

# Липецкий государственный технический университет

Факультет автоматизации и информатики  
Кафедра автоматизированных систем управления

Отчет по лабораторной работе № 1  
по дисциплине «OS Linux»  
на тему «Работа с файловой системой ОС 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
Вывод . . . . .	18
Контрольные вопросы . . . . .	19

Цель работы

Приобрести опыт работы с файлами и каталогами в ОС Linux, настройки прав на доступ к файлам и каталогам.

## 1 Ход выполнения

На компьютер была установлена вторая операционная система Ubuntu 18.04 LTS. На рисунке 1.1 представлен рабочий стол после загрузки ОС.



Рисунок 1.1 – Запуск Linux Ubuntu

На рисунке 1.2 продемонстрирована загрузка пользователя root с помощью команды `sudo su`.

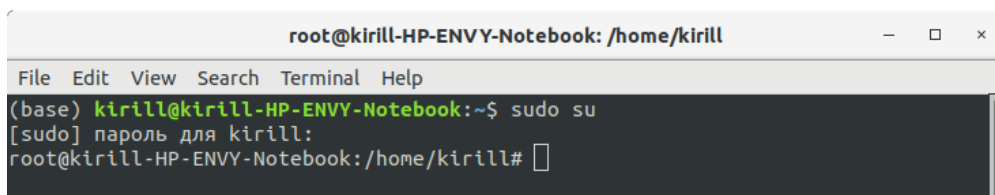


Рисунок 1.2 – Загрузка пользователя root

На рисунке 1.3 показана структура системного каталога. Для вывода данной структуры была использована команда `ls`.

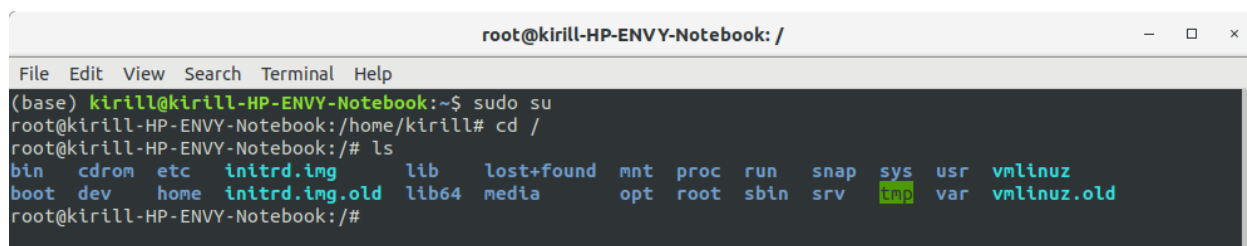


Рисунок 1.3 – Ознакомление со структурой системного каталога

Перечень каталогов, представленных на рисунке 1.3, и их назначение:

1. `/bin` – содержит команды, которые могут использоваться как системным администратором, так и рядовыми пользователями, причем только те команды, которые необходимы, когда никакая другая файловая система еще не смонтирована (например, в одно-пользовательском режиме). В этом каталоге могут также содержаться команды, которые используются не напрямую пользователем, а через скрипты;

2. `/boot` – каталог содержит все, что необходимо в процессе загрузки, исключая конфигурационные файлы и `the map installer`. Таким образом, в `/boot` хранятся данные, которые используются до того, как ядро начинает исполнять программы пользователя. Здесь же находятся резервные сохраненные копии главной загрузочной записи (`master boot sectors`), `sector map files`, и другие данные, которые не подлежат прямому редактированию;

3. `/dev` – это место расположения специальных файлов устройств;

4. `/etc` – содержит конфигурационные файлы и каталоги, специфичные для данной конкретной системы;

5. `/home` – домашняя директория пользователей, это достаточно стандартное решение, очевидно только, что этот каталог является специфичным для каждого отдельного компьютера;

6. `/lib` – содержит те разделяемые библиотеки, которые необходимы для загрузки системы и запуска команд, расположенных в корневой файловой системе, то есть в каталогах `/bin` и `/sbin`;

7. `/lib64` – обычно это используется для поддержки 64-битного или 32-битного формата в системах, поддерживающих несколько форматов исполняемых файлов, и требующих библиотек с одним и тем же названием. В этом случае `/lib32` и `/lib64` могут быть библиотечными каталогами, а `/lib` – символической ссылкой на один из них;

8. `/mnt` – эта директория предназначена для того, чтобы системный администратор мог временно монтировать файловые системы по мере необходимости. Содержимое этого каталога индивидуально для каждой системы и не должно никаким образом влиять на работу запускаемых программ;

9. /opt – зарезервирован для установки дополнительных пакетов программного обеспечения. Пакет, который устанавливается в каталог /opt, должен размещать свои статические файлы в отдельной каталоговой структуре /opt/<package>, где <package> - название соответствующего пакета программного обеспечения;

10. /root – домашний каталог пользователя root;

11. /sbin – утилиты для выполнения задач системного администрирования (и другие команды, используемые только пользователем root) размещаются в /sbin, /usr/sbin и /usr/local/sbin. Каталог /sbin содержит исполняемые файлы, необходимые для загрузки системы и ее восстановления в различных ситуациях (restoring, recovering, and/or repairing the system) и не попавшие в каталог /bin;

12. /tmp – каталог для хранения временных файлов программ. Каталог /tmp должен быть доступен для программ, которым необходимы временные файлы. Программы не должны предполагать, что какой-либо файл в каталоге /tmp сохранится при следующем запуске программы;

13. /media – этот каталог содержит подкаталоги, которые используются в качестве точек монтирования для съемных носителей, таких как гибкие диски, компакт-диски и zip-диски;

14. /run – этот каталог содержит данные системной информации, описывающие систему с момента ее загрузки. Файлы в этом каталоге должны быть очищены (при необходимости удалены или усечены) в начале процесса загрузки;

15. /srv – параметры, которые специфичны для окружения системы. Чаще всего данная директория пуста;

16. /usr – в этом каталоге хранятся все установленные пакеты программ, документация, исходный код ядра и система X Window. Все пользователи кроме суперпользователя root имеют доступ только для чтения. Может быть смонтирована по сети и может быть общей для нескольких машин;

17. /var – это каталог для часто меняющихся данных. Здесь находятся журналы операционной системы, системные log-файлы, cache-файлы и т. д.;

18. `/lost+found` – в `lost+found` скидываются файлы, на которых не было ссылок ни в одной директории, хотя их `inode` не были помечены как свободные;

19. `/proc` – это директория, к которой примонтирована виртуальная файловая система `procfs`. Различная информация, которую ядро может сообщить пользователям, находится в ”файлах” каталога `/proc`;

20. `/sys` – это директория, к которой примонтирована виртуальная файловая система `sysfs`, которая добавляет в пространство пользователя информацию ядра Linux о присутствующих в системе устройствах и драйверах;

21. `/snap` – каталог `/snap` по умолчанию является местом, где файлы и папки из установленных пакетов `snap` появляются в вашей системе.

