

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра автоматизированных систем управления**

**Лабораторная работа №1**

**по Дисциплине «Операционная система Linux»**

**Работа с файловой системой ОС Linux**

Студент

Глебов Д.А.

Группа АИ-18

Руководитель

Кургасов В.В.

к.п.н.

Липецк 2020 г.

## Оглавление

Цель работы .....	3
Задание кафедры.....	4
Ход работы.....	5
Вывод.....	26
Ответы на контрольные вопросы .....	27

## Цель работы

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

### Задание кафедры

1. Запустить виртуальную машину Linux Ubuntu.
2. Загрузиться пользователем root (sudo su).
3. Ознакомиться со структурой системных каталогов ОС Linux на рабочем месте. Изучить стандарт (2.1. Filesystem Hierarchy Standard).
4. Привести в отчете перечень каталогов с указанием их назначения.
5. Просмотреть содержимое каталога файлов физических устройств. В отчете привести перечень файлов физических устройств на рабочем месте с указанием назначения файлов.
6. Перейти в директорию пользователя root. Просмотреть содержимое каталога. Просмотреть содержимое файла vmlinuz. Просмотреть и пояснить права доступа к файлу vmlinuz.
7. Создать нового пользователя user.
8. Создать в директории пользователя user три файла 1.txt, 2.txt и 3.txt, используя команды touch, cat и текстовый редактор (на выбор vi/nano). Просмотреть и пояснить права доступа к файлам.
9. Перейти в директории пользователя root. В отчете описать результат.
10. Изменить права доступа на файл 1.txt в директории пользователя user.
11. Создать жесткую и символическую ссылки на файл 2.txt. Просмотреть результаты.
12. Создать каталог new в каталоге пользователя user.
13. Скопировать файл 1.txt в каталог new.
14. Переместить файл 2.txt в каталог new.
15. Изменить владельца файла 3.txt и каталога new.
16. Удалить файл 1.txt в каталоге new.
17. Удалить каталог new.
18. Найти, используя команду find, файл vga2iso (или другой файл по заданию преподавателя).

## Ход работы

В связи с тем, что на компьютере уже была установлена Linux Ubuntu 20.04 второй системой, было принято решение выполнить работу без установки виртуальной машины и выполнить работу в Ubuntu 20.04.

Начнём работу с того, что загрузимся пользователем root с помощью команды `sudo su`:

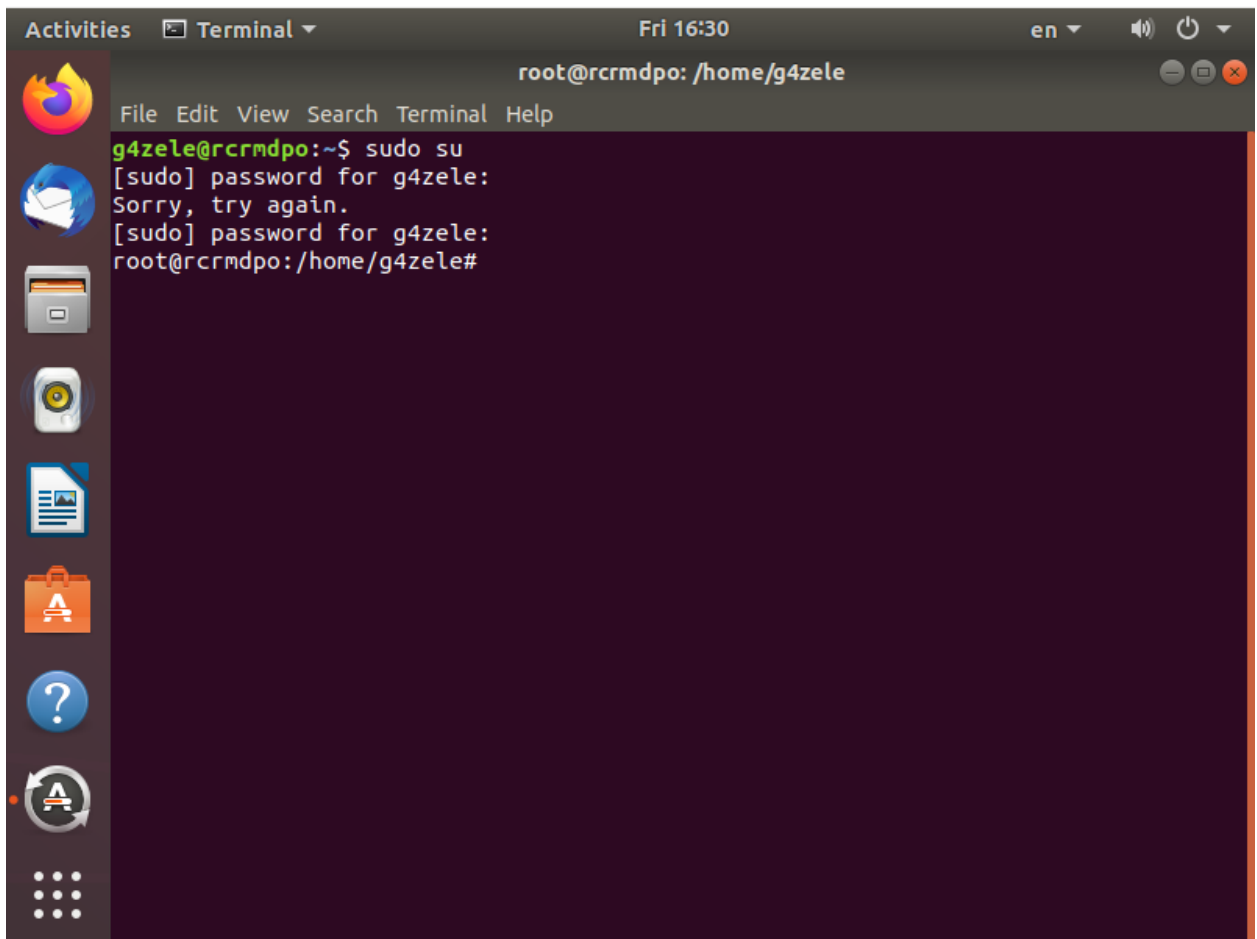


Рисунок 1 – Загрузка пользователем root

Все каталоги хранятся в корневой директории. Просмотрим её содержание. Для этого используем команду `ls`:

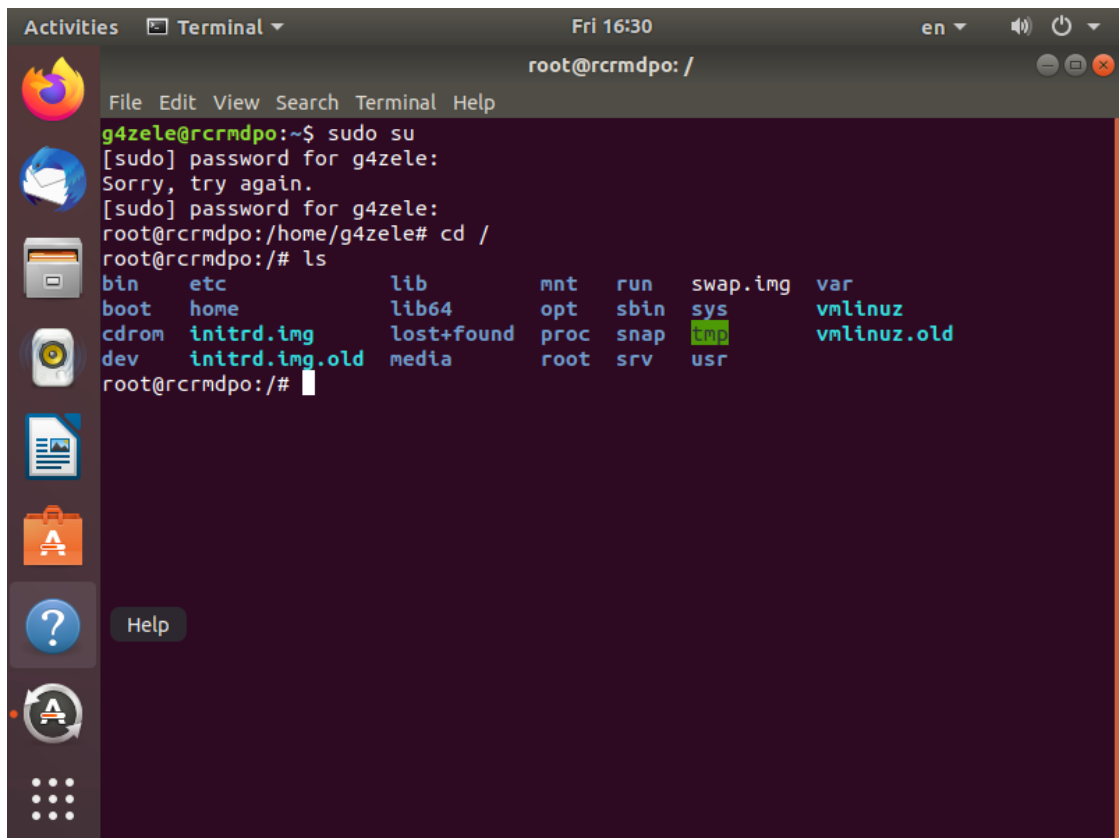


Рисунок 2 – Корневой каталог

Опишем каждый из этих каталогов:

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 однопользовательском 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 7 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/, где - 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 8 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 inod не были помечены как свободные;

