**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа № 4

По дисциплине «OS Linux»

Процессы в операционной системе Linux

Студент Бахмутский М.В.

Группа АС-18

Руководитель Кургасов В.В.

Липецк 2020 г.

Цель работы

Целью работы является знакомство со средствами управления процессами ОС Ubuntu.

Ход работы

1 Задание 1

Вывести информацию о текущем интерпретаторе команд

Для вывода информации о текущем интерпретаторе команд необходимо воспользоваться командой echo $SHELL. Результат выполнения данной команды представлен на рисунке 1.



Рисунок 1 – Информация о текущем интерпретаторе команд

2) Вывести информацию о текущем пользователе

Для вывода информации о текущем пользователе необходимо воспользоваться командой whoami. Результат выполнения данной команды представлен на рисунке 2.



Рисунок 2 – Информация о текущем пользователе

3) Вывести информацию о текущем каталоге

Для вывода информации о текущем каталоге необходимо воспользоваться командой pwd. Результат выполнения данной команды представлен на рисунке 3.



Рисунок 3 – Информация о текущем каталоге

4) Вывести информацию об оперативной памяти и области подкачки

Для вывода информации об оперативной памяти и области подкачки необходимо воспользоваться командой free. Результат выполнения данной команды представлен на рисунке 4.

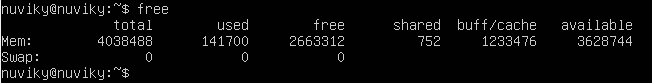


Рисунок 4 – Информация об оперативной памяти и области подкачки

5) Вывести информацию о дисковой памяти

Для вывода информации о дисковой памяти необходимо воспользоваться командой df. Результат выполнения данной команды представлен на рисунке 5.

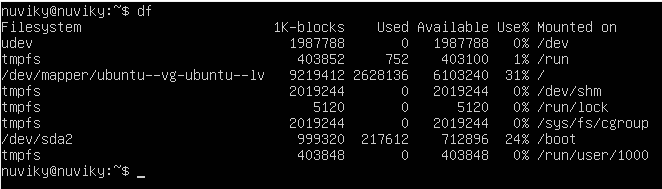


Рисунок 5 – Информация о дисковой памяти

2 Задание 2

1) Получить идентификатор текущего процесса (PID)

Для получения идентификатора текущего процесса (PID) необходимо воспользоваться командой echo $$. Результат выполнения данной команды представлен на рисунке 6.



Рисунок 6 – Идентификатор текущего процесса (PID)

2) Получить идентификатор родительского процесса (PPID)

Для получения идентификатора родительского процесса (PPID) необходимо воспользоваться командой echo $PPID. Результат выполнения данной команды представлен на рисунке 7.



Рисунок 7 – Идентификатор родительского процесса (PPID)

3) Получить идентификатор процесса инициализации системы

Для получения идентификатора процесса инициализации системы необходимо воспользоваться командой echo pidof init. Результат выполнения данной команды представлен на рисунке 8.



Рисунок 8 – Идентификатор процесса инициализации системы

4) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

Для получения информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд необходимо воспользоваться командой ps. Результат выполнения данной команды представлен на рисунке 9.

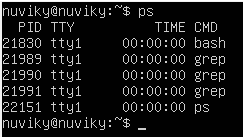


Рисунок 9 – Информация о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд

5) Отобразить все процессы

Для отображения всех процессов необходимо воспользоваться командой ps -e. Результат выполнения данной команды представлен на рисунке 10.

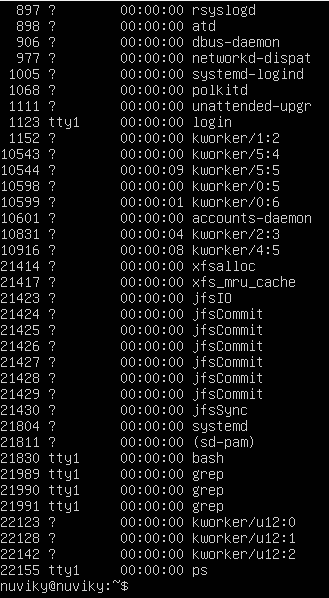


Рисунок 10 – Все процессы

3 Задание 3

1) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе

Для получения информации о выполняющихся процессах текущего пользователя в текущем интерпретаторе необходимо воспользоваться командой ps. Результат выполнения данной команды представлен на рисунке 11.

