

Липецкий государственный технический университет

Кафедра прикладной математики

Отчет по лабораторной работе № 2
«Работа с файловой системой ОС Linux»
по курсу «Операционная система Linux»

Студент

подпись, дата

Богомолов Е.А.
фамилия, инициалы

Группа

Руководитель

Доцент, к. пед. наук
ученая степень, ученое звание

подпись, дата

Кургасов В.В.
фамилия, инициалы

Липецк 2021 г.

Содержание

Цель работы	3
Задание кафедры	4
1. Ход работы	6
1.1. Запуск виртуальной машины с Linux Ubuntu и загрузка пользователем root	6
1.2. Перечень каталогов с указанием их назначения	7
1.3. Содержимое каталога физических устройств и перечень файлов физических устройств с указанием их назначения	11
1.4. Содержимое каталога «boot» и файла «vmlinuz»	14
1.5. Создание нового пользователя «user»	15
1.6. Создание в директории пользователя «user» трех файлов и пояснения прав доступа к файлам.	16
1.7. Переход в директорию пользователя «root».	17
1.8. Изменение прав доступа на файл «1.txt» в директории пользователя «user»	18
1.9. Создание жесткой и символической ссылки на файл «2.txt»	19
1.10. Создание каталога «new» в каталоге пользователей «user»	20
1.11. Копирование файла «1.txt» в каталог «new»	21
1.12. Перемещение файла «2.txt» в каталог «new»	22
1.13. Изменение владельца файла «3.txt» и каталога «new»	23
1.14. Удаление файла «1.txt» в каталоге «new»	24
1.15. Удаление каталога «new»	25
1.16. Поиск файла «vga2iso» с помощью команды «find»	26
Выводы	27
Контрольные вопросы	28

Цель работы

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

Задание кафедры

Необходимо:

1. Запустить виртуальную машину с 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 boot. 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.

1. Ход работы

1.1. Запуск виртуальной машины с Linux Ubuntu и загрузка пользователем root

Запуск виртуальной машины с Linux Ubuntu и загрузку пользователем root с помощью команды «sudo su» можно увидеть на рисунке 1.

```
Ubuntu 20.04.3 LTS ubuntu-server tty1
ubuntu-server login: ubuntu-user
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu 14 Oct 2021 07:03:24 PM UTC

System load:  0.54           Processes:            107
Usage of /:   46.1% of 8.79GB Users logged in:          0
Memory usage: 18%           IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%

27 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Thu Oct 14 18:51:29 UTC 2021 on tty1
sudubuntu-user@ubuntu-server:~$ sudo su
[sudo] password for ubuntu-user:
root@ubuntu-server:/home/ubuntu-user# _
```

Рисунок 1 – Загрузка пользователем root.

