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.

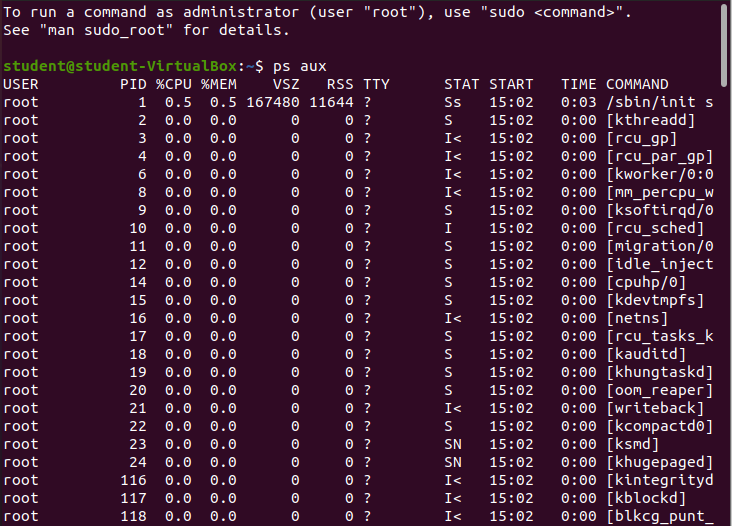


Рисунок 5.1 – Фрагмент вывода списка процессов системы

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 вывода списка процессов системы в виде дерева

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.

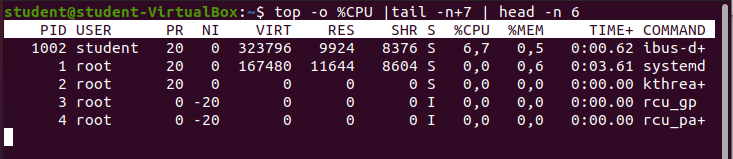


Рисунок 5.3 – Список 5 процессов, потребляющих наибольшее количество процессорного времени

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 – id процесса, cmd – команда запуска процессора, nlwp (Number Light-Weight Process) – число легковесных процессов в операционной системе, 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.

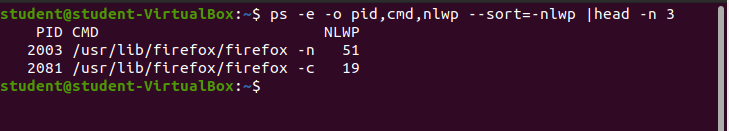


Рисунок 5.4 – Два процесса, имеющих более ДВУХ потоков

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).

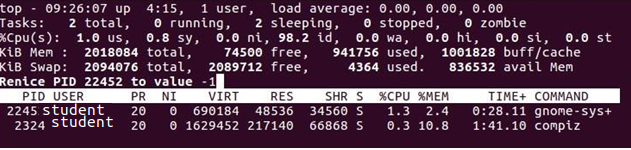


Рисунок 5.5– Информация о процессах

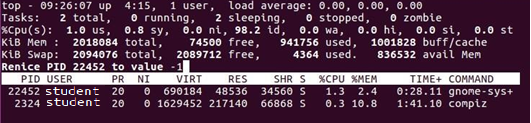


Рисунок 5.6 – Изменение приоритета процесса с PID 22452

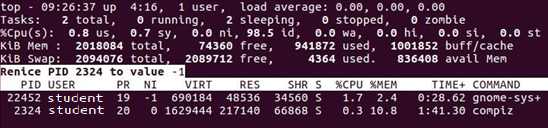


Рисунок 5.7– Изменение приоритета процесса с PID 2324

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

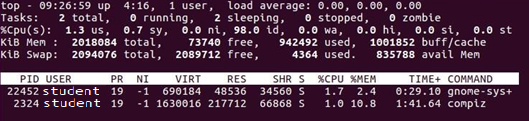


Рисунок 5.8 – Информация о процессах

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.

