Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение

высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

им. Н.П. ОГАРЁВА»

(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

Факультет математики и информационных технологий

Кафедра систем автоматизированного проектирования

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине: Операционные системы

КОНТРОЛЬ ИСПОЛЬЗОВАНИЯ РЕСУРСОВ ОС LINUX

Автор отчёта о лабораторной работе  А. Е. Конышев

подпись, дата

Обозначение лабораторной работы ЛР–02069964–02.03.02–08–23

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Руководитель работы

канд. техн. наук, доц.  А. В. Шамаев

подпись, дата

Саранск 2023

**Цель работы:** Практическое знакомство с командами, используемыми для контроля использования ресурсов и виртуальной файловой системой.

**Ход работы:**

1. Ознакомиться с теоретическим материалом.
2. Выполнить задания.
3. Ответить на контрольные вопросы.

**Задание:**

1. Вывести список всех процессов в системе.
2. Вывести дерево процессов.
3. С помощью команды top получить список процессов, потребляющих наибольшее количество процессорного времени.
4. Найти 2 процесса, имеющих более ДВУХ потоков.
5. Используя команду top, изменить приоритеты 2 процессов.
6. Получить список открытых файлов текущего пользователя.
7. Получить текущее состояние системной памяти.
8. Получить справку об использовании дискового пространства.
9. Вывести информацию о каком-либо процессе, используя содержимое каталога /proc.
10. Вывести информацию о процессоре ПК, используя содержимое каталога /proc.
11. Вывести список модулей, используемых в настоящий момент ядром ОС.

**Описание выполнения работы**

1. Вывести список всех процессов в системе.

Для вывода списка всех выполняющихся на компьютере в текущий момент процессах используется команда: ps aux

Значения используемых опций: а - all – процессы всех пользователей; u – ориентированная на пользователей (отображение информации о владельце); x – процессы, не контролируемые ttys.

Результат выполнения данной команды, представлен на рисунке 5.1.

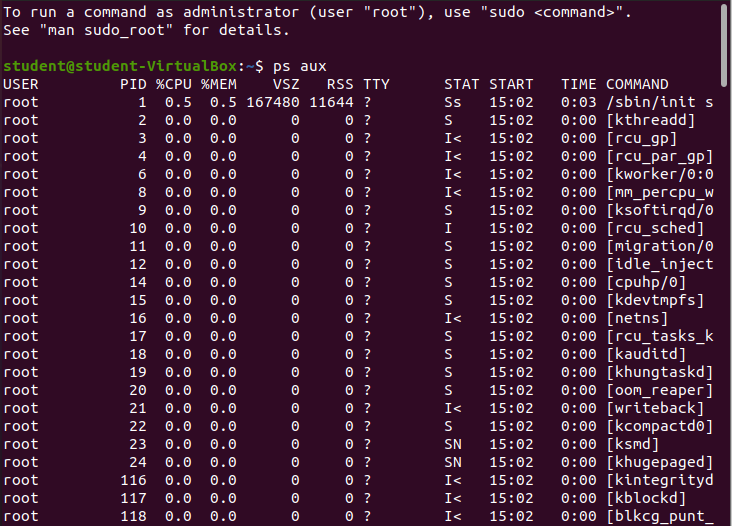


Рисунок 5.1 – Фрагмент вывода списка процессов системы

1. Вывести дерево процессов.

Для построения дерева процессов используются команды (результат работы каждой из них идентичен):

pstree, psejH, psaxjt

Результат выполнения первой из приведенных выше команд, представлен на рисунке 5.2.



Рисунок 5.2 – Фрагмент списка процессов системы …

1. С помощью команды top получить список 5 процессов, потребляющих наибольшее количество процессорного времени.

Для того чтобы вывести список процессов, отсортированных по потреблению наибольшего количество процессорного времени, необходимо воспользоваться командой: top -o %CPU.

Однако, данная команда выведет полный список всех процессов. Для того чтобы выбрать только первые 5 процессов, необходимо воспользоваться командами tail и head, которые отсекают от выходного результата команды строки с конца и с начала соответственно. В качестве параметра, определяющего количество отсекаемых строк, используется параметр n.

