**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра Прикладной математики

Лабораторная работа № 3

Процессы в Linux и управление ими

Вариант 7

Студент \_\_\_\_\_\_\_\_\_\_\_ Шишкина А.Л.

Группа ПМ-21-2

Руководитель

Доцент \_\_\_\_\_\_\_\_\_\_\_ Кургасов В.В.

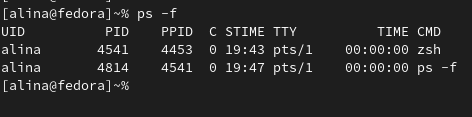
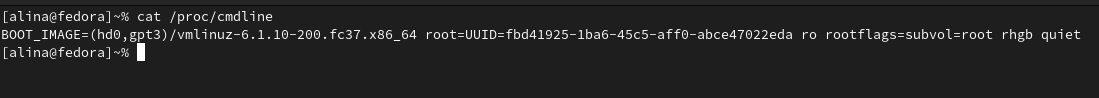
Цель работы:

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

Ход работы:

**Часть 1**

1. Зайдем в систему под пользовательской учетной записью, откроем терминал и введем команду cat /proc/cmdline, чтобы выснить номер версии linux.

  
2. Просмотрим процессы с помощью команды ps -f.

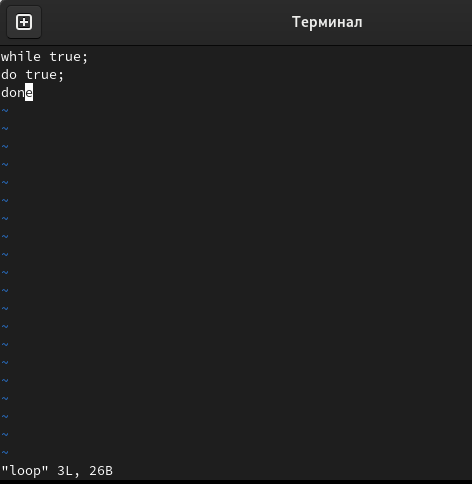
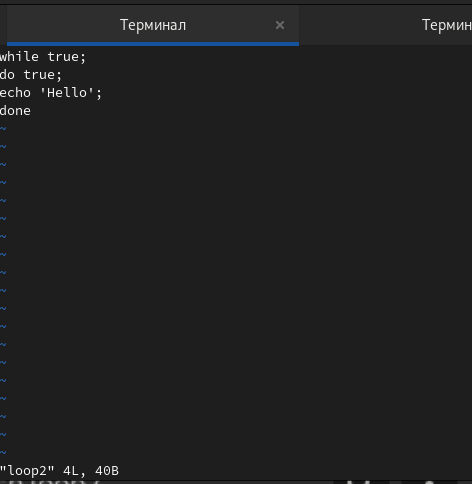
Вот значение основных колонок в выводе утилиты:

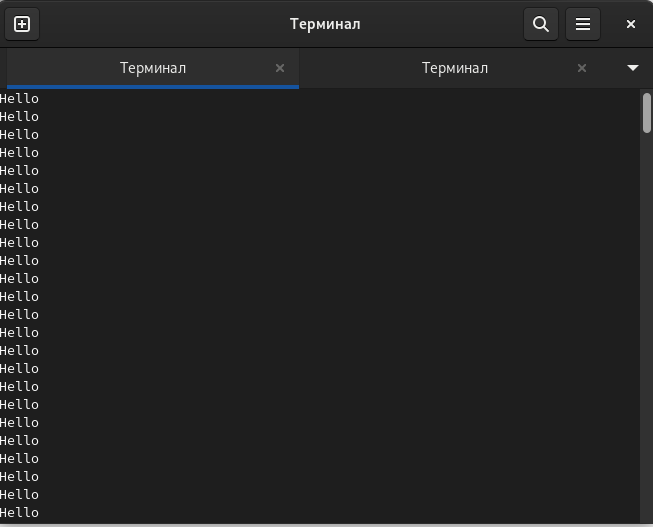
* **UID** - имя пользователя, от имени которого работает процесс;
* **PID** - идентификатор пользователя;
* **PPID** - идентификатор родительского процесса;
* **C** - расходование ресурсов процессора, в процентах;
* **TTY** - если процесс привязан к терминалу, то здесь будет выведен его номер;
* **TIME** - общее время выполнения процесса (user + system);
* **CMD** - команда, которой был запущен процесс, если программа не может прочитать аргументы процесса, он будет выведен в квадратных скобках;

Судя по рисунку, процессы не расходуют ресурсы процессора и общее время их выполнения 0 сек.

