

Липецкий государственный технический университет

Кафедра прикладной математики

Отчет по лабораторной работе № 3
«Процессы в операционной системе Linux»
по курсу «Операционная система Linux»

Студент

подпись, дата

Суханов Д.И.
фамилия, инициалы

Группа

Руководитель

Доцент, к. пед. наук
ученая степень, ученое звание

подпись, дата

Кургасов В.В.
фамилия, инициалы

Липецк 2021 г.

Содержание

Цель работы	3
Задание кафедры	4
Цель работы	4
Часть 1	4
Часть 2	5
Часть 3	5
Ход работы	6
Часть 1	6
Часть 2	14
Часть 3. Вариант 10.	18
Часть 4.	21
Выводы	25
1. Контрольные вопросы	26
Контрольные вопросы	26

Цель работы

Ознакомиться на практике с понятием процесса в операционной системе. Приобрести опыт и навыки управления процессами в операционной системе 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`: `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 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`) и 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. Вариант 10

1. Вывести информацию о состоянии процессов системы в реальном режиме с сортировкой по PID. Завершить один из процессов, владельцем которого является текущий пользователь, не выходя из команды.
2. С помощью сигнала `SIGKILL` завершить самый новый процесс, владельцем которого является текущий пользователь.
3. Определите информацию о работающем в системе пользователе (имя, время входа в систему, время простоя и т.д.).

Ход работы

Часть 1

1. Загрузка не root-ом, а пользователем. Поиск файла с образом ядра.

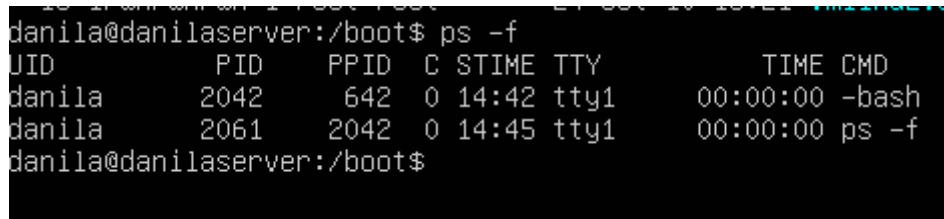
config-5.4.0-88-generic

```
danila@danilaserver:~$ cd /boot
danila@danilaserver:/boot$ ls -li
total 98808
 13 -rw-r--r-- 1 root root    237883 Sep 23 16:40 config-5.4.0-88-generic
8193 drwxr-xr-x 4 root root      4096 Oct 10 16:22 grub
 18 lrwxrwxrwx 1 root root        27 Oct 10 16:21 initrd.img -> initrd.img-5.4.0-88-generic
312 -rw-r--r-- 1 root root 84381291 Oct 10 16:23 initrd.img-5.4.0-88-generic
 16 lrwxrwxrwx 1 root root        27 Oct 10 16:21 initrd.img.old -> initrd.img-5.4.0-88-generic
 11 drwx----- 2 root root    16384 Oct 10 15:49 lost+found
 12 -rw----- 1 root root 4753783 Sep 23 16:40 System.map-5.4.0-88-generic
 17 lrwxrwxrwx 1 root root        24 Oct 10 16:21 vmlinuz -> vmlinuz-5.4.0-88-generic
 14 -rw----- 1 root root 11776256 Sep 23 16:41 vmlinuz-5.4.0-88-generic
 15 lrwxrwxrwx 1 root root        24 Oct 10 16:21 vmlinuz.old -> vmlinuz-5.4.0-88-generic
danila@danilaserver:/boot$
```

Рисунок 1 – Результат поиска

2. Просмотр процессов `ps -f`.

`ps -f`



```
danila@danilaserver:/boot$ ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
danila       2042      642  0  14:42 tty1        00:00:00 -bash
danila       2061     2042  0  14:45 tty1        00:00:00 ps -f
danila@danilaserver:/boot$
```

Рисунок 2 – Процессы

`-f` Генерировать полный листинг.

`-l` Генерировать листинг в длинном формате.

UID (f,l) Идентификатор владельца процесса; при указании опции `-f` выдается входное имя пользователя.

PID Идентификатор процесса (необходим для терминирования процесса).

PPID(f,l) Идентификатор родительского процесса.

C (f,l) Доля выделенного планировщиком времени ЦП.

STIME (f) Время запуска процесса (часы:минуты:секунды). Если процесс запущен более чем 24 часа назад, выдается месяц и день запуска.

PRI (l) Приоритет процесса; большее число означает меньший приоритет.

NI (l) Поправка к приоритету.

ADDR (l) Адрес процесса в памяти.

