

Липецкий государственный технический университет

Отчет по Лабораторной работе № 2 по дисциплине «Операционная система Linux» на тему «Процессы в операционной системе Linux »

Студент

Руководитель

доцент, к.п.н.

учёная степень, учёное звание

подпись, дата

подпись, дата

Елфимова Д.А.

фамилия, инициалы

Кургасов В.В.

фамилия, инициалы

Липецк 2019 г.

Задание

Порядок выполнения работы:

— Часть I.

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 II.

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 III. Вариант 4.

1. Сгенерировать следующую информацию — полный литинг в длинном формате о процессах текущего пользователя: PID, PPID, выделенное время ЦП, время запуска, размер образа.
2. С помощью сигнала SIGTSTP (используя комбинацию клавиш и команду kill) приостановить выполнение процесса, владельцем которого является текущий пользователь. Через несколько секунд возобновить выполнение процесса.
3. Определить идентификатор и имя процесса, созданного последним пользователем 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.

Оглавление

| | |
|-----------------------|----|
| 1. Часть I | 4 |
| 2. Часть II | 9 |
| 3. Часть III. | 13 |
| Заключение | 16 |

1. Часть I

1. Загрузиться не root, а пользователем. Найти файл с образом ядра. Выяснить по имени файла номер версии Linux.

```
test@ubuntu:~$ cd /boot
test@ubuntu:/boot$ ls -li
итого 32504
269948 -rw-r--r-- 1 root root 1236173 июля 13 2016 abi-4.4.0-31-generic
269946 -rw-r--r-- 1 root root 193009 июля 13 2016 config-4.4.0-31-generic
283729 drwxr-xr-x 5 root root 4096 нояб. 11 2018 grub
262643 -rw-r--r-- 1 root root 21545102 дек. 19 2018 initrd.img-4.4.0-31-generic
279863 -rw-r--r-- 1 root root 176500 марта 12 2014 memtest86+.bin
279864 -rw-r--r-- 1 root root 178176 марта 12 2014 memtest86+.elf
279862 -rw-r--r-- 1 root root 178680 марта 12 2014 memtest86+_multiboot.bin
269947 -rw----- 1 root root 3097433 июля 13 2016 System.map-4.4.0-31-generic
269949 -rw----- 1 root root 6654464 июля 13 2016 vmlinuz-4.4.0-31-generic
```

Рисунок 1. Номер версии Linux

2. Посмотреть процессы ps -f. Прокомментировать. Для этого почистить man ps.

```
test@ubuntu:/boot$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test     1339   1067  0  09:18 tty1        00:00:00 -bash
test     1355   1339  0  09:27 tty1        00:00:00 ps -f
```

Рисунок 2. Процессы ps -f

-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 ?.

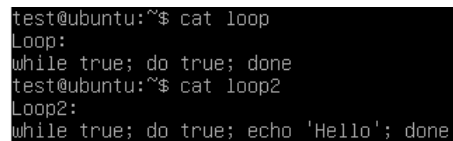
TIME Истраченное процессом время ЦП.

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

3. 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



```
test@ubuntu:~$ cat loop
Loop:
while true; do true; done
test@ubuntu:~$ cat loop2
Loop2:
while true; do true; echo 'Hello'; done
```

Рисунок 3. Сценарии

4. Запустить loop2 на переднем плане: sh loop2.

