

# Липецкий государственный технический университет

Факультет автоматизации и информатики  
Кафедра автоматизированных систем управления

Отчет по лабораторной работе № 2  
по дисциплине «OS Linux»  
на тему «Процессы в операционной системе Linux»

Студент

Группа АС-18-1

Руководитель

К.Н.

учёная степень, учёное звание

подпись, дата

подпись, дата

Сухоруков К.О.

фамилия, инициалы

Кургасов В.В.

фамилия, инициалы

Липецк 2020 г.

## СОДЕРЖАНИЕ

Цель работы . . . . .	2
1 Ход выполнения . . . . .	3
1.1 Часть 1 . . . . .	3
1.2 Часть 2 . . . . .	12
1.3 Часть 3 (вариант 10) . . . . .	16
Вывод . . . . .	21

## Цель работы

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

## 1 Ход выполнения

### 1.1 Часть 1

На рисунке 1.1 продемонстрирована загрузка в режиме обычного пользователя.

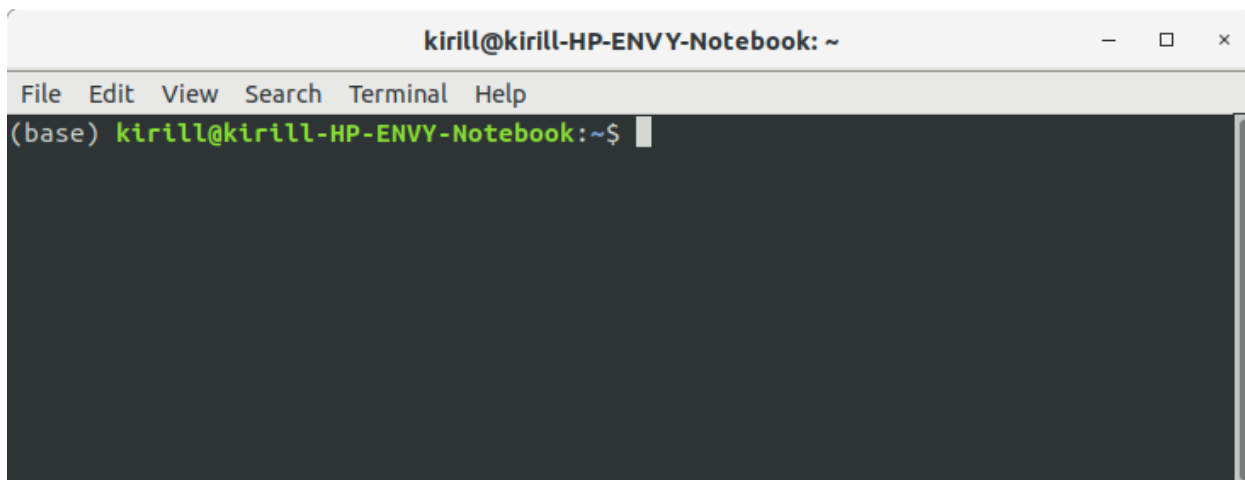


Рисунок 1.1 – Загрузка в режиме пользователя

На рисунке 1.2 произведен переход в корневую директорию и показана информация о файле ядра.

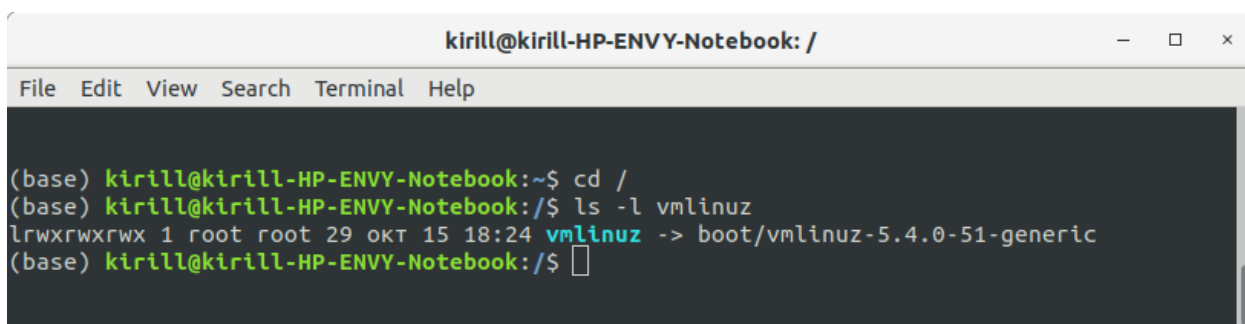


Рисунок 1.2 – Файл ядра Linux

По названию файла ядра с рисунка 1.2 видно, что версия ядра – 5.4.0.51

На рисунке 1.3 показан вывод всех текущих процессов Linux в текущей оболочке с помощью утилиты ps. Добавление опции -f дает возможность вывести максимум доступных данных.

```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill   4518  3309  0  18:44 pts/2    00:00:00 bash  
kirill   4577  4518  0  18:44 pts/2    00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.3 – Процессы в текущей оболочке

На рисунке 1.3 мы видим два запущенных процесса, проанализируем полученную информацию, используя файл справки:

1. UID – пользователь, от имени которого запущен процесс;
2. PID – идентификатор процесса;
3. PPID – идентификатор родительского процесса;
4. C – процент времени CPU, используемого процессом;
5. STIME – время запуска процесса;
6. TTY – терминал, из которого запущен процесс;
7. TIME – общее время процессора, затраченное на выполнение процессора;
8. CMD – команда запуска процессора;

Создадим два сценария loop и loop2 с помощью редактора vim, процесс создания продемонстрирован на рисунках 1.4 и 1.5.

