

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа №2

по Дисциплине «Операционная система Linux»

Процессы в операционной системе Linux

Студент

Глебов Д.А.

Группа АИ-18

Руководитель

Кургасов В.В.

к.п.н.

Липецк 2020 г.

Оглавление	
Цель работы	3
Задание кафедры.....	4
Ход работы.....	6
Часть II.....	14
Часть III	19
Вывод.....	22

Цель работы

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

Задание кафедры

Часть I (выполняют все).

- 1) Загрузиться не root, а пользователем.
- 2) Найти файл с образом ядра. Выяснить по имени файла номер версии Linux.
- 3) Посмотреть процессы `ps -f`. Прокомментировать. Для этого почитать `man ps`.
- 4) Написать с помощью редактора `vi` два сценария `loop` и `loop2`. Текст сценариев: `Loop: while true; do true; done` `Loop2: while true; do true; echo 'Hello'; done`
- 5) Запустить `loop2` на переднем плане: `sh loop2`. 6) Остановить, пошлав сигнал `STOP`.
- 7) Посмотреть последовательно несколько раз `ps -f`. Записать сообщение, объяснить.
- 8) Убить процесс `loop2`, пошлав сигнал `kill -9 PID`. Записать сообщение. Прокомментировать.
- 9) Запустить в фоне процесс `loop`: `sh loop&`. Не останавливая, посмотреть несколько раз: `ps -f`. Записать значение, объяснить.
- 10) Завершить процесс `loop` командой `kill -15 PID`. Записать сообщение, прокомментировать.
- 11) Третий раз запустить в фоне. Не останавливая убить командой `kill -9 PID`.
- 12) Запустить еще один экземпляр оболочки: `bash`.
- 13) Запустить несколько процессов в фоне. Останавливать их и снова запускать. Записать результаты просмотра командой `ps -f`.

Часть II (выполняют все).

1. Запустить в консоли на выполнение три задачи, две в интерактивном режиме, одну - в фоновом.
2. Перевести одну из задач, выполняющихся в интерактивном режиме, в фоновый режим.
3. Провести эксперименты по переводу задач из фонового режима в интерактивный и наоборот.
4. Создать именованный канал для архивирования и осуществить передачу в канал списка файлов домашнего каталога вместе с подкаталогами (ключ -R), одного каталога вместе с файлами и подкаталогами.

Часть III. Индивидуальные задания.

Вариант 12.

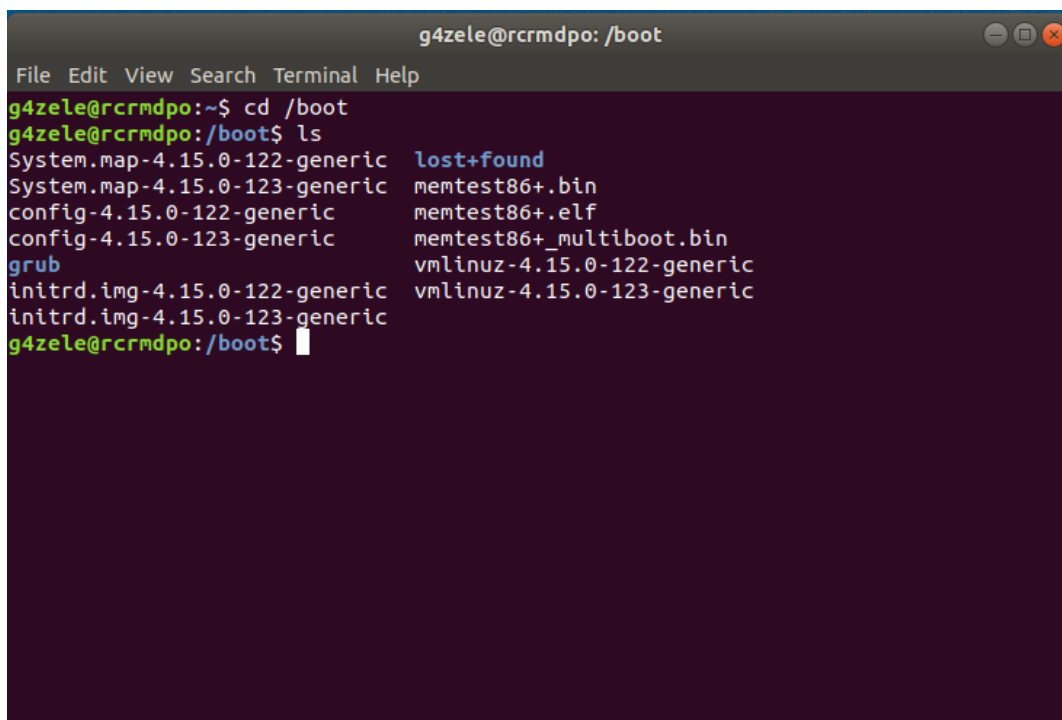
1. Вывести в реальном режиме информацию о состоянии процессов системы и их командных аргументах с сортировкой по идентификатору пользователя. Измените время обновления информации, не выходя из команды.
2. С помощью сигнала SIGINT завершить процесс, владельцем которого является текущий пользователь, с указанием имени процесса.
3. Откройте на редактирование один и тот же файл двумя редакторами. Определите PID и имя владельца процессов, которые используют данный файл. С помощью сигнала SIGTERM завершите процессы, которые используют данный файл с выдачей запроса на подтверждение от пользователя.
4. В отчете предоставьте все шаги ваших действий. То есть следует привести следующее: текст задания, а следом за ним снимок экрана консоли с результатами выполнения задания. Кроме того, перед скриншотом следует привести текстовую запись использованных команд. Кратко поясните результаты выполнения всех команд.

Ход работы

Часть I

Для начала работы нам нужно загрузиться не root, а пользователем. Для этого просто запустим терминал с помощью комбинации клавиш CTRL + ALT + T.

Затем найдём файл ядра в директории boot:



```
g4zele@rcrmdpo: /boot
File Edit View Search Terminal Help
g4zele@rcrmdpo:~$ cd /boot
g4zele@rcrmdpo:/boot$ ls
System.map-4.15.0-122-generic  lost+found
System.map-4.15.0-123-generic  memtest86+.bin
config-4.15.0-122-generic      memtest86+.elf
config-4.15.0-123-generic      memtest86+_multiboot.bin
grub                           vmlinuz-4.15.0-122-generic
initrd.img-4.15.0-122-generic  vmlinuz-4.15.0-123-generic
initrd.img-4.15.0-123-generic
g4zele@rcrmdpo:/boot$
```

Рисунок 1 – Версия ядра

По рисунку 1 видно (цифры в названии файла vmlinuz), что ядро имеет версию 4.15.0.

Затем посмотрим запущенные процессы с помощью команды `ps -f`:

```
g4zele@rcrmdpo: /boot$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
g4zele   3041   3030  0  17:45 pts/0    00:00:00 bash
g4zele   3053   3041  0  17:47 pts/0    00:00:00 ps -f
g4zele@rcrmdpo: /boot$
```

Рисунок 2 – Запущенные процессы

Поясним содержание рисунка 2:

UID - имя пользователя, от имени которого работает процесс;

PID - идентификатор пользователя;

PPID - идентификатор родительского процесса пользователя;

C - расходование ресурсов процессора, в процентах;

STIME - время, когда процесс был запущен;

TTY - если процесс привязан к терминалу, то здесь будет выведен его номер;

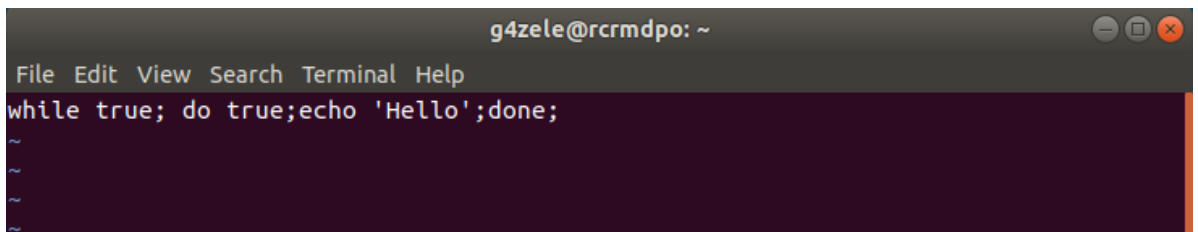
TIME - общее время выполнения процесса (user + system);

CMD - команда, которой был запущен процесс, если программа не может прочитать аргументы процесса, он будет выведен в квадратных скобках;