SZ (l) Размер (в блоках по 512 байт) образа процесса в памяти.

WCHAN (l) Адрес события, которого ожидает процесс. У активного процесса эта колонка пуста.

TTY Управляющий терминал. Если такового нет, выдается символ ?.

TIME Истраченное процессом время ЦП.

COMMAND Имя программы; если указана опция `-f`, то выводится полное имя команды и ее аргументы.

3. Запись с помощью редактора vi двух сценариев loop и loop2. Текст сценариев:

```
Loop: while true; do true; done
```

```
Loop2: while true; do true; echo 'Hello'; done
```

```
root@danilaserver:/# cat Loop
while true; do true; done
root@danilaserver:/# cat Loop2
while true; do true; echo 'Hello'; done
root@danilaserver:/#
```

Рисунок 3 – Содержимое файлов

4. Запуск и остановка loop2 на переднем плане: sh Loop2

```
sh Loop2
```

```
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
^Z
[2]+  Stopped                  sh Loop2
root@danilaserver:/# kill -1
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
root@danilaserver:/#
```

Рисунок 4 – Запуск и остановка

5. Посмотр несколько раз списка процессов.

```
ps -f
```

```
root@danilaserver:/# ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
root          642        1  0  14:38 tty1        00:00:00 /bin/login -p --
root         2085       2042  0  14:59 tty1        00:00:00 sudo su
root         2089       2085  0  14:59 tty1        00:00:00 su
root         2090       2089  0  14:59 tty1        00:00:00 bash
root         2099       2090  0  15:00 tty1        00:00:00 vi Loop2
root         2104       2090  0  15:04 tty1        00:00:00 sh Loop2
root         2112       2090  0  15:07 tty1        00:00:00 ps -f
root@danilaserver:/# ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
root          642        1  0  14:38 tty1        00:00:00 /bin/login -p --
root         2085       2042  0  14:59 tty1        00:00:00 sudo su
root         2089       2085  0  14:59 tty1        00:00:00 su
root         2090       2089  0  14:59 tty1        00:00:00 bash
root         2099       2090  0  15:00 tty1        00:00:00 vi Loop2
root         2104       2090  0  15:04 tty1        00:00:00 sh Loop2
root         2113       2090  0  15:07 tty1        00:00:00 ps -f
root@danilaserver:/# ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
root          642        1  0  14:38 tty1        00:00:00 /bin/login -p --
root         2085       2042  0  14:59 tty1        00:00:00 sudo su
root         2089       2085  0  14:59 tty1        00:00:00 su
root         2090       2089  0  14:59 tty1        00:00:00 bash
root         2099       2090  0  15:00 tty1        00:00:00 vi Loop2
root         2104       2090  0  15:04 tty1        00:00:00 sh Loop2
root         2114       2090  0  15:07 tty1        00:00:00 ps -f
root@danilaserver:/# _
```

Рисунок 5 – Поочередный просмотр списка процессов.

Изменяется время работы loop и идентификатор ps, так как каждый последующий вызов заменяет предыдущие.

6. Уничтожение процесса loop2 сигналом kill.

```
kill -9 2104
```

```
root@danilaserver:/# ps -f
UID          PID     PPID  C STIME TTY          TIME CMD
root          642        1   0 14:38 tty1        00:00:00 /bin/login -p --
root         2085       2042   0 14:59 tty1        00:00:00 sudo su
root         2089       2085   0 14:59 tty1        00:00:00 su
root         2090       2089   0 14:59 tty1        00:00:00 bash
root         2104       2090   0 15:04 tty1        00:00:00 sh Loop2
root         2120       2090   0 15:10 tty1        00:00:00 ps -f
root@danilaserver:/# kill -9 2104
[2]+  Killed                  sh Loop2
root@danilaserver:/# ps -f
UID          PID     PPID  C STIME TTY          TIME CMD
root          642        1   0 14:38 tty1        00:00:00 /bin/login -p --
root         2085       2042   0 14:59 tty1        00:00:00 sudo su
root         2089       2085   0 14:59 tty1        00:00:00 su
root         2090       2089   0 14:59 tty1        00:00:00 bash
root         2121       2090   0 15:10 tty1        00:00:00 ps -f
root@danilaserver:/#
```

Рисунок 6 – Результат уничтожения.

Процесс исчез из списка после его уничтожения.

