

Липецкий государственный технический университет

Факультет автоматизации и информатики
Кафедра автоматизированных систем управления

Отчет по лабораторной работе № 3 по дисциплине «OS Linux» на тему «Управление процессами в Linux»

Студент

Группа АС-18-1

Руководитель

К.Н.

учёная степень, учёное звание

подпись, дата

подпись, дата

Сухоруков К.О.

фамилия, инициалы

Кургасов В.В.

фамилия, инициалы

Липецк 2020 г.

СОДЕРЖАНИЕ

Цель работы	2
1 Ход выполнения	3
1.1 Задание 1	3
1.2 Задание 2	13
1.3 Задание 3	15
1.4 Задание 4	16
1.5 Задание 5	28
Вывод	29

Цель работы

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

1 Ход выполнения

1.1 Задание 1

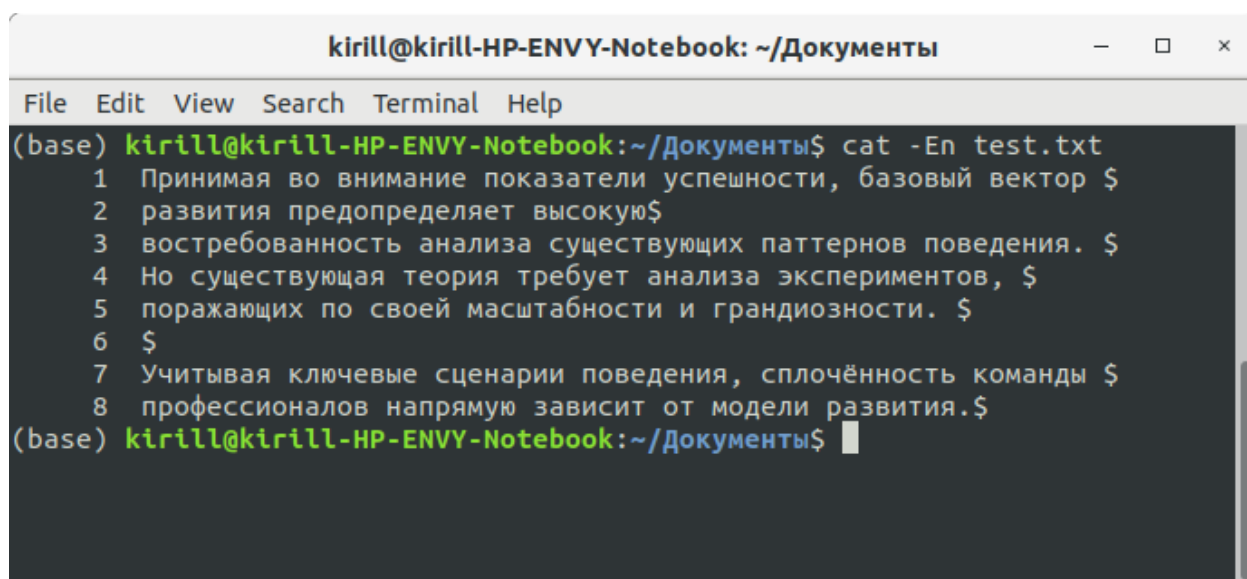
Повторить команды `cat`, `head`, `tail`, `more`, `less`, `grep`, `find`.

По сути, задача команды `cat` очень проста - она читает данные из файла или стандартного ввода и выводит их на экран. Это все, чем занимается утилита. Но с помощью ее опций и операторов перенаправления вывода можно сделать очень многое.

Рассмотрим основные опции команды `cat`:

1. `-b` – нумеровать только непустые строки;
2. `-E` – показывать символ `$` в конце каждой строки;
3. `-n` – нумеровать все строки;
4. `-s` – удалять пустые повторяющиеся строки;
5. `-T` – отображать табуляции в виде `^I`;
6. `-h` – отобразить справку;
7. `-v` – версия утилиты.

Для примера откроем какой-нибудь файл, пронумеруем все строки и покажем символ `$` в конце каждой строки. Результат выполнения представлен на рисунке 1.1.



```
kirill@kirill-HP-ENVY-Notebook: ~/Документы
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~/Документы$ cat -En test.txt
1 Принимая во внимание показатели успешности, базовый вектор $
2 развития предопределяет высокую$
3 востребованность анализа существующих паттернов поведения. $
4 Но существующая теория требует анализа экспериментов, $
5 поражающих по своей масштабности и грандиозности. $
6 $
7 Учитывая ключевые сценарии поведения, сплочённость команды $
8 профессионалов напрямую зависит от модели развития.$
(base) kirill@kirill-HP-ENVY-Notebook:~/Документы$
```

Рисунок 1.1 – Пример работы с командой `cat`

Команда `head` выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита.

Чаще всего к команде `head` применяются такие опции:

1. `-c` (`–bytes`) – позволяет задавать количество текста не в строках, а в байтах. При записи в виде `–bytes=[-]NUM` выводит на экран все содержимое файла, кроме `NUM` байт, расположенных в конце документа.

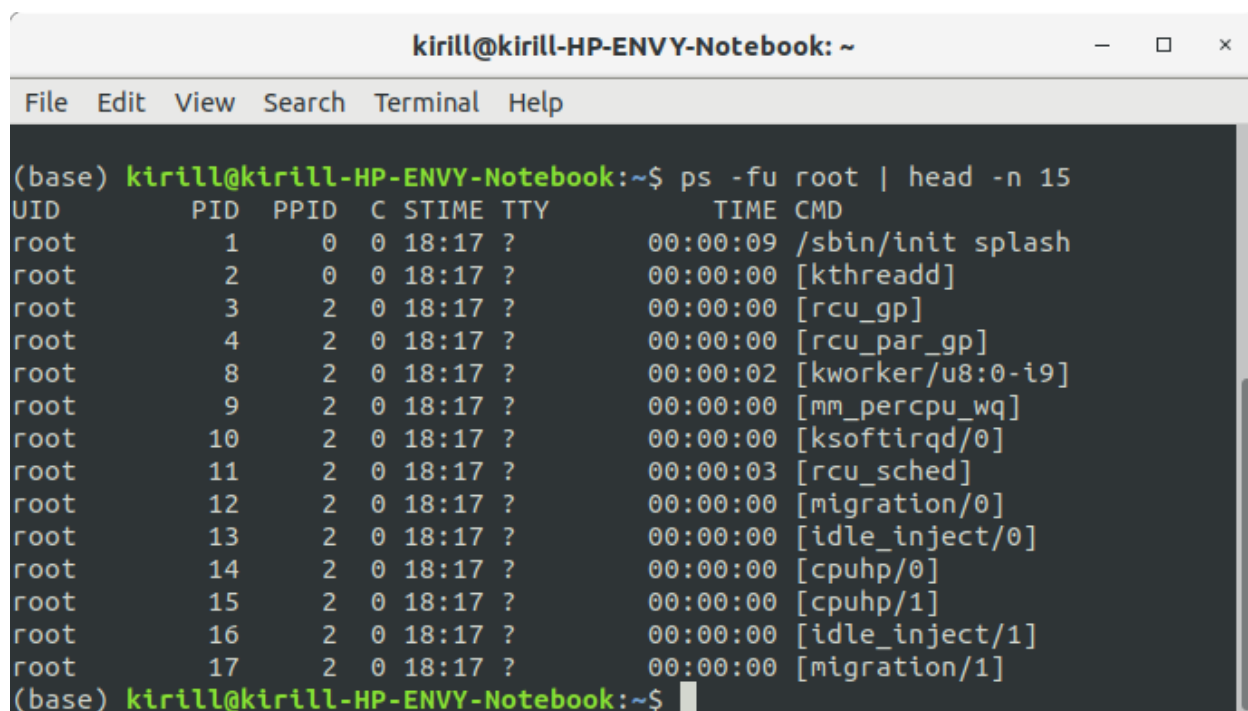
2. `-n` (`–lines`) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию. Если записать эту опцию в виде `–lines=[-]NUM`, будет показан весь текст кроме последних `NUM` строк.

3. `-q` (`–quiet`, `–silent`) – выводит только текст, не добавляя к нему название файла.

4. `-v` (`–verbose`) – перед текстом выводит название файла.

5. `-z` (`–zero-terminated`) – символы перехода на новую строку заменяет символами завершения строк.

В качестве примера выведем первые 15 строк выдаваемые командой `ps -fu root`. Результат выполнения представлен на рисунке 1.2.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -fu root | head -n 15  
UID      PID  PPID  C  STIME TTY          TIME CMD  
root         1     0  0  18:17 ?        00:00:09 /sbin/init splash  
root         2     0  0  18:17 ?        00:00:00 [kthreadd]  
root         3     2  0  18:17 ?        00:00:00 [rcu_gp]  
root         4     2  0  18:17 ?        00:00:00 [rcu_par_gp]  
root         8     2  0  18:17 ?        00:00:02 [kworker/u8:0-i9]  
root         9     2  0  18:17 ?        00:00:00 [mm_percpu_wq]  
root        10     2  0  18:17 ?        00:00:00 [ksoftirqd/0]  
root        11     2  0  18:17 ?        00:00:03 [rcu_sched]  
root        12     2  0  18:17 ?        00:00:00 [migration/0]  
root        13     2  0  18:17 ?        00:00:00 [idle_inject/0]  
root        14     2  0  18:17 ?        00:00:00 [cpuhp/0]  
root        15     2  0  18:17 ?        00:00:00 [cpuhp/1]  
root        16     2  0  18:17 ?        00:00:00 [idle_inject/1]  
root        17     2  0  18:17 ?        00:00:00 [migration/1]  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

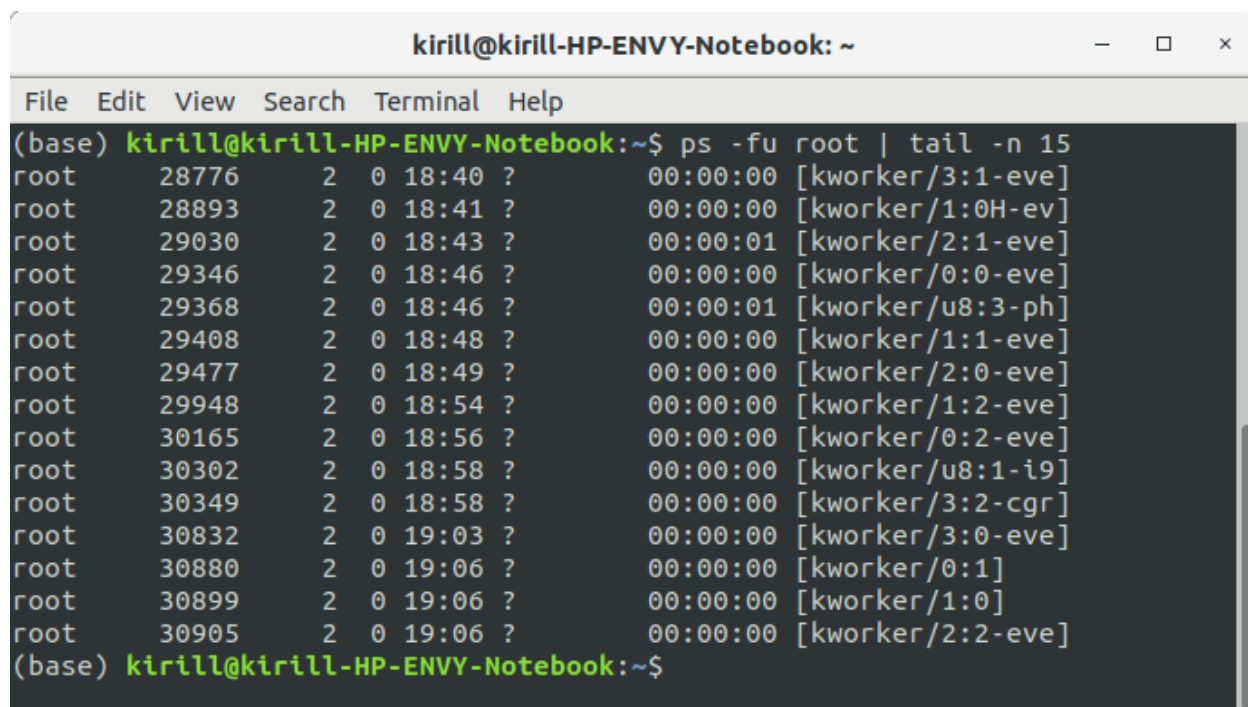
Рисунок 1.2 – Пример работы с командой `head`

Если необходимо просмотреть не все содержимое файла, а только то, что находится в конце, то можно воспользоваться командой `tail`. Она позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме.

По умолчанию утилита выводит десять последних строк из файла, но ее поведение можно настроить с помощью опций:

1. `-c` – выводить указанное количество байт с конца файла;
2. `-f` – обновлять информацию по мере появления новых строк в файле;
3. `-n` – выводить указанное количество строк из конца файла;
4. `-pid` – используется с опцией `-f`, позволяет завершить работу утилиты, когда завершится указанный процесс;
5. `-q` – не выводить имена файлов;
6. `-retry` – повторять попытки открыть файл, если он недоступен;
7. `-v` – выводить подробную информацию о файле.