Теперь напишем в редакторе Vi два сценария `loop` и `loop2` и сохраним их:

```
g4zele@rcrmdpo: ~
File Edit View Search Terminal Help
while true; do true; done;
~
~
~
~
1,3 All
```

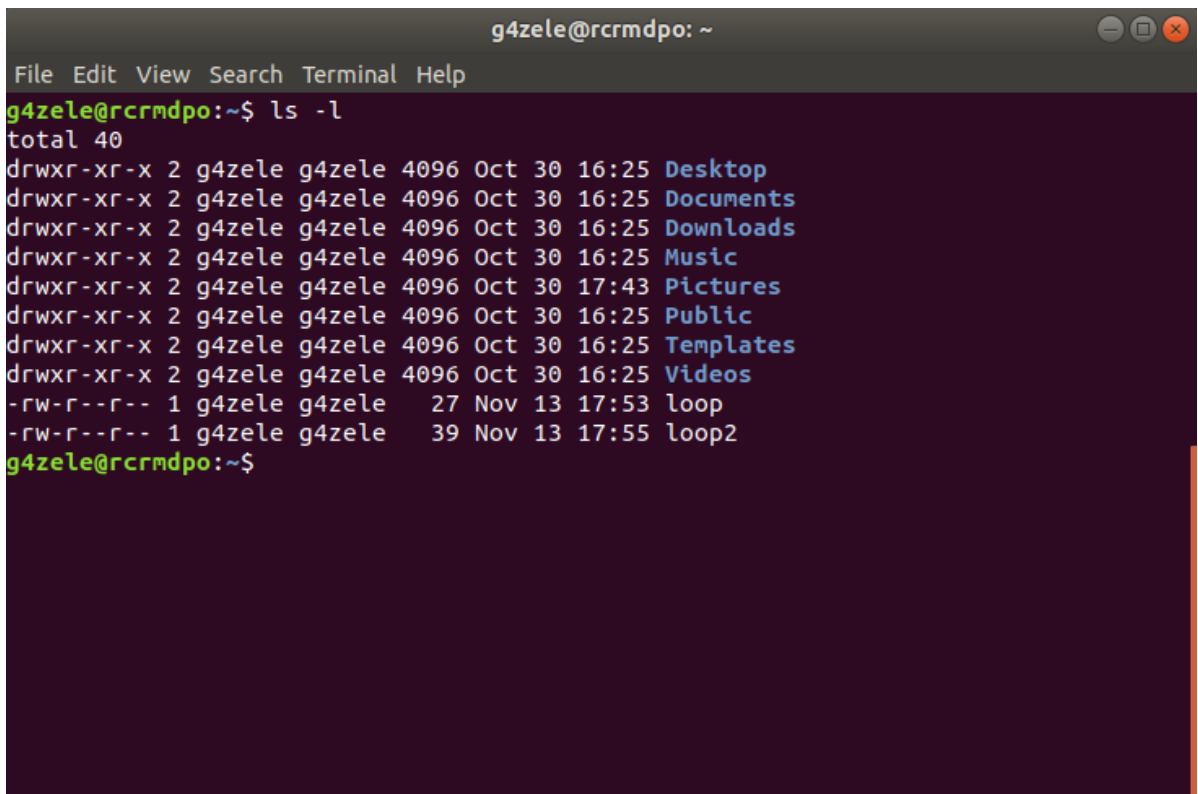
Рисунок 3 – Сценарий `loop`

A terminal window titled 'g4zele@rcrmdpo: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal contains the following text:

```
while true; do true;echo 'Hello';done;
~
~
~
~
```

Рисунок 4 – Сценарий loop2

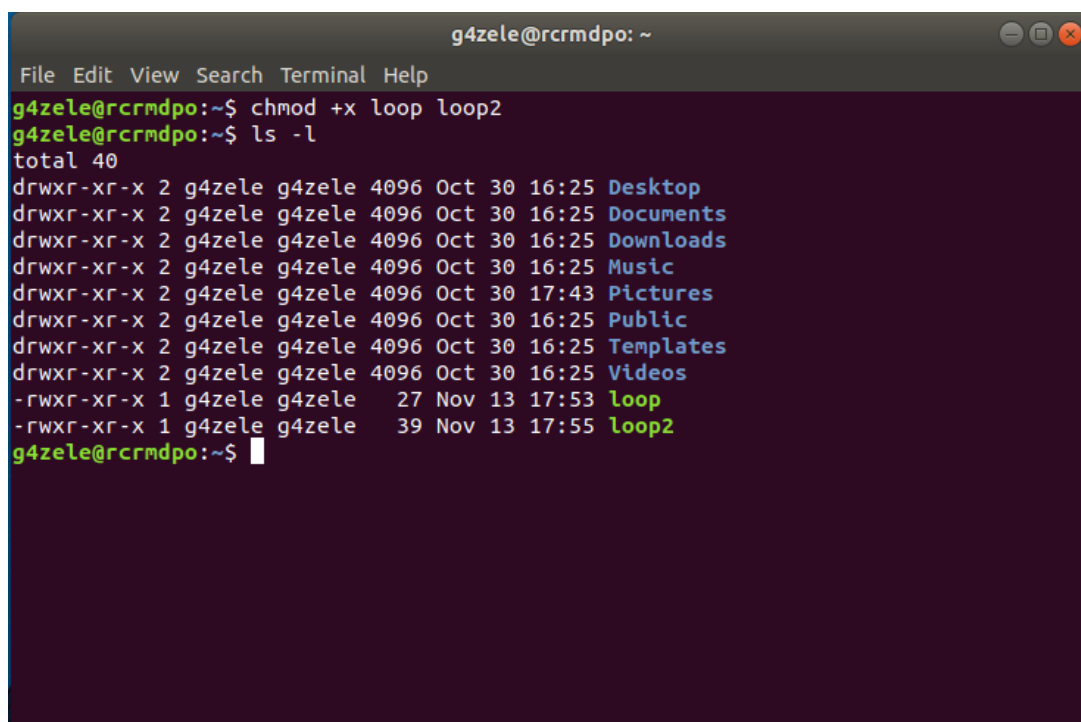
Проверим каталог на наличие созданных файлов:

A terminal window titled 'g4zele@rcrmdpo: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'ls -l' and its output:

```
g4zele@rcrmdpo:~$ ls -l
total 40
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Desktop
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Documents
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Downloads
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Music
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 17:43 Pictures
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Public
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Templates
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Videos
-rw-r--r-- 1 g4zele g4zele   27 Nov 13 17:53 loop
-rw-r--r-- 1 g4zele g4zele   39 Nov 13 17:55 loop2
g4zele@rcrmdpo:~$
```

Рисунок 5 – Каталог с файлами loop.sh и loop2.sh

Далее следует изменить права на файлы loop и loop2 и добавим право на исполнение с помощью команды chmod:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ chmod +x loop loop2  
g4zele@rcrmdpo:~$ ls -l  
total 40  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Desktop  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Documents  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Downloads  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Music  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 17:43 Pictures  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Public  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Templates  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Videos  
-rwxr-xr-x 1 g4zele g4zele 27 Nov 13 17:53 loop  
-rwxr-xr-x 1 g4zele g4zele 39 Nov 13 17:55 loop2  
g4zele@rcrmdpo:~$
```

Рисунок 6 – Изменение прав на файлы

Теперь мы можем запустить сценарий на исполнение в интерактивном режиме с помощью команды `sh`. Сделаем это, а также остановим процесс с помощью комбинации клавиш `CTRL + Z`:

```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
Hello  
^Z  
[1]+  Stopped                  sh loop2  
g4zele@rcrmdpo:~$
```

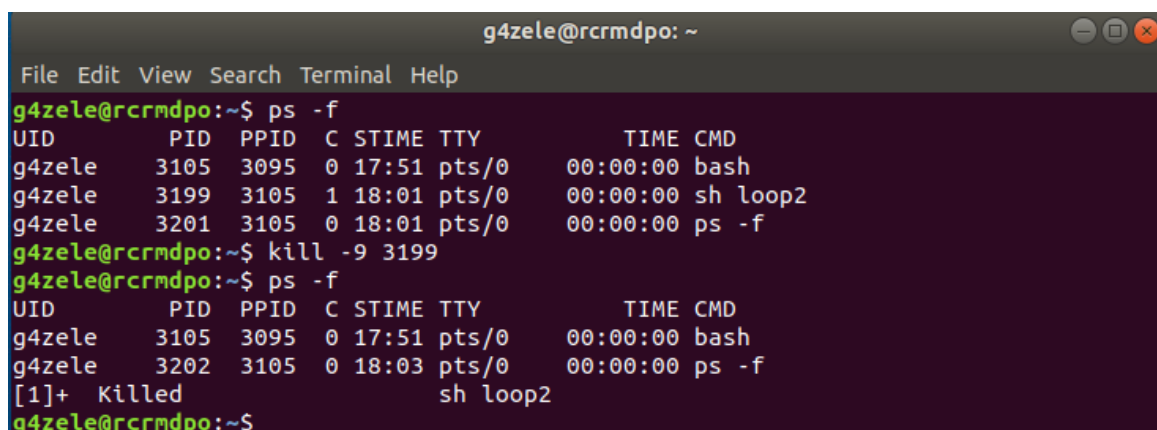
Рисунок 7 – Запуск сценария на исполнение и остановка процесса