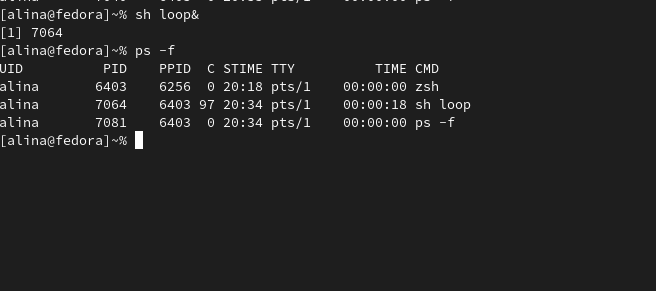
3. Создадим в редакторе vi сценарии loop и loop2.

Сценарии loop и loop2 представлены ниже

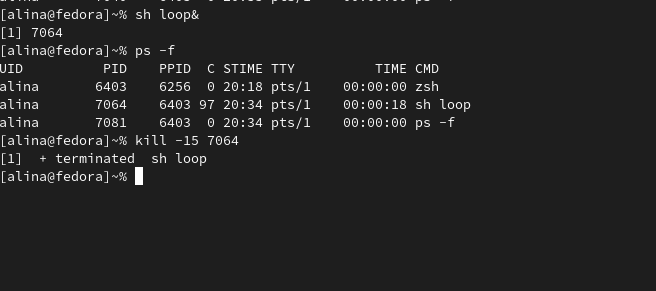
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
4. Запустим loop2 на переднем плане командой: sh loop2

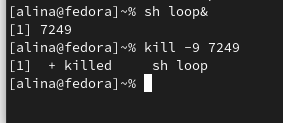
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Остановим процесс с помощью клавиш Ctrl + C.

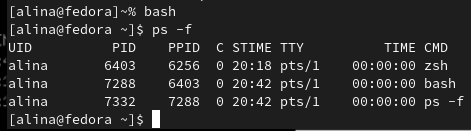
5. Запускаем loop свернутым и проверяем, что этот процесс выполняется



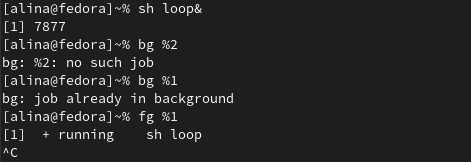
Завершим процесс loop командой kill -15 PID и проверим, что процесс был убит.

  
  
Снова запускаем процесс свернутым, отсылаем команду kill с кодом 9 и проверяем, что процесс был убит.

  
  
  
  
  
  
  
Запускаем еще один экземпляр оболочки: bash. Проверим, что bash действительно запущен.

  
  
  
  
  
  
  
**Часть 2**

1. Запускаем несколько процессов в фоне и один в интерактиве, а также перевожу процесс из фонового в интерактивный режим и наоборот



Создадим именованный канал для архивирования и осуществим передачу в канал.  
  
  
  
 **Часть 3.**

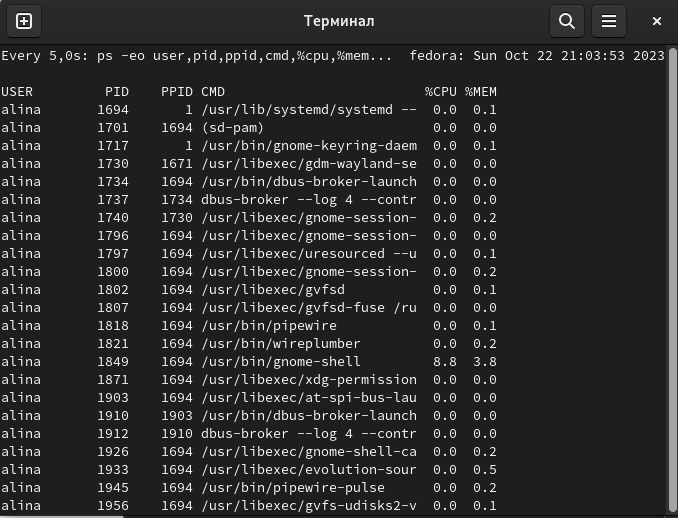
Вариант 7.