Используем команду `tail`, чтобы просмотреть последние 15 строк выдаваемые командой `ps -fu root`. Результат выполнения представлен на рисунке 1.3.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -fu root | tail -n 15  
root      28776      2  0 18:40 ?        00:00:00 [kworker/3:1-eve]  
root      28893      2  0 18:41 ?        00:00:00 [kworker/1:0H-ev]  
root      29030      2  0 18:43 ?        00:00:01 [kworker/2:1-eve]  
root      29346      2  0 18:46 ?        00:00:00 [kworker/0:0-eve]  
root      29368      2  0 18:46 ?        00:00:01 [kworker/u8:3-ph]  
root      29408      2  0 18:48 ?        00:00:00 [kworker/1:1-eve]  
root      29477      2  0 18:49 ?        00:00:00 [kworker/2:0-eve]  
root      29948      2  0 18:54 ?        00:00:00 [kworker/1:2-eve]  
root      30165      2  0 18:56 ?        00:00:00 [kworker/0:2-eve]  
root      30302      2  0 18:58 ?        00:00:00 [kworker/u8:1-i9]  
root      30349      2  0 18:58 ?        00:00:00 [kworker/3:2-cgr]  
root      30832      2  0 19:03 ?        00:00:00 [kworker/3:0-eve]  
root      30880      2  0 19:06 ?        00:00:00 [kworker/0:1]  
root      30899      2  0 19:06 ?        00:00:00 [kworker/1:0]  
root      30905      2  0 19:06 ?        00:00:00 [kworker/2:2-eve]  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.3 – Пример работы с командой `tail`

Утилита `more` предназначена для постраничного просмотра файлов в терминале Linux.

Список опций команды:

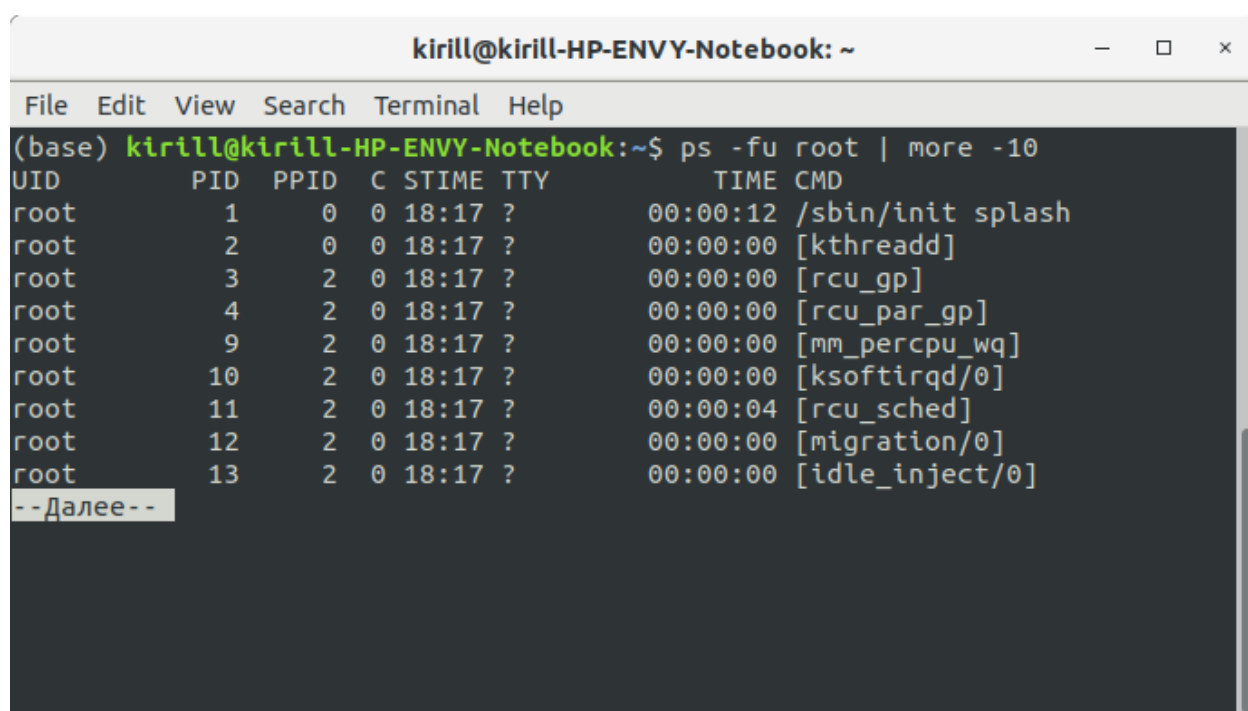
1. `-d` – вывод информации в конце страницы о клавишах, использующихся для продолжения работы, завершения её или получения инструкций;
2. `-l` – игнорирование в тексте символа разрыва страницы;
3. `-f` – подсчёт числа логических строк вместо экранных;
4. `-r` – очистка экрана терминала для того, чтобы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

Также у команды `more` есть собственные горячие клавиши и интерактивные команды:

1. `h (?)` – помощь (вывод информации только об интерактивных командах);
2. ПРОБЕЛ – отображение следующей порции текста (по умолчанию количество строк зависит от текущего размера окна терминала);
3. `z` – то же, что и ПРОБЕЛ; `ENTER` – вывод текста построчно (шаг команды – одна строка);
4. `d` – прокрутка текста на количество строк, соответствующее размеру терминала;
5. `q (Q)` – выход из утилиты;

6. s – переход на одну строку вперёд;
7. f – переход на одну экранную страницу вперёд;
8. b – переход на одну экранную страницу назад;
9. ' – возвращение к месту начала поиска;
10. = – отображение текущего количества строк;
11. /pattern – поиск с использованием регулярных выражений;
12. n – поиск слов и фраз, соответствующих последнему использованному регулярному выражению;
13. !command (:command) – выполнение команды в субоболочке;
14. v – открытие файла в текстовом редакторе, назначенном по умолчанию, а если таковой не найден, использование консольного текстового редактора для открытия файла;
15. :n – переход к следующему файлу;
16. :p – переход к предыдущему файлу;
17. :f – вывод названия текущего файла и количества строк в нём;
18. . – повторное выполнение предыдущей команды.