7. Запуск в фоне процесса loop. Просмотр несколько раз списка процессов. Завершение процесса loop командой kill. PID. Записать сообщение, прокомментировать.

```
sh Loop&
ps -f
kill -15
```

```
root@danilaserver:/# sh Loop&
[1] 2135
root@danilaserver:/# ps -f
  UID          PID    PPID  C  STIME TTY          TIME CMD
  root           642        1   0   14:38 tty1        00:00:00 /bin/login -p --
  root          2085       2042   0   14:59 tty1        00:00:00 sudo su
  root          2089       2085   0   14:59 tty1        00:00:00 su
  root          2090       2089   0   14:59 tty1        00:00:00 bash
  root          2135       2090  99   15:20 tty1        00:00:05 sh Loop
  root          2136       2090   0   15:20 tty1        00:00:00 ps -f
root@danilaserver:/# ps -f
  UID          PID    PPID  C  STIME TTY          TIME CMD
  root           642        1   0   14:38 tty1        00:00:00 /bin/login -p --
  root          2085       2042   0   14:59 tty1        00:00:00 sudo su
  root          2089       2085   0   14:59 tty1        00:00:00 su
  root          2090       2089   0   14:59 tty1        00:00:00 bash
  root          2135       2090  99   15:20 tty1        00:00:08 sh Loop
  root          2137       2090   0   15:20 tty1        00:00:00 ps -f
root@danilaserver:/# ps -f
  UID          PID    PPID  C  STIME TTY          TIME CMD
  root           642        1   0   14:38 tty1        00:00:00 /bin/login -p --
  root          2085       2042   0   14:59 tty1        00:00:00 sudo su
  root          2089       2085   0   14:59 tty1        00:00:00 su
  root          2090       2089   0   14:59 tty1        00:00:00 bash
  root          2135       2090  99   15:20 tty1        00:00:13 sh Loop
  root          2138       2090   0   15:20 tty1        00:00:00 ps -f
root@danilaserver:/# kill -15 2135
root@danilaserver:/# ps -f
  UID          PID    PPID  C  STIME TTY          TIME CMD
  root           642        1   0   14:38 tty1        00:00:00 /bin/login -p --
  root          2085       2042   0   14:59 tty1        00:00:00 sudo su
  root          2089       2085   0   14:59 tty1        00:00:00 su
  root          2090       2089   0   14:59 tty1        00:00:00 bash
  root          2139       2090   0   15:21 tty1        00:00:00 ps -f
[1]+  Terminated                  sh Loop
root@danilaserver:/#
```

Рисунок 7 – Результат запуска и удаления процесса

Процесс завершился.

8. Третий запуск в фоне. Уничтожение, не останавливая, командой kill

kill -9 PID

```
root@danilaserver:/# sh Loop&
[1] 2143
root@danilaserver:/# ps -f
UID      PID    PPID  C  STIME TTY          TIME CMD
root      642      1    0 14:38 tty1        00:00:00 /bin/login -p --
root     2085    2042    0 14:59 tty1        00:00:00 sudo su
root     2089    2085    0 14:59 tty1        00:00:00 su
root     2090    2089    0 14:59 tty1        00:00:00 bash
root     2143    2090   99 15:23 tty1        00:00:04 sh Loop
root     2144    2090    0 15:23 tty1        00:00:00 ps -f
root@danilaserver:/# kill -9 2143
root@danilaserver:/# ps -f
UID      PID    PPID  C  STIME TTY          TIME CMD
root      642      1    0 14:38 tty1        00:00:00 /bin/login -p --
root     2085    2042    0 14:59 tty1        00:00:00 sudo su
root     2089    2085    0 14:59 tty1        00:00:00 su
root     2090    2089    0 14:59 tty1        00:00:00 bash
root     2145    2090    0 15:23 tty1        00:00:00 ps -f
[1]+  Killed                  sh Loop
root@danilaserver:/# _
```

