Липецкий государственный технический университет
Факультет автоматизации и информатики
Кафедра автоматизированных систем управления

Лабораторная работа №4

по Дисциплине «Операционная система Linux»

Управление процессами в ОС Ubuntu

Студент Глебов Д.А.

Группа АИ-18

Руководитель Кургасов В.В.

к.п.н.

Липецк 2020 г.

Оглавление

Цель работы

Познакомиться со средствами управления процессами ОС Ubuntu.

Задание кафедры

1. Запустить программу виртуализации Oracle VM VirtualBox.

2. Запустить виртуальную машину Ubuntu.

3. Открыть окно интерпретатора команд.

4. Вывести общую информацию о системе.

4.1. Вывести информацию о текущем интерпретаторе команд.

4.2. Вывести информацию о текущем пользователе.

4.3. Вывести информацию о текущем каталоге.

4.4. Вывести информацию об оперативной памяти и области подкачки.

4.5. Вывести информацию о дисковой памяти.

5. Выполнить команды получения информации о процессах.

5.1. Получить идентификатор текущего процесса (PID).

5.2. Получить идентификатор родительского процесса (PPID).

5.3. Получить идентификатор процесса инициализации системы.

5.4. Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд.

5.5. Отобразить все процессы.

6. Выполнить команды управления процессами.

6.1. Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе.

6.2. Определить текущее значение nice по умолчанию.

6.3. Запустить интерпретатор bash с понижением приоритета.

6.4. Определить PID запущенного интерпретатора.

6.5. Установить приоритет запущенного интерпретатора равным 5.

6.6. Получить информацию о процессах bash.

Ход работы

Выведем информацию о текщем интерпритаторе команд, с помощью команды *echo $SHELL*:

```
g4zele@rcrmdpo:~$ echo $SHELL
/bin/bash
```

Рисунок 1 – Информация о текущем интерпретаторе команд

Просмотрим информацию о текущем пользователе командой *whoami*

```
g4zele@rcrmdpo:~$ whoami
g4zele
```

dfРисунок 2 – Информация о текущем пользователе

Выведем информацию о текущем каталоге с командой *pwd*:

```
g4zele@rcrmdpo:~$ pwd
/home/g4zele
```

Рисунок 3 – Информация о текущем пользователе

Теперь выведем информацию об оперативной памяти и области подкачки. (Команда *free*):

```
g4zele@rcrmdpo:~$ free
               total        used        free      shared  buff/cache   available
Mem:         2041492     1115260      200528       33836      725704      709212
Swap:        2097148       32452     2064696
```

Рисунок 4 – Информация об оперативной памяти и области подкачки

Проверим информацию о дисковой памяти командой *df*

```
g4zele@rcrmdpo:~$ df
Filesystem                         1K-blocks    Used Available Use% Mounted on
udev                                  987144       0    987144   0% /dev
tmpfs                                 204152    1688    202464   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv   20511312 8048844  11397508  42% /
tmpfs                                1020744       0   1020744   0% /dev/shm
tmpfs                                   5120       4      5116   1% /run/lock
tmpfs                                1020744       0   1020744   0% /sys/fs/cgroup
/dev/loop0                            166784  166784         0 100% /snap/gnome-3-28-1804/145
/dev/loop1                             56704   56704         0 100% /snap/core18/1932
/dev/loop2                             64384   64384         0 100% /snap/zenkit/12
/dev/loop4                            128896  128896         0 100% /snap/docker/471
/dev/loop5                            100096  100096         0 100% /snap/core/10185
/dev/loop6                             61056   61056         0 100% /snap/powershell/137
/dev/loop7                             66432   66432         0 100% /snap/gtk-common-themes/1514
/dev/loop8                             62592   62592         0 100% /snap/zenkit/10
/dev/loop9                             66944   66944         0 100% /snap/powershell/149
/dev/loop3                             63616   63616         0 100% /snap/gtk-common-themes/1506
/dev/sda2                             999320  156852    773656  17% /boot
tmpfs                                 204148      28    204120   1% /run/user/124
tmpfs                                 204148      32    204116   1% /run/user/1000
```

Рисунок 5 – Информация о дисковой памяти

Далее по заданию кафедры следуют задачи по выводу информации о процессах.

Получим идентификатор текущего процесса. (Команда *echo $$*):

```
g4zele@rcrmdpo:~$ echo $$
2908
```

Рисунок 6 – Идентификатор текущего процесса

Теперь получим идентификатор родителького процесса с помощью команды echo $PPID:

```
g4zele@rcrmdpo:~$ echo $PPID
2898
```

Рисунок 7 – Идентификатор родительского процесса

Для получения идентификатора системы мы будем использовать *pidof*:

```
g4zele@rcrmdpo:~$ pidof bash
2908
```

Рисунок 8 – Идентификатор процесса инициализации системы

Выведем информацию о выполняющихся процессах пользователя в текущем интерпретаторе команд командой *ps* ,без каких-либо допполнительных аттрибутов:

```
g4zele@rcrmdpo:~$ ps
  PID TTY          TIME CMD
 2908 pts/0    00:00:00 bash
 2947 pts/0    00:00:00 ps
```

Рисунок 9 – Процессы текущего пользователя в текущем интерпретаторе

Для того чтобы вывести все процессы используем команду *ps*, но уже с аттрибутом *-e*:

```
g4zele@rcrmdpo:~$ ps -e
  PID TTY          TIME CMD
    1 ?        00:00:02 systemd
    2 ?        00:00:00 kthreadd
    4 ?        00:00:00 kworker/0:0H
    5 ?        00:00:00 kworker/u2:0
    6 ?        00:00:00 mm_percpu_wq
    7 ?        00:00:00 ksoftirqd/0
    8 ?        00:00:00 rcu_sched
    9 ?        00:00:00 rcu_bh
   10 ?        00:00:00 migration/0
   11 ?        00:00:00 watchdog/0
   12 ?        00:00:00 cpuhp/0
   13 ?        00:00:00 kdevtmpfs
```

Рисунок 10 – Все процессы

Теперь рассмотрим команды управления процессами.