На рисунке 1.4 показано содержимое каталога физических устройств.

```

root@kirill-HP-ENVY-Notebook: /# ls dev
acpi_thermal_rel  hugepages  loop-control  sda3      tty11      tty31      tty51      ttyS12     ttyS4      vcsa3
autofs           hwrng      mapper        sda4      tty12      tty32      tty52      ttyS13     ttyS5      vcsa4
block           i2c-0      mcelog        sda5      tty13      tty33      tty53      ttyS14     ttyS6      vcsa5
bsg             i2c-1      media0        sda6      tty14      tty34      tty54      ttyS15     ttyS7      vcsa6
btrfs-control    i2c-2      mei0          sda7      tty15      tty35      tty55      ttyS16     ttyS8      vcsu
bus             i2c-3      mem           sdb       tty16      tty36      tty56      ttyS17     ttyS9      vcsu1
char           i2c-4      queue         sdb1      tty17      tty37      tty57      ttyS18     udmabuf    vcsu2
console         i2c-5      net           sdb2      tty18      tty38      tty58      ttyS19     uhid       vcsu3
core           initctl    null          sg0       tty19      tty39      tty59      ttyS20     uinput     vcsu4
cpu            input      nvram         sg1       tty2       tty4       tty6       ttyS21     urandom    vcsu5
cpu_dma_latency kmsg       port          shm       tty20      tty40      tty60      ttyS22     userio     vcsu6
cuse           kvm        ppp           snapshot  tty21      tty41      tty61      ttyS23     v4l        vfio
disk           lightnvme  psaux        snd       tty22      tty42      tty62      ttyS24     vcs        vga_arbiter
dri            log        ptmx         stderr    tty23      tty43      tty63      ttyS25     vcs1       vhost-net
drm_dp_aux0    loop0      pts          stdin     tty24      tty44      tty7       ttyS26     vcs2       vhost-vsock
ecryptfs       loop1      random        tpm0     tty25      tty45      tty8       ttyS27     vcs3       video0
fb0            loop2      rfkill        tpmrm0   tty26      tty46      tty9       ttyS28     vcs4       video1
fd             loop3      rtc           tty       tty27      tty47      tty10      ttyS29     vcs5       zero
freefall       loop4      rtc0          tty0     tty28      tty48      tty11      ttyS30     vcs6       zfs
full           loop5      sda           tty1     tty29      tty49      tty12      ttyS31     vcsa1
fuse           loop6      sda1          tty10    tty3       tty5       tty13      ttyS32     vcsa2
hpet           loop7      sda2          tty11    tty30      tty50      tty14      ttyS33     vcsa3

```

Рисунок 1.4 – Содержимое каталога /dev

Перечень файлов физических устройств, представленных на рисунке 1.4, и их назначение:

1. `acpi_thermal_rel` – обеспечивает функции управления температурой модуля ACPI;
2. `autofs` – цель `autofs` - обеспечить монтирование по требованию и автоматическое размонтирование других файловых систем;
3. `btrfs-control` – устройство принимает некоторые вызовы `ioctl`, которые могут выполнять следующие действия с модулем файловой системы: сканирование устройства на наличие файловой системы `btrfs` (т.е. позволить файловым системам с несколькими устройствами монтировать автоматически) и регистрировать их в модуле ядра, аналогично сканированию, но также дожидается завершения процесса сканирования устройства для данной файловой системы, получение поддерживаемые функции;
4. `console` – текстовый терминал и виртуальные консоли;
5. `cpu_dma_latency` – часть интерфейса качества и обслуживания в ядре Linux
6. `cuse` – символьные устройства в пространстве пользователя



7. `drm_dp_aux<N>` – канал DisplayPort AUX
8. `ecryptfs` – POSIX-совместимая промышленного уровня файловая система многоуровневого шифрования для Linux
9. `fb<N>` – устройство обеспечивает абстракцию для графического оборудования
10. `freefall` – это простой демон, обеспечивающий защиту жесткого диска от ударов для ноутбуков HP, поддерживающий функцию, официально называемую «HP Mobile Data Protection System 3D» или «HP 3D DriveGuard».
11. `fuse` – (filesystem in userspace — «файловая система в пользовательском пространстве») — свободный модуль для ядер Unix-подобных операционных систем, позволяет разработчикам создавать новые типы файловых систем, доступные для монтирования пользователями без привилегий (прежде всего — виртуальных файловых систем);
12. `hpet` – таймер событий высокой точности (HPET) - это аппаратный таймер, используемый в персональных компьютерах.
13. `hwrng` – генератор случайных чисел;
14. `i2c-<N>` – последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов;
15. `kmsg` – узел символьного устройства обеспечивает доступ пользователя к буферу `printk` ядра;
16. `kvm` – виртуальная машина на основе ядра
17. `loop<N>` – в Linux работа с образами дисков осуществляется через так называемые петлевые (`loop`) устройства. Образ привязывается к `loop`-устройству, после этого система может работать с этим устройством, как с обычным блочным;
18. `loop-control` – начиная с Linux 3.1, ядро предоставляет устройство `dev / loop-control`, которое позволяет приложению динамически находить свободное устройство, а также добавлять и удалять устройства `loop` из системы;
19. `mcelog` – серверная часть пользовательского пространства для регистрации ошибок машинных проверок, сообщаемых ядру аппаратными средствами. Ядро выполняет немедленные действия (например, завершает

процессы и т. д.), а mcelog декодирует ошибки и управляет различными другими расширенными ответами на ошибки, такими как отключение памяти, процессоров или запускающих событий. Кроме того, mcelog также обрабатывает исправленные ошибки, регистрируя их;

20. mei<N> – это изолированный и защищенный вычислительный ресурс (сопроцессор), находящийся внутри определенных наборов микросхем Intel;

21. mem – это файл символьного устройства, представляющий собой образ основной памяти компьютера. Его можно использовать, например, для проверки (и даже исправления) системы;

22. null – специальный файл в системах класса UNIX, представляющий собой так называемое «пустое устройство». Запись в него происходит успешно, независимо от объёма «записанной» информации. Чтение из /dev/null эквивалентно считыванию конца файла (EOF);

23. nvram – обеспечивает доступ к конфигурации BIOS NVRAM в системах i386 и amd64;

24. port – символьное устройство для чтения и / или записи;

25. rpp – обеспечивает реализацию функциональных возможностей, которые используются в любой реализации PPP, включая: блок сетевого интерфейса (rpp0 и т. д.), интерфейс к сетевому коду, многоканальный PPP: разделение дейтаграмм между несколькими ссылками, а также упорядочивание и объединение полученных фрагментов, интерфейс к rppd, через символьное устройство / dev / rpp, сжатие и распаковка пакетов, сжатие и распаковка заголовков TCP / IP, обнаружение сетевого трафика для набора по требованию и для тайм-аутов простоя, простая фильтрация пакетов;

26. psaux – устройство мыши PS / 2;

27. ptmx – используется для создания пары псевдотерминалов ведущего и ведомого;

28. random – предоставляет интерфейс к системному генератору случайных чисел, который выводит шум из драйверов устройств и других источников в «хаотичный» пул;

