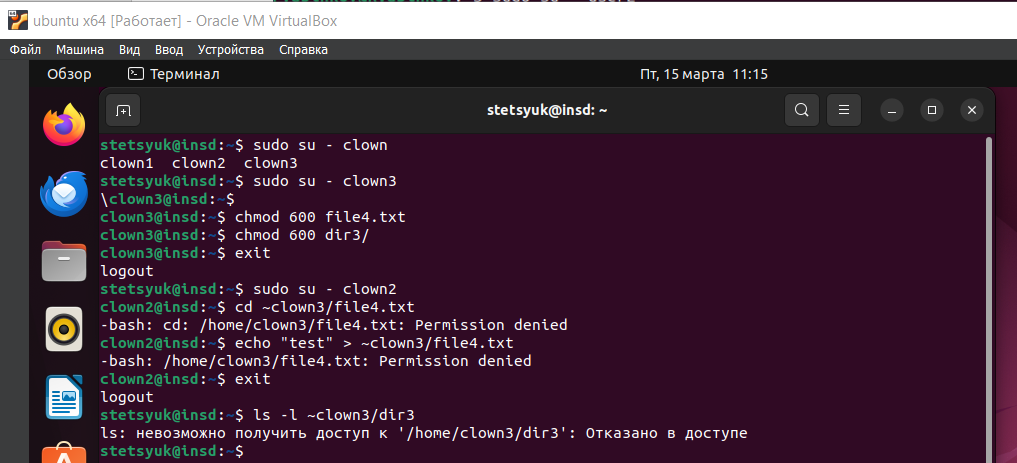


**Рисунок 3.3 – Проверка доступа к файлам других пользователей**



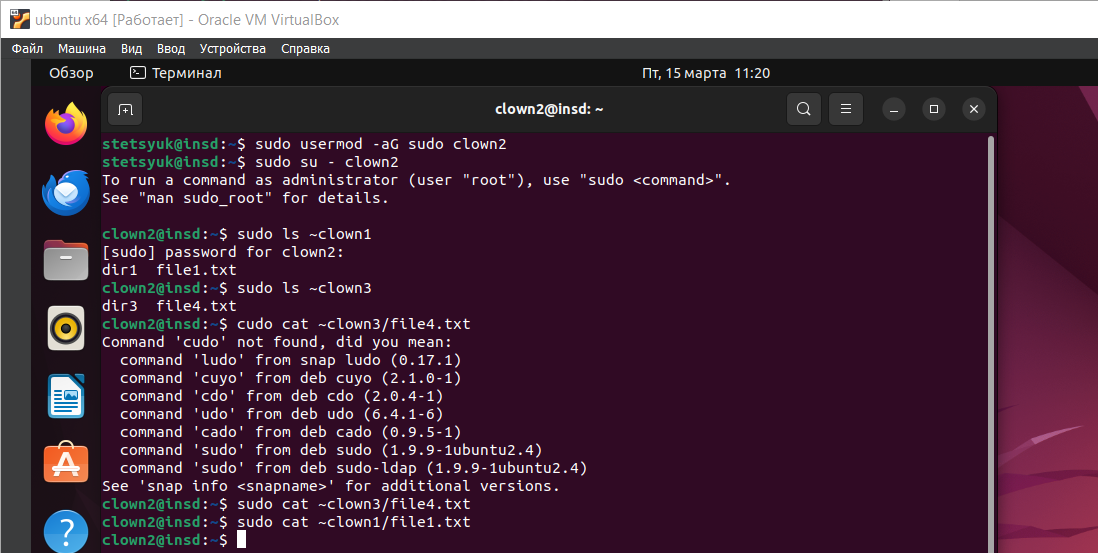
**Рисунок 3.4 – Проверка доступа к файлам других пользователей**

Используя команды управления доступом запретить возможность любым пользователям, кроме владельца, доступ к каталогам и файлам одного из пользователей. Проверить невозможность доступа (Рисунок 3.5).