Выведем результат выполнения команды `ps -fu root` постранично по 10 строк. Результат выполнения представлен на рисунке 1.4.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ ps -fu root | more -10
UID      PID  PPID  C  STIME TTY          TIME CMD
root         1     0  0  18:17 ?        00:00:12 /sbin/init splash
root         2     0  0  18:17 ?        00:00:00 [kthreadd]
root         3     2  0  18:17 ?        00:00:00 [rcu_gp]
root         4     2  0  18:17 ?        00:00:00 [rcu_par_gp]
root         9     2  0  18:17 ?        00:00:00 [mm_percpu_wq]
root        10     2  0  18:17 ?        00:00:00 [ksoftirqd/0]
root        11     2  0  18:17 ?        00:00:04 [rcu_sched]
root        12     2  0  18:17 ?        00:00:00 [migration/0]
root        13     2  0  18:17 ?        00:00:00 [idle_inject/0]
--Далее--
```

Рисунок 1.4 – Пример работы с командой `more`

Когда команда вывела часть текста и бездействует в ожидании дальнейших действий пользователя можно выполнять интерактивные команды. Они нужны для управления выводом.

Команда `less` позволяет перематывать текст не только вперёд, но и назад, осуществлять поиск в обоих направлениях, переходить сразу в конец или в начало файла. Особенность `less` заключается в том, что команда не считывает текст полностью, а загружает его небольшими фрагментами.

Наиболее популярные опции:

1. `-a`, `-search-skip-screen` – не осуществлять поиск в тексте, который в данный момент отображен на экране;
2. `-bn`, `-buffers=n` – задать размер буфера памяти;
3. `-c`, `-clear-screen` – листать текст, полностью стирая содержимое экрана (построчная прокрутка работать не будет);
4. `-Dxcolor`, `-color=xcolor` – задать цвет отображаемого текста;
5. `-E`, `-QUIT-AT-EOF` – выйти, когда утилита достигнет конца файла;
6. `-e`, `-quit-at-eof` – выйти, когда утилита второй раз достигнет конца файла;
7. `-F`, `-quit-if-one-screen` – выйти, если содержимое файла помещается на одном экране;
8. `-f`, `-force` – открыть специальный файл;
9. `-hn`, `-max-back-scroll=n` – задать максимальное количество строк для прокрутки назад;
10. `-yn`, `-max-forw-scroll=n` – задать максимальное количество строк для прокрутки вперёд;
11. `-i`, `-ignore-case` – игнорировать регистр;
12. `-I`, `-IGNORE-CASE` – игнорировать регистр, даже если паттерн для поиска содержит заглавные буквы;
13. `-jn`, `-jump-target=n` – указать, в какой строке должна быть выведена искомая информация;
14. `-J`, `-status-column` – пометить строки, соответствующие результатам поиска;

15. -n, -line-numbers – не выводить номера строк;
16. -N, -LINE-NUMBERS – вывести номера строк;
17. -s, -squeeze-blank-lines – заменить множество идущих подряд пустых строк одной пустой строкой;
18. -w, -hilite-unread – выделить первую строку нового фрагмента текста.

Во время просмотра текста утилитой можно управлять при помощи внутренних команд, набирая их на клавиатуре компьютера. Наиболее часто используемые из них:

1. h, H – справка;
2. Space, Ctrl+V, f, Ctrl+F – прокрутить текст на один экран вперёд;
3. Enter, Return, Ctrl+N, e, Ctrl+E, j, Ctrl+J – прокрутить текст на n строк вперед, по умолчанию n=1;
4. y, Ctrl+Y, Ctrl+P, k, Ctrl+K – прокрутить текст на n строк назад, по умолчанию n=1;
5. Ctrl+→ – прокрутить текст по горизонтали в конец строки;
6. Ctrl+← – прокрутить текст по горизонтали в начало строки;
7. :d – удалить текущий файл из списка файлов;
8. Ctrl+G, :f – вывести основную информацию о файле;
9. q, Q, :q, :Q, ZZ – выход.

Выведем с помощью команды less содержимое длинного текстового файла, убрав множественные пустые строки (используем опцию -s). Результат выполнения представлен на рисунке 1.5.

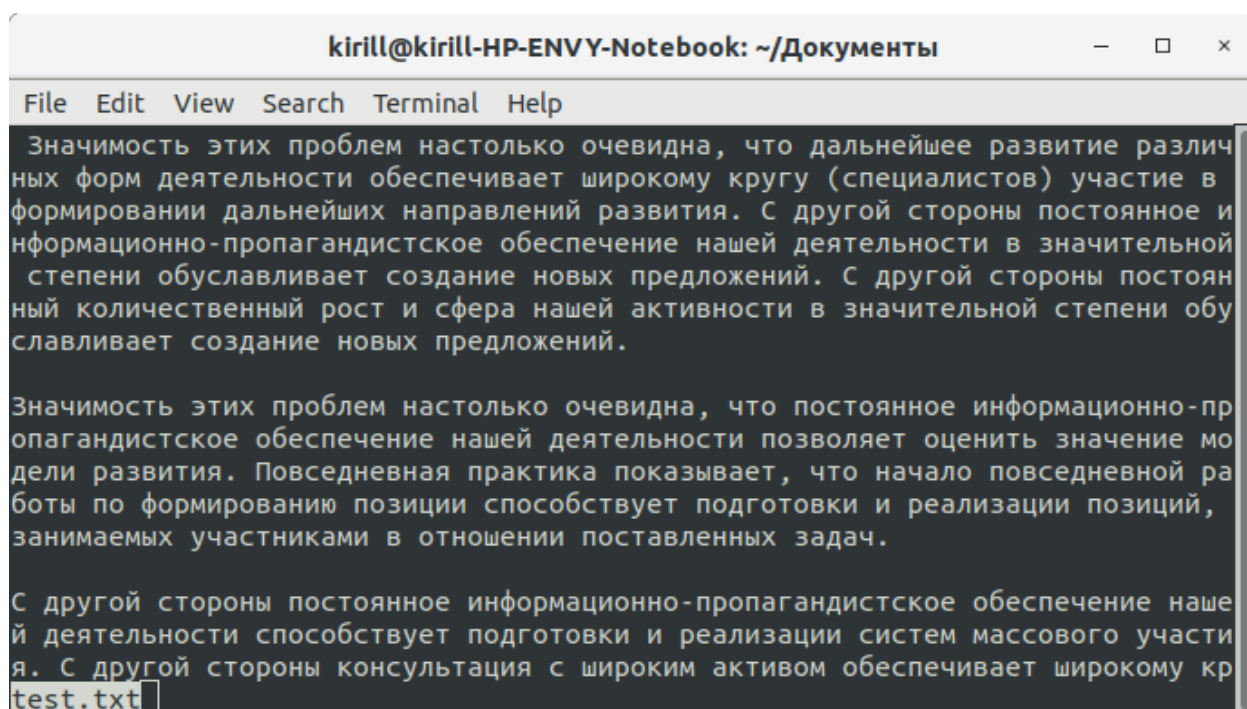


Рисунок 1.5 – Пример работы с командой less

Во время выполнения команды less можно пользоваться интерактивными командами. Найдём вхождения слова "другой" вниз по тексту, для этого введём /другой. Результат выполнения поиска представлен на рисунке 1.6.

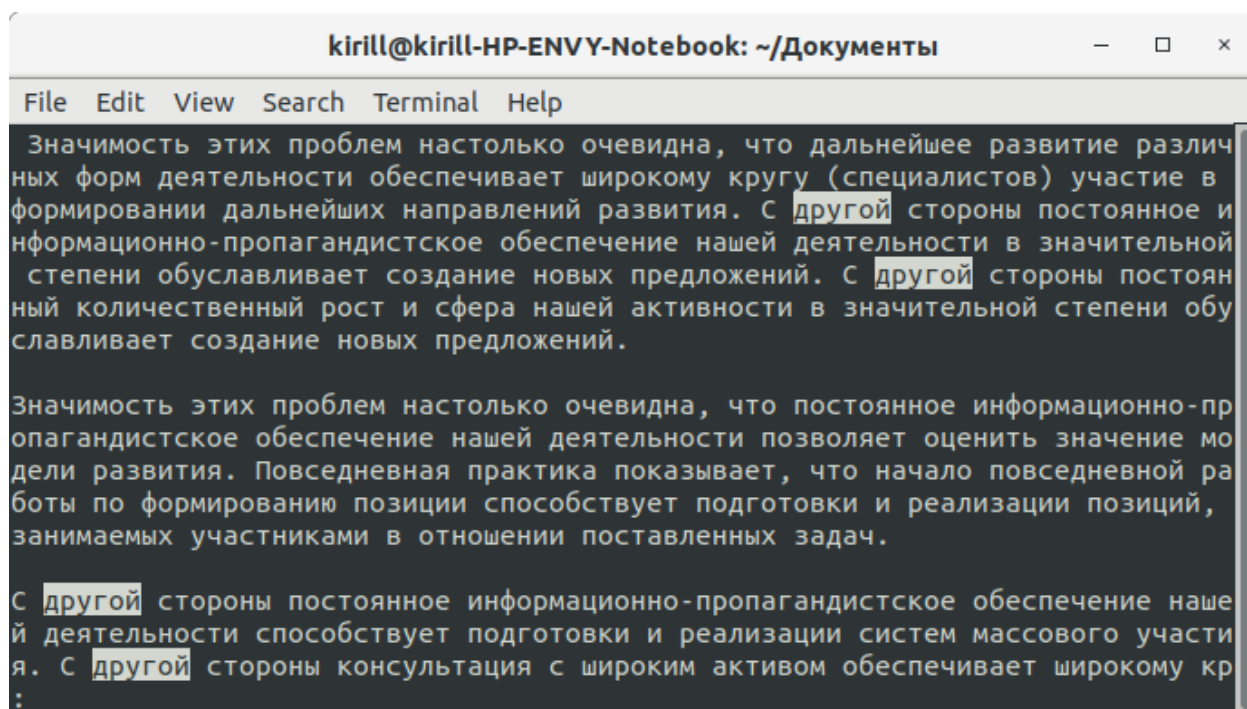


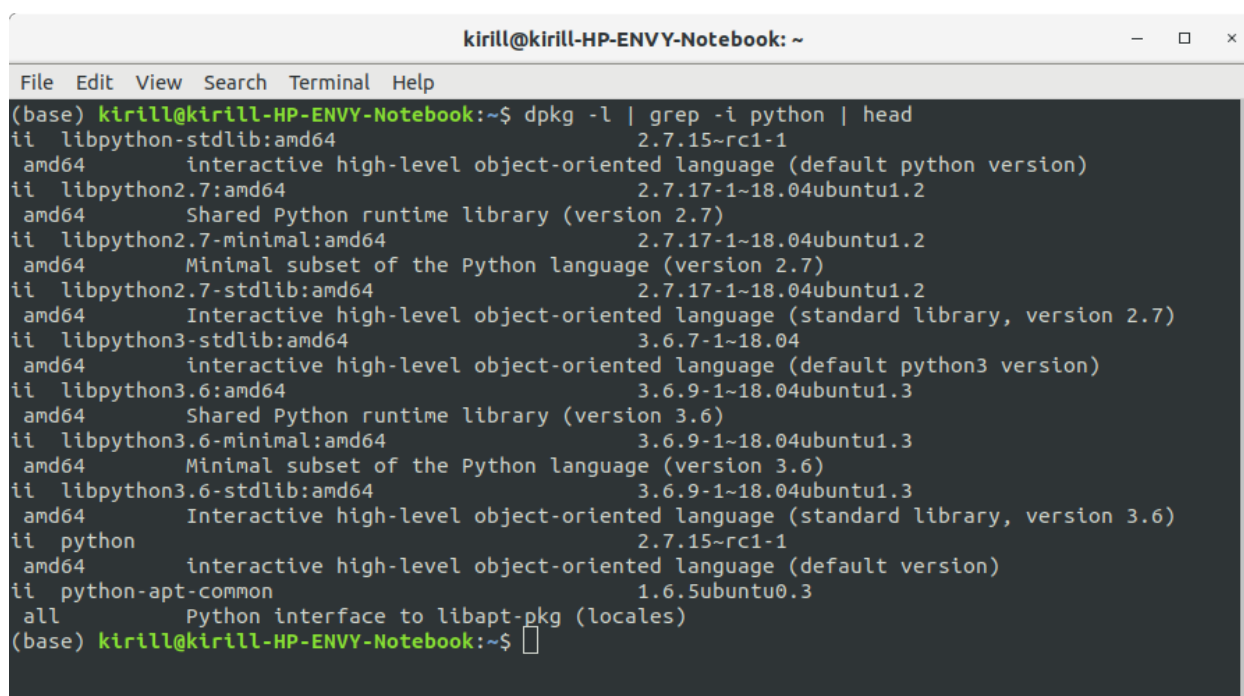
Рисунок 1.6 – Пример работы с поиском внутри команды less

Утилита `grep` решает множество задач, в основном она используется для поиска строк, соответствующих строке в тексте или содержимому файлов. Также она может находить по шаблону или регулярным выражениям.

Основные опции команды:

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

Посмотрим какие `python` пакеты и библиотеки установлены в данной системе. Для этого с помощью команды `dpkg -l` получим все установленные `*.deb` пакеты, далее передадим вывод этой команды в `grep -i python`, чтобы найти все `python` пакеты, чтобы сократить вывод распечатаем первые 10 строк с помощью команды `head`. Результат выполнения представлен на рисунке 1.7.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ dpkg -l | grep -i python | head  
ii  libpython-stdlib:amd64 2.7.15-rc1-1  
amd64 interactive high-level object-oriented language (default python version)  
ii  libpython2.7:amd64 2.7.17-1~18.04ubuntu1.2  
amd64 Shared Python runtime library (version 2.7)  
ii  libpython2.7-minimal:amd64 2.7.17-1~18.04ubuntu1.2  
amd64 Minimal subset of the Python language (version 2.7)  
ii  libpython2.7-stdlib:amd64 2.7.17-1~18.04ubuntu1.2  
amd64 Interactive high-level object-oriented language (standard library, version 2.7)  
ii  libpython3-stdlib:amd64 3.6.7-1~18.04  
amd64 interactive high-level object-oriented language (default python3 version)  
ii  libpython3.6:amd64 3.6.9-1~18.04ubuntu1.3  
amd64 Shared Python runtime library (version 3.6)  
ii  libpython3.6-minimal:amd64 3.6.9-1~18.04ubuntu1.3  
amd64 Minimal subset of the Python language (version 3.6)  
ii  libpython3.6-stdlib:amd64 3.6.9-1~18.04ubuntu1.3  
amd64 Interactive high-level object-oriented language (standard library, version 3.6)  
ii  python 2.7.15-rc1-1  
amd64 interactive high-level object-oriented language (default version)  
ii  python-apt-common 1.6.5ubuntu0.3  
all Python interface to libapt-pkg (locales)  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.7 – Пример работы с командой grep

Find – это команда для поиска файлов и каталогов на основе специальных условий.

Основные опции команды:

1. -P – никогда не открывать символические ссылки
2. -L – получает информацию о файлах по символическим ссылкам.

Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.

3. -maxdepth – максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.

4. -depth – искать сначала в текущем каталоге, а потом в подкаталогах

5. -mount – искать файлы только в этой файловой системе.

6. -version – показать версию утилиты find

7. -print – выводить полные имена файлов

8. -type f – искать только файлы

9. -type d – поиск папки в Linux

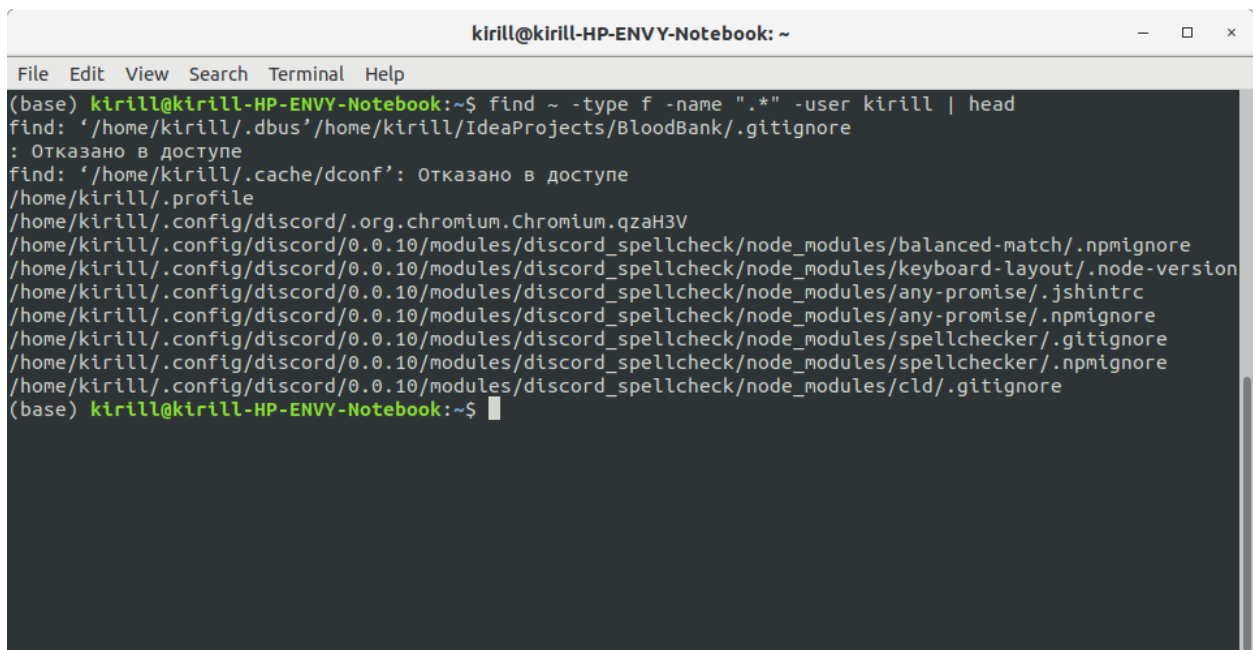
Основные критерии команды:

1. -name – поиск файлов по имени

2. -perm – поиск файлов в Linux по режиму доступа

3. -user – поиск файлов по владельцу
4. -group – поиск по группе
5. -mtime – поиск по времени модификации файла
6. -atime – поиск файлов по дате последнего чтения
7. -nogroup – поиск файлов, не принадлежащих ни одной группе
8. -nouser – поиск файлов без владельцев
9. -newer – найти файлы новее чем указанный
10. -size – поиск файлов в Linux по их размеру

Найдем все скрытые файлы пользователя kirill. Для этого воспользуемся следующей командой: `find -type f -name ".*" -user kirill`, выведем первые 10 из них с помощью команды `head`. Результат представлен на рисунке 1.8.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ find ~ -type f -name ".*" -user kirill | head  
find: '/home/kirill/.dbus'/home/kirill/IdeaProjects/BloodBank/.gitignore  
: Отказано в доступе  
find: '/home/kirill/.cache/dconf': Отказано в доступе  
/home/kirill/.profile  
/home/kirill/.config/discord/.org.chromium.Chromium.qzaH3V  
/home/kirill/.config/discord/0.0.10/modules/discord_spellcheck/node_modules/balanced-match/.npmignore  
/home/kirill/.config/discord/0.0.10/modules/discord_spellcheck/node_modules/keyboard-layout/.node-version  
/home/kirill/.config/discord/0.0.10/modules/discord_spellcheck/node_modules/any-promise/.jshintrc  
/home/kirill/.config/discord/0.0.10/modules/discord_spellcheck/node_modules/any-promise/.npmignore  
/home/kirill/.config/discord/0.0.10/modules/discord_spellcheck/node_modules/spellchecker/.gitignore  
/home/kirill/.config/discord/0.0.10/modules/discord_spellcheck/node_modules/spellchecker/.npmignore  
/home/kirill/.config/discord/0.0.10/modules/discord_spellcheck/node_modules/cld/.gitignore  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.8 – Пример работы с командой `find`

1.2 Задание 2

Операторы перенаправления способны изменять направление вывода и ввода информации. Имеются следующие операторы перенаправления ввода/вывода:

1. `>` – перенаправляет стандартный поток в файл (другой поток). При этом если файл существует, то он перезаписывается, если не существует – создается.

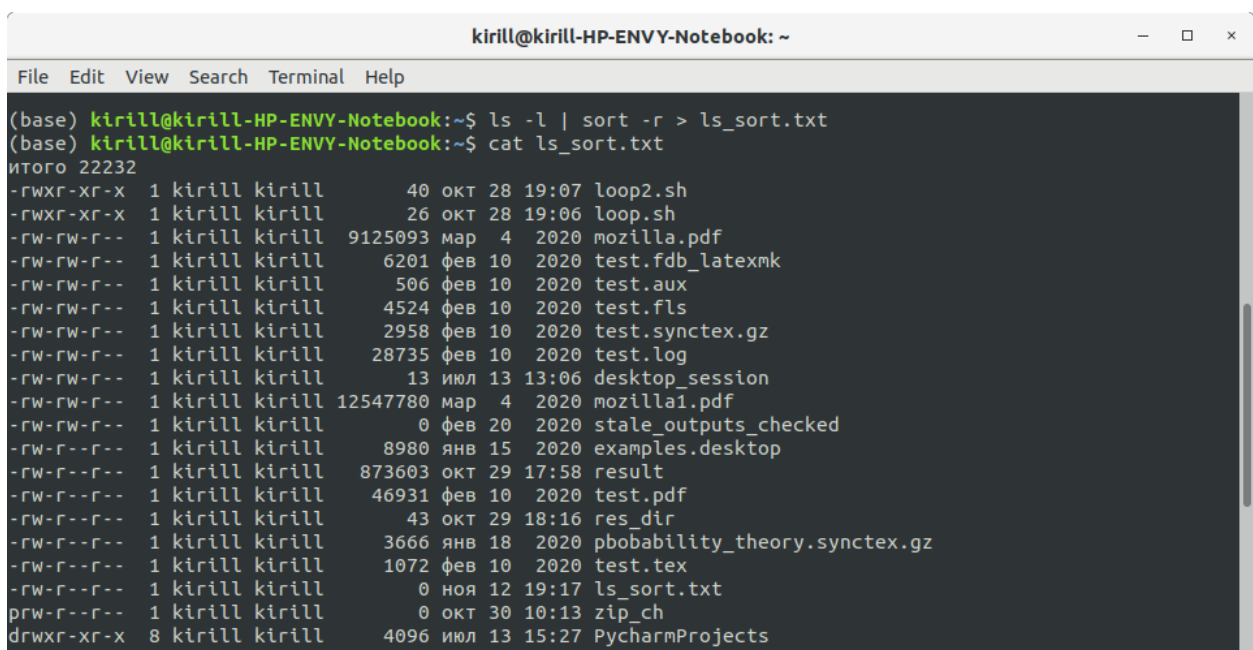
2. `>>` – перенаправляет стандартный поток в файл. При этом если файл существует, то информация добавляется в конец, если не существует – файл создается.

3. `<` – перенаправляет содержимое указанного файла на стандартный ввод программы.

4. `>&` – перенаправляет стандартные потоки вывода и ошибок друг в друга.

Конвейер (англ. pipeline) в терминологии операционных систем семейства Unix — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса. Конвейеры имеют следующий синтаксис `<команда> | <команда> | ... | <команда>`.

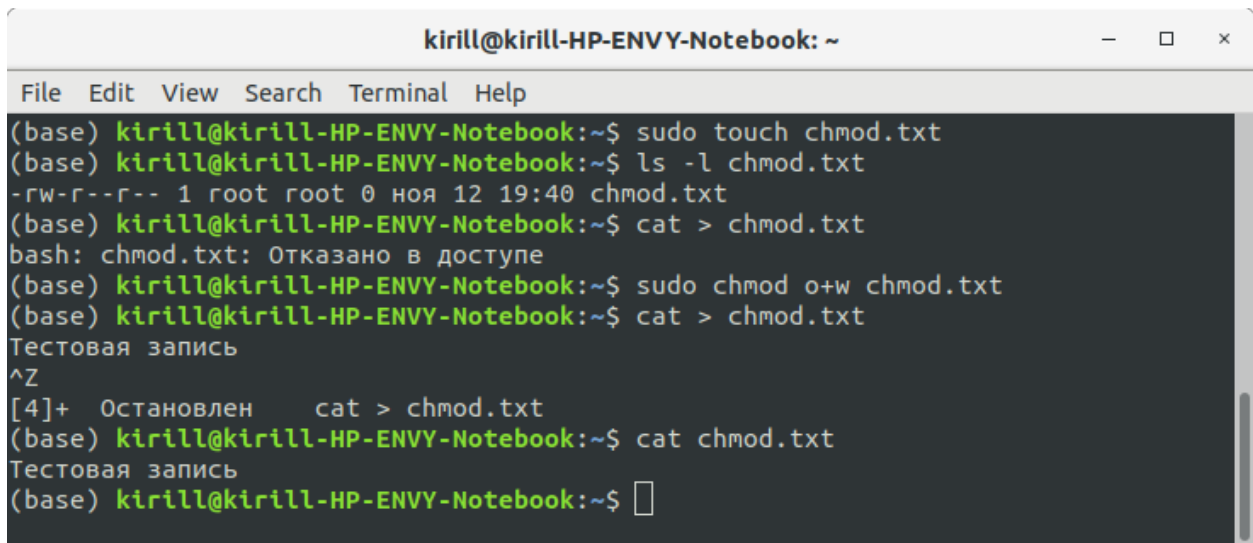
Воспользуемся конвейерами, чтобы перенаправит вывод команды `ls -l` в команду `sort`, а затем перенаправим вывод `sort` в файл. Результат представлен на рисунке 1.9.