29. rfkill – предоставляет общий интерфейс для отключения любого радиопередатчика в системе;
30. rtc<N> – часы реального времени;
31. sda – первый жесткий диск;
32. sda<N> – N-ый раздел первого жесткого диска;
33. sdb – второй жесткий диск;
34. sdb<N> – N-ый раздел второго жесткого диска;
35. sg<N> – SCSI Generic driver используется, среди прочего, для сканеров, устройств записи компакт-дисков и чтения аудио-компакт-дисков в цифровом формате;
36. snapshot – поддержка снимков устройства;
37. tpm<N> – разрешает доступ к устройству Trusted Platform Module (tpm);
38. tty<N> – виртуальная консоль;
39. ttyprintk – драйвер псевдо ТТУ, который позволяет пользователям создавать сообщения printk через вывод на устройство ttyprintk;
40. uhid – поддержка драйвера ввода-вывода пользовательского пространства для подсистемы HID;
41. uinput – поддержка драйвера уровня пользователя для ввода;
42. urandom – более быстрая и менее безопасная генерация случайных чисел;
43. userio – призван упростить жизнь разработчикам драйверов ввода, позволяя им тестировать различные устройства Serio (в основном, различные сенсорные панели на ноутбуках), не имея физического устройства перед ними;
44. vcs<N> – текущее текстовое содержимое виртуальной консоли;
45. vcsa<N> – текущее содержимое текстового атрибута виртуальной консоли;
46. vcsu<N> – текущее текстовое содержимое виртуальной консоли(юникод);

47. `vga_arbiter` – сканирует все устройства PCI и добавляет в арбитраж VGA. Затем арбитр включает / отключает декодирование на разных устройствах устаревших инструкций VGA;

48. `vhci` – виртуальный драйвер HCI Bluetooth;

49. `vhost-net` – ускоритель ядра хоста для virtio net;

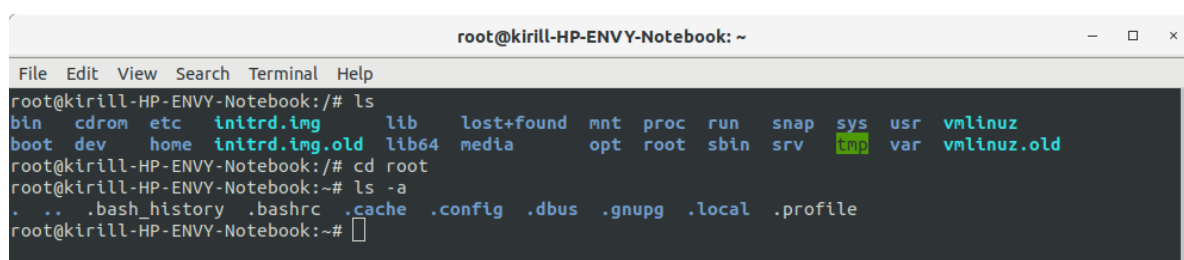
50. `vhost-vsock` – программное устройство, поэтому нет пробного вызова, который вызывает драйвер, чтобы зарегистрировать его узел устройства `misc char`. Это создает проблема с курицей и яйцом: приложения в пользовательском пространстве должны открываться/ `dev / vhost-vsock`, чтобы использовать драйвер, но файл не существует, пока модуль ядра загружен;

51. `video<N>` – устройство видеозахвата / наложения;

52. `zero` – источник нулевого байта;

53. `zfs` – настраивает пулы хранения ZFS.