Рисунок 8 – Третий запуск и удаление

9. Запуск еще одного экземпляра оболочки: `bash`. Запуск несколько процессов в фоне. Остановка и запускать. Запись результата просмотра командой `ps -f`.

```
kill -19 PID
```

```
ps -f
```

```
root@danilaserver:/# ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
root      642       1   0  14:38 tty1        00:00:00 /bin/login -p --
root     2085    2042   0  14:59 tty1        00:00:00 sudo su
root     2089    2085   0  14:59 tty1        00:00:00 su
root     2090    2089   0  14:59 tty1        00:00:00 bash
root     2156    2090   3  15:34 tty1        00:00:03 sh Loop
root     2162    2090  98  15:35 tty1        00:00:43 sh Loop
root     2164    2090  13  15:35 tty1        00:00:00 bash Loop
root     2166    2090   0  15:35 tty1        00:00:00 ps -f
root@danilaserver:/# kill -19 2162
root@danilaserver:/# ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
root      642       1   0  14:38 tty1        00:00:00 /bin/login -p --
root     2085    2042   0  14:59 tty1        00:00:00 sudo su
root     2089    2085   0  14:59 tty1        00:00:00 su
root     2090    2089   0  14:59 tty1        00:00:00 bash
root     2156    2090   2  15:34 tty1        00:00:03 sh Loop
root     2162    2090  93  15:35 tty1        00:01:04 sh Loop
root     2164    2090   2  15:35 tty1        00:00:00 bash Loop
root     2167    2090   0  15:36 tty1        00:00:00 ps -f

[3]+  Stopped                  sh Loop
root@danilaserver:/# kill -19 2162
root@danilaserver:/# ps -f
UID      PID     PPID  C  STIME TTY          TIME CMD
root      642       1   0  14:38 tty1        00:00:00 /bin/login -p --
root     2085    2042   0  14:59 tty1        00:00:00 sudo su
root     2089    2085   0  14:59 tty1        00:00:00 su
root     2090    2089   0  14:59 tty1        00:00:00 bash
root     2156    2090   2  15:34 tty1        00:00:03 sh Loop
root     2162    2090  69  15:35 tty1        00:01:04 sh Loop
root     2164    2090   1  15:35 tty1        00:00:00 bash Loop
root     2168    2090   0  15:36 tty1        00:00:00 ps -f
root@danilaserver:/# _
```

Рисунок 9 – Неоднократная остановка и запуск процессов.

Часть 2

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

```
sh Loop
sh Loop
sh Loop$
jobs
```

```
root@danilaserver:/# sh Loop
^Z
[1]+  Stopped                  sh Loop
root@danilaserver:/# sh Loop
^Z
[2]+  Stopped                  sh Loop
root@danilaserver:/# sh Loop&
[3] 2174
root@danilaserver:/# ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
root         642      1    0 14:38 tty1        00:00:00 /bin/login -p --
root        2085    2042    0 14:59 tty1        00:00:00 sudo su
root        2089    2085    0 14:59 tty1        00:00:00 su
root        2090    2089    0 14:59 tty1        00:00:00 bash
root        2171    2090    0 15:38 tty1        00:00:00 sh Loop
root        2173    2090    3 15:38 tty1        00:00:00 sh Loop
root        2174    2090   99 15:38 tty1        00:00:17 sh Loop
root        2175    2090    0 15:39 tty1        00:00:00 ps -f
root@danilaserver:/# jobs
[1]-  Stopped                  sh Loop
[2]+  Stopped                  sh Loop
[3]   Running                  sh Loop &
root@danilaserver:/#
```

Рисунок 10 – Запуск в разных режимах.

2. Перевод задачи, выполняющуюся в интерактивном режиме, в фоновый режим.

```
bg 2
bg 1
jobs
```

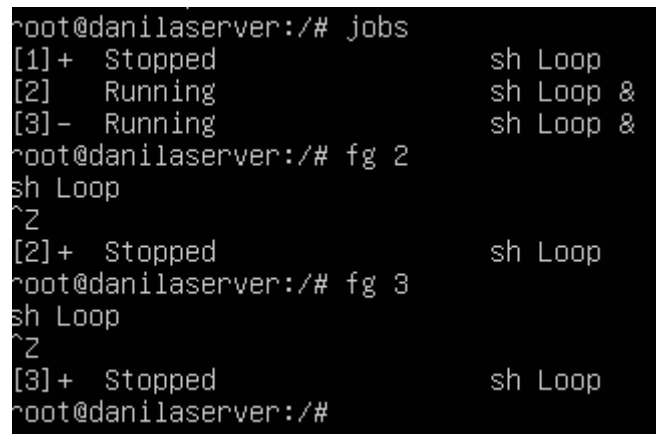


```
root@danilaserver:/# jobs
[1]-  Stopped                  sh Loop
[2]+  Stopped                  sh Loop
[3]   Running                  sh Loop &
root@danilaserver:/# bg 2
[2]+  sh Loop &
root@danilaserver:/# jobs
[1]+  Stopped                  sh Loop
[2]   Running                  sh Loop &
[3]-  Running                  sh Loop &
root@danilaserver:/#
```

Рисунок 11 – Результат перевода.

