Управление процессами

Майоров Дмитрий Андреевич

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получить навыки управления процессами операционной системы

2 Выполнение лабораторной работы

Получаем полномочия администратора. Вводим нужные команды. Последнюю команду мы запустили без &, поэтому мы пока что не имеем контроль над оболочкой. Нажимаем ctrl+z, чтобы остановить процесс

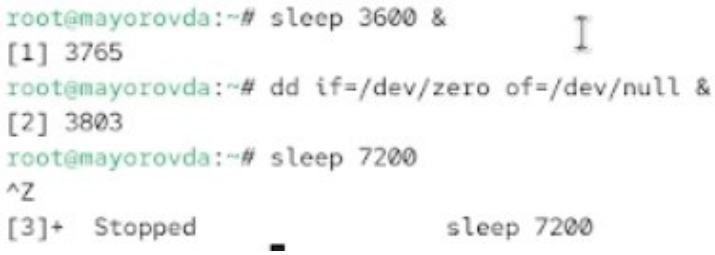

Рисунок 2.1

Вводим команду jobs. Видим три задания, которые мы запустили

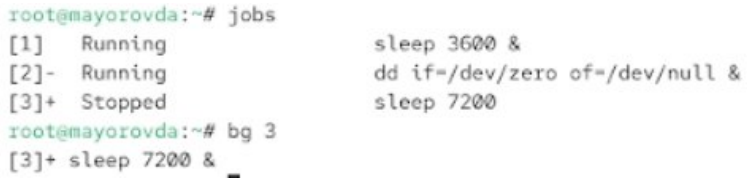

Рисунок 2.2

Перемещаем первое задание на передний план. Отменяем первое задание с помощью ctrl+c. Проводим аналогичные дейтсвия для заданий 2 и 3

```
root@mayorovda:~# fg 1
sleep 3600
^C
root@mayorovda:~# jobs
[2]-  Running                 dd if=/dev/zero of=/dev/null &
[3]+  Running                 sleep 7200 &
root@mayorovda:~# fg 2
dd if=/dev/zero of=/dev/null
^C257344997+0 records in
257344997+0 records out
131760638464 bytes (132 GB, 123 GiB) copied, 79.2569 s, 1.7 GB/s

root@mayorovda:~# fg 3
sleep 7200
^C
```

Рисунок 2.3

В новом терминале делаем новое задание. Закрываем терминал с помощью exit.

```
mayorovda@mayorovda:~$ dd if=/dev/zero of=/dev/null &
[1] 4068
mayorovda@mayorovda:~$ exit
```

Рисунок 2.4

В другом терминале запускаем top. Используем клавишу K, чтобы убить задание dd. Выходим из top

```
root@mayorovda:~# top

top - 14:35:41 up 3 min,  4 users,  load average: 1.52, 0.69, 0.27
Tasks: 257 total,   2 running, 254 sleeping,   0 stopped,   1 zombie
%Cpu(s): 18.1 us, 37.4 sy,  0.3 ni, 43.5 id,  0.0 wa,  0.6 hi,  0.0 si,
MiB Mem :   1961.2 total,     92.0 free,   1436.0 used,    633.7 buff/cac
MiB Swap:   2092.0 total,   1770.0 free,    322.0 used.    525.2 avail Me
```

Рисунок 2.5

Запускаем три новые программы

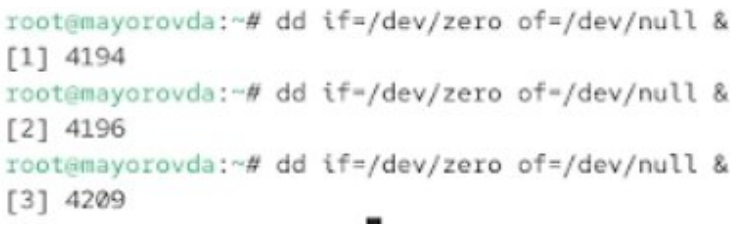

Рисунок 2.6

Вводим команду ps aux | grep dd. Она показывает все строки, в которых есть буквы dd