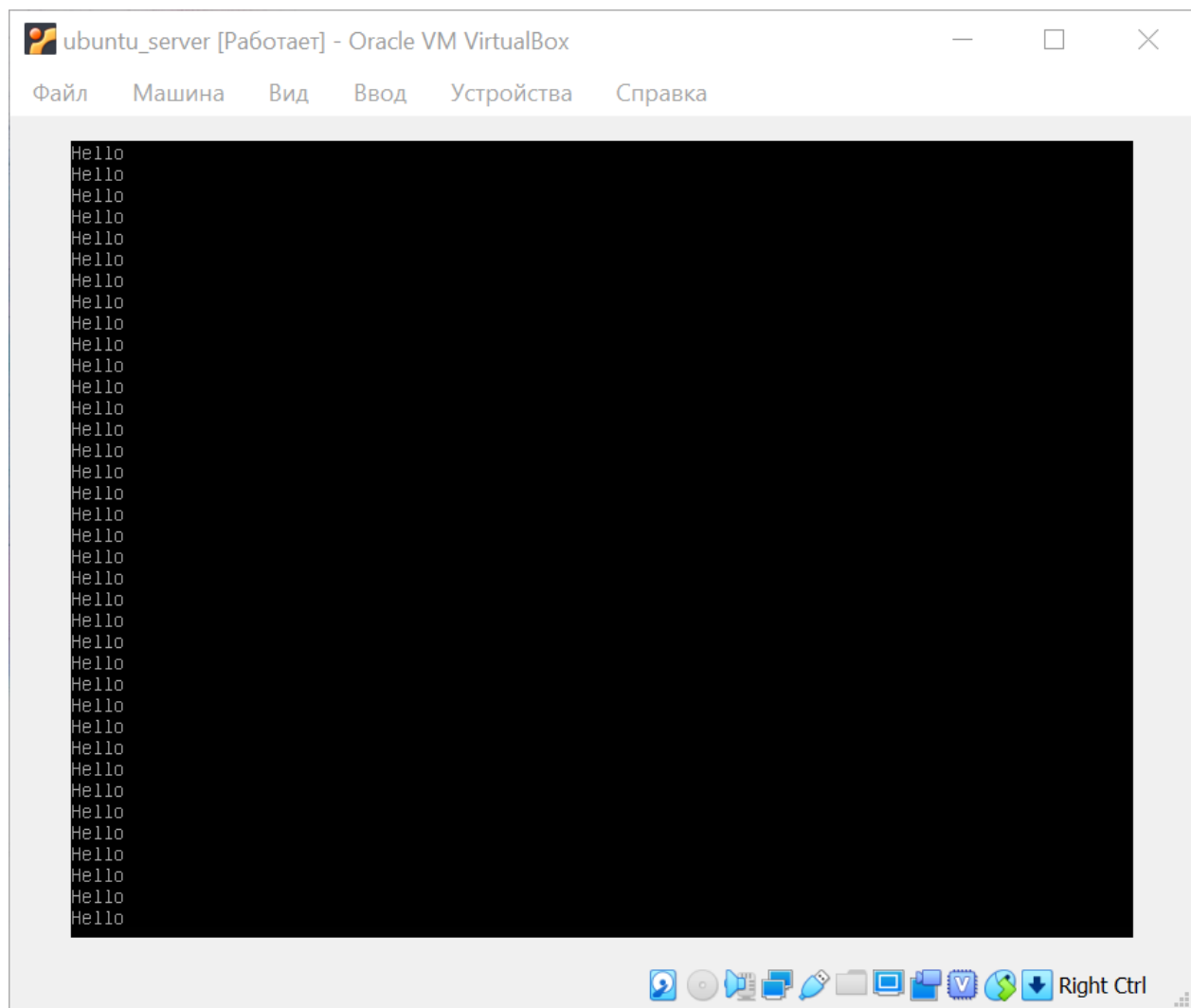


Рисунок 4. sh loop2

5. Остановить, послав сигнал STOP.