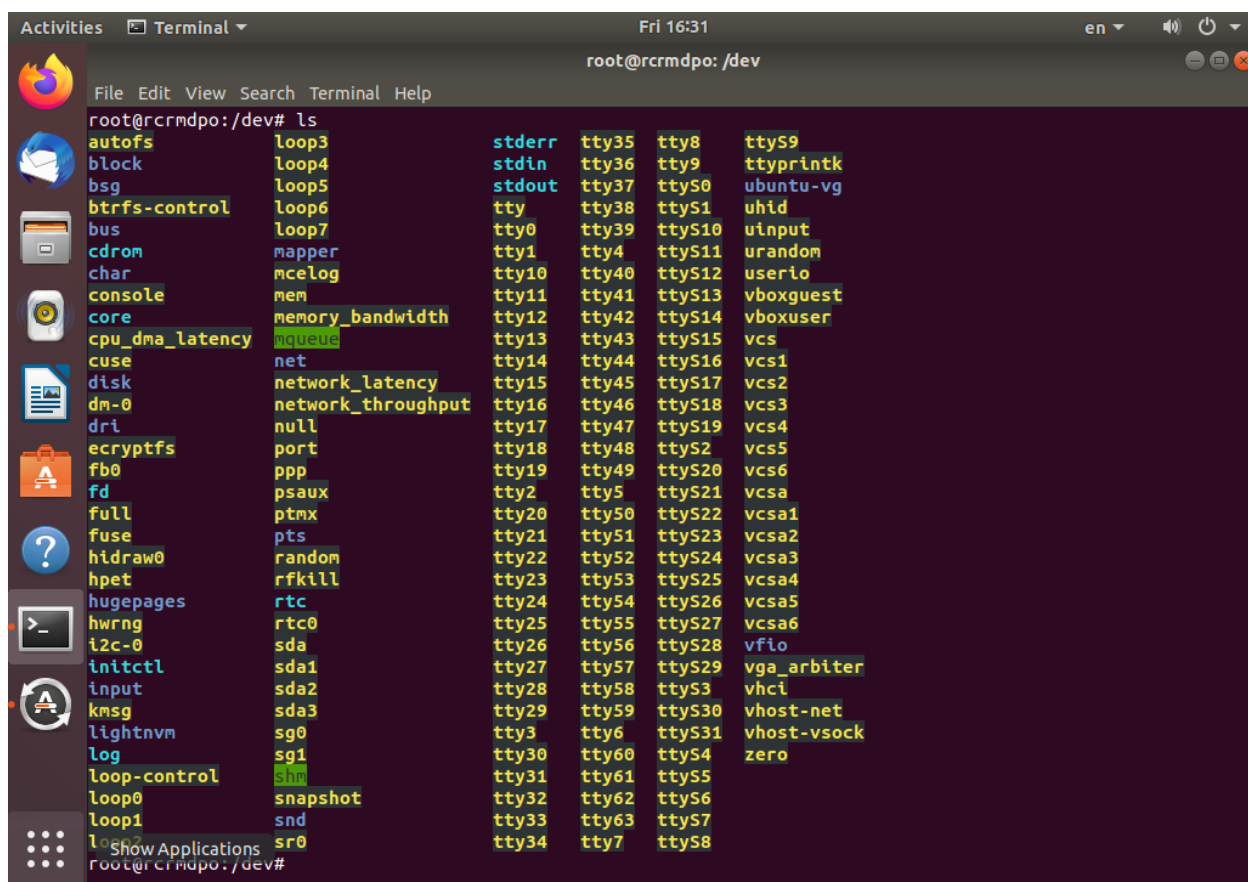
19. /proc – это директория, к которой примонтирована виртуальная файловая система procfs. Различная информация, которую ядро может сообщить пользователям, находится в “файлах” каталога /proc;

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.



Далее перейдём в директорию /dev, которая является каталогом файлов физических устройств, и посмотрим её:



```
root@rcrmdpo:/dev# ls
autofs          loop3          stderr         tty35          tty8           ttyS9
block           loop4          stdin          tty36          tty9           ttyprintk
bsg             loop5          stdout         tty37          ttyS0          ubuntu-vg
btrfs-control  loop6          tty            tty38          ttyS1          uhid
bus             loop7          tty0           tty39          ttyS10         uinput
cdrom           mapper         tty1           tty4           ttyS11         urandom
char            mcelog         tty10          tty40          ttyS12         userio
console         mem            tty11          tty41          ttyS13         vboxguest
core            memory_bandwidth tty12          tty42          ttyS14         vboxuser
cpu_dma_latency queue          tty13          tty43          ttyS15         vcs
cuse            net            tty14          tty44          ttyS16         vcs1
disk            network_latency tty15          tty45          ttyS17         vcs2
dm-0            network_throughput tty16          tty46          ttyS18         vcs3
dri             null           tty17          tty47          ttyS19         vcs4
ecryptfs        port           tty18          tty48          ttyS20         vcs5
fb0             ppp            tty19          tty49          ttyS21         vcs6
fd              psaux          tty2           tty5           ttyS22         vcsa
full            ptmx           tty20          tty50          ttyS23         vcsa1
fuse            pts            tty21          tty51          ttyS24         vcsa2
hidraw0         random          tty22          tty52          ttyS25         vcsa3
hpet            rfkill         tty23          tty53          ttyS26         vcsa4
hugepages       rtc            tty24          tty54          ttyS27         vcsa5
hwrng           rtc0           tty25          tty55          ttyS28         vcsa6
i2c-0           sda            tty26          tty56          ttyS29         vfio
initctl         sda1           tty27          tty57          ttyS30         vga_arbiter
input           sda2           tty28          tty58          ttyS31         vhci
kmsg            sda3           tty29          tty59          ttyS32         vhost-net
lightnvm        sg0            tty3            tty6           ttyS33         vhost-vsock
log             sg1            tty30          tty60          ttyS34         zero
loop-control    shw            tty31          tty61          ttyS35
loop0           snapshot       tty32          tty62          ttyS36
loop1           snd            tty33          tty63          ttyS37
loop2           sr0            tty34          tty64          ttyS38
```

Рисунок 3 – Каталог файлов физических устройств

Опишем эти файлы:

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` – устройства 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 дождаться 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` – канал DisplayPort AUX;
8. `ecryptfs` – POSIX-совместимая промышленного уровня файловая система многоуровневого шифрования для Linux;
9. `fb` – 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 Unixподобных 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-` – 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` – в Linux работа с образами дисков осуществляется через так называемые петлевые (`loop`) устройства. Образ привязывается к `loop-` 11 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 – 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 объема «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 (ppp0 и т. д.), интерфейс к сетевому коду, многоканальный PPP: разделение дейтаграмм между несколькими ссылками, а также упорядочивание и объединение полученных фрагментов, интерфейс к rppd, 12 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. `rftkill` – предоставляет общий интерфейс для отключения любого радиопередатчика в системе;

30. `rtc` – часы реального времени;

31. `sda` – первый жесткий диск;

32. `sda` – N-ый раздел первого жесткого диска;

33. `sdb` – второй жесткий диск;

34. `sdb` – N-ый раздел второго жесткого диска;

35. `sg` – 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. `tmp` – разрешает доступ к устройству Trusted Platform Module (tpm);

38. `tty` – 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; 13

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` – текущее текстовое содержимое виртуальной консоли;

45. `vcsa` – текущее содержимое текстового атрибута виртуальной консоли;

46. `vcsu` – 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`;

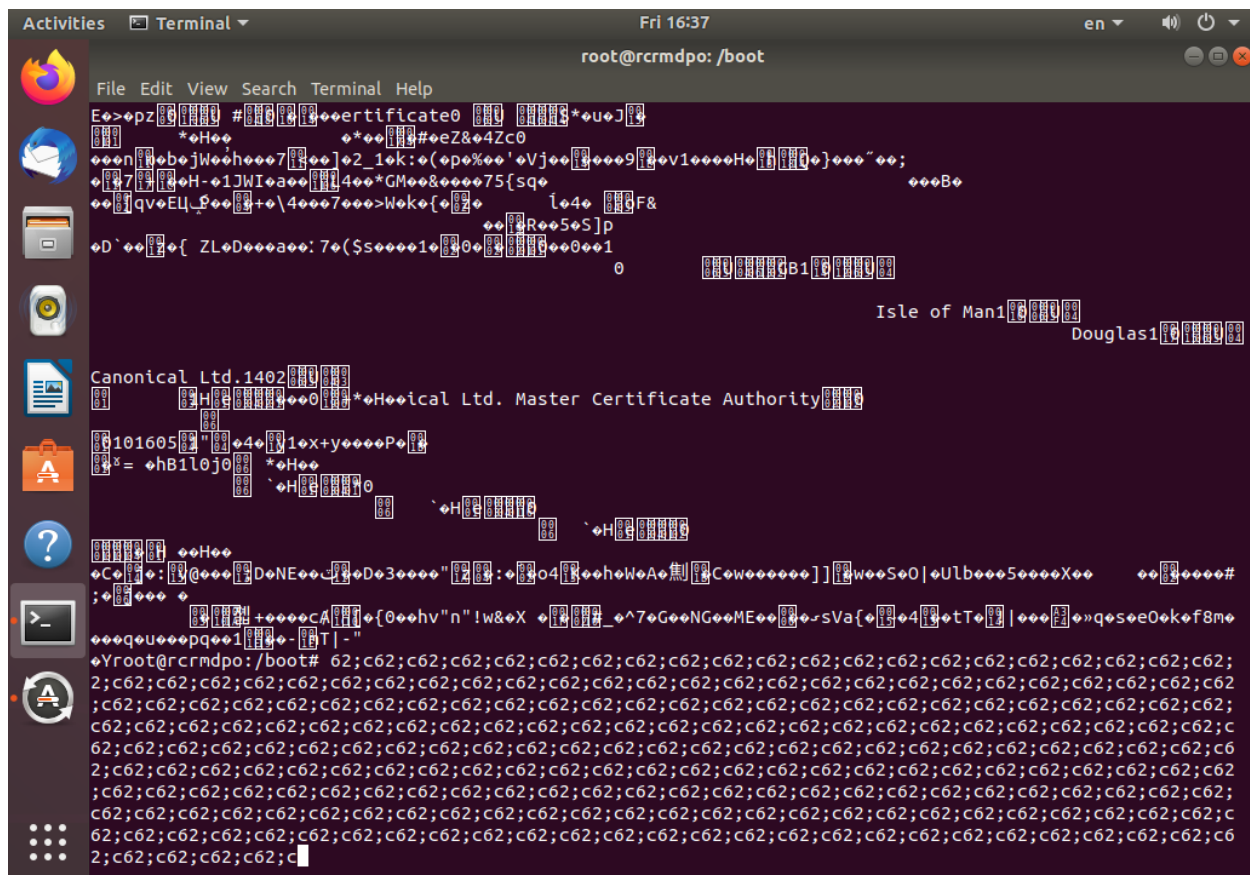
50. `vhost-vsock` – программное устройство, поэтому нет пробного вызова, который вызывает драйвер, чтобы зарегистрировать его узел устройства `misc char`. Это создает проблема с курицей и яйцом: приложения в пользовательском пространстве должны открываться `/dev / vhost-vsock`, чтобы использовать драйвер,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

51. `video` – устройство видеозахвата / наложения;

52. `zero` – источник нулевого байта;

53. `zfs` – настраивает пулы хранения ZFS.

