Лабараторная работа №6

Отчет

Славинский Владислав Вадимович

Содержание

# 1 Цель работы

Получить навыки управления процессами операционной системы.

# 2 Выполнение лабораторной работы

В консоли перейдем в режим работы суперпользователя, используя команду su -. (рис. 1)

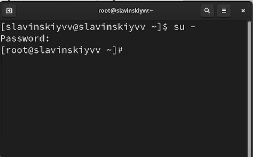


Рис. 1: Переход в режим суперпользователя

Введем следующие команды: sleep 3600 &, dd if=/dev/zero of=/dev/null & и sleep 7200(рис. 2)

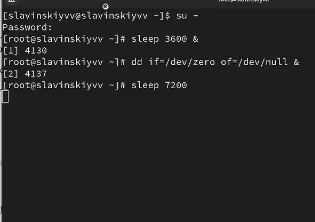


Рис. 2: Ввод команд

Поскольку мы запустили последнюю команду без & у нас есть 2 часа, прежде чем мы снова получите контроль над оболочкой. Введем ctrl+z, чтобы остановить процесс. (рис. 3)

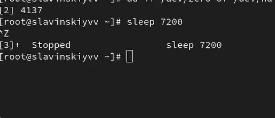


Рис. 3: Остановка процесса

Введем команду jobs. Мы видим три процесса, которые мы запустили. Два первых процесса имеют статус running, а последний имеет статус stopped. (рис. 4)

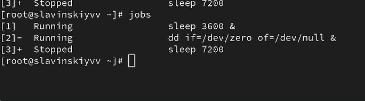


Рис. 4: Комада jobs

Для продолжения выполнения задания 3 в фоновом режиме введем bg 3 и посмотрим статус через jobs. Видим, что состояние изменилось на running. (рис. 5)

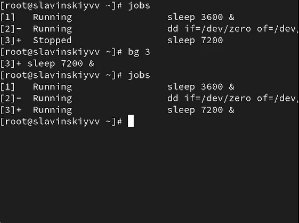


Рис. 5: Выполнение задания 3 в фоновом режиме

Для перемещения задания 1 на передний план введем fg 1. После чего отменим задание через ctrl+c и посмотрим статус (рис. 6)

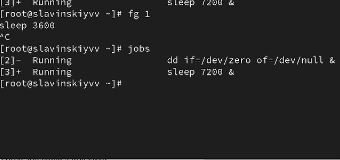


Рис. 6: Отмена задания 1

Проделаем то же самое с заданиями 2 и 3. (рис. 7)

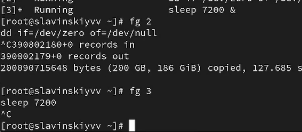


Рис. 7: Отмена заданий 2 и 3

Откроем второй терминал и под учётной записью своего пользователя введем в нём: dd if=/dev/zero of=/dev/null & и выйдем из него.(рис. 8)

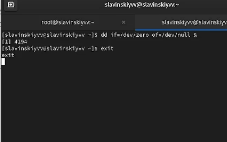


Рис. 8: Ввод команды во втором терминале

На другом терминале под учётной записью своего пользователя запустим top. Видим, что задание dd все еще запущено. Выйдем из top через q.(рис. 9)

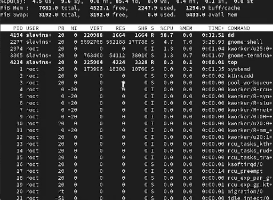


Рис. 9: Запуск top

Вновь запустим top и с помощью k убьем задание dd. Потом выйдем из top с помощью q. (рис. 10)

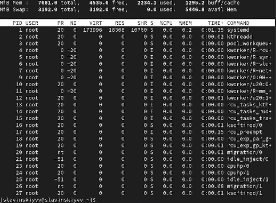


Рис. 10: Завершение задания dd

Перейдем в режим суперпользователя и введем следующую команду 3 раза dd if=/dev/zero of=/dev/null &. (рис. 11)

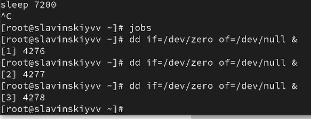


Рис. 11: Ввод команды

Введем ps aux | grep dd. Это показывает все строки, в которых есть буквы dd. Запущенные процессы dd идут последними. (рис. 12)