```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele   3105  3095  0  17:51 pts/0      00:00:00 bash  
g4zele   3199  3105  2  18:01 pts/0      00:00:00 sh loop2  
g4zele   3200  3105  0  18:01 pts/0      00:00:00 ps -f  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele   3105  3095  0  17:51 pts/0      00:00:00 bash  
g4zele   3199  3105  1  18:01 pts/0      00:00:00 sh loop2  
g4zele   3201  3105  0  18:01 pts/0      00:00:00 ps -f  
g4zele@rcrmdpo:~$
```

Рисунок 8 – Два листинга процесса подряд

Отсюда видим, что ресурс процессора, затраченный на процесс loop2, уменьшается, что говорит о приостановке процессора.

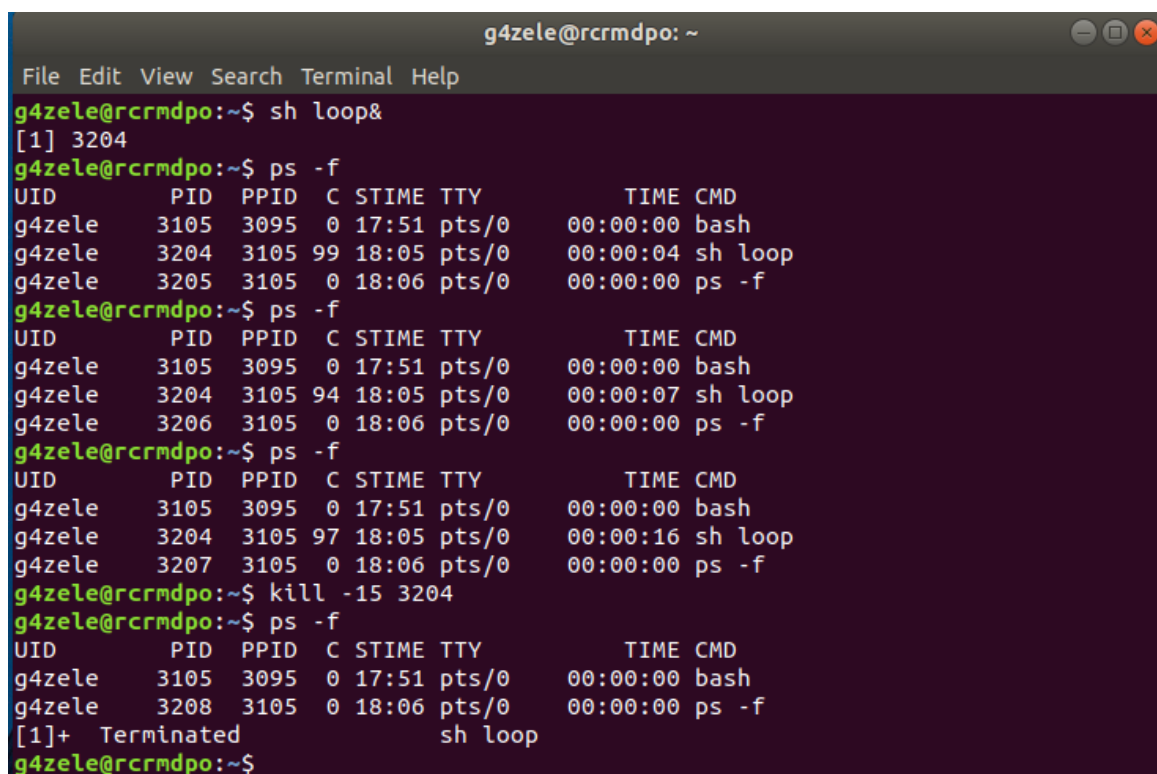
Теперь «убьём» процесс с помощью команды `kill -9` и в качестве аргумента передадим идентификатор процесса (PID):



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele    3105  3095  0  17:51 pts/0        00:00:00 bash  
g4zele    3199  3105  1  18:01 pts/0        00:00:00 sh loop2  
g4zele    3201  3105  0  18:01 pts/0        00:00:00 ps -f  
g4zele@rcrmdpo:~$ kill -9 3199  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele    3105  3095  0  17:51 pts/0        00:00:00 bash  
g4zele    3202  3105  0  18:03 pts/0        00:00:00 ps -f  
[1]+  Killed                  sh loop2  
g4zele@rcrmdpo:~$
```

Рисунок 9 – Уничтожение процесса `loop2.sh`

Теперь запустим сценарий `loop.sh` на исполнение в фоновом режиме с помощью команды `sh loop&`:

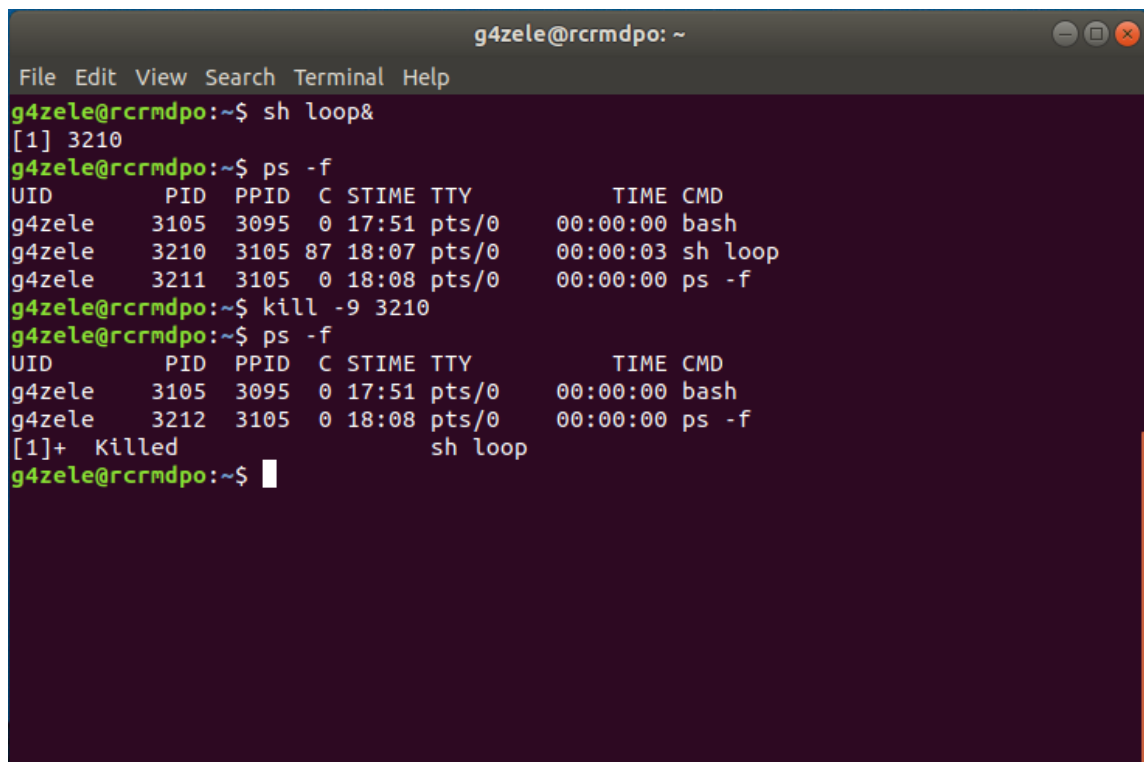


```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ sh loop&  
[1] 3204  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele    3105  3095  0  17:51 pts/0        00:00:00 bash  
g4zele    3204  3105  99  18:05 pts/0        00:00:04 sh loop  
g4zele    3205  3105  0  18:06 pts/0        00:00:00 ps -f  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele    3105  3095  0  17:51 pts/0        00:00:00 bash  
g4zele    3204  3105  94  18:05 pts/0        00:00:07 sh loop  
g4zele    3206  3105  0  18:06 pts/0        00:00:00 ps -f  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele    3105  3095  0  17:51 pts/0        00:00:00 bash  
g4zele    3204  3105  97  18:05 pts/0        00:00:16 sh loop  
g4zele    3207  3105  0  18:06 pts/0        00:00:00 ps -f  
g4zele@rcrmdpo:~$ kill -15 3204  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
g4zele    3105  3095  0  17:51 pts/0        00:00:00 bash  
g4zele    3208  3105  0  18:06 pts/0        00:00:00 ps -f  
[1]+  Terminated             sh loop  
g4zele@rcrmdpo:~$
```

Рисунок 10 – Запуск процесса `loop.sh` в фоновом режиме

По рисунку 10 видно, что ресурс процессора, затраченный на процесс, не уменьшается, что говорит о том, что процесс запущен.