Следующим шагом перейдём в директорию /root. В этом каталоге должен располагаться файл vmlinuz. Но в Linux Ubuntu 20.04 отсутствует данный файл, поэтому установим виртуальную машину VMware и запустим в ней Linux Ubuntu Server 18.04. И уже внутри виртуальной машины откроем файл vmlinuz:



#### Рисунок 4 – Содержимое файла vmlinuz

Все пользователи и группы пользователей имеют полные права на файл `vmlinux`. Владелцем файла указан пользователь `root`.

И продолжим работу в Linux Ubuntu 20.04. Создадим нового пользователя user с помощью команды useradd:

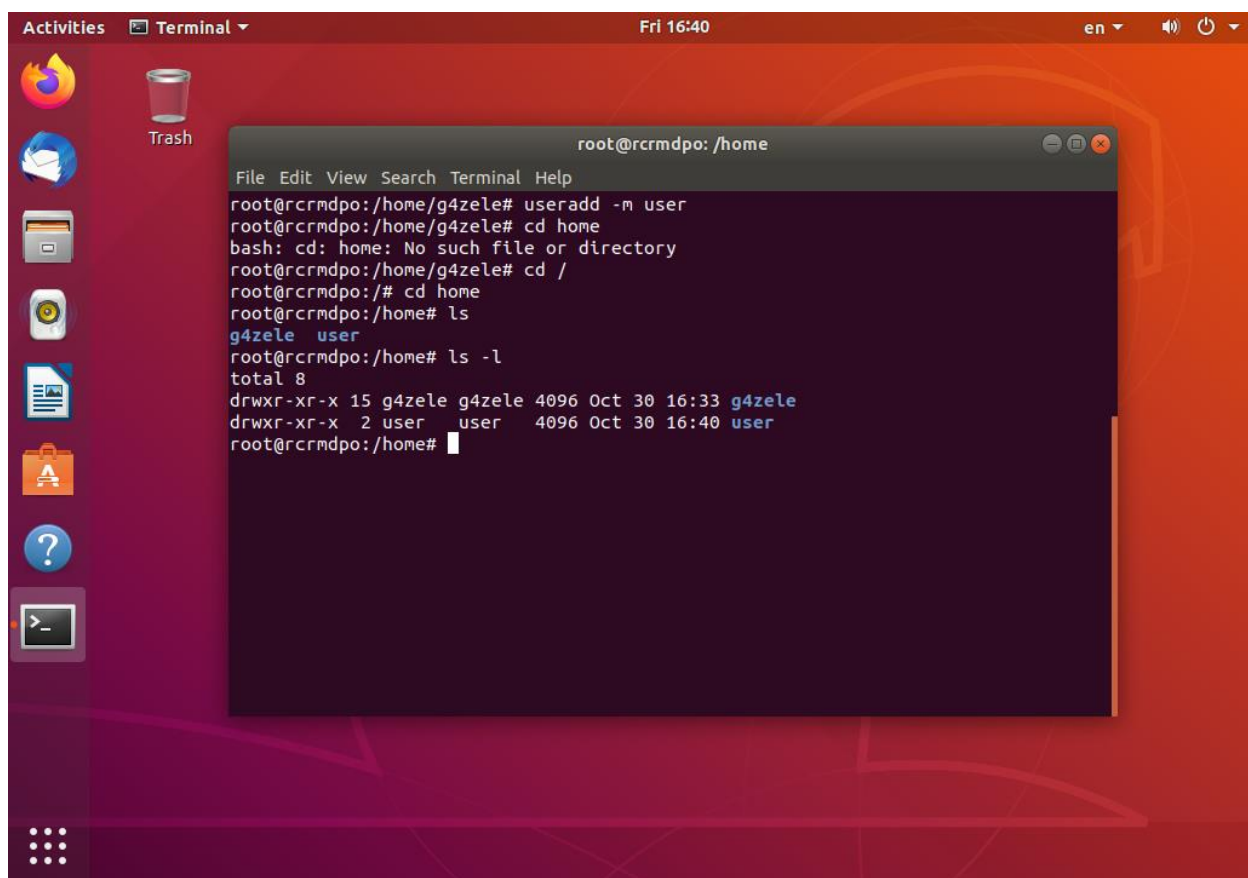
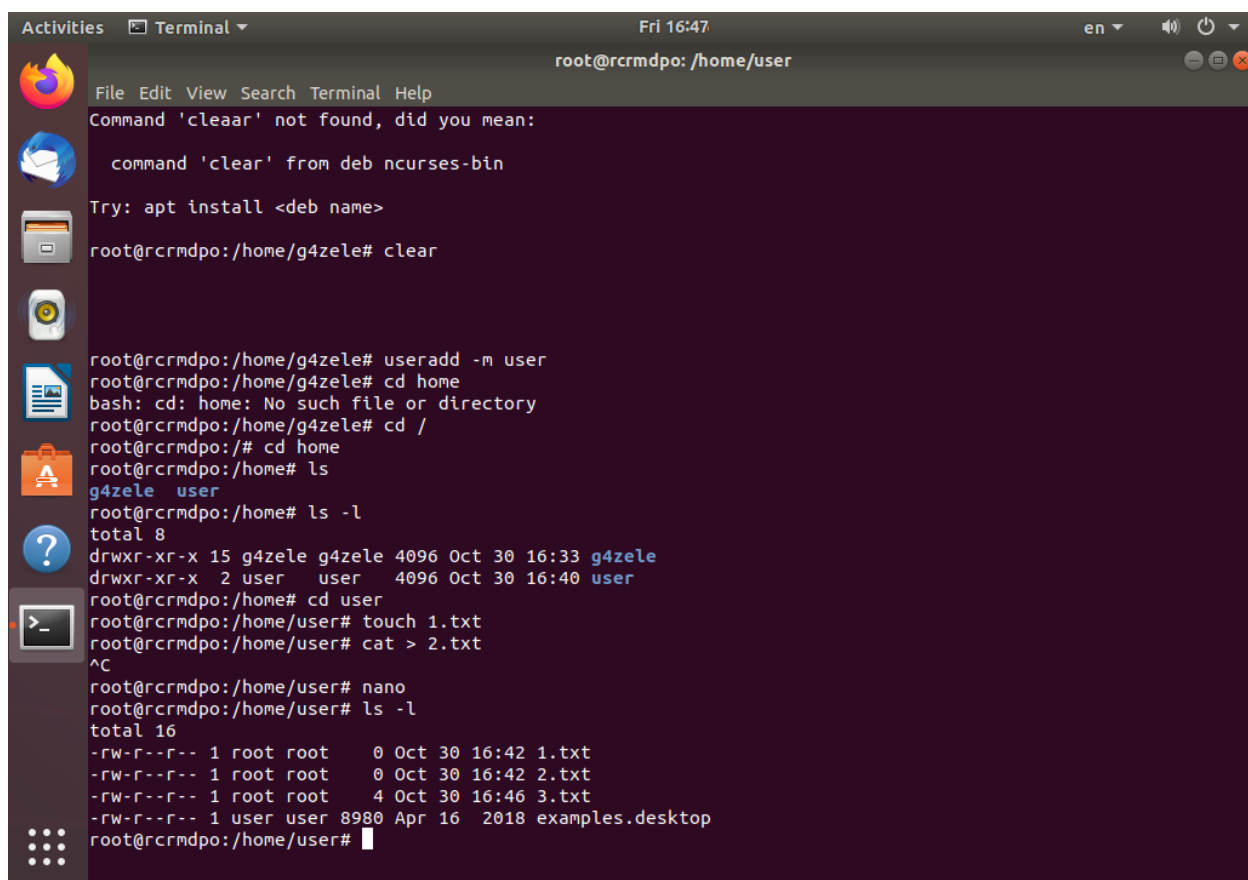


Рисунок 5 – Создание пользователя.

Создадим в директории пользователя /home/user 1.txt, 2.txt и 3.txt, используя команды touch, cat и текстовый редактор nano:



```
Activities Terminal Fri 16:47 root@rcrmdpo: /home/user
File Edit View Search Terminal Help
Command 'cleaar' not found, did you mean:
  command 'clear' from deb ncurses-bin
Try: apt install <deb name>
root@rcrmdpo:/home/g4zele# clear
root@rcrmdpo:/home/g4zele# useradd -m user
root@rcrmdpo:/home/g4zele# cd home
bash: cd: home: No such file or directory
root@rcrmdpo:/home/g4zele# cd /
root@rcrmdpo:/# cd home
root@rcrmdpo:/home# ls
g4zele user
root@rcrmdpo:/home# ls -l
total 8
drwxr-xr-x 15 g4zele g4zele 4096 Oct 30 16:33 g4zele
drwxr-xr-x  2 user  user  4096 Oct 30 16:40 user
root@rcrmdpo:/home# cd user
root@rcrmdpo:/home/user# touch 1.txt
root@rcrmdpo:/home/user# cat > 2.txt
^C
root@rcrmdpo:/home/user# nano
root@rcrmdpo:/home/user# ls -l
total 16
-rw-r--r-- 1 root root  0 Oct 30 16:42 1.txt
-rw-r--r-- 1 root root  0 Oct 30 16:42 2.txt
-rw-r--r-- 1 root root  4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
root@rcrmdpo:/home/user#
```

