

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра Автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине «Операционная система Linux»

Работа с файловой системой ОС Linux

Студент

Жидков И.А.

Группа АС-19

Руководитель

Кургасов В.В.

к.п.н.

Липецк 2020 г.

Оглавление

Цель работы.....	3
Задание кафедры.....	4
Ход работы.....	5
Вывод.....	18
Ответы на контрольные вопросы.....	19

Цель работы

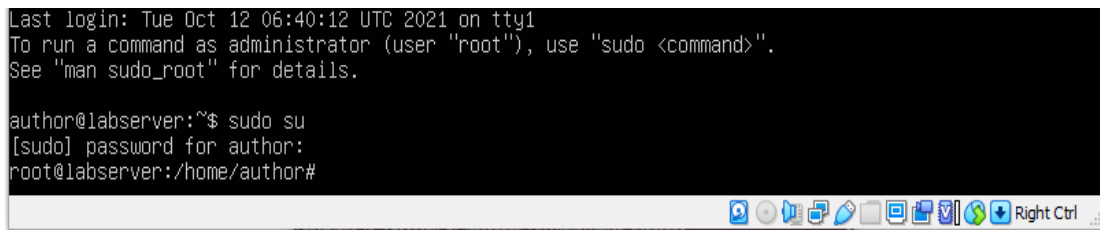
Приобрести опыт работы с файлами и каталогами в ОС Linux, настройки прав на доступ к файлам и каталогам.

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

Начнём работу с того, что запустим виртуальную машину и загрузимся пользователем root с помощью команды `sudo su`:



```
Last login: Tue Oct 12 06:40:12 UTC 2021 on tty1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

author@labserver:~$ sudo su
[sudo] password for author:
root@labserver:/home/author#
```

Рисунок 1 – Загрузка пользователем root

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



```
root@labserver:~# cd /
root@labserver:/# ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  swap.img  tmp  var
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
```