Выведем информацию о выполняющихся процессах пользователя в текущем интерпретаторе (*ps*), после чего просмотрим текущее значение nice. Далее запустим новый экземпляр интерпритатора bash с понижением проиритета (*nice -n 10 bash*) и узнаем его идентификатор (*echo $$*). После чего установим приоритет равным 5и и получим информацию о процессах bash (*ps lax | grep bash*):

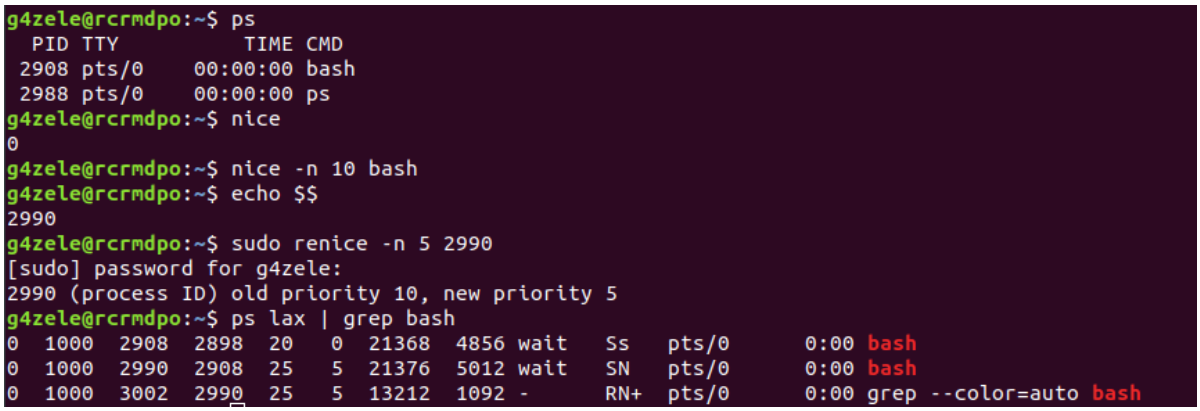

Рисунок 11 – Операции с созданным интерпретатором

Вывод

Мы познакомились со средствами управления процессами в ОС Ubuntu и изучили команды способные помочь в работе с процессами.

Ответы на контрольные вопросы

1. Перечислите состояния задачи в ОС Ubuntu.

1) Running (выполнение) – после выделения ей процесса;

2) Sleeping (спячки) – состояние блокировки;

3) Stopped (останов) – остановка работы;

4) Zombie (зомби) – выполнение задачи прекратилась, однако она не была удалена;

5) Dead (смерть) – задача может быть удалена из системы;

6) Active (активный) и inspired (неактивный) – используются при планировании выполнения процесса.

2. Как создаются задачи в ОС Ubuntu?

В системе Linux и процессы, и потоки называют задачами (task), и изнутри они представляют собой единую структуру данных.

Планировщик процессов хранит список всех задач в виде двух структур данных.

Первая структура это кольцевой список, каждая запись которого содержит указатели на предыдущую и последующую задачу. Обращение к данной структуре происходит в том случае, когда ядру необходимо проанализировать все задачи, которые должны быть выполнены в системе.

Второй структурой является хэш-таблица. При создании задачи ей присваивается уникальный идентификатор процесса (process identifier, PID). Идентификаторы процессов передаются хэш-функции для определения местоположения процесса в таблице процессов. Хэш-метод обеспечивает быстрый доступ к специфическим структурам данных задачи, если ядру известен ее PID.

Каждая задача таблицы процессов представляется в виде структуры task_struct, служащей в роли дескриптора процесса (т.е. блока управления процессором (PCB). В структуре task_struct хранятся переменные и вложенные структуры, описывающие процесс.

3. Назовите классы потоков в 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?

Планировщик использует приоритет и квант следующим образом. Сначала он вычисляет называемую в системе Linux «добродетелью» (goodness) величину каждого готового процесса по следующему алгоритму:

if (class == real_time) goodness = 1000 + priority;

if (class == timesharing & quantum > 0) goodness = quantum + priority;

if (class == timesharing && quantum == 0) goodness = 0;

В остальном алгоритм планирования очень прост: когда нужно принять решение, выбирается поток с максимальным значением «добродетели». Во время работы процесса его квант (переменная quantum) уменьшается на единицу на каждом тике. Центральный процессор отнимается у потока при выполнении одного из следующих условий:

1. Квант потока уменьшился до 0.

2. Поток блокируется на операции ввода-вывода и т.д.

3. В состояние готовности перешел ранее заблокированный поток с более высокой «добродетелью».

5. Как можно изменить приоритет для выполняющейся задачи? С этим может помочь команда renice –n

С этим может помочь команда *renice –n <новое значение приоритета> <PID процесса>*