

# Липецкий государственный технический университет

Кафедра прикладной математики

## Отчет по лабораторной работе №3 «Управление процессами в Linux.»

Студент

\_\_\_\_\_

подпись, дата

Комолых Т.О.  
фамилия, инициалы

Группа

ПМ-18

Руководитель

доц., к.п.н. кафедры АСУ  
ученая степень, ученое звание

\_\_\_\_\_

подпись, дата

Кургасов В. В.  
фамилия, инициалы

Липецк 2020 г.

# Содержание

|   |           |
|---|-----------|
| <b>Цель работы</b>  | <b>3</b>  |
| <b>Задание кафедры</b>  | <b>3</b>  |
| <b>Выполнение работы</b>  | <b>4</b>  |
| Повторение команд: cat, head, tail, more, less, grep, find. . . . . | 4         |
| Команда cat . . . . .   | 4         |
| Команда head . . . . .  | 5         |
| Команда tail . . . . .  | 6         |
| Команда more . . . . .  | 7         |
| Команда less . . . . .  | 9         |
| Команда grep . . . . .  | 11        |
| Команда find . . . . .  | 12        |
| Разбор понятий конвейера, перенаправления ввода-вывода. . . . .     | 14        |
| Перенаправление ввода вывода. . . . .                               | 14        |
| Конвейеры. . . . .  | 16        |
| Назначение прав доступа. Команды chmod, chown. . . . .              | 17        |
| Команда chmod. . . . .  | 17        |
| Команда chown. . . . .  | 18        |
| Процессы. Запуск процессов в supervisor. . . . .                    | 19        |
| Процессы. . . . .   | 19        |
| Запуск процессов в supervisor. . . . .                              | 22        |
| Автоматический запуск по расписанию. . . . .                        | 22        |
| <b>Вывод</b>  | <b>24</b> |
| <b>Список литературы</b>  | <b>25</b> |

## Цель работы

Ознакомиться на практике с управлением процессов в ОС Ubuntu.

## Задание кафедры

- 1) Повторить команды `cat`, `head`, `tail`, `more`, `less`, `grep`, `find`.
- 2) Разобраться с понятиями конвейер, перенаправление ввода-вывода.
- 3) Ознакомиться с информацией из рекомендованных источников про конвейеризации.
- 4) Повторить назначение прав доступа. Команды `chmod`, `chown`.
- 5) Ознакомиться с информацией по теме процессы, посмотреть и опробовать примеры наиболее распространенных команд, изучить возможность запуска процессов в `supervisor`.
- 6) Изучить возможность автоматического запуска программ по расписанию.

# Выполнение работы

## Повторение команд: cat, head, tail, more, less, grep, find.

### Команда cat

cat filename – команда, используемая для вывода текстовой информации из файла на экран. Для демонстрации работы данной команды создадим текстовый файл text (рисунок 1), заполнив его в текстовом редакторе vi (рисунок 2).

```
tatyana@arianrod:~$ vi text.txt_
```

Рисунок 1.

```
Длина тела – 55-95 см, хвоста – 26-55 см, масса – 6-10 кг.  
Лапы короткие, с плавательными перепонками.  
Хвост мускулистый, не пушистый.  
Половой диморфизм слабо выражен: около 50% самцов крупнее самок, масса и длина тела (без хвоста) самцов составляет соответственно 5,5-11,4 кг и 60-90 см, а у самок – 3,36-7,6 кг и 59-70 см.
```

Рисунок 2.

С помощью команды cat производится отображение на экране содержимого файла text.txt (рисунок 3).

```
tatyana@arianrod:~$ cat text.txt  
Выдра – крупный зверь с вытянутым гибким телом обтекаемой формы.  
Длина тела – 55-95 см, хвоста – 26-55 см, масса – 6-10 кг.  
Лапы короткие, с плавательными перепонками.  
Хвост мускулистый, не пушистый.  
Половой диморфизм слабо выражен: около 50% самцов крупнее самок, масса и длина тела (без хвоста) самцов составляет соответственно 5,5-11,4 кг и 60-90 см, а у самок – 3,36-7,6 кг и 59-70 см.
```

Рисунок 3.

Используя команду nl filename, можно пронумеровать строки файла (рисунок 4).

```
tatyana@arianrod:~$ nl text.txt  
1  Выдра – крупный зверь с вытянутым гибким телом обтекаемой формы.  
2  Длина тела – 55-95 см, хвоста – 26-55 см, масса – 6-10 кг.  
3  Лапы короткие, с плавательными перепонками.  
4  Хвост мускулистый, не пушистый.  
5  Половой диморфизм слабо выражен: около 50% самцов крупнее самок, масса и длина тела (без хвоста) самцов составляет соответственно 5,5-11,4 кг и 60-90 см, а у самок – 3,36-7,6 кг и 59-70 см.
```

Рисунок 4.

Параметр -b команды cat нумерует все непустые строки (рисунок 5).

```
tatyana@arianrod:~$ cat -b text.txt  
1  Выдра – крупный зверь с вытянутым гибким телом обтекаемой формы.  
2  Длина тела – 55-95 см, хвоста – 26-55 см, масса – 6-10 кг.  
3  Лапы короткие, с плавательными перепонками.  
4  Хвост мускулистый, не пушистый.  
5  Половой диморфизм слабо выражен: около 50% самцов крупнее самок, масса и длина тела (без хвоста) самцов составляет соответственно 5,5-11,4 кг и 60-90 см, а у самок – 3,36-7,6 кг и 59-70 см.
```

Рисунок 5.