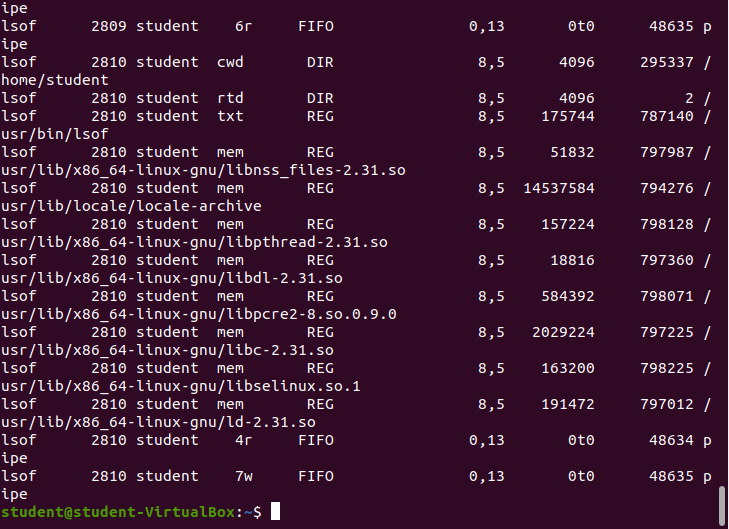


Рисунок 5.9– Фрагмент списка открытых файлов пользователя

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.

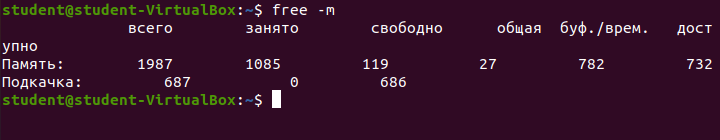


Рисунок 5.10– Текущее состояние системной памяти

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.

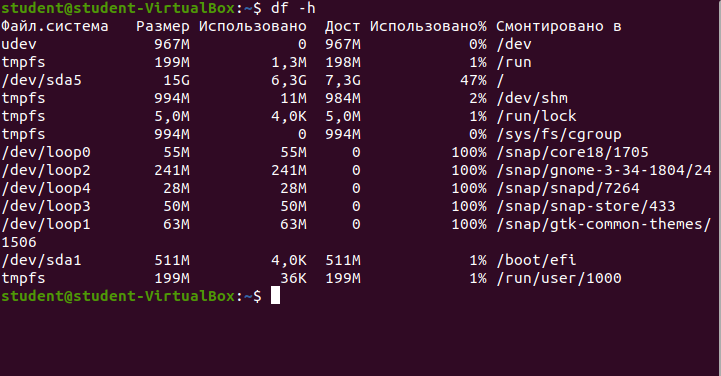


Рисунок 5.11– Справка об использовании дискового пространства

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.

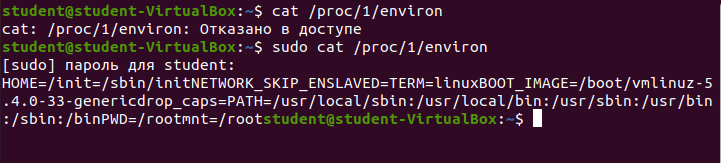


Рисунок 5.14 – Все переменные окружения, определенные для процесса с PID 1

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.

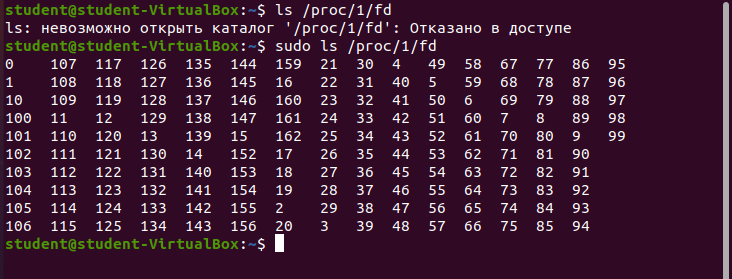


Рисунок 5.16– Список файловых дескрипторов, открытых в данный момент процессом с PID 1

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.