```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
1 while true; do true; done  
NORMAL [ +]  
:w loop.sh
```

Рисунок 1.4 – Создание сценария loop

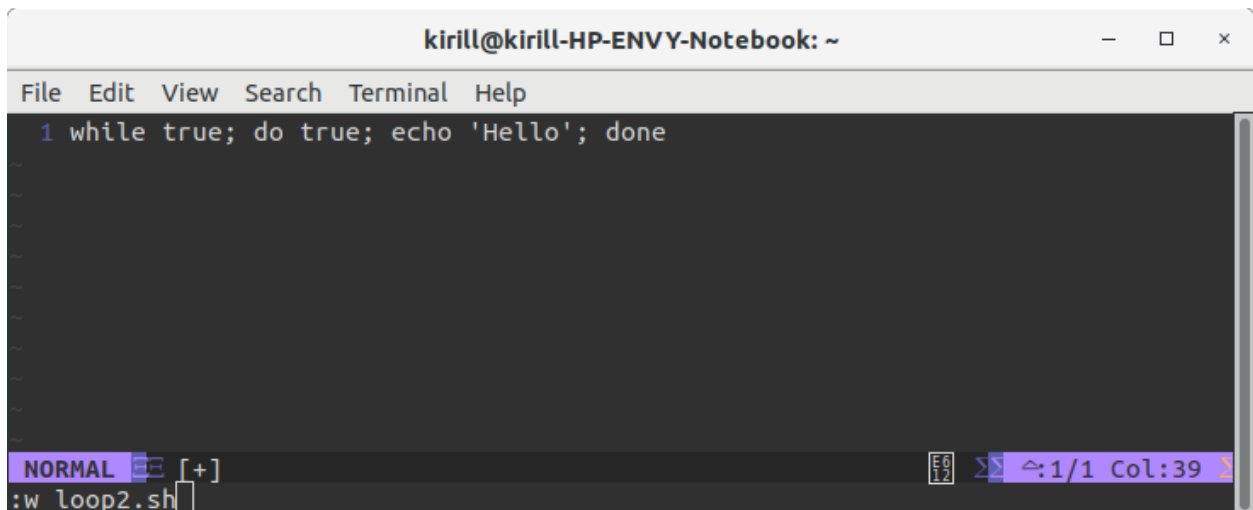


Рисунок 1.5 – Создание сценария loop2

На рисунке 1.6 показан результат создания сценариев.

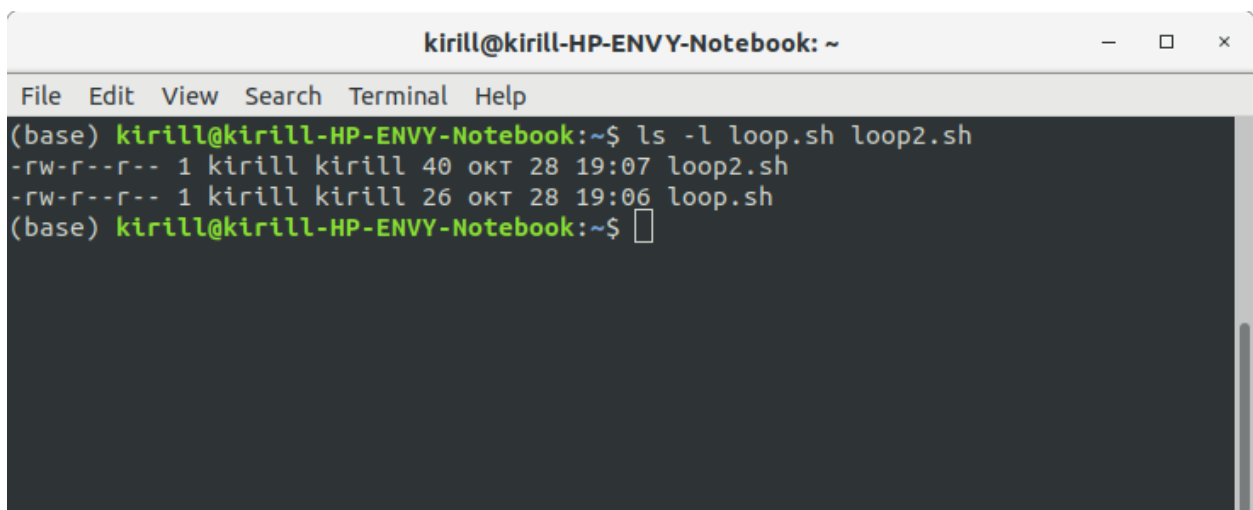


Рисунок 1.6 – Результат создания сценариев

Как видно из рисунка 1.6 файлы созданы, но для них не указаны права на исполнение. Добавим эти права с помощью команды `chmod`. Результат представлен на рисунке 1.7.

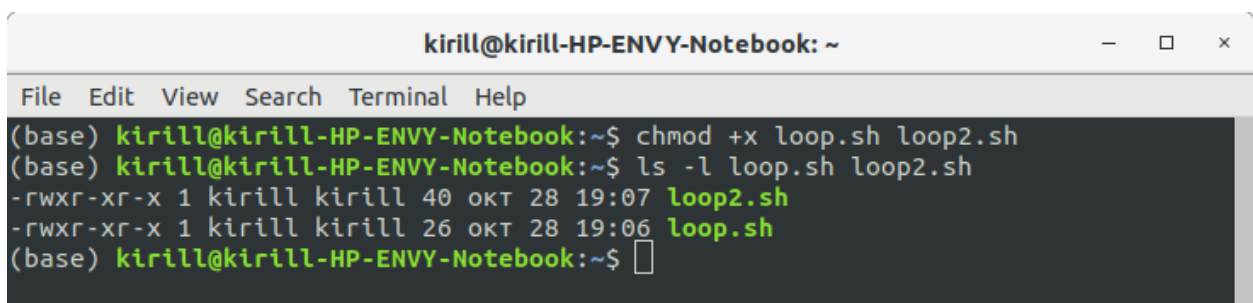
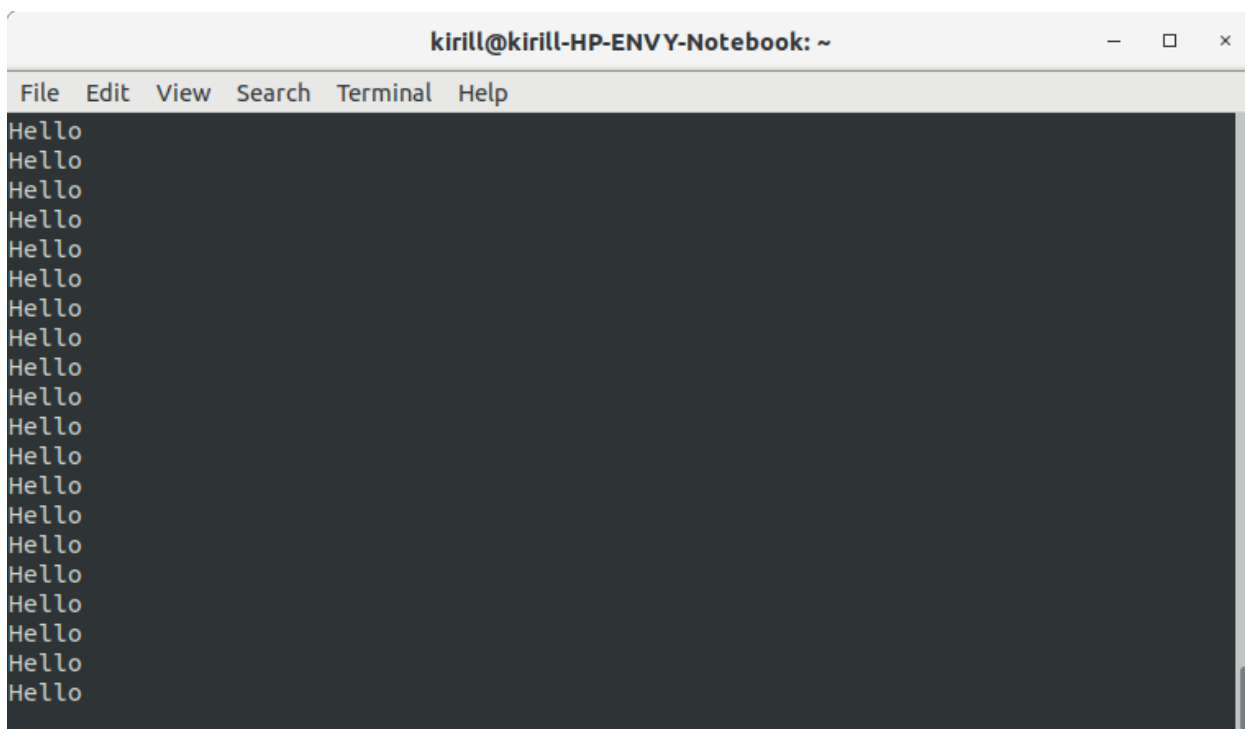


Рисунок 1.7 – Добавление прав на исполнение

Теперь оба процесса можно запустить на исполнение.

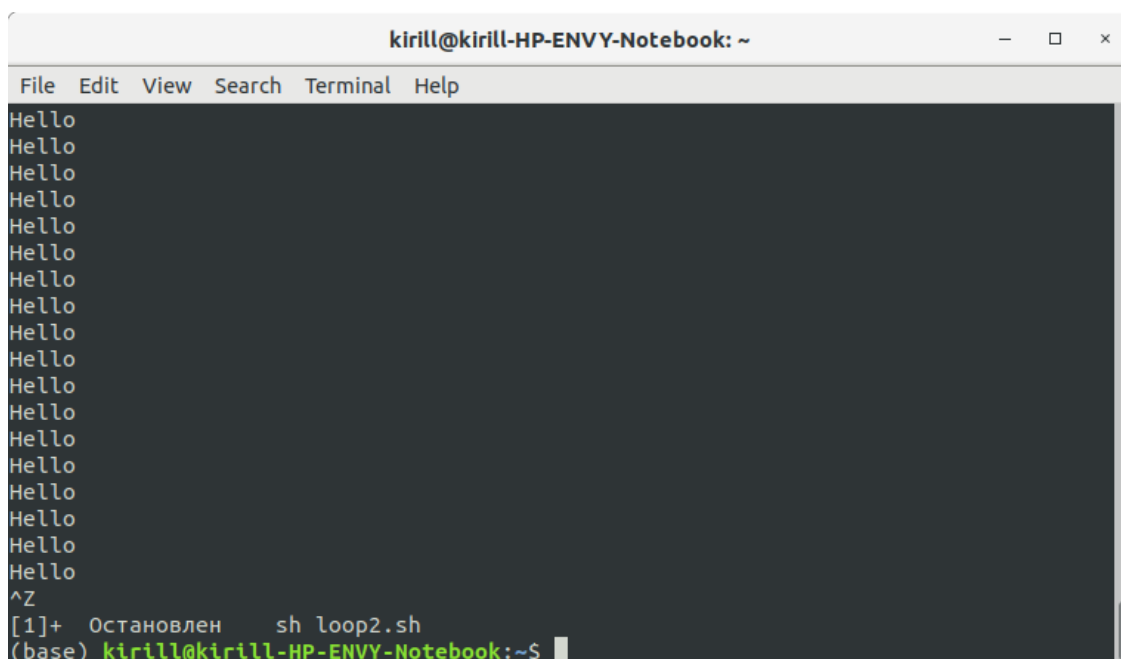
Запустим процесс `loop2` с помощью команды `sh`. результат запуска продемонстрирован на рисунке 1.8.



```
kirill@kirill-HP-ENVY-Notebook: ~  
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
```

Рисунок 1.8 – Запущенный процесс `loop2`

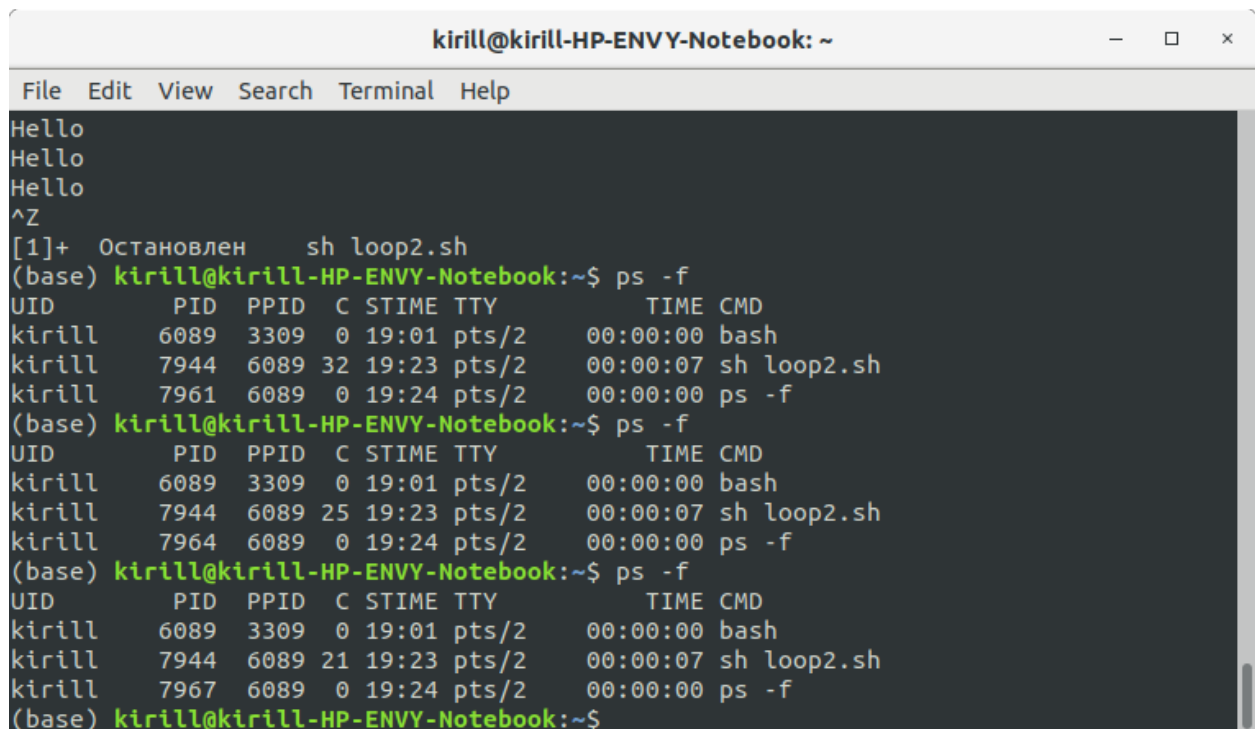
Чтобы послать процессу сигнал `STOP` нажмем комбинацию клавиш `Ctrl+Z`. Результат виден на рисунке 1.9.