Таким образом, получаем команду:

top -o %CPU | tail -n +7 | head -n 6

Результат выполнения данной команды, представлен на рисунке 5.3.

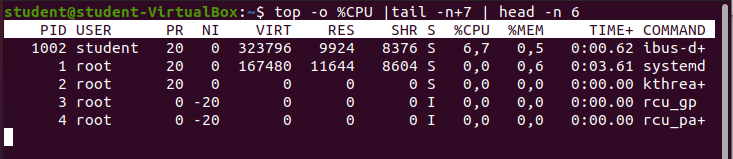


Рисунок 5.3 – Список 5 процессов, потребляющих наибольшее количество процессорного времени

1. Найти 2 процесса, имеющих более ДВУХ потоков.

Для того чтобы выбрать все процессы воспользуемся ключом «-e», который выводим список всех процессов. Получится команда:

ps –e

Для того чтобы получить необходимые нам колонки воспользуемся ключом «-o», параметры которого – необходимые для вывода колонки pid, cmd, nlwp, так как этот параметр позволяет пользователю использовать собственный формат ввода:

ps -e –o pid,cmd,nlwp

Отсортируем полученный результат по убыванию количества потоков

ps-e-o pid,cmd,nlwp --sort=-nlwp

Для того, чтобы получить только первые 2 процесса воспользуемся командой head –n 3. В конечном итоге получим команду:

ps -e -opid,cmd,nlwp --sort=-nlwp | head –n 3

Результат выполнения данной команды, представлен на рисунке 5.4.

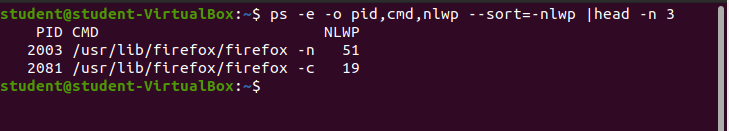


Рисунок 5.4 – Два процесса, имеющих более ДВУХ потоков

1. Используя команду top, изменить приоритеты 2 процессов. Изменим приоритеты двух процессов с PID 22452 и PID 2324.

Для просмотра информации об этих двух процессах воспользуемся командой top: top –p22452,2324

Результат выполнения данной команды, представлен на рисунке 5.5.

Как видим, значение приоритета обоих процессов равно 20. Повысим приоритет его у обоих процессов на 1. Это можно сделать интерактивно из утилиты top. Для этого запускаем утилиту top и нажимаем клавишу <R>. Нам будет предложено ввести уровень приоритета и PID процесса, аналогично вводу в режиме команд (рисунки 5.6 и 5.7).

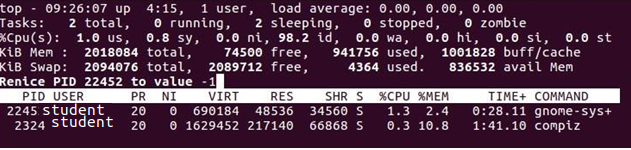


Рисунок 5.5– Информация о процессах

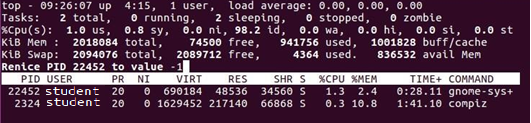


Рисунок 5.6 – Изменение приоритета процесса с PID 22452

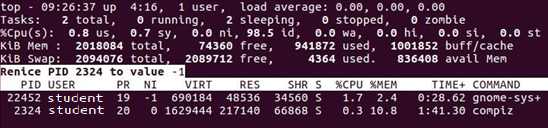


Рисунок 5.7– Изменение приоритета процесса с PID 2324

Теперь мы можем видеть, что приоритет обоих процессов изменен и равен значению 19 (рисунок 5.8.).

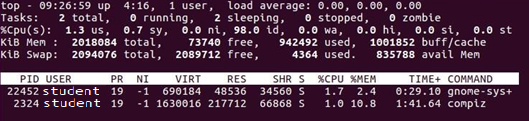


Рисунок 5.8 – Информация о процессах

1. Получить список открытых файлов текущего пользователя

Команда lsof (ListOpenFiles) без параметров выводит полный список открытых файлов. При этом пользователь-администратор получит несколько тысяч строк текста.