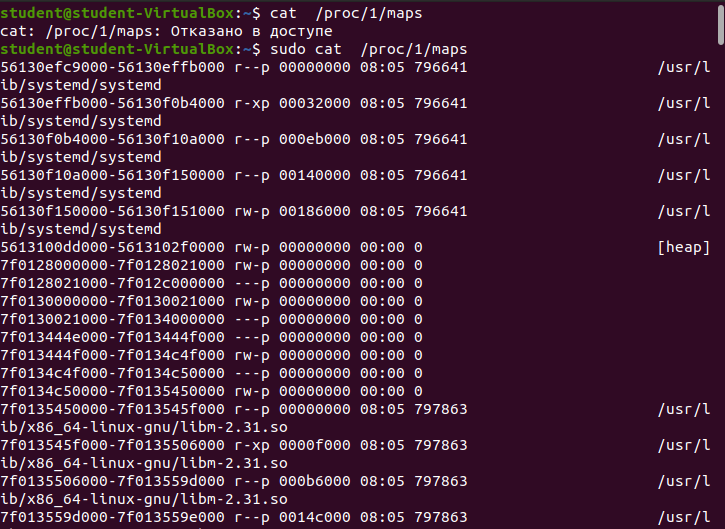


Рисунок 5.17– Фрагмент адресного пространства процесса с PID 1

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.

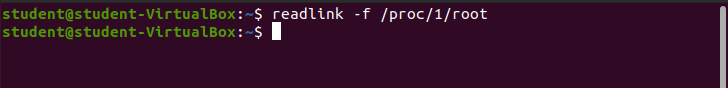


Рисунок 5.18– Корневой каталог, используемый процессом с PID 1

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.

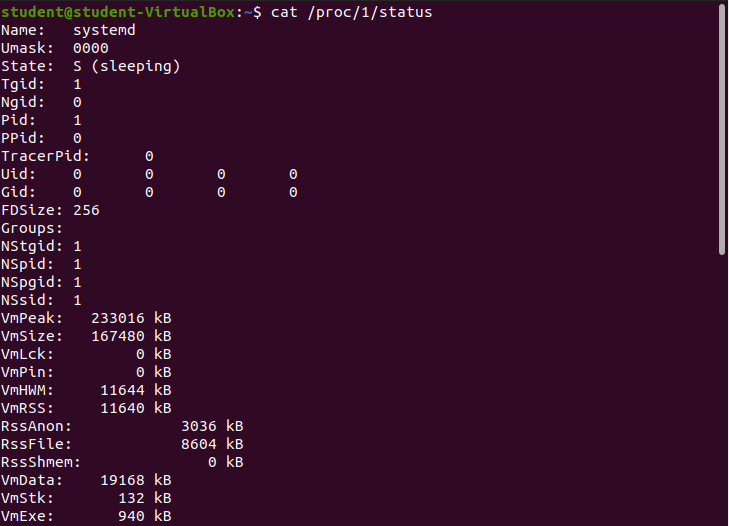


Рисунок 5.19–Фрагмент информации о процессе с PID 1

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.

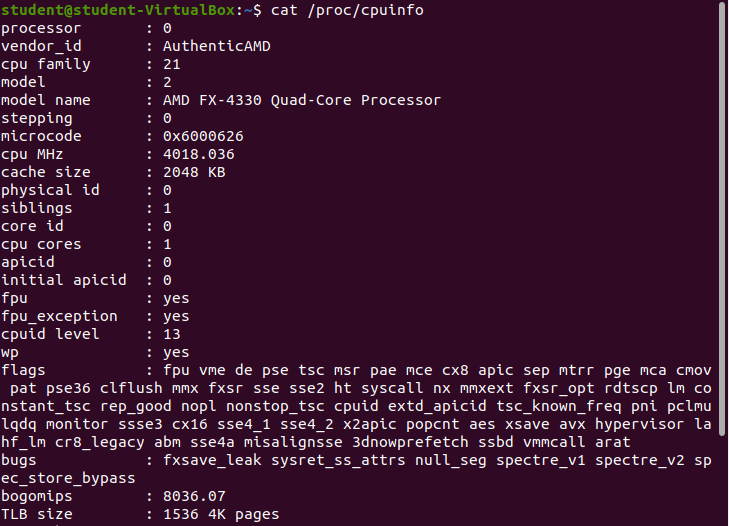


Рисунок 5.20– Фрагмент информации о процессоре ПК

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.