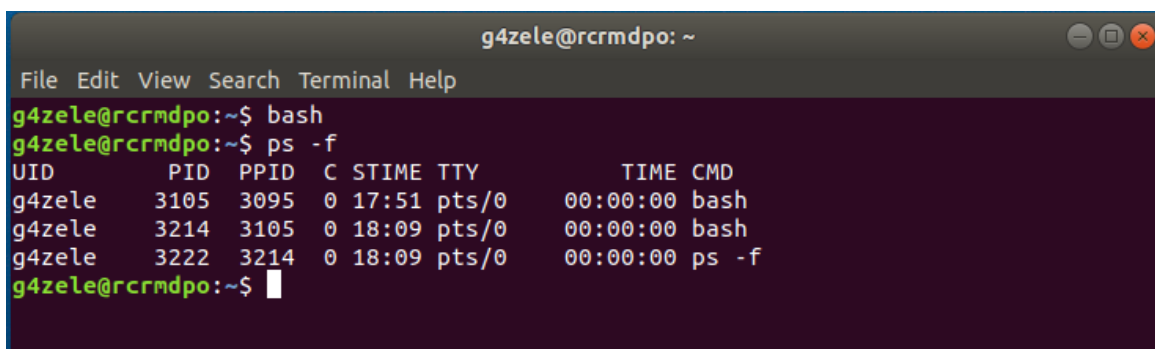
Теперь запустим процесс loop.sh снова и «убьём» его командой kill -9:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ sh loop&  
[1] 3210  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
g4zele   3105  3095  0  17:51 pts/0    00:00:00 bash  
g4zele   3210  3105  87  18:07 pts/0    00:00:03 sh loop  
g4zele   3211  3105  0  18:08 pts/0    00:00:00 ps -f  
g4zele@rcrmdpo:~$ kill -9 3210  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
g4zele   3105  3095  0  17:51 pts/0    00:00:00 bash  
g4zele   3212  3105  0  18:08 pts/0    00:00:00 ps -f  
[1]+  Killed                  sh loop  
g4zele@rcrmdpo:~$
```

Рисунок 11 – Запуск и убийство процесса loop.sh

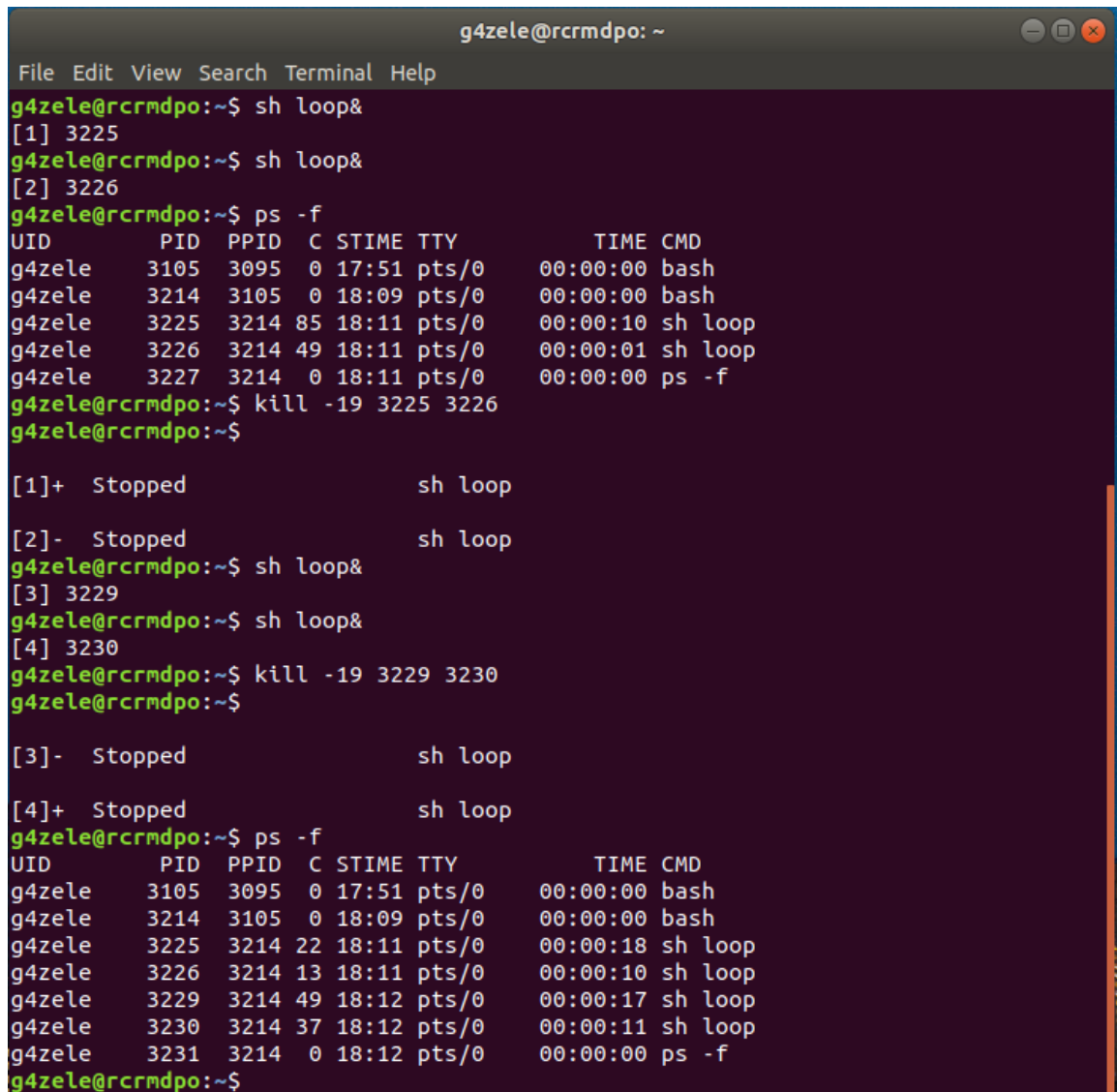
Следующим заданием будет запуск ещё одного экземпляра оболочки с помощью команды bash:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ bash  
g4zele@rcrmdpo:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
g4zele   3105  3095  0  17:51 pts/0    00:00:00 bash  
g4zele   3214  3105  0  18:09 pts/0    00:00:00 bash  
g4zele   3222  3214  0  18:09 pts/0    00:00:00 ps -f  
g4zele@rcrmdpo:~$
```

Рисунок 12 – Запуск экземпляра оболочки

Теперь запустим несколько процессов в фоне. Будем их останавливать и снова запускать:



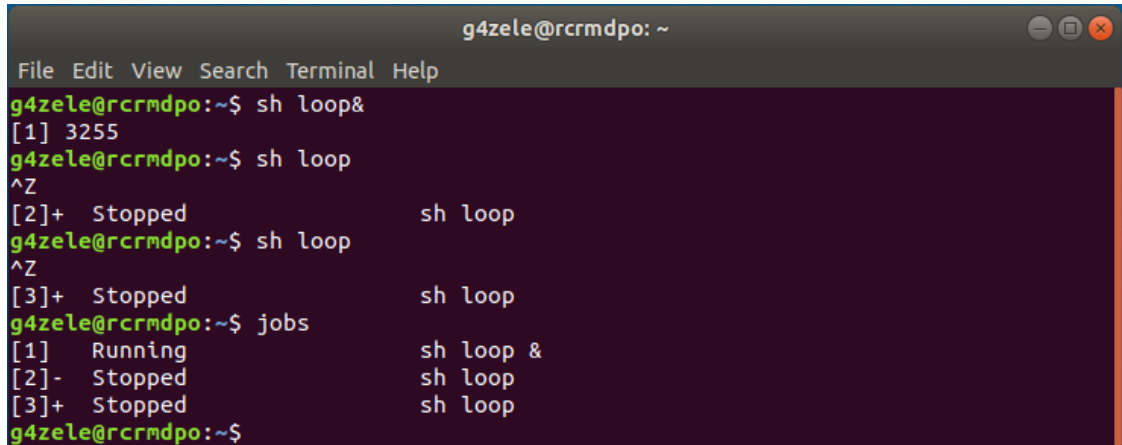
```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ sh loop&  
[1] 3225  
g4zele@rcrmdpo:~$ sh loop&  
[2] 3226  
g4zele@rcrmdpo:~$ ps -f  
UID          PID    PPID  C  STIME TTY          TIME CMD  
g4zele       3105    3095  0  17:51 pts/0        00:00:00 bash  
g4zele       3214    3105  0  18:09 pts/0        00:00:00 bash  
g4zele       3225    3214  85  18:11 pts/0        00:00:10 sh loop  
g4zele       3226    3214  49  18:11 pts/0        00:00:01 sh loop  
g4zele       3227    3214  0  18:11 pts/0        00:00:00 ps -f  
g4zele@rcrmdpo:~$ kill -19 3225 3226  
g4zele@rcrmdpo:~$  
[1]+  Stopped                  sh loop  
[2]-  Stopped                  sh loop  
g4zele@rcrmdpo:~$ sh loop&  
[3] 3229  
g4zele@rcrmdpo:~$ sh loop&  
[4] 3230  
g4zele@rcrmdpo:~$ kill -19 3229 3230  
g4zele@rcrmdpo:~$  
[3]-  Stopped                  sh loop  
[4]+  Stopped                  sh loop  
g4zele@rcrmdpo:~$ ps -f  
UID          PID    PPID  C  STIME TTY          TIME CMD  
g4zele       3105    3095  0  17:51 pts/0        00:00:00 bash  
g4zele       3214    3105  0  18:09 pts/0        00:00:00 bash  
g4zele       3225    3214  22  18:11 pts/0        00:00:18 sh loop  
g4zele       3226    3214  13  18:11 pts/0        00:00:10 sh loop  
g4zele       3229    3214  49  18:12 pts/0        00:00:17 sh loop  
g4zele       3230    3214  37  18:12 pts/0        00:00:11 sh loop  
g4zele       3231    3214  0  18:12 pts/0        00:00:00 ps -f  
g4zele@rcrmdpo:~$
```

Рисунок 13 – Запуск и остановка процессов несколько раз

На этом задания I части выполнены. Перейдём ко II части.

Часть II

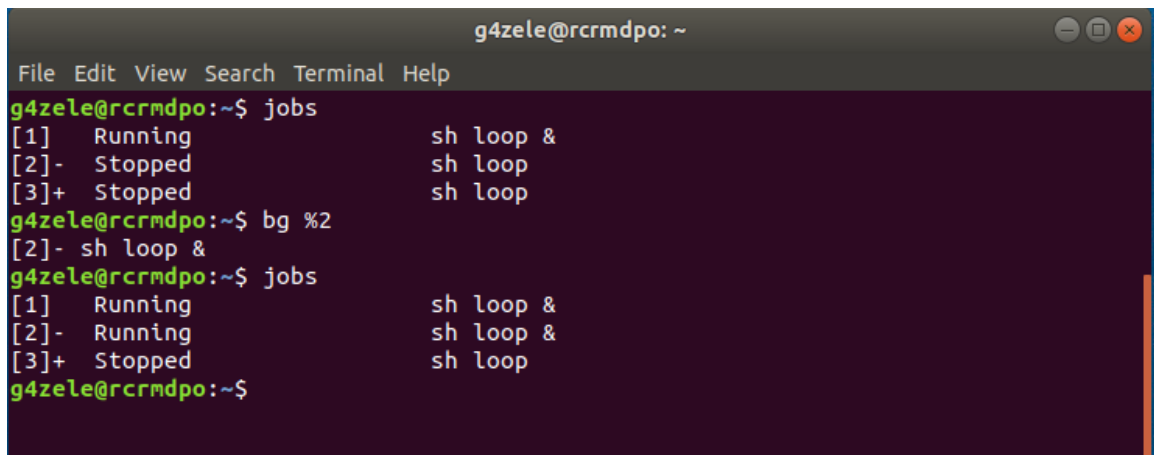
Первым заданием II части будет запуск трёх задач: двух в интерактивном режиме и одной – в фоновом. Затем выведем процессы с помощью команды `jobs`:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ sh loop&  
[1] 3255  
g4zele@rcrmdpo:~$ sh loop  
^Z  
[2]+  Stopped                  sh loop  
g4zele@rcrmdpo:~$ sh loop  
^Z  
[3]+  Stopped                  sh loop  
g4zele@rcrmdpo:~$ jobs  
[1]  Running                  sh loop &  
[2]-  Stopped                 sh loop  
[3]+  Stopped                 sh loop  
g4zele@rcrmdpo:~$
```

Рисунок 14 – Запуск трёх задач

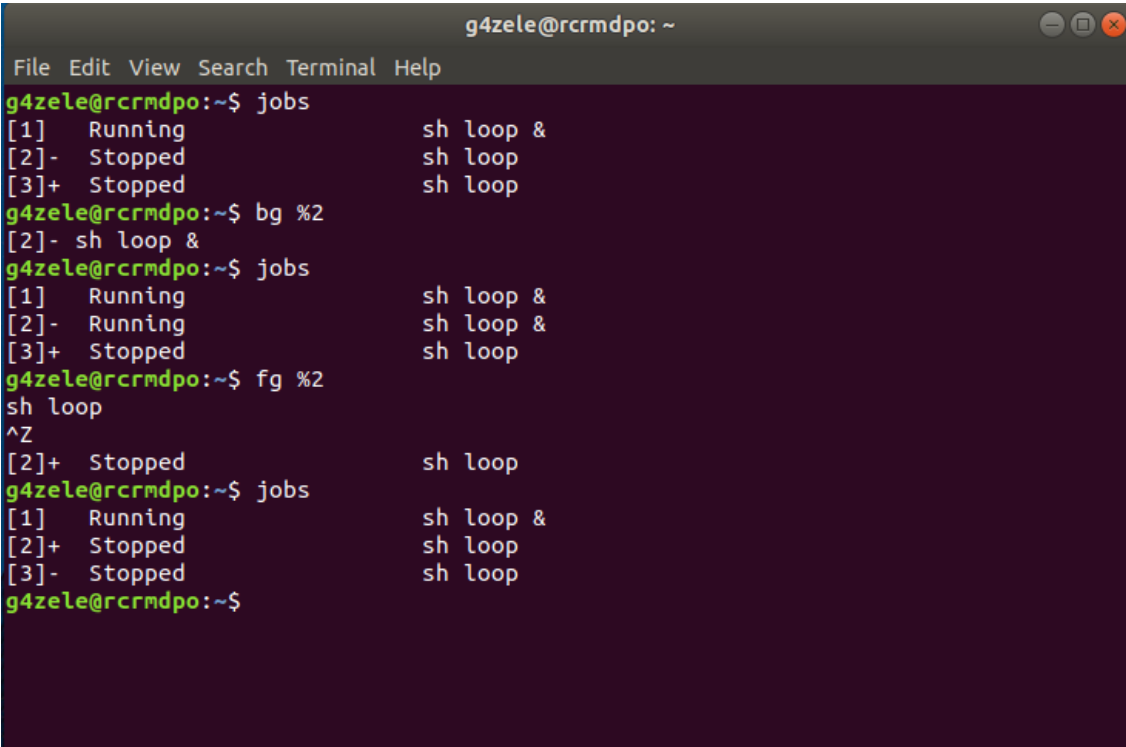
Теперь переведём одну из задач, выполняющихся в интерактивном режиме, в фоновый режим с помощью команды `bg`:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ jobs  
[1]  Running                  sh loop &  
[2]-  Stopped                 sh loop  
[3]+  Stopped                 sh loop  
g4zele@rcrmdpo:~$ bg %2  
[2]-  sh loop &  
g4zele@rcrmdpo:~$ jobs  
[1]  Running                  sh loop &  
[2]-  Running                 sh loop &  
[3]+  Stopped                 sh loop  
g4zele@rcrmdpo:~$
```