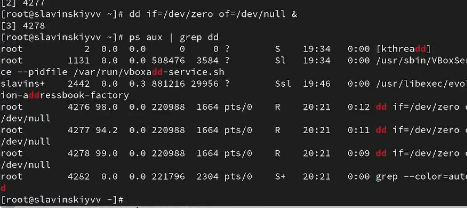


Рис. 12: Строки, в которых есть буквы dd

Используем PID одного из процессов dd, например 4276, чтобы изменить приоритет. Используем renice -n 5 . (рис. 13)

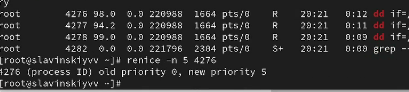


Рис. 13: Смена приоритета

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 увидим оболочку, из которой были запущены все процессы dd, и её PID.(рис. 14)

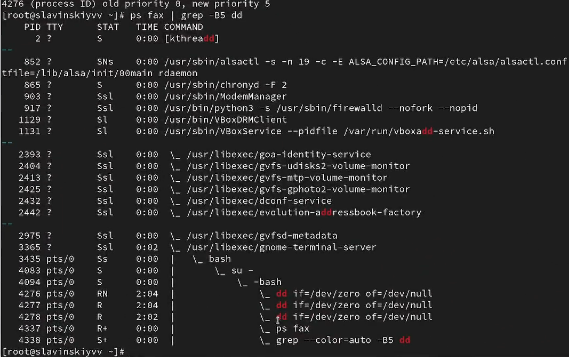


Рис. 14: Параметр -B5

Найдем PID корневой оболочки (у нас значение 4094), из которой были запущены процессы dd, и введем kill -9 (рис. 15)

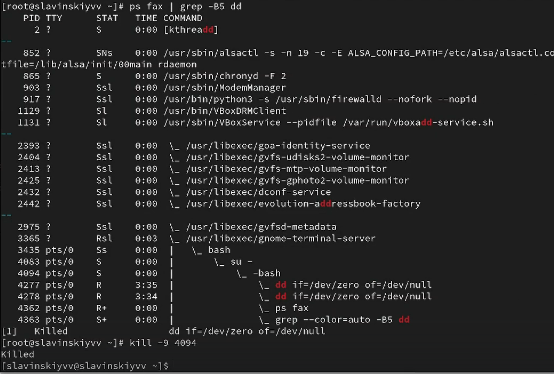


Рис. 15: Удаление корневой оболочки

Введем три раза команду dd if=/dev/zero of=/dev/null. Нам нужно запустить команду как фоновое значение, поэтому в конце добавляем &.(рис. 16)

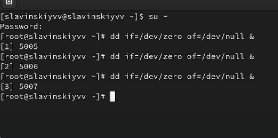


Рис. 16: Запуск команды в фоновом значении

Увеличим приоритет одной из этих команд, используя значение приоритета −5. Введем команду renice -n -5 5005(рис. 17)

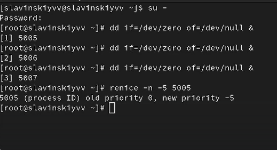


Рис. 17: Изменение приоритета

Изменим приоритет того же процесса ещё раз, но применим на этот раз значение −15. Разница в том, что при приоритете -15 процесс получает гораздо больше процессорного времени, чем при -5. Чем меньше значение, тем выше приоритет.(рис. 18)

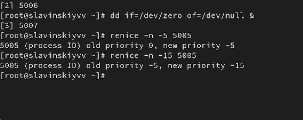


Рис. 18: Изменение приоритета

Завершим все процессы dd, которые мы запустили.(рис. 19)

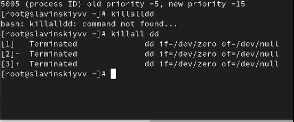


Рис. 19: Завершение процессов dd

Запустим программу yes в фоновом режиме с подавлением потока вывода с помощью команды yes > /dev/null &.(рис. 20)

yes в фоновом режиме с подавлением потока вывода

Рис. 20: yes в фоновом режиме с подавлением потока вывода

Запустим программу yes на переднем плане с подавлением потока вывода с помощью команды yes > /dev/null. Затем приостоновим программу через crtl+z. Потом заново запустим и завершим процесс через ctrl+c(рис. 21)

yes на переднем плане с подавлением потока вывода

Рис. 21: yes на переднем плане с подавлением потока вывода

Запустим программу yes на переднем плане без подавления потока вывода. Дальше приостановим выполнение программы через ctrl+z. Потом снова её запустим и завершим через ctrl+c(рис. 22)