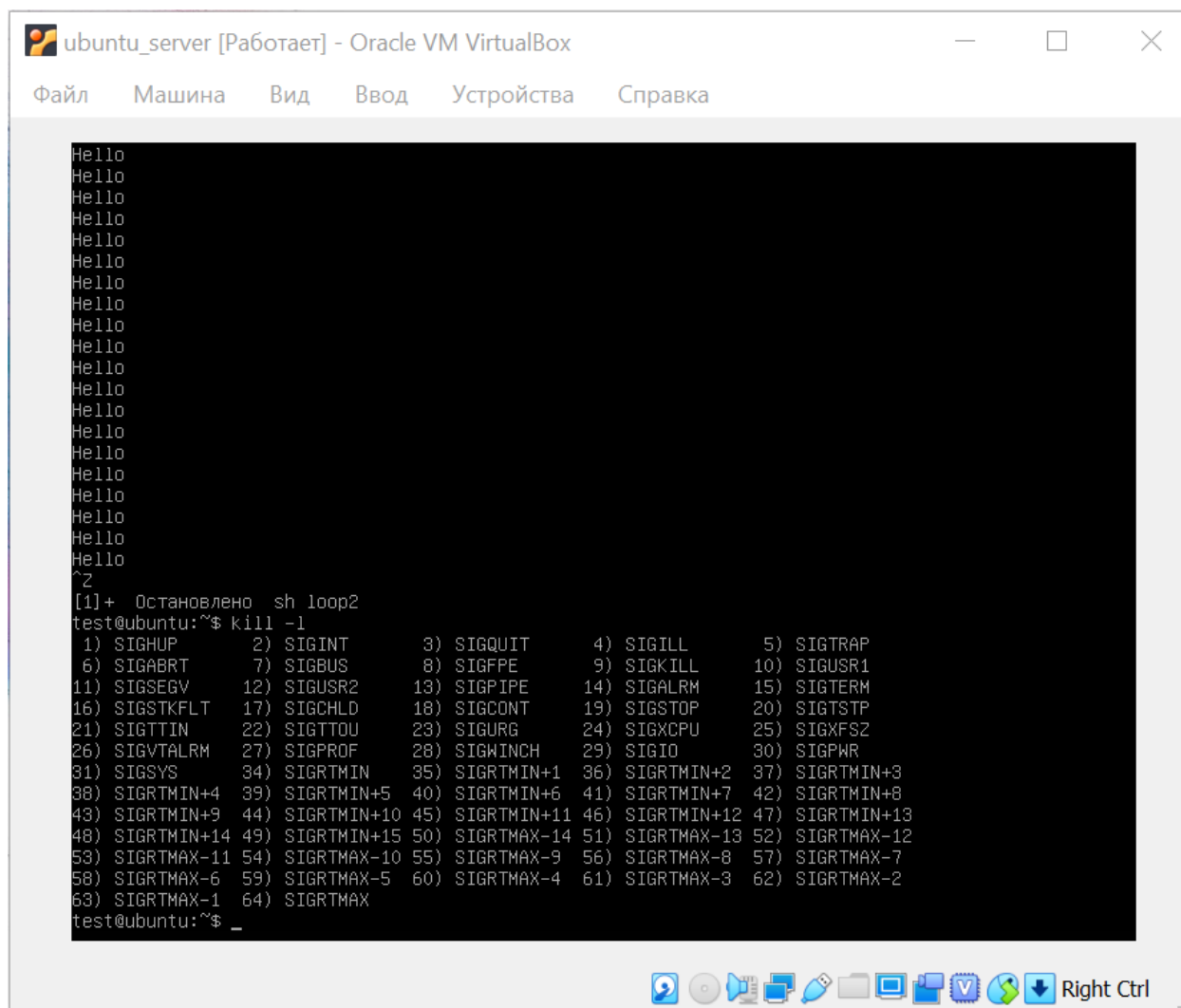


Рисунок 5. STOP

6. Посмотреть последовательно несколько раз `ps -f`. Записать сообщение, объяснить.

```
test@ubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
test        1336     1061  0 09:59 tty1          00:00:00 -bash
test        1355     1336 43 10:04 tty1          00:03:38 sh loop2
test        1362     1336  0 10:12 tty1          00:00:00 ps -f
test@ubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
test        1336     1061  0 09:59 tty1          00:00:00 -bash
test        1355     1336 42 10:04 tty1          00:03:38 sh loop2
test        1363     1336  0 10:12 tty1          00:00:00 ps -f
test@ubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
test        1336     1061  0 09:59 tty1          00:00:00 -bash
test        1355     1336 42 10:04 tty1          00:03:38 sh loop2
test        1364     1336  0 10:12 tty1          00:00:00 ps -f
```

Рисунок 6. `ps -f`

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

7. Убить процесс loop2, послав сигнал kill -9 PID. Записать сообщение. Прокомментировать.

```
test@ubuntu:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test    1336   1061  0  09:59 tty1        00:00:00 -bash
test    1355   1336  20  10:04 tty1        00:03:38 sh loop2
test    1368   1336  10  10:21 tty1        00:00:02 sh loop2
test    1369   1336  0  10:21 tty1        00:00:00 ps -f
test@ubuntu:~$ kill -9 1368
[2]+  Убито                  sh loop2
test@ubuntu:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test    1336   1061  0  09:59 tty1        00:00:00 -bash
test    1355   1336  19  10:04 tty1        00:03:38 sh loop2
test    1370   1336  0  10:22 tty1        00:00:00 ps -f
```

Рисунок 7. kill -9 PID

Как и ожидалось, процесс исчез из списка.