Параметр -E команды cat показывает символы конца строки, как \$ (рисунок 6).

```
tatyana@arianrod:~$ cat -E text.txt
Выдра - крупный зверь с вытянутым гибким телом обтекаемой формы.$
Длина тела - 55-95 см, хвоста - 26-55 см, масса - 6-10 кг.$
Лапы короткие, с плавательными перепонками.$
Хвост мускулистый, не пушистый. $
Половой диморфизм слабо выражен: около 50% самцов крупнее самок, масса и длина тела (без хвоста) самцов составляет соответственно 5,5-11,4 кг и 60-90 см, а у самок - 3,36-7,6 кг и 59-70 см. $
```

Рисунок 6.

Также команда cat имеет ещё несколько полезных параметров:

- -n - нумеровать все строки;
- -s - отобразить несколько подряд идущих пустых строк в виде одной пустой строки;
- -T - отображать табуляции в виде `^|`;
- -h - отобразить справку;
- -v - версия утилиты.

### Команда head

Команда head filename – по умолчанию выводит 10 первых строк файла. Для демонстрации работы данной команды был видоизменён изначальный файл text.txt (рисунок 7).

```
Длина тела - 55-95 см,
хвоста - 26-55 см,
масса - 6-10 кг.
Лапы короткие,
с плавательными перепонками.
Хвост мускулистый,
не пушистый.
Половой диморфизм слабо выражен: около 50% самцов крупнее самок,
масса и длина тела (без хвоста) самцов составляет соответственно
5,5-11,4 кг и 60-90 см,
а у самок -
3,36-7,6 кг и 59-70 см.
```

Рисунок 7.

Параметр -n команды head позволяет вывести определённое lines число строк (рисунок 8). В данном примере выводятся первые четыре строки.

```
tatyana@arianrod:~$ head -n 4 text.txt
Выдра - крупный зверь с вытянутым гибким телом обтекаемой формы.
Длина тела - 55-95 см,
хвоста - 26-55 см,
масса - 6-10 кг.
```

Рисунок 8.

Параметр -c команды head позволяет вывести указанное число bytes байт с начала файла (рисунок 9). В данном примере выводится первый 21 байт файла.

```
tatyana@arianrod:~$ head -c 21 text.txt
Выдра - крупtatyana@arianrod:~$
```

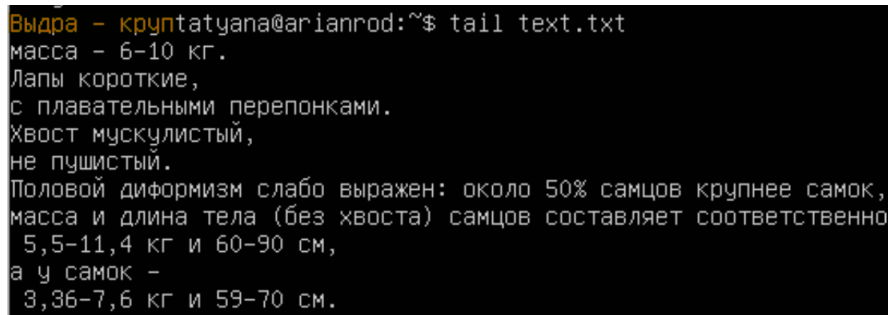
Рисунок 9.

Также чаще всего к команде head применяются такие опции:

- -q — выводит только текст, не добавляя к нему название файла.
- -v — перед текстом выводит название файла.
- -z — символы перехода на новую строку заменяет символами завершения строк.

### Команда tail

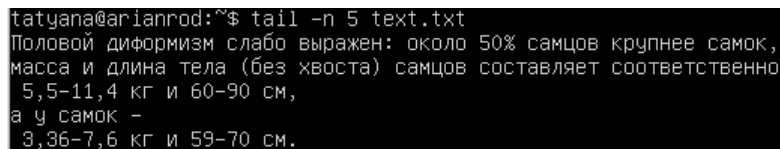
Команда tail filename – по умолчанию выводит 10 последних строк файла (рисунок 10).



```
Выдра - круптatyana@arianrod:~$ tail text.txt
масса - 6-10 кг.
Лапы короткие,
с плавательными перепонками.
Хвост мускулистый,
не пушистый.
Половой диморфизм слабо выражен: около 50% самцов крупнее самок,
масса и длина тела (без хвоста) самцов составляет соответственно
5,5-11,4 кг и 60-90 см,
а у самок -
3,36-7,6 кг и 59-70 см.
```

Рисунок 10.

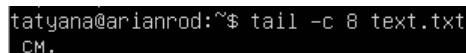
Параметр -n команды tail позволяет вывести определённое lines число последних строк (рисунок 11). В данном примере выводятся последние 5 строк.



```
tatyana@arianrod:~$ tail -n 5 text.txt
Половой диморфизм слабо выражен: около 50% самцов крупнее самок,
масса и длина тела (без хвоста) самцов составляет соответственно
5,5-11,4 кг и 60-90 см,
а у самок -
3,36-7,6 кг и 59-70 см.
```

Рисунок 11.

Параметр -с команды tail позволяет вывести указанное число bytes байт с конца файла (рисунок 12). В данном примере выводятся последние 8 байт файла.



```
tatyana@arianrod:~$ tail -с 8 text.txt
см.
```

Рисунок 12.

Также команда tail имеет параметр -f, который позволяет войти в постоянный цикл по считыванию конца файла. Таким образом при поступлении в файл новой информации пользователь может вести мониторинг её в реальном времени. Выход из этого режима осуществляется комбинацией клавиш <Ctrl>+<C> .