3. Проведение эксперимента по переводу задач из фонового режима в интерактивный и наоборот.

```
jobs
fg 2
fg 3
jobs
```

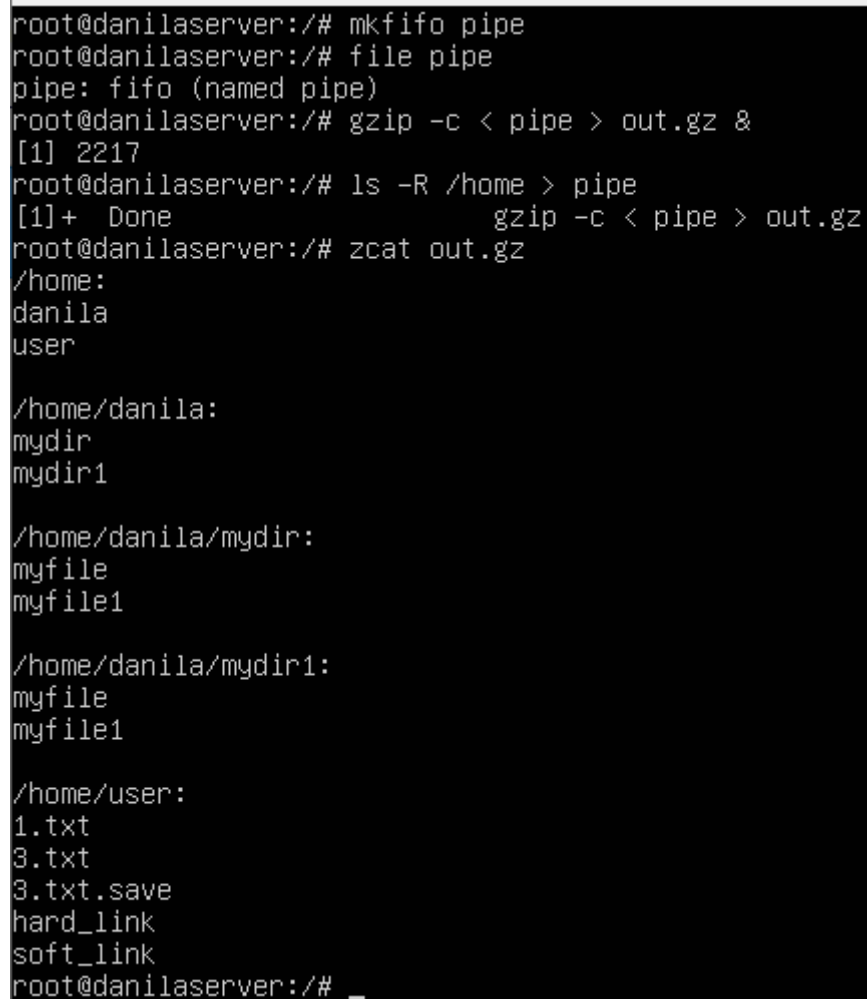


```
root@danilaserver:/# jobs
[1]+  Stopped                  sh Loop
[2]   Running                  sh Loop &
[3]-  Running                  sh Loop &
root@danilaserver:/# fg 2
sh Loop
^Z
[2]+  Stopped                  sh Loop
root@danilaserver:/# fg 3
sh Loop
^Z
[3]+  Stopped                  sh Loop
root@danilaserver:/#
```

Рисунок 12 – Результат эксперимента.

4. Создание именованного канала для архивирования и осуществление передачи в канал списка файлов домашнего каталога вместе с подкаталогами и одного каталога вместе с файлами и подкаталогами.

```
mkfifo pipe
file pipe
gzip -c < pipe > out.gz $
ls -R /home > pipe
zcat out.gz
gzip -c < pipe > outNEW.gz $
tar -cvf outNEW.tar /home > pipe
```



```
root@danilaserver:/# mkfifo pipe
root@danilaserver:/# file pipe
pipe: fifo (named pipe)
root@danilaserver:/# gzip -c < pipe > out.gz &
[1] 2217
root@danilaserver:/# ls -R /home > pipe
[1]+  Done                  gzip -c < pipe > out.gz
root@danilaserver:/# zcat out.gz
/home:
danila
user

/home/danila:
mydir
mydir1

/home/danila/mydir:
myfile
myfile1

/home/danila/mydir1:
myfile
myfile1

/home/user:
1.txt
3.txt
3.txt.save
hard_link
soft_link
root@danilaserver:/# _
```

Рисунок 13 – Архивирование одного каталога вместе с файлами и подкаталогами.

