|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_

Отчет по лабораторной работе № 1(Ubuntu)

**«ОС Ubuntu. Установка операционной системы. Интерфейс пользователя»**

по дисциплине «Операционные системы»

Студент ИУ5-51Б Т.А. Цыпышев

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель П.С. Семкин

(Подпись, дата) (И.О.Фамилия)

**Москва**

**2024**

**Цель работы**

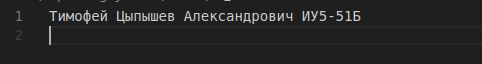
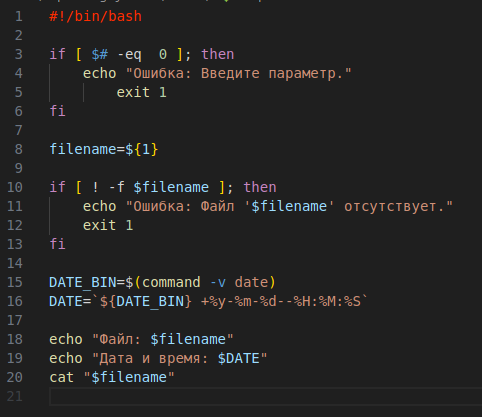
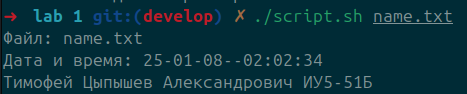
Целью работы является:

1. знакомство с концепцией виртуализации;
2. создание виртуальной машины и установка гостевой операционной системы Ubuntu;
3. знакомство и работа в графической оболочке Unity ОС Ubuntu;
4. знакомство и работа c командным интерпретатором bash ОС Ubuntu.

**Задание**

1. Создать виртуальную машину в среде менеджера виртуальных машин Oracle VM VirtualBox.
2. Установить на виртуальную машину гостевую операционную систему Ubuntu
3. Познакомиться с графической оболочкой и интерпретатором команд операционной системы
4. Создать текстовый файл и файл сценария вывода содержимого файла
5. Проверить выполнение сценария

**Порядок выполнения**

1. Создадим файл name.txt  
   Создадим в домашнем каталоге текстовый файл содержащий ФИО студента и учебную группу  
   
2. Создадим файл script.sh  
   Создать файл сценария с помощью команды touch и текстового редактора vi  
   После создания файла сделаем его исполняемым и запустим:  
   

**Контрольные вопросы**

1. **В чём различие между физическими и виртуальными ресурсами?**

*Физические ресурсы — это реальные компоненты оборудования, такие как процессор, оперативная память, жёсткий диск, видеокарта и периферийные устройства. Они существуют физически и ограничены в количестве.*

*Виртуальные ресурсы — это ресурсы, создаваемые программным обеспечением (например, гипервизором или операционной системой), которые имитируют физические устройства. Виртуальные ресурсы могут быть масштабируемыми и гибкими, позволяя эффективнее использовать физические ресурсы.*

1. **Что такое виртуальная машина?**

Виртуальная машина (ВМ) — это программная эмуляция компьютера, которая работает на физическом компьютере. Она предоставляет изолированную среду, в которой можно установить и использовать операционные системы и приложения так же, как на физическом устройстве.

Ключевые особенности:

* *Абстрагирует аппаратные ресурсы.*
* *Обеспечивает изоляцию между системами.*
* *Поддерживает работу нескольких операционных систем на одном физическом устройстве.*

1. ***В чём отличие мультипрограммных систем и систем виртуальных машин?***

* Мультипрограммные системы:
  + *Позволяют запускать несколько программ (процессов) одновременно на одной операционной системе.*
  + *Процессы разделяют физические ресурсы, такие как процессорное время и память.*
  + *Работают в рамках одной ОС, где задачи координируются ядром.*
* Системы виртуальных машин:
  + *Эмулируют целые операционные системы, включая их ядро, предоставляя изолированную среду для работы.*
  + *На одном физическом компьютере могут работать несколько независимых ВМ, каждая со своей ОС.*
  + *Используют гипервизор для управления виртуализацией.*

1. ***Для чего предназначена программа VirtulBox?***

Oracle VirtualBox — это инструмент для виртуализации, предназначенный для создания и управления виртуальными машинами.

Основные функции:

* *Позволяет запускать гостевые операционные системы на хостовой ОС.*
* *Поддерживает множество форматов ОС, включая Windows, Linux, macOS и другие.*
* *Обеспечивает доступ к виртуальным ресурсам, таким как виртуальная память, сеть, накопители.*
* *Подходит для тестирования ПО, обучения и работы с различными ОС.*

1. ***Как может быть установлена гостевая операционная система?***

Гостевая ОС может быть установлена в виртуальной машине следующим образом:

* Подготовка VirtualBox или другого гипервизора:
  + *Установите VirtualBox на хостовую операционную систему.*
  + *Создайте новую виртуальную машину, указав параметры (имя, тип ОС, объём памяти, размер диска).*
* Установка ISO-образа или загрузочного диска:
  + *Укажите ISO-файл или физический установочный носитель (диск, флешку) в настройках ВМ.*
  + *Запустите виртуальную машину.*
* Процесс установки ОС:
  + *Следуйте стандартному процессу установки гостевой ОС, как на физическом компьютере.*
  + *Настройте язык, разделы диска, параметры пользователя.*
* Настройка гостевых дополнений (опционально):
  + *Установите дополнения для улучшения взаимодействия между хостовой и гостевой ОС (например, общий буфер обмена, доступ к файлам).*

|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_

Отчет по лабораторной работе № 2(Ubuntu)

**«ОС Ubuntu. Управление пользователями»**

по дисциплине «Операционные системы»

Студент ИУ5-51Б Т.А. Цыпышев

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель П.С. Семкин

(Подпись, дата) (И.О.Фамилия)

**Москва**

**2024**

**Цель работы**

Целью работы является знакомство с политикой учётных записей пользователей и групп пользователей в операционных системах семейства Linux.