Также команда tail имеет следующие полезные опции:

- -f - обновлять информацию по мере появления новых строк в файле;
- -pid - используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс;
- -q - не выводить имена файлов;
- -retry - повторять попытки открыть файл, если он недоступен;
- -v - выводить подробную информацию о файле;

## Команда more

Команда `more filename` – эта команда позволяет просматривать длинные файлы по частям (рисунок 13). Для демонстрации работы команды `more` был видоизменён файл `text.txt`.

```
tatyana@arianrod:~$ more text.txt
```

Рисунок 13.

Команда `more` использует для прокрутки две клавиши – пробел (показать следующий экран) и Enter (показать следующую строку). Но у `more` есть один недостаток – она способна прокручивать текст только вперед.

Результат работы программы продемонстрирован на рисунках 14, 15.

```
Выдра –  
крупный  
зверь  
с  
вытянутым  
гибким  
телом  
обтекаемой  
формы.  
Длина  
тела  
– 55-95  
см,  
хвоста  
– 26-55  
см,  
масса  
– 6-10  
кг.  
Лапы  
короткие,  
с  
плавательными  
перепонками.  
Хвост  
мускулистый,  
не  
пушистый.  
Половой  
диформизм  
слабо  
выражен:  
около  
50%  
самцов  
крупнее  
--More-- (66%)
```

Рисунок 14.

```

плавательными
перепонками.
Хвост
мускулистый,
не
пушистый.
Половой
диформизм
слабо
выражен:
около
50%
самцов
крупнее
самок,
масса
и
длина
тела
(без хвоста)
самцов
составляет
соответственно
5,5-11,4
кг
и
60-90
см,
а
у
самок -
3,36-7,6
кг
и
59-70
см.

```

Рисунок 15.

Параметр +n команды more отображает текст, начиная со строки с номером n (рисунок 16).

```
tatyana@arianrod:~$ more +10 text.txt_
```

Рисунок 16.

Результат команды представлен на рисунке 17.

```

Длина
тела
- 55-95
см,
хвоста
- 26-55
см,
масса
- 6-10
кг.
Лапы
короткие,
с
плавательными
перепонками.
Хвост
мускулистый,
не
пушистый.
Половой
диформизм
слабо
выражен:
около
50%
самцов
крупнее
самок,
масса
и
длина
тела
(без хвоста)
самцов
составляет
соответственно
5,5-11,4
кг
и
60-90
см,
а
у
самок -
3,36-7,6
кг
и
59-70
см.
--More-- (86%)

```

Рисунок 17.

Параметр -help команды more осуществляет вызов справки (рисунок 18).



```
tatyana@arianrod:~$ more --help
Usage:
more [options] <file>...

A file perusal filter for CRT viewing.

Options:
-d      display help instead of ringing bell
-f      count logical rather than screen lines
-l      suppress pause after form feed
-c      do not scroll, display text and clean line ends
-p      do not scroll, clean screen and display text
-s      squeeze multiple blank lines into one
-u      suppress underlining
-n<number> the number of lines per screenful
+<number> display file beginning from line number
+<string> display file beginning from search string match

--help    display this help
-V, --version display version

For more details see more(1).
```

Рисунок 18.

Также команда `more` имеет следующие полезные опции:

- `-d` — вывод информации в конце страницы о клавишах, использующихся для продолжения работы, завершения её или получения инструкций;
- `-l` — игнорирование в тексте символа разрыва страницы;
- `-f` — подсчёт числа логических строк вместо экранных;
- `-p` — очистка экрана терминала для того, чтобы пользователю не пришлось пользоваться прокруткой перед выводом следующей порции текста;
- `-c` — устранение потребности в прокрутке (как и `-p`) — отображение текста, начиная с верха экрана, и стирание при этом предыдущего вывода построчно;
- `-s` — замена нескольких пустых строк, расположенных подряд, одной пустой строкой;
- `-u` — удаление подчёркивания;
- `-n` — отображение n-го количества строк;
- `+<строка>` — поиск в файле указанной строки и начало вывода текста именно с неё;
- `-v` — вывод на экран текущей версии утилиты.

### Команда `less`

Команда `less filename` — позволяет просматривать файлы любой длины, прокручивая их в любую сторону.

```
tatyana@arianrod:~$ less text.txt
```

Рисунок 19.

Результат работы программы продемонстрирован на рисунках 20, 21.

```

формы.
Длина
тела
- 55-95
см,
хвоста
26-55
см,
масса
- 6-10
кг.
Лапы
короткие,
с
плавательными
перепонками.
Хвост
мускулистый,
не
пушистый.
Половой
диформизм
слабо
выражен:
около
50%
самцов
крупнее
самок,
масса
и
длина
тела
(без хвоста)
самцов
составляет
:-

```

Рисунок 20.

```

крупный
зверь
с
вытянутым
гибким
телом
обтекаемой
формы.
Длина
тела
- 55-95
см,
хвоста
26-55
см,
масса
- 6-10
кг.
Лапы
короткие,
с
плавательными
перепонками.
Хвост
мускулистый,
не
пушистый.
Половой
диформизм
слабо
выражен:
около
50%
самцов
крупнее
самок,
:-

```

Рисунок 21.