```
kirill@kirill-HP-ENVY-Notebook: ~  
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]+  Остановлен      sh loop2.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.9 – Сигнал `STOP` для процесса `loop2`

Теперь несколько раз выполним команду `ps -f` (рисунок 1.10).

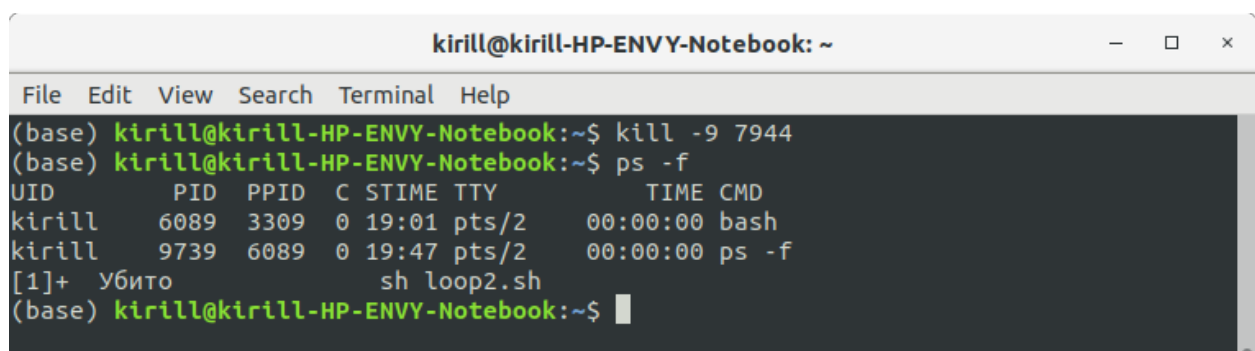


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
Hello  
Hello  
Hello  
^Z  
[1]+  Остановлен      sh loop2.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill    6089   3309   0  19:01 pts/2    00:00:00 bash  
kirill    7944   6089  32  19:23 pts/2    00:00:07 sh loop2.sh  
kirill    7961   6089   0  19:24 pts/2    00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill    6089   3309   0  19:01 pts/2    00:00:00 bash  
kirill    7944   6089  25  19:23 pts/2    00:00:07 sh loop2.sh  
kirill    7964   6089   0  19:24 pts/2    00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill    6089   3309   0  19:01 pts/2    00:00:00 bash  
kirill    7944   6089  21  19:23 pts/2    00:00:07 sh loop2.sh  
kirill    7967   6089   0  19:24 pts/2    00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.10 – Последовательный запуск команд `ps -f`

Как видно из рисунка 1.10 у процесса `loop2` меняется значение параметра `C`, т.е. процент времени CPU, используемого процессом. Т.к. мы послали сигнал `STOP` процесс был приостановлен, но не завершен, поэтому он отсаеся в выводе `ps -f`, но постепенно перестает занимать процессорное время. Данный процесс можно возобновить послав сигнал `CONT`.

Теперь завершим процесс `loop2` используя команду `kill -9` (рисунок 1.11).



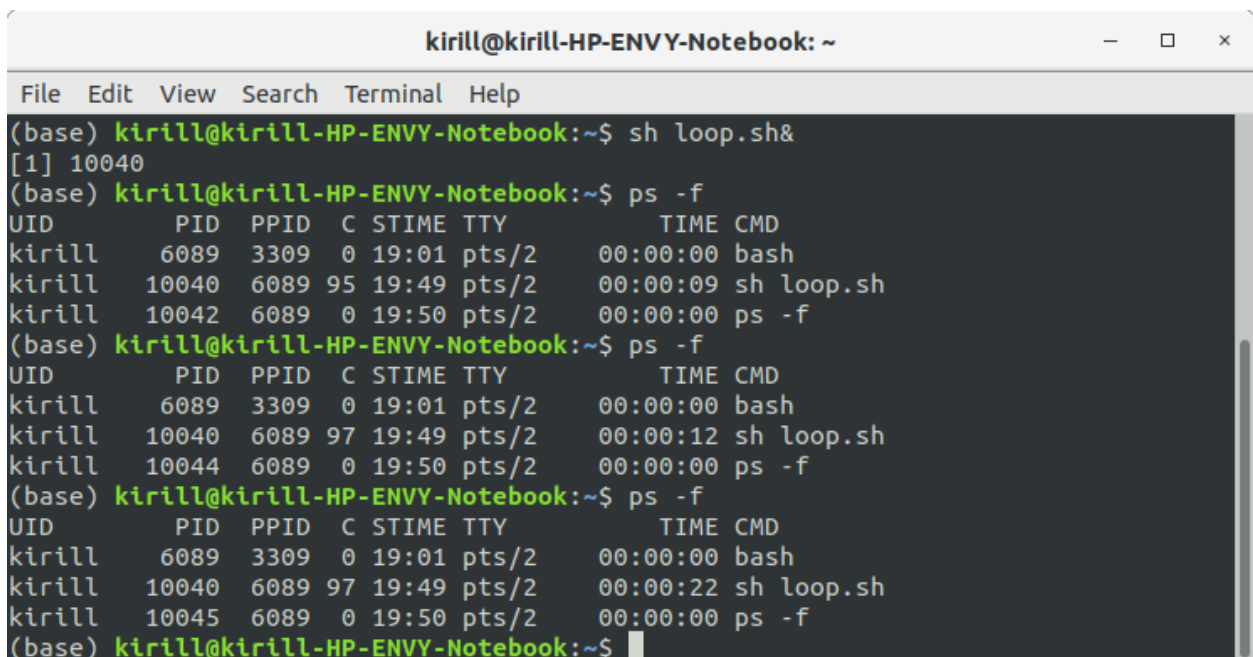
```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ kill -9 7944  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill    6089   3309   0  19:01 pts/2    00:00:00 bash  
kirill    9739   6089   0  19:47 pts/2    00:00:00 ps -f  
[1]+  убито          sh loop2.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.11 – Завершение процесса `loop2`



Как видно из рисунка 1.11 процесс loop2 был завершен, об этом свидетельствует сообщение от утилиты ps. Однако сигнал KILL завершает его не удостоверившись в его корректном завершении, что во многих случаях может приводить к ошибкам.

Запустим процесс loop в фоне с помощью команды `sh loop.sh&` и посмотрим, что выдает последовательный запуск команды `ps -f` (рисунок 1.12).

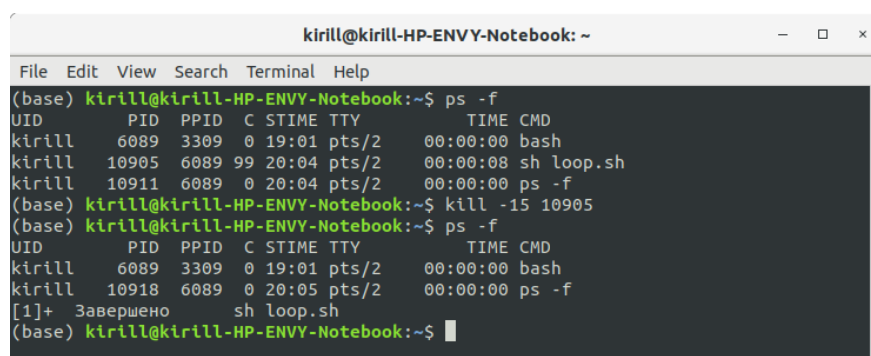


```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh loop.sh&
[1] 10040
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
kirill       6089    3309  0  19:01 pts/2        00:00:00 bash
kirill      10040    6089  95  19:49 pts/2        00:00:09 sh loop.sh
kirill      10042    6089  0  19:50 pts/2        00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
kirill       6089    3309  0  19:01 pts/2        00:00:00 bash
kirill      10040    6089  97  19:49 pts/2        00:00:12 sh loop.sh
kirill      10044    6089  0  19:50 pts/2        00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
kirill       6089    3309  0  19:01 pts/2        00:00:00 bash
kirill      10040    6089  97  19:49 pts/2        00:00:22 sh loop.sh
kirill      10045    6089  0  19:50 pts/2        00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.12 – Запуск процесса loop в фоне

Как видно из рисунке 1.12 видно, что процесс запущенный в фоне продолжает занимать большой процент процессорного времени.

Теперь завершим процесс loop и посмотрим, что выдаст команда `ps -f`(рисунок 1.13).