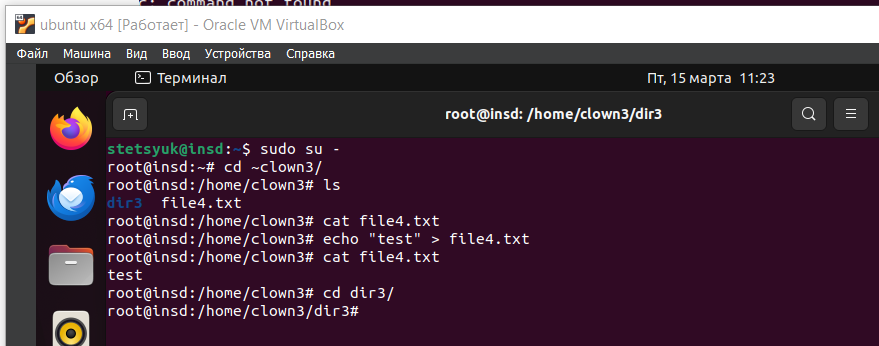
**Рисунок 3.5 – Изменение модификаторов доступа и проверка доступа к файлам**

Затем проведем попытку получения доступа к ранее запрещенным файлам с помощью sudo (Рисунок 3.6).



**Рисунок 3.6 – Попытка получения доступа к запрещенным файлам**

Затем перейдем в суперпользователя root и попробуем использовать любую команду (Рисунок 3.7).



**Рисунок 3.7 – Выполнение команд с помощью root**

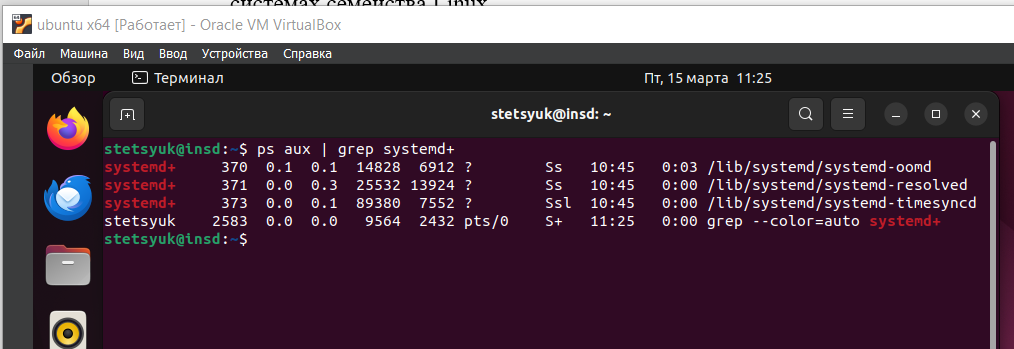
**Практическая работа №4**

**Цель работы:**

Изучить работу с процессами при помощи набора команд в операционных системах семейства Linux.

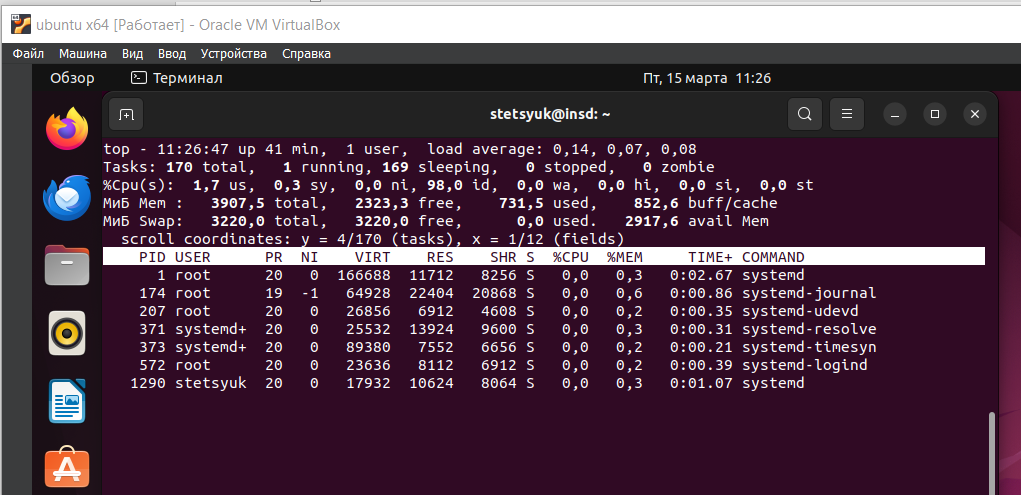
**Ход работы:**

C помощью команды ps и grep выведем все процессы принадлежащие пользователю systemd (Рисунок 4.1).



