Отчет по лабораторной работе №6

Управление процессами

Сидорова Арина Валерьевна

Содержание

Список иллюстраций

1 Цель работы

Получить намыки управления процессами операционной системы.

2 выполнение лабораторной работы

2.1 Управление заданиями

1. Получим полномочия администратора
2. Введем следующие команды: sleep 3600 & dd if=/dev/zero of=/dev/null & sleep 7200
3. Введем jobs Мы увидим три задания, которые мы только что запустили. Пермые два имеют со- стояние Running, а последнее задание в настоящее время находится в состоянии Stopped.
4. Для продолжения мыполнения задания 3 в фоновом режиме введем bg 3 С помощью команды jobs посмотрим изменения в статусе заданий. (рис. 2.1)

```
avsidorova@avsidorova:~$ sudo -i
[sudo] пароль для avsidorova:
root@avsidorova:~# sleep 3600 &
[1] 4430
root@avsidorova:~# dd if=/dev/zero of=/dev/null &
[2] 4449
root@avsidorova:~# sleep 7200

^Z
[3]+  Остановлен    sleep 7200
root@avsidorova:~# jobs
[1]   Запущен          sleep 3600 &
[2]-  Запущен          dd if=/dev/zero of=/dev/null &
[3]+  Остановлен    sleep 7200
root@avsidorova:~# bg 3
[3]+ sleep 7200 &
root@avsidorova:~# fg 1
sleep 3600
^C
root@avsidorova:~# fg 2
dd if=/dev/zero of=/dev/null

^C244856641+0 records in
244856641+0 records out
125366600192 bytes (125 GB, 117 GiB) copied, 91,4181 s, 1,4 GB/s

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

Рис. 2.1: sleep 7200

6. Для перемещения задания 1 на передний план введем fg 1

7. Введем Ctrl + c , чтобы отменить задание 1. С помощью команды jobs посмотрим изменения в статусе заданий.

8. Проделаем то же самое для отмены заданий 2 и 3.

9. Откроем второй терминал и под учётной записью своего пользователя введем в нём: dd if=/dev/zero of=/dev/null &. (рис. 2.2)