Рисунок 15 – Перевод процесса в фоновый режим

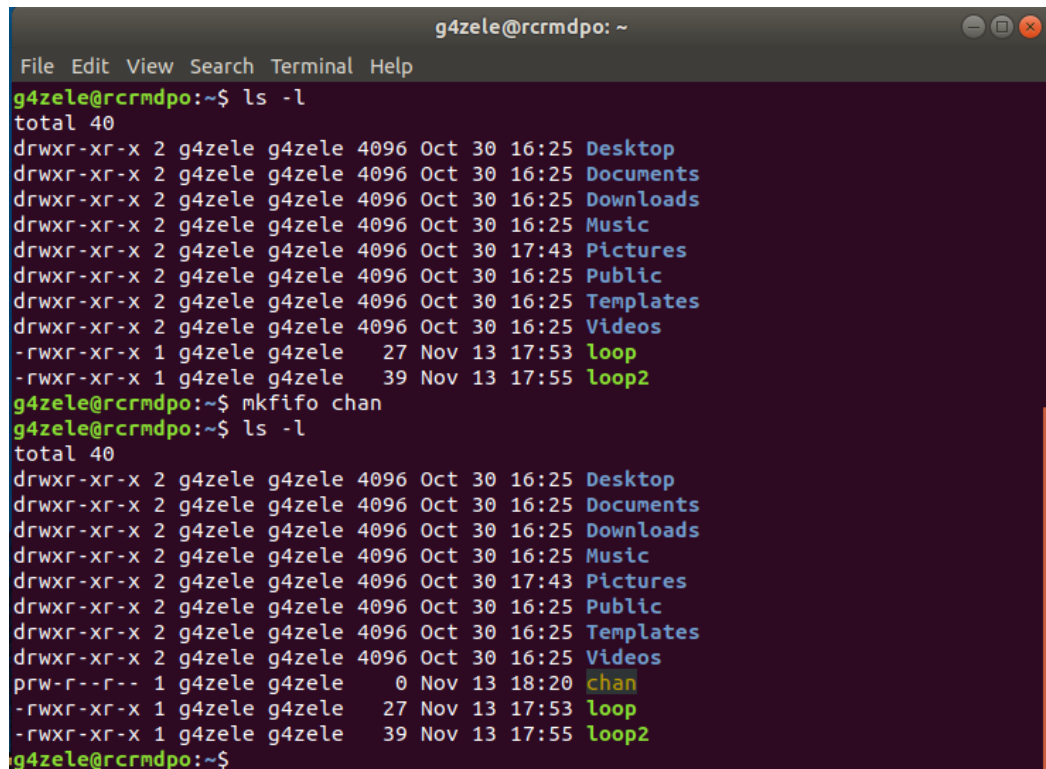
Попробуем переводить процессы в фоновый режим из интерактивного и наоборот – из фонового в интерактивный:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ jobs  
[1]  Running                sh loop &  
[2]-  Stopped                sh loop  
[3]+  Stopped                sh loop  
g4zele@rcrmdpo:~$ bg %2  
[2]-  sh loop &  
g4zele@rcrmdpo:~$ jobs  
[1]  Running                sh loop &  
[2]-  Running                sh loop &  
[3]+  Stopped                sh loop  
g4zele@rcrmdpo:~$ fg %2  
sh loop  
^Z  
[2]+  Stopped                sh loop  
g4zele@rcrmdpo:~$ jobs  
[1]  Running                sh loop &  
[2]+  Stopped                sh loop  
[3]-  Stopped                sh loop  
g4zele@rcrmdpo:~$
```

Рисунок 16 – Перевод задачи из фонового режима в интерактивный и наоборот

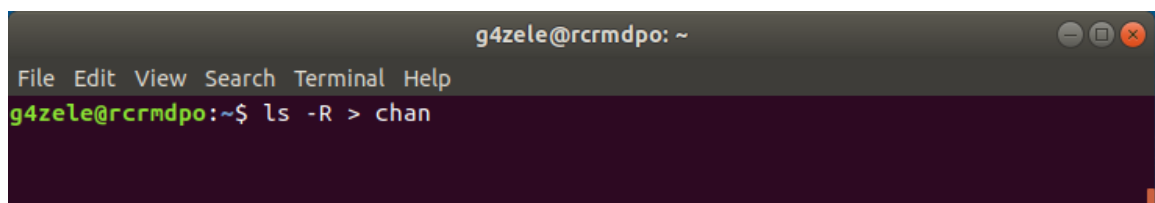
Теперь создадим именованный канал для архивирования с помощью команды `mkfifo`:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ ls -l  
total 40  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Desktop  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Documents  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Downloads  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Music  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 17:43 Pictures  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Public  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Templates  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Videos  
-rwxr-xr-x 1 g4zele g4zele 27 Nov 13 17:53 loop  
-rwxr-xr-x 1 g4zele g4zele 39 Nov 13 17:55 loop2  
g4zele@rcrmdpo:~$ mkfifo chan  
g4zele@rcrmdpo:~$ ls -l  
total 40  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Desktop  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Documents  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Downloads  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Music  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 17:43 Pictures  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Public  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Templates  
drwxr-xr-x 2 g4zele g4zele 4096 Oct 30 16:25 Videos  
prw-r--r-- 1 g4zele g4zele 0 Nov 13 18:20 chan  
-rwxr-xr-x 1 g4zele g4zele 27 Nov 13 17:53 loop  
-rwxr-xr-x 1 g4zele g4zele 39 Nov 13 17:55 loop2  
g4zele@rcrmdpo:~$
```

Рисунок 17 – Создание именованного канала

Затем занесём в канал листинг домашнего каталога:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ ls -R > chan
```

Рисунок 18 – Заносим в канал листинг домашнего каталога

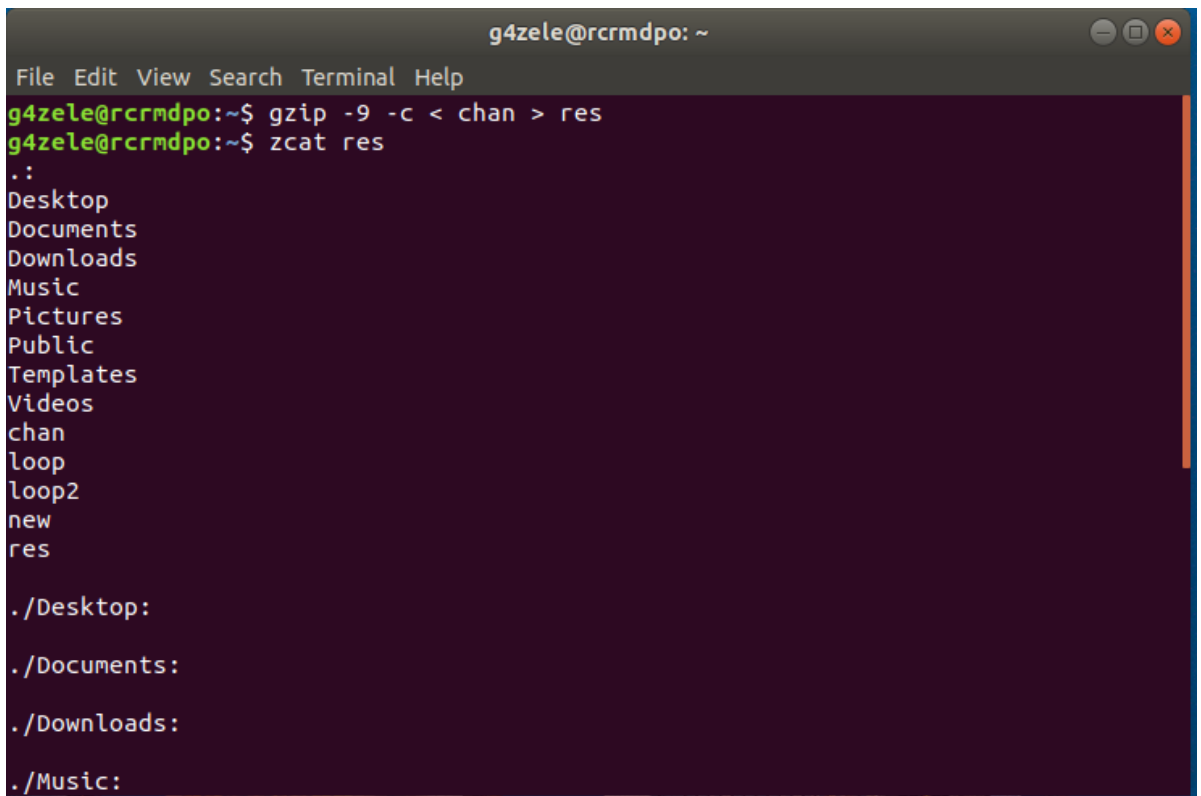
Создадим директорию:

A terminal window titled 'g4zele@rcrmdpo: ~/new' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
g4zele@rcrmdpo:~$ mkdir new
g4zele@rcrmdpo:~$ cd new
g4zele@rcrmdpo:~/new$ touch 1.txt
g4zele@rcrmdpo:~/new$ touch 2.txt
g4zele@rcrmdpo:~/new$ mkdir new1
g4zele@rcrmdpo:~/new$ cd new1
g4zele@rcrmdpo:~/new/new1$ touch 3.txt
g4zele@rcrmdpo:~/new/new1$ cd ..
g4zele@rcrmdpo:~/new$ ls -l
total 4
-rw-r--r-- 1 g4zele g4zele  0 Nov 13 18:23 1.txt
-rw-r--r-- 1 g4zele g4zele  0 Nov 13 18:24 2.txt
drwxr-xr-x 2 g4zele g4zele 4096 Nov 13 18:24 new1
g4zele@rcrmdpo:~/new$
```