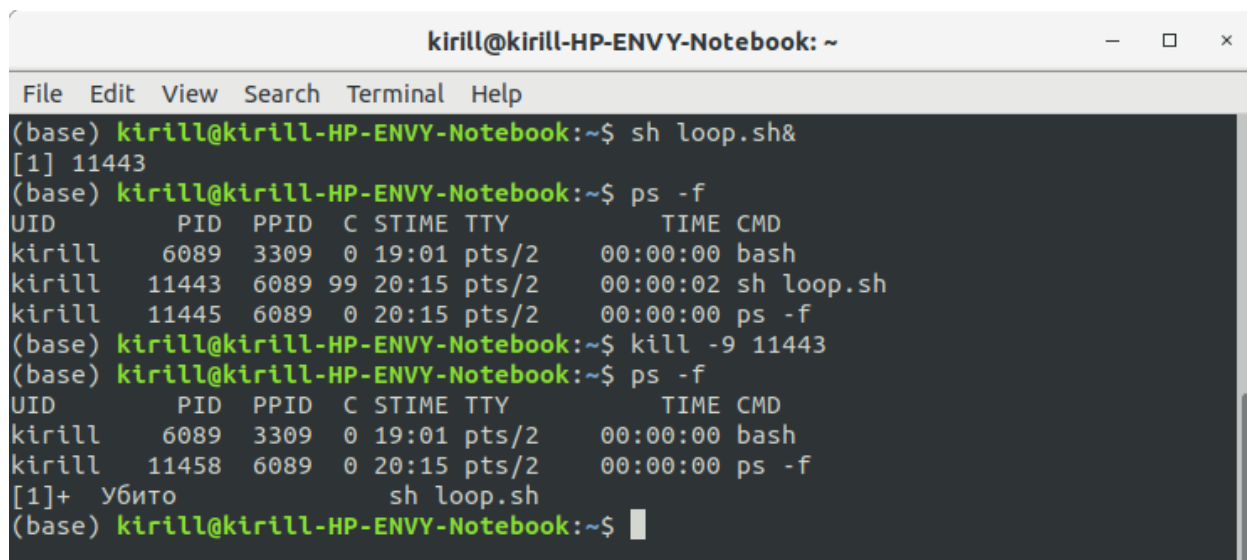


```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
kirill       6089    3309  0  19:01 pts/2        00:00:00 bash
kirill      10905    6089  99  20:04 pts/2        00:00:08 sh loop.sh
kirill      10911    6089  0  20:04 pts/2        00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ kill -15 10905
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
kirill       6089    3309  0  19:01 pts/2        00:00:00 bash
kirill      10918    6089  0  20:05 pts/2        00:00:00 ps -f
[1]+  Завершено sh loop.sh
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.13 – Завершение процесса loop сигналом TERM

Как видно из рисунка 1.13 процесс был завершен, при чем завершен после корректного завершения выполнения процесса, т.к. мы послали сигнал TERM, а не KILL.

Теперь запустим процесс loop фоне и ”убьем”его командой kill -9 (рисунок 1.14).

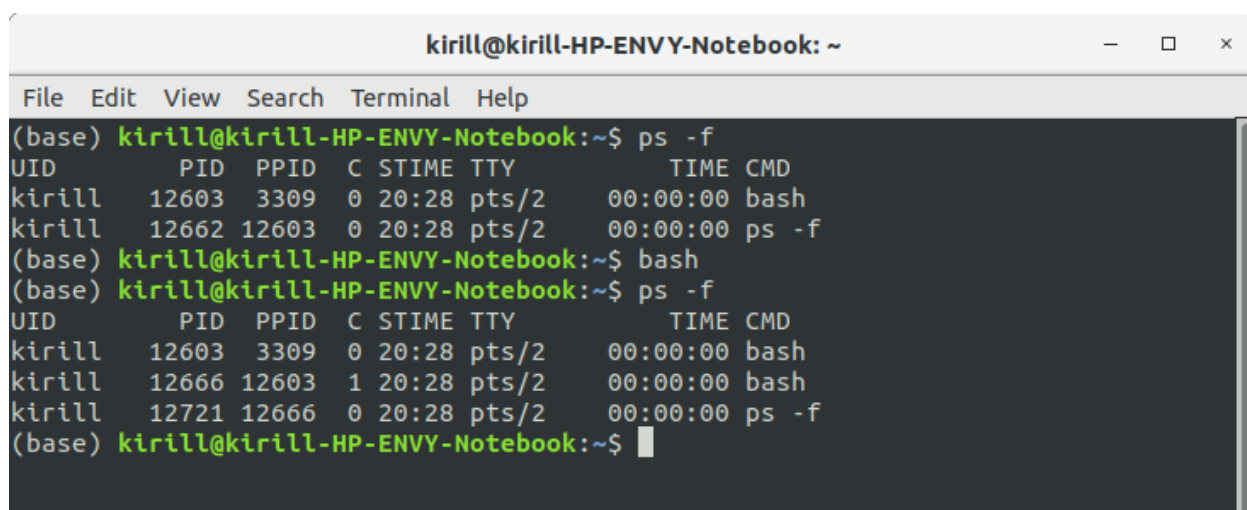


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh loop.sh&  
[1] 11443  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill    6089   3309  0  19:01 pts/2    00:00:00 bash  
kirill   11443   6089  99  20:15 pts/2    00:00:02 sh loop.sh  
kirill   11445   6089  0  20:15 pts/2    00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$ kill -9 11443  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill    6089   3309  0  19:01 pts/2    00:00:00 bash  
kirill   11458   6089  0  20:15 pts/2    00:00:00 ps -f  
[1]+  Убито                sh loop.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.14 – Завершение процесса loop сигналом KILL

Как видно из вывода команды ps -f на рисунке 1.14 процесс loop был именно ”убит не дожидаясь корректного завершения выполнения.

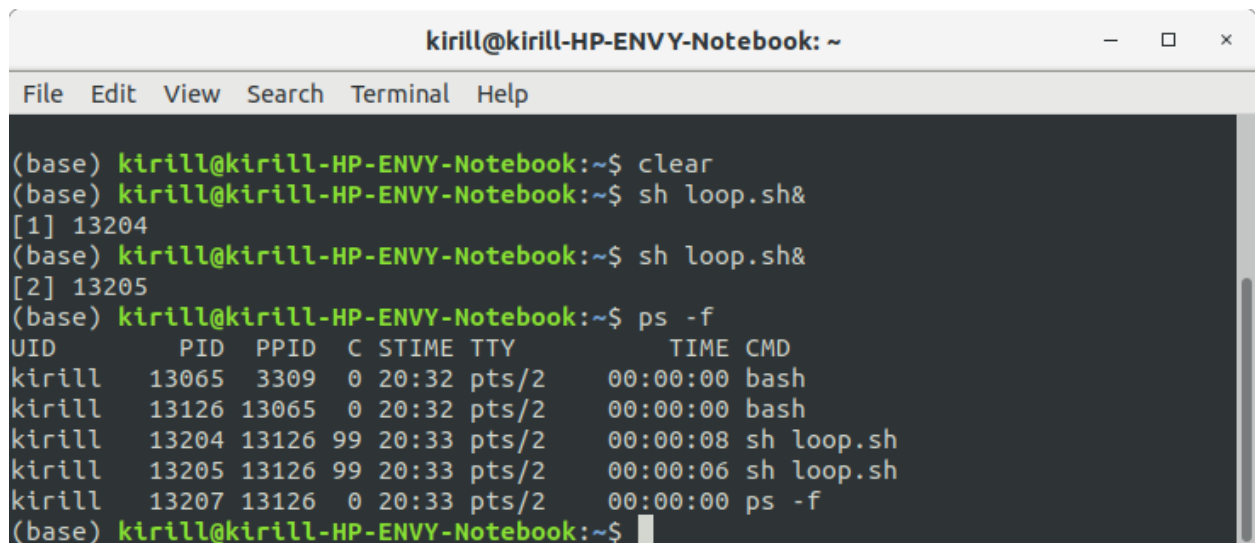
Запустим еще один экземпляр bash, результат продемонстрирован на рисунке 1.15.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill   12603   3309  0  20:28 pts/2    00:00:00 bash  
kirill   12662  12603  0  20:28 pts/2    00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$ bash  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY      TIME CMD  
kirill   12603   3309  0  20:28 pts/2    00:00:00 bash  
kirill   12666  12603  1  20:28 pts/2    00:00:00 bash  
kirill   12721  12666  0  20:28 pts/2    00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.15 – Запуск еще одного экземпляра bash

Запустим несколько процессов loop в фоне, результат продемонстрирован на рисунке 1.16.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help

(base) kirill@kirill-HP-ENVY-Notebook:~$ clear
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh loop.sh&
[1] 13204
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh loop.sh&
[2] 13205
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID      PID  PPID  C  STIME TTY      TIME CMD
kirill   13065   3309  0  20:32 pts/2    00:00:00 bash
kirill   13126  13065  0  20:32 pts/2    00:00:00 bash
kirill   13204  13126  99  20:33 pts/2    00:00:08 sh loop.sh
kirill   13205  13126  99  20:33 pts/2    00:00:06 sh loop.sh
kirill   13207  13126  0  20:33 pts/2    00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.16 – Запуск нескольких процессов loop в фоне

Как видно из рисунка 1.16 родительским процессом для новых loop является новый экземпляр bash.

Теперь остановим эти процессы, послав им сигнал STOP, с помощью команды kill -19. Затем возобновим эти процессы, послав сигнал CONT, с помощью команды kill -18. После посылок сигналов будем распечатывать процессы командой ps -f. Последовательность выполнения продемонстрирована на рисунке 1.17.

```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID      PID  PPID  C  STIME TTY      TIME CMD
kirill   13065  3309   0  20:32 pts/2    00:00:00 bash
kirill   13126  13065   0  20:32 pts/2    00:00:00 bash
kirill   13204  13126  99  20:33 pts/2    00:06:47 sh loop.sh
kirill   13205  13126  99  20:33 pts/2    00:06:45 sh loop.sh
kirill   13595  13126   0  20:40 pts/2    00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ kill -19 13204 13205

[1]- Остановлен      sh loop.sh

[2]+ Остановлен      sh loop.sh
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID      PID  PPID  C  STIME TTY      TIME CMD
kirill   13065  3309   0  20:32 pts/2    00:00:00 bash
kirill   13126  13065   0  20:32 pts/2    00:00:00 bash
kirill   13204  13126  95  20:33 pts/2    00:07:20 sh loop.sh
kirill   13205  13126  94  20:33 pts/2    00:07:18 sh loop.sh
kirill   13612  13126   0  20:41 pts/2    00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID      PID  PPID  C  STIME TTY      TIME CMD
kirill   13065  3309   0  20:32 pts/2    00:00:00 bash
kirill   13126  13065   0  20:32 pts/2    00:00:00 bash
kirill   13204  13126  87  20:33 pts/2    00:07:20 sh loop.sh
kirill   13205  13126  87  20:33 pts/2    00:07:18 sh loop.sh
kirill   13634  13126   0  20:41 pts/2    00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ kill -18 13204 13205
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID      PID  PPID  C  STIME TTY      TIME CMD
kirill   13065  3309   0  20:32 pts/2    00:00:00 bash
kirill   13126  13065   0  20:32 pts/2    00:00:00 bash
kirill   13204  13126  75  20:33 pts/2    00:08:12 sh loop.sh
kirill   13205  13126  75  20:33 pts/2    00:08:10 sh loop.sh
kirill   13675  13126   0  20:44 pts/2    00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID      PID  PPID  C  STIME TTY      TIME CMD
kirill   13065  3309   0  20:32 pts/2    00:00:00 bash
kirill   13126  13065   0  20:32 pts/2    00:00:00 bash
kirill   13204  13126  76  20:33 pts/2    00:08:31 sh loop.sh
kirill   13205  13126  76  20:33 pts/2    00:08:30 sh loop.sh
kirill   13681  13126   0  20:44 pts/2    00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f
UID      PID  PPID  C  STIME TTY      TIME CMD
kirill   13065  3309   0  20:32 pts/2    00:00:00 bash
kirill   13126  13065   0  20:32 pts/2    00:00:00 bash
kirill   13204  13126  76  20:33 pts/2    00:08:39 sh loop.sh
kirill   13205  13126  76  20:33 pts/2    00:08:38 sh loop.sh
kirill   13683  13126   0  20:44 pts/2    00:00:00 ps -f
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