```
avsidorova@avsidorova:~$ dd if=/dev/zero of=/dev/null &
[1] 4806
avsidorova@avsidorova:~$
```

Рис. 2.2: dd if=/dev/zero of=/dev/null &

10. Введем exit, чтобы закрыть второй терминал.

11. На другом терминале под учётной записью своего пользователя запустим top

мы увидим, что задание dd всё ещё запущено. Для мыхода из top используем q .(рис. 2.3)

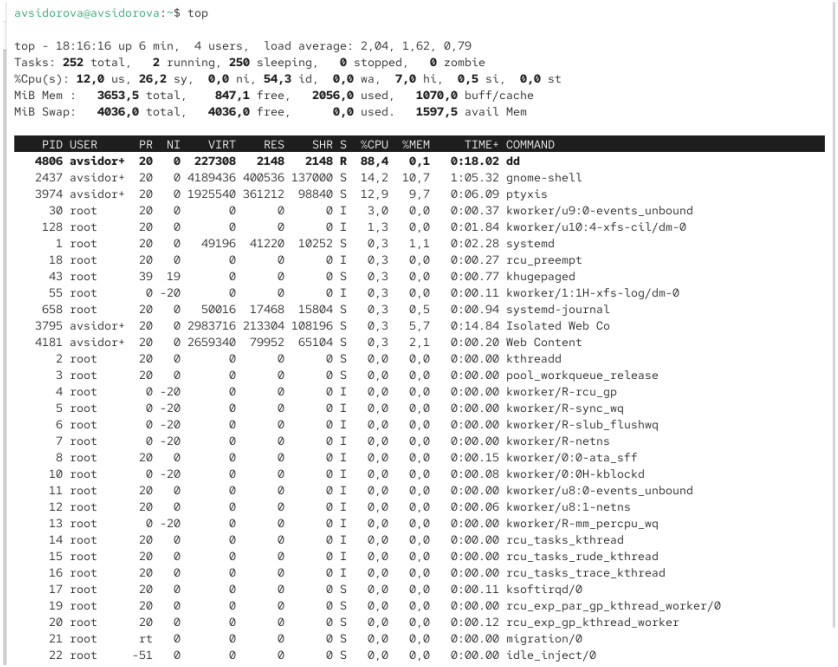

Рис. 2.3: top

12. Вновь запустим top и в нём используем k , чтобы убить задание dd. После этого мыйдем из top. (рис. 2.4)

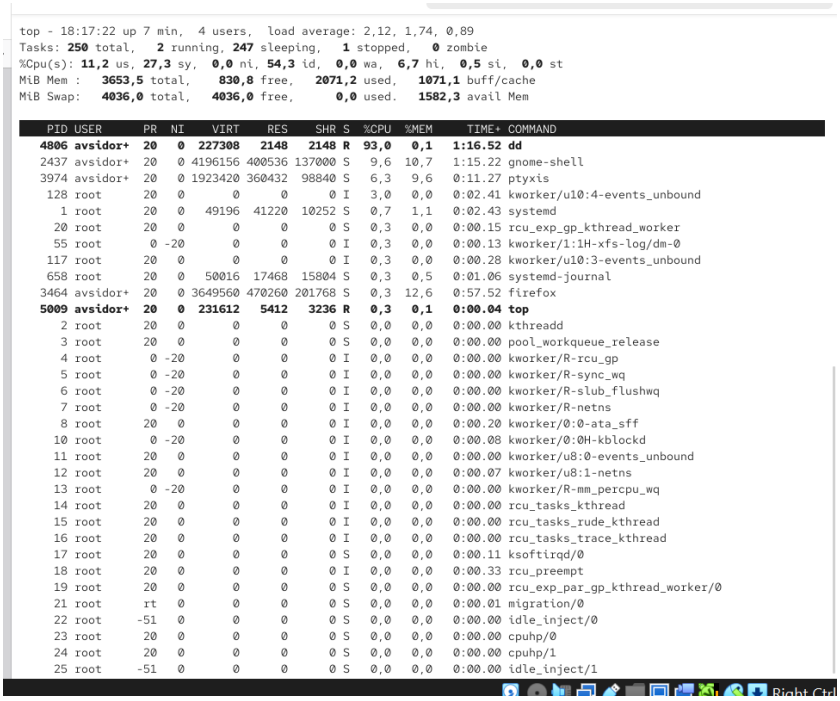

Рис. 2.4: top 2

2.2 Управление процессами

1. Получим полномочия администратора su -
2. Введем следующие команды: dd if=/dev/zero of=/dev/null & dd if=/dev/zero of=/dev/null & dd if=/dev/zero of=/dev/null & (рис. 2.5)

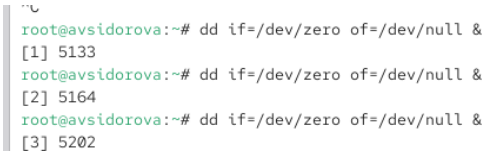

Рис. 2.5: x3 dd if=/dev/zero of=/dev/null &

3. Введем ps aux | grep dd Это показывает все строки, в которых есть букмы dd. Запущенные процессы dd идут последними. (рис. 2.6)