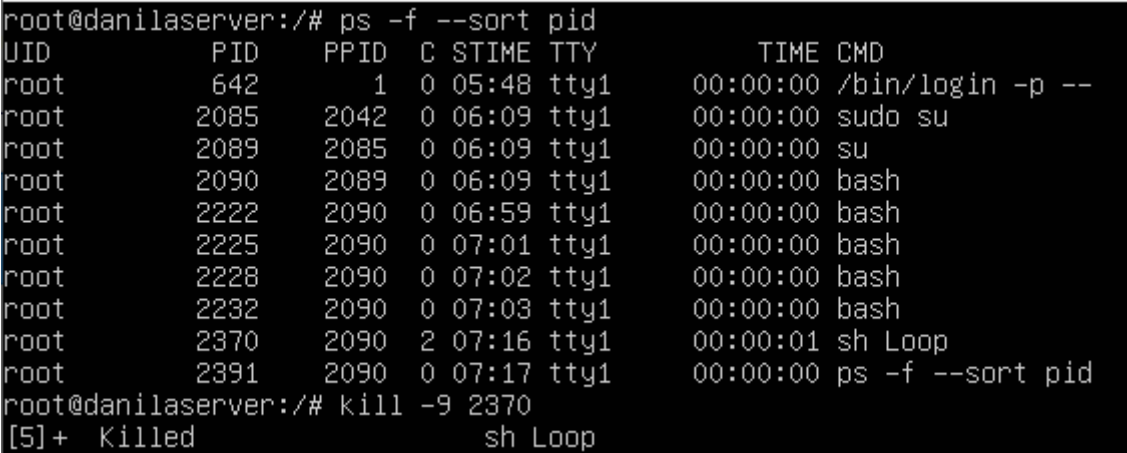

```
root@danilaserver:/# gzip -c < pipe > outNEW.gz &  
[5] 2234  
root@danilaserver:/# tar -cvf outNEW.tar /home > pipe  
tar: Removing leading `/' from member names  
[5] Done  
gzip -c < pipe > outNEW.gz  
root@danilaserver:/# _
```

Рисунок 14 – Архивирование одного каталога вместе с файлами и подкаталогами - 2

Часть 3. Вариант 10.

1. Вывод информации о состоянии процессов системы в реальном режиме с сортировкой по PID. Завершение одного из процессов, владельцем которого является текущий пользователь, не выходя из команды.

```
ps -f --sort pid  
kill -9 2370
```



```
root@danilaserver:/# ps -f --sort pid  
UID      PID     PPID  C  STIME TTY          TIME CMD  
root      642        1  0  05:48 tty1        00:00:00 /bin/login -p --  
root     2085     2042  0  06:09 tty1        00:00:00 sudo su  
root     2089     2085  0  06:09 tty1        00:00:00 su  
root     2090     2089  0  06:09 tty1        00:00:00 bash  
root     2222     2090  0  06:59 tty1        00:00:00 bash  
root     2225     2090  0  07:01 tty1        00:00:00 bash  
root     2228     2090  0  07:02 tty1        00:00:00 bash  
root     2232     2090  0  07:03 tty1        00:00:00 bash  
root     2370     2090  2  07:16 tty1        00:00:01 sh Loop  
root     2391     2090  0  07:17 tty1        00:00:00 ps -f --sort pid  
root@danilaserver:/# kill -9 2370  
[5]+  Killed                  sh Loop
```

Рисунок 15 – Сортировка и завершение процесса.

2. Завершение самого нового процесса с помощью сигнала SIGKILL, владельцем которого является текущий пользователь.

```
ps -f --sort time  
kill -9 985
```

```
danila@danilaserver:/$ ps -f --sort time  
UID      PID     PPID  C  STIME TTY      TIME CMD  
danila    935      644  0  07:34 tty1    00:00:00 -bash  
danila   1032      935  0  07:45 tty1    00:00:00 ps -f --sort time  
danila    983      935  0  07:36 tty1    00:00:01 sh Loop  
danila    985      935 99  07:37 tty1    00:07:46 sh Loop  
danila@danilaserver:/$ kill -9 985  
danila@danilaserver:/$ ps -f --sort time  
UID      PID     PPID  C  STIME TTY      TIME CMD  
danila    935      644  0  07:34 tty1    00:00:00 -bash  
danila   1033      935  0  07:45 tty1    00:00:00 ps -f --sort time  
danila    983      935  0  07:36 tty1    00:00:01 sh Loop  
[2]-  Killed                  sh Loop  
danila@danilaserver:/$
```

Рисунок 16 – Результат эксперимента.

3. Определение информации о работающем в системе пользователе (имя, время входа в систему, время простоя и т.д.).

```
last
last danila
lastb
```

```
root@danilaserver:/# last
danila  tty1          Fri Nov 12 07:34    still logged in
reboot  system boot      5.4.0-88-generic  Fri Nov 12 07:33    still running
danila  tty1          Thu Nov 11 14:42    - crash (16:51)
reboot  system boot      5.4.0-88-generic  Thu Nov 11 14:38    still running
danila  tty1          Fri Oct 15 17:46    - crash (26+20:52)
danila  tty1          Fri Oct 15 17:45    - 17:45 (00:00)
danila  tty1          Fri Oct 15 16:42    - 17:45 (01:02)
danila  tty1          Sun Oct 10 16:35    - 16:41 (5+00:06)
reboot  system boot      5.4.0-88-generic  Sun Oct 10 16:33    still running

wtmp begins Sun Oct 10 16:33:59 2021
root@danilaserver:/#
```