8. Запустить в фоне процесс loop: sh loop&. Не останавливая, посмотреть несколько раз: ps -f. Завершить процесс loop командой kill -15 PID. Записать сообщение, прокомментировать.

```
test@ubuntu:~$ sh loop&
[1] 1347
test@ubuntu:~$ loop: 1: loop: Loop:: not found
ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test    1333   1058  1  10:29 tty1        00:00:00 -bash
test    1347   1333  92  10:29 tty1        00:00:04 sh loop
test    1348   1333  0  10:30 tty1        00:00:00 ps -f
test@ubuntu:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test    1333   1058  1  10:29 tty1        00:00:00 -bash
test    1347   1333  96  10:29 tty1        00:00:08 sh loop
test    1349   1333  0  10:30 tty1        00:00:00 ps -f
test@ubuntu:~$ kill -15 1347
test@ubuntu:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test    1333   1058  0  10:29 tty1        00:00:00 -bash
test    1350   1333  0  10:31 tty1        00:00:00 ps -f
[1]+  Завершено              sh loop
```

Рисунок 8. kill -15 PID

Процесс завершился.

9. Третий раз запустить в фоне. Не останавливая убить командой kill -9 PID.

```
test@ubuntu:~$ sh loop&
[1] 1351
loop: 1: loop: Loop:: not found
test@ubuntu:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test    1333   1058  0  10:29 tty1        00:00:00 -bash
test    1351   1333  99  10:33 tty1        00:00:29 sh loop
test    1352   1333  0  10:33 tty1        00:00:00 ps -f
test@ubuntu:~$ kill -9 1351
test@ubuntu:~$ ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
test    1333   1058  0  10:29 tty1        00:00:00 -bash
test    1353   1333  0  10:33 tty1        00:00:00 ps -f
[1]+  Убито                  sh loop
```

Рисунок 9. kill -9 PID

10. Запустить еще один экземпляр оболочки: `bash`. Запустить несколько процессов в фоне. Останавливать их и снова запускать. Записать результаты просмотра командой `ps -f`.

```
ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
test        1339    1061  0 10:43 tty1        00:00:00 -bash
test        1353    1339  52 10:43 tty1        00:00:40 sh loop
test        1356    1339  47 10:43 tty1        00:00:32 bash loop
test        1359    1339  31 10:44 tty1        00:00:02 sh loop
test        1360    1339   0 10:44 tty1        00:00:00 ps -f
test@ubuntu:~$
test@ubuntu:~$ kill -19 1353
test@ubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
test        1339    1061  0 10:43 tty1        00:00:00 -bash
test        1353    1339  44 10:43 tty1        00:00:54 sh loop
test        1356    1339  42 10:43 tty1        00:00:48 bash loop
test        1359    1339  34 10:44 tty1        00:00:18 sh loop
test        1361    1339   0 10:45 tty1        00:00:00 ps -f
[1]+  Остановлено sh loop
test@ubuntu:~$ kill -19 1359
test@ubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
test        1339    1061  0 10:43 tty1        00:00:00 -bash
test        1353    1339  32 10:43 tty1        00:00:54 sh loop
test        1356    1339  45 10:43 tty1        00:01:13 bash loop
test        1359    1339  39 10:44 tty1        00:00:39 sh loop
test        1362    1339   0 10:46 tty1        00:00:00 ps -f
[3]+  Остановлено sh loop
test@ubuntu:~$ kill -18 1353
test@ubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
test        1339    1061  0 10:43 tty1        00:00:00 -bash
test        1353    1339  14 10:43 tty1        00:00:59 sh loop
test        1356    1339  77 10:43 tty1        00:05:17 bash loop
test        1359    1339  11 10:44 tty1        00:00:39 sh loop
test        1363    1339   0 10:50 tty1        00:00:00 ps -f
```

Рисунок 10. `bash`. Остановка и запуск.

2. Часть II

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

```
sh loop
sh loop
sh loop$
jobs
```

```

test@ubuntu:~$ sh loop
loop: 1: loop: Loop:: not found
^Z
[1]+  Остановлено  sh loop
test@ubuntu:~$ sh loop
loop: 1: loop: Loop:: not found
^Z
[2]+  Остановлено  sh loop
test@ubuntu:~$ sh loop&
[3] 1350
test@ubuntu:~$ loop: 1: loop: Loop:: not found
ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
test         1332   1059  0 21:46 tty1        00:00:00 -bash
test         1346   1332  6 21:47 tty1        00:00:01 sh loop
test         1349   1332 14 21:47 tty1        00:00:02 sh loop
test         1350   1332 94 21:47 tty1        00:00:07 sh loop
test         1351   1332  0 21:47 tty1        00:00:00 ps -f
test@ubuntu:~$ jobs
[1]-  Остановлено  sh loop
[2]+  Остановлено  sh loop
[3]   Выполняется sh loop &
test@ubuntu:~$