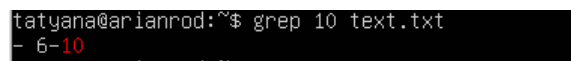
Команда less имеет следующие полезные опции:

- -a — не осуществлять поиск в тексте, который в данный момент отображен на экране;
- -bn — задать размер буфера памяти;
- -c — листать текст, полностью стирая содержимое экрана (построчная прокрутка работать не будет);
- -Dxcolor — задать цвет отображаемого текста;

- -E — выйти, когда утилита достигнет конца файла;
- -e — выйти, когда утилита второй раз достигнет конца файла;
- -F — выйти, если содержимое файла помещается на одном экране;
- -f — открыть специальный файл;
- -hn — задать максимальное количество строк для прокрутки назад;
- -up — задать максимальное количество строк для прокрутки вперёд;
- -i — игнорировать регистр;
- -I — игнорировать регистр, даже если паттерн для поиска содержит заглавные буквы;
- -jn — указать, в какой строке должна быть выведена искомая информация;
- -J — пометить строки, соответствующие результатам поиска;
- -n — не выводить номера строк;
- -N — вывести номера строк;
- -s — заменить множество идущих подряд пустых строк одной пустой строкой;
- -w — выделить первую строку нового фрагмента текста.

### Команда `grep`

Команда `grep regex filename` – утилита фильтрации текста. Ищет в файле `filename` строки, в которых встречается выражение, соответствующее шаблону `regex` и выводит их на стандартный вывод. На рисунке 22 осуществляется поиск и вывод строки, содержащей заданное значение 10.



```
tatyana@arianrod:~$ grep 10 text.txt
6-10
```

Рисунок 22.

Утилиту `grep` используют не только для фильтрации текстовых файлов, но и, например, для фильтрации вывода каких-либо команд.

На рисунке 23 осуществляется поиск всех файлов, начинающихся на букву `f` в каталоге `bin`.

```
tatyana@arianrod:~$ ls /bin | grep ^f
factor
faillog
fallocate
false
fgconsole
fgrep
file
finalrd
fincore
find
findmnt
flock
fmt
fold
free
from
ftp
fuser
fusermount
fwupdagent
fwupdate
fwupdmgr
fwupdtool
fwupdtpmvlog
```

Рисунок 23.

Команда `grep` имеет следующие полезные опции:

- `-b` - показывать номер блока перед строкой;
- `-c` - подсчитать количество вхождений шаблона;
- `-h` - не выводить имя файла в результатах поиска внутри файлов Linux;
- `-i` - не учитывать регистр;
- `-l` - отобразить только имена файлов, в которых найден шаблон;
- `-n` - показывать номер строки в файле;
- `-s` - не показывать сообщения об ошибках;
- `-v` - инвертировать поиск, выдавать все строки кроме тех, что содержат шаблон;
- `-w` - искать шаблон как слово, окружённое пробелами;
- `-e` - использовать регулярные выражения при поиске;
- `-An` - показать вхождение и `n` строк до него;
- `-Bn` - показать вхождение и `n` строк после него;
- `-Cn` - показать `n` строк до и после вхождения;

### Команда `find`

Команда `find` используется для осуществления поиска нужного файла. Она может искать файлы по имени, размеру, дате создания или модификации и некоторым другим критериям. Общий синтаксис команды `find` имеет следующий вид:

```
find [список_каталогов] критерий_поиска
```

Параметр "список\_каталогов" определяет, где искать нужный файл. Началом "критерия\_поиска" определяющего, что именно должна искать программа `find`, считается первый аргумент, начинающийся на: `-`, `(`, `)` или `!`

На рисунке 24 осуществляется поиск в корневой папке файла `text.txt`.

```
tatyana@arianrod:~$ find ./ -name text.txt
./text.txt
```

Рисунок 24.

На рисунке 25 осуществляется поиск файла по имени с третьим и четвёртым произвольными символами.

```
tatyana@arianrod:~$ find -name te??txt
./text.txt
```

Рисунок 25.

На рисунке 26 осуществляется поиск файла по указанному типу, где f - обычный файл.

```
tatyana@arianrod:~$ find -type f
./config/htop/htoprc
./lessht
./bash_history
./cache/motd.legal-displayed
./text.txt.swp
./bash_logout
./bashrc
./text.txt
./sudo_as_admin_successful
./profile
./viminfo
```

Рисунок 26.

На рисунке 27 осуществляется поиск файлов по имени с указанием расширения, где имя представляет собой произвольную строку.

```
tatyana@arianrod:~$ find -name *.txt
./text.txt
```

Рисунок 27.

Основные параметры:

- -P никогда не открывать символические ссылки
- -L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.
- -maxdepth - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.
- -depth - искать сначала в текущем каталоге, а потом в подкаталогах
- -mount искать файлы только в этой файловой системе.
- -version - показать версию утилиты find
- -print - выводить полные имена файлов
- -type f - искать только файлы
- -type d - поиск папки в Linux

Основные критерии:

- -name - поиск файлов по имени

- -perm - поиск файлов в Linux по режиму доступа
- -user - поиск файлов по владельцу
- -group - поиск по группе
- -mtime - поиск по времени модификации файла
- -atime - поиск файлов по дате последнего чтения
- -nogroup - поиск файлов, не принадлежащих ни одной группе
- -nouser - поиск файлов без владельцев
- -newer - найти файлы новее чем указанный
- -size - поиск файлов в Linux по их размеру

## Разбор понятий конвейера, перенаправления ввода-вывода.

### Перенаправление ввода вывода.

Перенаправление ввода-вывода — возможность командной оболочки ряда операционных систем перенаправлять стандартные потоки в определённое пользователем место, например, в файл.

Перенаправление обычно осуществляется вставкой специального символа > между командами.

На рисунке 28 осуществляется запись результата команды ps -f в файл loop.

```
tatyana@arianrod:~$ ps -f > loop
```

Рисунок 28.

Результат записи в файл приведён на рисунке 29.

```
UID          PID     PPID  C  STIME TTY          TIME CMD
tatyana      910      650  0  17:00 tty1          00:00:00 -bash
tatyana     1370      910  0  17:12 tty1          00:00:00 ps -f

"loop" 3L, 172C                                1,1      All
```

Рисунок 29.