1.2. Перечень каталогов с указанием их назначения

Просмотрим содержимое корневого диалога с помощью команды «ls» и опишем каждый из полученных каталогов. Результат выполнения представлен на рисунке 2.

```
root@ubuntu-server:/home/ubuntu-user# cd /
root@ubuntu-server:/# 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 – Перечень каталогов

Перечень каталогов

- **/bin** содержит команды, которые могут использоваться как системным администратором, так и пользователями, но которые требуются, когда другие файловые системы не подключены (например, в однопользовательском режиме). Он также может содержать команды, которые косвенно используются скриптами.
- **/boot** – этот каталог содержит все необходимое для процесса загрузки, за исключением файлов конфигурации, которые не нужны во время загрузки, и установщика карты. Таким образом, **/boot** хранит данные, которые используются до того, как ядро начнет выполнять программы пользовательского режима. Это может включать сохраненные основные загрузочные сектора и файлы карт секторов. Программы, необходимые для того, чтобы загрузчик мог загружать файл, должны быть размещены в **/sbin**. Файлы конфигурации для загрузчиков, которые не требуются во время загрузки, должны быть помещены в **/etc**.
- **/cdrom** – это временное место, где монтируются диски CD-ROM, когда они вставляются в компьютер. Однако, стандартное место для подключаемого носителя находится внутри каталога **/media**;
- **/dev** – это расположение специальных файлов или файлов устройств.
- **/etc** содержит файлы конфигурации. "Файл конфигурации это локальный файл, используемый для управления работой программы;

он должен быть статическим и не может быть исполняемым двоичным файлом.

- **/home** – личный каталог пользователя в операционной системе, где находятся его данные, настройки и т. д.
- **/lib** содержит образы общих библиотек, необходимые для загрузки системы и выполнения команд в корневой файловой системе, т. Е.. двоичными файлами в **/bin** и **/sbin**.
- **/lib32, /lib64, libx32** – альтернативный формат необходимых общих библиотек.
- **/lost+found** – в каждой файловой системе Linux есть каталог **lost+found**. Если произошел сбой в работе файловой системы, то при следующей загрузке будет выполнена проверка файловой системы. Все найденные поврежденные файлы будут размещены в каталоге **lost+found**.
- **/media** – этот каталог содержит подкаталоги, которые используются в качестве точек подключения для съемных носителей, таких как дискеты, компакт-диски и zip-диски.
- **/mnt** – этот каталог предоставляется для того, чтобы системный администратор мог временно смонтировать файловую систему по мере необходимости. Содержимое этого каталога является локальной проблемой и не должно влиять на способ запуска любой программы. Этот каталог не должен использоваться программами установки: вместо него должен использоваться подходящий временный каталог, не используемый системой.
- **/opt** зарезервирован для установки дополнительных пакетов прикладного программного обеспечения. Пакет, устанавливаемый в **/opt**, должен размещать свои статические файлы в отдельном **/opt/<пакете>** или / дерево каталогов **opt/<поставщик>**, где **<пакет>** - это имя, описывающее пакет программного обеспечения, а **<поставщик>** - зарегистрированное имя поставщика в локальной сети.

- **/proc** – это де-факто стандартный метод Linux для обработки информации о процессах и системе. Рекомендуется для хранения и извлечения информации о процессах, а также другой информации о ядре и памяти.
- **/root** – домашний каталог корневой учетной записи может определяться разработчиком или локальными предпочтениями, но это рекомендуемое расположение по умолчанию.
- **/run** – этот каталог содержит данные о системе, описывающие систему с момента ее загрузки. Файлы в этом каталоге должны быть очищены (удалены или усечены соответствующим образом) в начале процесса загрузки.
- **/sbin** –здесь системой Linux помещаются команды, относящиеся к обслуживанию файловой системы и управлению запуском системы.
- **/span** – по умолчанию является местом, где файлы и папки из установленных пакетов `snar` появляются в вашей системе.
- **/srv** –в каталоге находятся данные для сервисов, предоставляемых системой.
- **/sys** – это расположение, в котором представлена информация об устройствах, драйверах и некоторых функциях ядра. Его базовая структура определяется конкретным ядром Linux, используемым в данный момент, и в остальном не определена.
- **/tmp** – каталог должен быть доступен для программ, которым требуются временные файлы. Программы не должны предполагать, что какие-либо файлы или каталоги в `/tmp` сохраняются между вызовами программы.
- **/usr** – это второй основной раздел файловой системы. `/usr` - это общедоступные данные, доступные только для чтения. Это означает, что `/usr` должен быть доступен для совместного использования между различными хостами, совместимыми с FHS, и не должен записываться на них. Любая информация, относящаяся к конкретному хосту

или изменяющаяся со временем, хранится в другом месте. Большие пакеты программного обеспечения не должны использовать прямой подкаталог в иерархии /usr.

- /**var** содержит файлы переменных данных. Это включает каталоги и файлы буфера, административные данные и данные журнала, а также временные и временные файлы.

1.3. Содержимое каталога физических устройств и перечень файлов физических устройств с указанием их назначения

Просмотрим содержимое каталога физических устройств («/dev») с помощью команды «ls». Результат выполнения команды представлен на рисунке 3.