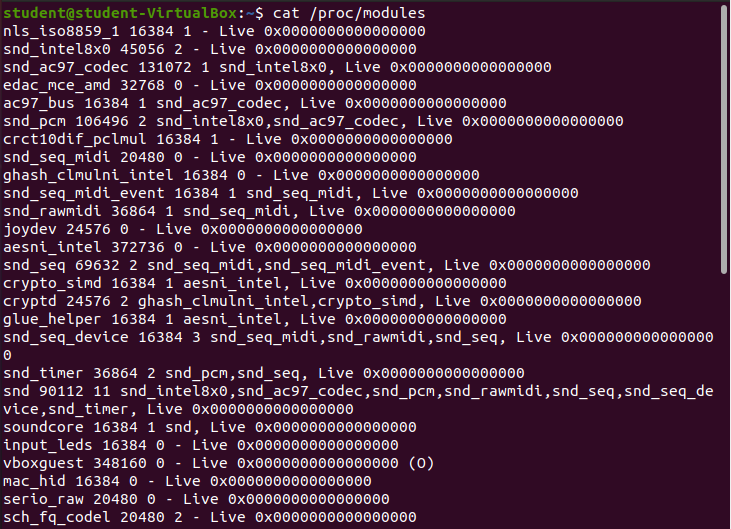


Рисунок 5.21 – Фрагмент списка модулей, используемых в настоящий момент ядром ОС

**Контрольные вопросы**

* 1. Команды вывода списка процессов.

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.

Полезные ключи: -e – вывод сведений обо всех процессах и o – пользовательский вывод.

* 1. Команда получения списка потоков.

Для получения информации о потоках заданного процесса используется опция –L, на пример ps –fLC swriter.bin выводит список потоков приложения writer Open Office.

* 1. Команда для завершения приложений.

Завершение процесса выполняется командой kill сигнал PID.

Сначала процессу посылается сигнал -15. Если это не помогает, 129, используется крайняя мера -посылается сигнал -9.

* 1. Состояния процесса Linux.

Каждый из каталогов содержит одинаковые пункты, краткое описание некоторых из них:

cmdline: этот (псевдо-) файл содержит полную командную строку, использованную для вызова процесса 134 программой и ее аргументами нет пробелов, а в конце строки нет разделителя строки. Чтобы просмотреть его, вы можете использовать: perl - ple 's,\00, ,g' cmdline.

cwd: эта символическая ссылка указывает на текущий рабочий каталог процесса (следует из имени).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. Единственным решением для его просмотра будет: perl -pl -e 's,\00,\n,g' environ exe: эта символическая ссылка указывает на исполняемый файл, соответствующий запущенному процессу.

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

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 распределеним; смещение от начала файла, где начинается распределение; старший и младший номера (в шестнадцатиричном виде) устройства, на котором находится распределенный файл; номер inode файла; и, наконец, имя самого файла.

root: эта символическая ссылка указывает на корневой каталог, используемый процессом. Обычно это будет /.

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.

1. Получение информации о потоках процесса.

Как известно, процесс может иметь параллельно выполняющиеся потоки (threads) или облегченные процессы (LWP, Light Weight Process). Для получения информации о потоках заданного процесса используется опция – L, например ps –fLC swriter.bin выводит список потоков приложения writer Open Office. Процессы, использующие более одного потока – редактор звуковых файлов audacity и soffice.bin, а также демоны (службы в по

терминологии Windows). Как указано выше, многопоточные процессы помечено символом l в колонке состояния.

1. Примеры многопоточных процессов.

Программные потоки являются базовым элементом многозадачного программного окружения. Программный поток может быть описан как среда выполнения процесса; поэтому каждый процесс имеет как минимум один программный поток. Многопоточность предполагает наличие нескольких параллельно работающих (на многопроцессорных системах) и обычно синхронизируемых сред выполнения процесса.

