

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

Лобов М.Ю.

Группа АИ-18

Руководитель

Кургасов В.В.

к.п.н.

Липецк 2020 г.

Оглавление

Цель работы	3
Задание кафедры.....	4
Ход работы.....	5
Вывод.....	23
Ответы на контрольные вопросы	24

Цель работы

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

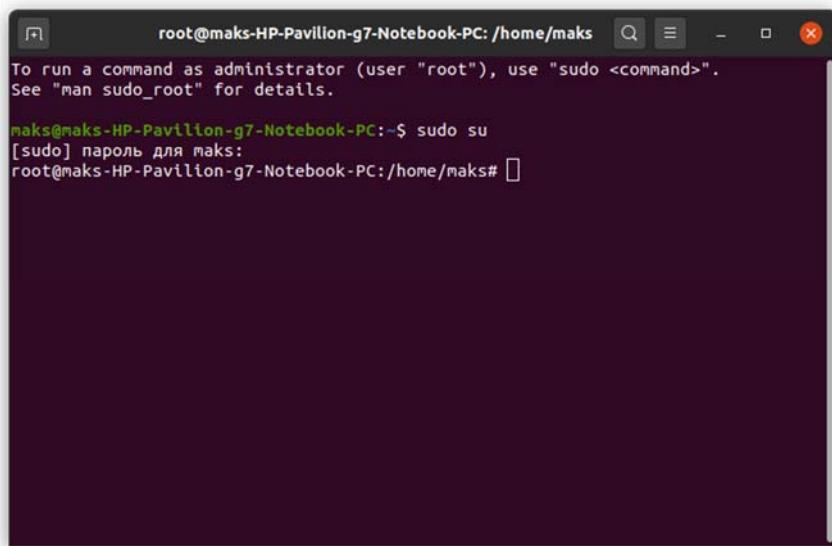
Задание кафедры

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 директорий 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 символьическую 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 в 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 запустим терминал с помощью комбинации клавиш Ctrl+Alt+T и загрузимся пользователем root с помощью команды sudo su:

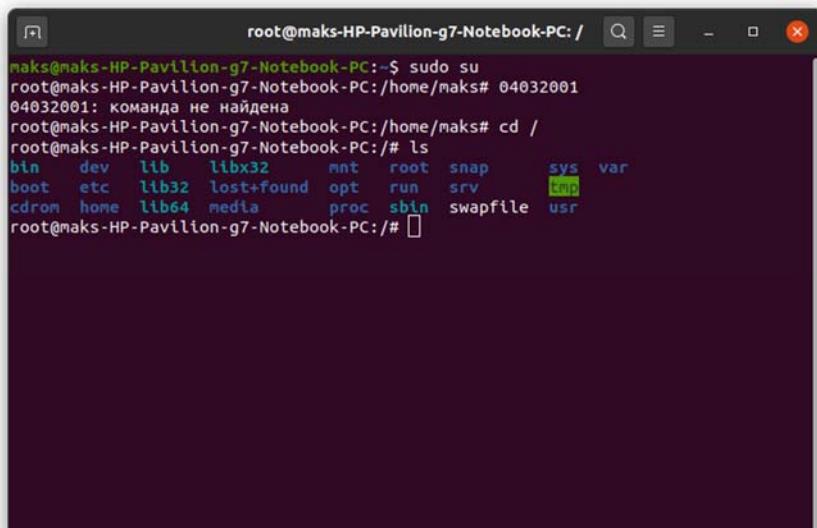


```
root@maks-HP-Pavilion-g7-Notebook-PC: /home/maks
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

maks@maks-HP-Pavilion-g7-Notebook-PC:~$ sudo su
[sudo] пароль для maks:
root@maks-HP-Pavilion-g7-Notebook-PC:/home/maks# 
```

Рисунок 1 – Загрузка пользователем root

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



```
root@maks-HP-Pavilion-g7-Notebook-PC:~$ sudo su
root@maks-HP-Pavilion-g7-Notebook-PC:/home/maks# 04032001
04032001: команда не найдена
root@maks-HP-Pavilion-g7-Notebook-PC:/home/maks# cd /
root@maks-HP-Pavilion-g7-Notebook-PC:/# ls
bin  dev  lib  libx32  mnt  root  snap  sys  var
boot  etc  lib32  lost+found  opt  run  srv  tmp
cdrom  home  lib64  media  proc  sbin  swapfile  usr
root@maks-HP-Pavilion-g7-Notebook-PC:/# 
```

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` - символьской 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 специфичные 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 складываются 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 просмотрим её:

```
root@maks-HP-Pavilion-g7-Notebook-PC:/dev# ls
autofs          i2c-4      psaux     tty16    tty45      ttyS15   vcs1
block          i2c-5      ptmx      tty17    tty46      ttyS16   vcs2
bsg            i2c-6      pts       tty18    tty47      ttyS17   vcs3
btrfs-control  i2c-7      random    tty19    tty48      ttyS18   vcs4
bus             i2c-8      rfkill    tty2     tty49      ttyS19   vcs5
char            i2c-9      rtc       tty20    tty5      ttyS2    vcs6
console        initctl   rtc0      tty21    tty50      ttyS20   vcsa
core            input     sda      tty22    tty51      ttyS21   vcsa1
cpu             kmsq     sda1     tty23    tty52      ttyS22   vcsa2
cpu_dma_latency kvm      sda2     tty24    tty53      ttyS23   vcsa3
cuse            lightnvm sdb      tty25    tty54      ttyS24   vcsa4
disk            log      sdb1     tty26    tty55      ttyS25   vcsa5
dri             loop0    sdb2     tty27    tty56      ttyS26   vcsa6
drm_dp_aux0    loop1    sdb3     tty28    tty57      ttyS27   vcsu
ecryptfs       loop2    sdb5     tty29    tty58      ttyS28   vcsu1
fb0            loop3    sg0      tty3     tty59      ttyS29   vcsu2
fd              loop4    sg1      tty30    tty6      ttyS3    vcsu3
full           loop5    shm      tty31    tty60      ttyS30   vcsu4
fuse           loop6    snapshot  tty32    tty61      ttyS31   vcsu5
hpet           loop7    snd      tty33    tty62      ttyS4    vcsu6
hugepages      loop-control stderr   tty34    tty63      ttyS5    vfio
hwng           mapper   stdin    tty35    tty7      ttyS6    vga_arbiter
i2c-0          mclog    stdout   tty36    tty8      ttyS7    vhci
i2c-1          media0   tty      tty37    tty9      ttyS8    vhost-net
i2c-10         mei0     tty0     tty38    ttyprintk  ttyS9    vhost-vsock
i2c-11         mem      tty1     tty39    tty50      udmabuf  video0
i2c-12         mqueue   tty10    tty4     tty51      uhid     video1
i2c-13         net      tty11    tty40    tty510     uinput    zero
i2c-14         null    tty12    tty41    tty511     urandom   zfs
i2c-15         nvram   tty13    tty42    tty512     userio   v4l
i2c-2          port    tty14    tty43    tty513     vcs
i2c-3          PPP     tty15    tty44    tty514
```