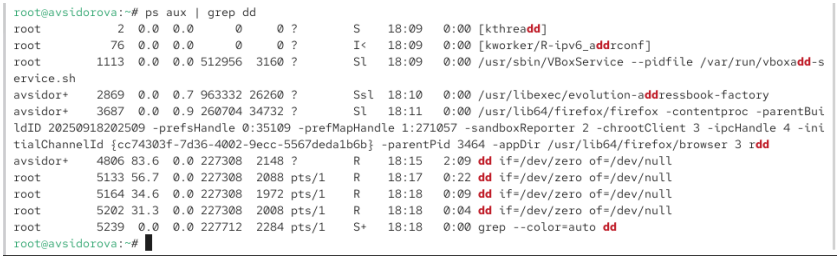

Рис. 2.6: ps aux | grep dd

4. Используем PID одного из процессов dd, чтобы изменить приоритет. Используем renice -n 5 (рис. 2.7)

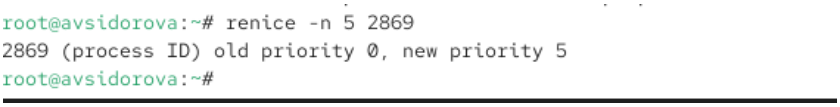

Рис. 2.7: renice -n 5

5. Введем ps fax | grep -B5 dd Параметр -B5 показывает соответствующие запросу строки, включая пять строк до этого. Поскольку ps fax показывает иерархию отношений между процессами, мы также увидем оболочку, из которой были запущены все процессы dd, и её PID. (рис. 2.8)

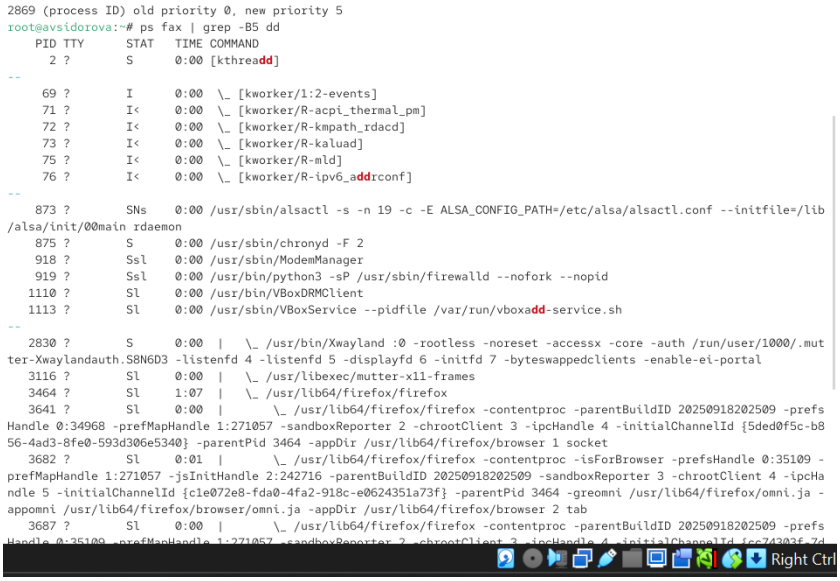

Рис. 2.8: ps fax | grep -B5 dd

6. Найдем PID корневой оболочки, из которой были запущены процессы dd, и введем kill -9 (заменив на значение PID оболочки). мы увидим, что наша корневая оболочка закрылась, а вместе с ней и все процессы dd. Остановка родительского процесса — простой и удобный способ остановить все его дочерние процессы. (рис. 2.9)