На рисунке 1.5 представлен переход в директорию пользователя `root` с помощью команды `cd root` и выведено ее содержимое - команда `ls -a`.



```
root@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook:/# ls  
bin  cdrom  etc  initrd.img  lib  lost+found  mnt  proc  run  snap  sys  usr  vmlinuz  
boot dev  home  initrd.img.old  lib64  media  opt  root  sbin  srv  tmp  var  vmlinuz.old  
root@kirill-HP-ENVY-Notebook:/# cd root  
root@kirill-HP-ENVY-Notebook:~# ls -a  
.  ..  .bash_history  .bashrc  .cache  .config  .dbus  .gnupg  .local  .profile  
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.5 – Содержимое директории пользователя `root`

Затем переходим в корневой каталог с помощью команды `cd /` и откроем файл `vmlinuz` с помощью текстового редактора `vim`, выполнив команду `vim vmlinuz` (рисунок 1.6).

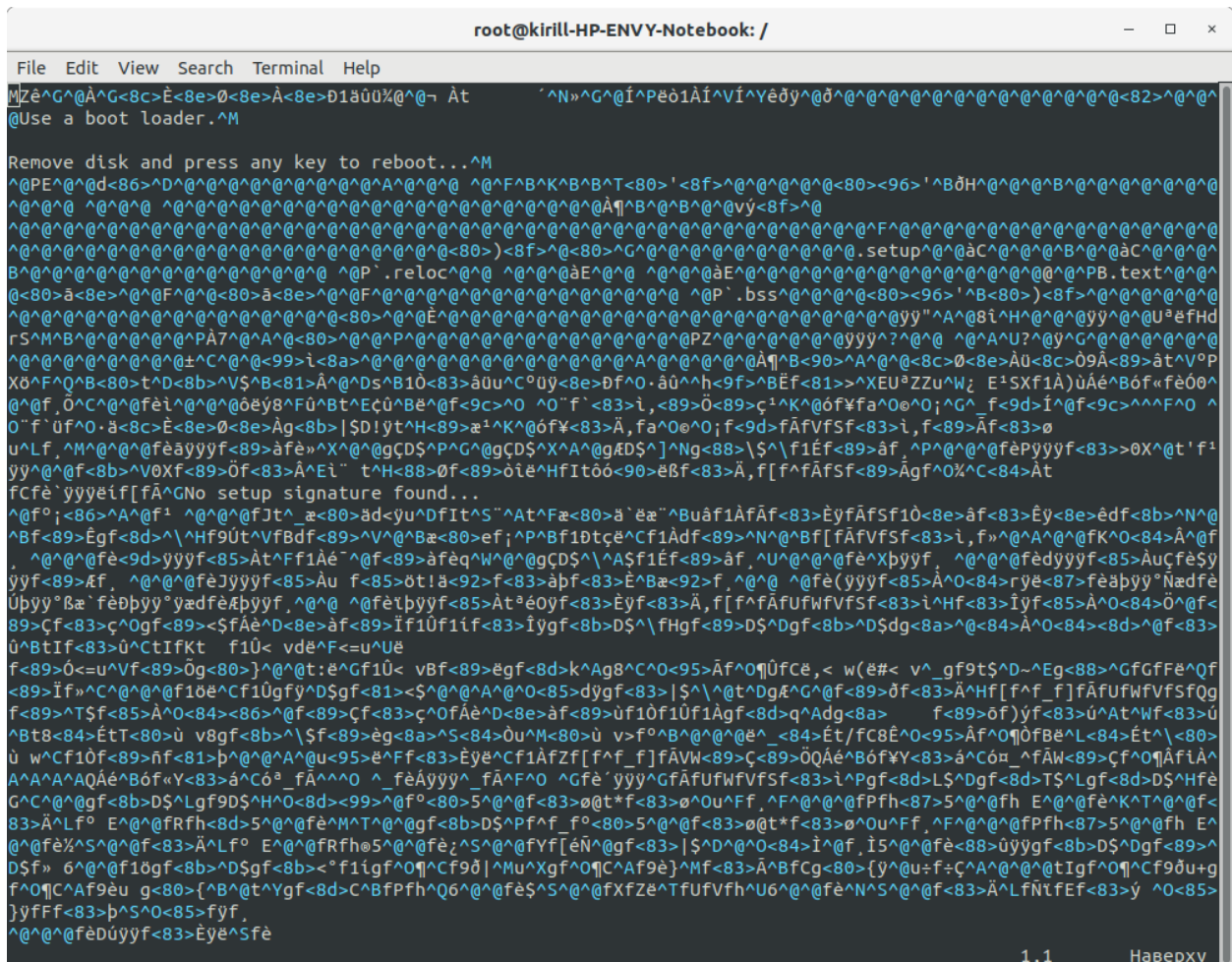
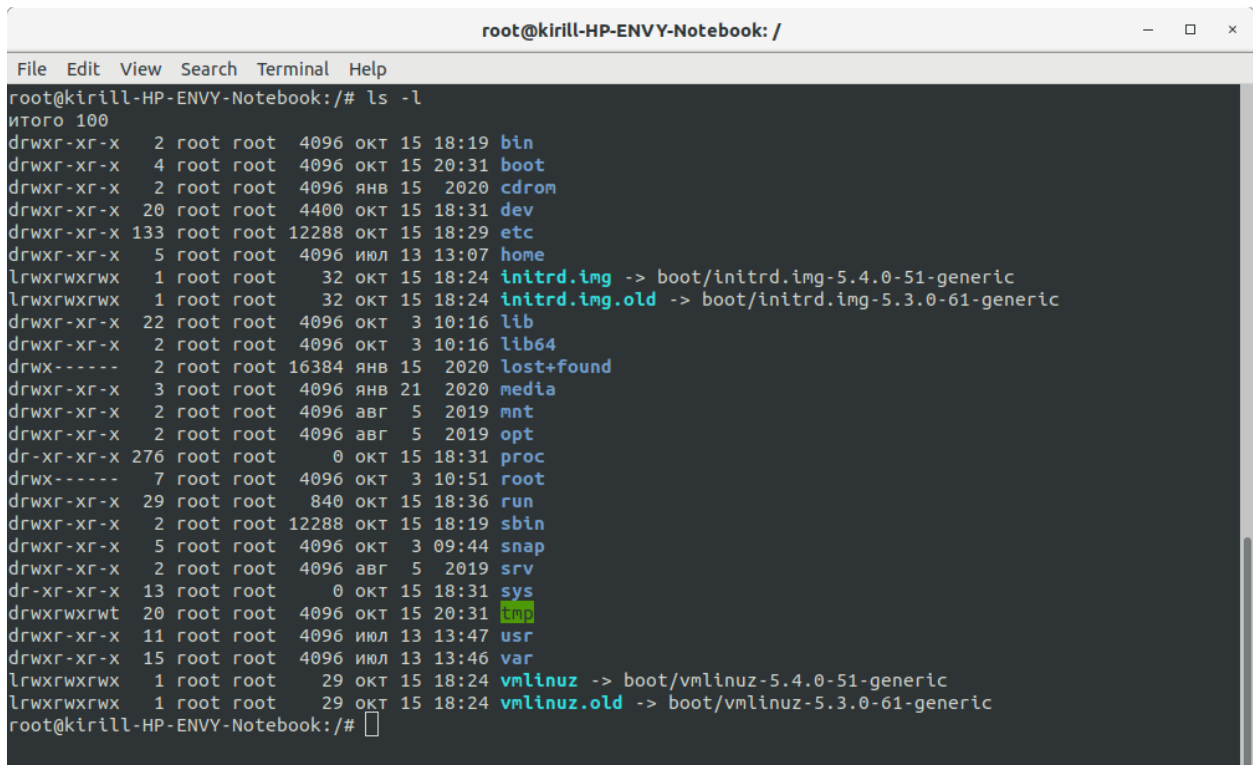


Рисунок 1.6 – Содержимое файла `vmlinuz`

Чтобы просмотреть права доступа к файлу `vmlinuz`, выполним команду `ls -l`, результат выполнения показан на рисунке 1.7.

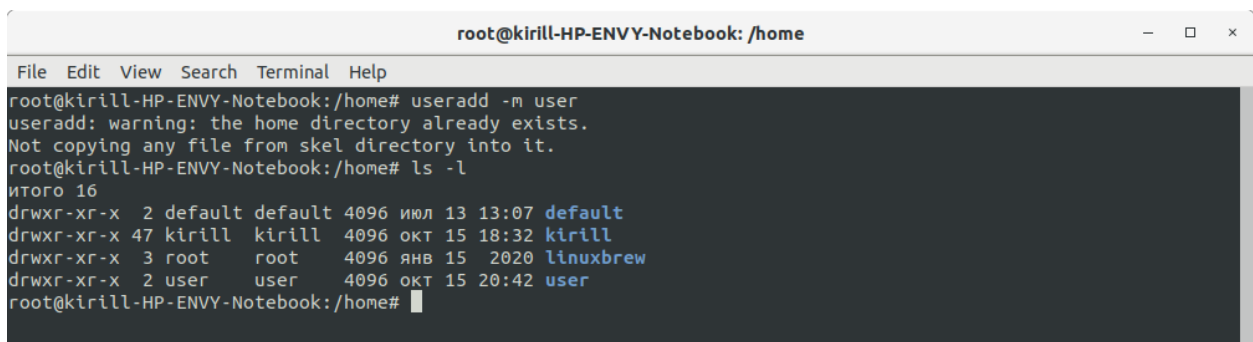