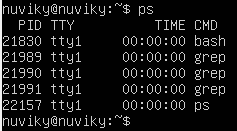


Рисунок 11 – Информация о выполняющихся процессах текущего пользователя в текущем интерпретаторе

2) Определить текущее значение nice по умолчанию

Для определения текущего значения nice по умолчанию необходимо воспользоваться командой nice. Результат выполнения данной команды представлен на рисунке 12.



Рисунок 12 – Текущее значение nice по умолчанию

3) Запустить интерпретатор bash с понижением приоритета

Для запуска интерпретатора bash с понижением приоритета необходимо воспользоваться командой nice -n 10 bash. Результат выполнения данной команды представлен на рисунке 13.

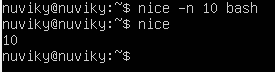


Рисунок 13 – Интерпретатор bash с понижением приоритета

4) Определить PID запущенного интерпретатора

Для определения PID запущенного интерпретатора необходимо воспользоваться командой ps -f. Результат выполнения данной команды представлен на рисунке 14.

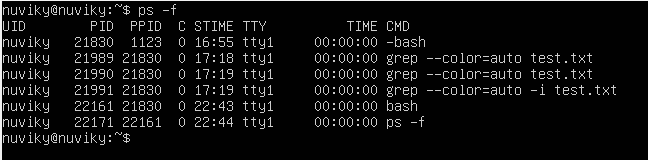


Рисунок 14 – PID запущенного интерпретатора

5) Установить приоритет запущенного интерпретатора равным 5

Для установки приоритета запущенного интерпретатора равным 5 необходимо воспользоваться командой renice –n 5 <PID процесса>. Результат выполнения данной команды представлен на рисунке 15.



Рисунок 15 – Установка приоритета запущенного интерпретатора

6) Получить информацию о процессах bash

Для получения информации о процессах bash необходимо воспользоваться командой ps lax | grep bash. Результат выполнения данной команды представлен на рисунке 16.

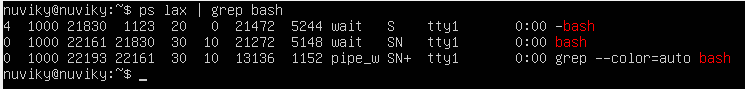


Рисунок 16 – Информация о процессах bash

Вывод

В ходе выполнения лабораторной работы я ознакомился на практике с понятием процесса в операционной системе, приобрел опыт и навыки управления процессами в операционной системе Linux.

Контрольные вопросы

1. Перечислите состояния задачи в OC Ubuntu

В ОС Ubuntu задачи могут находиться в следующих состояниях:

– Активен (R=Running) – процесс находится в очереди на выполнение, то есть либо выполняется в данный момент, либо ожидает выделения ему очередного кванта времени центрального процессора.

– «Спит» (S=Sleeping) – процесс находится в состоянии прерываемого ожидания, то есть ожидает какого-то события, сигнала или освобождения нужного ресурса.

– Находится в состоянии непрерываемого ожидания (D=Direct) – процесс ожидает определенного («прямого») сигнала от аппаратной части и не реагирует на другие сигналы;

– Приостановлен (T) – процесс находится в режиме трассировки (обычно такое состояние возникает при отладке программ).

«Зомби» (Z=Zombie) – это процесс, выполнение которого завершилось, но относящиеся к нему структуры ядра по каким-то причинам не освобождены.

2. Как создаются 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).

Когда дочерний или же родительский процесс пытается выполнить запись в страницу памяти, записывающая программа создает собственную копию страницы в памяти.

Процедура vfork приостанавливает работу родительского процесса в том случае, когда дочерний процесс вызывает функции execve или exit, чтобы обеспечить загрузку дочерним процессом новых страниц до того, как родительский процесс начнет выполнять бесполезные операции копирования при записи.

3. Назовите классы потоков ОС Ubuntu

1. Потоки реального времени, обслуживаемые по алгоритму FIFO.

2. Потоки реального времени, обслуживаемые в порядке циклической очереди.

3. Потоки разделения времени

4. Как используется приоритет планирования при запуске 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.

5. Как можно изменить приоритет для выполняющейся задачи?

Понижение приоритета выполняемого процесса:

renice –n [значение nice] –p [PID процесса]