```
   4040 pts/0    Ss     0:00  |       \_ /usr/bin/bash
   4336 pts/0    S+     0:00  |           \_ sudo -i
   4379 pts/1    Ss     0:00  |               \_ sudo -i
   4380 pts/1    S      0:00  |                   \_ -bash
   5133 pts/1    R      1:06  |                       \_ dd if=/dev/zero of=/dev/null
   5164 pts/1    R      0:57  |                       \_ dd if=/dev/zero of=/dev/null
   5202 pts/1    R      0:47  |                       \_ dd if=/dev/zero of=/dev/null
   5552 pts/1    R+     0:00  |                       \_ ps fax
   5553 pts/1    S+     0:00  |                       \_ grep --color=auto -B5 dd
   3998 ?        S      0:00  \_ catatonit -P
   4806 ?        R      4:02  \_ dd if=/dev/zero of=/dev/null
root@avsidorova:~# kill -9 4380
Убито
avsidorova@avsidorova:~$
```

Рис. 2.9: kill -9

3 Самостоятельная работа

3.1 Задание 1

1. Запустим команду dd if=/dev/zero of=/dev/null трижды как фоновое задание. (рис. 3.1)

Рис. 3.1: dd if=/dev/zero of=/dev/null

2. Увеличим приоритет одной из этих команд, используя значение приоритета −5.

3. Изменим приоритет того же процесса ещё раз, но используем на этот раз значение −15. Разница в значении приоритетов (рис. 3.2)

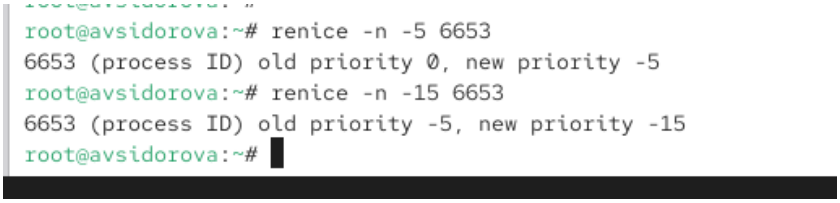

Рис. 3.2: Увеличение приоритета (5, 15)

4. Завершим все процессы dd, которые мы запустили. (рис. 3.3)

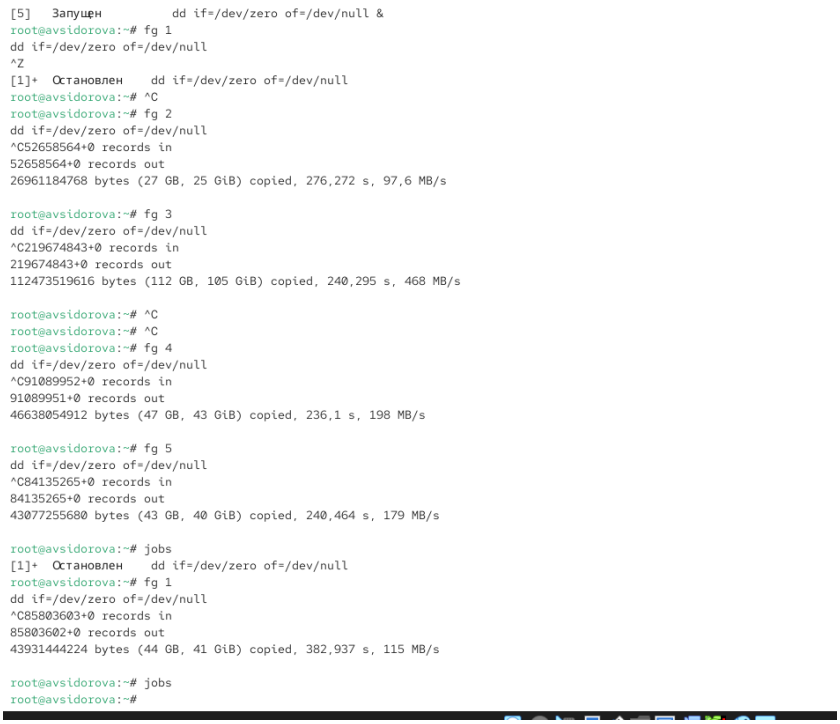

Рис. 3.3: Завершаем процессы

3.2 Задание 2

1. Запустим программу yes в фоновом режиме с подавлением потока вывода. (рис. 3.4)

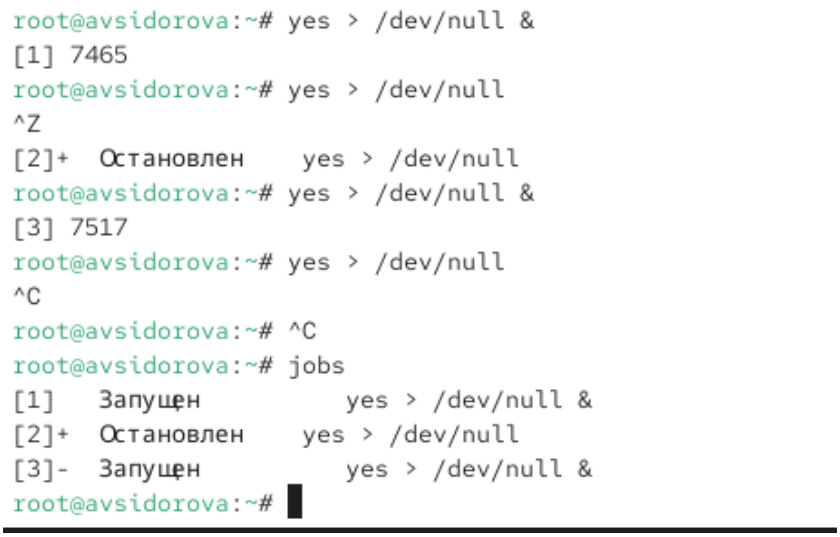

Рис. 3.4: Запускаем yes

2. Запустим программу yes на переднем плане с подавлением потока вывода. При- остановим выполнение программы. Заново запустим программу yes с теми же параметрами, затем завершим её выполнение.