Рисунок 6 – Создание файлов



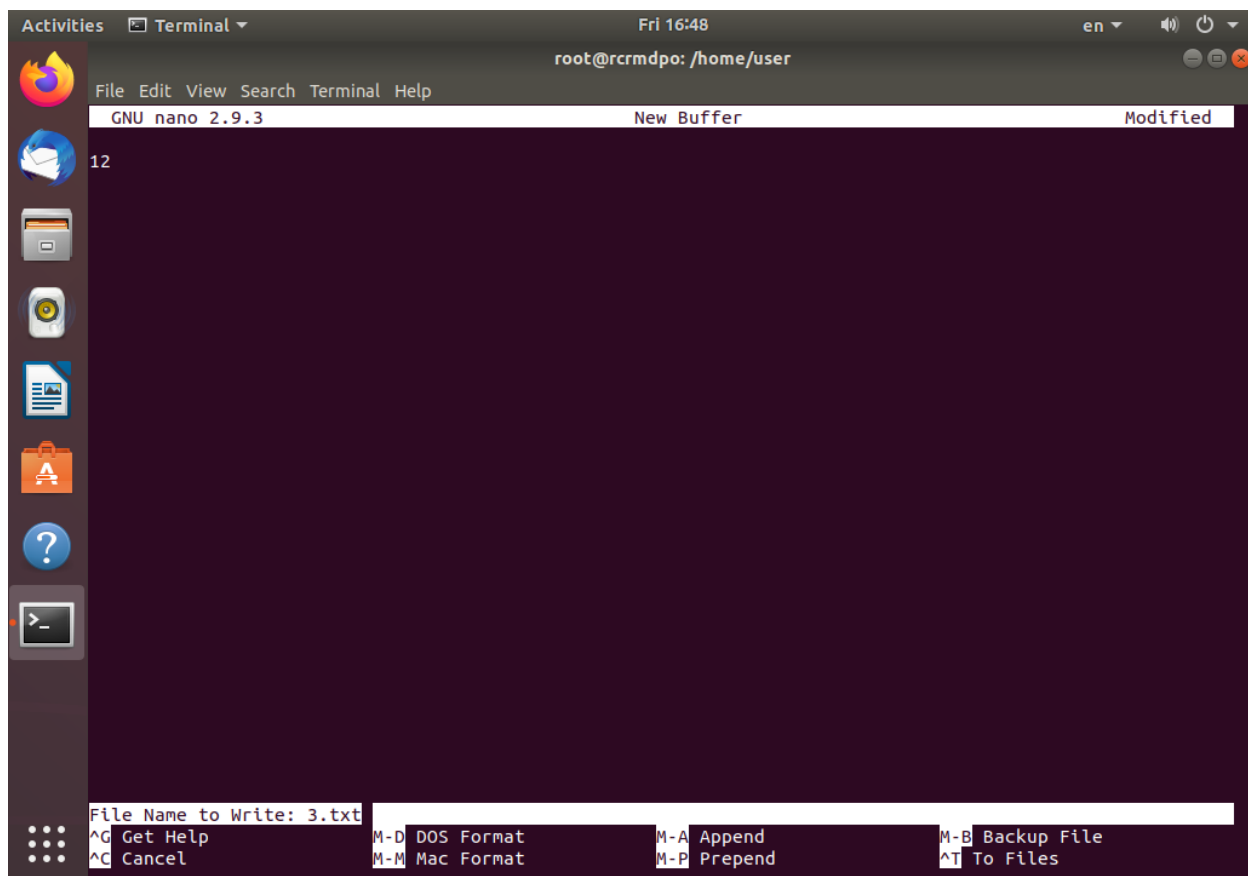
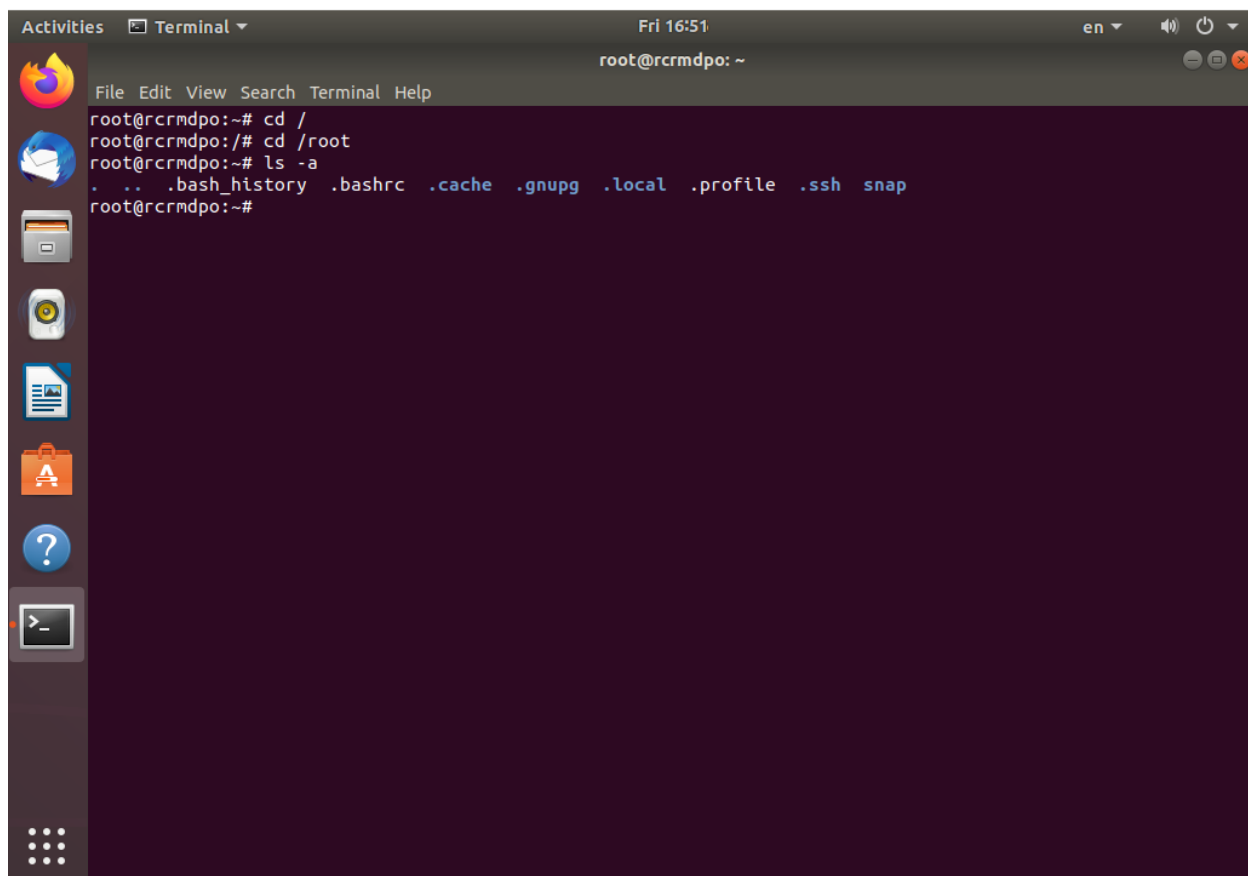


Рисунок 7 – Создание файла с помощью текстового редактора nano

Владельцем файлов является пользователь root, он имеет полные права на файлы, остальные пользователи имеют только право на чтение.

После этого перейдём в директорию /root:



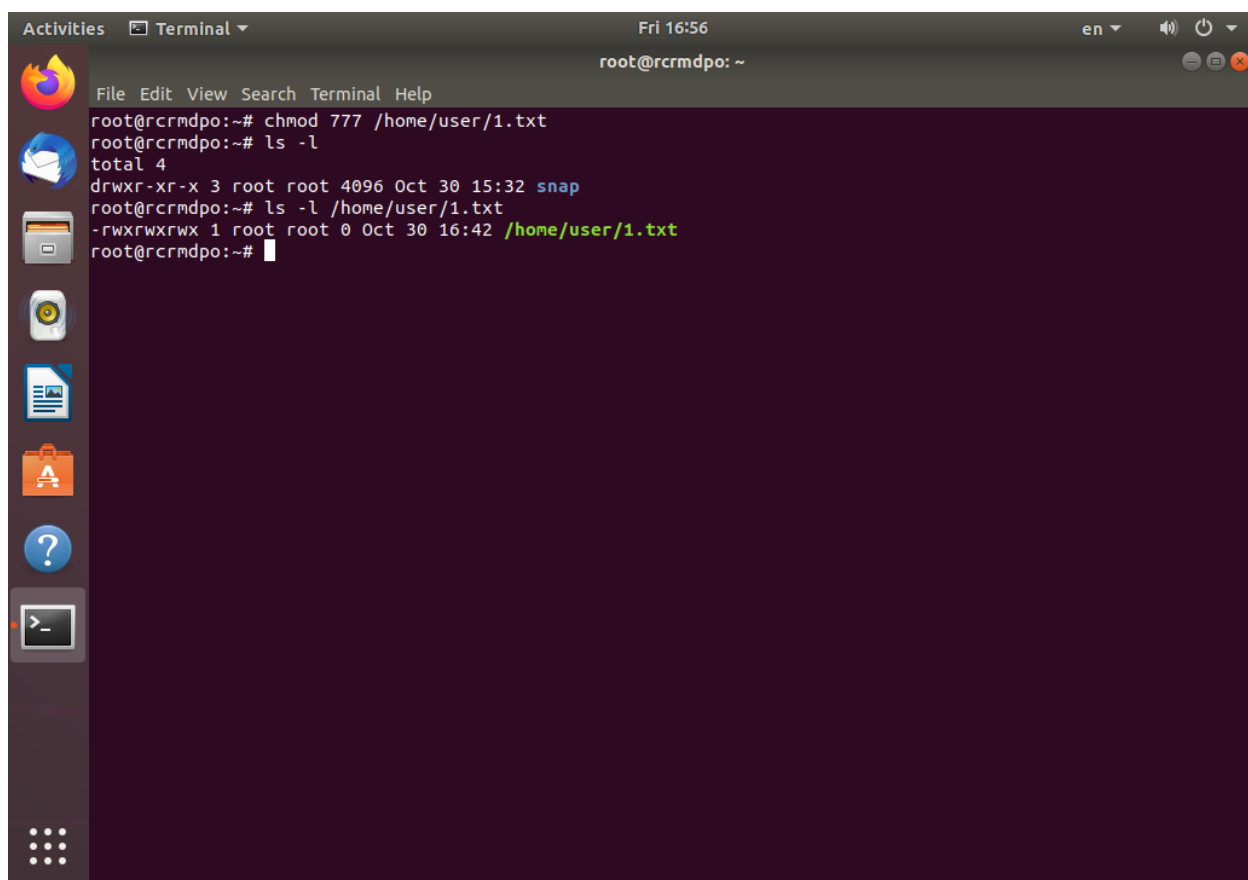
The image shows a terminal window titled "Terminal" with a menu bar containing "File", "Edit", "View", "Search", "Terminal", and "Help". The window's title bar also displays "Fri 16:51" and "en". The terminal content shows the following commands and output:

```
root@rcrmdpo:~# cd /
root@rcrmdpo:~# cd /root
root@rcrmdpo:~# ls -a
.  ..  .bash_history  .bashrc  .cache  .gnupg  .local  .profile  .ssh  snap
root@rcrmdpo:~#
```

The terminal window has a dark purple background. On the left side, there is a vertical dock with several application icons: a Firefox browser icon, a mail icon, a file manager icon, a disk icon, a document icon, a shopping bag icon, a question mark icon, and a terminal icon. At the bottom of the dock is a grid of dots representing the application menu.