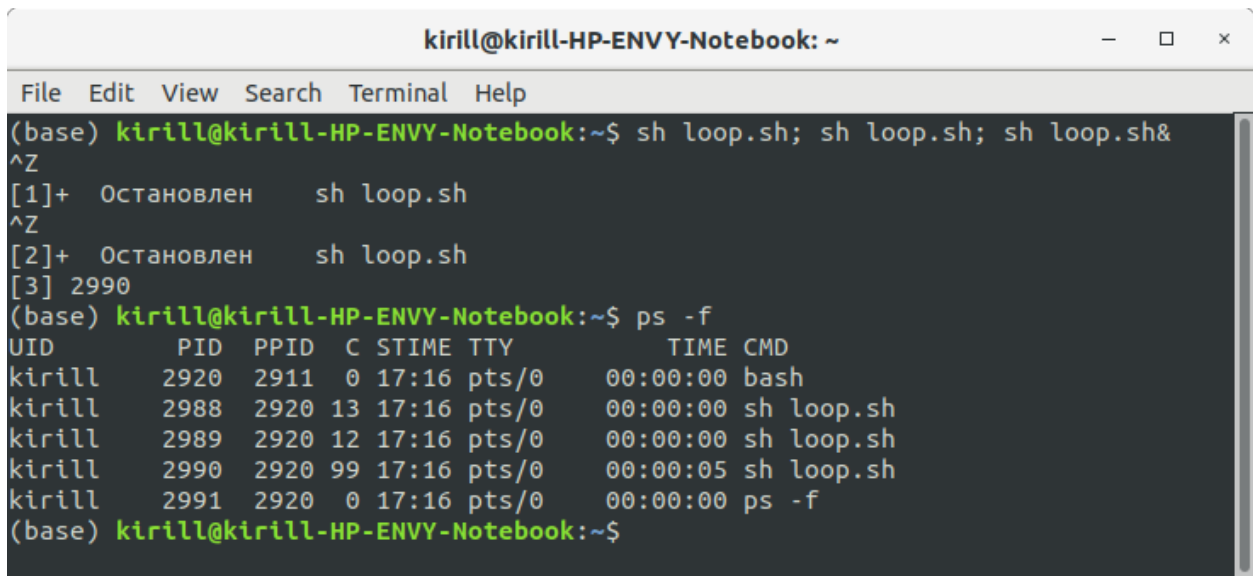
Рисунок 1.17 – Остановка и возобнослвление процессов

Как видно из рисунка 1.17 после выполнения команды kill -19 процессы приостанавливаются о чем свидетельствует уменьшение процента затрачиваемого процессорного времени. После выполнения команды kill -

18 процессы возобновляется, о чем свидетельствует увеличение процента затрачиваемого процессорного времени.

## 1.2 Часть 2

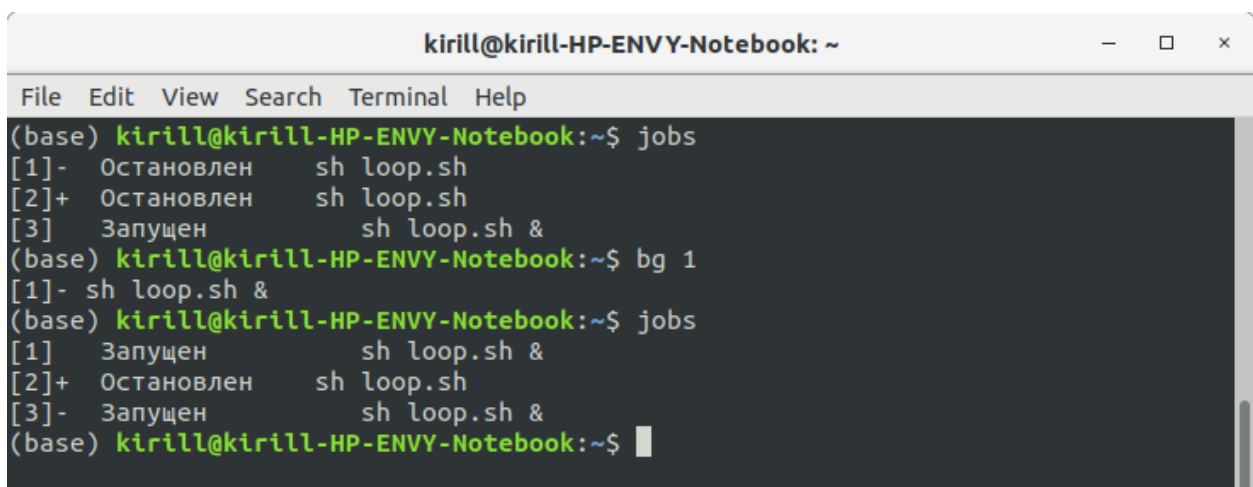
Запустим 3 процесса `loop`: 2 в интерактивном режиме (PID 2988 и 2989) и 1 (PID 2990) в фоновом. Процесс запуска и результат выполнения команды представлены на рисунке 1.18.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sh loop.sh; sh loop.sh; sh loop.sh&  
^Z  
[1]+  Остановлен      sh loop.sh  
^Z  
[2]+  Остановлен      sh loop.sh  
[3] 2990  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID          PID  PPID  C  STIME TTY          TIME CMD  
kirill      2920   2911  0  17:16 pts/0        00:00:00 bash  
kirill      2988   2920 13  17:16 pts/0        00:00:00 sh loop.sh  
kirill      2989   2920 12  17:16 pts/0        00:00:00 sh loop.sh  
kirill      2990   2920 99  17:16 pts/0        00:00:05 sh loop.sh  
kirill      2991   2920  0  17:16 pts/0        00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.18 – Запуск 3-х процессов

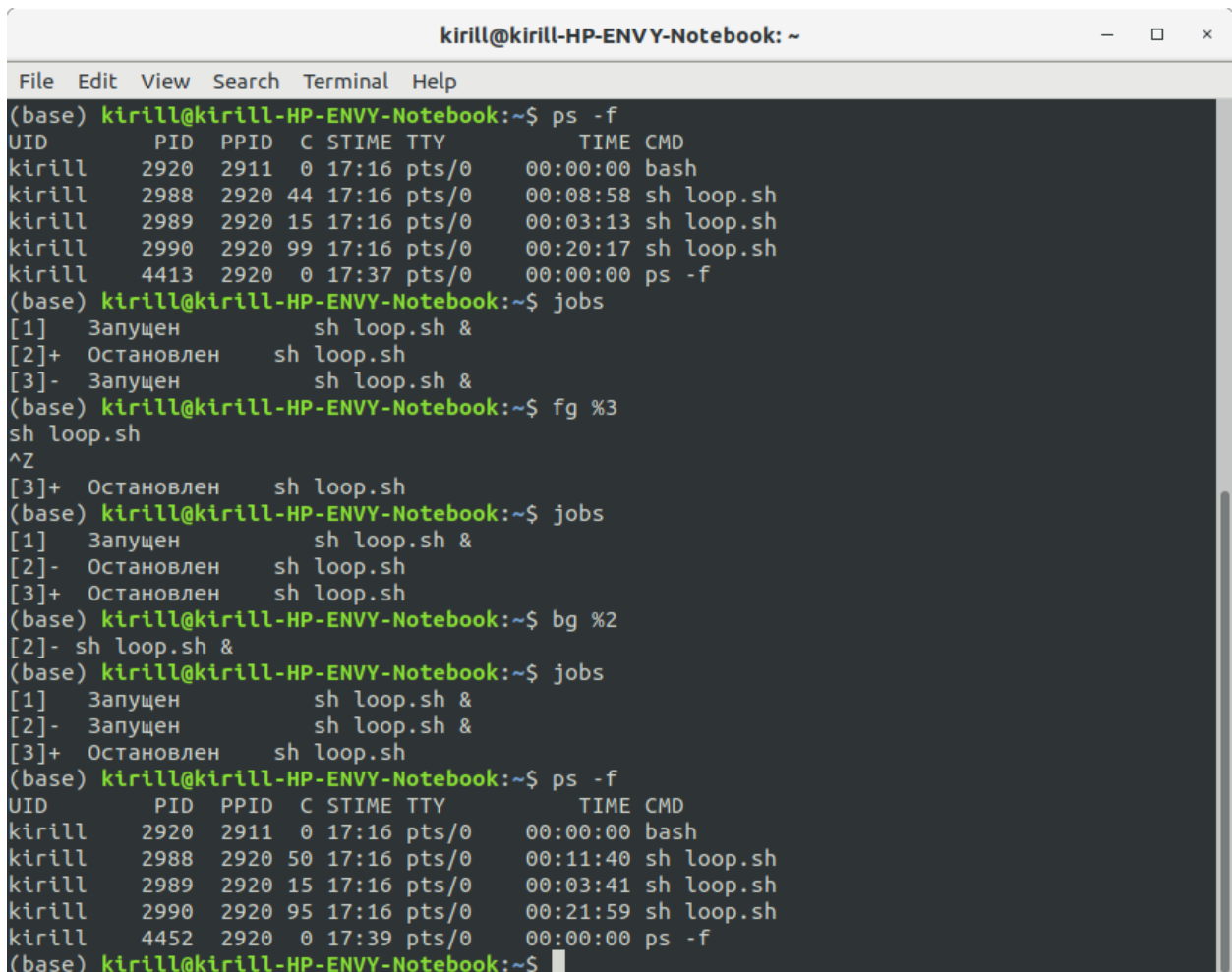
Переведем 1 из процессов в фоновый режим с помощью команды `bg`, указав номер процесса из вывода команды `jobs` (рисунок 1.19).