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 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- – 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-

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 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. `ppp` – 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 `pppd`,

через символьное устройство / dev / ppp, 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 – 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. ttysize – драйвер псевдо TTY, который позволяет пользователям создавать сообщения printk через вывод на устройство ttysize;

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 – 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 ne;
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:

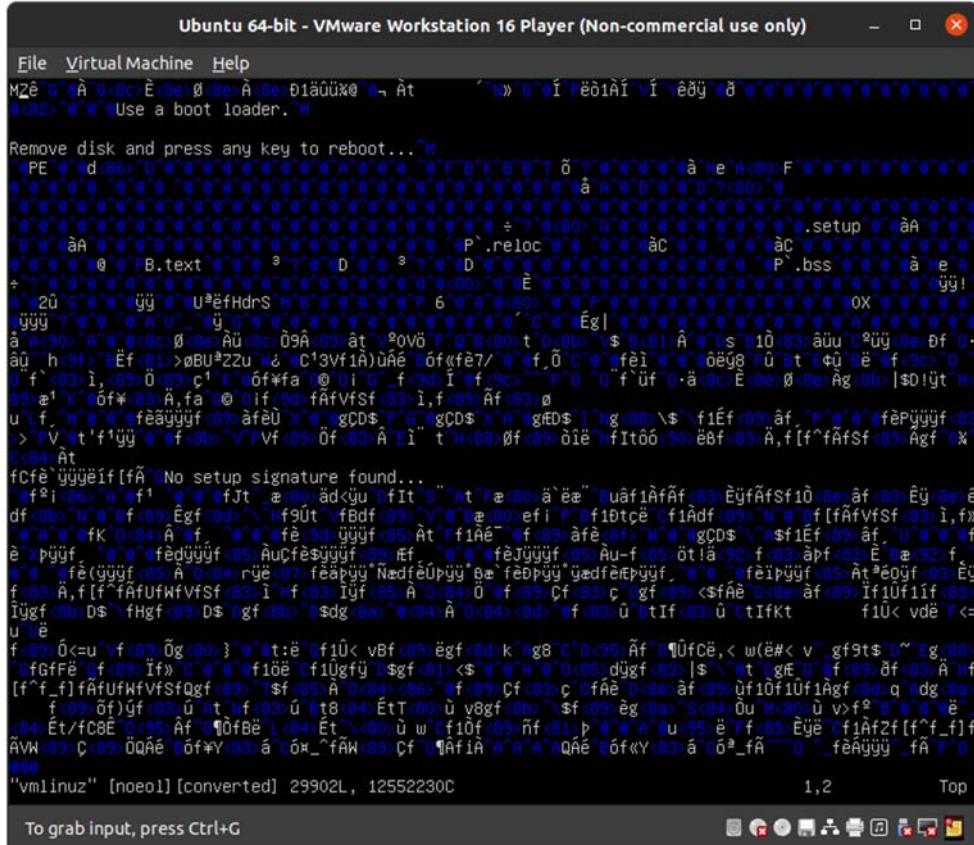
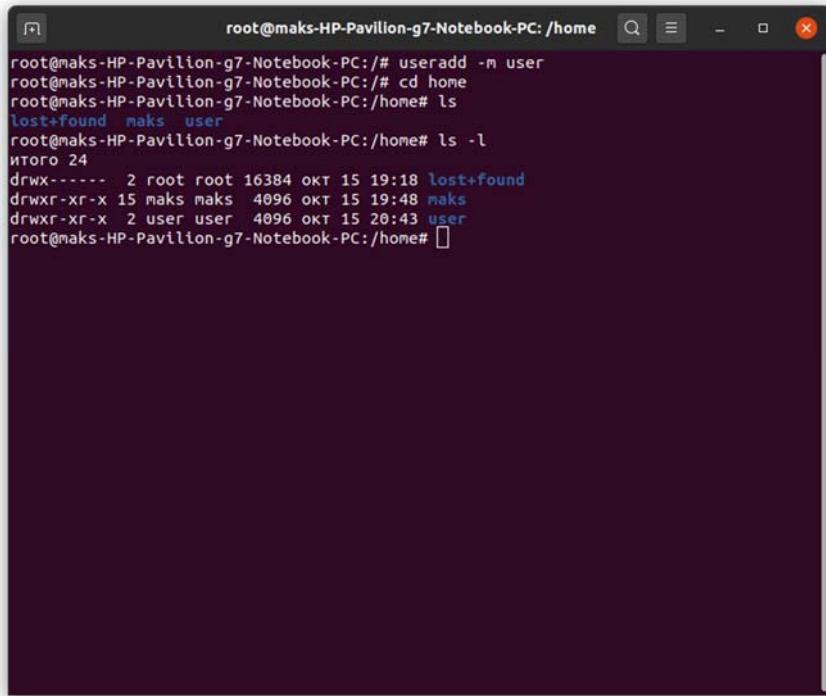


Рисунок 4 – Содержимое файла vmlinuz

Все пользователи и группы пользователей имеют полные права на файл vmlinuz. Владельцем файла указан пользователь root.