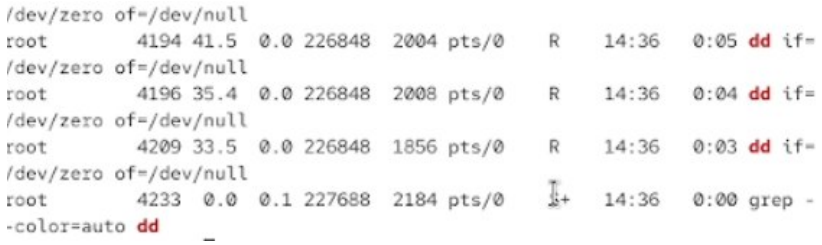

Рисунок 2.7

Используем PID первого процесса, чтобы изменить его приоритет

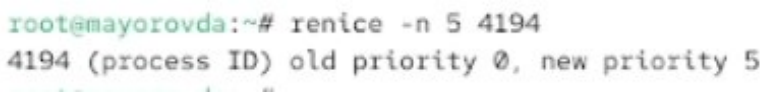

Рисунок 2.8

Вводим ps fax | grep -B5 dd. Параметр -B5 показывает соответствующие запросу строки, включая пять строк до этого

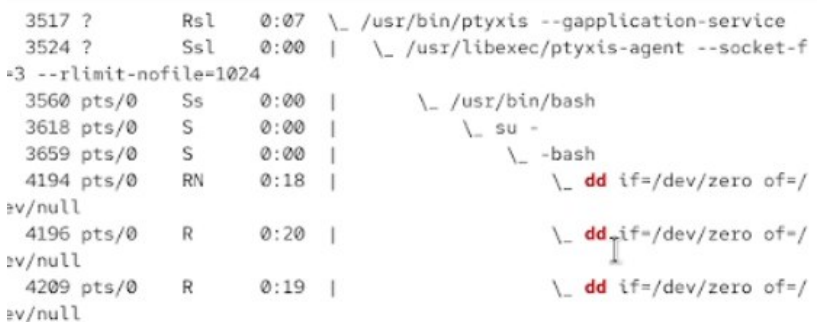

Рисунок 2.9

Находим PID корневой оболочки, из которой были запущены процессы dd, и вводим kill -9 3517. Корневая оболочка закрылась, а вместе с ней и все процессы dd

Рисунок 2.10

3 Самостоятельная работа

3.1 Задание 1

Запускаем команду dd if=/dev/zero of=/dev/nu трижды как фоновое значение

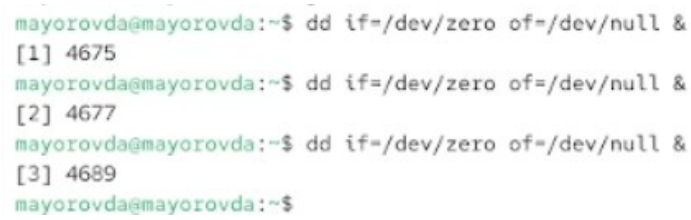

Рисунок 3.1

Изменяем приоритет первого процесса на -5, а потом изменяем его же на -15. -15 - это более высокий приоритет

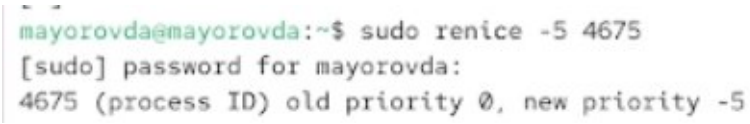

Рисунок 3.2

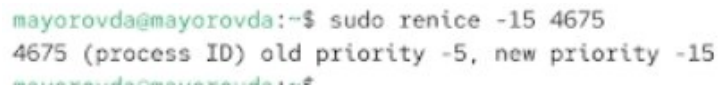

Рисунок 3.3

Завершаем все процессы

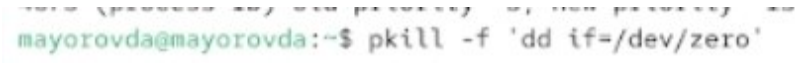

Рисунок 3.4

3.2 Задание 2

Запускаем программу yes в фоновом режиме с подавлением потока вывода

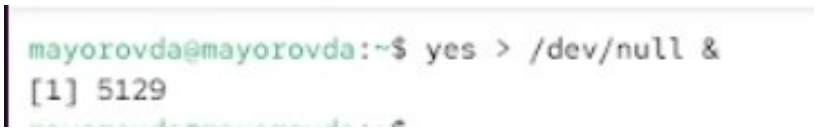

Рисунок 3.5

Запускаем программу yes на переднем плане с подавлением потока вывода. Приостановливаем выполнение программы. Заново запускаем программу yes с теми же параметрами, затем завершаем её выполнение