```
root@kirill-HP-ENVY-Notebook: /  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook: /# ls -l  
итого 100  
drwxr-xr-x  2 root root  4096 окт 15 18:19 bin  
drwxr-xr-x  4 root root  4096 окт 15 20:31 boot  
drwxr-xr-x  2 root root  4096 янв 15  2020 cdrom  
drwxr-xr-x 20 root root 4400 окт 15 18:31 dev  
drwxr-xr-x 133 root root 12288 окт 15 18:29 etc  
drwxr-xr-x  5 root root  4096 июл 13 13:07 home  
lrwxrwxrwx  1 root root   32 окт 15 18:24 initrd.img -> boot/initrd.img-5.4.0-51-generic  
lrwxrwxrwx  1 root root   32 окт 15 18:24 initrd.img.old -> boot/initrd.img-5.3.0-61-generic  
drwxr-xr-x 22 root root  4096 окт  3 10:16 lib  
drwxr-xr-x  2 root root  4096 окт  3 10:16 lib64  
drwx----- 2 root root 16384 янв 15  2020 lost+found  
drwxr-xr-x  3 root root  4096 янв 21  2020 media  
drwxr-xr-x  2 root root  4096 авг  5  2019 mnt  
drwxr-xr-x  2 root root  4096 авг  5  2019 opt  
dr-xr-xr-x 276 root root    0 окт 15 18:31 proc  
drwx----- 7 root root  4096 окт  3 10:51 root  
drwxr-xr-x 29 root root   840 окт 15 18:36 run  
drwxr-xr-x  2 root root 12288 окт 15 18:19 sbin  
drwxr-xr-x  5 root root  4096 окт  3 09:44 snap  
drwxr-xr-x  2 root root  4096 авг  5  2019 srv  
dr-xr-xr-x 13 root root    0 окт 15 18:31 sys  
drwxrwxrwt 20 root root  4096 окт 15 20:31 tmp  
drwxr-xr-x 11 root root  4096 июл 13 13:47 usr  
drwxr-xr-x 15 root root  4096 июл 13 13:46 var  
lrwxrwxrwx  1 root root   29 окт 15 18:24 vmlinuz -> boot/vmlinuz-5.4.0-51-generic  
lrwxrwxrwx  1 root root   29 окт 15 18:24 vmlinuz.old -> boot/vmlinuz-5.3.0-61-generic  
root@kirill-HP-ENVY-Notebook: /#
```

Рисунок 1.7 – Содержимое корневого каталога

Как видно из рисунка 1.7 все пользователи и группы пользователей имеют полные права на файл `vmlinuz`. Владелцем файла указан пользователь `root`.

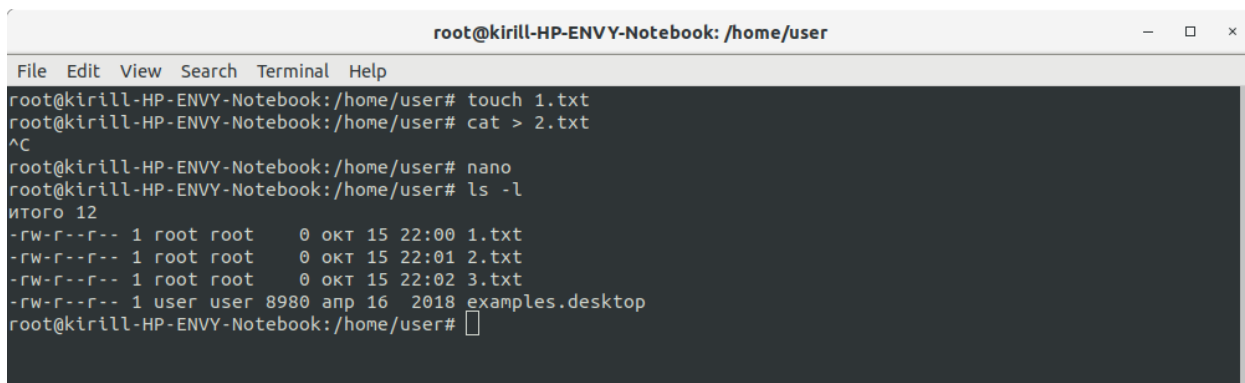
Далее добавим нового пользователя `user` с помощью команды `adduser -m user`. Результат выполнения команды представлен на рисунке 1.8.



```
root@kirill-HP-ENVY-Notebook: /home  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook: /home# useradd -m user  
useradd: warning: the home directory already exists.  
Not copying any file from skel directory into it.  
root@kirill-HP-ENVY-Notebook: /home# ls -l  
итого 16  
drwxr-xr-x  2 default default 4096 июл 13 13:07 default  
drwxr-xr-x 47 kirill kirill 4096 окт 15 18:32 kirill  
drwxr-xr-x  3 root root  4096 янв 15  2020 linuxbrew  
drwxr-xr-x  2 user user  4096 окт 15 20:42 user  
root@kirill-HP-ENVY-Notebook: /home#
```

Рисунок 1.8 – Добавление нового пользователя

Создадим 3 файла 1.txt, 2.txt, 3.txt с помощью команд touch, cat и nano. Процесс создания представлен на рисунке 1.9.

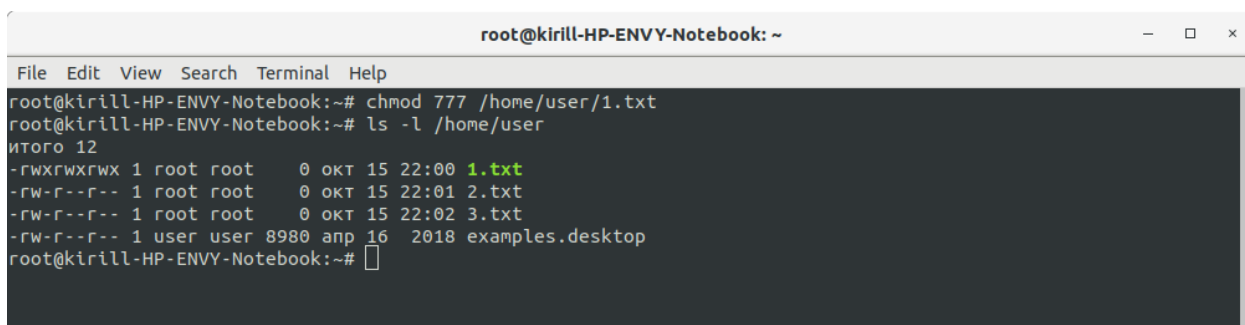


```
root@kirill-HP-ENVY-Notebook: /home/user
File Edit View Search Terminal Help
root@kirill-HP-ENVY-Notebook:/home/user# touch 1.txt
root@kirill-HP-ENVY-Notebook:/home/user# cat > 2.txt
^C
root@kirill-HP-ENVY-Notebook:/home/user# nano
root@kirill-HP-ENVY-Notebook:/home/user# ls -l
итого 12
-rw-r--r-- 1 root root 0 окт 15 22:00 1.txt
-rw-r--r-- 1 root root 0 окт 15 22:01 2.txt
-rw-r--r-- 1 root root 0 окт 15 22:02 3.txt
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop
root@kirill-HP-ENVY-Notebook:/home/user#
```

Рисунок 1.9 – Создание новых файлов различными способами

Как видно из рисунка 1.9 владельцем файлов является пользователь root, он имеет полные права на файлы, остальные пользователи имеют права только право на чтение.

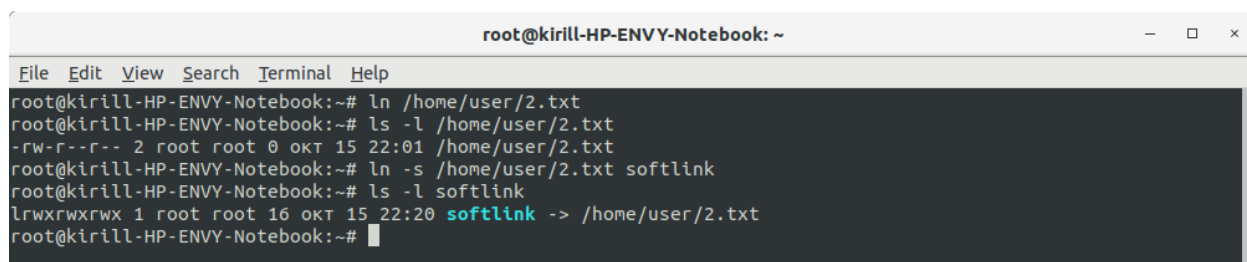
Изменим права доступа к файлу 1.txt с помощью команды chmod 777, таким образом все пользователи и группы пользователей получат полные права на файлы 1.txt. Результат выполнения команды представлен на рисунке 1.10.