```

Рисунок 11. Три задачи

2. Перевести задачу, выполняющуюся в интерактивном режиме, в фоновый режим.

```

bg 2
bg 1
jobs

```

```

test@ubuntu:~$ bg 2
[2]+ sh loop &
test@ubuntu:~$ bg 1
[1]+ sh loop &
test@ubuntu:~$ jobs
[1]   Выполняется sh loop &
[2]-  Выполняется sh loop &
[3]+  Выполняется sh loop &
test@ubuntu:~$ _

```

Рисунок 12. В фоновый режим

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

```

jobs
fg 2
fg 3
jobs

```

```

test@ubuntu:~$ jobs
[1]-  Остановлено  sh loop
[2]+  Остановлено  sh loop
[3]   Выполняется sh loop &
test@ubuntu:~$ fg 2
sh loop
^Z
[2]+  Остановлено  sh loop
test@ubuntu:~$ fg 3
sh loop
^Z
[3]+  Остановлено  sh loop
test@ubuntu:~$ jobs
[1]   Остановлено  sh loop
[2]-  Остановлено  sh loop
[3]+  Остановлено  sh loop

```

Рисунок 13. В интерактивный режим

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.

```
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/test > pipe
```

```

test@ubuntu:~$ mkfifo pipe
test@ubuntu:~$ file pipe
pipe: fifo (named pipe)
test@ubuntu:~$ gzip -c < pipe > out.gz &
[1] 1373
test@ubuntu:~$ ls -R /home > pipe
[1]+  Готово      gzip -c < pipe > out.gz
test@ubuntu:~$ zcat out.gz
/home:
test
user

/home/test:
1
123
123.pub
1.sh
dump.sql
fif2
first
in.txt
loop
loop2
mc
new
out.gz

```

Рисунок 14. Архивирование списка файлов домашнего каталога вместе с подкаталогами (ключ -R)

```

/home/test/
/home/test/123.gz
/home/test/.outNEW.gz.swp
/home/test/out.gz
/home/test/us.zip
/home/test/.out2.zip.swp
/home/test/.bashrc.gz
/home/test/pipe
/home/test/path.txt
/home/test/.bash_logout.gz
/home/test/.profile.gz
/home/test/.selected_editor.gz
/home/test/.viminfo.gz
/home/test/first/
/home/test/.out002.swp
/home/test/.config/
/home/test/.config/mc/
/home/test/.config/mc/ini.gz
/home/test/.config/mc/panels.ini
/home/test/.config/mc/ini
/home/test/.config/mc/panels.ini.gz
/home/test/.config/htop/
/home/test/.config/htop/htoprc.gz
/home/test/.loop.swp.gz
/home/test/.bash_history.gz
/home/test/new/
/home/test/loop2.gz
/home/test/one.zip
/home/test/in.txt.gz
/home/test/mc.gz
/home/test/123.pub.gz
/home/test/1.gz
/home/test/fif2
/home/test/.mano.swp.gz
/home/test/1.sh.gz
/home/test/loop.gz
"outNEW.gz" [только для чтения] [noeol] 1L, 328C

```

Рисунок 15. Архивирование одного каталога вместе с файлами и подкаталогами

```

test@ubuntu:~$ gzip -c < pipe > outNEW.gz &
[1] 1358
test@ubuntu:~$ tar -cvf outNEW.tar /home/test > pipe
tar: Удаляется начальный '/' из имен объектов
tar: /home/test/outNEW.tar: файл является архивом; не сброшен
[1]+  Готово      gzip -c < pipe > outNEW.gz

```

Рисунок 16. Архивирование одного каталога вместе с файлами и подкаталогами - 2

Если же попробовать создать канал архивирования tar и направить в него путь каталога, который необходимо заархивировать, то получится вот такая ошибка:

```

test@ubuntu:~$ tar -cvf out002.tar < pipe2 &
[11] 1415
test@ubuntu:~$ ls /home/test > pipe2
tar: Робкий отказ от создания пустого архива
test@ubuntu:~$ Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.

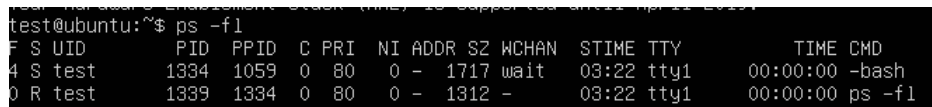
```

Рисунок 17. Ошибка

3. Часть III.