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ jobs  
[1]-  Остановлен      sh loop.sh  
[2]+  Остановлен      sh loop.sh  
[3]   Запущен         sh loop.sh &  
(base) kirill@kirill-HP-ENVY-Notebook:~$ bg 1  
[1]-  sh loop.sh &  
(base) kirill@kirill-HP-ENVY-Notebook:~$ jobs  
[1]   Запущен         sh loop.sh &  
[2]+  Остановлен      sh loop.sh  
[3]-  Запущен         sh loop.sh &  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.19 – Перевод процесса в фоновый режим

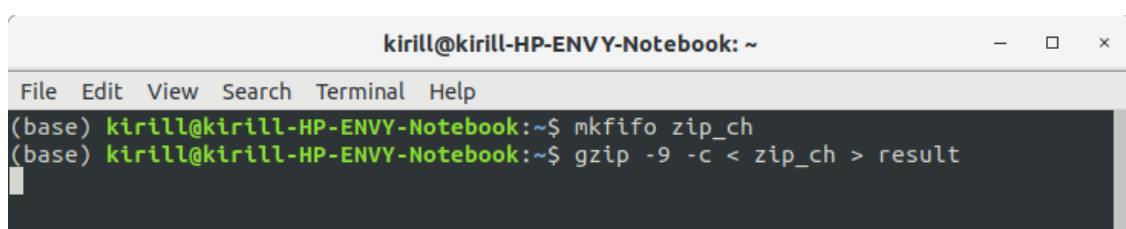
Поэкспериментируем с переводом процессов в разные режимы работы. Переведем процесс 3 в интерактивный режим, затем переведем процесс 2 в фоновый режим. Результат выполнения представлен на рисунке 1.20.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
kirill    2920   2911  0  17:16 pts/0      00:00:00 bash  
kirill    2988   2920  44  17:16 pts/0      00:08:58 sh loop.sh  
kirill    2989   2920  15  17:16 pts/0      00:03:13 sh loop.sh  
kirill    2990   2920  99  17:16 pts/0      00:20:17 sh loop.sh  
kirill    4413   2920  0  17:37 pts/0      00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$ jobs  
[1]  Запущен          sh loop.sh &  
[2]+  Остановлен      sh loop.sh  
[3]-  Запущен          sh loop.sh &  
(base) kirill@kirill-HP-ENVY-Notebook:~$ fg %3  
sh loop.sh  
^Z  
[3]+  Остановлен      sh loop.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$ jobs  
[1]  Запущен          sh loop.sh &  
[2]-  Остановлен      sh loop.sh  
[3]+  Остановлен      sh loop.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$ bg %2  
[2]-  sh loop.sh &  
(base) kirill@kirill-HP-ENVY-Notebook:~$ jobs  
[1]  Запущен          sh loop.sh &  
[2]-  Запущен          sh loop.sh &  
[3]+  Остановлен      sh loop.sh  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -f  
UID      PID  PPID  C  STIME TTY          TIME CMD  
kirill    2920   2911  0  17:16 pts/0      00:00:00 bash  
kirill    2988   2920  50  17:16 pts/0      00:11:40 sh loop.sh  
kirill    2989   2920  15  17:16 pts/0      00:03:41 sh loop.sh  
kirill    2990   2920  95  17:16 pts/0      00:21:59 sh loop.sh  
kirill    4452   2920  0  17:39 pts/0      00:00:00 ps -f  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.20 – Перевод процессов в различные режимы работы

Для создания именованного канала воспользуемся командой `mkfifo`. После его создания мы можем организовать связь между выводами и вводами различных процессов. С помощью полученного канала сделаем так, чтобы утилита для архивирования `gzip` получала ввод из канала и выводила результат в файл `result` (рисунок 1.21).

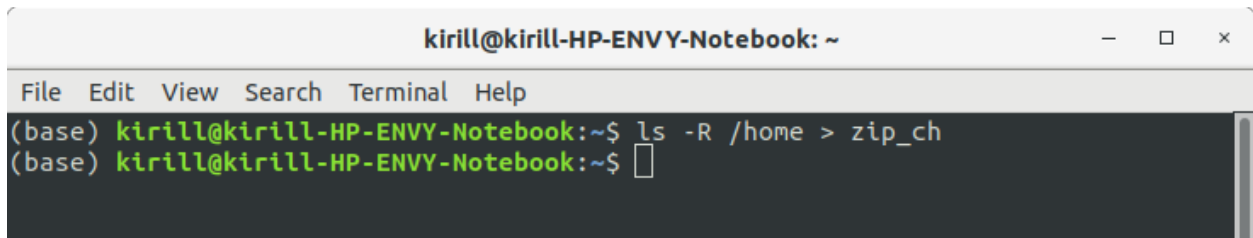


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ mkfifo zip_ch  
(base) kirill@kirill-HP-ENVY-Notebook:~$ gzip -9 -c < zip_ch > result
```

Рисунок 1.21 – Создание именованного канала для архивирования данных



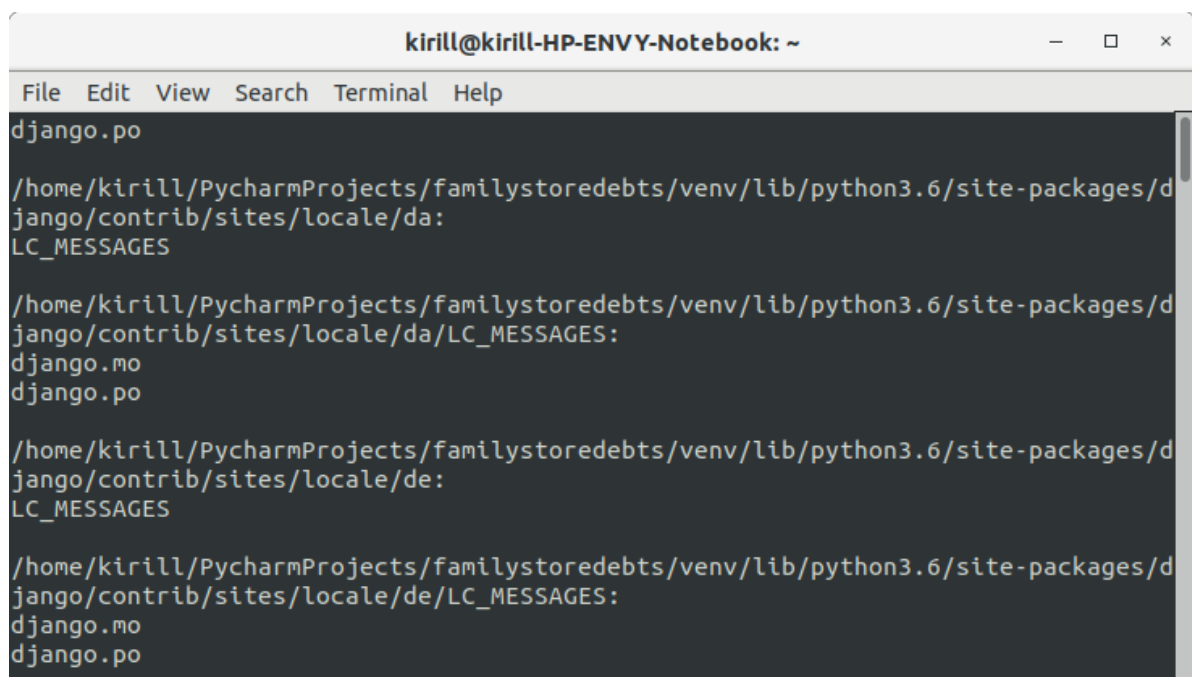
Теперь откроем второй терминал и передадим в канал список содержимое домашнего каталога и его подкаталогов с помощью команды `ls -R` (рисунок 1.22).



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ls -R /home > zip_ch  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.22 – Передача в канал для архивирования список содержимого домашнего каталога

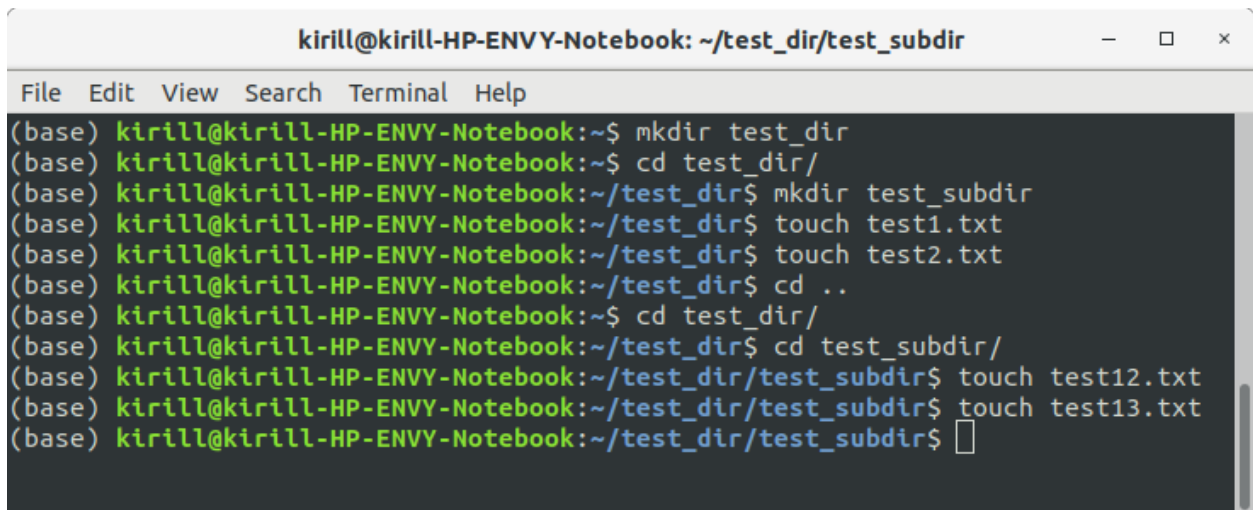
Вернемся в 1-ый терминал, чтобы посмотреть содержимое файла `result`. Для просмотра содержимого архивированного файла воспользуемся командой `zcat`. Результат выполнения представлен на рисунке 1.23.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
django.po  
  
/home/kirill/PycharmProjects/familystoredebts/venv/lib/python3.6/site-packages/d  
jango/contrib/sites/locale/da:  
LC_MESSAGES  
  
/home/kirill/PycharmProjects/familystoredebts/venv/lib/python3.6/site-packages/d  
jango/contrib/sites/locale/da/LC_MESSAGES:  
django.mo  
django.po  
  
/home/kirill/PycharmProjects/familystoredebts/venv/lib/python3.6/site-packages/d  
jango/contrib/sites/locale/de:  
LC_MESSAGES  
  
/home/kirill/PycharmProjects/familystoredebts/venv/lib/python3.6/site-packages/d  
jango/contrib/sites/locale/de/LC_MESSAGES:  
django.mo  
django.po
```