**Задание**

1. Создать учётные записи пользователей
2. Создать учётную запись группы пользователей
3. Включить пользователей в группу
4. Назначить пользователю права администратора

**Порядок выполнения**

1. Создание пользователей  
   Создание учётных записей пользователей с использованием useradd  
   sudo useradd -m -s /bin/bash admin\_kaf  
   sudo useradd -m -s /bin/bash stud\_51  
   sudo useradd -m -s /bin/bash stud\_52  
   sudo useradd -m -s /bin/bash stud\_53  
   sudo useradd -m -s /bin/bash stud\_54  
   sudo useradd -m -s /bin/bash stud\_55
2. Установка паролей для пользователей  
   echo "admin\_kaf:adminkaf" | sudo chpasswd  
   echo "stud\_51:stud51" | sudo chpasswd  
   echo "stud\_52:stud52" | sudo chpasswd  
   echo "stud\_53:stud53" | sudo chpasswd  
   echo "stud\_54:stud54" | sudo chpasswd  
   echo "stud\_55:stud55" | sudo chpasswd
3. Создание группы пользователей  
   Создание группы  
   sudo groupadd group\_stud3k
4. Добавление пользователей в группу  
   sudo usermod -aG group\_stud3k stud\_51  
   sudo usermod -aG group\_stud3k stud\_52  
   sudo usermod -aG group\_stud3k stud\_53  
   sudo usermod -aG group\_stud3k stud\_54  
   sudo usermod -aG group\_stud3k stud\_55
5. Назначение пользователю admin\_kaf прав администратора  
   Добавление пользователя admin\_kaf в группу sudo  
   sudo usermod -aG wheel admin\_kaf
6. Удаление учётной записи student  
   sudo useradd -m -s /bin/bash student   
   sudo userdel -r student
7. Проверка правильности создания пользователей  
   Просмотр информации о пользователях в файлах /etc/passwd и /etc/group  
   cat /etc/passwd | grep 'admin\_kaf\|stud\_51\|stud\_52\|stud\_53\|stud\_54\|stud\_55'/  
   cat /etc/group | grep 'group\_stud3k'
8. Наблюдение за пользователями  
   find / -user admin\_kaf - Ищет все файлы и директории на системе, принадлежащие пользователю admin\_kaf  
   du -sh ~/ - Отображает размер домашней директории текущего пользователя  
   fuser -m /home/student - Показывает процессы, использующие файлы или директории в /home/student  
   ps aux - Отображает список всех запущенных процессов в системе  
   top - Запускает интерактивный монитор системы, отображающий обновляющийся список текущих процессов  
   last - Показывает историю входов пользователей в систему  
   lastb - Показывает историю неудачных попыток входа в систем  
   who -u - Показывает информацию о текущих пользователях, находящихся в системе  
   users - Выводит список пользователей, которые в данный момент вошли в систему  
   finger - Показывает подробную информацию о пользователях, которые в данный момент вошли в систему

**Контрольные вопросы**

1. Какие типы пользователей существуют в ОС Linux?

В Linux существуют три основных типа пользователей:

1. Суперпользователь (root):
   * *Имеет полные права доступа ко всей системе.*
   * *Может выполнять любые действия, включая управление пользователями, установку и удаление программ, изменение системных настроек.*
2. Обычные пользователи:
   * *Ограничены в правах.*
   * *Имеют доступ только к своим личным файлам и каталогам.*
   * *Для выполнения административных задач требуется использование команды sudo.*
3. Системные пользователи:
   * *Создаются для работы системных сервисов и процессов.*
   * *Обычно не имеют пароля и не используют графический интерфейс.*
   * *Примеры: www-data для веб-серверов, nobody для программ с минимальными привилегиями.*

2. Какими правами обладает пользователь root?

Пользователь root обладает следующими правами:

* *Полный доступ ко всем файлам и каталогам в системе, независимо от их владельца и настроек доступа.*
* *Возможность устанавливать, удалять и обновлять программное обеспечение.*
* *Управление пользователями и их правами (создание, удаление, изменение паролей и групп).*
* *Изменение системных настроек, таких как конфигурация сети или ядра.*
* *Исполнение любых команд без ограничений.*
* *Монтирование и размонтирование файловых систем.*

3. В чём назначение утилиты sudo?

Утилита sudo (SuperUser DO) позволяет обычным пользователям выполнять команды с правами суперпользователя, не входя в систему под root.

Назначение:

* *Увеличение безопасности: root-пароль не передаётся обычным пользователям.*
* *Логирование действий: все команды, выполняемые через sudo, записываются в системный журнал.*
* *Гибкое управление правами: можно предоставить доступ только к определённым командам.*

4. Как можно назначить пользователю права sudo?

Чтобы назначить пользователю права *sudo*, выполните следующие шаги:

1. Добавление в группу sudo:  
    *sudo usermod -aG sudo имя\_пользователя*
2. Проверка конфигурации в файле /etc/sudoers:
   * *Отредактируйте файл с помощью команды visudo для предотвращения ошибок:  
      sudo visudo*
   * *Убедитесь, что запись для группы sudo или пользователя присутствует:  
      %sudo ALL=(ALL:ALL) ALL*
3. Проверка прав:
   * *Выйдите и снова войдите в систему под именем пользователя.*
   * *Убедитесь, что команды с sudo работают.*

5. В какие группы может входить пользователь?

Пользователь может входить в различные группы, которые определяют его права доступа.

Основные типы групп:

1. *Основная группа:*
   * Назначается при создании пользователя.
   * Обычно совпадает с именем пользователя.
2. *Дополнительные группы:*
   * Определяют доступ к определённым ресурсам.
   * Примеры:
     + *sudo* — права суперпользователя.
     + *docker* — управление Docker-контейнерами.
     + *adm* — доступ к системным журналам.
     + *dialout* — доступ к последовательным портам.
     + *www-data* — права для работы с веб-серверами.

Просмотреть группы, в которых состоит пользователь, можно командой: *groups имя\_пользователя*

|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_

Отчет по лабораторной работе № 3(Ubuntu)

**«ОС Ubuntu. Основные права доступа к файлам и каталогам»**

по дисциплине «Операционные системы»

Студент ИУ5-51Б Т.А. Цыпышев

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель П.С. Семкин

(Подпись, дата) (И.О.Фамилия)

**Москва**

**2024**

**Цель работы**

Целью работы является изучение объектов файловой системы ОС Ubuntu и установка основных прав доступа к файлам и каталогам.