```
root@ubuntu-server:/# cd dev
root@ubuntu-server:/dev# ls
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          pis           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           mqueue        stderr        tty26          tty47          ttyS0          ttyS3          vcs6
hidraw0        net           stdin         tty27          tty48          ttyS1          ttyS30         vcsa
hpet           null          stdout        tty28          tty49          ttyS10         ttyS31         vcsa1
```

Рисунок 3 – Содержимое каталога файлов физических устройств
перечень файлов физических устройств:

- autofs – даёт возможность автоматического монтирования съёмных носителей и сетевых ресурсов при вставке или обращении к ним.
- block – блочные устройства.
- bsg – альтернативный сквозной драйвер SCSI.
- btrfs-control – файловая система.
- bus – внешние носители.
- cdrom – дисковод.
- char – символьное устройство.

- `console` – системная консоль, т.е. физически подключенные клавиатура и монитор.
- `core` – уменьшенный вариант крошечного ядра без графического рабочего стола, хотя могут быть добавлены дополнительные расширения для создания системы с графической средой рабочего стола.
- `cpu` – выполняет все виды операций по обработке данных, и его часто называют мозгом компьютера.
- `_dma_latency` – используется для сбора показателей задержки для однонаправленного и двунаправленного трафика.
- `cuse` – открывается программой, которая хочет реализовать определенное символьное устройство. CUSE определяет, какое устройство он реализует, а затем создает узел устройства.
- `disk` – предоставляет дополнительную информацию о разделах в вашей системе.
- `dm-o` – блочное устройство, которое всегда возвращает нулевые данные при чтении и беззвучно удаляет записи. Пользователь может записывать данные в любом месте разреженного устройства и считывать их обратно, как обычное устройство.
- `dri` – это платформа, обеспечивающая безопасный и эффективный прямой доступ к графическому оборудованию в системе X Window. `dvd` – цифровой видеодиск.
- `ecryptfs` – это пакет программного обеспечения для шифрования дисков для Linux. `fbo` – это расширение OpenGL для гибкого рендеринга за пределами экрана, включая рендеринг в текстуру.
- `fd` – файл дисководов для гибких дисков. `full` – создает непрерывный поток нулевых символов как выводится при чтении из и генерирует ошибку `ENOSPC` («диск заполнен») при попытке записи в него. `fuse` – это простой интерфейс для программ пользовательского пространства для экспорта виртуальной файловой системы в ядро Linux.

- hidraw0 – предоставляет необработанный интерфейс для устройств USB и Bluetooth с человеческим интерфейсом (HIDS). hpet – таймер событий высокой точности.

1.4. Содержимое каталога «boot» и файла «vmlinuz»

Перейдем к каталогу «boot» с помощью команды «cd» и посмотрим его содержимое с помощью «ls». Результат выполнения представлен на рисунке 4.

```
root@ubuntu-server:/# cd boot
root@ubuntu-server:/boot# ls
config-5.4.0-88-generic  initrd.img-5.4.0-88-generic  System.map-5.4.0-88-generic  vmlinuz.old
grub                    initrd.img                   vmlinuz
initrd.img              lost+found                  vmlinuz-5.4.0-88-generic
root@ubuntu-server:/boot# _
```

Рисунок 4 – Содержимое каталога файлов физических устройств

Данный файл содержит сжатое ядро Linux – программу, которая запускается в первую очередь при запуске операционной системы, и остается в памяти пока компьютер не будет выключен.

Посмотрим содержимое файла «vmlinuz» с помощью команды «cat». Результат выполнения представлен на рисунке 5.