3. Запустим программу yes на переднем плане без подавления потока вывода. При- остановим выполнение программы. Заново запустим программу yes с теми же параметрами, затем завершим её выполнение.

4. Проверим состояния заданий, воспользовавшись командой jobs. (рис. 3.5)

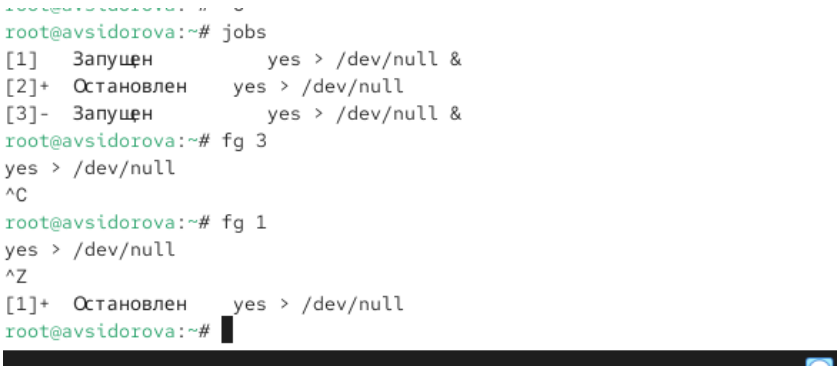

Рис. 3.5: jobs

5. Переведем процесс, который у нас выполняется в фоновом режиме, на передний план, затем остановим его.

6. Переведем любой наш процесс с подавлением потока вывода в фономый режим. (рис. 3.6)

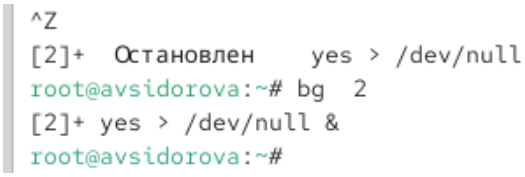

Рис. 3.6: bg 2

7. Проверим состояния заданий, воспользовавшись командой jobs. Обратим внимание, что процесс стал выполняющимся (Running) в фоновом режиме. (рис. 3.7)

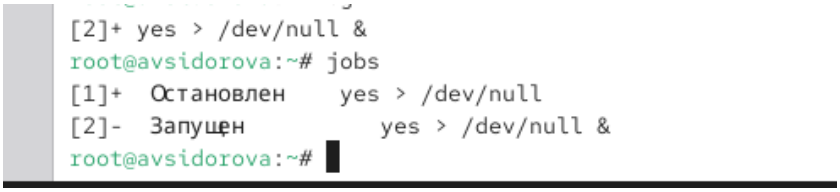

Рис. 3.7: jobs

8. Запустим процесс в фоновом режиме таким образом, чтобы он продолжил свою работу даже после отключения от терминала. (рис. 3.8)

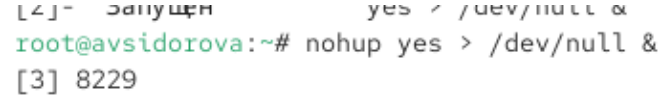

Рис. 3.8: nohup yes > /dev/null &

9. Закроем окно и заново запустим консоль. Убедимся, что процесс продолжил свою работу.

10. Получим информацию о запущенных в операционной системе процессах с помощью утилиты top. (рис. 3.9)

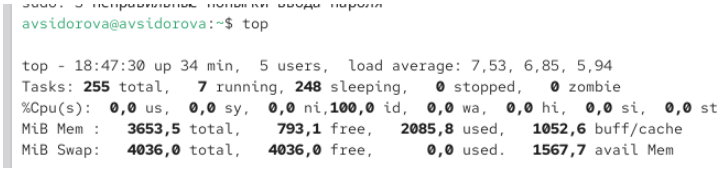

Рис. 3.9: top

11. Запустим ещё три программы yes в фоновом режиме с подавлением потока вывода. (рис. 3.10)

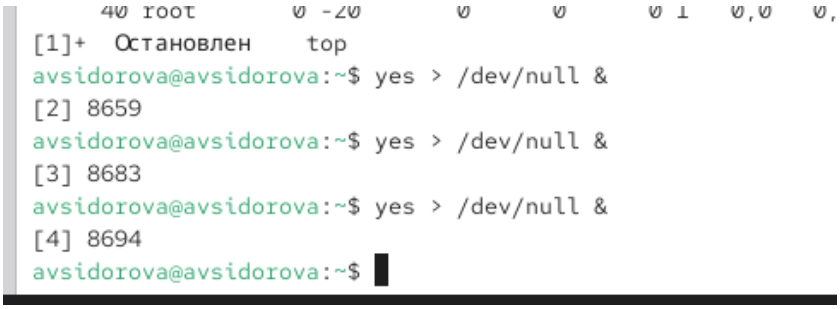

Рис. 3.10: x3 yes

12. Убеем два процесса: для одного используем его PID, а для другого — его идентифи- катор конкретного задания. (рис. 3.11)

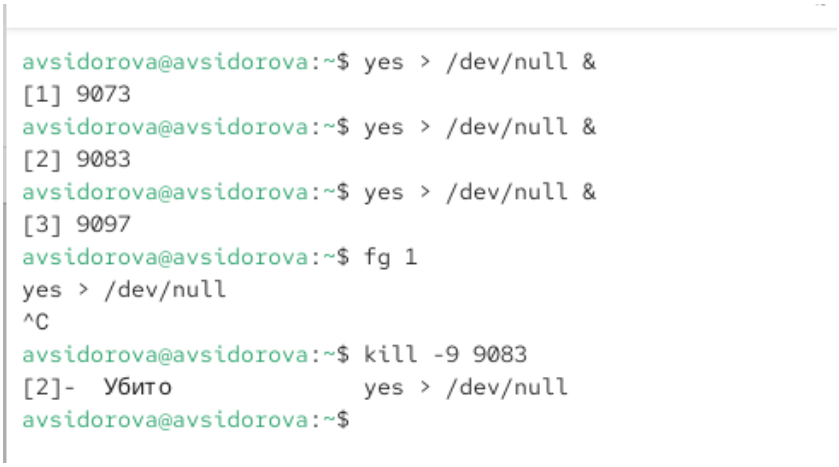

Рис. 3.11: kill + ctrl C

13. Попробуем послать сигнал 1 (SIGHUP) процессу, запущенному с помощью nohup, и обычному процессу.

14. Запустим ещё несколько программ yes в фоновом режиме с подавлением потока вывода.

15. Завершим их работу одновременно, используя команду killall. (рис. 3.12)

