**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №4

по курсу "Операционная система Linux"

Управление процессами ОС Ubuntu

Курбатова К.А.

|  |
| --- |
| Студент |
| Группа АИ-17-1 | Кургасов В.В. |  |
| Руководитель  Доцент |  |  |

Липецк 2019г.

Цель работы

Знакомство со средствами управления процессами ОС Ubuntu.

Задание

1. Запустить программу виртуализации Oracle VM VirtualBox.

2. Запустить виртуальную машину Uduntu.

1. Открыть окно интерпретатора команд
2. Вывести общую информацию о системе
   1. Вывести информацию о текущем интерпретаторе команд
   2. Вывести информацию о текущем пользователе
   3. Вывести информацию о текущем каталоге
   4. Вывести информацию об оперативной памяти и области подкачки
   5. Вывести информацию о дисковой памяти

5.Выполнить команды получения информации о процессах

* 1. Получить идентификатор текущего процесса(PID)
  2. Получить идентификатор родительского процесса(PPID)
  3. Получить идентификатор процесса инициализации системы
  4. Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд
  5. Отобразить все процессы

1. Выполнить команды управления процессами
   1. Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе
   2. Определить текущее значение nice по умолчанию
   3. Запустить интерпретатор bash с понижением приоритета nice –n 10 bash
   4. Определить PID запущенного интерпретатора
   5. Установить приоритет запущенного интерпретатора равным 5 renice –n 5 <PID процесса>
   6. Получить информацию о процессах bash

# ps lax | grep bash

Ход работы



Рисунок 1 – Информация о текущем интерпретаторе команд.



Рисунок 2 – Информация о текущем пользователе.



Рисунок 3 – Информация о текущем каталоге.

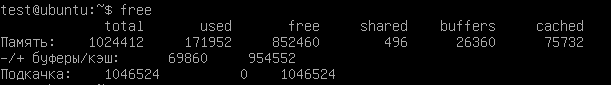


Рисунок 4 – Информация об оперативной памяти и области подкачки.

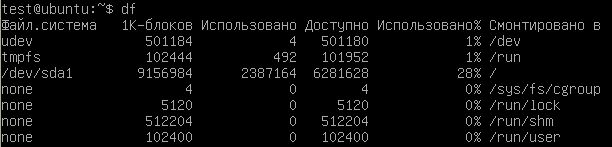


Рисунок 5 – Информация о дисковой памяти.



Рисунок 6 - Идентификатор текущего процесса (PID).



Рисунок 7 - Идентификатор родительского процесса (PPID).



Рисунок 8 - Идентификатор процесса инициализации системы.

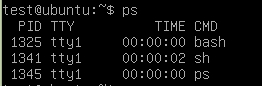


Рисунок 9 – Информация о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд.

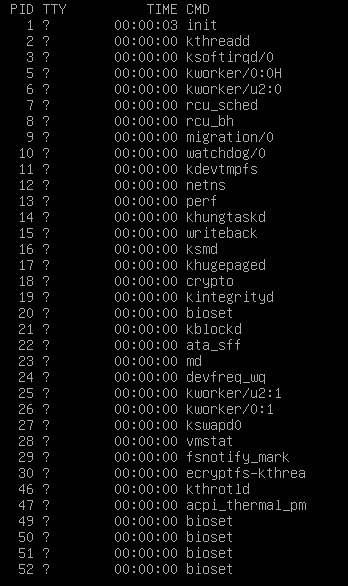


Рисунок 10 – Отображение всех процессов командой ps –e(часть 1).

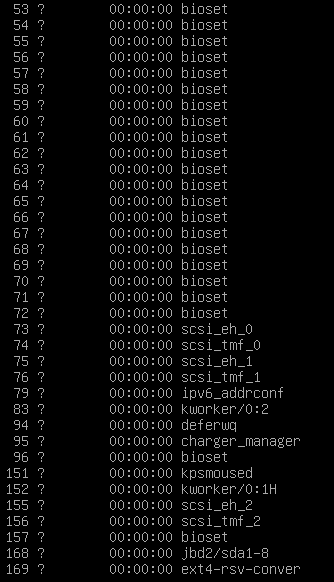


Рисунок 11 - Отображение всех процессов командой ps –e(часть 2).