Данное действие можно усовершенствовать, указав номер (файловый дескриптор) непосредственно перед символом перенаправления. Этот номер указывает, какой именно поток используется для перенаправления. В Unix существуют следующие стандартные потоки ввода-вывода:

| Дескриптор | Название            | Описание                 |
|------------|---------------------|--------------------------|
| 0          | <code>stdin</code>  | Стандартный ввод         |
| 1          | <code>stdout</code> | Стандартный вывод        |
| 2          | <code>stderr</code> | Стандартный вывод ошибок |

Рисунок 30.

Например, команда `1 2> файл1` выполняет команду `1`, направляя стандартный поток ошибок в файл `1`. По правилам синтаксиса для указания потока, в который осуществляется перенаправление, нужно добавлять символ `2` после символа перенаправления.

На рисунке 31 осуществляется попытка найти все файлы с именем `.profile`, направляя на вход другой программы объединённый результат посредством конвейера.

```
tatyana@arianrod:~$ find / -name .profile 2>&1 | less
```

Рисунок 31.

Результаты поиска приведены на рисунках 32, 33.

```
find: '/etc/polkit-1/localauthority': Permission denied
find: '/etc/multipath': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/snap/core18/1932/etc/ssl/private': Permission denied
find: '/snap/core18/1932/root': Permission denied
find: '/snap/core18/1932/var/cache/ldconfig': Permission denied
find: '/snap/core18/1932/var/lib/private': Permission denied
find: '/snap/core18/1885/etc/ssl/private': Permission denied
find: '/snap/core18/1885/root': Permission denied
find: '/snap/core18/1885/var/cache/ldconfig': Permission denied
find: '/snap/core18/1885/var/lib/private': Permission denied
find: '/snap/core18/1885/var/lib/snapd/void': Permission denied
find: '/snap/core/10185/etc/ssl/private': Permission denied
find: '/snap/core/10185/root': Permission denied
find: '/snap/core/10185/var/cache/ldconfig': Permission denied
find: '/snap/core/10185/var/lib/machines': Permission denied
find: '/snap/core/10185/var/lib/private': Permission denied
find: '/snap/core/10185/var/lib/snapd/void': Permission denied
find: '/snap/core/10185/var/lib/waagent': Permission denied
find: '/snap/core/10185/var/spool/cron/crontabs': Permission denied
find: '/snap/core/10185/var/spool/rsyslog': Permission denied
find: '/snap/core/10126/etc/ssl/private': Permission denied
find: '/snap/core/10126/root': Permission denied
find: '/snap/core/10126/var/cache/ldconfig': Permission denied
find: '/snap/core/10126/var/lib/machines': Permission denied
find: '/snap/core/10126/var/lib/private': Permission denied
find: '/snap/core/10126/var/lib/snapd/void': Permission denied
find: '/snap/core/10126/var/lib/waagent': Permission denied
find: '/snap/core/10126/var/spool/cron/crontabs': Permission denied
find: '/snap/core/10126/var/spool/rsyslog': Permission denied
find: '/run/user/1000/inaccessible': Permission denied
find: '/run/sudo': Permission denied
find: '/run/cryptsetup': Permission denied
find: '/run/systemd/unit-root': Permission denied
find: '/run/systemd/inaccessible': Permission denied
find: '/run/lvm': Permission denied
```

Рисунок 32.

```
find: '/proc/904/ns': Permission denied
find: '/proc/1421/task/1421/fd': Permission denied
find: '/proc/1421/task/1421/fdinfo': Permission denied
find: '/proc/1421/task/1421/ns': Permission denied
find: '/proc/1421/fd': Permission denied
find: '/proc/1421/map_files': Permission denied
find: '/proc/1421/fdinfo': Permission denied
find: '/proc/1421/ns': Permission denied
find: '/proc/1883/task/1883/fd': Permission denied
find: '/proc/1883/task/1883/fdinfo': Permission denied
find: '/proc/1883/task/1883/ns': Permission denied
find: '/proc/1883/fd': Permission denied
find: '/proc/1883/map_files': Permission denied
find: '/proc/1883/fdinfo': Permission denied
find: '/proc/1883/ns': Permission denied
find: '/proc/1898/task/1898/fd': Permission denied
find: '/proc/1898/task/1898/fdinfo': Permission denied
find: '/proc/1898/task/1898/ns': Permission denied
find: '/proc/1898/fd': Permission denied
find: '/proc/1898/map_files': Permission denied
find: '/proc/1898/fdinfo': Permission denied
find: '/proc/1898/ns': Permission denied
find: '/proc/1901/task/1901/fd': Permission denied
find: '/proc/1901/task/1901/fdinfo': Permission denied
find: '/proc/1901/task/1901/ns': Permission denied
find: '/proc/1901/fd': Permission denied
find: '/proc/1901/map_files': Permission denied
find: '/proc/1901/fdinfo': Permission denied
find: '/proc/1901/ns': Permission denied
find: '/proc/1902/task/1902/fd': Permission denied
find: '/proc/1902/task/1902/fdinfo': Permission denied
find: '/proc/1902/task/1902/ns': Permission denied
find: '/proc/1902/fd': Permission denied
find: '/proc/1902/map_files': Permission denied
find: '/proc/1902/fdinfo': Permission denied
find: '/proc/1902/ns': Permission denied
(END)
```

Рисунок 33.