```
0I+++
0F@r^&F`+5+S"~C+c0+2++
+^+ Eб"0A+A+K+`L9+tt+90++A+щ+tt+
[ ,1++5+*?+ap++fz0+?+z@H+E+?+?+%\<++++N++++
++{!+{+}*+{ ++E+{++2+(+?U++:++[++++p++H++R9++++2E+tt+e++++@++<E++MA+AA++++RQ+D      +++
+++83++++H+
Afh++++>+H++++4+++@+ E+++@+G@++$+z|+\\+} ++J#+[+07++++[1++++=+++++|++z0++y++b+I++++=+I++0+B+I++wr@_h++f4+H+++
Afh++++>+H++++4+++@+ E+++@+G@++$+z|+\\+} ++J#+[+07++++[1++++=+++++|++z0++y++b+I++++=+I++0+B+I++wr@_h++f4+H+++
`MD+MD+IDASID+ARHE+V+H++R+?+QLE+I++++|!+RM _f ++}#++z1(+LL+e+})(++{1IAπ'"÷üe=[o
cêö'+ÿÉâü35ôÜð$≥4ç%Ü≥7èiç||ë=Liô!É'±I±I rÿÿê++[0[0K0++++488(41++++V+A++73=7xDF++++'ozpJh,?+Z+.+++++^G~
#fA      ++++/ON++++Ja$E+(3++(++I++H++6L+3,H}
@++"D+ CL++aM+I++++++<++I++3++Rg ++çÊÊMîè&ÊððAî|Hî@Hî<Éæ
ÛÂçH#èèiVimîðÃ
f<EBBNðÇ Bÿö`÷HîLî→Ã&
-cÜ!ÜÇ63%±EöIÁ13AÜæÃÃÃ!AÜÜî`DèðÈS%Âèp+A+U++++{2+B+i ++0+V+ TM+j"++Qor+s~^H{@
L-Ãâ(â&|Iâð$ÜÉÉÃeHÜRsã·Ã"è@Tÿ|ÜéÃÿ
++8;B+!+,+n%++{++++A+H++B2++++ZgF4H+%3+T UBKK"R+}I+EI+T@++[FT+T+T+TÜT"L'%"=iÃ²b&e++U+I+E jAT+}++++#+@
Z_
++++
P+
++~+4Pu
I++E}++t++
++{+1++E+L+;+++++++8M+&+" F++F+I++S+Rd+3L+] ++V++Va++++ ++m+*] +Z+@++QZB+D++M+n= ++1>+ ++^2+1!++AM+8[_
+B+I++t+
++7++++ ++%+!+++++++3P+tp++<+`H+] ++
p@-+s+I++++A+N+} ++}'n+~+E+I+D+`S+=+
.U0+c+; I++++#X+!V+ ++a+m+~+%+)#G }++
#9+7+K++"/
+.tr0+ t++tF+~w+$ ++KE+A+tt.^+) +SH+D++H++V0+++
++Vg+0++8+M+D+] +I++" +P+@++=+P+D+] +H+4RH+<+L+D+ +M++++#9M+*N+"r+H+L+3 ++!++++bM+I++++ ++8+E++++ +
++8+L+~++++-9 K++h++L+KI++!+jR++I+D+`++ +C++jq+L+N A+P++*!A++p' %/k/-u+aaI+L+0'++H+++++++c0
++ +[a++E++++A++A++P+D+E+W'; +WE++g++7+++++j
0++++L+] ++#+] ++
++J+/.7+/
}+
++!++DD+E++8D++r
```

Рисунок 5 – Содержимое файла «vmlinuz»

К «vmlinuz» имеют доступ все пользователи.

1.5. Создание нового пользователя «user»

Создадим нового пользователя «user» с помощью команды «adduser». Результат выполнения представлен на рисунках 6 и 7.

```
root@ubuntu-server:/# sudo adduser user
Adding user `user' ...
Adding new group `user' (1001) ...
Adding new user `user' (1001) with group `user' ...
Creating home directory `/home/user' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user
Enter the new value, or press ENTER for the default
  Full Name []: evgeniy
   Room Number []:
   Work Phone []:
   Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Рисунок 6 – Создание нового пользователя

```
root@ubuntu-server:/# cd home
root@ubuntu-server:/home# ls
ubuntu-user  user
root@ubuntu-server:/home#
```

Рисунок 7 – Результат создания нового пользователя

1.6. Создание в директории пользователя «user» трех файлов и пояснения прав доступа к файлам.

Создадим в директории нового пользователя «user» три файла «1.txt», «2.txt», «3.txt» с помощью команд «touch», «cat», «nano». Результат выполнения команд представлен на рисунках 8 и 9.

```
root@ubuntu-server:/home/user# touch 1.txt
root@ubuntu-server:/home/user# cat > 2.txt
Hello, World!
root@ubuntu-server:/home/user# nano 3.txt_
```

Рисунок 8 – Создание файлов


```
root@ubuntu-server:/home/user# ls
1.txt 2.txt 3.txt
root@ubuntu-server:/home/user# ls -l
total 8
-rw-r--r-- 1 root root  0 Oct 14 20:34 1.txt
-rw-r--r-- 1 root root 14 Oct 14 20:34 2.txt
-rw-r--r-- 1 root root  1 Oct 14 20:34 3.txt
root@ubuntu-server:/home/user# _
```

Рисунок 9 – Результат создания файлов

Благодаря рисунку 9, можно сделать вывод, что к этому файлу полный доступ имеет только владелец (root), а остальные имеют право только на чтение.