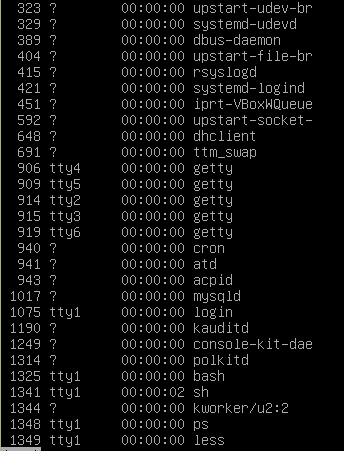


Рисунок 12 - Отображение всех процессов командой ps –e(часть 3).



Рисунок 13 - Текущее значение nice по умолчанию.

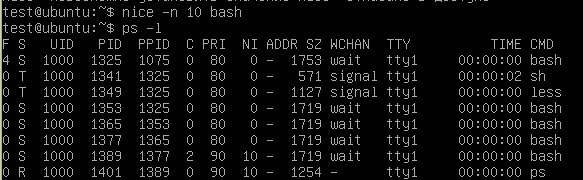


Рисунок 14 – Запущен интерпретатор bash с понижением приоритета.



Рисунок 15 - PID запущенного интерпретатора.



Рисунок 16 – Установка приоритета запущенного интерпретатора равным 5.

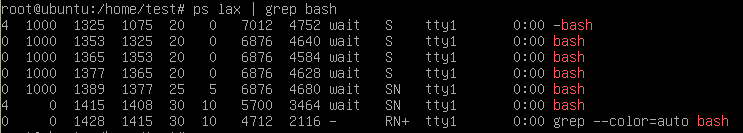


Рисунок 17 - Информация о процессах bash.

Контрольные вопросы

1. Перечислите состояния задачи в OC Ubuntu.

Состояния в ОС Ubuntu следующие:

* Zombie (зомби) - показывает, что выполнение задачи прекратилось, однако она еще не была удалена из системы.
* Dead (смерти) - может быть удалена из системы.
* Active (активный) и expired (неактивный) - используются при планировании выполнения процесса, и поэтому они не сохраняются в переменной state.

1. Как создаются задачи задачи в OC Ubuntu?

Задачи создаются путем вызова системной функции clone.

Любые обращения к fork или vfork преобразуются в системные вызовы clone во время компиляции.

Функция fork создает дочернюю задачу, виртуальная память для которой выделяется по принципу копирования при записи (copy-on-write).

Поскольку дочерний процесс не использует родительские страницы (если в начале работы им была сразу вызвана функция execve), данная операция абсолютно бесполезна и только приводит к увеличению накладных расходов. Поэтому в Linux поддерживается вызов функции vfork, позволяющей повысить быстродействие при вызове дочерними процессами процедуры execve.

1. Назовите классы потоков ОС Ubuntu.

* Потоки реального времени, обслуживаемые по алгоритму FIFO.
* Потоки реального времени, обслуживаемые в порядке циклической очереди.
* Потоки разделения времени.

1. Как используется приоритет планирования при запуске задачи

Потоки реального времени, обслуживаемые по алгоритму FIFO, имеют наивысшие приоритеты и не могут прерываться другими потоками, за исключением такого же потока реального времени FIFO, перешедшего в состояние готовности.

Потоки реального времени, обслуживаемые в порядке циклической очереди, представляют собой то же самое, что и потоки реального времени FIFO, но с тем отличием, что они могут прерываться таймером.

Эти классы имеют более высокий приоритет, чем у потоков стандартного класса разделения времени.

1. Как можно изменить приоритет для выполняющейся задачи?

У каждого потока есть приоритет планирования. Значение по умолчанию равно 20, но оно может быть изменено при помощи системного вызова nice(value), вычитающего значение value из 20. Поскольку value должно находиться в диапазоне от -20 до +19, приоритеты всегда попадают в промежуток от 1 до 40.

Вывод

Я ознакомилась на практике со средствами управления процессами ОС Ubuntu.