Рисунок 8 – Каталог root

Совершим некоторые операции с созданными нами файлами. Для начала изменим права доступа на файл 1.txt с помощью команды `chmod`:

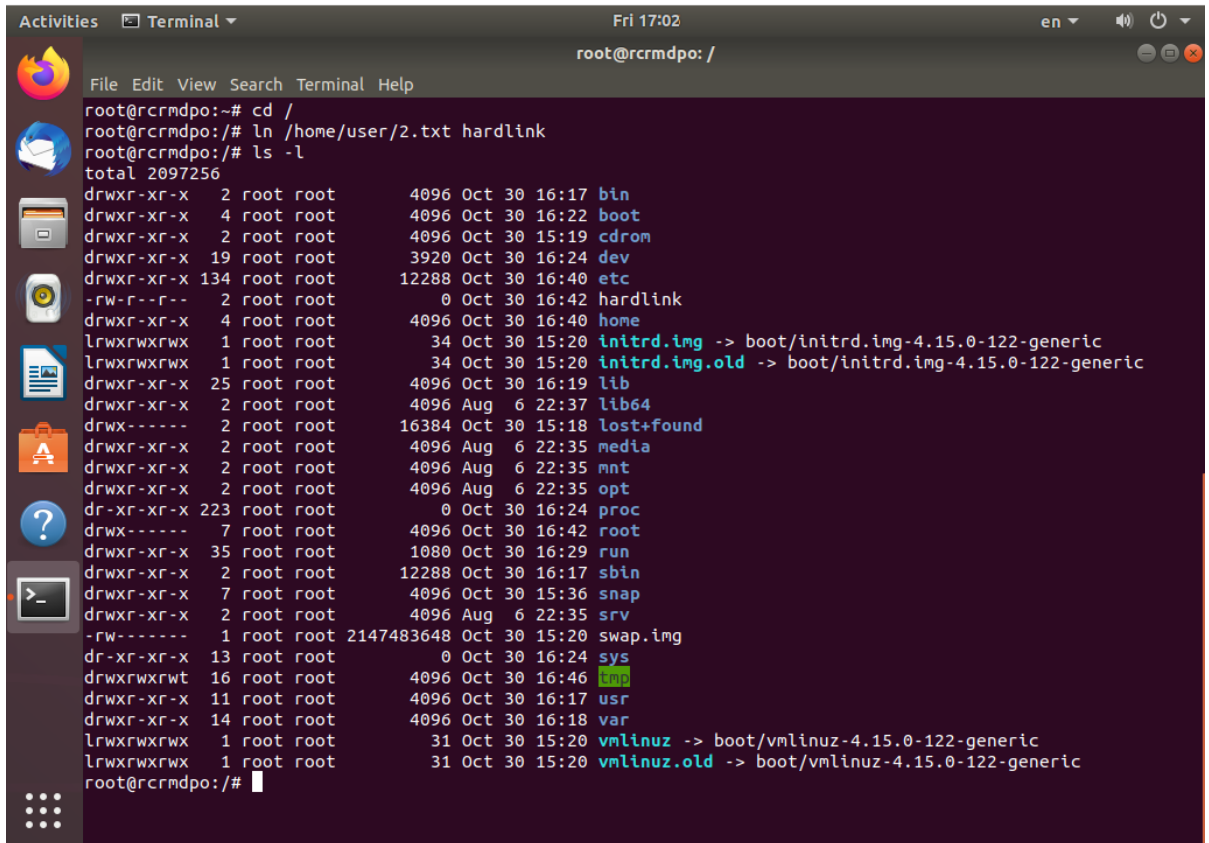


```
Activities Terminal Fri 16:56 en
root@rcrmdpo: ~
File Edit View Search Terminal Help
root@rcrmdpo:~# chmod 777 /home/user/1.txt
root@rcrmdpo:~# ls -l
total 4
drwxr-xr-x 3 root root 4096 Oct 30 15:32 snap
root@rcrmdpo:~# ls -l /home/user/1.txt
-rwxrwxrwx 1 root root 0 Oct 30 16:42 /home/user/1.txt
root@rcrmdpo:~#
```

Рисунок 9 – Изменение прав доступа к файлу

Так как после команды `chmod` было указано значение `777`, то все пользователи имеют право на чтение, изменение и исполнение файла.

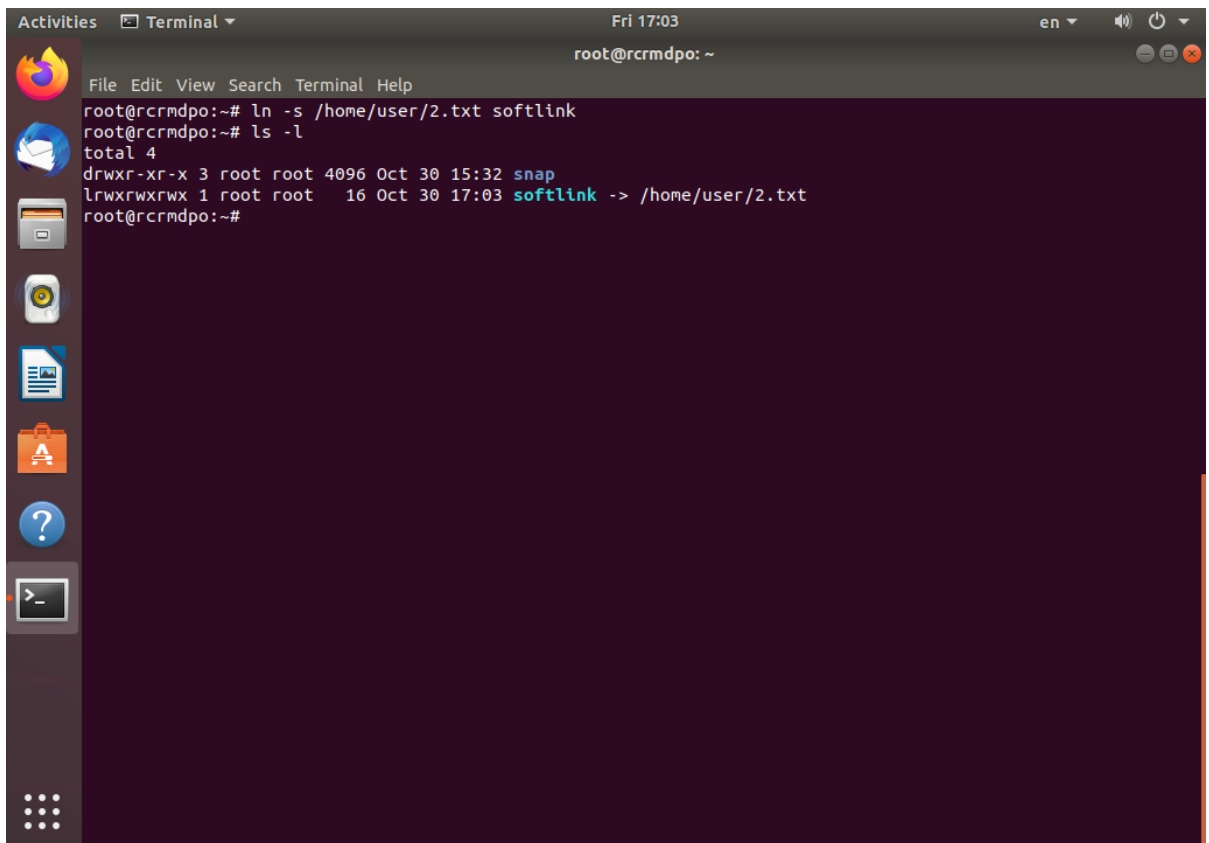
Далее создадим жёсткую и символическую ссылки на файл 2.txt:



A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Fri 17:02, root@rcrmdpo: /). The terminal shows the following commands and output:

```
root@rcrmdpo:~# cd /
root@rcrmdpo:~# ln /home/user/2.txt hardlink
root@rcrmdpo:~# ls -l
total 2097256
drwxr-xr-x  2 root root      4096 Oct 30 16:17 bin
drwxr-xr-x  4 root root      4096 Oct 30 16:22 boot
drwxr-xr-x  2 root root      4096 Oct 30 15:19 cdrom
drwxr-xr-x 19 root root     3920 Oct 30 16:24 dev
drwxr-xr-x 134 root root    12288 Oct 30 16:40 etc
-rw-r--r--  2 root root         0 Oct 30 16:42 hardlink
drwxr-xr-x  4 root root      4096 Oct 30 16:40 home
lrwxrwxrwx  1 root root        34 Oct 30 15:20 initrd.img -> boot/initrd.img-4.15.0-122-generic
lrwxrwxrwx  1 root root        34 Oct 30 15:20 initrd.img.old -> boot/initrd.img-4.15.0-122-generic
drwxr-xr-x 25 root root      4096 Oct 30 16:19 lib
drwxr-xr-x  2 root root      4096 Aug  6 22:37 lib64
drwxr-xr-x  2 root root    16384 Oct 30 15:18 lost+found
drwxr-xr-x  2 root root      4096 Aug  6 22:35 media
drwxr-xr-x  2 root root      4096 Aug  6 22:35 mnt
drwxr-xr-x  2 root root      4096 Aug  6 22:35 opt
dr-xr-xr-x 223 root root         0 Oct 30 16:24 proc
drwxr-xr-x  7 root root      4096 Oct 30 16:42 root
drwxr-xr-x 35 root root     1080 Oct 30 16:29 run
drwxr-xr-x  2 root root    12288 Oct 30 16:17/sbin
drwxr-xr-x  7 root root      4096 Oct 30 15:36 snap
drwxr-xr-x  2 root root      4096 Aug  6 22:35 srv
-rw-r--r--  1 root root 2147483648 Oct 30 15:20 swap.img
dr-xr-xr-x 13 root root         0 Oct 30 16:24 sys
drwxrwxrwt 16 root root      4096 Oct 30 16:46 tmp
drwxr-xr-x 11 root root      4096 Oct 30 16:17 usr
drwxr-xr-x 14 root root      4096 Oct 30 16:18 var
lrwxrwxrwx  1 root root        31 Oct 30 15:20 vmlinuz -> boot/vmlinuz-4.15.0-122-generic
lrwxrwxrwx  1 root root        31 Oct 30 15:20 vmlinuz.old -> boot/vmlinuz-4.15.0-122-generic
root@rcrmdpo:~#
```