Для получения списка файлов, открытых конкретным пользователем, служит команда: lsof -u имя\_пользователя

Таким образом, в нашем случае (имя пользователя: student) необходимо выполнить команду:

lsof –u student

Результат выполнения данной команды, представлен на рисунке 5.9.

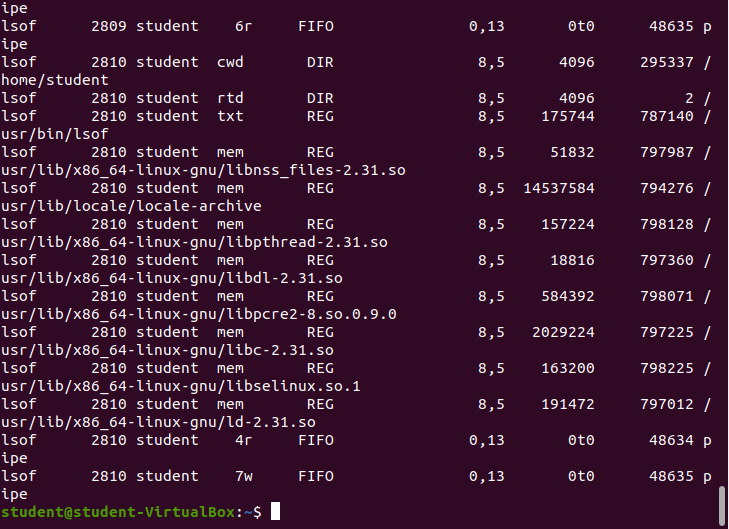


Рисунок 5.9– Фрагмент списка открытых файлов пользователя

1. Получить текущее состояние системной памяти

Текущее состояние системной памяти позволяет получить команда free.

По умолчанию все выводимые значения представлены в килобайтах. Значения в мегабайтах позволяет получить опция –m.

Таким образом, получаем команду: free –m

Результат выполнения данной команды, представлен на рисунке 5.10.

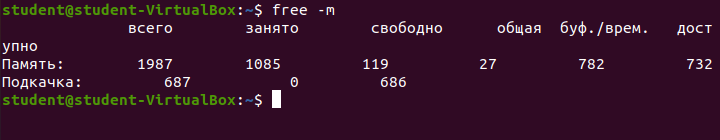


Рисунок 5.10– Текущее состояние системной памяти

1. Получить справку об использовании дискового пространства.

Команда df выводит данные об объеме доступного дискового пространства (в килобайтах). Опция -h улучшает восприятие результатов (данные выводятся в табличной форме).

Команда du дает возможность узнать объем дисковой памяти, занимаемой каталогами и файлами.

Таким образом, получаем команду: df –h

Результат выполнения данной команды, представлен на рисунке 5.11.

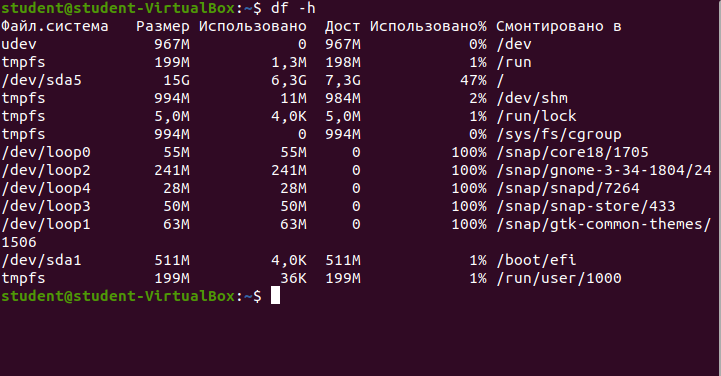


Рисунок 5.11– Справка об использовании дискового пространства

1. Вывести информацию о каком-либо процессе, используя содержимое каталога /proc

В каталоге /proc содержится информация о каждом процессе, приведем информацию об основных файлах в этом каталоге.