```
root@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
root@kirill-HP-ENVY-Notebook:~# chmod 777 /home/user/1.txt
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user
итого 12
-rwxrwxrwx 1 root root 0 окт 15 22:00 1.txt
-rw-r--r-- 1 root root 0 окт 15 22:01 2.txt
-rw-r--r-- 1 root root 0 окт 15 22:02 3.txt
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.10 – Изменения прав на файл 1.txt

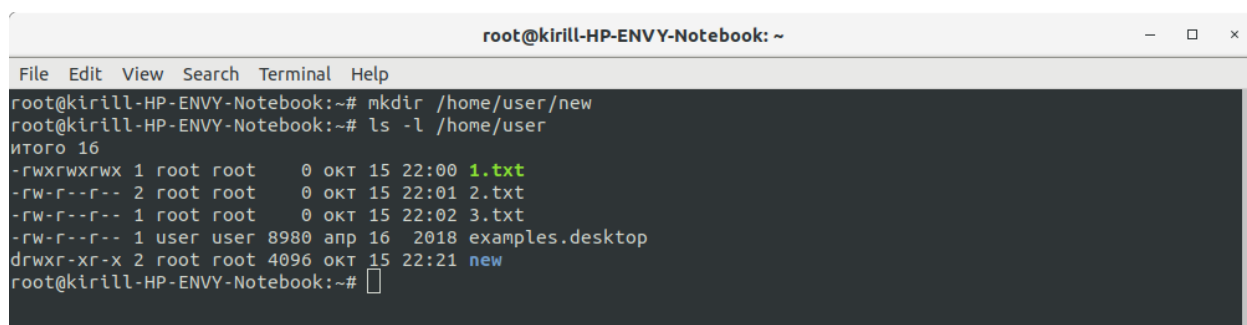
Создадим жесткую и символические ссылки с помощью команд `ln` и `ln -s` соответственно. Результат выполненных команд представлен на рисунке 1.11.



```
root@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
root@kirill-HP-ENVY-Notebook:~# ln /home/user/2.txt
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user/2.txt
-rw-r--r-- 2 root root 0 окт 15 22:01 /home/user/2.txt
root@kirill-HP-ENVY-Notebook:~# ln -s /home/user/2.txt softlink
root@kirill-HP-ENVY-Notebook:~# ls -l softlink
lrwxrwxrwx 1 root root 16 окт 15 22:20 softlink -> /home/user/2.txt
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.11 – Создание жесткой и символической ссылки на файл 2.txt

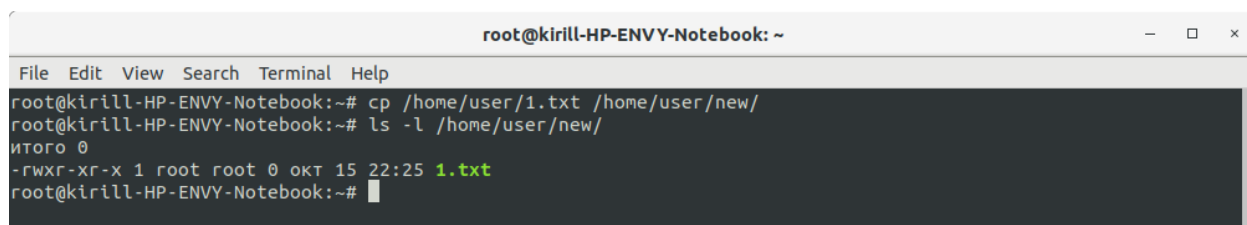
Создадим новую директорию с помощью команды `mkdir`, результат представлен на рисунке 1.12.



```
root@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
root@kirill-HP-ENVY-Notebook:~# mkdir /home/user/new
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user
итого 16
-rwxrwxrwx 1 root root 0 окт 15 22:00 1.txt
-rw-r--r-- 2 root root 0 окт 15 22:01 2.txt
-rw-r--r-- 1 root root 0 окт 15 22:02 3.txt
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop
drwxr-xr-x 2 root root 4096 окт 15 22:21 new
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.12 – Создание новой директории

Для копирования файла 1.txt воспользуемся командой `cp` (рисунок 1.13).