### Конвейеры.

Конвейеры — это возможность нескольких программ работать совместно, когда выход одной программы непосредственно идет на вход другой без использования промежуточных временных файлов. Синтаксис: команда1 | команда2, выполняет команду1 используя её поток вывода как поток ввода при выполнении команды2, что равносильно использованию двух перенаправлений и временного файла:

команда1 > ВременныйФайл

команда2 < ВременныйФайл

rm ВременныйФайл

На рисунке 34 команда перенаправления и конвейеризации объединены в цепочку для получения более сложных команд.

```
tatyana@arianrod:~$ ls | grep '.txt' | sort > loop
```

Рисунок 34.

На рисунке 35 выводится список содержимого текущего каталога, который фильтруется, оставляя только строки содержащие '.txt', затем этот отфильтрованный список лексически сортируется и окончательный результат помещается в файл loop.

```
loop.txt
loop1.txt
text.txt
top-output.txt
```

Рисунок 35.



## Назначение прав доступа. Команды chmod, chown.

### Команда chmod.

Команда chmod – изменяет права доступа к файлу. Для использования этой команды также необходимо иметь права владельца файла или права root . Синтаксис команды имеет следующий вид:

chmod mode filename , где filename – имя файла, у которого изменяются права доступа;  
mode – права доступа, устанавливаемые на файл.

Права доступа можно записать в 2 вариантах – символьном и абсолютном.

В символьном виде использование команды chmod будет выглядеть следующим образом:

chmod группа\_пользователей действие вид\_прав filename.

где: u,g,o,a – установка прав для пользователя, группы, остальных пользователей, всех групп прав доступа соответственно.

+, -, = – добавить, удалить, установить разрешение соответственно.

r,w,x,X,u,g,o – право чтения, записи, выполнения, выполнения если есть такое право еще у какой либо из групп доступа, такие же как у владельца, такие же как у группы, такие же как у остальных пользователей.

На рисунке 36 осуществляется просмотр разрешений, установленных на файл text.txt. Результат работы команды показывает, что все группы пользователей имеют право на чтение, а право записи доступно только суперпользователю и группе пользователей.

```
root@arianrod:/home/tatyana# ls -l text.txt
-rw-rw-r-- 1 tatyana tatyana 683 Nov 11 17:12 text.txt
```

Рисунок 36.

На рисунке 37 группе пользователей предоставляется право на выполнение.

```
root@arianrod:/home/tatyana# chmod g+x text.txt
```

Рисунок 37.

Для просмотра разрешений, установленных на этот файл используется команда ls с ключом -l (рисунок 38).

```
root@arianrod:/home/tatyana# ls -l text.txt
-rw-rwxr-- 1 tatyana tatyana 683 Nov 11 17:12 text.txt
```

Рисунок 38.

На рисунке 39 суперпользователь наделяется такими же правами, что и группа пользователей.

```
root@arianrod:/home/tatyana# chmod u=g text.txt
root@arianrod:/home/tatyana# ls -l text.txt
-rwxrwxr-- 1 tatyana tatyana 683 Nov 11 17:12 text.txt
```

Рисунок 39.

На рисунке 40 все группы пользователей лишаются права выполнять указанный файл.

```
root@arianrod:/home/tatyana# chmod a-x text.txt
root@arianrod:/home/tatyana# ls -l text.txt
-rw-rw-r-- 1 tatyana tatyana 683 Nov 11 17:12 text.txt
```

Рисунок 40.

Для использования абсолютного режима необходимо представить права доступа к файлу в виде 3-х восьмеричных групп. Соответственно, первая группа задаёт права для пользователя, вторая - для группы пользователей, а третья - для остальных пользователей.

- 0 - никаких прав;
- 1 - только выполнение;
- 2 - только запись;
- 3 - выполнение и запись;
- 4 - только чтение;
- 5 - чтение и выполнение;
- 6 - чтение и запись;
- 7 - чтение запись и выполнение;

На рисунке 41 пользователь получает права на чтение, запись и выполнение, группа пользователей - права на чтение и выполнение, а остальные пользователи - право чтения файла.

```
root@arianrod:/home/tatyana# chmod 754 text.txt
root@arianrod:/home/tatyana# ls -l text.txt
-rwxr-xr-- 1 tatyana tatyana 683 Nov 11 17:12 text.txt
```

Рисунок 41.

### Команда chown.

Команда chown – позволяет сменить владельца файла. Для использования этой команды необходимо либо иметь права владельца текущего файла или права root.

chown username:groupname filename

где username – имя пользователя – нового владельца файла;

groupname – имя группы – нового владельца файла;

filename – имя файла, у которого сменяется владелец.

На рисунке 42 осуществляется смена владельца файла с tatyana на user.

```
root@arianrod:/home/tatyana# ls -l text.txt
-rwxr-xr-- 1 tatyana tatyana 683 Nov 11 17:12 text.txt
root@arianrod:/home/tatyana# chown user text.txt
root@arianrod:/home/tatyana# ls -l text.txt
-rwxr-xr-- 1 user tatyana 683 Nov 11 17:12 text.txt
```

Рисунок 42.

Также команда chown имеет следующие основные опции:

- -c - подробный вывод всех выполняемых изменений;
- -f - минимум информации, скрыть сообщения об ошибках;
- -h - изменять права символических ссылок и не трогать файлы, к которым они ведут;
- -R - рекурсивная обработка всех подкаталогов;
- -H - если передана символическая ссылка на директорию - перейти по ней;
- -L - переходить по всем символическим ссылкам на директории;
- -P - не переходить по символическим ссылкам на директории (по умолчанию).