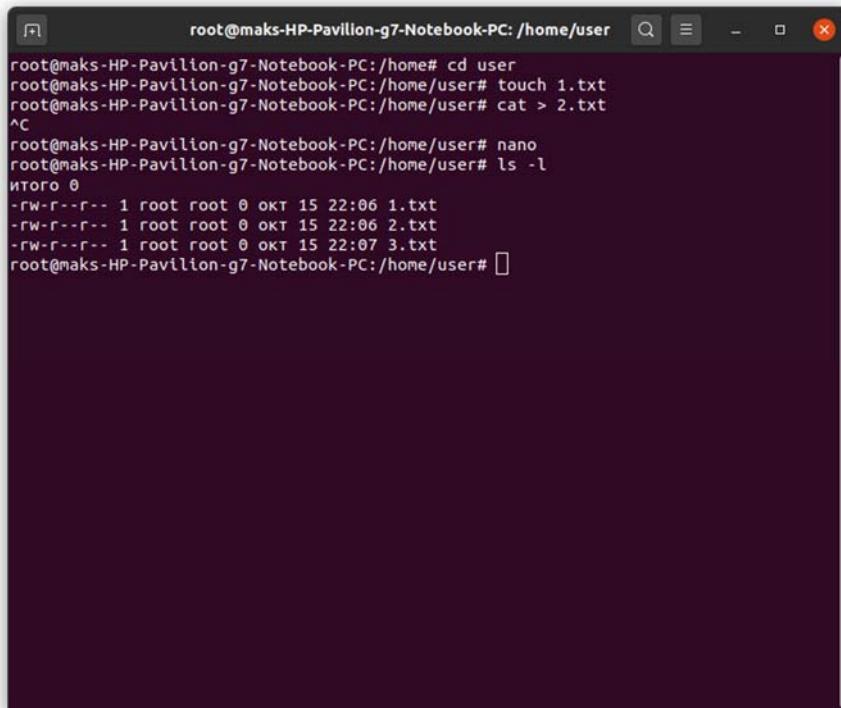
И продолжим работу в Linux Ubuntu 20.04. Создадим нового пользователя user с помощью команды useradd:



```
root@maks-HP-Pavilion-g7-Notebook-PC: /home
root@maks-HP-Pavilion-g7-Notebook-PC:/# useradd -m user
root@maks-HP-Pavilion-g7-Notebook-PC:/# cd home
root@maks-HP-Pavilion-g7-Notebook-PC:/home# ls
lost+found maks user
root@maks-HP-Pavilion-g7-Notebook-PC:/home# ls -l
итого 24
drwx----- 2 root root 16384 окт 15 19:18 lost+found
drwxr-xr-x 15 maks maks 4096 окт 15 19:48 maks
drwxr-xr-x 2 user user 4096 окт 15 20:43 user
root@maks-HP-Pavilion-g7-Notebook-PC:/home#
```

Рисунок 5 – Создание пользователя

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



```
root@maks-HP-Pavilion-g7-Notebook-PC: /home
root@maks-HP-Pavilion-g7-Notebook-PC:/# cd user
root@maks-HP-Pavilion-g7-Notebook-PC:/home/user# touch 1.txt
root@maks-HP-Pavilion-g7-Notebook-PC:/home/user# cat > 2.txt
^C
root@maks-HP-Pavilion-g7-Notebook-PC:/home/user# nano
root@maks-HP-Pavilion-g7-Notebook-PC:/home/user# ls -l
итого 0
-rw-r--r-- 1 root root 0 окт 15 22:06 1.txt
-rw-r--r-- 1 root root 0 окт 15 22:06 2.txt
-rw-r--r-- 1 root root 0 окт 15 22:07 3.txt
root@maks-HP-Pavilion-g7-Notebook-PC:/home/user#
```

