

# Липецкий государственный технический университет

Кафедра автоматизированных систем управления

Отчет по лабораторной работе №3  
по предмету "Операционная система Линукс"  
на тему  
"Процессы в Линукс"

Студент

\_\_\_\_\_  
подпись, дата

Акельев А.Р.  
фамилия, инициалы

Группа

Руководитель

доцент, кандидат наук  
ученая степень, ученое звание

\_\_\_\_\_  
подпись, дата

Кургасов В.В.  
фамилия, инициалы

Липецк 2022 г.

# Содержание

Цель работы	3
Задание кафедры	4
1. Часть 1	6
2. Часть 2	16
3. Часть 3	20
Выводы	22
Список литературы	23

## Цель работы

Приобрести опыт работы с процессами Linux.

# Задание кафедры

## Часть 1

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, послав сигнал `STOP`.
7. Посмотреть последовательно несколько раз `ps -f`. Записать сообщение, объяснить.
8. Убить процесс `loop2`, послав сигнал `kill -9 PID`. Записать сообщение. Прокомментировать.
9. Запустить в фоне процесс `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 2

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.
5. 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.

### **Часть 3**

1. Сгенерировать информацию — полный листинг о всех процессах системы.
2. Завершить выполнение двух процессов, владельцем которых является текущий пользователь. Первый процесс завершить с помощью сигнала SIGTERM, задав его имя, второй — с помощью сигнала SIGKILL, задав его номер.
3. Определить идентификаторы процессов, владельцем которых не является root.
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.

# 1. Часть 1

```
artem@debian-ART:~$ cd /boot
artem@debian-ART:/boot$ ls
config-5.10.0-18-amd64  initrd.img-5.10.0-18-amd64  System.map-5.10.0-19-amd64
config-5.10.0-19-amd64  initrd.img-5.10.0-19-amd64  vmlinuz-5.10.0-18-amd64
grub                  System.map-5.10.0-18-amd64  vmlinuz-5.10.0-19-amd64
artem@debian-ART:/boot$
```

Рисунок 1 – Просмотр файла с образом ядра

Номер версии линукс: 5.10.0

```
artem@debian-ART:/boot$ ps -f
UID          PID    PPID  C  STIME TTY          TIME CMD
artem        449      334  0  15:27 tty1        00:00:00 -bash
artem        472      449  0  15:30 tty1        00:00:00 ps -f
```

Рисунок 2 – Просмотр процессов

1. UID – идентификатор пользователя.
2. PID – идентификатор процесса. Он принудительно назначается планировщиком при запуске процесса.
3. PPID – идентификатор родительского процесса.
4. C – численное значение расходования ресурсов процессора в процентах.
5. STIME – это время начала процесса.
6. TTY – имя управляющего терминала - терминала, с которого запущен процесс.
7. TIME – это общее время использования процессорного времени процессом.
8. CMD – команда, которой был запущен процесс, если программа не может прочитать аргументы процесса, он будет выведен в квадратных скобках.

[illegible]