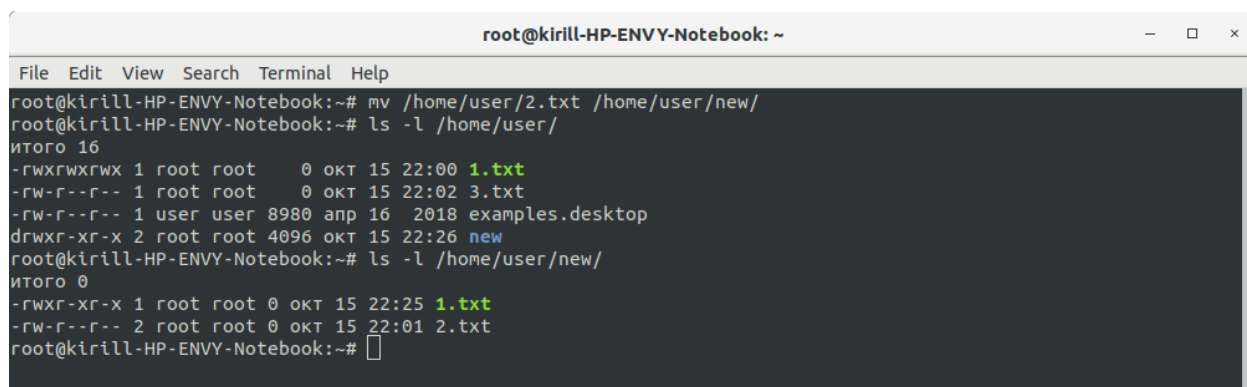


```
root@kirill-HP-ENVY-Notebook: ~
File Edit View Search Terminal Help
root@kirill-HP-ENVY-Notebook:~# cp /home/user/1.txt /home/user/new/
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user/new/
итого 0
-rwxr-xr-x 1 root root 0 окт 15 22:25 1.txt
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.13 – Копирование файла 1.txt в директорию /new



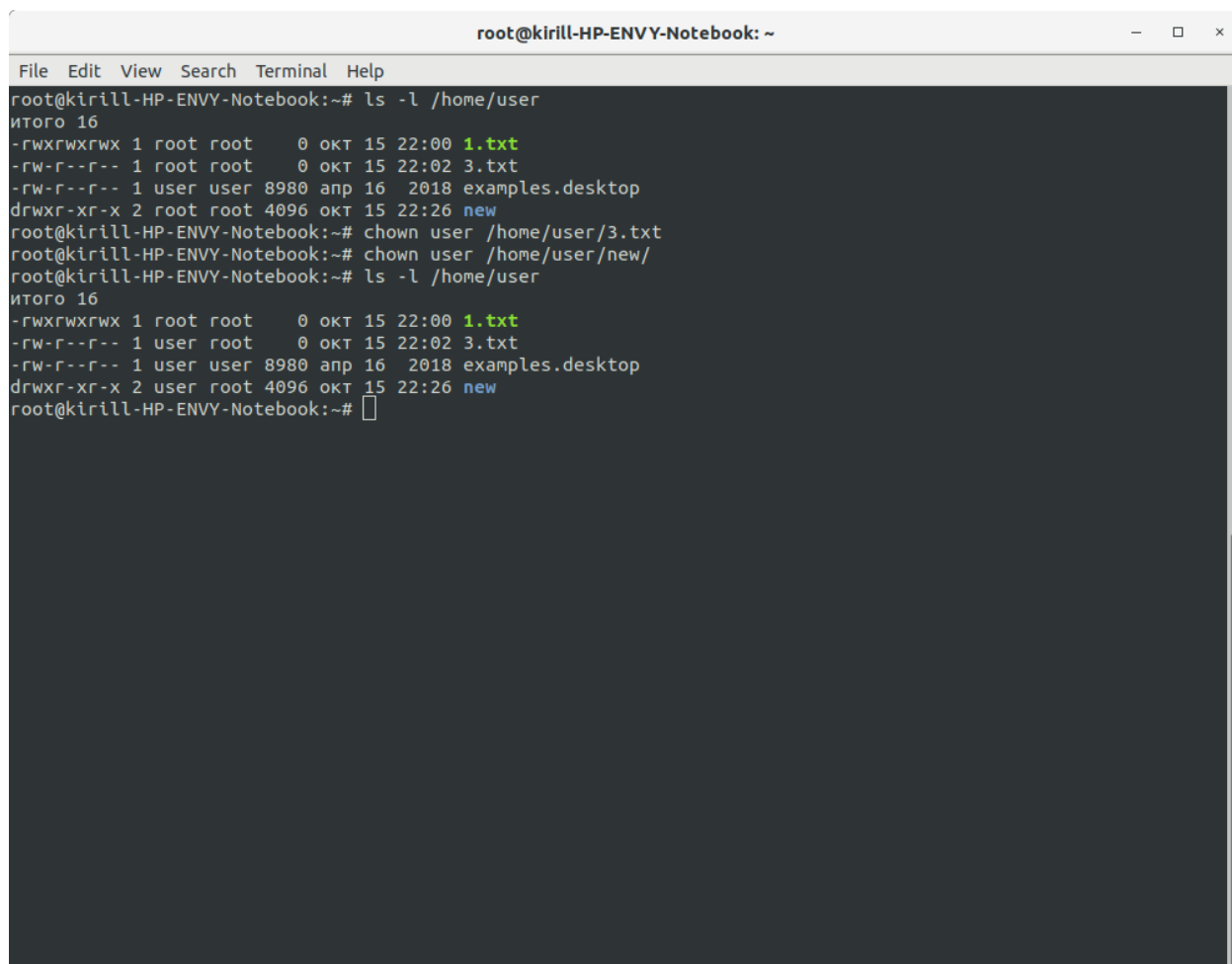
Перемещение файла 2.txt представлено на рисунке 1.14. Чтобы переместить файл была использована команда mv.



```
root@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook:~# mv /home/user/2.txt /home/user/new/  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user/  
итого 16  
-rwxrwxrwx 1 root root 0 окт 15 22:00 1.txt  
-rw-r--r-- 1 root root 0 окт 15 22:02 3.txt  
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop  
drwxr-xr-x 2 root root 4096 окт 15 22:26 new  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user/new/  
итого 0  
-rwxr-xr-x 1 root root 0 окт 15 22:25 1.txt  
-rw-r--r-- 2 root root 0 окт 15 22:01 2.txt  
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.14 – Перемещение файла 2.txt в директорию /new

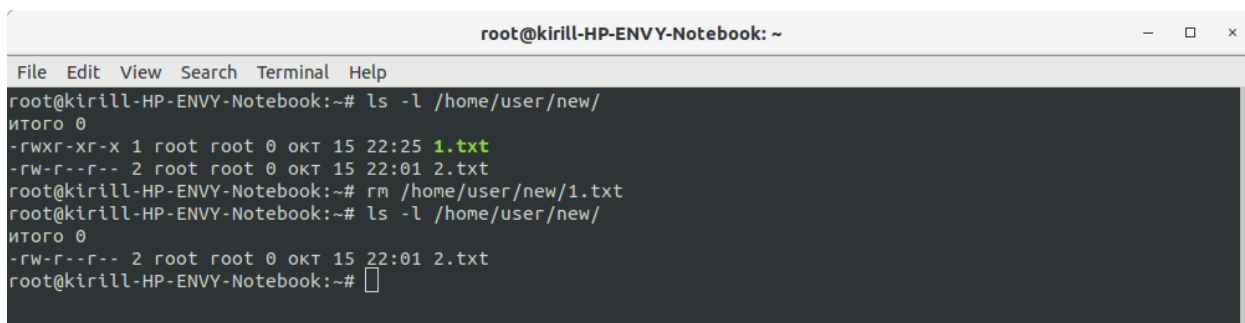
Изменим владельца файла 3.txt на user с помощью команды chown. Результат выполнения команды представлен на рисунке 1.15.



```
root@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user  
итого 16  
-rwxrwxrwx 1 root root 0 окт 15 22:00 1.txt  
-rw-r--r-- 1 root root 0 окт 15 22:02 3.txt  
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop  
drwxr-xr-x 2 root root 4096 окт 15 22:26 new  
root@kirill-HP-ENVY-Notebook:~# chown user /home/user/3.txt  
root@kirill-HP-ENVY-Notebook:~# chown user /home/user/new/  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user  
итого 16  
-rwxrwxrwx 1 root root 0 окт 15 22:00 1.txt  
-rw-r--r-- 1 user root 0 окт 15 22:02 3.txt  
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop  
drwxr-xr-x 2 user root 4096 окт 15 22:26 new  
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.15 – Изменение владельца файла 3.txt и директории /new

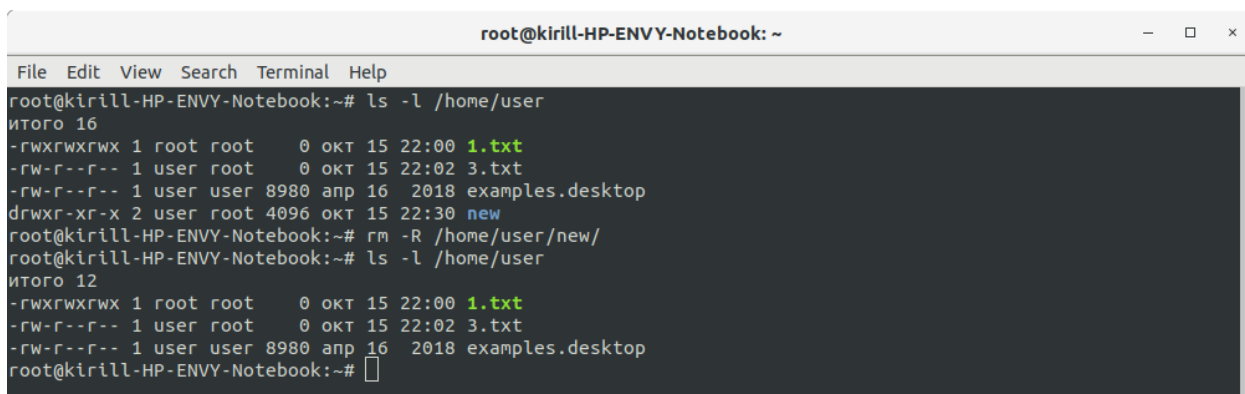
На рисунке 1.16 представлена процедура удаление файла 1.txt с помощью команды `rm`.