Рисунок 6 – Создание файлов

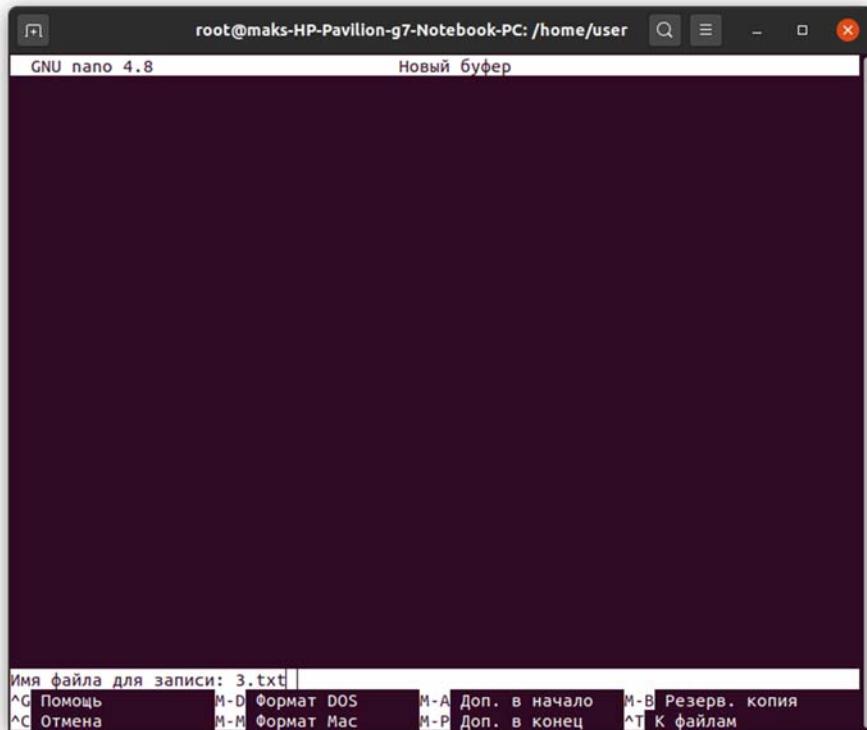


Рисунок 7 – Создание файла с помощью текстового редактора nano

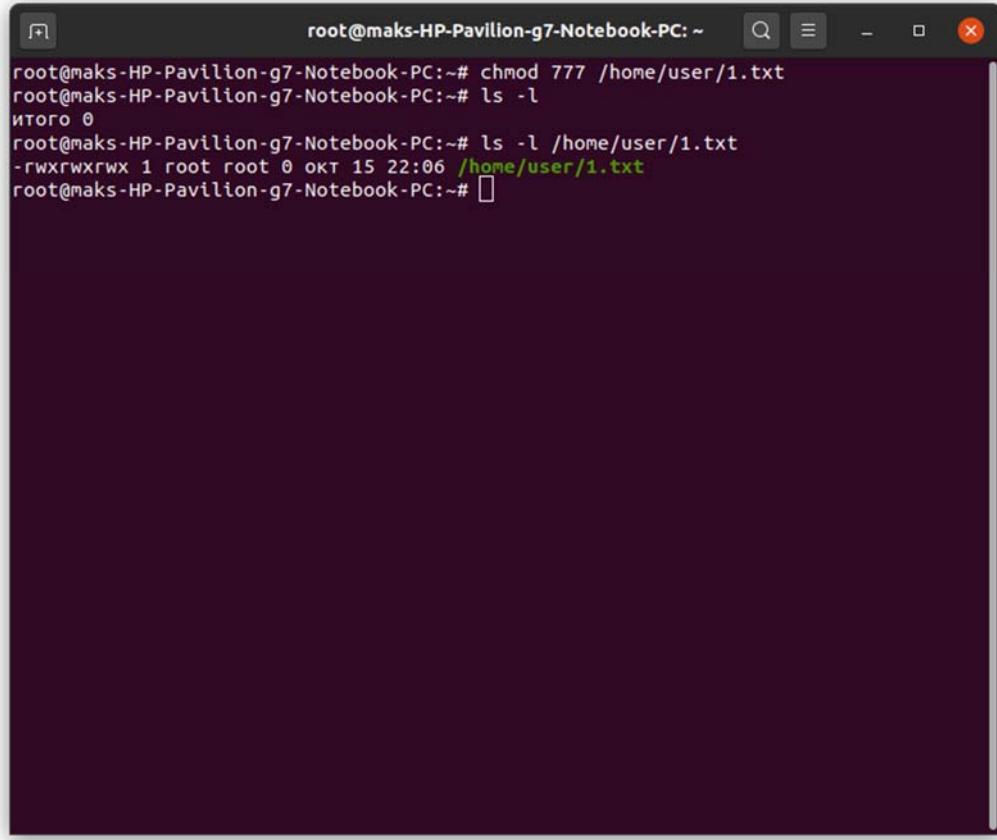
Владельцем файлов является пользователь root, он имеет полные права на файлы, остальные пользователи имеют только право на чтение.

После этого перейдём в директорию /root:

```
root@maks-HP-Pavilion-g7-Notebook-PC:~/dev# cd root
bash: cd: root: Нет такого файла или каталога
root@maks-HP-Pavilion-g7-Notebook-PC:~/dev# cd /
root@maks-HP-Pavilion-g7-Notebook-PC:/# cd root
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -a
. . . .bash_history .bashrc .cache .profile
root@maks-HP-Pavilion-g7-Notebook-PC:~# 
```