Рисунок 19 – Создание нового каталога

Откроем листинг домашнего каталога:

A terminal window titled 'g4zele@rcrmdpo: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
g4zele@rcrmdpo:~$ gzip -9 -c < chan > res
g4zele@rcrmdpo:~$ zcat res
.:
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
chan
loop
loop2
new
res

./Desktop:

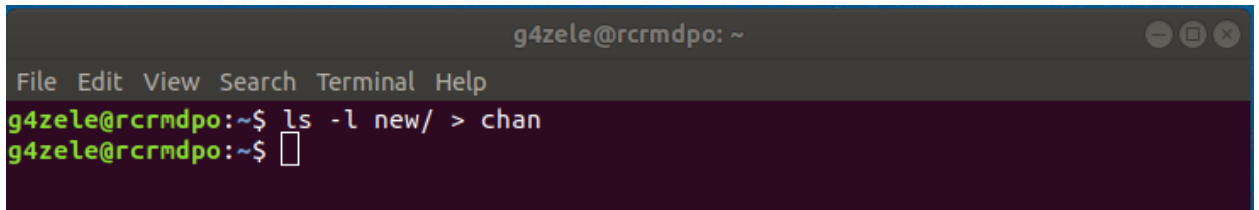
./Documents:

./Downloads:

./Music:
```

Рисунок 20 – Листинг домашнего каталога

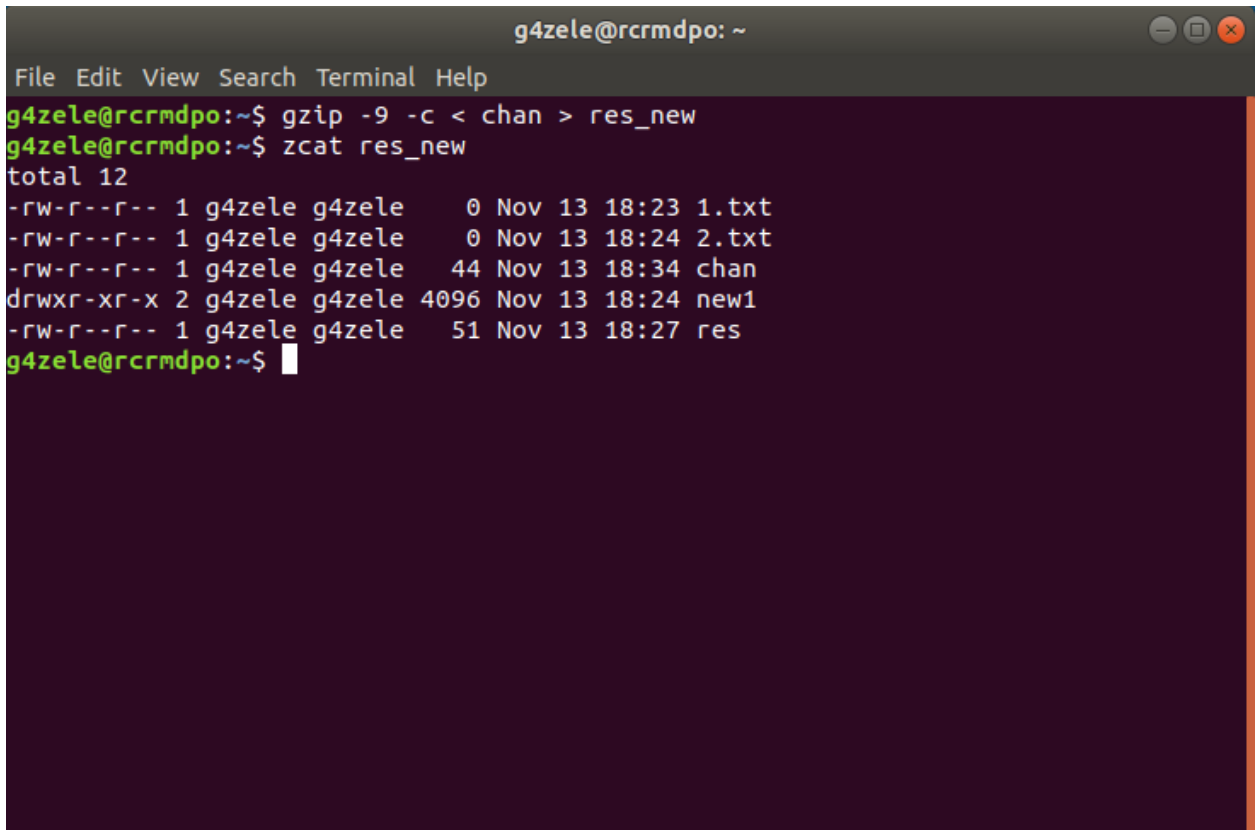
Теперь занесём в канал листинг созданного каталога:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ ls -l new/ > chan  
g4zele@rcrmdpo:~$
```

Рисунок 21 – Заносим в канал листинг созданного каталога

Теперь архивируем содержимое каталога и откроем архив:



```
g4zele@rcrmdpo: ~  
File Edit View Search Terminal Help  
g4zele@rcrmdpo:~$ gzip -9 -c < chan > res_new  
g4zele@rcrmdpo:~$ zcat res_new  
total 12  
-rw-r--r-- 1 g4zele g4zele  0 Nov 13 18:23 1.txt  
-rw-r--r-- 1 g4zele g4zele  0 Nov 13 18:24 2.txt  
-rw-r--r-- 1 g4zele g4zele 44 Nov 13 18:34 chan  
drwxr-xr-x 2 g4zele g4zele 4096 Nov 13 18:24 new1  
-rw-r--r-- 1 g4zele g4zele  51 Nov 13 18:27 res  
g4zele@rcrmdpo:~$
```

Рисунок 22 – Листинг созданного каталога

Это было последнее задание II части лабораторной работы. Переходим к выполнению заданий III части лабораторной работы.

Часть III

Для того чтобы вывести информацию о процессах воспользуемся утилитой “Тор”

```
g4zele@rcrmdpo: ~
File Edit View Search Terminal Help
top - 19:14:00 up 1:34, 1 user, load average: 0.00, 0.02, 0.05
Tasks: 208 total, 2 running, 164 sleeping, 1 stopped, 0 zombie
%Cpu(s): 3.6 us, 4.6 sy, 0.0 ni, 91.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2041492 total, 191768 free, 1064276 used, 785448 buff/cache
KiB Swap: 2097148 total, 2096880 free, 268 used. 787516 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 2502 g4zele    20   0 3028616 278636 113312 S   1.7  13.6   1:21.64 gnome-shell
 1836 gdm       20   0 2933544 210416  91136 S   0.0  10.3   0:07.35 gnome-shell
 2891 g4zele    20   0 1069352 159676  35016 S   0.0   7.8   0:04.61 gnome-softw+
2366 g4zele    20   0 504412 118152  62040 R   2.3   5.8   0:36.80 Xorg
 1461 root      20   0  419640  79812  44108 S   0.0   3.9   0:09.06 dockerd
 2772 g4zele    20   0  892196  66332  26312 S   0.0   3.2   0:00.51 evolution-c+
 2743 g4zele    20   0  896736  61760  47932 S   0.0   3.0   0:03.63 nautilus-de+
 2818 g4zele    20   0  939288  61564  22380 S   0.0   3.0   0:00.48 evolution-c+
 2120 gdm       20   0  488208  50036  37864 S   0.0   2.5   0:00.16 ibus-x11
 1875 root      20   0  430644  41656  23312 S   0.0   2.0   0:04.89 containerd
 1883 gdm       20   0  227552  38172  27196 S   0.0   1.9   0:00.09 Xwayland
 3379 g4zele    20   0  859164  36936  27136 S   0.7   1.8   0:05.26 gnome-termi+
 3010 g4zele    20   0  862708  33044  26952 S   0.0   1.6   0:00.10 deja-dup-mo+
 2589 g4zele    20   0  775496  32400  26600 S   0.0   1.6   0:00.10 goa-daemon
 1232 root      20   0  683028  27924  15108 S   0.0   1.4   0:01.39 snapd
 2847 g4zele    20   0  877840  25552  22084 S   0.0   1.3   0:00.05 evolution-a+
 2576 g4zele    20   0  970648  24824  20780 S   0.0   1.2   0:00.12 evolution-s+
```

Рисунок 23 – Работа утилиты Тор.