1.7. Переход в директорию пользователя «root».

Перейдем к каталогу «root» с помощью команды «cd» и посмотрим его содержимое с помощью «ls». Результат выполнения представлен на рисунке 10.



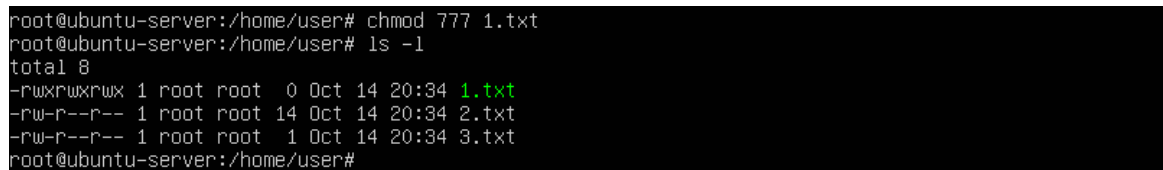
```
root@ubuntu-server:/# cd root
root@ubuntu-server:~# ls -a
.  ..  .bashrc  .local  .profile  snap  .ssh
root@ubuntu-server:~#
```

Рисунок 10 – Создание файлов

В данном каталоге представлены персональные данные и данные профиля.

1.8. Изменение прав доступа на файл «1.txt» в директории пользователя «user»

Изменим права доступа на файл «1.txt» с помощью команды «chmod». Результат выполнения команды представлен на рисунке.



```
root@ubuntu-server:/home/user# chmod 777 1.txt
root@ubuntu-server:/home/user# ls -l
total 8
-rwxrwxrwx 1 root root 0 Oct 14 20:34 1.txt
-rw-r--r-- 1 root root 14 Oct 14 20:34 2.txt
-rw-r--r-- 1 root root 1 Oct 14 20:34 3.txt
root@ubuntu-server:/home/user#
```

Рисунок 11 – Изменение прав доступа на файл «1.txt»

Данная команда предоставляет все права (чтение, запись, выполнение) для всех типов пользователей (владелец, группа и другие).

1.9. Создание жесткой и символической ссылки на файл «2.txt»

С помощью команды «ln» создадим жесткую ссылку на файл «2.txt». С помощью команды «ln -s» создадим символическую ссылку на файл «2.txt». Результат выполнения команд представлены на рисунках 11 и 12.

```
-rw-r--r-- 1 root root 14 Oct 14 20:34 2.txt
-rw-r--r-- 1 root root 1 Oct 14 20:34 3.txt
root@ubuntu-server:/home/user# ln /home/user/2.txt hard_link
root@ubuntu-server:/home/user# ln -s /home/user/2.txt soft_link
```

Рисунок 12 – Создание жесткой и символической ссылки на файл «2.txt»

```
root@ubuntu-server:/home/user# ls -l
total 12
-rwxrwxrwx 1 root root 0 Oct 14 20:34 1.txt
-rw-r--r-- 2 root root 14 Oct 14 20:34 2.txt
-rw-r--r-- 1 root root 1 Oct 14 20:34 3.txt
-rw-r--r-- 2 root root 14 Oct 14 20:34 hard_link
lrwxrwxrwx 1 root root 16 Oct 14 21:03 soft_link -> /home/user/2.txt
root@ubuntu-server:/home/user# _
```

Рисунок 13 – Результат создания жесткой и символической ссылки на файл «2.txt»

1.10. Создание каталога «new» в каталоге пользователей «user»

С помощью команды «mkdir» создадим директорию «new» в каталоге пользователя «user». Результат выполнения представлен на рисунке 13.

```
root@ubuntu-server:/home/user# mkdir new
root@ubuntu-server:/home/user# ls
1.txt 2.txt 3.txt hard_link new soft_link
root@ubuntu-server:/home/user# cd new
root@ubuntu-server:/home/user/new# _
```

Рисунок 14 – Создание директории «new»

1.11. Копирование файла «1.txt» в каталог «new»

