**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Программа ДПО Intaro

ЛАБОРАТОРНАЯ РАБОТА №3

“Процессы и управление ими в операционной системе Linux”

Студент \_\_\_\_\_\_\_\_\_\_ Целищев А.Е.

ПМ-21-2

Руководитель \_\_\_\_\_\_\_\_\_\_ Кургасов В.В.

Доцент каф. АСУ

Липецк 2023 г

**Цель работы**

Ознакомиться на практике с понятием процесса в операционной системе. Приобрести опыт и навыки управления процессами в операционной системе Linux.

**Ход работы**

*Часть 1*

Войдем в систему под пользовательской учетной записью и найдем файл с образом ядра. Видим несколько файлов, содержащих “vmlinuz”, среди которых два содержат в названии версии ядра. Опираясь на старшую версию, делаем вывод, что версия Linux в данной системе: 5.15.0-87

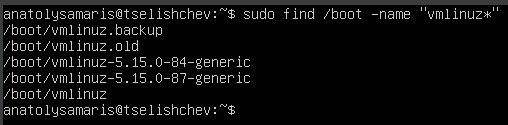


Рис 1. Результат поиска файла с образом ядра Linux.

Посмотрим процессы с помощью команды ps -f. В результате получим информацию в виде таблицы, где UID - имя пользователя, от имени которого работает процесс; PID - идентификатор процесса; PPID - идентификатор родительского процесса (так, процесс 1487 - команда ps - была вызвана из терминала bash, которая как процесс имеет PID 1064, и потому для команды ps PPID будет равен 1064); С - расходование ресурсов процессора в процентах; TTY - если процесс привязан к терминалу, то будет выведен его номер; TIME - общее время выполнения процесса; CMD - команда, которой процесс был запущен.

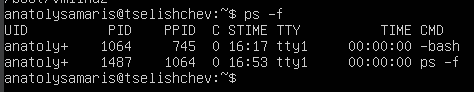


Рис 2. Результат вывода команды ps -f.

Создадим в текстовом редакторе два сценария:

- loop: while true; do true; done.

- loop2: while true; do true; echo ‘Hello’; done.

Запустим loop2 на переднем плане командой sh loop2

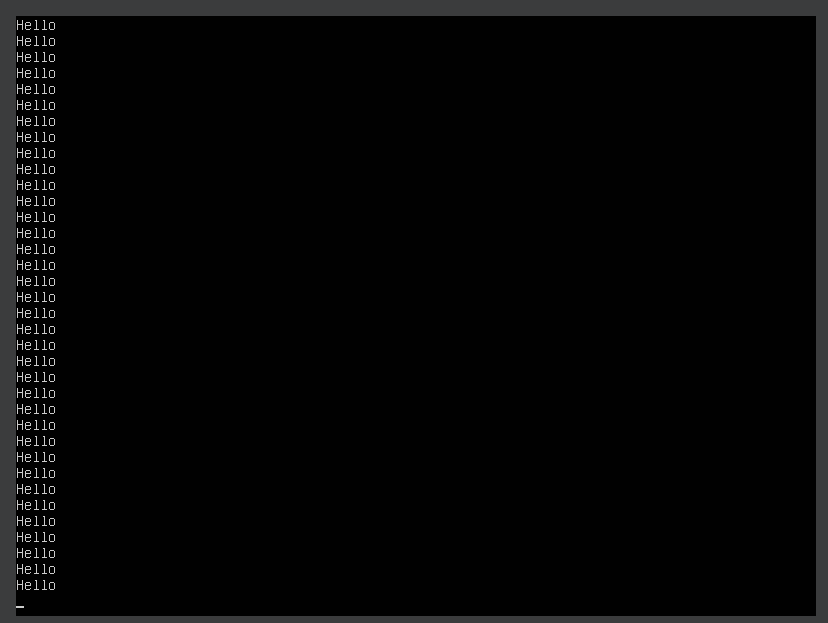


Рис 3. Работа сценария loop2.

Остановим процесс, отправив сигнал STOP (SIGINT). Для этого можно использовать сочетание клавиш Ctrl + C.

Запустим loop свернутым. Убедимся, что процесс выполняется. Видим также, что процесс loop потребляет ~99% ресурсов среди процессов (здесь за 100% принимаются все выделенные под процессы ресурсы, а не все возможные ресурсы процессора.).

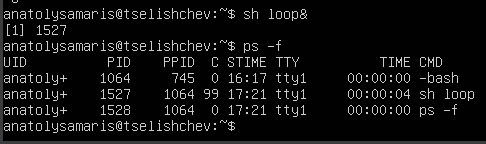


Рис 4. Запуск процесса loop и просмотр работающих процессов.

Командой kill -15 1061 (процесс loop был перезапущен, и ему был присвоен PID 1061, это значение и передаем команде kill) отправим сигнал SIGTERM для завершения процесса. В выводе ps -f увидим запись: Terminated sh loop.