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. 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);
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;
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 съёмных 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 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 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 – 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@labserver:/# ls dev
autofs          hugepages      nvram          tty            tty29          tty5           ttyS11         ttyS4          vcsa2
block           hwrng          port           tty0           tty3           tty50          ttyS12         ttyS5          vcsa3
bsg             i2c-0          ppp            tty1           tty30          tty51          ttyS13         ttyS6          vcsa4
btrfs-control   initctl        psaux          tty10          tty31          tty52          ttyS14         ttyS7          vcsa5
bus             input          ptmx           tty11          tty32          tty53          ttyS15         ttyS8          vcsa6
cdrom           kmsg           pts            tty12          tty33          tty54          ttyS16         ttyS9          vcsu
char            lightnvm       random          tty13          tty34          tty55          ttyS17         ubuntu-vg      vcsu1
console         log            rfkill         tty14          tty35          tty56          ttyS18         udmabuf       vcsu2
core            loop0          rtc            tty15          tty36          tty57          ttyS19         uhid          vcsu3
cpu             loop1          rtc0           tty16          tty37          tty58          ttyS2           uinput       vcsu4
cpu_dma_latency loop2          sda            tty17          tty38          tty59          ttyS20         urandom      vcsu5
cuse            loop3          sda1           tty18          tty39          tty6           ttyS21         userio       vcsu6
disk            loop4          sda2           tty19          tty4           tty60          ttyS22         vboxguest    vfio
dm-0            loop5          sda3           tty2           tty40          tty61          ttyS23         vboxuser     vga_arbiter
dri             loop6          sg0            tty20          tty41          tty62          ttyS24         vcs          vhci
dvd             loop7          sg1            tty21          tty42          tty63          ttyS25         vcs1         vhost-net
ecryptfs        loop-control   shm            tty22          tty43          tty7           ttyS26         vcs2         vhost-vsock
fb0             mapper         snapshot       tty23          tty44          tty8           ttyS27         vcs3         zero
fd              mcelog         snd            tty24          tty45          tty9           ttyS28         vcs4         zfs
full            mem            sr0            tty25          tty46          ttyprintk      ttyS29         vcs5
fuse            mqqueue        stderr         tty26          tty47          ttyS0          ttyS3           vcs6
hidraw0         net            stdin          tty27          tty48          ttyS1          ttyS30          vcsa
hpet            null           stdout         tty28          tty49          ttyS10         ttyS31          vcsa1
```

Рисунок 3 – Каталог файлов физических устройств

Опишем эти файлы:

1. autofs – цель autofs - обеспечить монтирование по требованию и автоматическое размонтирование других файловых систем;
2. 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;
3. console – текстовый терминал и виртуальные консоли;
4. cpu_dma_latency – часть интерфейса качества и обслуживания в ядре Linux;

5. `cuse` – символьные устройства в пространстве пользователя;
6. `dm-0` – часть устройства отображения в ядре, используемого LVM;
7. `ecryptfs` – POSIX-совместимая промышленного уровня файловая система многоуровневого шифрования для Linux;
8. `fb0` – устройство обеспечивает абстракцию для графического оборудования;
9. `full` – специальный файл, представляющий собой «полное устройство». Запись в него ненулевого количества байт происходит с ошибкой «недостаточно места»;
10. `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);
11. `hpet` – таймер событий высокой точности (HPET) - это аппаратный таймер, используемый в персональных компьютерах;
12. `hwrng` – генератор случайных чисел;
13. `i2c-0` – последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов;
14. `kmsg` – узел символьного устройства обеспечивает доступ пользователя к буферу `printk` ядра;
15. `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;
16. `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;

17. `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;

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

19. `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);

20. `nvgm` – обеспечивает доступ к конфигурации BIOS NVRAM в системах `i386` и `amd64`;

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

22. `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;

23. `psaux` – устройство мыши PS / 2;

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

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

26. `rftkill` – предоставляет общий интерфейс для отключения любого радиопередатчика в системе;

27. `rtc` – часы реального времени;

28. `sda` – первый жесткий диск;

29. `sg` – SCSI Generic driver используется, среди прочего, для сканеров, устройств записи компакт-дисков и чтения аудио-компакт-дисков в цифровом формате;

30. `snapshot` – поддержка снимков устройства;

31. `tty` – виртуальная консоль;

32. `ttyprintk` – драйвер псевдо ТТУ, который позволяет пользователям создавать сообщения `printk` через вывод на устройство `ttyprintk`;

33. `uhid` – поддержка драйвера ввода-вывода пользовательского пространства для подсистемы HID;

34. `uinput` – поддержка драйвера уровня пользователя для ввода;

35. `urandom` – более быстрая и менее безопасная генерация случайных чисел;

36. `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;

37. `vcs` – текущее текстовое содержимое виртуальной консоли;

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

39. `vcsu` – текущее текстовое содержимое виртуальной консоли (юникод);

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

41. `vhci` – виртуальный драйвер HCI Bluetooth;

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

43. `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;

44. `zero` – источник нулевого байта;

45. `zfs` – настраивает пулы хранения ZFS.

Следующим шагом перейдём в директорию `/root`. В этом каталоге должен располагаться файл `vmlinuz`.

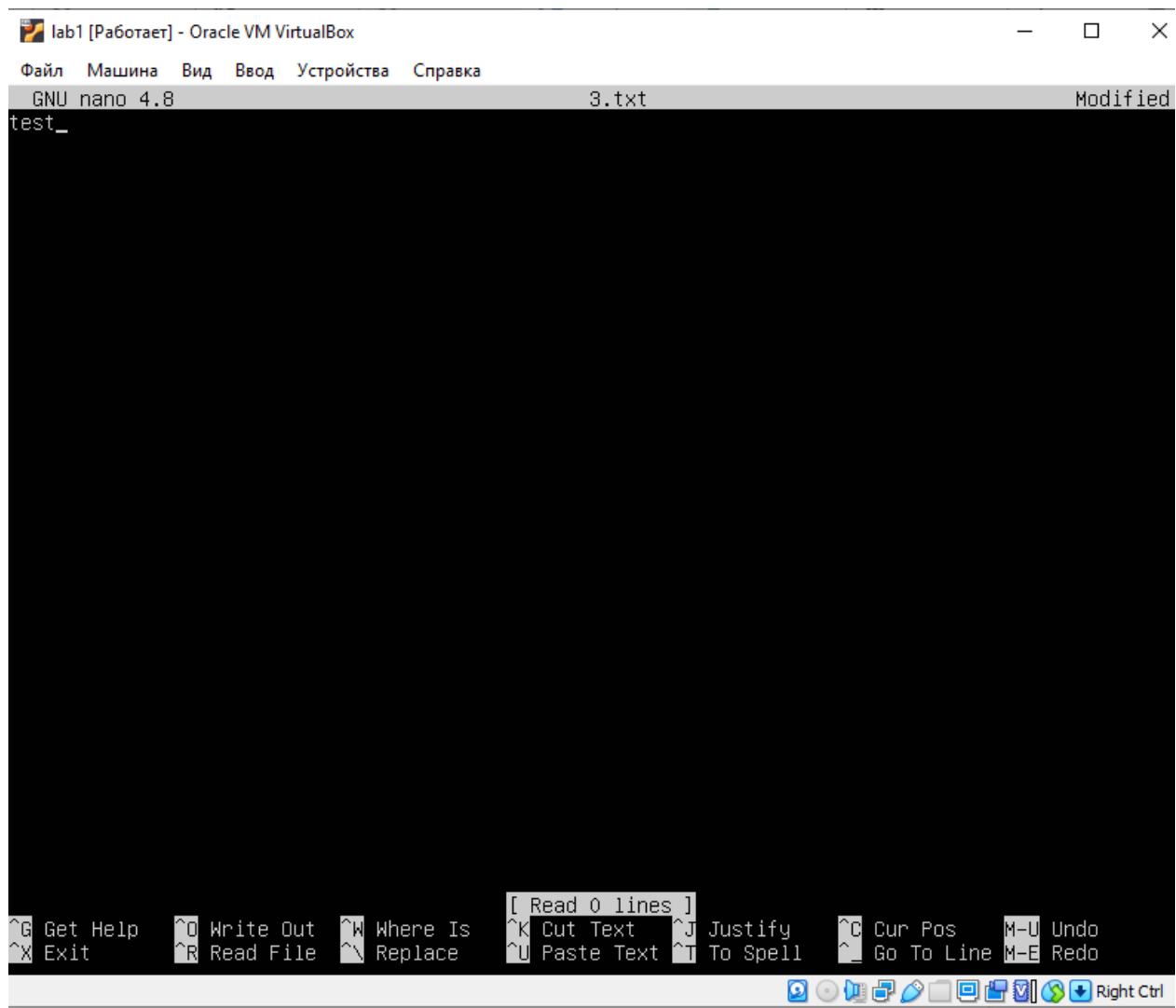


Рисунок 7 – Создание файла с помощью текстового редактора nano

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@labserver:~# cd root
bash: cd: root: No such file or directory
root@labserver:~# cd /
root@labserver:/# cd root
root@labserver:~# ls -a
.  ..  .bashrc  .local  .profile  snap  .ssh
root@labserver:~#
```