С помощью команды «ср» скопируем файл «1.txt» в каталог «new». Результат выполнения представлен на рисунке 14.

```
root@ubuntu-server:/home/user/new# cd /
root@ubuntu-server:/# cp /home/user/1.txt /home/user/new/1.txt
root@ubuntu-server:/# ls -l /home/user/new
total 0
-rwxr-xr-x 1 root root 0 Oct 14 21:08 1.txt
root@ubuntu-server:/# _
```

Рисунок 15 – Копирование файла «1.txt»

1.12. Перемещение файла «2.txt» в каталог «new»

С помощью команды «mv» скопируем файл «2.txt» в каталог «new». Результат выполнения представлен на рисунке 15.

```
root@ubuntu-server:/# mv /home/user/2.txt /home/user/new/2.txt
root@ubuntu-server:/# ls -l /home/user/new
total 4
-rwxr-xr-x 1 root root  0 Oct 14 21:08 1.txt
-rw-r--r-- 2 root root 14 Oct 14 20:34 2.txt
root@ubuntu-server:/# _
```

Рисунок 16 – Перемещение файла «2.txt»

1.13. Изменение владельца файла «3.txt» и каталога «new»

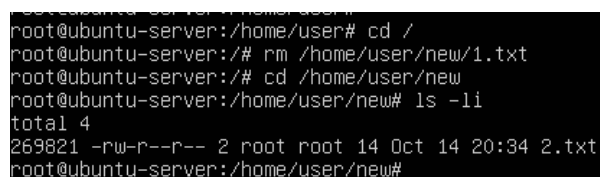
С помощью команды «chown» реализуем смену владельца у файла «3.txt» и каталога «new». Результат выполнения представлен на рисунке 16.

```
root@ubuntu-server:~# chown ubuntu-user /home/user/3.txt
root@ubuntu-server:~# chown ubuntu-user /home/user/new
root@ubuntu-server:~# cd /home/user
root@ubuntu-server:/home/user# ls -li
total 12
269819 -rwxrwxrwx 1 root      root      0 Oct 14 20:34 1.txt
269823 -rw-r--r-- 1 ubuntu-user root      1 Oct 14 20:34 3.txt
269821 -rw-r--r-- 2 root      root     14 Oct 14 20:34 hard_link
286996 drwxr-xr-x 2 ubuntu-user root  4096 Oct 14 21:09 new
269822 lrwxrwxrwx 1 root      root     16 Oct 14 21:03 soft_link -> /home/user/2.txt
root@ubuntu-server:/home/user# _
```

Рисунок 17 – Смена владельца у файла «3.txt» и каталога «new»

1.14. Удаление файла «1.txt» в каталоге «new»

С помощью команды «rm» удалим файл «1.txt» в каталоге «new». Результат выполнения представлен на рисунке 17.

A screenshot of a terminal window with a black background and white text. The text shows a sequence of commands and their outputs in a root shell on an Ubuntu server. The commands are: 'cd /', 'rm /home/user/new/1.txt', 'cd /home/user/new', and 'ls -li'. The output of 'ls -li' shows a file named '2.txt' with permissions '-rw-r--r--', size 269821, owned by root, and a timestamp of 14 Oct 14 20:34.

```
root@ubuntu-server:/home/user# cd /
root@ubuntu-server:/# rm /home/user/new/1.txt
root@ubuntu-server:/# cd /home/user/new
root@ubuntu-server:/home/user/new# ls -li
total 4
269821 -rw-r--r-- 2 root root 14 Oct 14 20:34 2.txt
root@ubuntu-server:/home/user/new#
```

Рисунок 18 – Удаление файла «1.txt» в каталоге «new»

1.15. Удаление каталога «new»


С помощью команды «rm -r» удалим каталог «new» вместе с его содержимым. Результат выполнения представлен на рисунке 18.

```
root@ubuntu-server:/home/user# rm -r /home/user/new
root@ubuntu-server:/home/user# ls -li
total 8
269819 -rwxrwxrwx 1 root      root   0 Oct 14 20:34 1.txt
269823 -rw-r--r-- 1 ubuntu-user root   1 Oct 14 20:34 3.txt
269821 -rw-r--r-- 1 root      root  14 Oct 14 20:34 hard_link
269822 lrwxrwxrwx 1 root      root  16 Oct 14 21:03 soft_link -> /home/user/2.txt
root@ubuntu-server:/home/user# _
```

Рисунок 19 – Удаление каталога «new»