Рисунок 3 – Создание файла loop

```
"loop2" [New File]
```

Рисунок 4 – Создание файла loop2



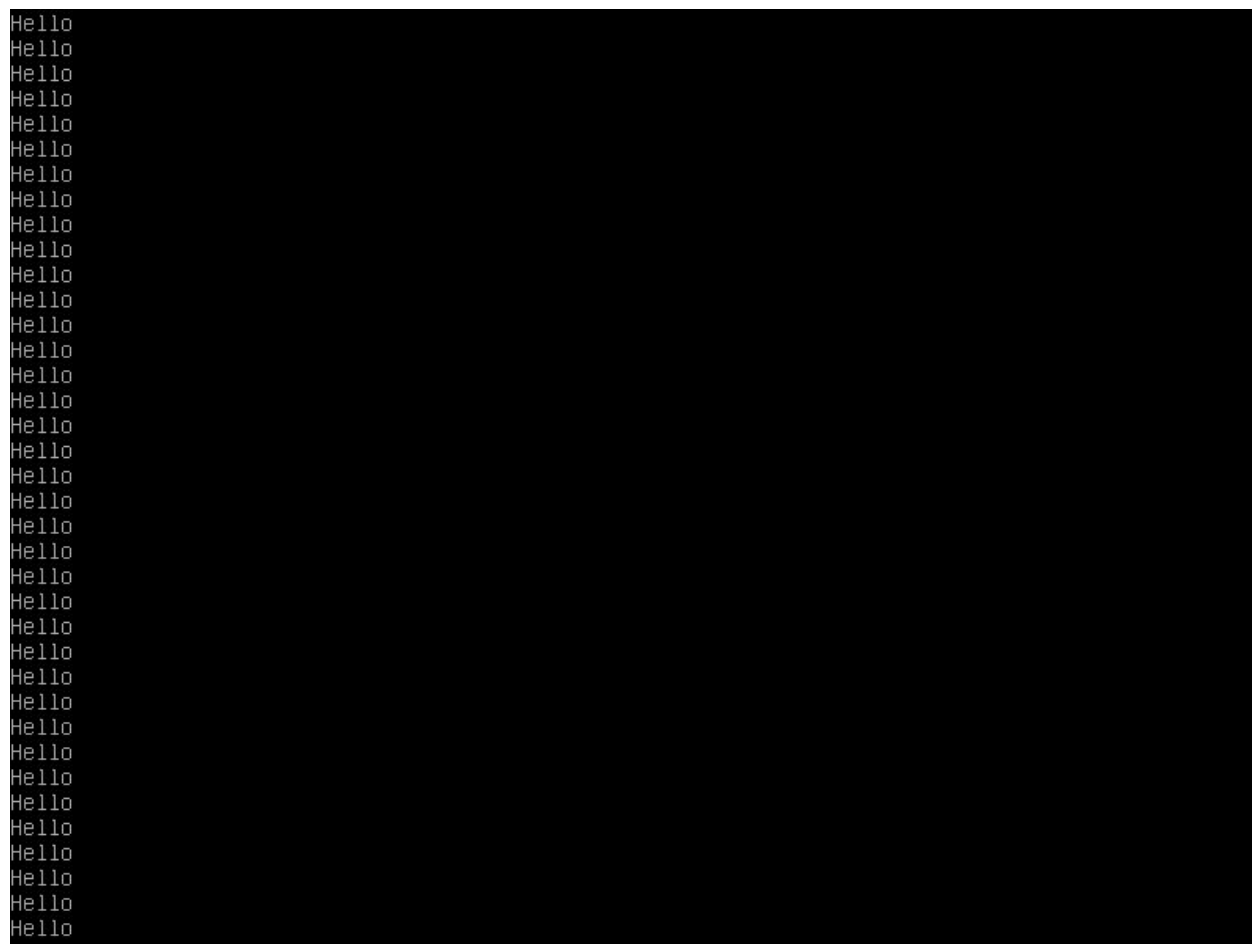


Рисунок 5 – Запуск файла loop2

[illegible]

Рисунок 6 – Остановка файла loop2

```

artem@debian-ART:~$ ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
artem    449      334  0  15:27 tty1        00:00:00 -bash
artem    555      449 22  15:49 tty1        00:00:24 sh loop2
artem    561      449  0  15:51 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
artem    449      334  0  15:27 tty1        00:00:00 -bash
artem    555      449 19  15:49 tty1        00:00:24 sh loop2
artem    562      449  0  15:51 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
artem    449      334  0  15:27 tty1        00:00:00 -bash
artem    555      449 18  15:49 tty1        00:00:24 sh loop2
artem    563      449  0  15:51 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
artem    449      334  0  15:27 tty1        00:00:00 -bash
artem    555      449 18  15:49 tty1        00:00:24 sh loop2
artem    564      449  0  15:51 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
artem    449      334  0  15:27 tty1        00:00:00 -bash
artem    555      449 17  15:49 tty1        00:00:24 sh loop2
artem    565      449  0  15:52 tty1        00:00:00 ps -f
artem@debian-ART:~$

```

Рисунок 7 – Команда ps -f несколько раз

Отличие состоит в изменении PID у "ps -f".

```

artem@debian-ART:~$ kill -9 555
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         570      449  0 15:56 tty1        00:00:00 ps -f
[1]+  Убито                  sh loop2