Для сортировки процессов по имени пользователя используем интерактивную программу [u] + имя пользователя для сортировки.

```
g4zele@rcrmdpo: ~
File Edit View Search Terminal Help
top - 19:16:34 up 1:36, 1 user, load average: 0.00, 0.00, 0.04
Tasks: 208 total, 1 running, 165 sleeping, 1 stopped, 0 zombie
%Cpu(s): 1.7 us, 0.7 sy, 0.0 ni, 97.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2041492 total, 191768 free, 1064276 used, 785448 buff/cache
KiB Swap: 2097148 total, 2096880 free, 268 used. 787516 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 2502 g4zele    20   0 3028616 278568 113312 S   1.7  13.6   1:22.64 gnome-shell
 2891 g4zele    20   0 1069352 159676  35016 S   0.0   7.8   0:04.62 gnome-softw+
2366 g4zele    20   0 504412 118152  62040 S   1.7   5.8   0:37.41 Xorg
 2772 g4zele    20   0  892196  66332  26312 S   0.0   3.2   0:00.51 evolution-c+
 2743 g4zele    20   0  896736  61760  47932 S   0.0   3.0   0:03.68 nautilus-de+
 2818 g4zele    20   0  939288  61564  22380 S   0.0   3.0   0:00.48 evolution-c+
 3379 g4zele    20   0  859164  36936  27136 S   0.7   1.8   0:05.57 gnome-termi+
 3010 g4zele    20   0  862708  33044  26952 S   0.0   1.6   0:00.10 deja-dup-mo+
 2589 g4zele    20   0  775496  32400  26600 S   0.0   1.6   0:00.10 goa-daemon
 2847 g4zele    20   0  877840  25552  22084 S   0.0   1.3   0:00.05 evolution-a+
 2576 g4zele    20   0  970648  24824  20780 S   0.0   1.2   0:00.12 evolution-s+
 2893 g4zele    20   0  585708  24548  19548 S   0.0   1.2   0:00.25 update-noti+
 2663 g4zele    20   0  858252  23128  17936 S   0.0   1.1   0:00.20 gsd-media-k+
 2838 g4zele    20   0  724532  22804  19412 S   0.0   1.1   0:00.05 evolution-a+
 2652 g4zele    20   0  659784  22300  17328 S   0.0   1.1   0:00.88 gsd-color
 2619 g4zele    20   0  508892  21960  16808 S   0.0   1.1   0:00.16 gsd-power
 2631 g4zele    20   0  486204  21812  16628 S   0.0   1.1   0:00.18 gsd-xsettin+
```

Рисунок 24 – Отсортированные процессы.

Для изменения времени обновления информации вызовем интерактивную команду с помощью кнопки d (delay) и поставим вместо трёх секунд одну.

```

g4zele@rcrmdpo: ~
File Edit View Search Terminal Help
top - 19:18:47 up 1:38, 1 user, load average: 0.00, 0.00, 0.02
Tasks: 208 total, 1 running, 165 sleeping, 1 stopped, 0 zombie
%Cpu(s): 0.7 us, 0.3 sy, 0.0 ni, 99.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2041492 total, 191388 free, 1064580 used, 785524 buff/cache
KiB Swap: 2097148 total, 2096880 free, 268 used. 787208 avail Mem
Change delay from 3.0 to 1

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2502	g4zele	20	0	3028616	278544	113312	S	0.3	13.6	1:22.99	gnome-shell
2891	g4zele	20	0	1069352	159676	35016	S	0.0	7.8	0:04.62	gnome-softw+
2366	g4zele	20	0	504412	118152	62040	S	0.0	5.8	0:37.59	Xorg
2772	g4zele	20	0	892196	66332	26312	S	0.0	3.2	0:00.51	evolution-c+
2743	g4zele	20	0	896736	61760	47932	S	0.0	3.0	0:03.72	nautilus-de+
2818	g4zele	20	0	939288	61564	22380	S	0.0	3.0	0:00.48	evolution-c+
3379	g4zele	20	0	859164	36936	27136	S	0.3	1.8	0:05.68	gnome-termi+
3010	g4zele	20	0	862708	33044	26952	S	0.0	1.6	0:00.10	deja-dup-mo+
2589	g4zele	20	0	775496	32400	26600	S	0.0	1.6	0:00.10	goa-daemon
2847	g4zele	20	0	877840	25552	22084	S	0.0	1.3	0:00.05	evolution-a+
2576	g4zele	20	0	970648	24824	20780	S	0.0	1.2	0:00.12	evolution-s+
2893	g4zele	20	0	585708	24548	19548	S	0.0	1.2	0:00.25	update-noti+
2663	g4zele	20	0	858252	23128	17936	S	0.0	1.1	0:00.20	gsd-media-k+
2838	g4zele	20	0	724532	22804	19412	S	0.0	1.1	0:00.05	evolution-a+
2652	g4zele	20	0	659784	22300	17328	S	0.0	1.1	0:00.89	gsd-color
2619	g4zele	20	0	508892	21960	16808	S	0.0	1.1	0:00.16	gsd-power
2631	g4zele	20	0	486204	21812	16628	S	0.0	1.1	0:00.18	gsd-xsettin+

Рисунок 25 – Изменение времени обновления.

Для отключения процесса по имени с помощью сигнала SIGINT воспользуемся командой pkill -SIGINT “Имя процесса”

```

g4zele@rcrmdpo: ~
File Edit View Search Terminal Help
g4zele@rcrmdpo:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
g4zele    3645   3636  0  19:54 pts/0        00:00:00 bash
g4zele    3657   3645  0  19:55 pts/0        00:00:00 ps -f
g4zele@rcrmdpo:~$ sh loop&
[1] 3658
g4zele@rcrmdpo:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
g4zele    3645   3636  0  19:54 pts/0        00:00:00 bash
g4zele    3658   3645  99  19:55 pts/0        00:00:01 sh loop
g4zele    3659   3645  0  19:55 pts/0        00:00:00 ps -f
g4zele@rcrmdpo:~$ pkill -SIGINT sh
pkill: killing pid 208 failed: Operation not permitted
pkill: killing pid 1687 failed: Operation not permitted
pkill: killing pid 2076 failed: Operation not permitted

g4zele@rcrmdpo:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
g4zele    3645   3636  0  19:54 pts/0        00:00:00 bash
g4zele    3787   3645  0  19:55 pts/0        00:00:00 ps -f
[1]+  Interrupt                  sh loop
g4zele@rcrmdpo:~$

```

Рисунок 26 – Создание процесса и использование команды `ps` с сигналом `SIGINT`.

Откроем файл `1.txt` на редактирование думя редакторами «Nano» после чего используем команду `ls -l`

```
g4zele@rcrmdpo:~$ ls -l new
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE NAME
bash     3921 g4zele  cwd  DIR   253,0    4096  919464 new
bash     3936 g4zele  cwd  DIR   253,0    4096  919464 new
nano     4065 g4zele  cwd  DIR   253,0    4096  919464 new
nano     4066 g4zele  cwd  DIR   253,0    4096  919464 new
```

Рисунок 27 – Использование команды `ps` с сигналом `SIGINT`.

Мы можем видеть 2 процесса от команды `nano` с PID 4065 и 4066.

Процессы необходимо закрыть с сигналом `Terminate (SIGTERM)` Так что запишем команду `killall -TERM --interactive nano`

```
g4zele@rcrmdpo:~$ killall -TERM --interactive nano
Kill nano(4065) ? (y/N) y
Kill nano(4066) ? (y/N) y
g4zele@rcrmdpo:~$
```

Рисунок 27 – `killall -TERM --interactive nano`.

Вывод

В ходе выполнения лабораторной работы я ознакомился на практике с понятием процесса в операционной системе и приобрёл опыт и навыки управления процессами в операционной системе Linux.