## Процессы. Запуск процессов в supervisor.

### Процессы.

Процесс – понятие совокупности программного кода и данных, загруженных в память ЭВМ. Процесс – это не запущенная программа (приложение) или команда, так как приложение может создавать несколько процессов одновременно. Код процесса не обязательно должен выполняться в текущий момент времени, так как процесс может находиться в состоянии спящего. В этом случае выполнение кода такого процесса приостановлено. Существует всего 3 состояния, в которых может находиться процесс:

Работающий процесс – в данный момент код этого процесса выполняется.

Спящий процесс – в данный момент код процесса не выполняется в ожидании какого либо события (нажатия клавиши на клавиатуре, поступление данных из сети и т.д.)

Процесс-зомби – сам процесс уже не существует, его код и данные выгружены из оперативной памяти, но запись в таблице процессов остается по тем или иным причинам.

Каждому процессу в системе назначаются числовые идентификаторы (личные номера) и идентификаторы родительского процесса. PID является именем процесса, по которому можно адресовать процесс в операционной системе. PPID определяет родственные отношения между процессами.

Для работы с процессами удобно использовать top (рисунок 43).

```
root@arianrod:/home/tatyana# top_
```

Рисунок 43.

Результат ввода команды приведён на рисунке 44.

```
top - 18:32:36 up 2:29, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 96 total, 1 running, 95 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 981.3 total, 422.9 free, 137.2 used, 421.3 buff/cache
MiB Swap: 1207.0 total, 1207.0 free, 0.0 used, 692.1 avail Mem
```

| PID | USER | PR  | NI  | VIRT   | RES   | SHR  | S | %CPU | %MEM | TIME+   | COMMAND              |
|-----|------|-----|-----|--------|-------|------|---|------|------|---------|----------------------|
| 1   | root | 20  | 0   | 167684 | 11804 | 8572 | S | 0.0  | 1.2  | 0:01.08 | systemd              |
| 2   | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | kthreadd             |
| 3   | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | rcu_gp               |
| 4   | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | rcu_par_gp           |
| 6   | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | kworker/0:0H-kblockd |
| 9   | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | mm_percpu_wq         |
| 10  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.08 | ksoftirqd/0          |
| 11  | root | 20  | 0   | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:01.02 | rcu_sched            |
| 12  | root | rt  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.02 | migration/0          |
| 13  | root | -51 | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | idle_inject/0        |
| 14  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | cpuhp/0              |
| 15  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | kdevtmpfs            |
| 16  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | netns                |
| 17  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | rcu_tasks_kthre      |
| 18  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | kauditd              |
| 19  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | khungtaskd           |
| 20  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | oom_reaper           |
| 21  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | writeback            |
| 22  | root | 20  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | kcompactd0           |
| 23  | root | 25  | 5   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | ksmd                 |
| 24  | root | 39  | 19  | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | khugepaged           |
| 70  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | kintegrityd          |
| 71  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | kblockd              |
| 72  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | blkcg_punt_bio       |
| 73  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | tpm_dev_wq           |
| 74  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | ata_sff              |
| 75  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | md                   |
| 76  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | edac-poller          |
| 77  | root | 0   | -20 | 0      | 0     | 0    | I | 0.0  | 0.0  | 0:00.00 | devfreq_wq           |
| 78  | root | rt  | 0   | 0      | 0     | 0    | S | 0.0  | 0.0  | 0:00.00 | watchdogd            |

Рисунок 44.

Для просмотра списка процессов в Linux существует команда ps (рисунок 45).

```
root@arianrod:/home/tatyana# ps
  PID TTY          TIME CMD
   653 tty1      00:00:00 login
   938 tty1      00:00:00 sudo
   939 tty1      00:00:00 su
   940 tty1      00:00:00 bash
  1277 tty1      00:00:00 ps
```

Рисунок 45.

Чтобы отобразить конкретные детали процесса пользователя, используется команда top с параметром -u.

```
root@arianrod:/home/tatyana# top -u tatyana_
```

Рисунок 46.

Результат команды представлен на рисунке 47.

```
top - 18:37:07 up 2:33, 1 user, load average: 0.01, 0.00, 0.00
Tasks: 95 total, 1 running, 94 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 981.3 total, 422.9 free, 137.2 used, 421.3 buff/cache
MiB Swap: 1207.0 total, 1207.0 free, 0.0 used, 692.1 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
  922 tatyana    20   0 18648 9956 8264 S   0.0   1.0   0:00.03 systemd
  924 tatyana    20   0 168900 3612   16 S   0.0   0.4   0:00.00 (sd-pam)
  929 tatyana    20   0  7072 5088 3384 S   0.0   0.5   0:00.01 bash
```

Рисунок 47.

Опция z переключает между цветным и одноцветным режимами (рисунок 48).

```
top - 18:38:35 up 2:35, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 97 total, 1 running, 95 sleeping, 1 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 981.3 total, 422.1 free, 137.9 used, 421.3 buff/cache
MiB Swap: 1207.0 total, 1207.0 free, 0.0 used, 691.4 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 1395 root        20   0  7916 3732 3200 R   0.3   0.4   0:00.01 top
    1 root        20   0 167684 11804 8572 S   0.0   1.2   0:01.08 systemd
    2 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
    9 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
   10 root        20   0      0      0      0 S   0.0   0.0   0:00.09 ksoftirqd/0
   11 root        20   0      0      0      0 I   0.0   0.0   0:01.05 rcu_sched
   12 root        rt    0      0      0      0 S   0.0   0.0   0:00.03 migration/0
   13 root       -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   14 root        20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   16 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
   17 root        20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_kthre
   18 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kauditd
   19 root        20   0      0      0      0 S   0.0   0.0   0:00.00 khungtaskd
   20 root        20   0      0      0      0 S   0.0   0.0   0:00.00 oom_reaper
   21 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 writeback
   22 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kcompactd0
   23 root        25   5      0      0      0 S   0.0   0.0   0:00.00 ksmd
   24 root        39  19      0      0      0 S   0.0   0.0   0:00.00 khugepaged
   70 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 kintegrityd
   71 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 kblockd
   72 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 blkcg_punt_bio
   73 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
   74 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 ata_sff
   75 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 md
   76 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 edac-poller
   77 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 devfreq_wq
```

Рисунок 48.