```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ls -l | sort -r > ls_sort.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$ cat ls_sort.txt  
итого 22232  
-rwxr-xr-x 1 kirill kirill 40 окт 28 19:07 loop2.sh  
-rwxr-xr-x 1 kirill kirill 26 окт 28 19:06 loop.sh  
-rw-rw-r-- 1 kirill kirill 9125093 мар 4 2020 mozilla.pdf  
-rw-rw-r-- 1 kirill kirill 6201 фев 10 2020 test.fdb_latexmk  
-rw-rw-r-- 1 kirill kirill 506 фев 10 2020 test.aux  
-rw-rw-r-- 1 kirill kirill 4524 фев 10 2020 test.flx  
-rw-rw-r-- 1 kirill kirill 2958 фев 10 2020 test.synctex.gz  
-rw-rw-r-- 1 kirill kirill 28735 фев 10 2020 test.log  
-rw-rw-r-- 1 kirill kirill 13 июл 13 13:06 desktop_session  
-rw-rw-r-- 1 kirill kirill 12547780 мар 4 2020 mozilla1.pdf  
-rw-rw-r-- 1 kirill kirill 0 фев 20 2020 stale_outputs_checked  
-rw-rw-r-- 1 kirill kirill 8980 янв 15 2020 examples.desktop  
-rw-rw-r-- 1 kirill kirill 873603 окт 29 17:58 result  
-rw-rw-r-- 1 kirill kirill 46931 фев 10 2020 test.pdf  
-rw-rw-r-- 1 kirill kirill 43 окт 29 18:16 res_dir  
-rw-rw-r-- 1 kirill kirill 3666 янв 18 2020 pbobability_theory.synctex.gz  
-rw-rw-r-- 1 kirill kirill 1072 фев 10 2020 test.tex  
-rw-rw-r-- 1 kirill kirill 0 ноя 12 19:17 ls_sort.txt  
prw-rw-r-- 1 kirill kirill 0 окт 30 10:13 zip_ch  
drwxr-xr-x 8 kirill kirill 4096 июл 13 15:27 PycharmProjects
```

Рисунок 1.9 – Пример работы с конвейерами и перенапрвлением вывода

1.3 Задание 3

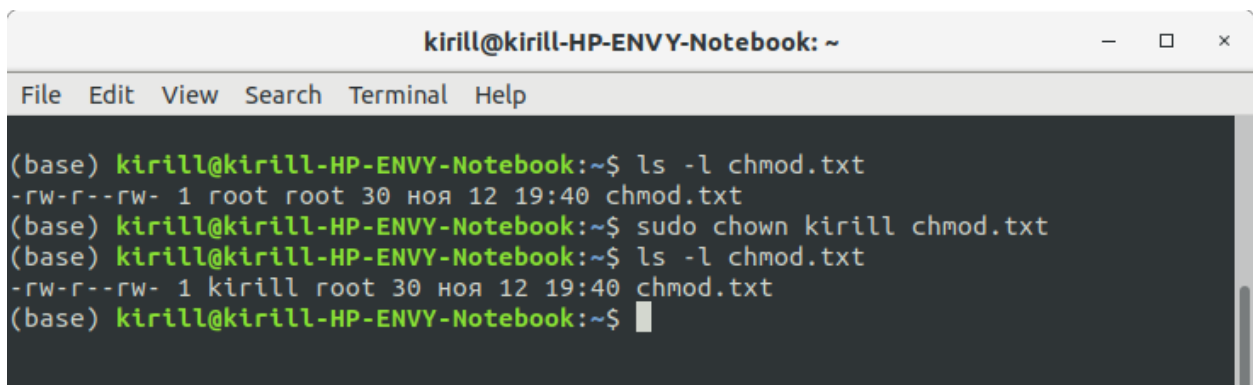


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sudo touch chmod.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ls -l chmod.txt  
-rw-r--r-- 1 root root 0 ноя 12 19:40 chmod.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$ cat > chmod.txt  
bash: chmod.txt: Отказано в доступе  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sudo chmod o+w chmod.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$ cat > chmod.txt  
Тестовая запись  
^Z  
[4]+ Остановлен cat > chmod.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$ cat chmod.txt  
Тестовая запись  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.10 – Пример работы с командой chmod

Создадим файл `chmod.txt` с правами суперпользователя `root`, для этого выполним команду `sudo touch chmod.txt`. Теперь с помощью команды `cat` попробуем записать что-нибудь в этот файл, как видно на рисунке 1.10 нам отказано в доступе, т.к. у пользователя `kirill` нет прав на запись в этот файл. Чтобы исправить данную ситуацию выдадим ему эти права с помощью команды `sudo chmod o+w chmod.txt`. Произведем еще одну тестовую запись, которая теперь происходит успешно.

Теперь воспользуемся командой `chown`, чтобы изменить владельца файла `chmod.txt` на пользователя `kirill`. Для этого используем команду `sudo chown kirill chmod.txt`. Результат выполнения команды `chown`, а также информация о владении файлом представлена на рисунке 1.11.

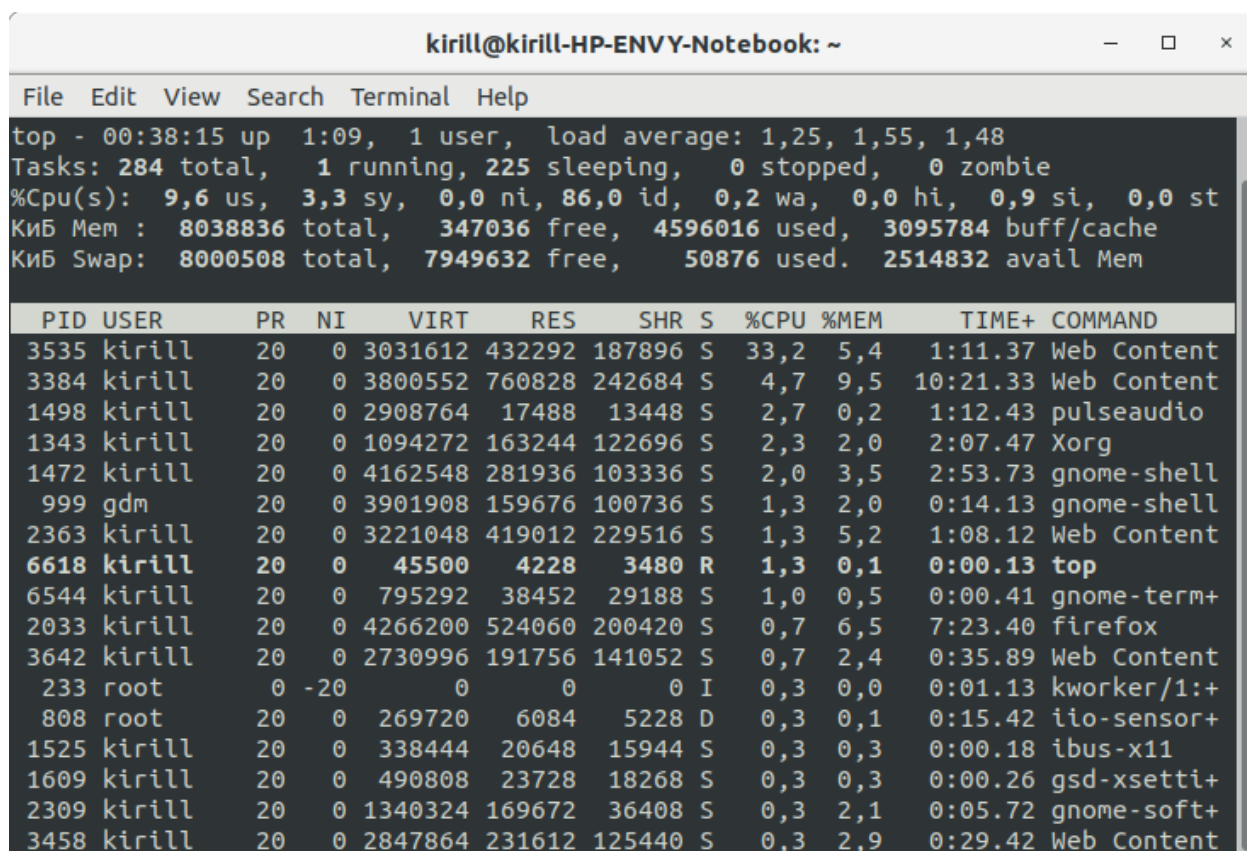


```
kirill@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ls -l chmod.txt  
-rw-r--r-- 1 root root 30 ноя 12 19:40 chmod.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$ sudo chown kirill chmod.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$ ls -l chmod.txt  
-rw-r--r-- 1 kirill root 30 ноя 12 19:40 chmod.txt  
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.11 – Пример работы с командой chown