Рисунок 10 – Создание жёсткой ссылки

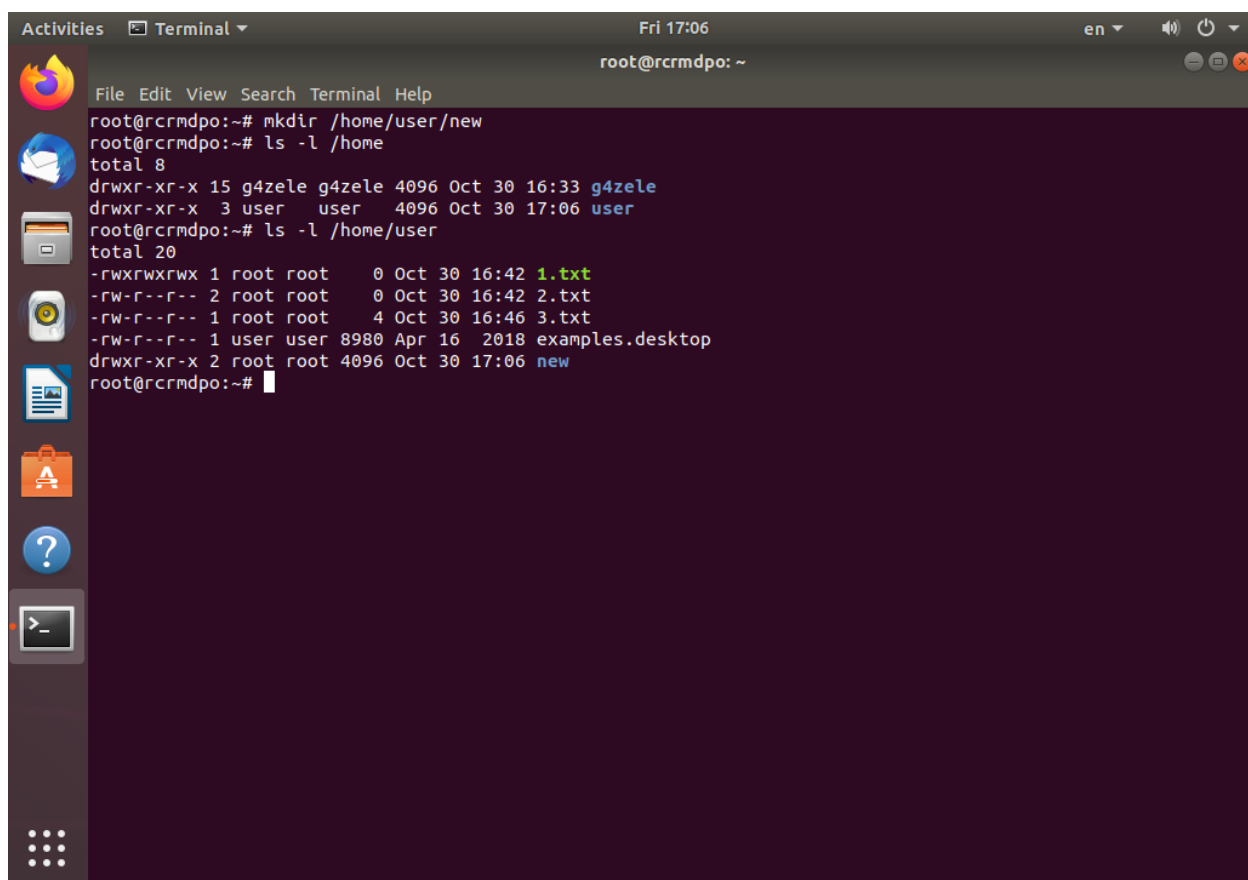


A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Fri 17:03, root@rcrmdpo: ~). The terminal shows the following commands and output:

```
root@rcrmdpo:~# ln -s /home/user/2.txt softlink
root@rcrmdpo:~# ls -l
total 4
drwxr-xr-x  3 root root      4096 Oct 30 15:32 snap
lrwxrwxrwx  1 root root        16 Oct 30 17:03 softlink -> /home/user/2.txt
root@rcrmdpo:~#
```

Рисунок 11 – Создание символической ссылки

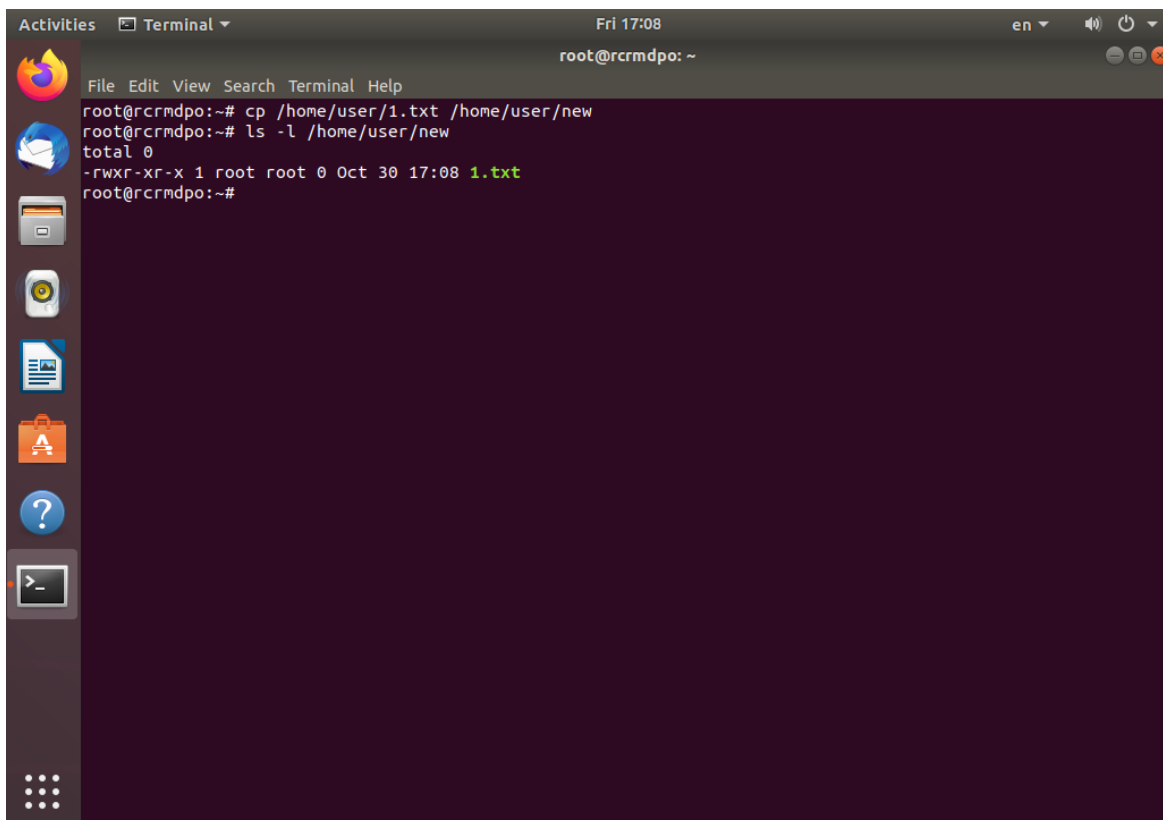
После этого требуется создать новую директорию `new` в каталоге пользователя `user`. Для этого используем команду `mkdir`:

A screenshot of a Linux terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a root user at a machine named "rcrmdpo". The user enters the command "mkdir /home/user/new". Then, they run "ls -l /home" showing two directories: "g4zele" and "user". Next, they run "ls -l /home/user" showing several files and the newly created "new" directory. The terminal output is as follows:

```
root@rcrmdpo:~# mkdir /home/user/new
root@rcrmdpo:~# ls -l /home
total 8
drwxr-xr-x 15 g4zele g4zele 4096 Oct 30 16:33 g4zele
drwxr-xr-x  3 user   user   4096 Oct 30 17:06 user
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root    0 Oct 30 16:42 1.txt
-rw-r--r-- 2 root root    0 Oct 30 16:42 2.txt
-rw-r--r-- 1 root root    4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16  2018 examples.desktop
drwxr-xr-x 2 root root 4096 Oct 30 17:06 new
root@rcrmdpo:~#
```

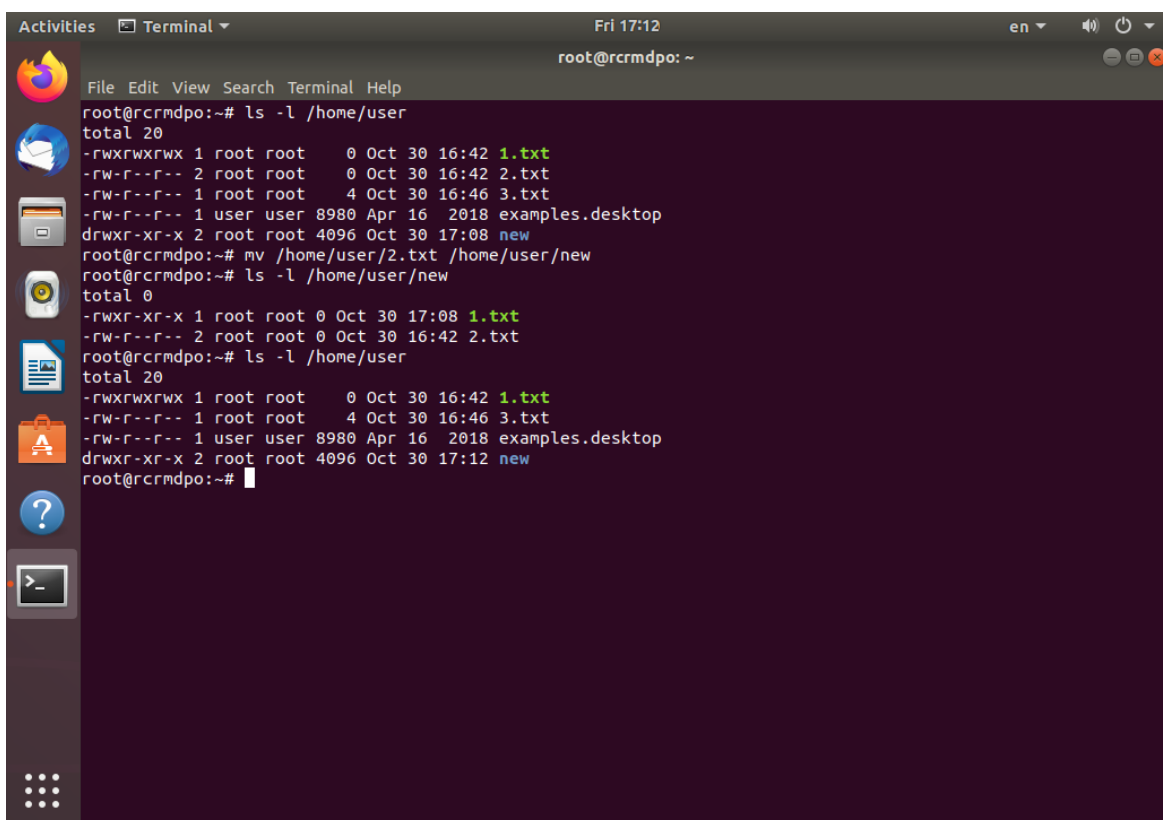
Рисунок 12 – Создание каталога в директории пользователя