1. Вывести информацию о состоянии процессов системы в реальном режиме с обновлением один раз в 5 секунд. Отсортировать вывод по идентификатору пользователя по возрастанию и убыванию.

Команда:

watch -n 5 "ps -eo user,pid,ppid,cmd,%cpu,%mem --sort=user | less"

Результат работы команды

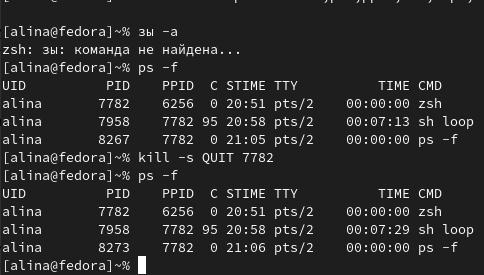
  
  
2. Завершить выполнение процесса, владельцем которого является текущий пользователь, с помощью сигнала SIGQUIT двумя способами: задав имя сигнала и используя комбинацию клавиш.  
  
Команда 1:

kill -s QUIT <PID>

Команда 2:

CTRL+\

Результат выполнения команд

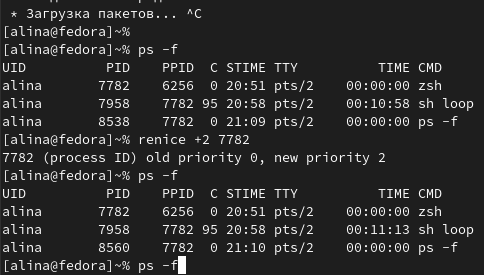


3. Изменить на 2 единицы приоритет процесса, запущенного из командного интерпретатора.

Команда:

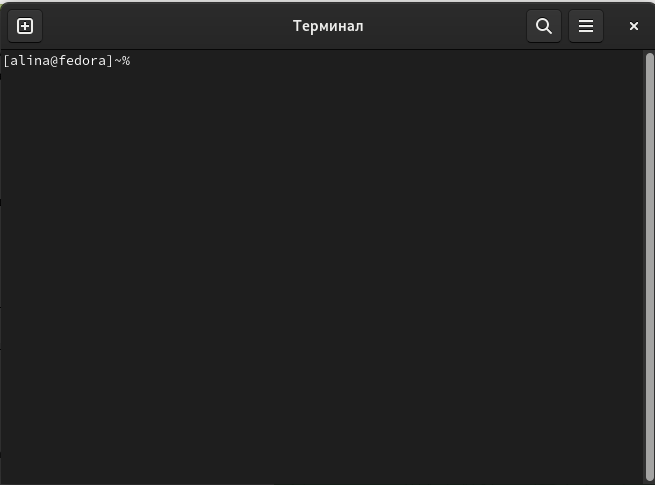
renice +2 <PID>

Результат работы команды



**Часть 4.**

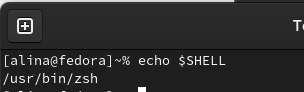
1. Открыть окно интерпретатора команд

  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
2. Вывести общую информацию о системе:

a) Вывести информацию о текущем интерпретаторе команд.

Команда:

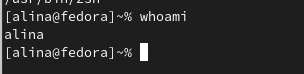
echo $SHELL



b) Вывести информацию о текущем пользователе.

Команда:

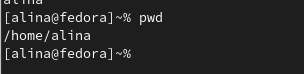
whoami



c) Вывести информацию о текущем каталоге.

Команда:

pwd

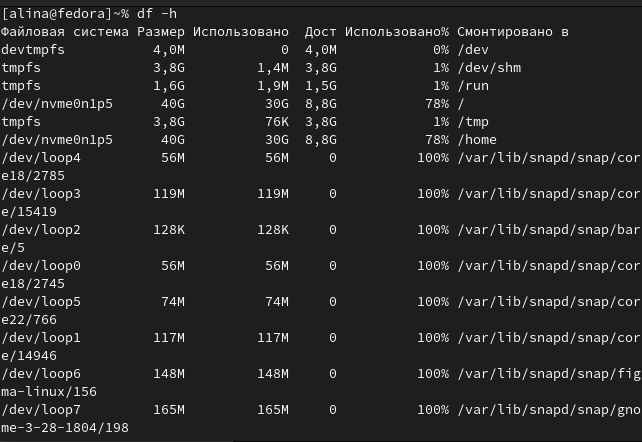


d) Вывести информацию об оперативной памяти и области подкачки.