1.4 Задание 4

Выведем в терминал все процессы, запущенные в данный момент с помощью команды `top`. результат представлен на рисунке 1.12.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 00:38:15 up 1:09, 1 user, load average: 1,25, 1,55, 1,48
Tasks: 284 total, 1 running, 225 sleeping, 0 stopped, 0 zombie
%Cpu(s): 9,6 us, 3,3 sy, 0,0 ni, 86,0 id, 0,2 wa, 0,0 hi, 0,9 si, 0,0 st
КиБ Mem : 8038836 total, 347036 free, 4596016 used, 3095784 buff/cache
КиБ Swap: 8000508 total, 7949632 free, 50876 used. 2514832 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 3535 kirill    20   0 3031612 432292 187896 S   33,2    5,4   1:11.37 Web Content
 3384 kirill    20   0 3800552 760828 242684 S    4,7    9,5  10:21.33 Web Content
 1498 kirill    20   0 2908764  17488  13448 S    2,7    0,2   1:12.43 pulseaudio
 1343 kirill    20   0 1094272 163244 122696 S    2,3    2,0   2:07.47 Xorg
 1472 kirill    20   0 4162548 281936 103336 S    2,0    3,5   2:53.73 gnome-shell
   999 gdm        20   0 3901908 159676 100736 S    1,3    2,0   0:14.13 gnome-shell
 2363 kirill    20   0 3221048 419012 229516 S    1,3    5,2   1:08.12 Web Content
 6618 kirill    20   0   45500    4228   3480 R    1,3    0,1   0:00.13 top
 6544 kirill    20   0  795292   38452  29188 S    1,0    0,5   0:00.41 gnome-term+
 2033 kirill    20   0 4266200 524060 200420 S    0,7    6,5   7:23.40 firefox
 3642 kirill    20   0 2730996 191756 141052 S    0,7    2,4   0:35.89 Web Content
   233 root       0 -20      0      0      0 I    0,3    0,0   0:01.13 kworker/1:++
   808 root      20   0  269720    6084   5228 D    0,3    0,1   0:15.42 iio-sensor+
 1525 kirill    20   0  338444   20648  15944 S    0,3    0,3   0:00.18 ibus-x11
 1609 kirill    20   0  490808   23728  18268 S    0,3    0,3   0:00.26 gsd-xsetti+
 2309 kirill    20   0 1340324 169672  36408 S    0,3    2,1   0:05.72 gnome-soft+
 3458 kirill    20   0 2847864 231612 125440 S    0,3    2,9   0:29.42 Web Content
```

Рисунок 1.12 – Процессы выполняемые в данный момент

Теперь отсортируем процессы по полю `PID`, для этого во время выполнения команды `top` нажмем комбинацию клавиш `Shift+F`, затем выберем из предложенного списка `PID`, подтвердим выбор нажатием клавиши `s` и выйдем из текущего окна, нажав клавишу `q`. Страница выбора поля для сортировки представлено на рисунке 1.13, результат сортировки представлен на рисунке 1.14.

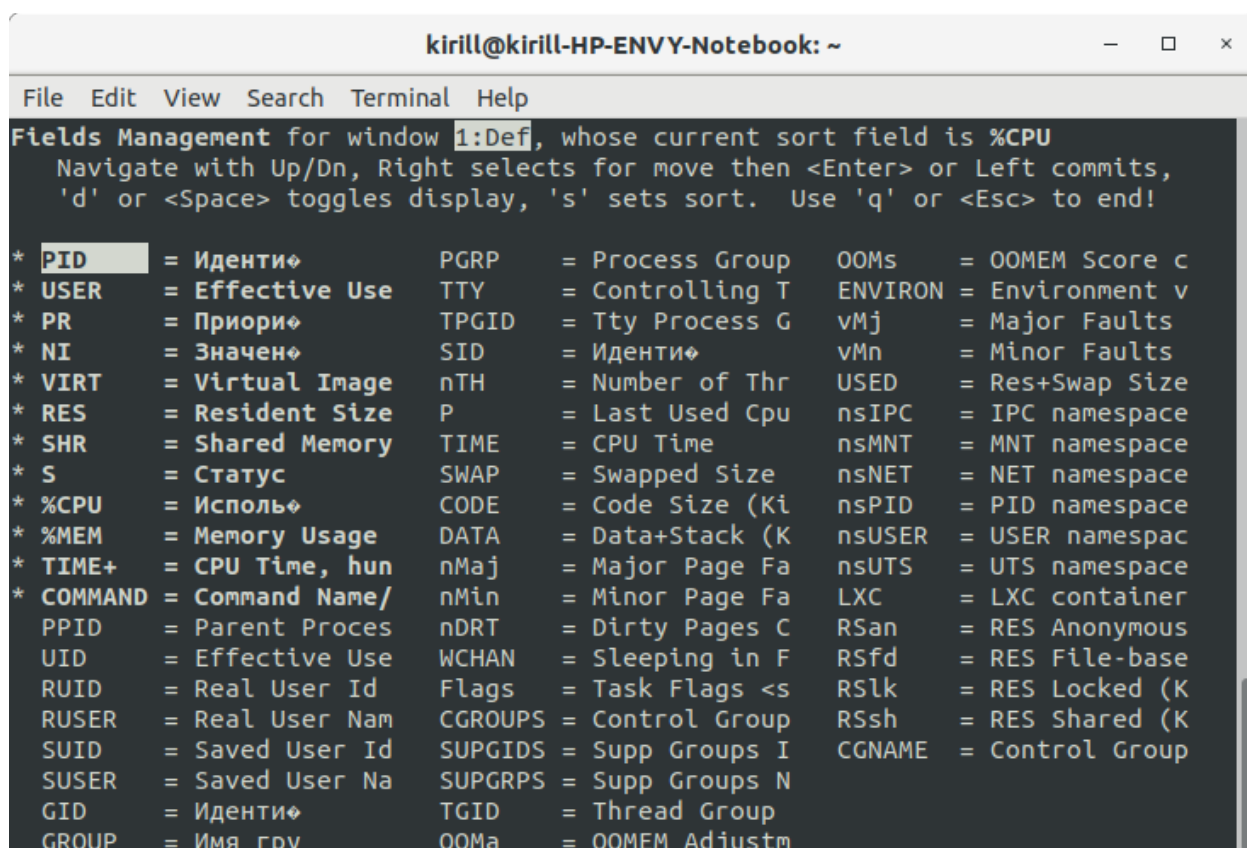


Рисунок 1.13 – Страница выбора поля для сортировки

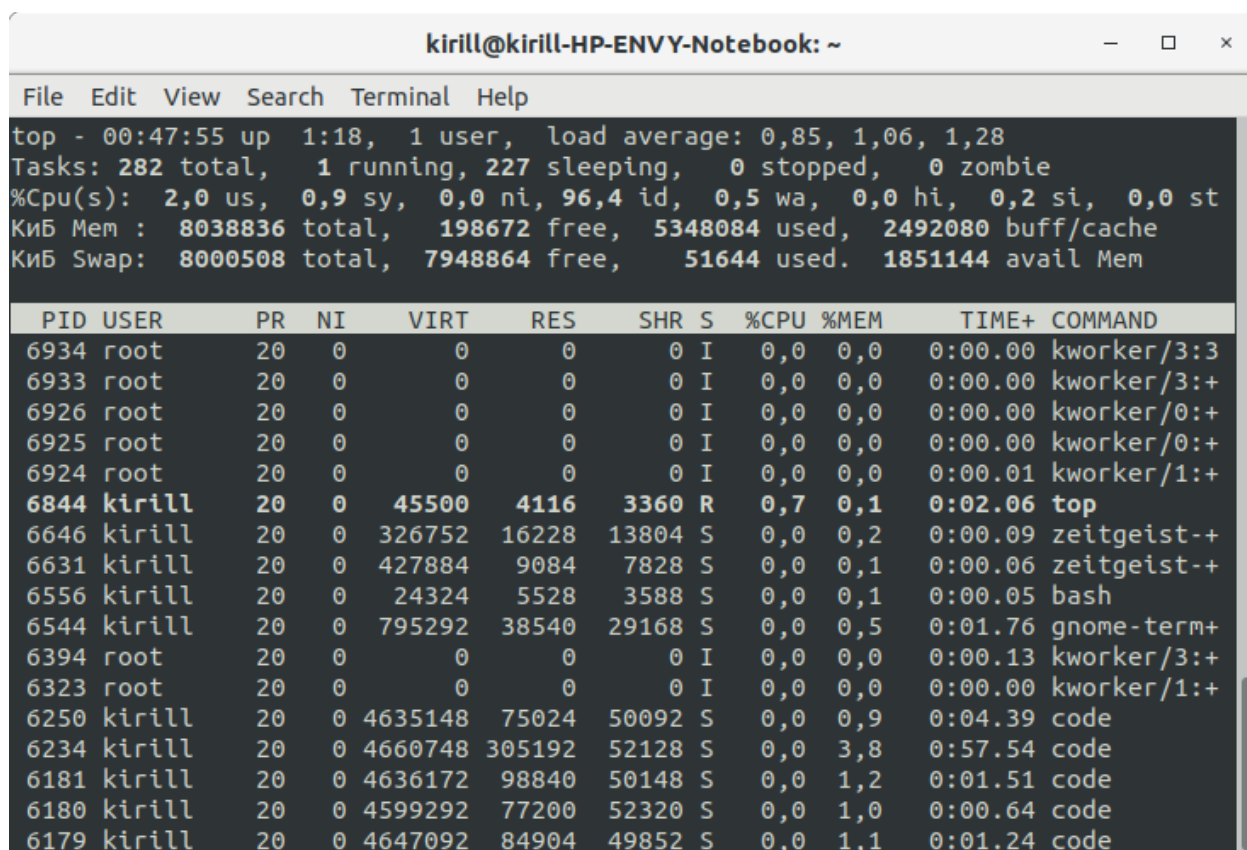


Рисунок 1.14 – Процессы, отсортированные по PID

Выведем все процессы пользователя kirill, для этого выполним команду `top -u kirill`. Результат выполнения данной команды представлен на рисунке 1.15.

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 00:54:45 up 1:25, 1 user, load average: 2,47, 1,90, 1,57
Tasks: 280 total, 1 running, 227 sleeping, 0 stopped, 0 zombie
%Cpu(s): 9,0 us, 1,9 sy, 0,0 ni, 88,1 id, 0,4 wa, 0,0 hi, 0,6 si, 0,0 st
КиБ Mem : 8038836 total, 350204 free, 5398576 used, 2290056 buff/cache
КиБ Swap: 8000508 total, 7940672 free, 59836 used. 1812528 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1472 kirill    20   0 4165800 286840 106912 S   16,2   3,6   3:50.26 gnome-shell
 2363 kirill    20   0 3231288 445456 229516 S    5,9   5,5   1:24.01 Web Content
 2033 kirill    20   0 4195000 550300 179628 S    4,6   6,8   8:32.16 firefox
 6120 kirill    20   0 18,716g 292688 111632 S    4,0   3,6   2:42.98 code
 3384 kirill    20   0 3801240 762132 243260 S    3,6   9,5  10:58.77 Web Content
 1343 kirill    20   0 1094220 163336 122788 S    2,3   2,0   2:42.10 Xorg
 5808 kirill    20   0 438920 69012 49924 S    2,3   0,9   0:33.86 code
 1498 kirill    20   0 3171344 18520 14240 S    1,0   0,2   1:28.55 pulseaudio
 1515 kirill    20   0 429928 7716 5884 S    1,0   0,1   0:05.00 ibus-daemon
 7592 kirill    20   0 45500 4296 3544 R    1,0   0,1   0:00.06 top
 3535 kirill    20   0 3038820 307272 137316 S    0,7   3,8   3:37.06 Web Content
 3642 kirill    20   0 3551800 941108 138892 S    0,7  11,7   0:59.87 Web Content
 1726 kirill    20   0 199352 6128 5376 S    0,3   0,1   0:01.36 ibus-engin+
 2309 kirill    20   0 1340324 169672 36408 S    0,3   2,1   0:05.79 gnome-soft+
 3458 kirill    20   0 2834352 209268 118028 S    0,3   2,6   0:31.62 Web Content
 5748 kirill    20   0 4965992 169360 103120 S    0,3   2,1   0:18.27 code
 6544 kirill    20   0 795292 38660 29236 S    0,3   0,5   0:02.23 gnome-term+

```

Рисунок 1.15 – Процессы пользователя kirill

Выделим процессы, которые выполняются в данный момент. Для этого во время выполнения команды `top` необходимо нажать клавишу `z`. Результат выполнения представлен на рисунке 1.16.