Теперь копируем файл 1.txt и переместим файл 2.txt в созданную директорию:

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Fri 17:08, en, and system icons). The prompt is 'root@rcrmdpo: ~'. The user enters 'cp /home/user/1.txt /home/user/new'. The prompt changes to 'root@rcrmdpo:~#'. The user enters 'ls -l /home/user/new'. The output shows 'total 0' and '-rwxr-xr-x 1 root root 0 Oct 30 17:08 1.txt'. The prompt returns to 'root@rcrmdpo:~#'.

```
root@rcrmdpo:~# cp /home/user/1.txt /home/user/new
root@rcrmdpo:~# ls -l /home/user/new
total 0
-rwxr-xr-x 1 root root 0 Oct 30 17:08 1.txt
root@rcrmdpo:~#
```

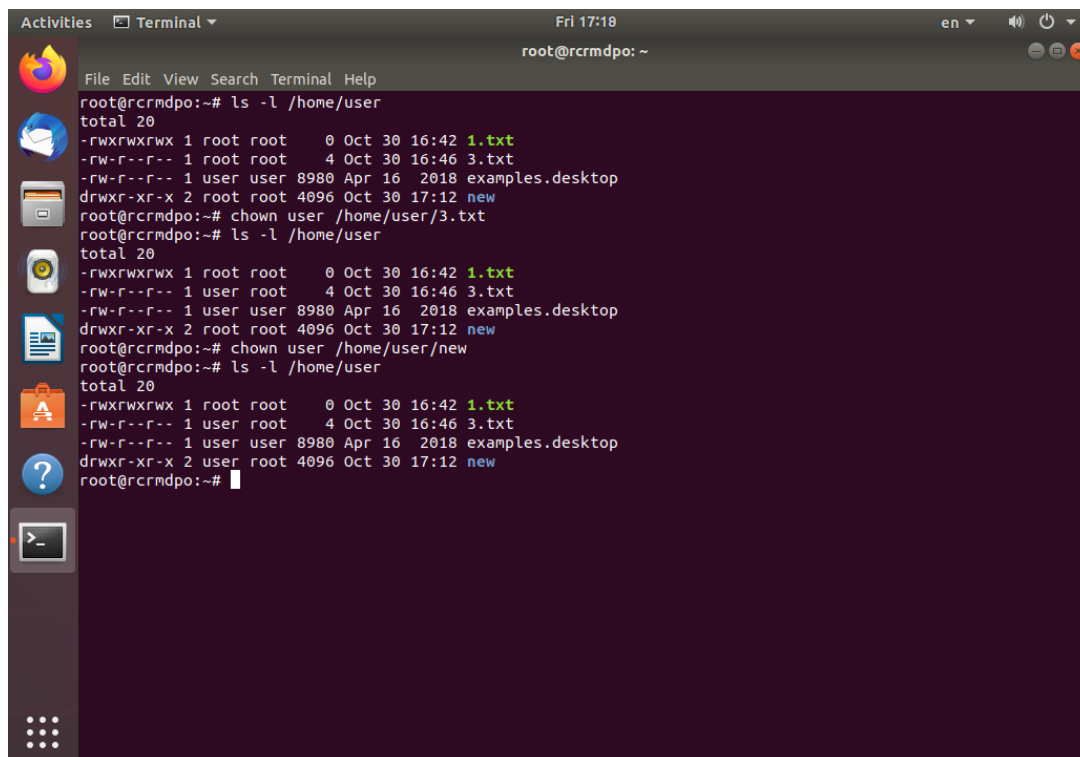
Рисунок 13 – Копирование файла

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Fri 17:12, en, and system icons). The prompt is 'root@rcrmdpo: ~'. The user enters 'ls -l /home/user'. The output shows 'total 20' and a list of files including 1.txt, 2.txt, 3.txt, examples.desktop, and new. The prompt changes to 'root@rcrmdpo:~#'. The user enters 'mv /home/user/2.txt /home/user/new'. The prompt changes to 'root@rcrmdpo:~#'. The user enters 'ls -l /home/user/new'. The output shows 'total 0' and '-rwxr-xr-x 1 root root 0 Oct 30 17:08 1.txt'. The prompt returns to 'root@rcrmdpo:~#'. The user enters 'ls -l /home/user'. The output shows 'total 20' and the same list of files as before. The prompt returns to 'root@rcrmdpo:~#'.

```
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root  0 Oct 30 16:42 1.txt
-rw-r--r-- 2 root root  0 Oct 30 16:42 2.txt
-rw-r--r-- 1 root root  4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16  2018 examples.desktop
drwxr-xr-x 2 root root 4096 Oct 30 17:08 new
root@rcrmdpo:~# mv /home/user/2.txt /home/user/new
root@rcrmdpo:~# ls -l /home/user/new
total 0
-rwxr-xr-x 1 root root 0 Oct 30 17:08 1.txt
-rw-r--r-- 2 root root 0 Oct 30 16:42 2.txt
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root  0 Oct 30 16:42 1.txt
-rw-r--r-- 1 root root  4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16  2018 examples.desktop
drwxr-xr-x 2 root root 4096 Oct 30 17:12 new
root@rcrmdpo:~#
```

Рисунок 14 – Перемещение файла

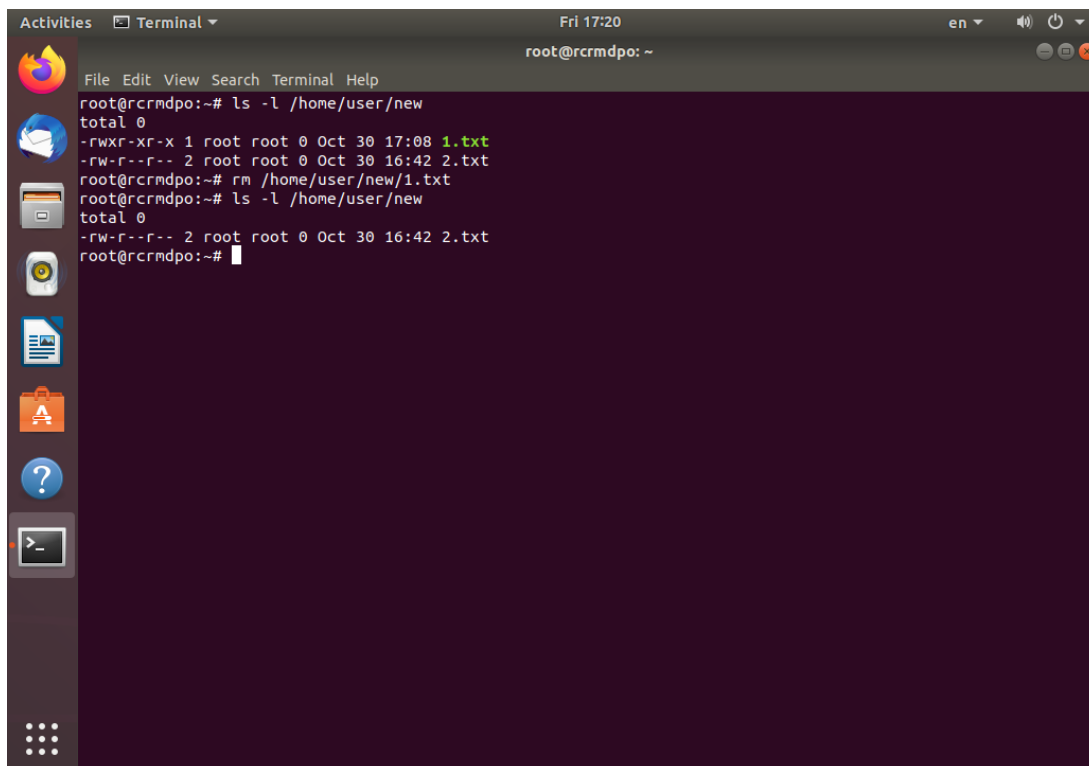
После этого поменяем владельцев файла 3.txt и каталога new. Сделаем это с помощью команды `chown`:



```
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root  0 Oct 30 16:42 1.txt
-rw-r--r-- 1 root root  4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
drwxr-xr-x 2 root root 4096 Oct 30 17:12 new
root@rcrmdpo:~# chown user /home/user/3.txt
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root  0 Oct 30 16:42 1.txt
-rw-r--r-- 1 user root  4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
drwxr-xr-x 2 root root 4096 Oct 30 17:12 new
root@rcrmdpo:~# chown user /home/user/new
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root  0 Oct 30 16:42 1.txt
-rw-r--r-- 1 user root  4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
drwxr-xr-x 2 user root 4096 Oct 30 17:12 new
root@rcrmdpo:~#
```