Рисунок 1.23 – Содержимое файла `result`

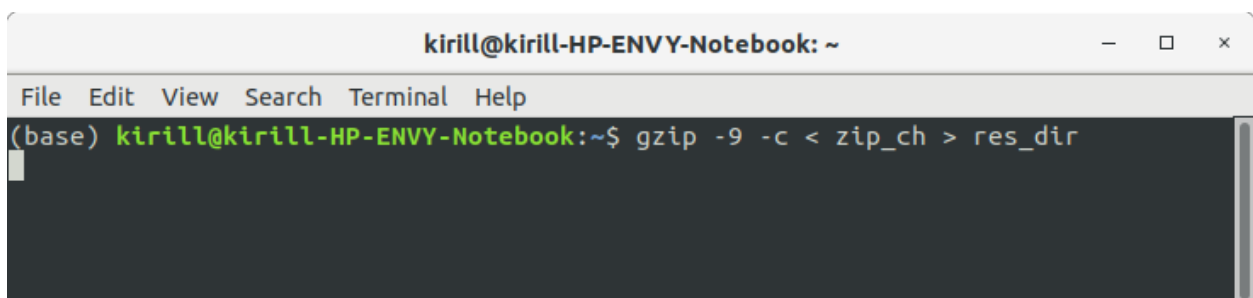
Для выполнения следующего задания подготовим тестовый каталог, поместив в него несколько файлов и подкаталог. Для создания каталогов воспользуемся командой `mkdir`, для создания файлов командой `touch`, для перемещения между каталогами используем команду `cd`. Процесс создания тестового каталога представлен на рисунке 1.24.



```
kirill@kirill-HP-ENVY-Notebook: ~/test_dir/test_subdir
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ mkdir test_dir
(base) kirill@kirill-HP-ENVY-Notebook:~$ cd test_dir/
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir$ mkdir test_subdir
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir$ touch test1.txt
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir$ touch test2.txt
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir$ cd ..
(base) kirill@kirill-HP-ENVY-Notebook:~$ cd test_dir/
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir$ cd test_subdir/
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir/test_subdir$ touch test12.txt
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir/test_subdir$ touch test13.txt
(base) kirill@kirill-HP-ENVY-Notebook:~/test_dir/test_subdir$
```

Рисунок 1.24 – Создание тестового каталога

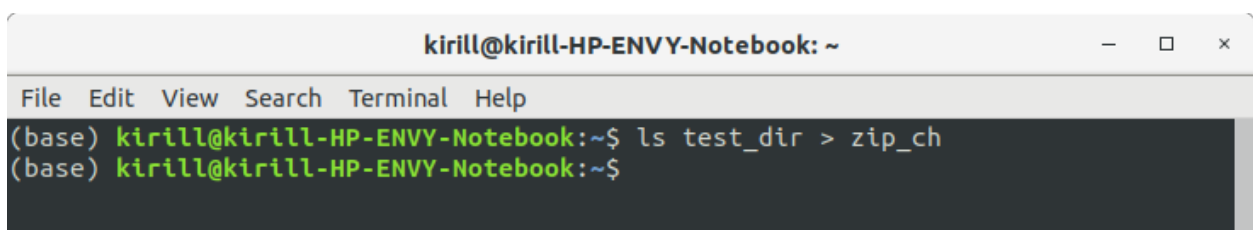
Теперь выполним аналогичные предыдущему заданию действия. Создадим новое соединение и передадим туда список содержимого тестового каталога. Создание нового соединения представлено на рисунке 1.25.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ gzip -9 -c < zip_ch > res_dir
```

Рисунок 1.25 – Создание нового соединения

Теперь из второго терминала передадим список содержимого тестового каталога с помощью команды `ls -l` (рисунок 1.26).



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ ls test_dir > zip_ch
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.26 – Передача списка содержимого тестового каталога

Перейдем в первый терминал и посмотрим содержимое файла `res_dir` с помощью утилиты `zcat` (рисунок 1.27).



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ gzip -9 -c < zip_ch > res_dir  
(base) kirill@kirill-HP-ENVY-Notebook:~$ zcat res_dir  
test1.txt  
test2.txt  
test_subdir  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.27 – Содержимое файла res\_dir

### 1.3 Часть 3 (вариант 10)

Чтобы просмотреть процессы в реальном времени воспользуемся командой `top`. Чтобы отсортировать процессы по PID необходимо передать команде `top` опцию `-o` и указать по какому параметру необходимо выполнить сортировку, в данном случае – PID. Выполняемая команжа представлена на рисунке 1.28.

```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ top -o PID
```

Рисунок 1.28 – Команда для отображения процессов в реальном времени отсортированных по PID

Результат выполнения данной команды представлен на рисунке 1.29.

```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
top - 10:57:56 up 1:39, 1 user, load average: 0,44, 0,72, 0,92  
Tasks: 275 total, 1 running, 221 sleeping, 1 stopped, 0 zombie  
%Cpu(s): 1,7 us, 0,8 sy, 0,0 ni, 97,4 id, 0,0 wa, 0,0 hi, 0,1 si, 0,0 st  
КиБ Mem : 8038836 total, 2010404 free, 3348456 used, 2679976 buff/cache  
КиБ Swap: 8000508 total, 8000508 free, 0 used. 4115116 avail Mem  


| PID  | USER   | PR | NI | VIRT    | RES   | SHR   | S | %CPU | %MEM | TIME+   | COMMAND     |
|------|--------|----|----|---------|-------|-------|---|------|------|---------|-------------|
| 7446 | kirill | 20 | 0  | 45500   | 4056  | 3408  | R | 0,3  | 0,1  | 0:00.07 | top         |
| 7425 | kirill | 20 | 0  | 45500   | 4220  | 3392  | T | 0,0  | 0,1  | 0:00.19 | top         |
| 7366 | kirill | 20 | 0  | 24324   | 5624  | 3704  | S | 0,0  | 0,1  | 0:00.05 | bash        |
| 7345 | root   | 20 | 0  | 0       | 0     | 0     | I | 0,0  | 0,0  | 0:00.05 | kworker/u8+ |
| 7344 | root   | 20 | 0  | 0       | 0     | 0     | I | 0,0  | 0,0  | 0:00.00 | kworker/1:2 |
| 7343 | root   | 20 | 0  | 0       | 0     | 0     | I | 0,0  | 0,0  | 0:00.00 | kworker/1:+ |
| 7342 | root   | 20 | 0  | 0       | 0     | 0     | I | 0,0  | 0,0  | 0:00.01 | kworker/3:+ |
| 7269 | kirill | 20 | 0  | 2579204 | 84200 | 67352 | S | 0,0  | 1,0  | 0:00.30 | Web Content |
| 6858 | root   | 20 | 0  | 0       | 0     | 0     | I | 0,0  | 0,0  | 0:00.21 | kworker/0:+ |


```

Рисунок 1.29 – Запущенная команда top

Чтобы остановить процесс во время работы команды `top`, необходимо нажать клавишу `k`, затем ввести PID процесса, после этого можно ввести какой сигнал подать процессу (по умолчанию подается сигнал `TERM`). Процесс выполнения вышеуказанных действий продемонстрирован на рисунках 1.30 и 1.31.

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 10:57:59 up 1:39, 1 user, load average: 0,44, 0,72, 0,92
Tasks: 275 total, 1 running, 221 sleeping, 1 stopped, 0 zombie
%Cpu(s): 4,2 us, 1,2 sy, 0,0 ni, 92,3 id, 1,7 wa, 0,0 hi, 0,5 si, 0,0 st
КиБ Mem : 8038836 total, 2008768 free, 3347404 used, 2682664 buff/cache
КиБ Swap: 8000508 total, 8000508 free, 0 used. 4113600 avail Mem
PID to signal/kill [default pid = 7446] 7446

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7446	kirill	20	0	45500	4056	3408	R	0,3	0,1	0:00.08	top
7425	kirill	20	0	45500	4220	3392	T	0,0	0,1	0:00.19	top
7366	kirill	20	0	24324	5624	3704	S	0,0	0,1	0:00.05	bash
7345	root	20	0	0	0	0	I	0,0	0,0	0:00.05	kworker/u8+
7344	root	20	0	0	0	0	I	0,0	0,0	0:00.00	kworker/1:2
7343	root	20	0	0	0	0	I	0,0	0,0	0:00.00	kworker/1:+
7342	root	20	0	0	0	0	I	0,0	0,0	0:00.01	kworker/3:+
7269	kirill	20	0	2579204	84200	67352	S	0,0	1,0	0:00.30	Web Content
6858	root	20	0	0	0	0	I	0,0	0,0	0:00.21	kworker/0:+
6688	root	20	0	0	0	0	I	0,0	0,0	0:00.16	kworker/1:+
6516	root	20	0	0	0	0	I	0,0	0,0	0:00.04	kworker/0:+
6502	root	20	0	0	0	0	I	0,0	0,0	0:00.10	kworker/3:+
6405	root	20	0	0	0	0	I	0,3	0,0	0:01.17	kworker/u8+
6402	root	20	0	0	0	0	I	0,0	0,0	0:02.64	kworker/2:+
5852	kirill	20	0	751456	47380	34128	S	0,7	0,6	0:01.28	eog
5792	kirill	20	0	24456	2084	0	S	0,0	0,0	0:00.00	bash
5702	kirill	20	0	24456	6004	3920	S	0,0	0,1	0:00.08	bash

Рисунок 1.30 – Вызов команды для отправки сигнала процессу и указание PID

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 10:57:59 up 1:39, 1 user, load average: 0,44, 0,72, 0,92
Tasks: 275 total, 1 running, 221 sleeping, 1 stopped, 0 zombie
%Cpu(s): 4,2 us, 1,2 sy, 0,0 ni, 92,3 id, 1,7 wa, 0,0 hi, 0,5 si, 0,0 st
КиБ Mem : 8038836 total, 2008768 free, 3347404 used, 2682664 buff/cache
КиБ Swap: 8000508 total, 8000508 free, 0 used. 4113600 avail Mem
Send pid 7446 signal [15/sigterm]

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7446	kirill	20	0	45500	4056	3408	R	0,3	0,1	0:00.08	top
7425	kirill	20	0	45500	4220	3392	T	0,0	0,1	0:00.19	top
7366	kirill	20	0	24324	5624	3704	S	0,0	0,1	0:00.05	bash
7345	root	20	0	0	0	0	I	0,0	0,0	0:00.05	kworker/u8+
7344	root	20	0	0	0	0	I	0,0	0,0	0:00.00	kworker/1:2
7343	root	20	0	0	0	0	I	0,0	0,0	0:00.00	kworker/1:+

Рисунок 1.31 – Указание сигнала для выбранного процесса

Для определения последнего процесса запущенного текущим пользователем передадим команде `top` опции `-o -TIME+` – сортировка по возрастанию процессорного времени и `-u kirill` – вывод процессов только пользователя `kirill` (рисунок 1.32).