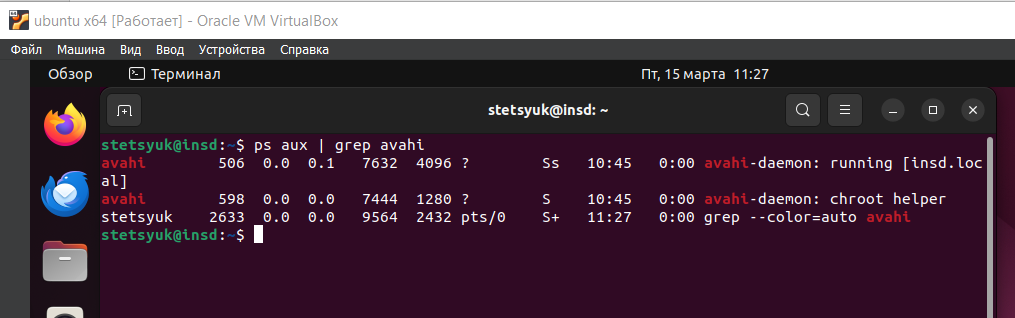
**Рисунок 4.1 – Сортировка процессов, принадлежащих пользователю systemd через ps**

В программе top введем команду COMMAND=systemd (Рисунок 4.2).



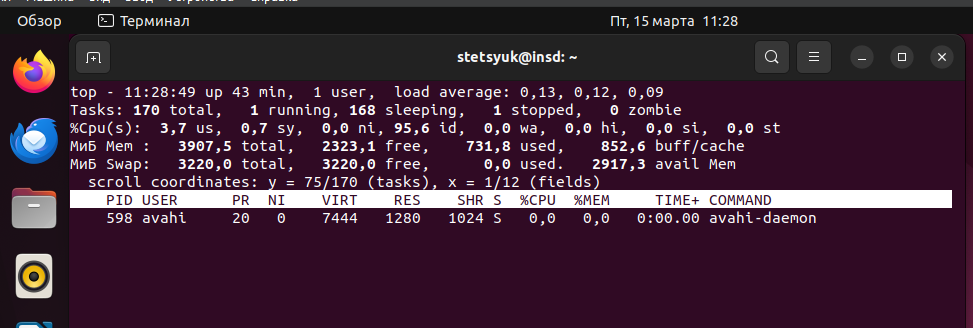
**Рисунок 4.2 – Сортировка процессов, принадлежащих пользователю systemd через top**

C помощью команды ps и grep выведем все процессы принадлежащие пользователю avahi (Рисунок 4.3).



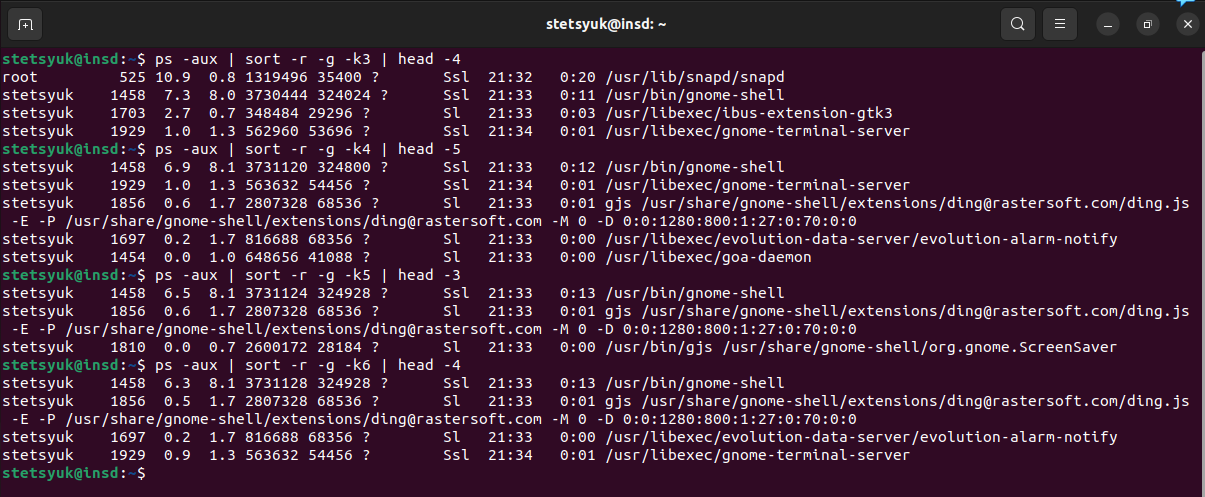
**Рисунок 4.3 – Сортировка процессов, принадлежащих пользователю avahi через ps**