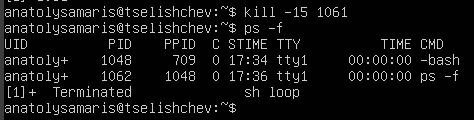


Рис 5. В результате SIGTERM процесс loop остановлен и это указано в ps -f.

Если запустить процесс еще раз и исполнить kill -9 PID, то в выводе ps -f будет показано, что процесс «убит», то есть безусловно завершен сигналом SIGKILL.

Запустим ещё один терминал bash. В выводе ps -f новый терминал не имеет префикса -, а также мы можем выявить его по большему значению PID либо времени начала выполнения STIME.

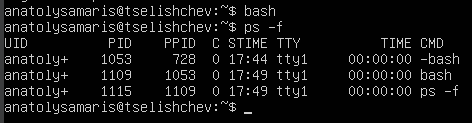


Рис 6. Результат запуска еще одного терминала.

*Часть 2*

Создадим задачу в фоновом режиме (например, терминал bash), а затем переведём его в интерактивный режим. На скриншоте видно, что сначала запущенный терминал помечается как «Stopped». С помощью jobs узнаем фоновые процессы (там под номером 1 будет запущенный в фоне bash), и с помощью команды fg 1 мы, по сути, перейдём в запущенный терминал. Если посмотреть из него jobs, ничего не будет выведено.

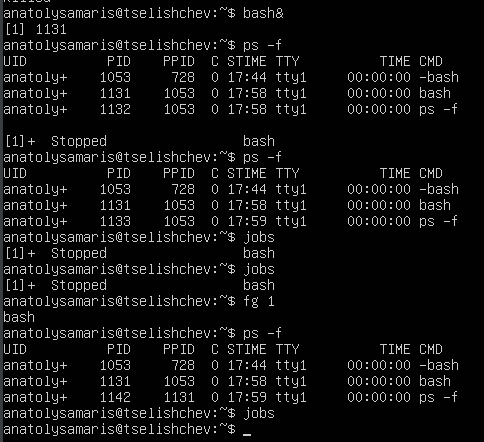
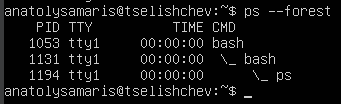


Рис 7. Эксперимент с переводом bash из фона в интерактивный режим.

*Часть 3*

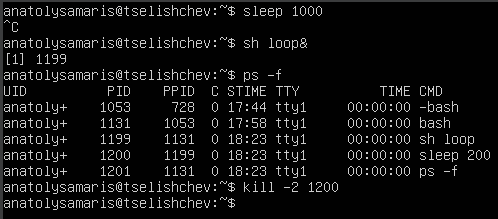
*Вариант 6*

1. Отобразить информацию о процессах, начиная с указанного идентификатора, с выделением цветом текущего процесса и его предков.



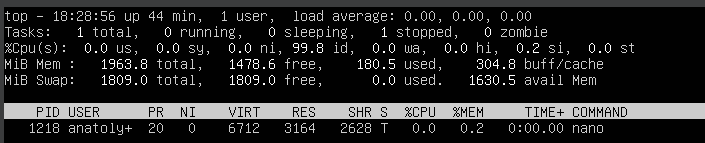
1. Завершить выполнение процесса, владельцем которого является текущий пользователь, с помощью сигнала SIGINT двумя способами: задав имя сигнала и используя комбинацию клавиш.

- Перепишем сценарий loop, описав в нем команду sleep 200, чтобы не вызывать ошибок из-за бесконечного цикла.

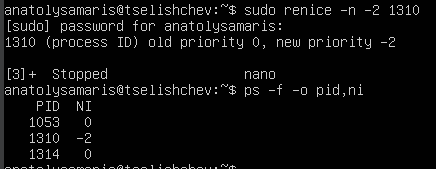


1. Запустить редактор nano, определить приоритет редактора. Запустить новый процесс данного редактора с увеличенным на 2 значением приоритета.

- Исполним nano file &, чтобы запустить редактор в фоне, а затем с помощью top -p PID посмотрим информацию о процессе, в том числе его приоритет в таблице внизу, в колонке NI (niceness). Приоритет процесса равен 0



- Запустим еще раз процесс редактора, повысив его приоритет на 2. Так как значения процессов лежат в диапазоне между -20 и 19, где -20 является наивысшим приоритетом, в данном случае повышение приоритета редактора означает, что надо запустить его процесс с приоритетом -2. Используем команду sudo renice -n -2 PID для переопределения приоритета процесса.



**Выводы**

Были рассмотрены процессы в операционной системе Linux, способы управления ими; проведены эксперименты с утилитами для управления процессами. Получены практические навыки управления процессами в операционной системе Linux.