**Задание**

1. Создать каталоги и файлы и установить права доступа к ним для различных пользователей.
2. В корневом каталоге файловой системы создать каталог КАФЕДРА.
3. В каталоге КАФЕДРА создать каталоги

ДОКУМЕНТЫ\_КАФЕДРЫ

ДОКУМЕНТЫ\_ОБЩИЕ

ДОКУМЕНТЫ\_ГРУППЫ\_51, … ДОКУМЕНТЫ ГРУППЫ\_55

1. Установить права доступа к каталогам и файлам:
   1. Пользователь admin\_kaf должен иметь полный доступ к каталогам

ДОКУМЕНТЫ\_КАФЕДРЫ

ДОКУМЕНТЫ\_ОБЩИЕ

* 1. К каталогу ДОКУМЕНТЫ\_КАФЕДРЫ пользователям stud51-stud55 разрешить доступ только для чтения
  2. К каталогу ДОКУМЕНТЫ\_ОБЩИЕ разрешить полный доступ пользователям группы group\_stud3k
  3. К каталогам ДОКУМЕНТЫ\_ГРУППЫ\_51, … ДОКУМЕНТЫ ГРУППЫ\_55 разрешить полный доступ только для пользователей stud51 –stud55 соответственно. Остальным членам группы разрешить чтение. Прочим пользователям доступ запретить

**Порядок выполнения**

1. Создание каталога КАФЕДРА  
   sudo mkdir /КАФЕДРА
2. Изменение владельца каталога КАФЕДРА  
   Измените владельца на admin\_kaf и группу на admin\_kaf:  
   sudo chown admin\_kaf:admin\_kaf /КАФЕДРА
3. Создание каталогов в КАФЕДРА  
   Создайте необходимые каталоги:  
   sudo mkdir /КАФЕДРА/ДОКУМЕНТЫ\_КАФЕДРЫ  
   sudo mkdir /КАФЕДРА/ДОКУМЕНТЫ\_ОБЩИЕ  
   sudo mkdir /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_51  
   sudo mkdir /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_52  
   sudo mkdir /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_53  
   sudo mkdir /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_54  
   sudo mkdir /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_55
4. Установка владельцев каталогов  
   Установите владельцев для каталогов ДОКУМЕНТЫ\_ГРУППЫ\_51 и т.д.:  
   sudo chown stud\_51:group\_stud3k /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_51  
   sudo chown stud\_52:group\_stud3k /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_52  
   sudo chown stud\_53:group\_stud3k /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_53  
   sudo chown stud\_54:group\_stud3k /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_54  
   sudo chown stud\_55:group\_stud3k /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_55
5. Установка разрешений для файлов и каталогов  
   Установите необходимые разрешения. Например:  
   sudo chmod 770 /КАФЕДРА  
   sudo chmod 755 /КАФЕДРА/ДОКУМЕНТЫ\_КАФЕДРЫ  
   sudo chmod 755 /КАФЕДРА/ДОКУМЕНТЫ\_ОБЩИЕ  
   sudo chmod 755 /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_51  
   sudo chmod 755 /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_52  
   sudo chmod 755 /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_53  
   sudo chmod 755 /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_54  
   sudo chmod 755 /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_55
6. Создание файлов в каталогах  
   Создайте файлы в соответствующих каталогах:  
   echo "Список группы ИУ5-XX" > /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_51/Список\_группы\_ИУ5-XX.txt  
   echo "Список группы ИУ5-XX" > /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_52/Список\_группы\_ИУ5-XX.txt  
   echo "Список группы ИУ5-XX" > /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_53/Список\_группы\_ИУ5-XX.txt  
   echo "Список группы ИУ5-XX" > /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_54/Список\_группы\_ИУ5-XX.txt  
   echo "Список группы ИУ5-XX" > /КАФЕДРА/ДОКУМЕНТЫ\_ГРУППЫ\_55/Список\_группы\_ИУ5-XX.txt
7. Проверка правильности установки разрешений  
   Проверьте права доступа с помощью команды:  
   ls -l /КАФЕДРА

**Контрольные вопросы**

1. Какие типы файлов поддерживает ОС Ubuntu?

В ОС Ubuntu (и Linux в целом) поддерживаются следующие типы файлов:

1. Обычные файлы (regular files):
   * *Это текстовые, бинарные или исполняемые файлы.*
   * *Обозначаются символом - в начале строки при выводе команды ls -l.*
2. Каталоги (directories):
   * *Представляют собой контейнеры для других файлов и каталогов.*
   * *Обозначаются символом d.*
3. Символические ссылки (symbolic links):
   * *Указатели на другие файлы или каталоги.*
   * *Обозначаются символом l.*
4. Устройства (device files):
   * *Файлы, представляющие аппаратные устройства:*
     + *Блочные устройства (block devices): например, жёсткие диски (b).*
     + *Символьные устройства (character devices): например, последовательные порты (c).*
5. FIFO или именованные каналы:
   * *Используются для обмена данными между процессами.*
   * *Обозначаются символом p.*
6. Сокеты (sockets):
   * *Используются для сетевого взаимодействия и межпроцессного обмена.*
   * *Обозначаются символом s.*

2. Что такое файловые разрешения и файловые атрибуты?

Файловые разрешения — это правила, определяющие, кто и какие действия может выполнять с файлом или каталогом. Разрешения определяются тремя типами операций:

* ***r*** *(read) — чтение файла или просмотр содержимого каталога.*
* ***w*** *(write) — запись в файл или изменение содержимого каталога.*
* ***x*** *(execute) — выполнение файла или доступ к каталогу.*

Файловые атрибуты — это дополнительные метаданные, задающие поведение файла в системе:

* *Атрибуты прав доступа (например, sticky bit, setuid, setgid).*
* *Расширенные атрибуты (например, метки SELinux).*
* *Флаги файловой системы (например, защита от удаления или изменения).*

Атрибуты можно просмотреть с помощью команды:  
*lsattr имя\_файла*

3. Какие категории пользователей определяют файловые разрешения?

Файловые разрешения в Linux определяются для следующих категорий пользователей:

1. Владелец (user):
   * *Учетная запись, создавшая файл или каталог.*
   * *Имеет персональный набор разрешений.*
2. Группа (group):
   * *Группа, к которой принадлежит файл или каталог.*
   * *Все пользователи этой группы наследуют разрешения, установленные для группы.*
3. Прочие (others):
   * *Все остальные пользователи системы, не являющиеся владельцем или членами группы файла.*

Пример вывода команды ls -l:  
*-rwxr-xr-- 1 user group 4096 Jan 7 filename*

Обозначает:

* *rwx — права владельца.*
* *r-x — права группы.*
* *r-- — права для всех остальных.*

|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_

Отчет по лабораторной работе № 4(Ubuntu)

**«ОС Ubuntu. Расширенные права доступа к файлам и каталогам»**

по дисциплине «Операционные системы»

Студент ИУ5-51Б Т.А. Цыпышев

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель П.С. Семкин

(Подпись, дата) (И.О.Фамилия)

**Москва**

**2024**

**Цель работы**

Целью работы является знакомство с расширенными правами доступа к файлам и каталогам ОС Ubuntu.