Рисунок 17 – Вывод информации.

```
root@danilaserver:/# last danila
danila  tty1          Fri Nov 12 07:34    still logged in
danila  tty1          Thu Nov 11 14:42    - crash (16:51)
danila  tty1          Fri Oct 15 17:46    - crash (26+20:52)
danila  tty1          Fri Oct 15 17:45    - 17:45 (00:00)
danila  tty1          Fri Oct 15 16:42    - 17:45 (01:02)
danila  tty1          Sun Oct 10 16:35    - 16:41 (5+00:06)

wtmp begins Sun Oct 10 16:33:59 2021
root@danilaserver:/# _
```

Рисунок 18 – Вывод информации о конкретном пользователе.

```
root@danilaserver:/# lastb
root    tty1          Tue Oct 12 04:13    - 04:13 (00:00)
root    tty1          Tue Oct 12 04:09    - 04:09 (00:00)
root    tty1          Tue Oct 12 04:09    - 04:09 (00:00)

btmp begins Tue Oct 12 04:09:16 2021
root@danilaserver:/# _
```

Рисунок 19 – Просмотр неудачных попыток входа в систему.

Часть 4.

1. Запустить программу виртуализации Oracle VM VirtualBox.
2. Запустить виртуальную машину Ubuntu.
3. Открыть окно интерпретатора команд.
4. Вывести общую информацию о системе

- (a) Вывести информацию о текущем интерпретаторе команд.

```
echo $SHELL
```

- (b) Вывести информацию о текущем пользователе.

```
whoami
```

- (c) Вывести информацию о текущем каталоге.

```
pwd
```

- (d) Вывести информацию об оперативной памяти и области подкачки.

```
free
```

- (e) Вывести информацию о дисковой памяти.

```
df
```

5. Выполнить команды получения информации о процессах

- (a) Получить идентификатор текущего процесса(PID).

```
echo $$
```

- (b) Получить идентификатор родительского процесса(PPID).

```

danila@danilaserver:/$ echo $SHELL
/bin/bash
danila@danilaserver:/$ whoami
danila
danila@danilaserver:/$ pwd
/
danila@danilaserver:/$ free

```

	total	used	free	shared	buff/cache	available
Mem:	1004796	198912	435164	1056	370720	653780
Swap:	1779708	0	1779708			

```

danila@danilaserver:/$ df

```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev	457244	0	457244	0%	/dev
tmpfs	100480	1056	99424	2%	/run
/dev/mapper/ubuntu--vg-ubuntu--lv	9219412	4400456	4330920	51%	/
tmpfs	502396	0	502396	0%	/dev/shm
tmpfs	5120	0	5120	0%	/run/lock
tmpfs	502396	0	502396	0%	/sys/fs/cgroup
/dev/loop1	63360	63360	0	100%	/snap/core20/1169
/dev/loop2	33280	33280	0	100%	/snap/snapd/13640
/dev/loop0	56832	56832	0	100%	/snap/core18/2128
/dev/loop3	68864	68864	0	100%	/snap/lxd/21835
/dev/loop4	68864	68864	0	100%	/snap/lxd/21545
/dev/loop5	56832	56832	0	100%	/snap/core18/2246
/dev/loop6	33152	33152	0	100%	/snap/snapd/13170
/dev/sda2	999320	108572	821936	12%	/boot
tmpfs	100476	0	100476	0%	/run/user/1000

```

danila@danilaserver:/$ _

```

Рисунок 20 – Общая информация о системе.

```
echo $PPID
```

- (с) Получить идентификатор процесса инициализации системы.

```
pidof init
```

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

```
ps
```

- (е) Отобразить все процессы.

```
ps -e
```

6. Выполнить команды управления процессами.