```

Рисунок 8 – Убийство файла loop2

```

artem@debian-ART:~$ sh loop&
[1] 572
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         572      449  99 15:58 tty1        00:00:09 sh loop
artem         573      449  0 15:58 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         572      449  99 15:58 tty1        00:00:10 sh loop
artem         574      449  0 15:58 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         572      449  99 15:58 tty1        00:00:11 sh loop
artem         575      449  0 15:58 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         572      449  99 15:58 tty1        00:00:12 sh loop
artem         576      449  0 15:58 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         572      449  99 15:58 tty1        00:00:18 sh loop
artem         577      449  0 15:58 tty1        00:00:00 ps -f

```

Рисунок 9 – Запуск в фоне loop

```

artem@debian-ART:~$ kill -15 572
artem@debian-ART:~$ ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
artem         449       334  0  15:27 tty1        00:00:00 -bash
artem         579       449  0  16:04 tty1        00:00:00 ps -f
[1]+  Завершено      sh loop
artem@debian-ART:~$ _

```

Рисунок 10 – Убийство loop

```

artem@debian-ART:~$ sh loop&
[1] 580
artem@debian-ART:~$ kill -9 580
artem@debian-ART:~$ ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
artem         449       334  0  15:27 tty1        00:00:00 -bash
artem         581       449  0  16:06 tty1        00:00:00 ps -f
[1]+  Убито          sh loop
artem@debian-ART:~$

```

Рисунок 11 – Еще один запуск loop

```

artem@debian-ART:~$ ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
artem         449       334  0  15:27 tty1        00:00:00 -bash
artem         584       449  0  16:10 tty1        00:00:00 bash
artem         587       584  0  16:11 tty1        00:00:00 ps -f

```

Рисунок 12 – Экземпляр оболочки

```

artem@debian-ART:~$ sh loop&
[1] 588
artem@debian-ART:~$ sh loop&
[2] 589
artem@debian-ART:~$ sh loop&
[3] 590
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         584      449  0 16:10 tty1        00:00:00 bash
artem         588      584 82 16:12 tty1        00:00:09 sh loop
artem         589      584 66 16:12 tty1        00:00:06 sh loop
artem         590      584 68 16:12 tty1        00:00:04 sh loop
artem         591      584  0 16:12 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         584      449  0 16:10 tty1        00:00:00 bash
artem         588      584 83 16:12 tty1        00:00:11 sh loop
artem         589      584 70 16:12 tty1        00:00:07 sh loop
artem         590      584 75 16:12 tty1        00:00:06 sh loop
artem         592      584  0 16:12 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         584      449  0 16:10 tty1        00:00:00 bash
artem         588      584 82 16:12 tty1        00:00:12 sh loop
artem         589      584 70 16:12 tty1        00:00:08 sh loop
artem         590      584 75 16:12 tty1        00:00:06 sh loop
artem         593      584  0 16:12 tty1        00:00:00 ps -f

```

Рисунок 13 – Запуск нескольких фоновых процессов

```

artem@debian-ART:~$ kill -15 588
artem@debian-ART:~$ kill -15 589
[1]  Завершено      sh loop
artem@debian-ART:~$ kill -15 590
[2]-  Завершено      sh loop
artem@debian-ART:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
artem         449      334  0 15:27 tty1        00:00:00 -bash
artem         584      449  0 16:10 tty1        00:00:00 bash
artem         595      584  0 16:14 tty1        00:00:00 ps -f
[3]+  Завершено      sh loop