Рисунок 8 – Каталог root

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



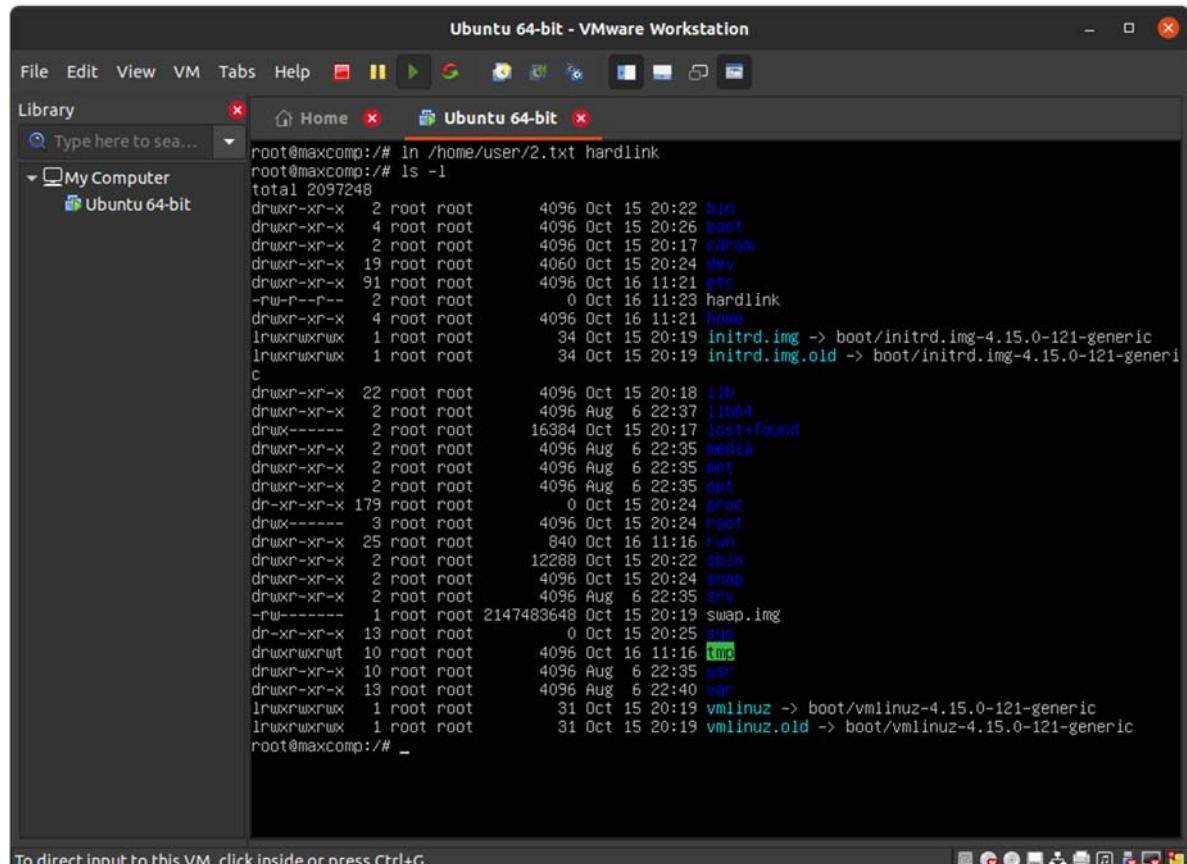
The screenshot shows a terminal window with a dark background and light-colored text. At the top, it says "root@maks-HP-Pavilion-g7-Notebook-PC: ~". Below that, the user runs the command "chmod 777 /home/user/1.txt", which changes the permissions of the file to be readable, writable, and executable by all users. Then, the user runs "ls -l" to list the file's details, showing it has full permissions (rwxrwxrwx) and was created by root on October 15 at 22:06. The file path "/home/user/1.txt" is highlighted in green.

```
root@maks-HP-Pavilion-g7-Notebook-PC:~# chmod 777 /home/user/1.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l
итого 0
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/1.txt
-rwxrwxrwx 1 root root 0 окт 15 22:06 /home/user/1.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

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.

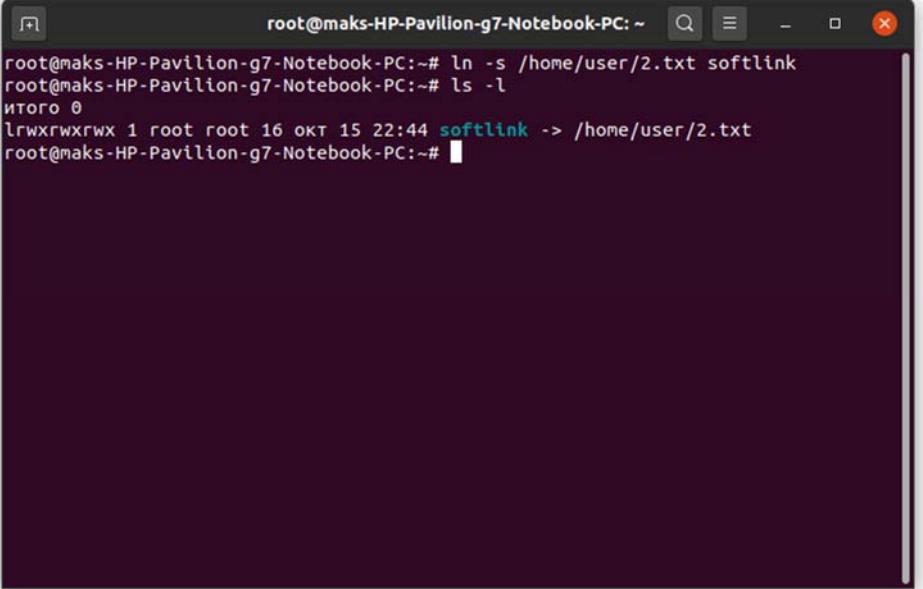
Далее создадим жёсткую и символьическую ссылки на файл 2.txt. При создании символьической ссылки проблем не возникло, но для создания жёсткой ссылки пришлось снова воспользоваться Linux Ubuntu Server 18.04, установленной в виртуальной машине VMware, так как при попытке создания жёсткой ссылки на файл в Linux Ubuntu происходит ошибка создания ссылки между устройствами. Причиной этой ошибки могла стать неверная разметка диска при установке. Приступим к созданию ссылок:



The screenshot shows a terminal window titled "Ubuntu 64-bit - VMware Workstation". The terminal is running as root on a host named "maxcomp". The user has run the command "ln /home/user/2.txt hardlink". The output of the command is displayed, showing the creation of a hard link named "hardlink" in the current directory. The terminal also lists various system files and directories like bin, boot, cdrom, dev, etc, media, proc, root, run, sbin, snap, srv, swap.img, tmp, usr, and var.