**Задание**

1. Создать каталог с общим доступом(shared) ДЕКАНАТ и обеспечить доступ к данному каталогу группы пользователей, используя расширенные разрешения (биты setgid и sticky).
2. Обеспечить разграничение доступа пользователей группы group\_dek к подкаталогам каталога ДЕКАНАТ, используя ACL.

**Порядок выполнения**

Создание каталога с общим доступом(shared)

1. Создание учетных записей пользователей  
   sudo adduser dekan  
   sudo adduser sotrudnik  
   sudo adduser admin\_dek  
   При создании каждого пользователя будет предложено ввести пароль. Установите пароль dekanat для всех пользователей, но будет dek4nat!
2. Создание группы и добавление пользователей в группу
3. Создайте группу group\_dek:  
   sudo groupadd group\_dek

Добавьте пользователей в группу group\_dek:  
sudo usermod -aG group\_dek dekan  
sudo usermod -aG group\_dek sotrudnik  
sudo usermod -aG group\_dek admin\_dek

Добавьте пользователей в группу sudo:  
sudo usermod -aG sudo dekan  
sudo usermod -aG sudo sotrudnik  
sudo usermod -aG sudo admin\_dek

1. Создание каталога ДЕКАНАТ  
   Создайте каталог ДЕКАНАТ в корневом каталоге:  
   sudo mkdir /ДЕКАНАТ

Установите группу владельцев каталога на group\_dek:  
sudo chown admin\_dek:group\_dek /ДЕКАНАТ

1. Изменение владельца и группы каталога  
   Убедитесь, что владелец каталога ДЕКАНАТ — admin\_dek:  
   ls -ld /ДЕКАНАТ  
   Вывод должен показывать admin\_dek как владельца и group\_dek как группу.
2. Установка прав доступа  
   Установите права доступа для каталога, чтобы члены группы могли создавать файлы:  
   sudo chmod 2770 /ДЕКАНАТ  
   Здесь 2 устанавливает бит SGID, который позволяет всем файлам, создаваемым в каталоге, наследовать группу group\_dek.

Убедитесь, что все файлы, созданные в каталоге, будут принадлежать группе group\_dek:  
sudo chmod g+s /ДЕКАНАТ

Чтобы пользователи могли удалять только свои файлы, нужно установить права так, чтобы разрешать удаление файлов только их владельцам. В данном случае стандартные права rwx для пользователя, группы и --x для других подходят.

1. Создание подкаталогов  
   Создайте каталоги ПРИКАЗЫ и ИНФОРМАЦИЯ внутри ДЕКАНАТ:  
   sudo mkdir /ДЕКАНАТ/ПРИКАЗЫ  
   sudo mkdir /ДЕКАНАТ/ИНФОРМАЦИЯ

Установите для них права:  
sudo chown admin\_dek:group\_dek /ДЕКАНАТ/ПРИКАЗЫ  
sudo chown admin\_dek:group\_dek /ДЕКАНАТ/ИНФОРМАЦИЯ  
sudo chmod 2770 /ДЕКАНАТ/ПРИКАЗЫ  
sudo chmod 2770 /ДЕКАНАТ/ИНФОРМАЦИЯ

1. Проверка корректности созданных каталогов  
   Проверьте права доступа к каталогам:  
   ls -l /ДЕКАНАТ
2. Вход с учетными записями и создание файлов  
   Создайте файл Приказ061121.txt в каталоге /ДЕКАНАТ/ПРИКАЗЫ:  
   touch /ДЕКАНАТ/ПРИКАЗЫ/Приказ061121.txt

Для пользователя sotrudnik создайте файл Новости.txt в каталоге /ДЕКАНАТ/ИНФОРМАЦИЯ:  
touch /ДЕКАНАТ/ИНФОРМАЦИЯ/Новости.txt

1. Проверка прав доступа к созданным файлам  
   Проверьте права доступа к созданным файлам:  
   ls -l /ДЕКАНАТ/ПРИКАЗЫ/Приказ061121.txt  
   ls -l /ДЕКАНАТ/ИНФОРМАЦИЯ/Новости.txt  
   Убедитесь, что файлы принадлежат соответствующим пользователям и группе group\_dek.

Управление расширенными правами доступа к файлам и каталогам с использованием ACL

1. Проверка текущих прав доступа к каталогу ПРИКАЗЫ  
   В терминале выполните команду getfacl, чтобы проверить текущие права доступа к каталогу ПРИКАЗЫ:  
   getfacl /ДЕКАНАТ/ПРИКАЗЫ  
   Это покажет текущие права доступа, включая стандартные права и любые ACL, которые могут быть установлены.
2. Установка прав доступа для пользователей dekan и sotrudnik  
   Установите права доступа для пользователя dekan (полный доступ) и sotrudnik (только чтение) к каталогу ПРИКАЗЫ с помощью команды setfacl:  
   sudo setfacl -m u:dekan:rwx /ДЕКАНАТ/ПРИКАЗЫ # Полный доступ для dekan  
   sudo setfacl -m u:sotrudnik:r /ДЕКАНАТ/ПРИКАЗЫ # Только чтение для sotrudnik
3. Проверка текущих прав доступа к каталогу ПРИКАЗЫ  
   Снова выполните команду getfacl, чтобы убедиться, что права были установлены корректно:  
   getfacl /ДЕКАНАТ/ПРИКАЗЫ  
   Убедитесь, что в выводе присутствуют записи для пользователей dekan и sotrudnik с правильными правами доступа.
4. Проверка текущих прав доступа к каталогу ИНФОРМАЦИЯ  
   Выполните команду getfacl, чтобы проверить текущие права доступа к каталогу ИНФОРМАЦИЯ:  
   getfacl /ДЕКАНАТ/ИНФОРМАЦИЯ
5. Установка прав доступа для пользователей sotrudnik и dekan к каталогу ИНФОРМАЦИЯ  
   Установите права доступа для пользователя sotrudnik (полный доступ) и dekan (только чтение) к каталогу ИНФОРМАЦИЯ:  
   sudo setfacl -m u:sotrudnik:rwx /ДЕКАНАТ/ИНФОРМАЦИЯ # Полный доступ для sotrudnik  
   sudo setfacl -m u:dekan:r /ДЕКАНАТ/ИНФОРМАЦИЯ # Только чтение для dekan
6. Проверка текущих прав доступа к каталогу ИНФОРМАЦИЯ  
   Выполните команду getfacl, чтобы убедиться, что права были установлены корректно:  
   getfacl /ДЕКАНАТ/ИНФОРМАЦИЯ