Опция h выводит справки по утилите (рисунок 49).

```

Help for Interactive Commands - procps-ng UNKNOWN
Window 1:Def: Cumulative mode Off. System: Delay 3.0 secs; Secure mode Off.

Z,B,E,e  Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale
l,t,m    Toggle Summary: 'l' load avg; 't' task/cpu stats; 'm' memory info
0,1,2,3,I Toggle: '0' zeros; '1/2/3' cpus or numa node views; 'I' Irix mode
f,F,X    Fields: 'f'/'F' add/remove/order/sort; 'X' increase fixed-width

L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right
R,H,J,C . Toggle: 'R' Sort; 'H' Threads; 'J' Num justify; 'C' Coordinates
c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify
x,y      . Toggle highlights: 'x' sort field; 'y' running tasks
z,b      . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
u,U,o,O . Filter by: 'u'/'U' effective/any user; 'o'/'O' other criteria
n,#,^O . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'O' other filter(s)
V,v      . Toggle: 'V' forest view; 'v' hide/show forest view children

k,r      Manipulate tasks: 'k' kill; 'r' renice
d or s   Set update interval
W,Y      Write configuration file 'W'; Inspect other output 'Y'
q        Quit
          ( commands shown with '.' require a visible task display window )
Press 'h' or '?' for help with Windows,
Type 'q' or <Esc> to continue █

```

Рисунок 49.

Также полезны следующие интерактивные команды, которые можно выполнять во время работы программы:

- q или Esc - выход из top;
- A - выбор цветовой схемы;
- d или s - изменить интервал обновления информации;
- H - выводить потоки процессов;
- k - послать сигнал завершения процессу;
- W - записать текущие настройки программы в конфигурационный файл;
- Y - посмотреть дополнительные сведения о процессе, открытые файлы, порты, логи и т.д.;
- Z - изменить цветовую схему;
- l - скрыть или вывести информацию о средней нагрузке на систему;
- m - выключить или переключить режим отображения информации о памяти;
- x - выделять жирным колонку, по которой выполняется сортировка;
- y - выделять жирным процессы, которые выполняются в данный момент;
- c - переключение режима вывода команды, доступен полный путь и только команда;
- F - настройка полей с информацией о процессах;
- o - фильтрация процессов по произвольному условию;
- u - фильтрация процессов по имени пользователя;
- V - отображение процессов в виде дерева;
- i - переключение режима отображения процессов, которые сейчас не используют ресурсы процессора;

- n - максимальное количество процессов, для отображения в программе;
- L - поиск по слову;
- <> - перемещение поля сортировки вправо и влево;

### Запуск процессов в supervisor.

Чтобы установить supervisord необходимо выполнить команду (рисунок 50).

```
root@arianrod:/home/tatyana# apt-get install supervisor
```

Рисунок 50.

Для запуска сценария test.sh создаётся конфигурационный файл (рисунок 51).

```
tatyana@arianrod:~$ sudo vi /etc/supervisor/conf.d/testloop1.conf
```

Рисунок 51.

Конфигурационный файл приведён на рисунке 52.

```
[program:testloop1]
command=/home/tatyana/loop1
autostart=true
autorestart=true
stdout_logfile=/var/log/loop1_out.log
stderr_logfile=/var/log/loop1_err.log
stopsignal=KILL_
```

Рисунок 52.

Создав конфигурационный файл, нужно оповестить Supervisor о появлении новой программы. Для этого используется команда supervisorctl. Сперва нужно просмотреть наличие новых файлов, а затем активировать их (рисунок 53).

```
tatyana@arianrod:~$ sudo supervisorctl reread
testloop1: available
tatyana@arianrod:~$ sudo supervisorctl update
testloop1: added process group
```

Рисунок 53.

### Автоматический запуск по расписанию.

cron – программа-демон, предназначенная для выполнения заданий в определенное время, или через определенные промежутки времени. Для редактирования заданий используется утилита crontab. Для редактирования расписания текущего файла необходимо выполнить команду crontab -e (рисунок 54).

```

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 23 * * 1 /home/tatyana/loop1

```

crontab: installing new crontab

Рисунок 54.

Синтаксис настройки одной задачи cron в crontab:

минута час день месяц день\_недели /путь\_к\_исполняемому\_файлу

С помощью команды `crontab -l` осуществляется проверка наличия поставленного на расписание процесса (рисунок 55).

```

tatyana@arianrod:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 23 * * 1 /home/tatyana/loop1

```

Рисунок 55.

## Вывод

В ходе лабораторной работы были усвоены знания по управлению процессами в ОС Ubuntu.



## Список литературы

- [1] Львовский, С.М. Набор и верстка в системе  $\text{\LaTeX}$  [Текст] / С.М. Львовский. М.: МЦНМО, 2006. — 448 с.
- [2] SEDICOMM. 30 полезных команд «ps» для мониторинга процессов Linux: <https://blog.sedicom.com/2018/05/28/30-poleznyh-komand-ps-dlya-monitoringa-protsessov-linux/> (дата обращения: 29.10.2020). - Текст: электронный.