1. Сгенерировать следующую информацию — полный литинг в длинном формате о процессах текущего пользователя: PID, PPID, выделенное время ЦП, время запуска, размер образа.

```
ps -fl
```



```
test@ubuntu:~$ ps -fl
F S UID      PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
4 S test      1334 1059  0  80   0 - 1717 wait  03:22 tty1      00:00:00 -bash
0 R test      1339 1334  0  80   0 - 1312 -    03:22 tty1      00:00:00 ps -fl
```

Рисунок 18. Полный литинг в длинном формате о процессах текущего пользователя

2. С помощью сигнала SIGTSTP (используя комбинацию клавиш и команду kill) приостановить выполнение процесса, владельцем которого является текущий пользователь. Через несколько секунд возобновить выполнение процесса.

```
ps -l
kill -l
kill -20 1336
ps -l
kill -8 1336
ps -l
```

```

test@ubuntu:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000  1331  1058  0 80   0 -  1717 wait  tty1        00:00:00 bash
0 R  1000  1336  1331  78 80   0 -   571 -    tty1        00:00:47 sh
0 R  1000  1340  1331  49 80   0 -   571 -    tty1        00:00:13 sh
0 R  1000  1342  1331  0 80   0 -  1254 -    tty1        00:00:00 ps
test@ubuntu:~$ kill -l
 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
test@ubuntu:~$ kill -20 1336
test@ubuntu:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000  1331  1058  0 80   0 -  1717 wait  tty1        00:00:00 bash
0 T  1000  1336  1331  65 80   0 -   571 signal tty1        00:00:59 sh
0 R  1000  1340  1331  55 80   0 -   571 -    tty1        00:00:32 sh
0 R  1000  1343  1331  0 80   0 -  1254 -    tty1        00:00:00 ps

[1]+  Остановлено sh loop
test@ubuntu:~$ kill -18 1336
test@ubuntu:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000  1331  1058  0 80   0 -  1717 wait  tty1        00:00:00 bash
0 R  1000  1336  1331  46 80   0 -   571 -    tty1        00:01:07 sh
0 R  1000  1340  1331  70 80   0 -   571 -    tty1        00:01:18 sh
0 R  1000  1344  1331  0 80   0 -  1254 -    tty1        00:00:00 ps
test@ubuntu:~$

```

Рисунок 19. SIGTSTP

3. Определить идентификатор и имя процесса, созданного последним пользователем root. Последний процесс это процесс пользователя test с идентификатором 1345 и именем ps -f.

```

test@ubuntu:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000  1331  1058  0 80   0 -  1717 wait  tty1        00:00:00 bash
0 R  1000  1336  1331  78 80   0 -   571 -    tty1        00:00:47 sh
0 R  1000  1340  1331  49 80   0 -   571 -    tty1        00:00:13 sh
0 R  1000  1342  1331  0 80   0 -  1254 -    tty1        00:00:00 ps
test@ubuntu:~$ kill -l
 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
test@ubuntu:~$ kill -20 1336
test@ubuntu:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000  1331  1058  0 80   0 -  1717 wait  tty1        00:00:00 bash
0 T  1000  1336  1331  65 80   0 -   571 signal  tty1        00:00:59 sh
0 R  1000  1340  1331  55 80   0 -   571 -    tty1        00:00:32 sh
0 R  1000  1343  1331  0 80   0 -  1254 -    tty1        00:00:00 ps

[1]+  Остановлено  sh loop
test@ubuntu:~$ kill -18 1336
test@ubuntu:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000  1331  1058  0 80   0 -  1717 wait  tty1        00:00:00 bash
0 R  1000  1336  1331  46 80   0 -   571 -    tty1        00:01:07 sh
0 R  1000  1340  1331  70 80   0 -   571 -    tty1        00:01:18 sh
0 R  1000  1344  1331  0 80   0 -  1254 -    tty1        00:00:00 ps
test@ubuntu:~$

```

Рисунок 20. Идентификатор и имя процесса

Заключение

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.