```

Рисунок 14 – Остановка процессов

```

artem@debian-ART:~$ sh loop&
[1] 596
artem@debian-ART:~$ sh loop&
[2] 597
artem@debian-ART:~$ sh loop&
[3] 598
artem@debian-ART:~$ ps -f
UID          PID     PPID  C STIME TTY          TIME CMD
artem         449       334  0 15:27 tty1        00:00:00 -bash
artem         584       449  0 16:10 tty1        00:00:00 bash
artem         596       584  84 16:15 tty1        00:00:04 sh loop
artem         597       584  79 16:15 tty1        00:00:03 sh loop
artem         598       584  58 16:15 tty1        00:00:02 sh loop
artem         599       584  0 16:15 tty1        00:00:00 ps -f
artem@debian-ART:~$ ps -f
UID          PID     PPID  C STIME TTY          TIME CMD
artem         449       334  0 15:27 tty1        00:00:00 -bash
artem         584       449  0 16:10 tty1        00:00:00 bash
artem         596       584  73 16:15 tty1        00:00:08 sh loop
artem         597       584  70 16:15 tty1        00:00:07 sh loop
artem         598       584  58 16:15 tty1        00:00:05 sh loop
artem         600       584  0 16:15 tty1        00:00:00 ps -f
artem@debian-ART:~$ kill -15 596
artem@debian-ART:~$ kill -15 597
[1]  Завершено      sh loop
artem@debian-ART:~$ kill -15 598
artem@debian-ART:~$ ps -f
UID          PID     PPID  C STIME TTY          TIME CMD
artem         449       334  0 15:27 tty1        00:00:00 -bash
artem         584       449  0 16:10 tty1        00:00:00 bash
artem         601       584  0 16:15 tty1        00:00:00 ps -f
[2]-  Завершено      sh loop
[3]+  Завершено      sh loop

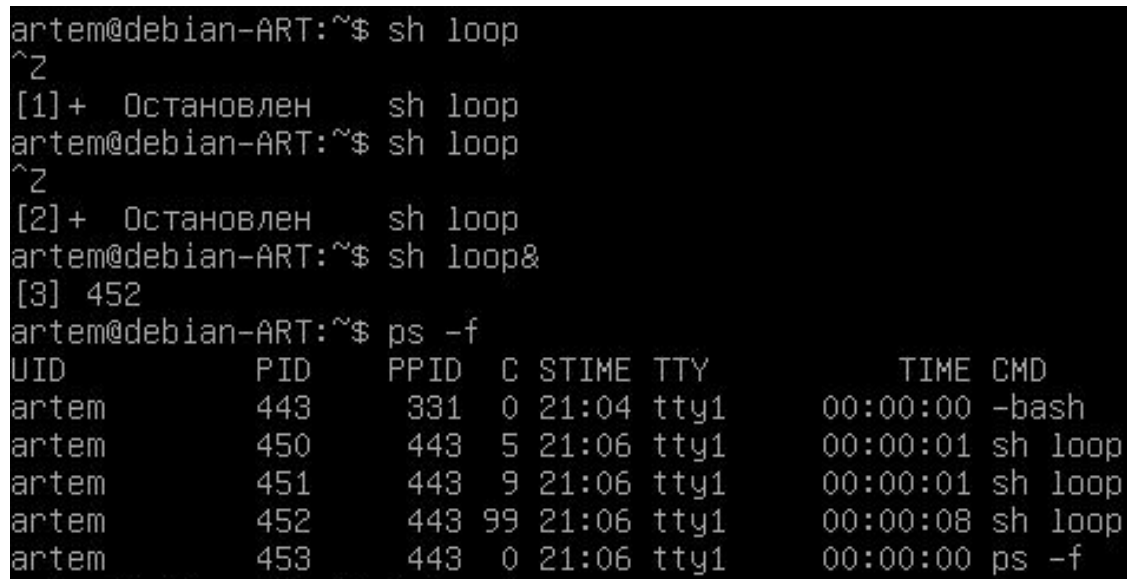
```

Рисунок 15 – Запуск и остановка

## 2. Часть 2

Запустить в консоли на выполнение три задачи, две в интерактивном режиме, одну - в фоновом.

```
sh loop
sh loop&
```



```
artem@debian-ART:~$ sh loop
^Z
[1]+  Остановлен    sh loop
artem@debian-ART:~$ sh loop
^Z
[2]+  Остановлен    sh loop
artem@debian-ART:~$ sh loop&
[3] 452
artem@debian-ART:~$ ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
artem	443	331	0	21:04	tty1	00:00:00	-bash
artem	450	443	5	21:06	tty1	00:00:01	sh loop
artem	451	443	9	21:06	tty1	00:00:01	sh loop
artem	452	443	99	21:06	tty1	00:00:08	sh loop
artem	453	443	0	21:06	tty1	00:00:00	ps -f

Рисунок 16 – Запуск 3-х задач

Перевести одну из задач, выполняющихся в интерактивном режиме, в фоновый режим.

```
bg %1
```



```
artem@debian-ART:~$ bg %1
[1]- sh loop &
```

Рисунок 17 – Перевод loop в фоновый режим

Провести эксперименты по переводу задач из фонового режима в интерактивный и наоборот.