```
root@maxcomp:/# ln /home/user/2.txt hardlink
root@maxcomp:/# ls -l
total 2097248
drwxr-xr-x  2 root root      4096 Oct 15 20:22 bin
drwxr-xr-x  4 root root      4096 Oct 15 20:26 boot
drwxr-xr-x  2 root root      4096 Oct 15 20:17 cdrom
drwxr-xr-x 19 root root     4060 Oct 15 20:24 dev
drwxr-xr-x 91 root root     4096 Oct 16 11:21 etc
-rw-r--r--  2 root root       0 Oct 16 11:23 hardlink
drwxr-xr-x  4 root root     4096 Oct 16 11:21 home
lrwxrwxrwx  1 root root     34 Oct 15 20:19 initrd.img -> boot/initrd.img-4.15.0-121-generic
lrwxrwxrwx  1 root root     34 Oct 15 20:19 initrd.img.old -> boot/initrd.img-4.15.0-121-generic
c
drwxr-xr-x 22 root root     4096 Oct 15 20:18 lib
drwxr-xr-x  2 root root     4096 Aug  6 22:37 lib64
drwx-----  2 root root    16384 Oct 15 20:17 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 179 root root      0 Oct 15 20:24 proc
drwx-----  3 root root     4096 Oct 15 20:24 root
drwxr-xr-x  25 root root     840 Oct 16 11:16 run
drwxr-xr-x  2 root root   12288 Oct 15 20:22 sbin
drwxr-xr-x  2 root root     4096 Oct 15 20:24 snap
drwxr-xr-x  2 root root     4096 Aug  6 22:35 srv
-rw-----  1 root root  2147483648 Oct 15 20:19 swap.img
dr-xr-xr-x 13 root root      0 Oct 15 20:25 sys
drwxrwxrwt 10 root root     4096 Oct 16 11:16 tmp
drwxr-xr-x 10 root root     4096 Aug  6 22:35 usr
drwxr-xr-x 13 root root     4096 Aug  6 22:40 var
lrwxrwxrwx  1 root root     31 Oct 15 20:19 vmlinuz -> boot/vmlinuz-4.15.0-121-generic
lrwxrwxrwx  1 root root     31 Oct 15 20:19 vmlinuz.old -> boot/vmlinuz-4.15.0-121-generic
root@maxcomp:/#
```

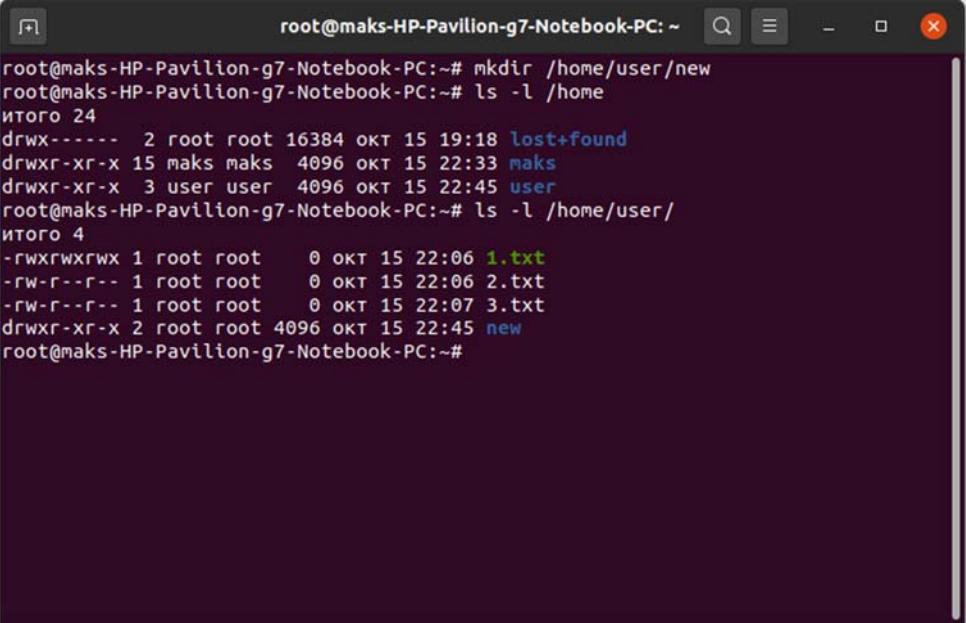
Рисунок 10 – Создание жёсткой ссылки



```
root@maks-HP-Pavilion-g7-Notebook-PC:~# ln -s /home/user/2.txt softlink
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l
итого 0
lrwxrwxrwx 1 root root 16 окт 15 22:44 softlink -> /home/user/2.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

Рисунок 11 – Создание символьической ссылки

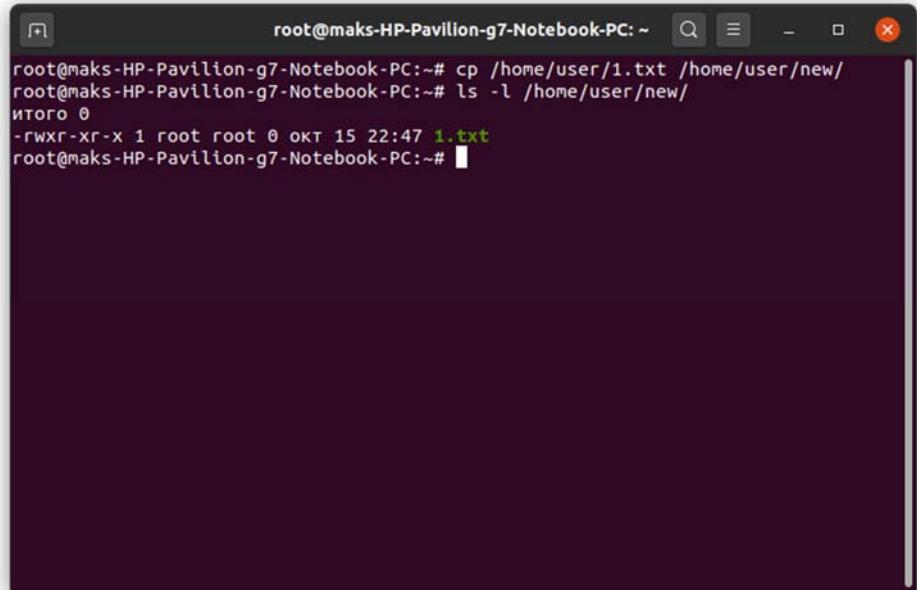
После этого требуется создать новую директорию new в каталоге пользователя user. Для этого используем команду mkdir:



```
root@maks-HP-Pavilion-g7-Notebook-PC:~# mkdir /home/user/new
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home
итого 24
drwx----- 2 root root 16384 окт 15 19:18 lost+found
drwxr-xr-x 15 maks maks 4096 окт 15 22:33 maks
drwxr-xr-x  3 user user 4096 окт 15 22:45 user
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/
итого 4
-rw-rw-rwx 1 root root    0 окт 15 22:06 1.txt
-rw-r--r-- 1 root root    0 окт 15 22:06 2.txt
-rw-r--r-- 1 root root    0 окт 15 22:07 3.txt
drwxr-xr-x 2 root root 4096 окт 15 22:45 new
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

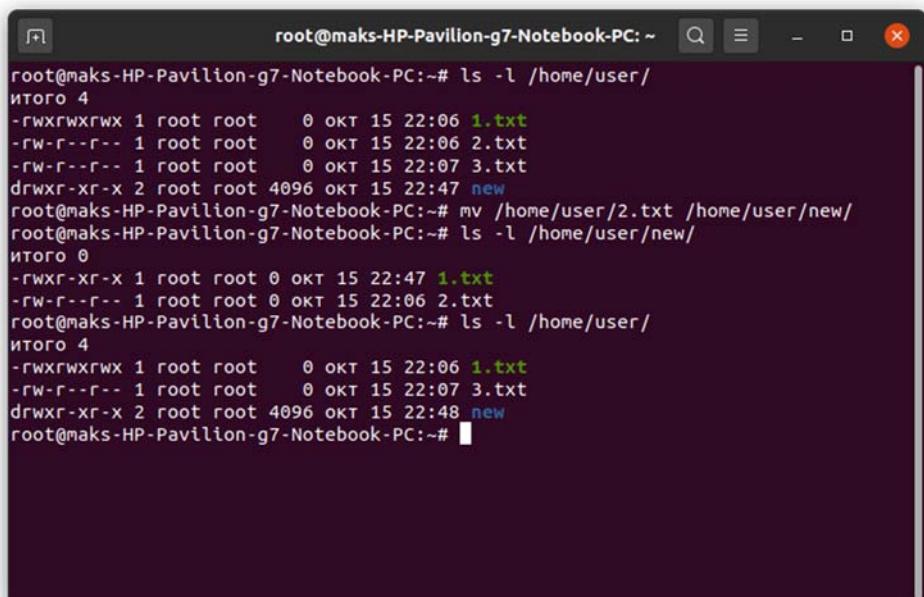
Рисунок 12 – Создание каталога в директории пользователя

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