В программе top введем команду COMMAND=avahi (Рисунок 4.4).



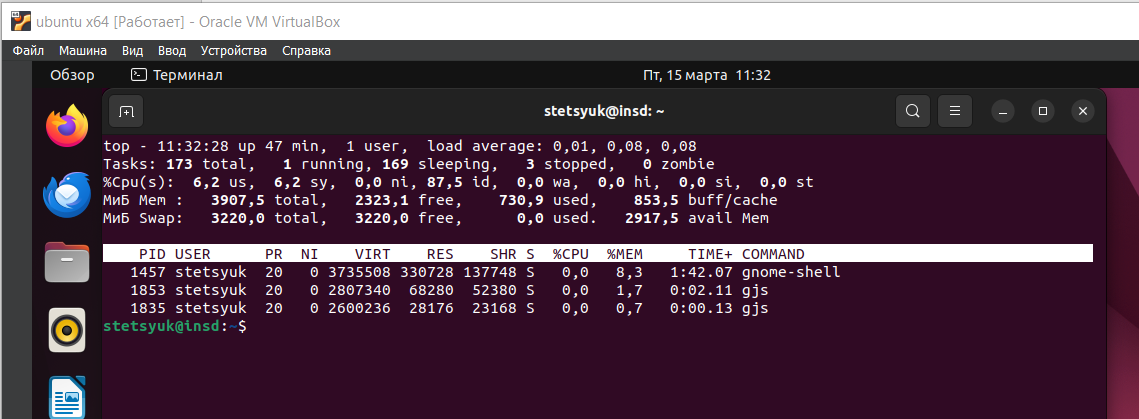
**Рисунок 4.4 – Сортировка процессов, принадлежащих пользователю avahi через top**

Выведем с помощью команд ps, sort и head топ 4 процессов, занимающих наибольшее количество ресурсов процессора в процентах; Топ 5 процессов, занимающих наибольшее количество ресурсов памяти в процентах; Топ 3 процессов, занимающих наибольший объем виртуальной памяти; Топ 4 процессов, занимающих наибольший объем физической памяти (ОЗУ) (Рисунок 4.5).



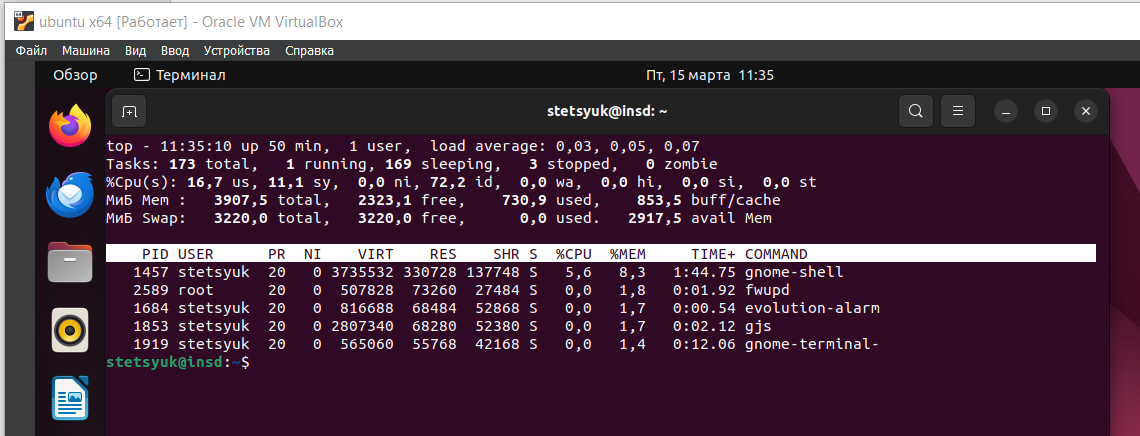
**Рисунок 4.5 – Сортировка процессов командами ps, sort и head**

Выведем в программе top 4 процесса, занимающих наибольшее количество ресурсов процессора в процентах с помощью команды top -o VIRT | head -n10 (рисунок 4.6).