Рисунок 8 – Каталог root

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

```
root@labserver:~# chmod 777 /home/new_user/1.txt
root@labserver:~# ls /home/new_user
1.txt 2.txt 3.txt
root@labserver:~#
```

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`.

```
root@labserver:/home/new_user# cat 2.txt
hello it's me
root@labserver:/home/new_user# ln 2.txt hardlink
root@labserver:/home/new_user# ls
1.txt 2.txt 3.txt hardlink
root@labserver:/home/new_user# cat hardlink
hello it's me
root@labserver:/home/new_user#
```

Рисунок 10 – Создание жёсткой ссылки

```
root@labserver:/home/new_user# ln -s 2.txt softlink
root@labserver:/home/new_user# ls -l
total 8
-rwxrwxrwx 1 root root 0 окт 14 12:36 1.txt
-rw-r--r-- 2 root root 14 окт 14 12:45 2.txt
-rw-r--r-- 1 root root 0 окт 14 12:35 3.txt
-rw-r--r-- 2 root root 14 окт 14 12:45 hardlink
lrwxrwxrwx 1 root root 5 окт 14 12:49 softlink -> 2.txt
root@labserver:/home/new_user# _
```

Рисунок 11 – Создание символической ссылки

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

```
root@labserver:/home/new_user# mkdir new
root@labserver:/home/new_user# ls
1.txt 2.txt 3.txt hardlink new softlink
root@labserver:/home/new_user# _
```

Рисунок 12 – Создание каталога в директории пользователя

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

```

root@labserver:/home/new_user# cp 1.txt new
root@labserver:/home/new_user# ls
1.txt 2.txt 3.txt hardlink new softlink
root@labserver:/home/new_user# ls new
1.txt
root@labserver:/home/new_user# _

```

Рисунок 13 – Копирование файла

```

root@labserver:/home/new_user# mv 2.txt new
root@labserver:/home/new_user# ls
1.txt 3.txt hardlink new softlink
root@labserver:/home/new_user# ls new
1.txt 2.txt
root@labserver:/home/new_user# _

```

Рисунок 14 – Перемещение файла

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

```

root@labserver:/home/new_user# chown new_user 3.txt
root@labserver:/home/new_user# chown new_user 2.txt
chown: cannot access '2.txt': No such file or directory
root@labserver:/home/new_user# chown new_user new/2.txt
root@labserver:/home/new_user# ls
1.txt 3.txt hardlink new softlink
root@labserver:/home/new_user# _

```

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@labserver:/home/new_user# rm 1.txt
root@labserver:/home/new_user# rm new
rm: cannot remove 'new': Is a directory
root@labserver:/home/new_user# ls
3.txt hardlink new softlink
root@labserver:/home/new_user# _

```

Рисунок 16 – Удаление файла

```

root@labserver:/home/new_user# rm -R new
root@labserver:/home/new_user# ls
3.txt hardlink softlink
root@labserver:/home/new_user#

```

Рисунок 17 – Удаление каталога

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


```
root@labserver:/# find -name vga2iso
root@labserver:/# _
```

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.

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`

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`.