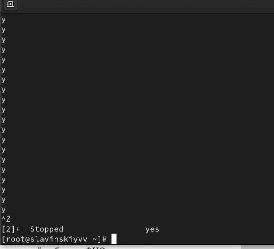


Рис. 22: yes на переднем плане без подавления потока вывода

Проверим состояния заданий, воспользовавшись командой jobs.(рис. 23)

Состояние заданий

Рис. 23: Состояние заданий

Переведем процесс, который у нас выполняется в фоновом режиме, на передний план, затем остановим его.(рис. 24)

Перевод процесса на передний план и его остановка

Рис. 24: Перевод процесса на передний план и его остановка

Переведем второй процесс с подавлением потока вывода в фоновый режим.(рис. 25)

Перевод процесса в фоновый режим

Рис. 25: Перевод процесса в фоновый режим

Проверим состояние заданий, воспользовавшись командой jobs. И видим, что второй процесс стал выполняться(рис. 26)

Проверка состояния заданий

Рис. 26: Проверка состояния заданий

Запустим процесс в фоновом режиме таким образом, чтобы он продолжил свою работу даже после отключения от терминала.(рис. 27)

Запуск процесса, чтобы продолжал работу после закрытия терминала

Рис. 27: Запуск процесса, чтобы продолжал работу после закрытия терминала

Закроем и откроем заново консоль. Убедимся, что процесс продолжил свою работу. Получим информацию с помощью утилиты top.(рис. 28)

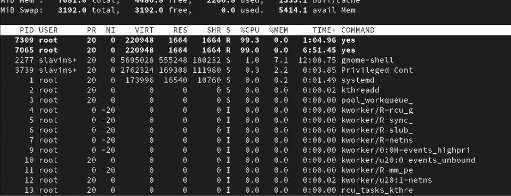


Рис. 28: Информация о запущенных в операционной системе процессах

Запустим ещё три программы yes в фоновом режиме с подавлением потока вывода..(рис. 29)

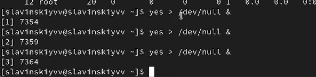


Рис. 29: Запуск программ yes в фоновом режиме с подавлением потока вывода

Убьем два процесса: для одного используйте его PID, а для другого — его идентификатор конкретного задания.(рис. 30)

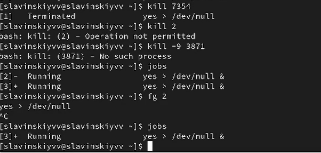


Рис. 30: Уничтожение процессов

Попробуем послать сигнал 1 (SIGHUP) процессу, запущенному с помощью nohup, и обычному процессу.(рис. 31)

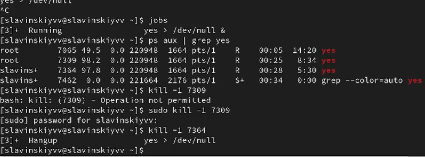


Рис. 31: Послание сигнала 1 процессам

Запустим ещё несколько программ yes в фоновом режиме с подавлением потока вывода.(рис. 32)

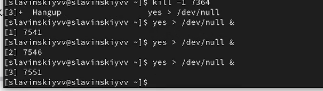


Рис. 32: Запуск программ yes в фоновом режиме с подавлением потока вывода

Завершим их работу одновременно, используя команду killall.(рис. 33)

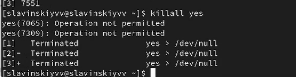


Рис. 33: Завершение их работы с помощью killall

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. Видим, что приоритеты у них разные.(рис. 34)

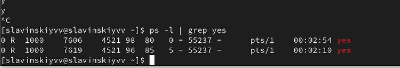


Рис. 34: Запуск двух программ yes, но у одной программы приоритет больше на 5

Используя утилиту renice, изменим приоритет у одного из потоков yes таким образом, чтобы у обоих потоков приоритеты были равны. Меняем приоритет у 7506 на 5 и получается, что приоритеты теперь равны.(рис. 35)

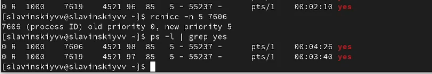


Рис. 35: Установка равных приоритетов

# 3 Выводы

В ходе выполнения лабораторной работы были получены навыки управления процессами операционной системы.

# 4 Ответы на контрольные вопросы

1. jobs
2. ctrl+z, bg
3. ctrl+c
4. kill
5. ps fax
6. renice -n -5 1234
7. killall -9 dd
8. killall mycommand
9. В интерфейсе top, чтобы убить процесс нужно нажать клавишу k.
10. nice -n mycommand