**Контрольные вопросы**

1. Каково назначение расширенных разрешений setuid, setgid и sticky bit?

setuid (Set User ID):

* *Позволяет исполняемому файлу запускаться с правами владельца файла, а не пользователя, который его запускает.*
* *Обычно используется для выполнения задач, требующих повышенных привилегий.*
* *Пример: файл /bin/passwd (используется для изменения пароля), который должен иметь доступ к защищённым файлам.  
  Установка setuid:  
  chmod u+s имя\_файла*

setgid (Set Group ID):

* *Для файлов: аналогично setuid, но действие распространяется на группу файла. Исполняемый файл запускается с правами группы, а не текущего пользователя.*
* *Для каталогов: файлы, создаваемые внутри каталога с setgid, наследуют группу каталога (а не группу текущего пользователя).  
  Установка setgid:  
  chmod g+s имя\_файла\_или\_каталога*

Sticky bit:

* *Применяется к каталогам. Ограничивает удаление или переименование файлов в каталоге только их владельцем или суперпользователем, даже если у других есть права на запись.*
* *Обычно используется для общедоступных каталогов, таких как /tmp.  
  Установка sticky bit:  
  chmod +t имя\_каталога*

2. В чём отличие задания расширенных разрешений в абсолютном и относительном режимах?

Абсолютный режим:

* *Указывается числовое значение разрешений (включая расширенные флаги, такие как setuid, setgid и sticky bit).*
* *Формат: четырёхзначное число, где первый разряд отвечает за расширенные разрешения, а остальные три — за стандартные.*
  + *Пример:  
    4755 — устанавливает setuid (4), а также права владельца, группы и остальных (755).*

*Установка в абсолютном режиме:  
chmod 4755 имя\_файла*

Относительный режим:

* *Устанавливает или снимает конкретные разрешения относительно текущих прав.*
* *Используются символы (u, g, o, a) для указания категорий пользователей и флаги (+, -, =) для добавления, удаления или назначения прав.*
  + *Пример:*
    - *chmod u+s — добавляет флаг setuid.*
    - *chmod g-s — удаляет флаг setgid.*

3. Как назначаются права при использовании ACL?

ACL (Access Control List) — это расширенная система прав доступа, позволяющая задавать индивидуальные права для конкретных пользователей или групп.

1. Установка прав:

*setfacl -m u:имя\_пользователя:права имя\_файла*

* + *u:имя\_пользователя — права для пользователя.*
  + *g:имя\_группы — права для группы.*
  + *права — комбинация r, w, x.*

*Пример:  
 setfacl -m u:john:rwx example.txt*

1. Просмотр текущих прав ACL:  
   *getfacl имя\_файла*
2. Удаление прав:  
   *setfacl -x u:имя\_пользователя имя\_файла*
3. Установка маски:  
   *Маска (mask) ограничивает максимальные права для всех дополнительных записей ACL.  
   setfacl -m m::rwx имя\_файла*

Преимущество ACL:  
Позволяет задавать более гибкие настройки прав, чем традиционные разрешения, например, одновременно предоставлять доступ нескольким конкретным пользователям без изменения группы.

|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления\_\_\_\_\_\_\_\_\_\_\_

Отчет по лабораторной работе № 5(Ubuntu)

**«ОС Ubuntu. Управление процессами»**

по дисциплине «Операционные системы»

Студент ИУ5-51Б Т.А. Цыпышев

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель П.С. Семкин

(Подпись, дата) (И.О.Фамилия)

**Москва**

**2024**

**Цель работы**

Целью работы является знакомство со средствами мониторинга и управления процессами ОС Ubuntu.