```
root@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user/new/  
итого 0  
-rwxr-xr-x 1 root root 0 окт 15 22:25 1.txt  
-rw-r--r-- 2 root root 0 окт 15 22:01 2.txt  
root@kirill-HP-ENVY-Notebook:~# rm /home/user/new/1.txt  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user/new/  
итого 0  
-rw-r--r-- 2 root root 0 окт 15 22:01 2.txt  
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.16 – Удаление файла 1.txt

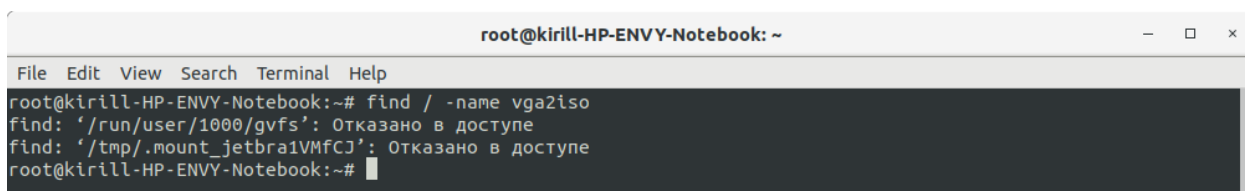
Для того, чтобы удалить непустую директорию необходимо воспользоваться командой `rm` с опцией `-R`, тогда удалиться директория и все ее содержимое. Результат выполнения представлен на рисунке 1.17.



```
root@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user  
итого 16  
-rwxrwxrwx 1 root root 0 окт 15 22:00 1.txt  
-rw-r--r-- 1 user root 0 окт 15 22:02 3.txt  
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop  
drwxr-xr-x 2 user root 4096 окт 15 22:30 new  
root@kirill-HP-ENVY-Notebook:~# rm -R /home/user/new/  
root@kirill-HP-ENVY-Notebook:~# ls -l /home/user  
итого 12  
-rwxrwxrwx 1 root root 0 окт 15 22:00 1.txt  
-rw-r--r-- 1 user root 0 окт 15 22:02 3.txt  
-rw-r--r-- 1 user user 8980 апр 16 2018 examples.desktop  
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.17 – Удаление директории /new

Чтобы найти файл по имени необходимо воспользоваться командой `find` и после опции `-name` указать имя искомого файла `vga2iso`. Результат поиска представлен на рисунке 1.18.



```
root@kirill-HP-ENVY-Notebook: ~  
File Edit View Search Terminal Help  
root@kirill-HP-ENVY-Notebook:~# find / -name vga2iso  
find: '/run/user/1000/gvfs': Отказано в доступе  
find: '/tmp/.mount_jetbra1VMfCJ': Отказано в доступе  
root@kirill-HP-ENVY-Notebook:~#
```

Рисунок 1.18 – Выполнение поиска файла vga2iso

## Вывод

В ходе выполнения лабораторной работы я приобрел опыт работы с файлами и каталогами в ОС Linux, научился настраивать права на доступ к файлам и каталогам.

## Контрольные вопросы

### 1. Что такое файловая система?

Файловая система - это структура, с помощью которой ядро операционной системы предоставляет пользователям (и процессам) ресурсы долговременной памяти системы, т. е. памяти на различного вида долговременных носителях информации - жестких дисках, магнитных лентах, CD-ROM и т. п. С точки зрения пользователя, файловая система — это логическая структура каталогов и файлов.

### 2. Права доступа к файлам. Назначение прав доступа.

В основе механизмов разграничения доступа лежат имена пользователей и имена групп пользователей. В ОС Linux пользователи и группы пользователей могут иметь следующие права на доступ к файлу: чтение из файла, запись в файл и запуск файла на исполнение. Чтобы узнать какие права доступа назначены какому-либо файлу необходимо выполнить команду `ls -l <имя файла>`. Данная команда выведет следующую информацию о файле: тип файла (файл, каталог, символическая ссылка), информация о правах доступа к файлу владельца, системной группы, для всех остальных пользователей. Чтобы задать новые права для доступа к файлу необходимо воспользоваться командой `chmod`, которую можно использовать в двух форматах: явное указание кому какие права доступа назначить или отнять и цифровое кодирование прав доступа.

### 3. Жесткая ссылка в Linux. Основные сведения.

Жёсткая ссылка (HardLink) - это просто ещё одна запись в папке для данного файла.

Когда создаётся жёсткая ссылка, сам файл не копируется физически, а только появляется под ещё одним именем или в ещё одном месте, а его старые имя и местонахождение остаются нетронутыми. С этого момента жёсткая ссылка неотличима от первоначальной записи в папке. Единственное отличие - то, что для жёсткой ссылки не создается короткое имя файла, поэтому из ДОС-программ она не видна.

Когда меняется размер или дата файла, все соответствующие записи в папках обновляются автоматически. При удалении файла он не удаляется физически до тех пор, пока все жёсткие ссылки, указывающие на него, не будут удалены. Порядок их удаления значения не имеет. При удалении жёсткой ссылки в корзину количество ссылок у файла сохраняется.

#### 4. Команда поиска в Linux. Основные сведения.

Find - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Утилита `find` предустановлена по умолчанию во всех Linux дистрибутивах, поэтому вам не нужно будет устанавливать никаких дополнительных пакетов. Это очень важная находка для тех, кто хочет использовать командную строку наиболее эффективно.

Команда `find` имеет такой синтаксис:

`find [папка] [параметры] критерий шаблон [действие]`

#### 5. Перечислите основные команды работы с каталогами.

Основными командами для работы с файлами и каталогами в ОС Linux являются:

1. `ls` – список файлов в директории;
2. `cd` – переход между директориями;
3. `rm` – удалить файл;
4. `rmdir` – удалить папку;
5. `mv` – переместить файл;
6. `cp` – скопировать файл;
7. `mkdir` – создать папку;
8. `ln` – создать ссылку;
9. `chmod` – изменить права файла;
10. `touch` – создать пустой файл.