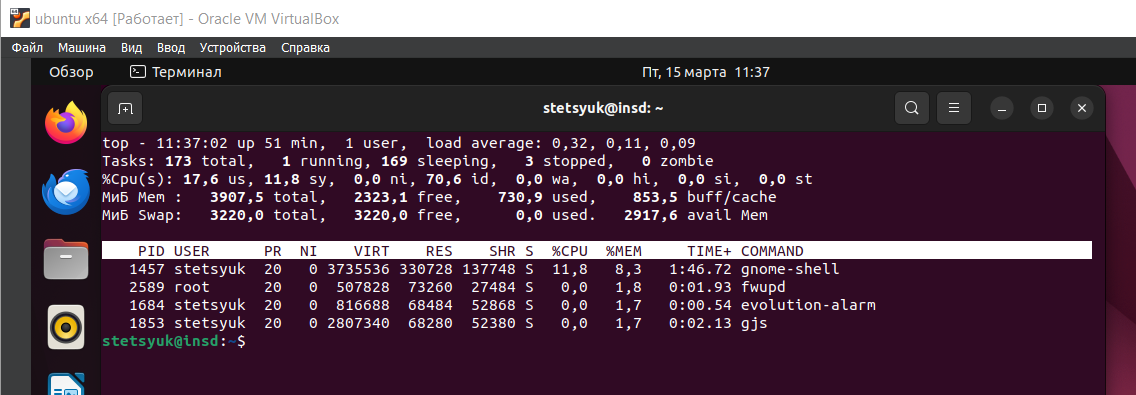
**Рисунок 4.6 – Сортировка процессов командами top и head**

Выведем в программе top 5 процессов, занимающих наибольшее количество ресурсов памяти, top -o %MEM | head -n12 (рисунок 4.7).



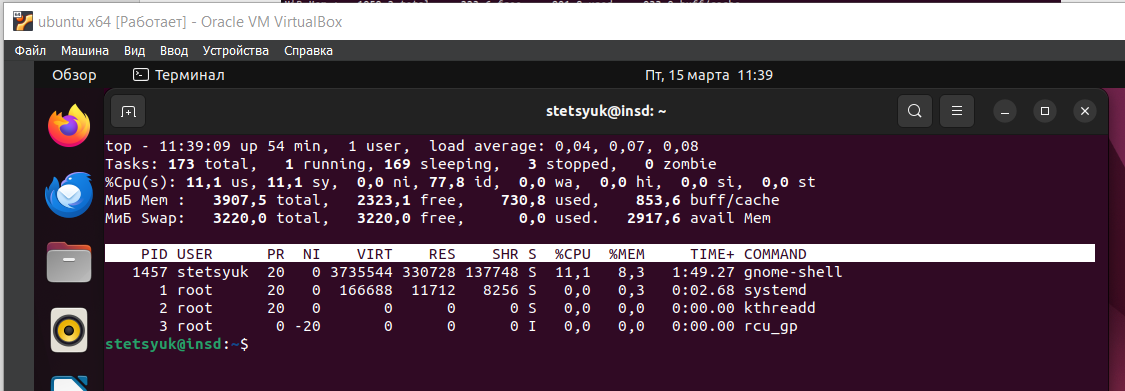
**Рисунок 4.7 – Сортировка процессов командами top и head**

Выведем в программе top 3 процесса, занимающих наибольший объем виртуальной памяти, top -o RES | head -n11 (рисунок 4.8).