kirill@kirill-HP-ENVY-Notebook: ~											
File Edit View Search Terminal Help											
top - 00:59:38 up 1:30, 1 user, load average: 2,96, 2,33, 1,82											
Tasks: 281 total, 2 running, 226 sleeping, 0 stopped, 0 zombie											
%Cpu(s): 17,3 us, 3,7 sy, 0,0 ni, 76,6 id, 0,2 wa, 0,0 hi, 2,1 si, 0,0 st											
КиБ Mem : 8038836 total, 404288 free, 5452660 used, 2181888 buff/cache											
КиБ Swap: 8000508 total, 7939392 free, 61116 used. 1644388 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3535	kirill	20	0	3125540	495852	169820	S	58,5	6,2	5:44.28	Web Content
2033	kirill	20	0	4166408	548732	163256	S	15,9	6,8	9:07.21	firefox
1472	kirill	20	0	4165760	285196	106200	S	6,3	3,5	4:10.83	gnome-shell
1343	kirill	20	0	1094328	163276	122728	S	2,7	2,0	2:53.88	Xorg
3384	kirill	20	0	3819696	780680	259996	R	2,3	9,7	11:08.12	Web Content
2363	kirill	20	0	3184148	399416	173860	S	1,0	5,0	1:29.07	Web Content
1498	kirill	20	0	3171344	18520	14240	S	0,7	0,2	1:30.87	pulseaudio
3642	kirill	20	0	3525180	913320	111392	S	0,7	11,4	1:01.60	Web Content
7832	kirill	20	0	45500	4292	3544	R	0,7	0,1	0:00.13	top
808	root	20	0	269720	6084	5228	S	0,3	0,1	0:19.37	iiio-sensor+
1515	kirill	20	0	429928	7716	5884	S	0,3	0,1	0:05.18	ibus-daemon
1632	kirill	20	0	662636	22908	17556	S	0,3	0,3	0:00.71	gsd-color
1794	kirill	20	0	4347956	226588	40024	S	0,3	2,8	0:23.52	jetbrains-+
4737	root	20	0	0	0	0	I	0,3	0,0	0:02.75	kworker/u8+
6139	kirill	20	0	4756224	174524	56016	S	0,3	2,2	0:51.34	code
6544	kirill	20	0	795292	38696	29236	S	0,3	0,5	0:02.57	gnome-term+
7617	root	20	0	0	0	0	I	0,3	0,0	0:00.64	kworker/2:+

Рисунок 1.16 – Выделение выполняемых процессов

Чтобы отобразить абсолютные пути запущенных процессов, необходимо во время выполнения команды top нажать клавишу с. Результат выполнения представлен на рисунке 1.17.

kirill@kirill-HP-ENVY-Notebook: ~											
File Edit View Search Terminal Help											
top - 01:03:04 up 1:33, 1 user, load average: 2,07, 2,14, 1,85											
Tasks: 286 total, 2 running, 228 sleeping, 0 stopped, 0 zombie											
%Cpu(s): 14,8 us, 2,0 sy, 0,0 ni, 81,0 id, 0,9 wa, 0,0 hi, 1,3 si, 0,0 st											
КиБ Mem : 8038836 total, 355144 free, 5632256 used, 2051436 buff/cache											
КиБ Swap: 8000508 total, 7934784 free, 65724 used. 1604304 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3535	kirill	20	0	3139652	462072	162768	R	41,2	5,7	6:55.00	/usr/lib/f+
6120	kirill	20	0	18,722g	282916	111532	S	9,3	3,5	4:00.86	/snap/code+
1472	kirill	20	0	4166616	342660	105108	S	5,6	4,3	4:25.38	/usr/bin/g+
3384	kirill	20	0	3810036	766768	246352	S	3,0	9,5	11:14.60	/usr/lib/f+
5808	kirill	20	0	441576	72180	53088	S	3,0	0,9	0:45.96	/snap/code+
1343	kirill	20	0	1094360	162712	122164	S	2,0	2,0	3:02.15	/usr/lib/x+
1498	kirill	20	0	3171344	17892	13612	S	1,0	0,2	1:32.46	/usr/bin/p+
7832	kirill	20	0	45500	4292	3544	R	1,0	0,1	0:01.26	top
999	gdm	20	0	3901908	158256	99300	S	0,7	2,0	0:17.09	/usr/bin/g+
2033	kirill	20	0	4154008	551796	158704	S	0,7	6,9	9:20.39	/usr/lib/f+
2363	kirill	20	0	3191316	401856	170236	S	0,7	5,0	1:31.50	/usr/lib/f+
6139	kirill	20	0	4756224	172076	55004	S	0,7	2,1	0:59.63	/snap/code+
11	root	20	0	0	0	0	I	0,3	0,0	0:07.03	[rcu_sched]
785	root	20	0	32540	2744	2464	S	0,3	0,0	0:00.01	/usr/sbin/+
808	root	20	0	269720	6080	5224	S	0,3	0,1	0:19.87	/usr/sbin/+
1794	kirill	20	0	4347956	225196	38632	S	0,3	2,8	0:23.69	./jetbrain+
2326	kirill	20	0	2678184	140452	88660	S	0,3	1,7	0:15.08	/usr/lib/f+

Рисунок 1.17 – Вывод абсолютных путей процессов

Чтобы поменять интервал обновления списка процессов, необходимо во время выполнения команды `top` нажать клавишу `d` и ввести новое значение времени обновления. Результат выполнения представлен на рисунке 1.18.

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 01:05:18 up 1:36, 1 user, load average: 2,08, 2,11, 1,88
Tasks: 280 total, 1 running, 227 sleeping, 0 stopped, 0 zombie
%Cpu(s): 10,5 us, 1,2 sy, 0,0 ni, 87,1 id, 0,0 wa, 0,0 hi, 1,2 si, 0,0 st
КиБ Mem : 8038836 total, 371816 free, 5616208 used, 2050812 buff/cache
КиБ Swap: 8000508 total, 7934784 free, 65724 used. 1629956 avail Mem
Change delay from 3,0 to 2
  PID USER   PR   NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 3535 kirill  20    0 3135576 433828 162964 S   41,3   5,4    7:35.50 Web Content
 3384 kirill  20    0 3810036 767664 246352 S    3,0   9,5   11:18.54 Web Content
 1498 kirill  20    0 3171344 17892 13612 S    1,3   0,2    1:33.47 pulseaudio
 2363 kirill  20    0 3193364 409180 170236 S    1,0   5,1    1:33.20 Web Content
 1472 kirill  20    0 4166416 342852 105004 S    0,7   4,3    4:32.38 gnome-shell
   781 root    20    0 110480 3440 3120 S    0,3   0,0    0:00.42 irqbalance
   808 root    20    0 269720 6080 5224 S    0,3   0,1    0:20.18 iio-sensor+
 2033 kirill  20    0 4154008 558552 165440 S    0,3   6,9    9:22.38 firefox
 3642 kirill  20    0 3525120 910108 107928 S    0,3  11,3    1:03.13 Web Content
 3865 kirill  20    0 3017300 297168 134300 S    0,3   3,7    2:13.24 Web Content
 5808 kirill  20    0 438472 69088 49996 S    0,3   0,9    0:49.98 code
 6120 kirill  20    0 18,719g 284480 108388 S    0,3   3,5    4:24.29 code
 6544 kirill  20    0 795428 38564 29060 S    0,3   0,5    0:02.96 gnome-term+
 7832 kirill  20    0 45632 4500 3604 R    0,3   0,1    0:02.19 top
    1 root    20    0 225600 7700 5548 S    0,0   0,1    0:05.91 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
  
```

Рисунок 1.18 – Изменение интервала оновления

Чтобы послать процессу какой-либо сигнал, необходимо во время выполнения команды `top` нажать клавишу `k`, затем ввести PID целевого процесса и сигнал, который необходимо отомлать. Результат выполнения представлен на рисунках 1.19 и 1.20.

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 01:07:33 up 1:38, 1 user, load average: 1,72, 1,95, 1,85
Tasks: 283 total, 1 running, 227 sleeping, 0 stopped, 0 zombie
%Cpu(s): 16,8 us, 3,0 sy, 0,0 ni, 78,6 id, 0,5 wa, 0,0 hi, 1,1 si, 0,0 st
КиБ Mem : 8038836 total, 365040 free, 5616796 used, 2057000 buff/cache
КиБ Swap: 8000508 total, 7934272 free, 66236 used. 1618040 avail Mem
PID to signal/kill [default pid = 3535] 3642

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3535	kirill	20	0	3135576	416456	162876	S	44,1	5,2	8:16.04	Web Content
1472	kirill	20	0	4166832	343032	105384	S	14,4	4,3	4:39.24	gnome-shell
1343	kirill	20	0	1095040	163184	122636	S	4,0	2,0	3:11.38	Xorg
2033	kirill	20	0	4155032	566604	170908	S	3,5	7,0	9:23.79	firefox
6120	kirill	20	0	18,723g	289680	111896	S	3,0	3,6	4:48.30	code
3384	kirill	20	0	3810036	766240	246352	S	2,5	9,5	11:22.59	Web Content
6544	kirill	20	0	795428	38624	29060	S	2,0	0,5	0:03.17	gnome-term+
2363	kirill	20	0	3193364	415116	170236	S	1,0	5,2	1:34.50	Web Content
5808	kirill	20	0	442824	71340	52248	S	1,0	0,9	0:53.77	code
7832	kirill	20	0	45632	4500	3604	R	1,0	0,1	0:03.07	top
808	root	20	0	269720	6080	5224	S	0,5	0,1	0:20.52	ilo-sensor+
1498	kirill	20	0	3171344	17892	13612	S	0,5	0,2	1:34.51	pulseaudio
1632	kirill	20	0	662636	22636	17284	S	0,5	0,3	0:00.75	gsd-color
3642	kirill	20	0	3523072	906760	107928	S	0,5	11,3	1:03.64	Web Content
3865	kirill	20	0	3021396	316948	134300	S	0,5	3,9	2:13.67	Web Content
5748	kirill	20	0	4967016	168204	102376	S	0,5	2,1	0:26.23	code
7020	root	20	0	0	0	0	I	0,5	0,0	0:00.47	kworker/2:+

Рисунок 1.19 – Выбор процесса для отправки сигнала

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
top - 01:07:33 up 1:38, 1 user, load average: 1,72, 1,95, 1,85
Tasks: 283 total, 1 running, 227 sleeping, 0 stopped, 0 zombie
%Cpu(s): 16,8 us, 3,0 sy, 0,0 ni, 78,6 id, 0,5 wa, 0,0 hi, 1,1 si, 0,0 st
КиБ Mem : 8038836 total, 365040 free, 5616796 used, 2057000 buff/cache
КиБ Swap: 8000508 total, 7934272 free, 66236 used. 1618040 avail Mem
Send pid 3642 signal [15/sigterm] sigkill

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3535	kirill	20	0	3135576	416456	162876	S	44,1	5,2	8:16.04	Web Content
1472	kirill	20	0	4166832	343032	105384	S	14,4	4,3	4:39.24	gnome-shell
1343	kirill	20	0	1095040	163184	122636	S	4,0	2,0	3:11.38	Xorg
2033	kirill	20	0	4155032	566604	170908	S	3,5	7,0	9:23.79	firefox
6120	kirill	20	0	18,723g	289680	111896	S	3,0	3,6	4:48.30	code
3384	kirill	20	0	3810036	766240	246352	S	2,5	9,5	11:22.59	Web Content
6544	kirill	20	0	795428	38624	29060	S	2,0	0,5	0:03.17	gnome-term+
2363	kirill	20	0	3193364	415116	170236	S	1,0	5,2	1:34.50	Web Content
5808	kirill	20	0	442824	71340	52248	S	1,0	0,9	0:53.77	code
7832	kirill	20	0	45632	4500	3604	R	1,0	0,1	0:03.07	top
808	root	20	0	269720	6080	5224	S	0,5	0,1	0:20.52	ilo-sensor+
1498	kirill	20	0	3171344	17892	13612	S	0,5	0,2	1:34.51	pulseaudio
1632	kirill	20	0	662636	22636	17284	S	0,5	0,3	0:00.75	gsd-color
3642	kirill	20	0	3523072	906760	107928	S	0,5	11,3	1:03.64	Web Content
3865	kirill	20	0	3021396	316948	134300	S	0,5	3,9	2:13.67	Web Content
5748	kirill	20	0	4967016	168204	102376	S	0,5	2,1	0:26.23	code
7020	root	20	0	0	0	0	I	0,5	0,0	0:00.47	kworker/2:+

Рисунок 1.20 – Выбор сигнала для отправки процессу

Чтобы отсортировать процессы по нагрузке на процессор, необходимо во время выполнения команды `top` нажать комбинацию клавиш `Shift+p`. Результат выполнения представлен на рисунке 1.21.