**Задание**

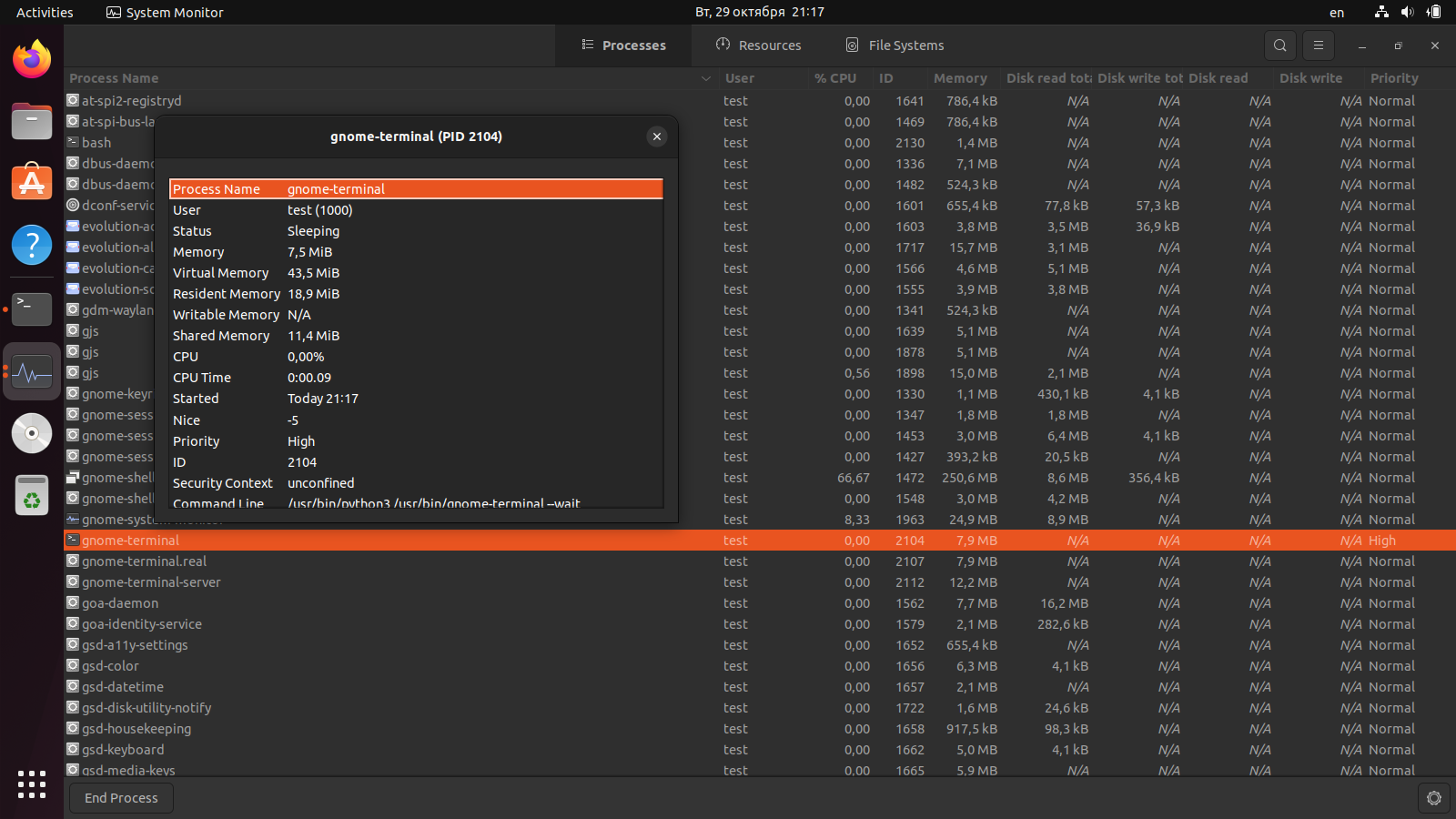
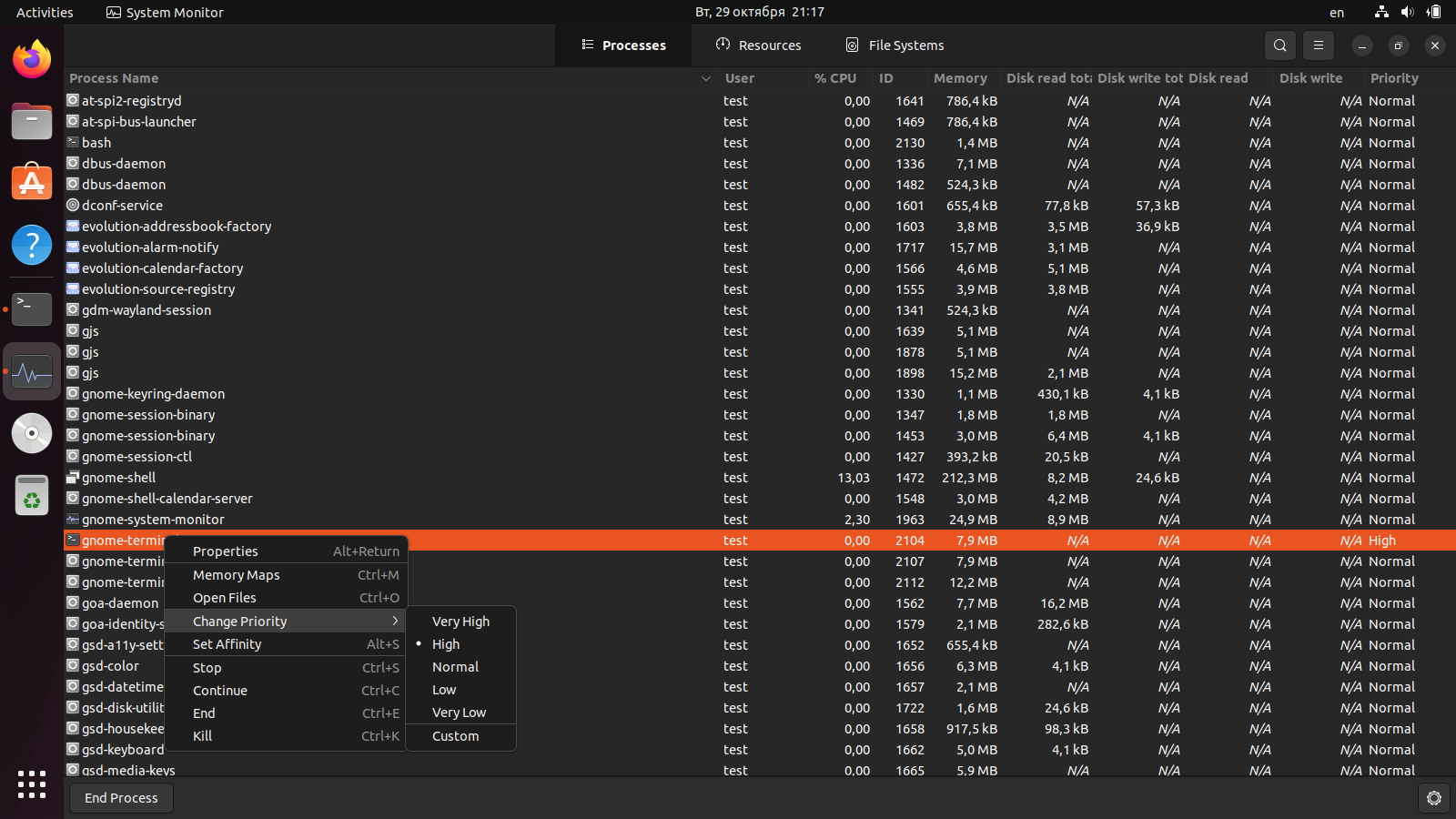
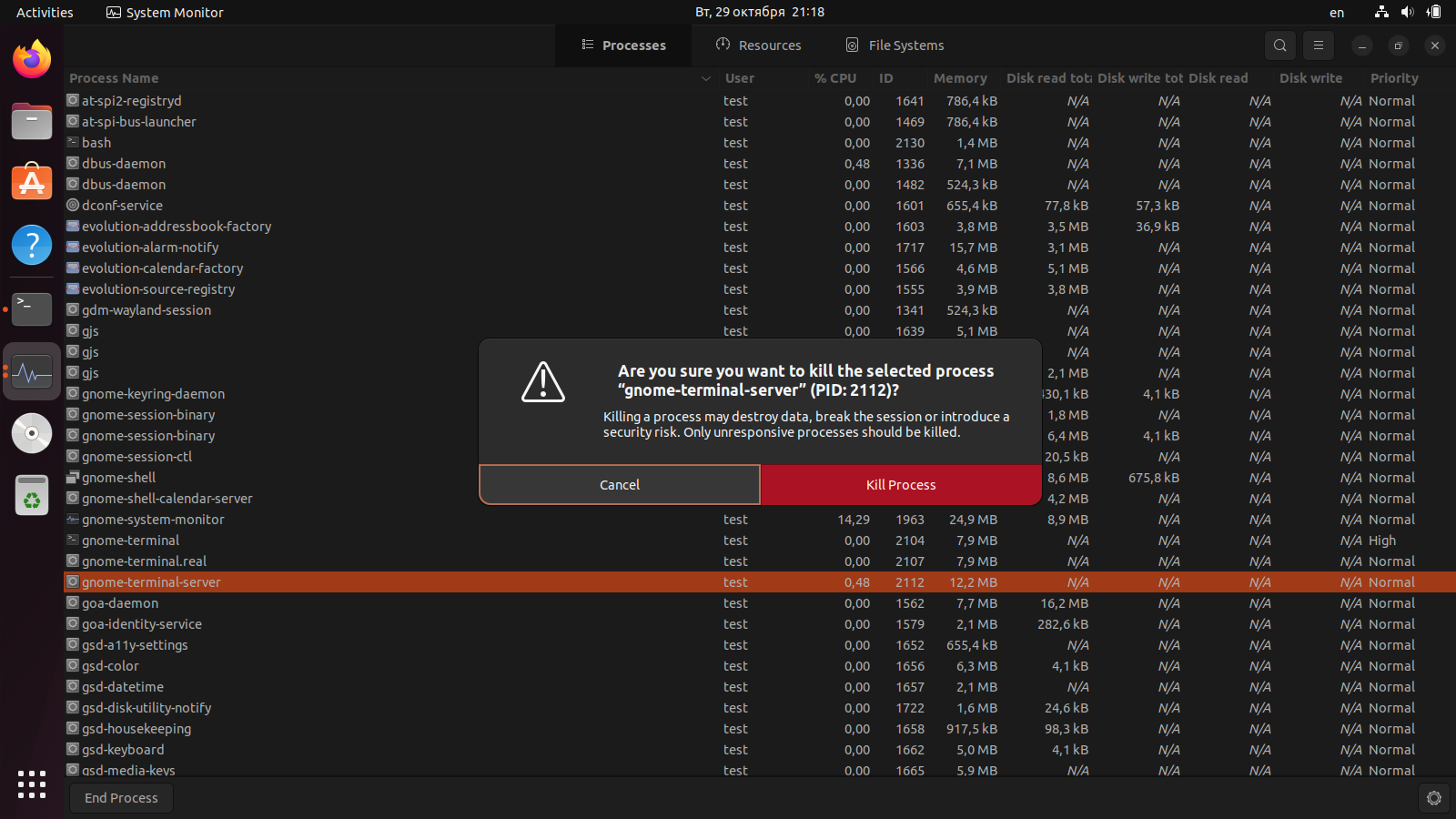