В качестве процесса возьмем процесс с PID 1.

cmdline: этот (псевдо-) файл содержит полную командную строку, использованную для вызова процесса.

Результат выполнения данной команды cat /proc/1/cmdline , представлен на рисунке 5.12.



Рисунок 5.12– Информация о процессе с PID 1, используя содержимое каталога /proc

cwd: эта символическая ссылка указывает на текущий рабочий каталог процесса (как следует из её имени).

Получим адрес ссылки при помощи команды: readlink -f/proc/1/cwd

Результат выполнения данной команды, представлен на рисунке 5.13.



Рисунок 5.13– Текущий рабочий каталог процесса с PID 1

environ: этот файл содержит все переменные окружения, определенные для этого процесса, в виде ПЕРЕМЕННАЯ = значение. Как и в cmdline вывод вообще не отформатирован: нет разделителей строк для отделения различных переменных, и в конце нет разделителя строки. cat/proc/1/environ

Результат выполнения данной команды, представлен на рисунке 5.14.

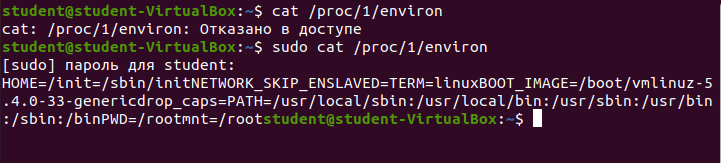


Рисунок 5.14 – Все переменные окружения, определенные для процесса с PID 1

exe: эта символическая ссылка указывает на исполняемый файл, соответствующий запущенному процессу.

Получим адрес ссылки при помощи команды: readlink -f /proc/1/exe

Результат выполнения данной команды, представлен на рисунке 5.15.



Рисунок 5.15 – Исполняемый файл, соответствующий запущенному процессу с PID 1

fd: этот подкаталог содержит список файловых дескрипторов, открытых в данный момент процессом.

Результат выполнения команды ls /proc/1/fd, представлен на рисунке 5.16.

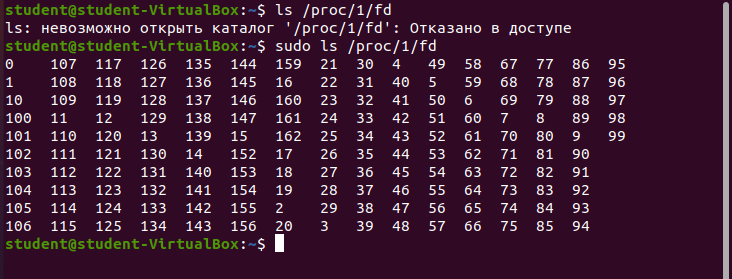


Рисунок 5.16– Список файловых дескрипторов, открытых в данный момент процессом с PID 1

maps: когда вы выводите содержимое этого именованного канала (при помощи команды cat, например), вы можете увидеть части адресного пространства процесса, которые в текущий момент распределены для файла. Вот эти поля (слева направо): адресное пространство, связанное с этим распределением; разрешения, связанные с этим распределением; смещение от начала файла, где начинается распределение; старший и младший номера (в шестнадцатеричном виде) устройства, на котором находится распределенный файл; номер inode файла; и, наконец, имя самого файла. Результат выполнения данной команды cat /proc/1/maps, представлен на рисунке 5.17.

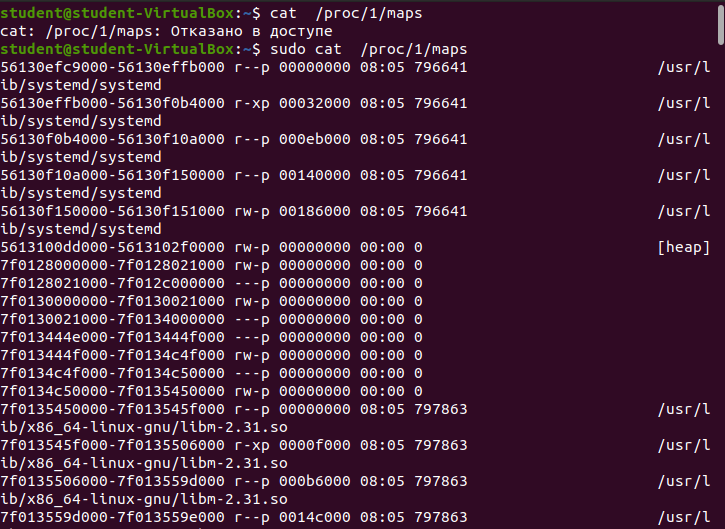


Рисунок 5.17– Фрагмент адресного пространства процесса с PID 1

root: эта символическая ссылка указывает на корневой каталог, используемый процессом. Обычно это будет readlink -f /proc/1/root