```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ top -o -TIME+ -u kirill
```

Рисунок 1.32 – Вывод процессов пользователя `kirill` по возрастанию времени запуска

Теперь в 1-ой строке указан последний процесс запущенный пользователем, чтобы послать ему сигнал необходимо нажать клавишу `k`, ввести PID процесса и сигнал, который необходимо ему передать. Выполнение данной последовательности действий представлено на рисунках 1.33 и 1.34.

```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 19:19:30 up 10:00, 1 user, load average: 1,73, 2,02, 1,93
Tasks: 286 total, 2 running, 226 sleeping, 4 stopped, 0 zombie
%Cpu(s): 23,8 us, 16,2 sy, 0,0 ni, 60,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
КиБ Mem : 8038836 total, 459452 free, 4698004 used, 2881380 buff/cache
КиБ Swap: 8000508 total, 7999472 free, 1036 used. 1939712 avail Mem
PID to signal/kill [default pid = 3] 1525
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1524	kirill	20	0	349324	6296	5712	S	0,0	0,1	0:00.00	at-spi-bus-laun
1598	kirill	20	0	275088	6828	6172	S	0,0	0,1	0:00.00	ibus-dconf
1609	kirill	20	0	265896	4956	4520	S	0,0	0,1	0:00.00	xdg-permission-
10532	kirill	20	0	170344	2124	1852	S	0,0	0,0	0:00.00	speech-dispatch
1422	kirill	20	0	206248	5816	5152	S	0,0	0,1	0:00.01	gdm-x-session
1761	kirill	20	0	502880	12664	10900	S	0,0	0,2	0:00.01	gsd-printer
1567	kirill	20	0	416112	5372	4828	S	0,0	0,1	0:00.02	gvfsd-fuse
1634	kirill	20	0	269984	4936	4472	S	0,0	0,1	0:00.02	gvfs-mtp-volume
1638	kirill	20	0	268160	5980	5404	S	0,0	0,1	0:00.02	gvfs-goa-volume
1664	kirill	20	0	372936	7816	6936	S	0,0	0,1	0:00.02	gvfs-afc-volume

Рисунок 1.33 – Ввод PID процесса для остановки

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 19:19:30 up 10:00, 1 user, load average: 1,73, 2,02, 1,93
Tasks: 286 total, 2 running, 226 sleeping, 4 stopped, 0 zombie
%Cpu(s): 23,8 us, 16,2 sy, 0,0 ni, 60,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
КиБ Mem : 8038836 total, 459452 free, 4698004 used, 2881380 buff/cache
КиБ Swap: 8000508 total, 7999472 free, 1036 used. 1939712 avail Mem
Send pid 1525 signal [15/sigterm] sigkill

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1524	kirill	20	0	349324	6296	5712	S	0,0	0,1	0:00.00	at-spi-bus-laun
1598	kirill	20	0	275088	6828	6172	S	0,0	0,1	0:00.00	ibus-dconf
1609	kirill	20	0	265896	4956	4520	S	0,0	0,1	0:00.00	xdg-permission-
10532	kirill	20	0	170344	2124	1852	S	0,0	0,0	0:00.00	speech-dispatch
1422	kirill	20	0	206248	5816	5152	S	0,0	0,1	0:00.01	gdm-x-session
1761	kirill	20	0	502880	12664	10900	S	0,0	0,2	0:00.01	gsd-printer
1567	kirill	20	0	416112	5372	4828	S	0,0	0,1	0:00.02	gvfsd-fuse
1634	kirill	20	0	269984	4936	4472	S	0,0	0,1	0:00.02	gvfs-mtp-volume
1638	kirill	20	0	268160	5980	5404	S	0,0	0,1	0:00.02	gvfs-goa-volume
1664	kirill	20	0	372936	7816	6936	S	0,0	0,1	0:00.02	gvfs-afc-volume

Рисунок 1.34 – Ввод сигнала, отсылаемого процессу

На рисунке 1.35 видно, что процесс был завершен.

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 19:19:49 up 10:00, 1 user, load average: 1,72, 2,00, 1,93
Tasks: 285 total, 3 running, 222 sleeping, 4 stopped, 0 zombie
%Cpu(s): 14,9 us, 10,0 sy, 0,0 ni, 74,6 id, 0,5 wa, 0,0 hi, 0,1 si, 0,0 st
КиБ Mem : 8038836 total, 464168 free, 4699388 used, 2875280 buff/cache
КиБ Swap: 8000508 total, 7999472 free, 1036 used. 1944756 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1598	kirill	20	0	275088	6828	6172	S	0,0	0,1	0:00.00	ibus-dconf
1609	kirill	20	0	265896	4956	4520	S	0,0	0,1	0:00.00	xdg-permission-
10532	kirill	20	0	170344	2124	1852	S	0,0	0,0	0:00.00	speech-dispatch
1422	kirill	20	0	206248	5816	5152	S	0,0	0,1	0:00.01	gdm-x-session
1761	kirill	20	0	502880	12664	10900	S	0,0	0,2	0:00.01	gsd-printer
1567	kirill	20	0	416112	5372	4828	S	0,0	0,1	0:00.02	gvfsd-fuse
1634	kirill	20	0	269984	4936	4472	S	0,0	0,1	0:00.02	gvfs-mtp-volume
1638	kirill	20	0	268160	5980	5404	S	0,0	0,1	0:00.02	gvfs-goa-volume
1664	kirill	20	0	372936	7816	6936	S	0,0	0,1	0:00.02	gvfs-afc-volume
1669	kirill	20	0	282912	6700	5916	S	0,0	0,1	0:00.02	gvfs-gphoto2-vo

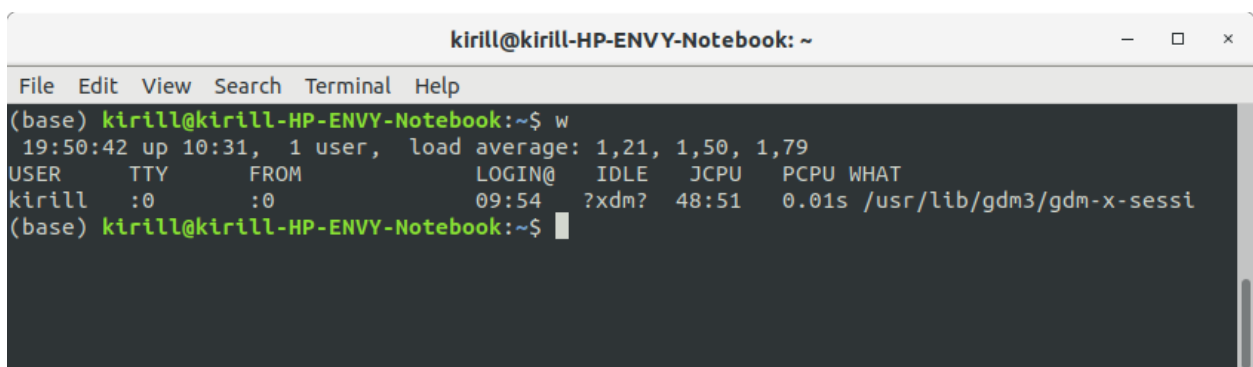
Рисунок 1.35 – Результат отправки сигнала

Для вывода информации о работающем в системе пользователе воспользуемся командой w, эта команда выводит следующую информацию:

1. USER – имя учетной записи;
2. TTY – название терминала;
3. FROM – имя хоста или IP адрес, с которого пользователь вошел в систему под конкретной учетной записью;
4. LOGIN – время, когда та или иная учетная запись впервые с момента включения машины зарегистрировалась в системе;

5. IDLE – время, когда пользователь проявил активность с определенного аккаунта (время простоя);
6. JCPU – время, использованное всеми процессами, запущенными в терминале tty;
7. PCPU – время, использованное текущим процессом (каким именно — можно посмотреть в столбце WHAT);
8. WHAT – текущие процессы и команды, которыми занят пользователь под конкретной учетной записью.

Результат выполнения данной команды представлен на рисунке 1.36.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ w  
19:50:42 up 10:31, 1 user, load average: 1,21, 1,50, 1,79  
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT  
kirill    :0        :0             09:54    ?xdm?  48:51  0.01s /usr/lib/gdm3/gdm-x-sessi  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.36 – Информация о работающем в системе пользователе

## Вывод

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