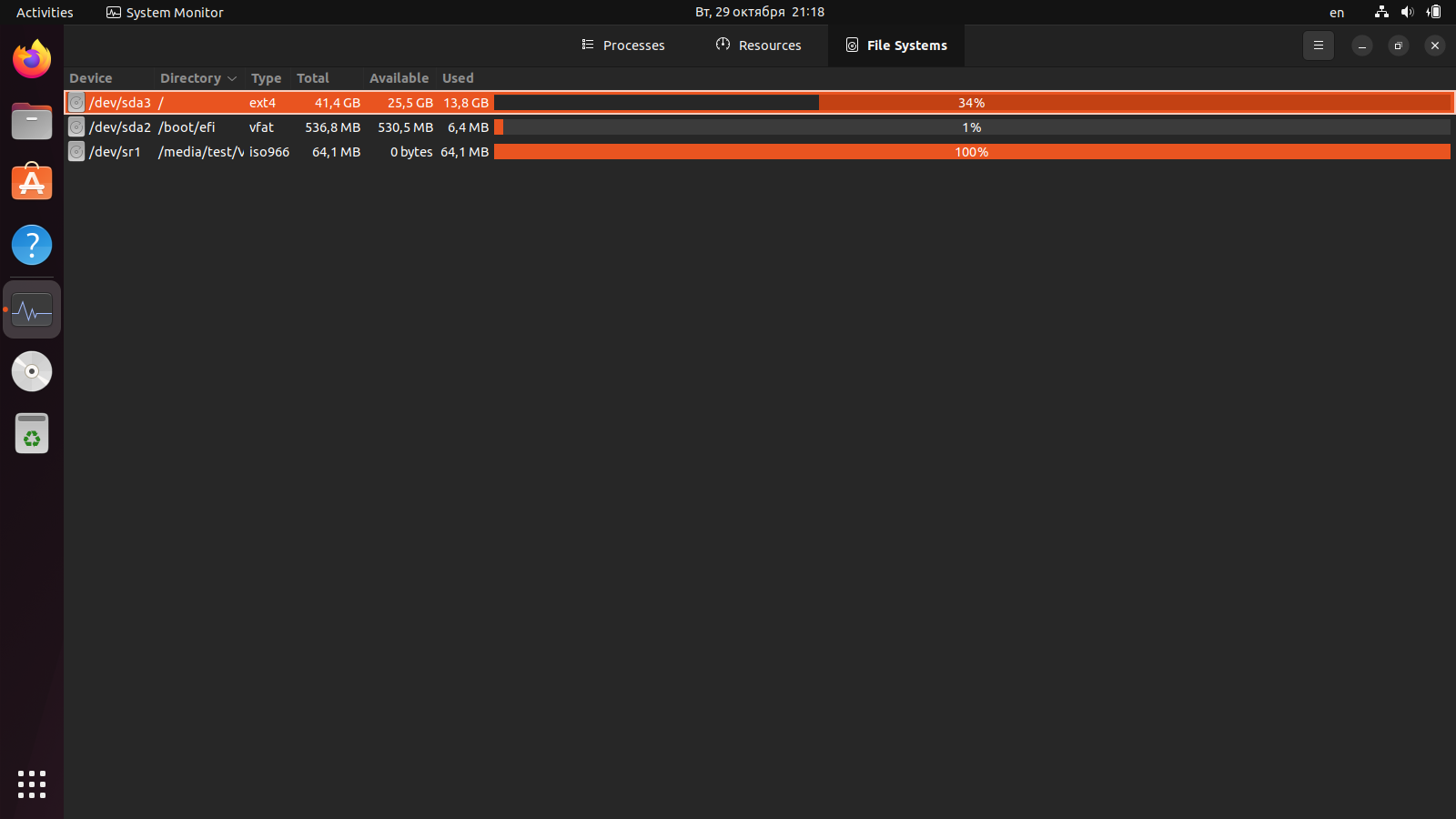
Познакомиться с возможности утилит Ubuntu для решения задач мониторинга и управления процессами операционной системы.