Результат выполнения данной команды, представлен на рисунке 5.18.

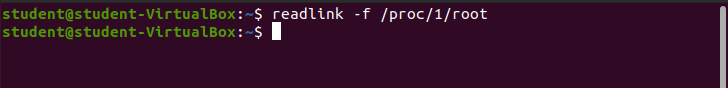


Рисунок 5.18– Корневой каталог, используемый процессом с PID 1

status: этот файл содержит разнообразную информацию о процессе: имя исполняемого файла, его текущее состояние, его PID и PPID, его реальные и эффективные UID и GID, его использование памяти и другие данные.

Результат выполнения команды cat /proc/1/status, представлен на рисунке 5.19.

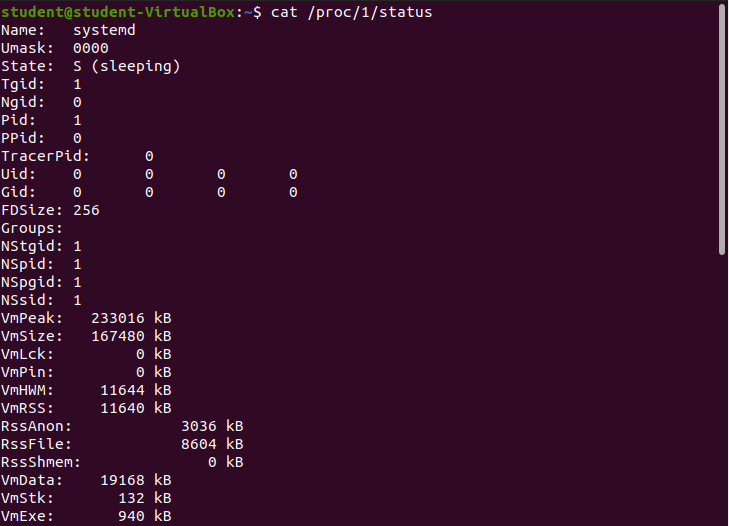


Рисунок 5.19–Фрагмент информации о процессе с PID 1

1. Вывести информацию о процессоре ПК, используя содержимое каталога /proc

/proc/cpuinfo: этот файл содержит, как видно из его имени, информацию о процессорах машины: cat /proc/cpuinfo

Результат выполнения данной команды, представлен на рисунке 5.20.

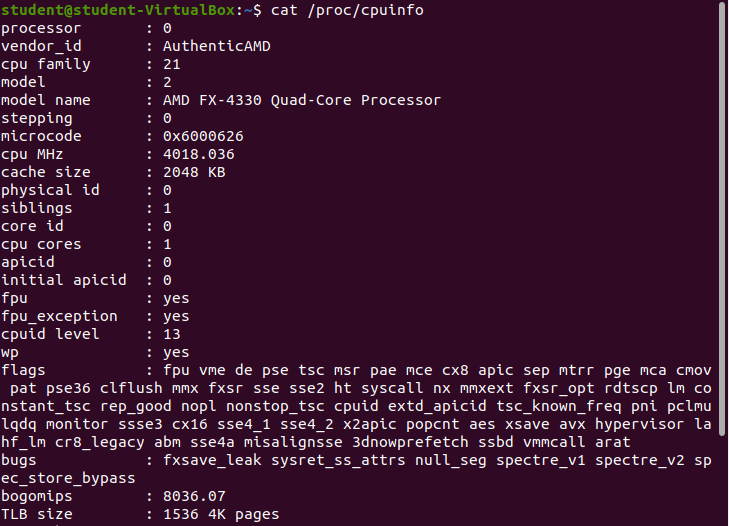


Рисунок 5.20– Фрагмент информации о процессоре ПК

Вывести список модулей, используемых в настоящий момент ядром /proc/modules: этот файл содержит список модулей, используемых ядром в настоящий момент, вместе со счетчиком использования каждого из модулей. Эта информация используется командой lsmod, которая отображает её в более удобной для чтения формe. Для вывода воспользуемся командой cat.