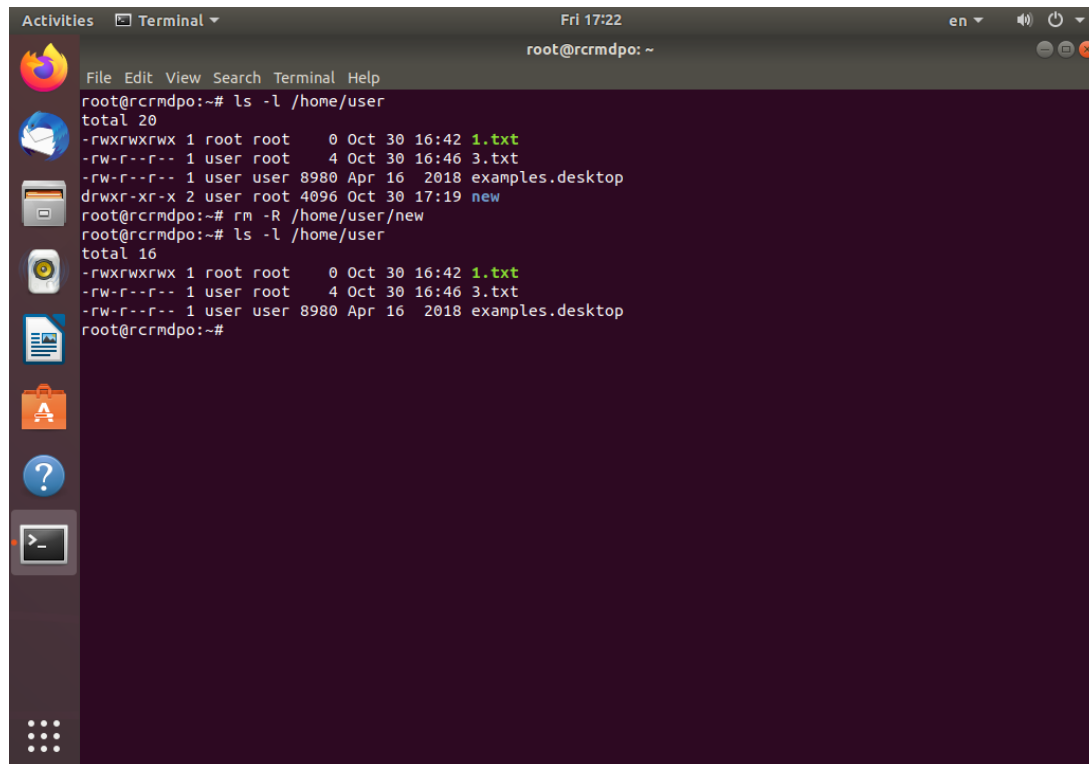
Рисунок 15 – Изменение владельца файла и каталога

Теперь удалим файл 1.txt из директории new, а затем удалим и саму директорию. Используем для этого команду rm:

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Fri 17:20, root@rcrmdpo: ~). The terminal shows the following commands and output:

```
root@rcrmdpo:~# ls -l /home/user/new
total 0
-rwxr-xr-x 1 root root 0 Oct 30 17:08 1.txt
-rw-r--r-- 2 root root 0 Oct 30 16:42 2.txt
root@rcrmdpo:~# rm /home/user/new/1.txt
root@rcrmdpo:~# ls -l /home/user/new
total 0
-rw-r--r-- 2 root root 0 Oct 30 16:42 2.txt
root@rcrmdpo:~#
```

Рисунок 16 – Удаление файла

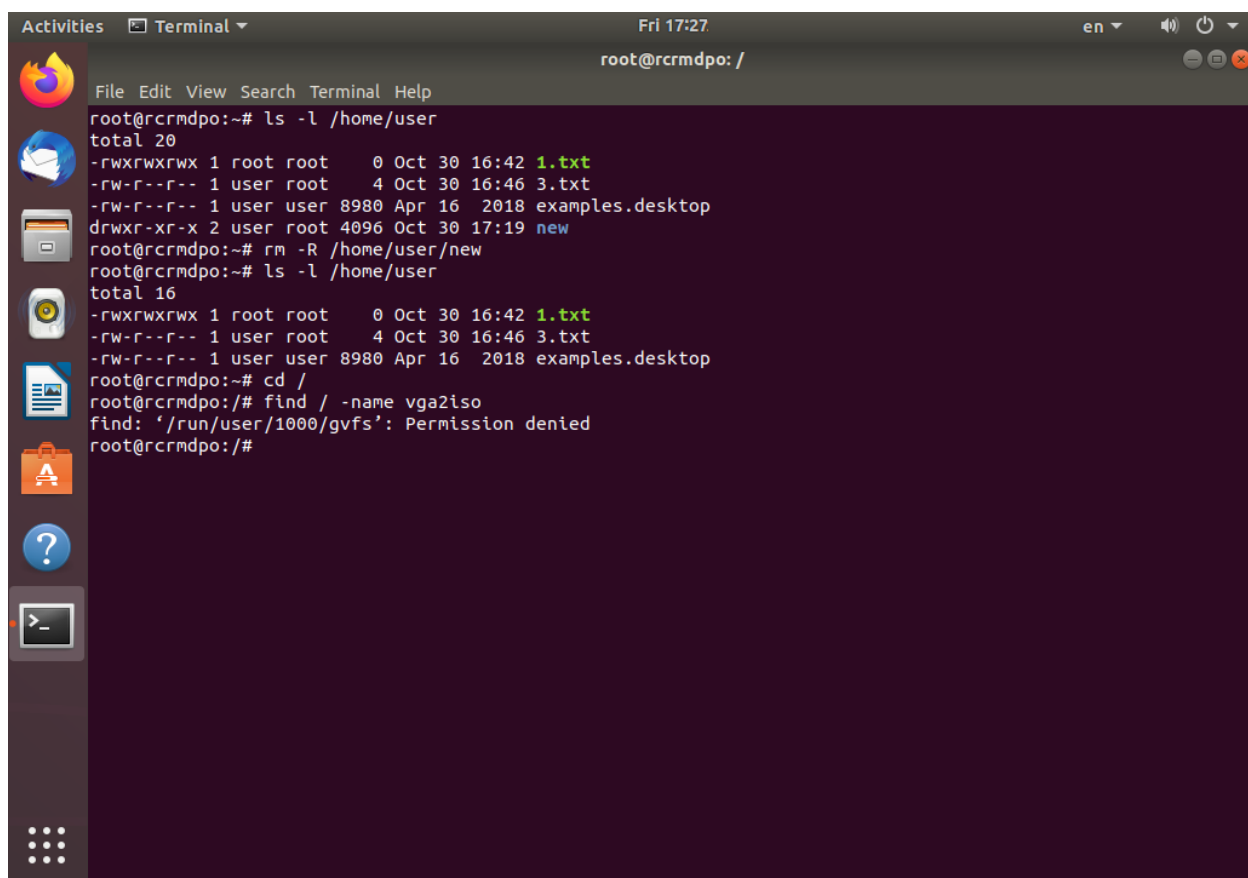
A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Fri 17:22, root@rcrmdpo: ~). The terminal shows the following commands and output:

```
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root 0 Oct 30 16:42 1.txt
-rw-r--r-- 1 user root 4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
drwxr-xr-x 2 user root 4096 Oct 30 17:19 new
root@rcrmdpo:~# rm -R /home/user/new
root@rcrmdpo:~# ls -l /home/user
total 16
-rwxrwxrwx 1 root root 0 Oct 30 16:42 1.txt
-rw-r--r-- 1 user root 4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
root@rcrmdpo:~#
```

Рисунок 17 – Удаление каталога



Последним заданием лабораторной работы является поиск файла `vga2iso` с использованием команды `find`. Осуществим эту операцию:



```
Activities  Terminal  Fri 17:27  en  [volume icon] [power icon]
root@rcrmdpo: /
File Edit View Search Terminal Help
root@rcrmdpo:~# ls -l /home/user
total 20
-rwxrwxrwx 1 root root    0 Oct 30 16:42 1.txt
-rw-r--r-- 1 user root    4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
drwxr-xr-x 2 user root 4096 Oct 30 17:19 new
root@rcrmdpo:~# rm -R /home/user/new
root@rcrmdpo:~# ls -l /home/user
total 16
-rwxrwxrwx 1 root root    0 Oct 30 16:42 1.txt
-rw-r--r-- 1 user root    4 Oct 30 16:46 3.txt
-rw-r--r-- 1 user user 8980 Apr 16 2018 examples.desktop
root@rcrmdpo:~# cd /
root@rcrmdpo:/# find / -name vga2iso
find: '/run/user/1000/gvfs': Permission denied
root@rcrmdpo:/#
```

Рисунок 18 – Поиск файла

## Вывод

В процессе выполнения лабораторной работы была изучена файловая система ОС Linux и основные операции, а именно: просмотр директории, создание нового пользователя, различные операции с файлами (создание, перемещение, копирование, удаление, изменение прав доступа на файл), создание директории, поиск файла и изменение прав доступа на файл.

Также изучены особенности установки виртуальной машины с последующим запуском в ней дистрибутива Linux Ubuntu Server.

## Ответы на 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 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.

Права доступа и информация о типе файла в UNIX-системах хранятся в индексных дескрипторах в отдельной структуре, состоящей из двух байтов. Четыре бита из этих 16-ти отведены для кодированной записи о типе файла. И, наконец, оставшиеся 9 бит определяют права доступа к файлу. Право на чтение (r) файла означает, что пользователь может просматривать содержимое файла. Но вы не сможете сохранить изменения в файле, если не имеете права на запись (w) в этот файл. Право на выполнение (x) означает, что вы можете попытаться запустить его на выполнение как исполняемую программу.

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

Жесткая ссылка является просто другим именем для исходного файла. После создания такой ссылки ее невозможно отличить от исходного имени файла. «Настоящего» имени у файла нет, точнее, все такие имена будут настоящими. Удаление файла по любому из его имён уменьшает на единицу количество ссылок, и окончательно файл будет удален только тогда, когда это количество станет равным нулю. Поэтому удобно использовать жесткие ссылки для того, чтобы предотвратить случайное удаление важного файла.

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

Команда `find` 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. Проще всего задать в качестве начального каталога поиска корневой каталог /, однако, в таком случае поиск может затянуться очень надолго, так как будет просматриваться вся структура каталогов, включая смонтированные файловые системы.

5. Перечислите основные команды работы с каталогами.

- 1) Просмотр каталога (list): ls -ключи путь/имя\_файла;
- 2) Узнать текущий каталог: pwd;
- 3) Сменить текущий каталог: cd имя\_каталога;
- 4) Создание нового каталога: mkdir путь/имя\_каталога;
- 5) Удаление пустого каталога: rmdir путь/имя\_каталога.