Выполнить команды управления процессами в соответствие с порядком

выполнения работы и объяснить полученные результаты.

**Порядок выполнения**

Мониторинг и управление процессами с помощью графической утилиты «Системный монитор»

1. Информация о процессе  
   
2. Приоритет процесса  
   
3. Убийство процесса  
   
4. Ресурсы  
   
5. Файловое хранилище  
   

Мониторинг и управление процессами с помощью утилит командной строки

1. Получить общую информацию о системе
   1. Информация о текущем интерпретаторе команд:  
      echo $SHELL
   2. Информация о текущем пользователе:  
      whoami
   3. Информация о текущем каталоге:  
      pwd
   4. Информация об оперативной памяти и области подкачки:  
      free -h
   5. Информация о дисковой памяти:  
      df -h
2. Получение информации о процессах
   1. Получение идентификатора текущего процесса (PID):  
      echo $$
   2. Получение идентификатора родительского процесса (PPID):  
      echo $PPID
   3. Получение идентификатора процесса по его имени (например, init):  
      pgrep init
   4. Получение информации о выполняющихся процессах с помощью команды ps:  
      ps aux

Этот вывод покажет все процессы, выполняющиеся в системе, с информацией о пользователе, PID, CPU, памяти и т.д.

1. Выполнение команд управления процессами
   1. Получение информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе:  
      ps -u $(whoami)
   2. Определение текущего значения приоритета по умолчанию:  
      nice -n 0
   3. Запуск командного процессора bash с понижением приоритета:  
      nice -n 10 bash  
      (Здесь 10 — это значение nice, указывающее понижение приоритета.)
   4. Определение PID запущенного командного процессора:  
      После запуска команды, в новом терминале выполните:  
      echo $$  
      Если вы запустили новый процесс bash, PID будет отображен.
   5. Установка приоритета командного процессора равным 5:  
      Вы можете использовать команду renice:  
      sudo renice 5 -p <PID>  
      Замените <PID> на идентификатор процесса, полученный на предыдущем шаге.
   6. Получение информации об активных процессах пользователя, включая их приоритеты, с помощью утилиты top:  
      Запустите команду:  
      top  
      В верхней части экрана будет показана информация о текущих процессах и их приоритетах (в столбце NI).

**Контрольные вопросы**

1. Перечислите состояния задачи в ОС Ubuntu

В ОС Ubuntu задача (процесс) может находиться в одном из следующих состояний:

1. Running (R) — выполняется:
   * *Задача активно выполняется на процессоре или готова к выполнению.*
2. Sleeping (S) — спящий режим:
   * *Ожидание события (например, ввода/вывода).*
   * *Состояние может быть прерываемым (задачу можно пробудить сигналом).*
3. Uninterruptible Sleep (D) — непрерываемый сон:
   * *Задача ожидает завершения операции ядра (например, операций с устройствами).*
4. Stopped (T) — остановлена:
   * *Выполнение задачи приостановлено сигналом (например, SIGSTOP).*
5. Zombie (Z) — зомби-процесс:
   * *Завершённый процесс, который ещё не был удалён родительским процессом.*
6. Idle (I) — бездействующий:
   * *Процессы ядра, которые не выполняют активной работы (обычно применимо к системным потокам).*

2. Как создаются задачи в ОС Ubuntu?

Задачи (процессы) в Ubuntu создаются следующими способами:

1. Создание нового процесса (fork):  
   *Используется системный вызов fork(), который создаёт копию текущего процесса.*
2. Создание нового процесса и выполнение программы (exec):  
   *После fork() можно загрузить новую программу в память с помощью вызова exec().*
3. Использование утилит:  
   *Например, команда bash для запуска нового процесса в оболочке:  
    ./my\_program*
4. Планировщики задач:  
   *Использование команд типа cron или at для выполнения задач по расписанию.*
5. Создание потоков:  
   *Используется библиотека pthread или аналогичные механизмы для создания потоков внутри одного процесса.*

3. Назовите классы потоков ОС Ubuntu

В Ubuntu (и Linux в целом) потоки классифицируются в соответствии с их приоритетом и политикой планирования:

1. SCHED\_NORMAL (или SCHED\_OTHER):
   * *Потоки обычного пользовательского пространства с динамическим приоритетом.*
   * *Подразумевает time-sharing (разделение времени).*
2. SCHED\_FIFO (First-In-First-Out):
   * *Реальное время с фиксированным приоритетом.*
   * *Поток выполняется до завершения или пока его не прервёт поток с более высоким приоритетом.*
3. SCHED\_RR (Round-Robin):
   * *Реальное время с фиксированным приоритетом.*
   * *Использует циклическую очередность: каждому потоку выделяется квант времени.*
4. SCHED\_IDLE:
   * *Низкий приоритет, используется для задач, выполняемых только в случае простоя системы.*
5. SCHED\_BATCH:
   * *Для фоновых задач, которые не требуют высокой интерактивности.*
6. SCHED\_DEADLINE:
   * *Для задач реального времени с дедлайном (доступно в современных ядрах Linux).*

4. Как используется приоритет планирования при запуске задачи?

Приоритет планирования определяет, сколько процессорного времени выделяется задаче.

1. Нормальные задачи (SCHED\_NORMAL):
   * *Используется динамический приоритет.*
   * *Процесс запускается с начальным приоритетом (от 0 до 19, где 0 — максимальный приоритет).*
2. Реа*льное время (SCHED\_FIFO, SCHED\_RR):*
   * *Фиксированный приоритет в диапазоне от 1 до 99, где 99 — максимальный.*

*При запуске задачи можно указать её приоритет с помощью утилиты nice:  
nice -n <приоритет> ./my\_program*

* + *Диапазон приоритетов: от -20 (высший) до 19 (низший).*

5. Как можно изменить приоритет для выполняющейся задачи?

Для изменения приоритета выполняющейся задачи используется команда renice:

1. Синтаксис команды:  
    *renice <новый\_приоритет> -p <PID>*
   * *<новый\_приоритет>: от -20 (высший) до 19 (низший).*
   * *<PID>: идентификатор процесса.*

*Пример:  
 renice 10 -p 1234*

1. Изменение приоритета задачи реального времени:  
   * *Используйте утилиту chrt для управления политикой и приоритетом реального времени:  
      chrt -p -r <приоритет> <PID>*
   * *Например:  
      chrt -p -r 50 1234*
2. Права доступа:
   * *Только суперпользователь (или пользователь с правами sudo) может повышать приоритет процесса или изменять приоритеты для задач реального времени.*