Программные потоки имеют свои идентификаторы (thread ID) и могут выполняться независимо друг от друга. Они делят между собой одно адресное пространство процесса и используют эту особенность в качестве преимущества, позволяющего не использовать каналы IPC (систем меж процессного взаимодействия - разделяемой памяти, каналов и других систем) для обмена данными. Программные потоки процесса могут взаимодействовать-например, независимые потоки могут получать/изменять значение глобальной переменной. Эта модель взаимодействия исключает лишние затраты ресурсов на вызовы IPC на уровне ядра. Поскольку потоки работают в едином адресном пространстве, переключения контекста для потока быстры и не ресурсоемки.

Многопоточные процессы помечено символом l в колонке состояния ps.

Примером может служить браузер Firefox.

1. Необходимость использования потоков.

Выделим основные преимущества использования потоков:

Потоки удобно использовать при необходимости выполнения в процессе нескольких действий сразу (пример: одновременная

обработка на сервере запросов нескольких пользователей)

Ускорение работы приложений, использующих ввод, обработку и вывод данных за счет возможности распределения этих операций по отдельным потокам. Это дает возможность не прекращать выполнение

программы во время возможных простоев из-за ожидания при чтении/записи данных

Как правило, переключение между потоками происходит быстрее и требует меньших затрат системных ресурсов, по сравнению с переключением между процессами

1. Процессы-зомби: как они появляются, как их найти и что с ними делать?

Потоки удобно использовать при необходимости выполнения в процессе нескольких действий сразу (пример: одновременная обработка на сервере запросов нескольких пользователей).

Как правило, переключение между потоками происходит быстрее и требует меньших затрат системных ресурсов, по сравнению с переключением между процессами.

1. Содержимое вывода команды top.

Команда top представляет динамически обновляемые сведения о процессах и о том, какой объем системных ресурсов использует каждый из них.

В первых пяти строках команда top отображает подробную информацию о системе, а затем описывает каждый выполняющийся процесс. Выходные данные сортируются по уровню загрузки ЦП данным процессом.

В первой строке программа сообщает текущее время, время работы системы (1 час 15 мин), количество зарегистрированных (login) пользователей (3 users), общая средняя загрузка системы (load average). Общей средней загрузкой системы называется среднее число процессов, находящихся в состоянии выполнения (R) или в состоянии ожидания (D). Общая средняя загрузка измеряется каждые 1, 5 и 15 минут.

Во второй строке вывода программы top сообщается, что в списке процессов находятся 132 процесса, из них 131 спит (состояние готовности или ожидания), 1 выполняется (на виртуальной машине только 1 процессор), 0 процессов зомби и 0 остановленных процессов.

В третьей-пятой строках приводится информация о загрузке процессора CPU в режиме пользователя и системном режиме, использования памяти и файла подкачки.

В таблице отображается различная информация о процессе. Рассмотрим колонки PID (идентификатор процесса), USER (пользователь, запустивший процесс), S (состояние процесса) и COMMAND (команда, которая была введена для запуска процесса).

Колонка S может содержать следующие значения:

R – процесс выполняется или готов к выполнению (состояние готовности); помечено символом l в колонке состояния ps. Примером может служить браузер Firefox.

1. Как получить информацию о процессах системы, используя файловую систему /proc?

Каталог /procсодержит всю информацию о всех процессах в системе, для доступа к информации о процессе используется следующая команда:

/proc/[pid]/stat

Где на месте stat может быть любой другой тип возвращаемой информации.

1. Команды для получения информации об открытых файлах.

Команда lsof (List open files) без параметров выводит полный список открытых файлов. Пользователь-администратор получит несколько тысяч строк текста.

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

1. Получение информации о состоянии системной памяти.

Текущее состояние системной памяти позволяет получить команда

По умолчанию все значения представлены в килобайтах. Значения в М

позволяет получить опция –m.

1. Получение информации об использовании дискового пространства.

Команда df выводит данные об объеме доступного дискового пространства (в Кбайтах). Опция –h улучшает восприятие результатов.

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

1. Назначение файловой системы /proc.

Файловая система /proc является механизмом для ядра и его модулей, позволяющим посылать информацию процессам (отсюда и название /proc). С помощью этой виртуальной файловой системы можно работать с внутренними структурами ядра, получать полезную информацию о процессах и изменять установки (меняя параметры ядра) на лету. Файловая система /proc располагается в памяти в отличие от других файловых систем, которые располагаются на диске.