- (а) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе.

```
ps
```

```

danila@danilaserver:/$ echo $$
935
danila@danilaserver:/$ echo $PPID
644
danila@danilaserver:/$ pidof init
1
danila@danilaserver:/$ ps
  PID TTY          TIME CMD
   935 tty1      00:00:00 bash
   983 tty1      00:00:01 sh
  1199 tty1      00:00:00 ps
danila@danilaserver:/$

```

Рисунок 21 – Получение информации о процессах.

```

527 ?          00:00:00 loop0
531 ?          00:00:00 loop1
534 ?          00:00:00 loop2
535 ?          00:00:00 loop3
537 ?          00:00:00 loop4
538 ?          00:00:00 loop5
539 ?          00:00:00 loop6
541 ?          00:00:00 jbd2/sda2-8
543 ?          00:00:00 ext4-rsv-conver
553 ?          00:00:00 systemd-timesyn
603 ?          00:00:00 systemd-network
605 ?          00:00:00 systemd-resolve
617 ?          00:00:00 accounts-daemon
620 ?          00:00:00 cron
621 ?          00:00:00 dbus-daemon
626 ?          00:00:00 networkd-dispat
627 ?          00:00:00 rsyslogd
629 ?          00:00:02 snapd
630 ?          00:00:00 systemd-logind
632 ?          00:00:00 udisksd
633 ?          00:00:00 atd
643 ?          00:00:00 kworker/0:5-events
644 tty1        00:00:00 login
670 ?          00:00:00 unattended-upgr
675 ?          00:00:00 polkitd
929 ?          00:00:00 systemd
930 ?          00:00:00 (sd-pam)
935 tty1        00:00:00 bash
983 tty1        00:00:01 sh
1035 ?         00:00:00 kworker/u2:1-events_power_efficient
1071 ?         00:00:00 kworker/0:0-cgroup_destroy
1100 ?         00:00:01 kworker/0:2-events
1113 ?         00:00:01 fwupd
1125 ?         00:00:00 upowerd
1179 ?         00:00:00 kworker/u2:0
1201 tty1        00:00:00 ps
danila@danilaserver:/$

```

Рисунок 22 – Все процессы.

(b) Определить текущее значение nice по умолчанию.

```
nice
```

(c) Запустить интерпретатор bash с понижением приоритета nice -n 10 bash

```
nice -n 10 bash
```

(d) Определить PID запущенного интерпретатора.

```
echo $$
```

(e) Установить приоритет запущенного интерпретатора равным 5

```
renice -n 5 -p <PID процесса>
```

(f) Получить информацию о процессах bash

```
ps lax | grep bash
```

```
danila@danilaserver:/$ ps
  PID TTY          TIME CMD
   935 tty1      00:00:00 bash
   983 tty1      00:00:01 sh
  1203 tty1      00:00:00 ps
danila@danilaserver:/$ nice
0
danila@danilaserver:/$ nice -n 10 bash
danila@danilaserver:/$ echo $$
1205
danila@danilaserver:/$ renice -n 5 1205
renice: failed to set priority for 1205 (process ID): Permission denied
danila@danilaserver:/$ ps lax | grep bash
 4 1000    935      644  20   0   8456   5392 do_wai S    tty1      0:00 -bash
 0 1000    1205     935  30  10   8272   5200 do_wai SN  tty1      0:00 bash
 0 1000    1213    1205  30  10   6300    736 -      RN+  tty1      0:00 grep --color=auto bash
danila@danilaserver:/$ sudo -i
[sudo] password for danila:
root@danilaserver:~# echo $$
1215
root@danilaserver:~# renice -n 5 1215
1215 (process ID) old priority 10, new priority 5
root@danilaserver:~#
```

Рисунок 23 – Процессы.

Вывод

В ходе данной лабораторной работы были приобретены навыки использования виртуальной машины, а также освоены некоторые моменты работы в терминале ОС Linux, в особенности порядок работы с текстовым редактором Vi, создание сценариев процессов, работа с командами и сигналами для управления процессами: запуском, остановкой, переводом на передний план, удалением процесса и др.

1. Контрольные вопросы

1. Перечислите состояния задачи в ОС Ubuntu

- running (выполнение) – после выделения ей процессора.
- sleeping (спячка) – при блокировке экрана
- stopped (остановлена) – выполнение задачи прекращено, но из системы не удалена.
- dead (смерть) – может быть удалена из системы
- active (активный) – используются при планировании выполнения процесса.
- expired (неактивный) – используются при планировании выполнения процесса

2. Как создаются задачи в ОС Ubuntu?

Функция clone позволяет создавать задачи.

3. Назовите классы потоков в ОС Ubuntu

- Потоки реального времени, обслуживаемые по алгоритму FIFO.
- Потоки реального времени, обслуживаемые в порядке циклической очереди.
- Потоки разделения времени.

4. Как используется приоритет планирования при запуске задачи.

У каждого потока есть приоритет планирования. Значение по умолчанию равно 20, но оно может быть изменено при помощи системного вызова `nice(value)`, вычитающего значение `value` из 20. Поскольку `value` должно находиться в диапазоне от -20 до +19, приоритеты всегда попадают в промежуток от 1 до 40.

5. Как можно изменить приоритет планирования для выполняющейся задачи?

Используя команду `nice`.