```
root@maks-HP-Pavilion-g7-Notebook-PC:~# cp /home/user/1.txt /home/user/new/
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/new/
итого 0
-rwxr-xr-x 1 root root 0 окт 15 22:47 1.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

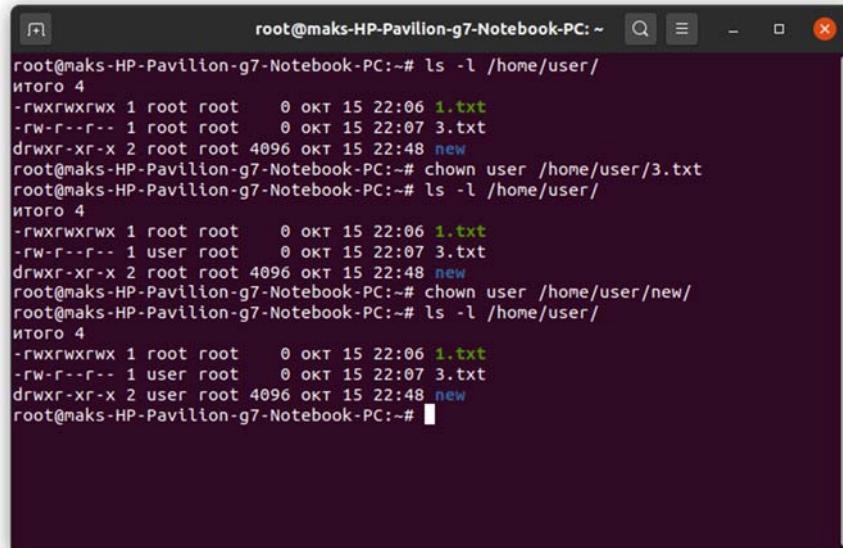
Рисунок 13 – Копирование файла



```
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/
итого 4
-rwxrwxrwx 1 root root    0 окт 15 22:06 1.txt
-rw-r--r-- 1 root root    0 окт 15 22:06 2.txt
-rw-r--r-- 1 root root    0 окт 15 22:07 3.txt
drwxr-xr-x 2 root root 4096 окт 15 22:47 new
root@maks-HP-Pavilion-g7-Notebook-PC:~# mv /home/user/2.txt /home/user/new/
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/new/
итого 0
-rwxr-xr-x 1 root root 0 окт 15 22:47 1.txt
-rw-r--r-- 1 root root 0 окт 15 22:06 2.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/
итого 4
-rwxrwxrwx 1 root root    0 окт 15 22:06 1.txt
-rw-r--r-- 1 root root    0 окт 15 22:07 3.txt
drwxr-xr-x 2 root root 4096 окт 15 22:48 new
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

Рисунок 14 – Перемещение файла

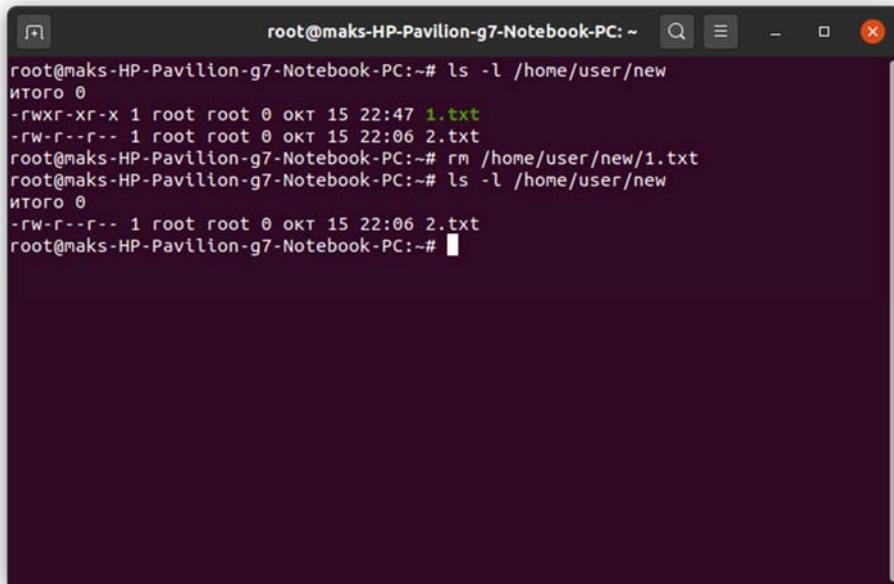
После этого поменяем владельцев файла 3.txt и каталога new. Сделаем это с помощью команды chown:



```
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/
итого 4
-rwxrwxrwx 1 root root    0 окт 15 22:06 1.txt
-rw-r--r-- 1 root root    0 окт 15 22:07 3.txt
drwxr-xr-x 2 root root 4096 окт 15 22:48 new
root@maks-HP-Pavilion-g7-Notebook-PC:~# chown user /home/user/3.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/
итого 4
-rwxrwxrwx 1 root root    0 окт 15 22:06 1.txt
-rw-r--r-- 1 user root    0 окт 15 22:07 3.txt
drwxr-xr-x 2 root root 4096 окт 15 22:48 new
root@maks-HP-Pavilion-g7-Notebook-PC:~# chown user /home/user/new
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/
итого 4
-rwxrwxrwx 1 root root    0 окт 15 22:06 1.txt
-rw-r--r-- 1 user root    0 окт 15 22:07 3.txt
drwxr-xr-x 2 user root 4096 окт 15 22:48 new
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

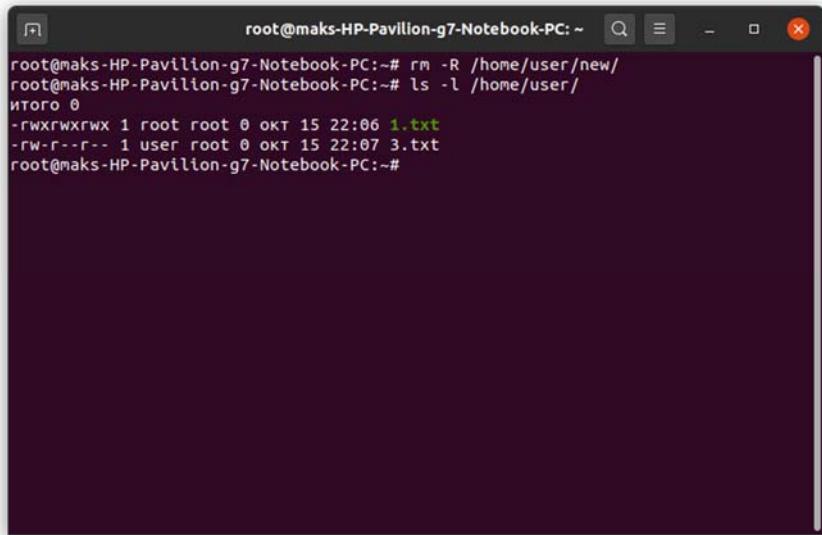
Рисунок 15 – Изменение владельца файла и каталога

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



```
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/new
итого 0
-rwxr-xr-x 1 root root 0 окт 15 22:47 1.txt
-rw-r--r-- 1 root root 0 окт 15 22:06 2.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~# rm /home/user/new/1.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/new
итого 0
-rw-r--r-- 1 root root 0 окт 15 22:06 2.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

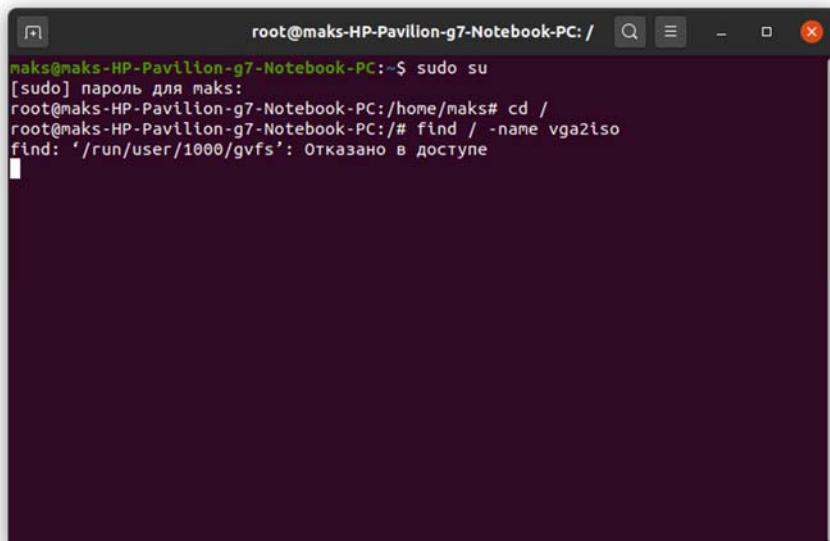
Рисунок 16 – Удаление файла



```
root@maks-HP-Pavilion-g7-Notebook-PC:~# rm -R /home/user/new/
root@maks-HP-Pavilion-g7-Notebook-PC:~# ls -l /home/user/
итого 0
-rwxrwxrwx 1 root root 0 окт 15 22:06 1.txt
-rw-r--r-- 1 user root 0 окт 15 22:07 3.txt
root@maks-HP-Pavilion-g7-Notebook-PC:~#
```

Рисунок 17 – Удаление каталога

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



```
maks@maks-HP-Pavilion-g7-Notebook-PC:~$ sudo su
[sudo] пароль для maks:
root@maks-HP-Pavilion-g7-Notebook-PC:/home/maks# cd /
root@maks-HP-Pavilion-g7-Notebook-PC:/# find / -name vga2iso
find: '/run/user/1000/gvfs': Отказано в доступе
```

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 и, 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.