```

top - 01:16:24 up 1:47, 1 user, load average: 0,89, 1,76, 1,87
Tasks: 281 total, 1 running, 227 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3,7 us, 1,6 sy, 0,0 ni, 94,2 id, 0,0 wa, 0,0 hi, 0,5 si, 0,0 st
КиБ Mem : 8038836 total, 480628 free, 5500868 used, 2057340 buff/cache
КиБ Swap: 8000508 total, 7934272 free, 66236 used. 1764728 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 3384 kirill    20   0 3810036 767420 246352 S   6,6   9,5   11:40.27 Web Content
 3535 kirill    20   0 3090360 339568 132068 S   4,7   4,2    9:56.44 Web Content
 3865 kirill    20   0 3019348 305224 134300 S   1,9   3,8    2:16.35 Web Content
9044 kirill    20   0  45500   4236   3488 R   1,9   0,1    0:00.51 top
   11 root       20   0      0      0      0 I   0,9   0,0    0:07.83 rcu_sched
   787 root       20   0 560596 14856 11480 S   0,9   0,2    0:03.23 NetworkMan+
 1472 kirill    20   0 4166420 341684 104920 S   0,9   4,3    5:05.63 gnome-shell
 1498 kirill    20   0 3171344 17892 13612 S   0,9   0,2    1:39.12 pulseaudio
 1515 kirill    20   0 429928   7712  5880 S   0,9   0,1    0:05.90 ibus-daemon
 2033 kirill    20   0 4138672 463124 173500 S   0,9   5,8    9:40.20 firefox
 2363 kirill    20   0 3211796 447268 170236 S   0,9   5,6    1:40.46 Web Content
    1 root       20   0 225600   7700  5548 S   0,0   0,1    0:06.42 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:++
    9 root        0 -20      0      0      0 I   0,0   0,0    0:00.00 mm_percpu_+
  
```

Рисунок 1.21 – Процессы отсортированные по нагрузке на CPU

Чтобы изменить приоритет процесса, необходимо во время выполнения команды `top` нажать клавишу `r`, затем ввести PID процесса и новый приоритет. Для примера установим процессу с PID=3535 приоритет равный 8. Результат выполнения представлен на рисунках 1.22 и 1.23.

kirill@kirill-HP-ENVY-Notebook: ~											
File Edit View Search Terminal Help											
top - 01:20:54 up 1:51, 1 user, load average: 0,37, 1,22, 1,64											
Tasks: 282 total, 2 running, 227 sleeping, 0 stopped, 0 zombie											
%Cpu(s): 3,1 us, 1,0 sy, 0,0 ni, 95,2 id, 0,1 wa, 0,0 hi, 0,5 si, 0,0 st											
КиБ Mem : 8038836 total, 516276 free, 5462224 used, 2060336 buff/cache											
КиБ Swap: 8000508 total, 7934272 free, 66236 used. 1808300 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1343	kirill	20	0	1094560	162888	122340	S	4,2	2,0	3:39.71	Xorg
3384	kirill	20	0	3810036	767392	246352	S	3,6	9,5	11:49.23	Web Content
1472	kirill	20	0	4166496	342168	104980	S	2,2	4,3	5:17.08	gnome-shell
3535	kirill	20	0	3092664	341848	132288	S	2,1	4,3	10:12.69	Web Content
1498	kirill	20	0	3171344	17892	13612	S	1,0	0,2	1:41.77	pulseaudio
2363	kirill	20	0	3221012	432404	170236	S	1,0	5,4	1:44.35	Web Content
6544	kirill	20	0	795428	38632	29060	S	0,9	0,5	0:04.83	gnome-term+
2033	kirill	20	0	4138928	458148	167708	S	0,6	5,7	9:47.76	firefox
808	root	20	0	269720	6080	5224	D	0,4	0,1	0:22.83	ilo-sensor+
1515	kirill	20	0	429928	7712	5880	S	0,4	0,1	0:06.08	ibus-daemon
3865	kirill	20	0	3017300	298896	134300	S	0,4	3,7	2:17.74	Web Content
6139	kirill	20	0	4756224	177720	55004	S	0,3	2,2	1:34.30	code
9044	kirill	20	0	45500	4236	3488	R	0,3	0,1	0:01.46	top
787	root	20	0	560596	14856	11480	S	0,1	0,2	0:03.33	NetworkMan+
1631	kirill	20	0	338052	19904	15204	S	0,1	0,2	0:00.28	gsd-clipbo+
1726	kirill	20	0	199352	6128	5376	S	0,1	0,1	0:01.65	ibus-engin+
1794	kirill	20	0	4347956	225196	38632	S	0,1	2,8	0:24.78	jetbrains-+

Рисунок 1.22 – Процессы до установки нового приоритета

kirill@kirill-HP-ENVY-Notebook: ~											
File Edit View Search Terminal Help											
top - 01:21:34 up 1:52, 1 user, load average: 0,72, 1,20, 1,61											
Tasks: 283 total, 2 running, 226 sleeping, 0 stopped, 0 zombie											
%Cpu(s): 4,0 us, 1,4 sy, 0,0 ni, 94,2 id, 0,2 wa, 0,0 hi, 0,3 si, 0,0 st											
КиБ Mem : 8038836 total, 522852 free, 5455800 used, 2060184 buff/cache											
КиБ Swap: 8000508 total, 7934272 free, 66236 used. 1815132 avail Mem											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3384	kirill	20	0	3810036	766672	246352	S	4,4	9,5	11:50.98	Web Content
1472	kirill	20	0	4166528	342236	105056	S	3,6	4,3	5:18.51	gnome-shell
2033	kirill	20	0	4138928	458228	167708	S	3,1	5,7	9:48.99	firefox
1343	kirill	20	0	1094504	162816	122268	S	2,2	2,0	3:40.57	Xorg
3535	kirill	28	8	3092664	342392	132288	S	2,1	4,3	10:13.54	Web Content
2363	kirill	20	0	3221012	433692	170236	S	1,5	5,4	1:44.94	Web Content
1498	kirill	20	0	3171344	17892	13612	S	1,2	0,2	1:42.25	pulseaudio
8856	kirill	20	0	3496024	921520	129912	S	0,9	11,5	0:12.11	Web Content
6544	kirill	20	0	795428	38632	29060	R	0,5	0,5	0:05.01	gnome-term+
5709	kirill	20	0	1088568	72824	40016	S	0,4	0,9	0:22.06	nautilus
808	root	20	0	269720	6080	5224	S	0,4	0,1	0:22.98	ilo-sensor+
6139	kirill	20	0	4756224	177720	55004	S	0,4	2,2	1:34.44	code
3865	kirill	20	0	3017300	299192	134300	S	0,3	3,7	2:17.84	Web Content
999	gdm	20	0	3901908	158248	99300	S	0,2	2,0	0:19.10	gnome-shell
1515	kirill	20	0	429928	7712	5880	S	0,2	0,1	0:06.15	ibus-daemon
1794	kirill	20	0	4347956	225196	38632	S	0,2	2,8	0:24.84	jetbrains-+
2326	kirill	20	0	2678184	142956	88660	S	0,1	1,8	0:15.87	WebExtensi+

Рисунок 1.23 – Процессы после установки нового приоритета

Чтобы сохранить список выполняемых процессов в файл воспользуемся перенаправлением ввода/вывода. Для этого выполним следующую команду: `top -n 1 -b > top-output.txt`. Затем выведем первые 10 строк полученного файла с помощью команды `head`. Результат выполнения представлен на рисунке 1.24.

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
(base) kirill@kirill-HP-ENVY-Notebook:~$ top -n 1 -b > top-output.txt
(base) kirill@kirill-HP-ENVY-Notebook:~$ head top-output.txt
top - 01:29:05 up 1:59, 1 user, load average: 1,98, 1,93, 1,82
Tasks: 280 total, 1 running, 227 sleeping, 0 stopped, 0 zombie
%Cpu(s): 21,3 us, 3,7 sy, 0,2 ni, 72,2 id, 1,2 wa, 0,0 hi, 1,4 si, 0,0 st
КиБ Mem : 8038836 total, 422336 free, 5560412 used, 2056088 buff/cache
КиБ Swap: 8000508 total, 7932992 free, 67516 used. 1675112 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 3535 kirill    28   8 3137624 423264 162556 S   27,8   5,3   11:27.85 Web Content
 9761 kirill    20   0  45492   4084   3388 R   16,7   0,1    0:00.05 top
   999 gdm        20   0 3901908 158332 99308 S    5,6   2,0    0:19.82 gnome-shell
(base) kirill@kirill-HP-ENVY-Notebook:~$

```

Рисунок 1.24 – Перенаправление вывода команды `top`

Чтобы получить справку по команде `top`, необходимо во время выполнения команды `top` нажать клавишу `h`. Результат выполнения представлен на рисунке 1.25.

```

kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
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,V,J . Toggle: 'R' Sort; 'H' Threads; 'V' Forest view; 'J' Num justify
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)
C,... . Toggle scroll coordinates msg for: up,down,left,right,home,end

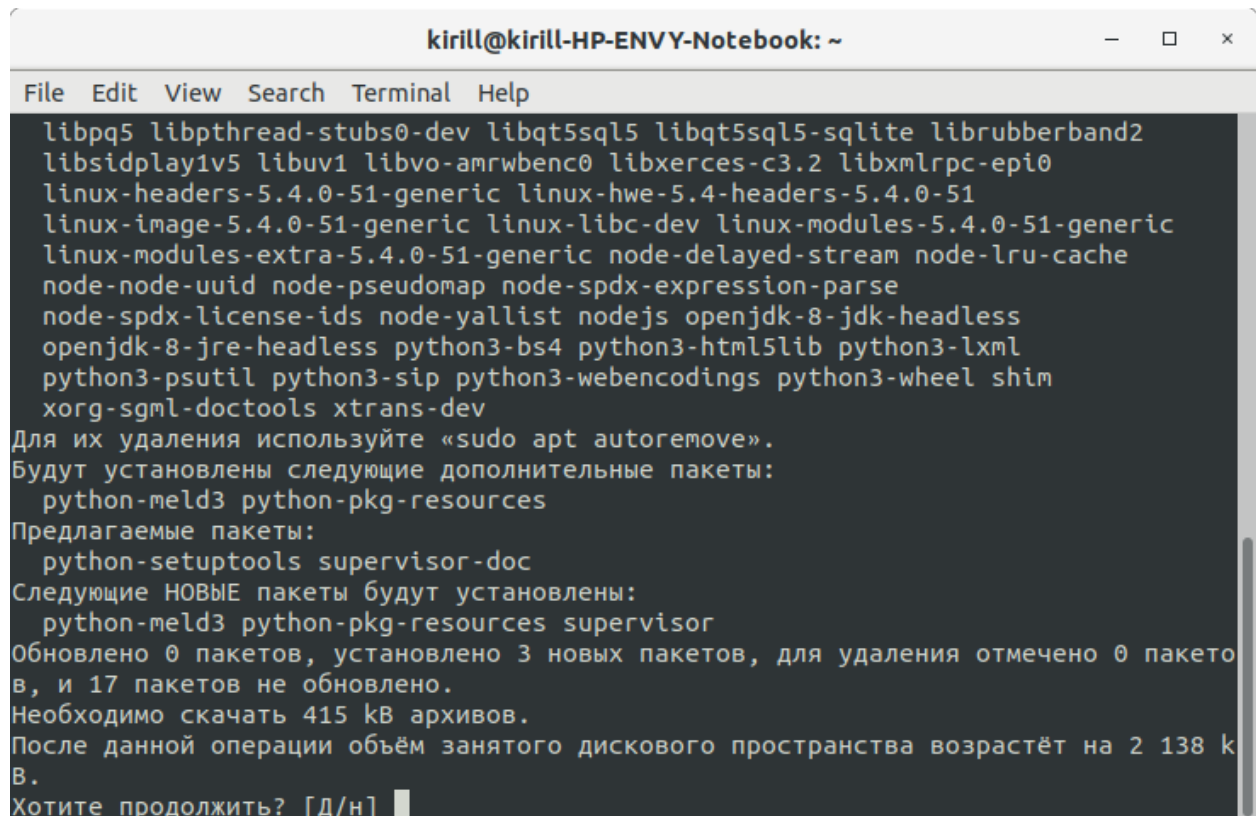
k,r Manipulate tasks: 'k' kill; 'r' renice
d or s Set update interval
W,Y Write configuration file 'W'; Inspect other output 'Y'
q Quit
(commands shown with '.' require a visible task display window)
Press 'h' or '?' for help with Windows,
Type 'q' or <Esc> to continue

```

Рисунок 1.25 – Справка по команде `top`

Вывод команды `top` будет обновляться пока не будет нажата клавиша `q`, чтобы выполнение прекратилось после определенного числа обновлений можно воспользоваться опцией `-n <число>`.

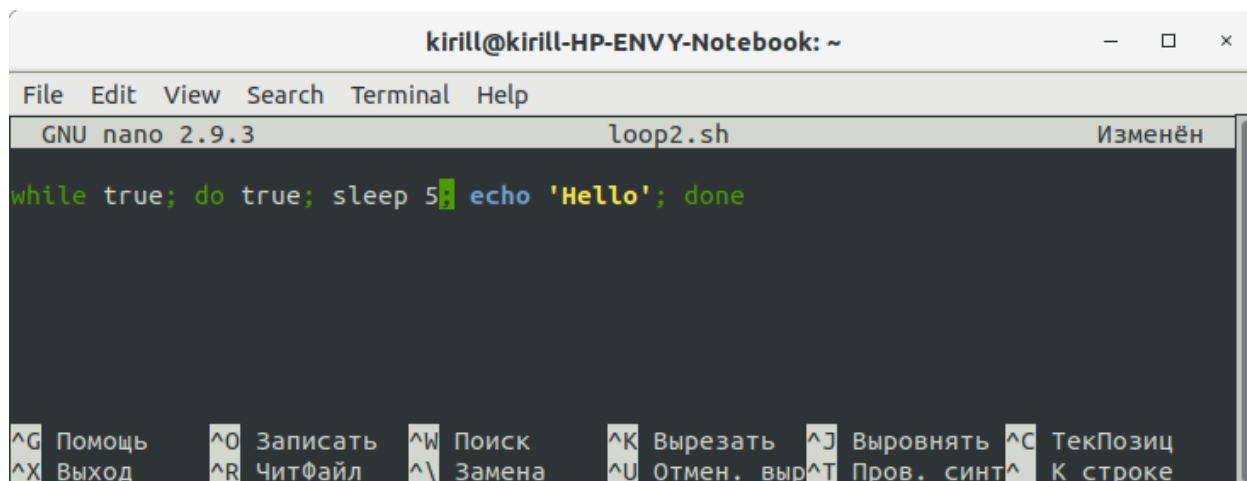
Теперь разберемся с запуском процессов в `supervision`. Для начала установим `supervisor`, для этого выполним команду `sudo apt install supervisor`. Процесс установки представлен на рисунке 1.26.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
libpq5 libpthread-stubs0-dev libqt5sql5 libqt5sql5-sqlite librubberband2
libsidplay1v5 libuv1 libvo-amrwbenc0 libxerces-c3.2 libxmlrpc-epi0
linux-headers-5.4.0-51-generic linux-hwe-5.4-headers-5.4.0-51
linux-image-5.4.0-51-generic linux-libc-dev linux-modules-5.4.0-51-generic
linux-modules-extra-5.4.0-51-generic node-delayed-stream node-lru-cache
node-node-uuid node-pseudomap node-spdx-expression-parse
node-spdx-license-ids node-yallist nodejs openjdk-8-jdk-headless
openjdk-8-jre-headless python3-bs4 python3-html5lib python3-lxml
python3-psutil python3-sip python3-webencodings python3-wheel shim
xorg-sgml-doctools xtrans-dev
Для их удаления используйте «sudo apt autoremove».
Будут установлены следующие дополнительные пакеты:
python-meld3 python-pkg-resources
Предлагаемые пакеты:
python-setuptools supervisor-doc
Следующие НОВЫЕ пакеты будут установлены:
python-meld3 python-pkg-resources supervisor
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 17 пакетов не обновлено.
Необходимо скачать 415 kB архивов.
После данной операции объем занятого дискового пространства возрастёт на 2 138 kB.
Хотите продолжить? [Д/н]
```

Рисунок 1.26 – Установка `supervisor`

Для тестирования запуска процессов в `supervision` создадим `bash` скрипт, который будет выводить слово `Hello` каждые 5 секунд. Результат создания скрипта представлен на рисунке 1.27.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 loop2.sh Изменён
while true; do true; sleep 5; echo 'Hello'; done
^G Помощь ^O Записать ^W Поиск ^K Вырезать ^J Выводить ^C ТекПозиц
^X Выход ^R ЧитФайл ^\ Замена ^U Отмен. выр ^T Пров. синт ^_ К строке
```