Результат выполнения команды cat/proc/modules, представлен на рисунке 5.21.

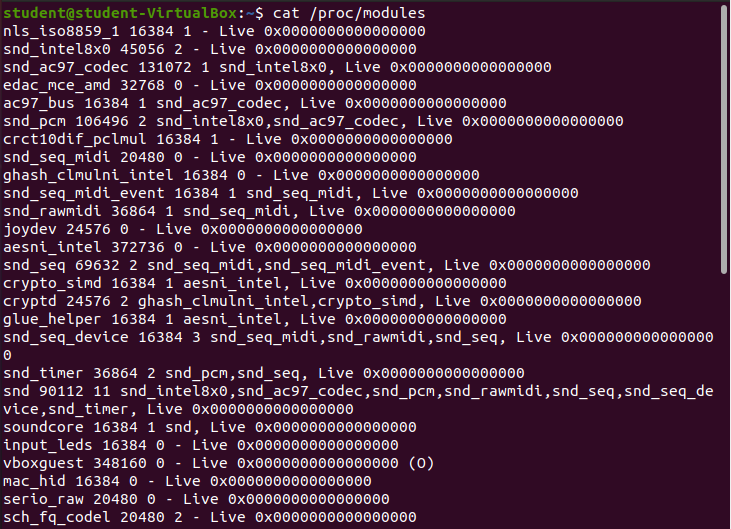


Рисунок 5.21 – Фрагмент списка модулей, используемых в настоящий момент ядром ОС

**Контрольные вопросы**

1. Назначение счетчиков производительности.

В ОС MS Windows имеется ряд объектов производительности, обычно соответствующих аппаратным компонентам, таким как память, процессоры, внешние устройства и т. д. Каждый объект производительности предоставляет счетчики, которые собирают данные производительности (performance counters).

Счетчик производительности представляет собой механизм, с помощью которого в MS Windows производится сбор сведений о производительности различных системных ресурсов. В MS Windows имеется предопределенный набор счетчиков производительности, с которыми можно взаимодействовать. Каждый счетчик относится к определенной области функций системы.

1. Категории и экземпляры счетчиков.

Счетчик производительности следит за поведением объектов производительности компьютера. Эти объекты включают в себя физические компоненты, такие как процессоры, диски, память и системные объекты, такие как процессы, потоки и задания.

Системные счетчики, относящиеся к одному и тому же объекту производительности, группируются в категории, отражающие их общую направленность. При создании экземпляра компонента PerformanceCounter сначала указывается категория, с которой будет взаимодействовать компонент, затем внутри этой категории выбирается счетчик, с которым будет осуществляться взаимодействие.

Некоторые объекты (такие как Память и Сервер) имеют только один экземпляр, другие объекты производительности могут иметь множество экземпляров. Если объект имеет множество экземпляров, то можно добавить счетчики для отслеживания статистики по каждому экземпляру или для всех экземпляров одновременно.

Например, если в системе установлены несколько процессоров, или процессор имеет несколько ядер, то объект Процессор будет иметь множество экземпляров. В случае, если объект поддерживает множество экземпляров, то при объединении экземпляров в группу появятся родительский экземпляр и дочерние экземпляры, которые будут принадлежать данному родительскому экземпляру.

1. Управление параметрами создаваемых графиков (масштаб, цвет и толщина линий).

Управление формой представления графиков производится с помощью окна свойств, которое открывается с помощью кнопки Свойства. Диапазон значений вертикальной шкалы задается в окне Свойства: системный монитор

1. Влияние активности окна приложения на текущий приоритет его потоков.

При перемещении окна приложения по экрану текущий приоритет изменяется только у потока с номером 0.