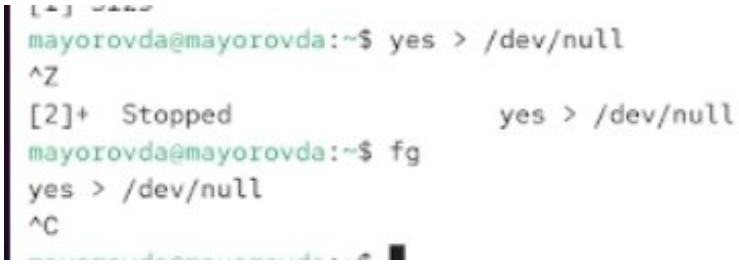

Рисунок 3.6

Запускаем программу yes на переднем плане без подавления потока вывода. Приостановливаем выполнение программы. Заново запускаем программу yes с теми же параметрами, затем завершаем её выполнение

```
y
y
y
y^Z
[2]+  Stopped                 yes
mayorovda@mayorovda:~$
```

Рисунок 3.7

```
y
y
y
^C
mayorovda@mayorovda:~$
```

Рисунок 3.8

Проверяем состояния заданий, воспользовавшись командой jobs. Переводим процесс, который у нас выполняется в фоновом режиме, на передний план, затем остановливаем его

```
mayorovda@mayorovda:~$ jobs
[1]+  Running                 yes > /dev/null &
mayorovda@mayorovda:~$ fg 1
yes > /dev/null
^Z
[1]+  Stopped                 yes > /dev/null
mayorovda@mayorovda:~$
```

Рисунок 3.9

Запускаем ещё три программы yes в фоновом режиме с подавлением потока вывода

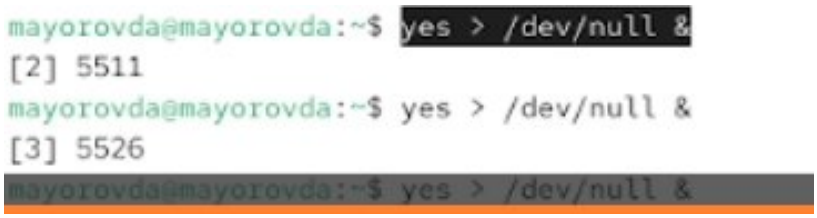

Рисунок 3.10

Убиваем два процесса: для одного используем его PID, а для другого — его идентификатор конкретного задания

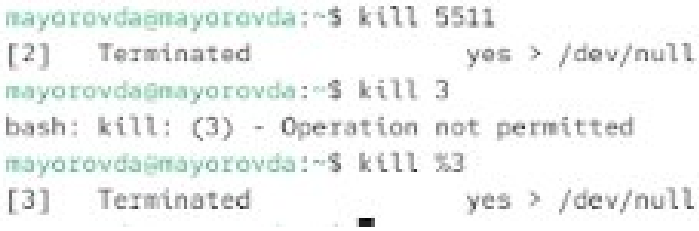

Рисунок 3.11

Запускаем ещё несколько программ yes в фоновом режиме с подавлением потока вывода

Рисунок 3.12

Завершаем их работу одновременно, используя команду killall

Рисунок 3.13

Запускаем программу yes в фоновом режиме с подавлением потока вывода. Используя утилиту nice, запускаем программу yes с теми же параметрами и с приоритетом, большим на 5

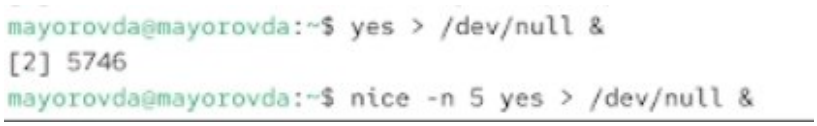

Рисунок 3.14

Используя утилиту renice, изменяем приоритет у одного из потоков yes таким образом, чтобы у обоих потоков приоритеты были равны

Рисунок 3.15

4 Выводы

Получены навыки управления процессами операционной системы