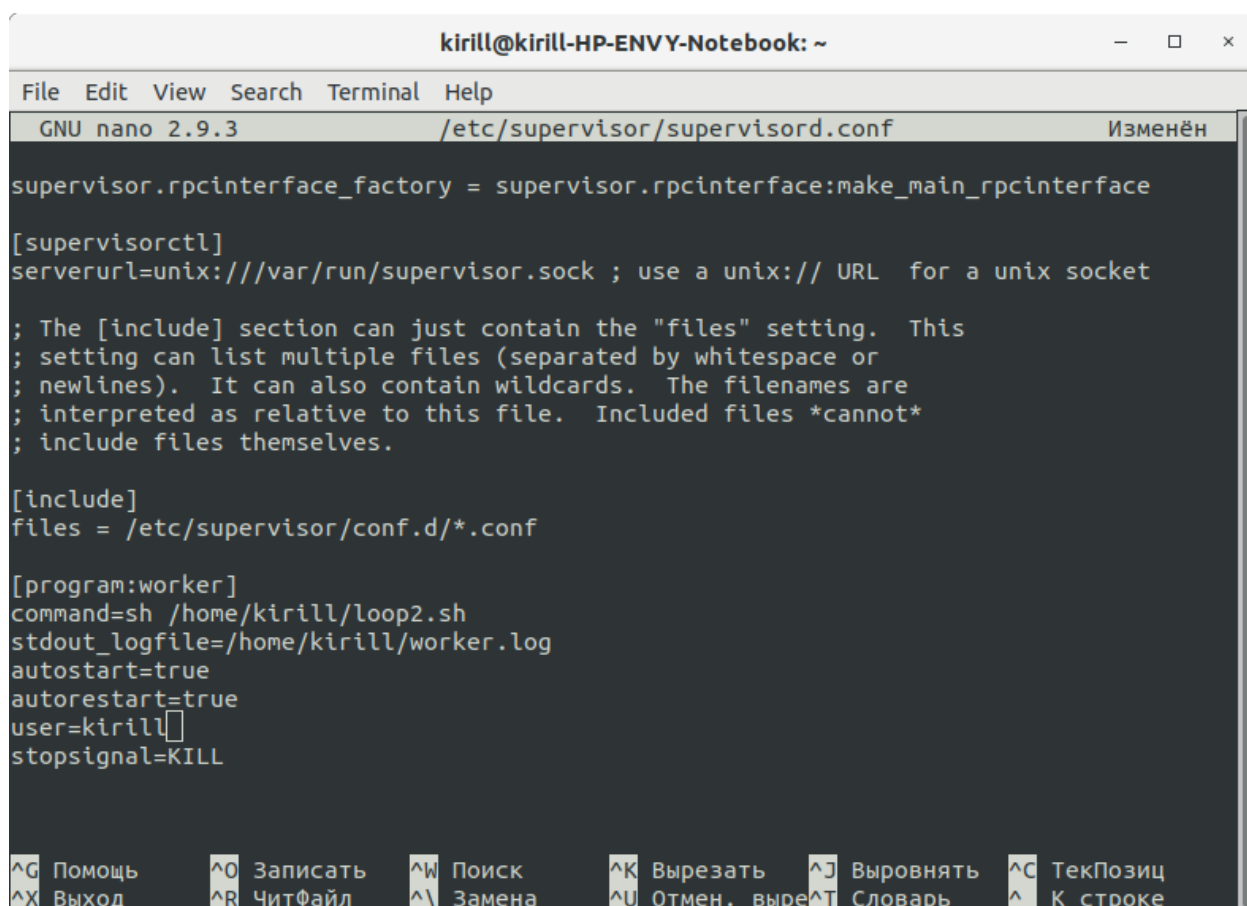
Рисунок 1.27 – Скрипт для тестирования

Теперь произведем конфигурацию supervisor и добавим созданный скрипт, чтобы он управлял им. Файл конфигурации по умолчанию находится в `/etc/supervisor/supervisord.conf`. В данном файле представлены следующие параметры:

`program:worker` — название процесса/воркера, к которому будут относиться все последующие параметры секции;

1. `command=` — команда на запуск файла, то есть путь к нужному файлу;
2. `stdout_logfile=` — вывод консоли в файл;
3. `autostart=` — запуск воркера вместе с запуском supervisor;
4. `autorestart=` — перезапуск воркера, если тот по какой-то причине упал;
5. `user=` — запуск процесса под определенным пользователем;
6. `stopsignal=` — сигнал остановки (убийства) процесса. Если не определяется, то используется команда по умолчанию — `TERM`;
7. `numprocs=` — количество инстансов заданного воркера

Теперь откроем наш файл и сконфигурируем его для запуска написанного скрипта. Результат выполнения представлен на рисунке 1.28.



```
kirill@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/supervisor/supervisord.conf Изменён

supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface

[supervisorctl]
serverurl=unix:///var/run/supervisor.sock ; use a unix:// URL  for a unix socket

; The [include] section can just contain the "files" setting.  This
; setting can list multiple files (separated by whitespace or
; newlines).  It can also contain wildcards.  The filenames are
; interpreted as relative to this file.  Included files *cannot*
; include files themselves.

[include]
files = /etc/supervisor/conf.d/*.conf

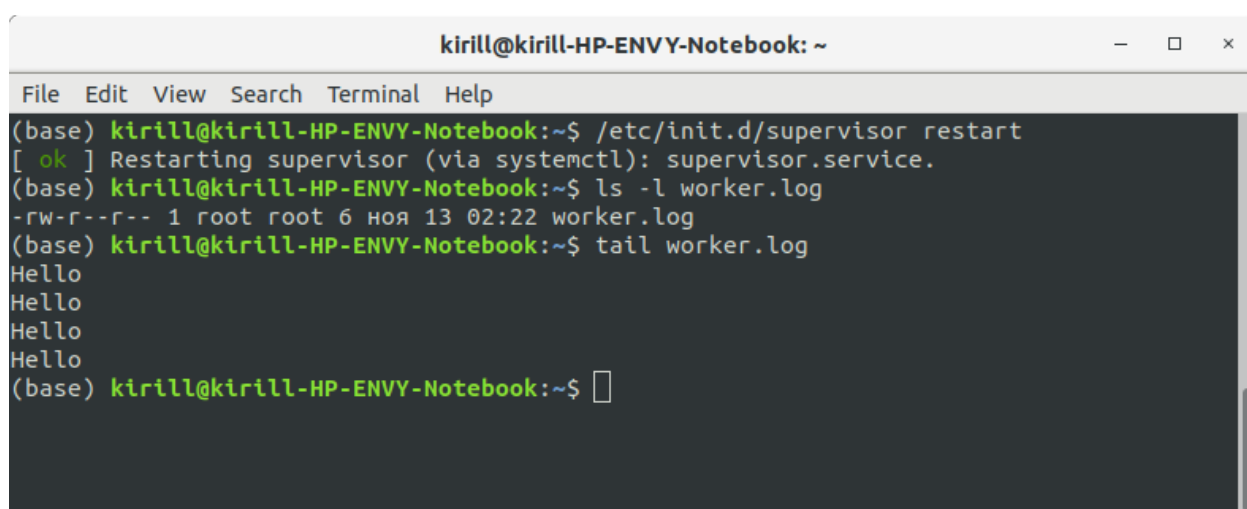
[program:worker]
command=sh /home/kirill/loop2.sh
stdout_logfile=/home/kirill/worker.log
autostart=true
autorestart=true
user=kirill
stopsignal=KILL

^G Помощь      ^O Записать    ^W Поиск      ^K Вырезать    ^J Выводить
^X Выход      ^R ЧитФайл    ^_ Замена     ^U Отмен. выр ^T Словарь    ^C ТекПозиц

```

Рисунок 1.28 – Конфигурация supervisor

После добавления процесса перезапустим supervisor командой: `/etc/init.d/supervisor restart`. После перезапуска посмотрим на содержимое файла логгирования. Результат выполнения представлен на рисунке 1.29.



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

(base) kirill@kirill-HP-ENVY-Notebook:~$ /etc/init.d/supervisor restart
[ ok ] Restarting supervisor (via systemctl): supervisor.service.
(base) kirill@kirill-HP-ENVY-Notebook:~$ ls -l worker.log
-rw-r--r-- 1 root root 6 ноя 13 02:22 worker.log
(base) kirill@kirill-HP-ENVY-Notebook:~$ tail worker.log
Hello
Hello
Hello
Hello
(base) kirill@kirill-HP-ENVY-Notebook:~$ 

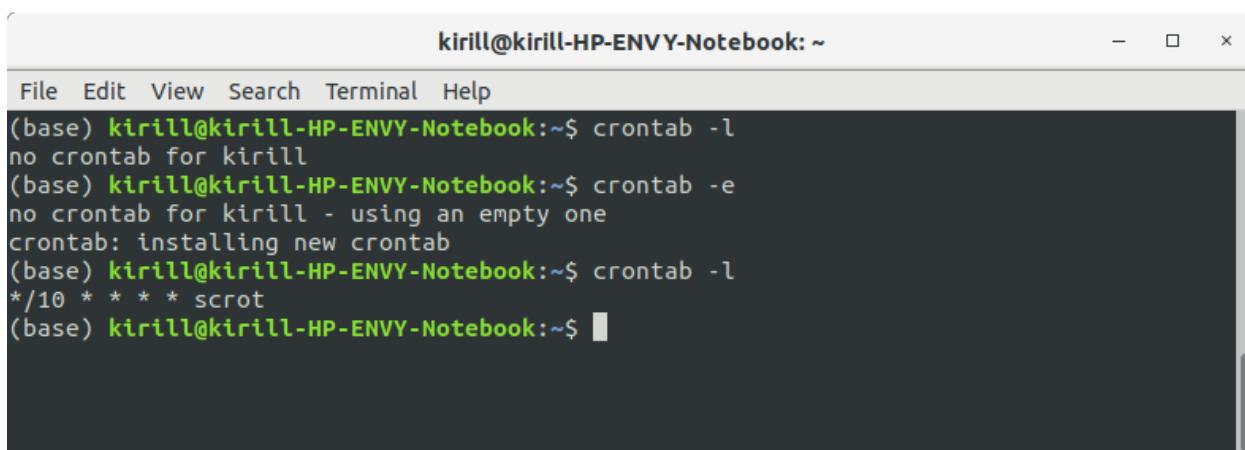
```

Рисунок 1.29 – Лог файл выполняемого процесса

1.5 Задание 5

Для запуска процессов по расписанию используется команда `cron`. `Cron` – программа-демон, предназначенная для выполнения заданий в определенное время, или через определенные промежутки времени. Для редактирования заданий используется утилита `crontab`.

Чтобы задать расписание для процесса выполним команду `crontab -e`, в открывшемся текстовом редакторе запишем расписание и команду, сделаем так, чтобы каждые 10 минут выполнялся снимок экрана, для этого запишем строку `*/10 * * * * scrot`, затем проверим, что изменения успешно вступили в силу с помощью команды `crontab -l`. Результат выполнения представлен на рисунке 1.30.

A screenshot of a terminal window titled "kirill@kirill-HP-ENVY-Notebook: ~". The terminal shows the following commands and output:

```
(base) kirill@kirill-HP-ENVY-Notebook:~$ crontab -l
no crontab for kirill
(base) kirill@kirill-HP-ENVY-Notebook:~$ crontab -e
no crontab for kirill - using an empty one
crontab: installing new crontab
(base) kirill@kirill-HP-ENVY-Notebook:~$ crontab -l
*/10 * * * * scrot
(base) kirill@kirill-HP-ENVY-Notebook:~$
```

Рисунок 1.30 – Создания расписания с помощью `cron`

Вывод

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