1.16. Поиск файла «vga2iso» с помощью команды «find»

С помощью команды «find» не получилось найти файл «vga2iso», так как используется более новая версия Ubuntu Server, поэтому дополнительно был выполнен поиск «1.txt». Результат выполнения представлен на рисунке 19.



```
root@ubuntu-server:/# find / -name "vga2iso"  
root@ubuntu-server:/# find / -name "1.txt"  
/home/user/1.txt  
root@ubuntu-server:/#
```

Рисунок 20 – Поиск файлов «vga2iso» и «1.txt»

Выводы

В ходе выполнения данной лабораторной работы мной был получен опыт работы с файлами и каталогами в ОС Linux, настройки прав на доступ к файлам и каталогам.

Контрольные вопросы

1. Что такое файловая система? Файловая система - часть операционной системы, которая обеспечивает чтение и запись файлов на дисковых носителях информации. Файловая система устанавливает физическую и логическую структуру файлов, правила их создания и управления ими, а также сопутствующие данные файла и идентификацию. Конкретная файловая система определяет размер имени файла, максимальный возможный размер файла.
2. Права доступа к файлам. Назначение прав доступа.

Каждый файл имеет три параметра доступа. Вот они:

- Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;
- Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;
- Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Но все эти права были бы бессмысленными, если бы применялись сразу для всех пользователей. Поэтому каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

- Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение.
- Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу.

- Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла.

Именно с помощью этих наборов полномочий устанавливаются права файлов в linux. Каждый пользователь может получить полный доступ только к файлам, владельцем которых он является или к тем, доступ к которым ему разрешен. Только пользователь Root может работать со всеми файлами независимо от их набора их полномочий

3. Жесткая ссылка в Linux. Основные сведения. Жесткая ссылка (hard link) является своего рода синонимом для существующего файла. Когда вы создаете жесткую ссылку, создается дополнительный указатель на существующий файл, но не копия файла.

Жесткие ссылки выглядят в файловой структуре как еще один файл. Если вы создаете жесткую ссылку в том же каталоге, где находится целевой файл, то они должны иметь разные имена. Жесткая ссылка на файл должна находиться в той же файловой системе, где и другие жесткие ссылки на этот файл.

В Linux каждый файл имеет уникальный идентификатор - индексный дескриптор (inode). Это число, которое однозначно идентифицирует файл в файловой системе. Жесткая ссылка и файл, для которой она создавалась имеют одинаковые inode. Поэтому жесткая ссылка имеет те же права доступа, владельца и время последней модификации, что и целевой файл. Различаются только имена файлов. Фактически жесткая ссылка это еще одно имя для файла.

Жесткие ссылки нельзя создавать для директорий.

Жесткая ссылка не может указывать на несуществующий файл.

Жесткие ссылки появились раньше, чем символические, но сейчас уже устаревают. В повседневной работе жесткие ссылки используются редко.

4. Команда поиска в Linux. Основные сведения. Команда find - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных

условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

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

Команда `find` имеет такой синтаксис:

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

Папка - каталог в котором будем искать

Параметры - дополнительные параметры, например, глубина поиска, и т д

Критерий - по какому критерию будем искать: имя, дата создания, права, владелец и т д.

Шаблон - непосредственно значение по которому будем отбирать файлы. Параметры:

- `-P` - никогда не открывать символические ссылки.
- `-L` - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.
- `-maxdepth` - максимальная глубина поиска по подкаталогам, для поиска только в текущем каталоге установите 1.
- `-depth` - искать сначала в текущем каталоге, а потом в подкаталогах.
- `-mount` искать файлы только в этой файловой системе.
- `-version` - показать версию утилиты `find`.
- `-print` - выводить полные имена файлов.
- `-type f` - искать только файлы.
- `-type d` - поиск папки в Linux.

5. Перечислите основные команды работы с каталогами.

- `ls` - список файлов в директории;
- `cd` - переход между директориями;
- `rm` - удалить файл;
- `rmdir` - удалить папку;
- `mkdir` - создать папку;
- `chmod` - изменить права файла;
- `mv` - переместить файл;
- `cp` - скопировать файл;
- `ln` - создать ссылку;
- `pwd` - узнать текущий каталог;
- `touch` - создать пустой файл.