**Рисунок 4.8 – Сортировка процессов командами top и head**

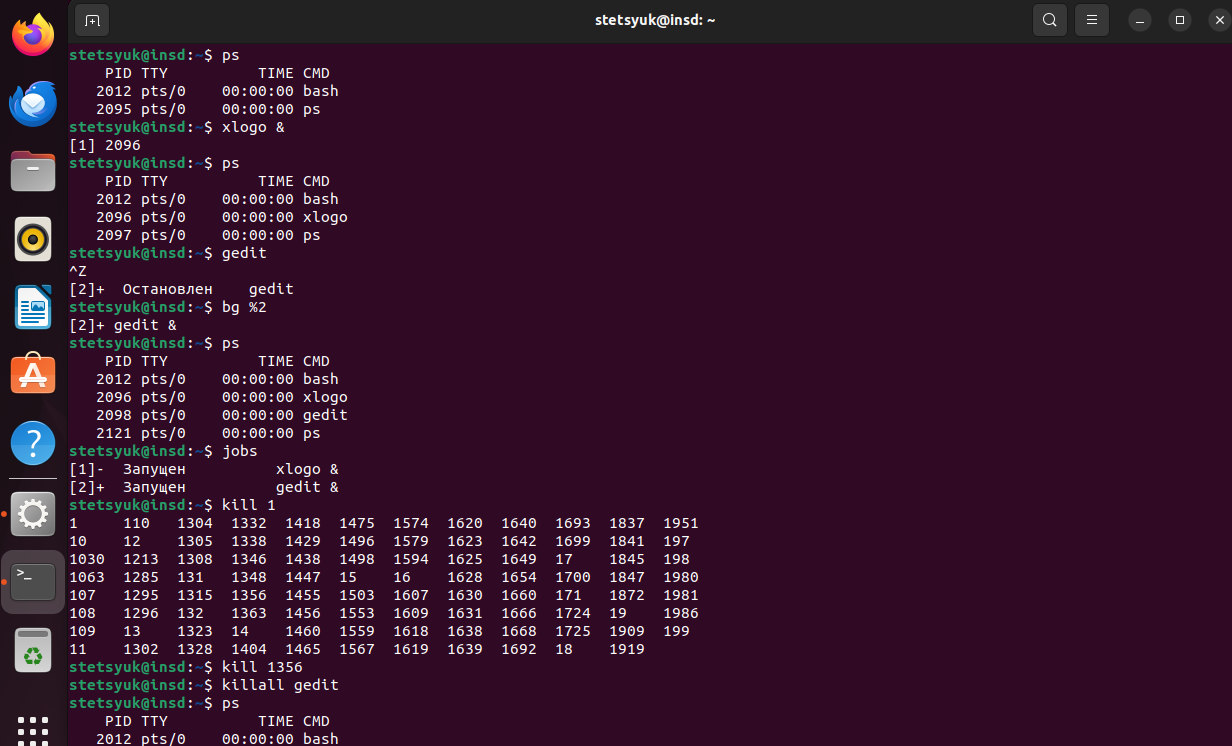
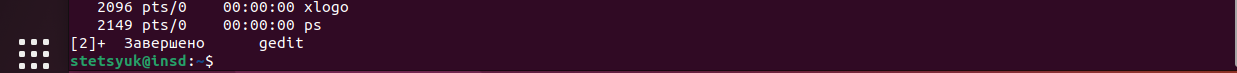
Выведем в программе top 4 процесса, занимающих наибольший объем физической памяти (ОЗУ), top -o %CPU | head -n11 (рисунок 4.9).



**Рисунок 4.9 – Сортировка процессов командами top и head**

Запустим xlogo в фоновом режиме командой xlogo &, а также запустим gedit и переместим его в фоновый режим, выведем список находящихся в фоне процессов командой jobs, завершим процесс xlogo командой kill и

процесс gedit командой killall (рисунок 4.10).



**Рисунок 4.10 – Работа с процессами xlogo и gedit.**

# **ЗАКЛЮЧЕНИЕ**

В данном блоке практических работ были получены навыки взаимодействия с командной строкой, работы с файловым деревом Linux, взаимодействия с пользователями, их создании, изменении модификаторов доступа файлов/директорий, наделении пользователя правами sudo и работы под root пользователем. Также получены знания работы с процессами Linux, их завершение, сортировка, мониторинг и работа в фоновом режиме.