```
root@avsidorova:~# killall yes
root@avsidorova:~# jobs
[1]-  Завершено      yes > /dev/null
[2]+  Завершено      yes > /dev/null
```

Рис. 3.12: killall yes

16. Запустим программу yes в фоновом режиме с подавлением потока вывода. Используя утилиту nice, запустим программу yes с теми же параметрами и с приоритетом, большим на 5. Сравним абсолютные и относимльные приоритеты у этих двух про- цессов.

17. Используя утилиту renice, изменим приоритет у одного из потоков yes таким обра- зом, чтобы у обоих потоков приоритеты были равны. (рис. 3.13)

```
root@avsidorova:~# killall yes
root@avsidorova:~# jobs
[1]-  Завершено      yes > /dev/null
[2]+  Завершено      yes > /dev/null
```

Рис. 3.13: renice

4 Ответы на контрольные вопросы

1. jobs
2. Ctrl+Z, затем bg
3. Ctrl+C
4. Использовать kill или killall из другой сессии.
5. ps fax или pstree
6. renice -n -5 1234 (требуются права суперпользователя)
7. killall dd
8. killall mycommand
9. Нажать k, ввести PID процесса.
10. nice -n -10 command (с учетом прав доступа).

5 выводы

Получили намыки управления процессами операционной системы.