```
bg %1
fg %1
```



```

artem@debian-ART:~$ bg %1
[1]- sh loop &
artem@debian-ART:~$ ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
artem        443       331  0  21:04 tty1        00:00:00 -bash
artem        450       443 16  21:06 tty1        00:00:25 sh loop
artem        451       443  0  21:06 tty1        00:00:01 sh loop
artem        452       443 99  21:06 tty1        00:02:24 sh loop
artem        454       443  0  21:08 tty1        00:00:00 ps -f
artem@debian-ART:~$ fg 1
sh loop
^Z
[1]+  Остановлен      sh loop

```

Рисунок 18 – Перевод в фоновый режим и наоборот

Создать именованный канал для архивирования

mkfifo chan

```

artem@debian-ART:~$ mkfifo chan
artem@debian-ART:~$ ls -l
итого 24
drwxr-xr-x  4 artem artem 4096 ноя 23 21:01 1
-rw-r--r--  1 artem artem  243 ноя 23 20:16 1.txt
-rw-r--r--  1 artem artem  300 ноя 23 20:29 2.txt
-rw-r--r--  1 artem artem  228 ноя 23 20:46 3.txt
prw-r--r--  1 artem artem    0 ноя 30 21:12 chan
-rw-r--r--  1 artem artem   26 ноя 28 15:44 loop
-rw-r--r--  1 artem artem   40 ноя 28 15:48 loop2

```

Рисунок 19 – Создание именованного канала

Осуществить передачу в канал списка файлов домашнего каталога вместе с подкаталогами

```
gzip -9 -c < chan > out.gz &  
ls -R chan
```

```
root@debian-ART:/home/artem# gzip -9 -c < chan > out.gz &  
[1] 463  
root@debian-ART:/home/artem# ls -R > chan  
[1]+  Завершён          gzip -9 -c < chan > out.gz  
root@debian-ART:/home/artem# zcat out.gz  
.:  
1  
1.txt  
2.txt  
3.txt  
chan  
loop  
loop2  
out.gz  
  
./1:  
1.txt  
21  
22  
  
./1/21:  
  
./1/22:  
31  
  
./1/22/31:  
2.txt  
3.txt  
41  
  
./1/22/31/41:
```

Рисунок 20 – Передача списка файлов домашнего каталога пользователя

Осуществить передачу в канал одного каталога вместе с файлами и подкаталогами

```
gzip -9 -c < chan > out.gz &  
tar -cvf out.tar 1 > chan
```



```
root@debian-ART:/home/artem# gzip -9 -c < chan > out.gz &  
[1] 470  
root@debian-ART:/home/artem# tar -cvf out.tar 1 > chan  
[1]+  Завершён      gzip -9 -c < chan > out.gz  
root@debian-ART:/home/artem# zcat out.gz  
1/  
1/22/  
1/22/31/  
1/22/31/2.txt  
1/22/31/3.txt  
1/22/31/41/  
1/1.txt  
1/21/
```

Рисунок 21 – Передача подкаталога с подкаталогами и файлами

### 3. Часть 3

Сгенерировать информацию — полный листинг о всех процессах системы.

`ps -eF`