Команда:

free -h

  
e) Вывести информацию о дисковой памяти.

Команда: df -h  
  


3. Выполнить команды получения информации о процессах:

a) Получить идентификатор текущего процесса (PID).

Команда:

echo $$

  
  
  
  
b) Получить идентификатор родительского процесса (PPID).

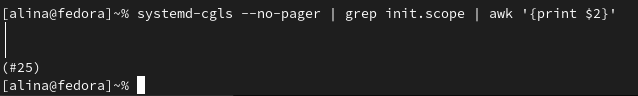
Команда:

echo $PPID

  
  
  
  
c) Получить идентификатор процесса инициализации системы.

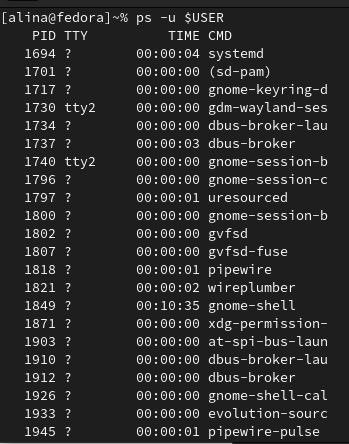
Команда:

systemd-cgls --no-pager | grep init.scope | awk '{print $2}'

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

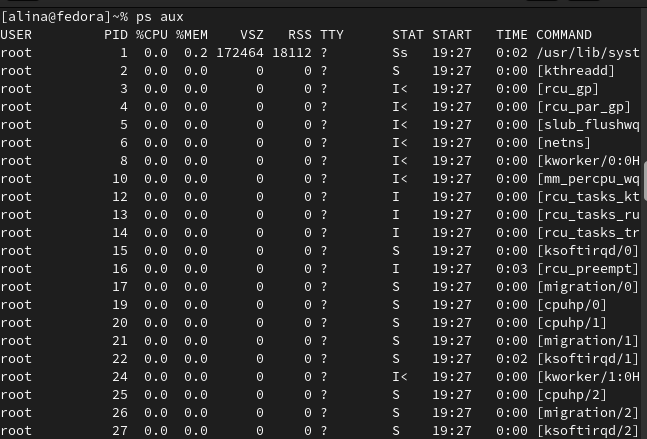
Команда:

ps -u $USER

  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
e) Отобразить все процессы.

Команда:

ps aux

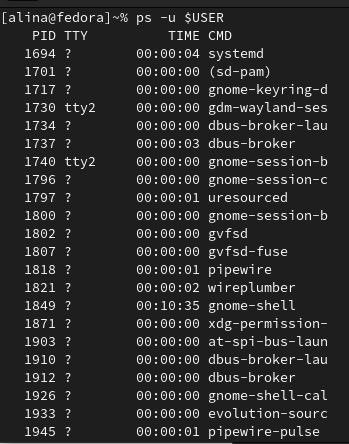


4. Выполнить команды управления процессами:

a) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе.

Команда:

ps -u $USER



b) Определить текущее значение nice по умолчанию.

Команда:

nice  
  


c) Запустить интерпретатор bash с понижением приоритета nice –n 10 bash.

Команда:

nice -n 10 bash

  
  
  
  
d) Определить PID запущенного интерпретатора.

Команда:

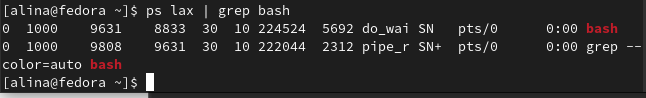
echo $$

  
  
  
  
e) Установить приоритет запущенного интерпретатора равным 5: renice –n 5 <PID процесса>.

Команда:

renice -n 5 <PID>  
  
  
  
  
  
f) Получить информацию о процессах bash: ps lax | grep bash.

Команда:

ps lax | grep bash  
  


Вывод:

Ознакомилась на практике с понятием процесса в операционной системе.  
Приобрела опыт и навыки управления процессами в операционной системе  
Linux.