```
root      123      2 0      0      0 0 14:49 ?      00:00:00 [scsi_eh_2]
root      124      2 0      0      0 0 14:49 ?      00:00:00 [scsi_tmf_2]
root      130      2 0      0      0 0 14:49 ?      00:00:00 [kworker/0:3-mm_percpu_wq]
root      159      2 0      0      0 1 14:49 ?      00:00:00 [jbd2/sda1-8]
root      160      2 0      0      0 0 14:49 ?      00:00:00 [ext4-rsv-conver]
root      194      1 0 14148 21884 1 14:49 ?      00:00:00 /lib/systemd/systemd-journald
root      218      1 0 5398 5180 1 14:49 ?      00:00:00 /lib/systemd/systemd-udevd
systemd+  266      1 0 22110 5968 0 14:49 ?      00:00:00 /lib/systemd/systemd-timesyncd
root      271      2 0      0      0 0 14:49 ?      00:00:00 [cryptd]
root      280      2 0      0      0 1 14:49 ?      00:00:00 [irq/18-vmwgfx]
root      282      2 0      0      0 0 14:49 ?      00:00:00 [ttm_swap]
root      284      2 0      0      0 0 14:49 ?      00:00:00 [card0-crtc0]
root      285      2 0      0      0 0 14:49 ?      00:00:00 [card0-crtc1]
root      286      2 0      0      0 1 14:49 ?      00:00:00 [card0-crtc2]
root      287      2 0      0      0 0 14:49 ?      00:00:00 [card0-crtc3]
root      288      2 0      0      0 0 14:49 ?      00:00:00 [card0-crtc4]
root      289      2 0      0      0 1 14:49 ?      00:00:00 [card0-crtc5]
root      292      2 0      0      0 1 14:49 ?      00:00:00 [card0-crtc6]
root      295      2 0      0      0 1 14:49 ?      00:00:00 [card0-crtc7]
root      296      1 0 1687 2668 0 14:49 ?      00:00:00 /usr/sbin/cron -f
message+  298      1 0 2069 4088 0 14:49 ?      00:00:00 /usr/bin/dbus-daemon --system --
root      300      1 0 55200 6080 0 14:49 ?      00:00:00 /usr/sbin/rsyslogd -n -iNONE
root      304      1 0 5512 7024 1 14:49 ?      00:00:00 /lib/systemd/systemd-logind
root      351      1 0 24972 5568 1 14:49 ?      00:00:00 /sbin/dhclient -4 -v -i -pf /run
root      367      1 0 3339 6916 0 14:49 ?      00:00:00 sshd: /usr/sbin/sshd -D [listene
root      596      2 0      0      0 1 15:51 ?      00:00:00 [kworker/1:1-events]
root      602      1 0 1826 4304 1 15:55 tty1    00:00:00 /bin/login -p --
root      603      2 0      0      0 0 15:56 ?      00:00:00 [kworker/u4:2-events_unbound]
root      605      2 0      0      0 0 15:56 ?      00:00:00 [kworker/0:1-ata_sff]
artem     611      1 0 3795 8252 1 15:56 ?      00:00:00 /lib/systemd/systemd --user
artem     612      611 0 25305 2560 1 15:56 ?      00:00:00 (sd-pam)
artem     617      602 0 2011 4668 1 15:56 tty1    00:00:00 -bash
root      633      2 0      0      0 1 16:01 ?      00:00:00 [kworker/1:0-ata_sff]
root      634      2 0      0      0 1 16:07 ?      00:00:00 [kworker/1:2-ata_sff]
root      638      2 0      0      0 0 16:09 ?      00:00:00 [kworker/0:0-ata_sff]
artem     642      617 0 2440 3220 0 16:10 tty1    00:00:00 ps -eF
artem@debian-ART:~$ _
```

Рисунок 22 – Передача подкаталога с подкаталогами и файлами

Завершить выполнение двух процессов, владельцем которых является текущий пользователь. Первый процесс завершить с помощью сигнала SIGTERM, задав его имя, второй — с помощью сигнала SIGKILL, задав его номер.

`kill -SIGTERM 658`

`kill -SIGKILL 662`

```

artem@debian-ART:~$ kill -SIGTERM 658
artem@debian-ART:~$ ps -f

```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
artem	617	602	0	15:56	tty1	00:00:00	-bash
artem	643	617	0	16:12	tty1	00:00:00	bash
artem	647	643	0	16:13	tty1	00:00:00	bash
artem	660	647	0	16:18	tty1	00:00:00	ps -f

```

[1]+  Завершено      sh loop

```

Рисунок 23 – Передача подкаталога с подкаталогами и файлами

```

artem@debian-ART:~$ sh loop&
[1] 662
artem@debian-ART:~$ kill -SIGKILL 662
artem@debian-ART:~$ ps -f

```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
artem	617	602	0	15:56	tty1	00:00:00	-bash
artem	643	617	0	16:12	tty1	00:00:00	bash
artem	647	643	0	16:13	tty1	00:00:00	bash
artem	663	647	0	16:20	tty1	00:00:00	ps -f

```

[1]+  Убито          sh loop

```

Рисунок 24 – Передача подкаталога с подкаталогами и файлами

Определить идентификаторы процессов, владельцем которых не является root.

Если смотреть по 1 скриншоту 3 части задания, то можно сказать, что это идентификаторы: 266, 298, 611, 612, 617, 642.

## Выводы

В ходе данной лабораторной работы я изучил команды Linux.

## Список литературы

- [1] Львовский, С.М. Набор и верстка в системе  $\text{\LaTeX}$  [Текст] / С.М. Львовский. М.: МЦНМО, 2006. — 448 с.