

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

Мастылина А.А.

Группа АИ-18

Руководитель

Кургасов В.В.

Липецк 2020 г.

Оглавление

Цель работы	3
Задание кафедры.....	4
Ход работы	5
Вывод	20
Ответы на контрольные вопросы	21

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

Запустим виртуальную машину Linux Ubuntu и загрузимся пользователем root при помощи команды `sudo su`. Пример выполнения работы представлен на рисунке 1.

```
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

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

System information disabled due to load higher than 1.0

60 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Fri Oct 23 08:03:22 UTC 2020 on tty1
anna@annaserver:~$ sudo su
[sudo] password for anna:
Sorry, try again.
[sudo] password for anna:
root@annaserver:/home/anna# _
```

Рисунок 1 – Загрузка пользователем root (`sudo su`)

Посмотрим содержание корневой директории с помощью команды `ls` и опишем каждый из этих каталогов. Пример выполнения работы представлен на рисунке 2.

```
root@annaserver:/home/anna# cd /
root@annaserver:/# 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 – Содержание корневой директории

Описание каталогов

1. `/bin` – содержит основные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям (пример: `cat`, `ls`, `cp`, `tail`, `ps`), исполняемые файлы, а также символичные ссылки на исполняемые файлы;

2. `/boot` – это каталог в котором находятся файлы, необходимые для загрузки системы такие как GRUB и ядра Linux. Здесь нет конфигурационных файлов, используемых загрузчиком - они находятся в каталоге `/etc` вместе с другими конфигурационными файлами. В `/boot` хранятся данные, которые используются до того, как ядро начинает исполнять программы пользователя;

3. /cdrom – это временное место, где монтируются диски CD-ROM, когда они вставляются в компьютер. Однако, стандартное место для подключаемого носителя находится внутри каталога /media;

4. /dev – содержит файлы устройств;

5. /etc – содержит конфигурационные файлы операционной системы и всех сетевых служб;

6. /home – содержит домашние каталоги всех пользователей, зарегистрированных в системе;

7. /lib – содержит основные библиотеки и модули ядра, необходимые для работы программ из /bin и /sbin;

8. /lib 64 – 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;

9. /lost+found – этот каталог нужен для хранения испорченных файлов при проблемах с файловой системой, которые были восстановлены после, например, некорректного размонтирования файловой системы;

10. /media – данный каталог содержит - точки монтирования для сменных носителей, таких как CD-ROM, DVD-ROM (впервые описано в FHS-2.3);

11. /mnt – точки монтирования. В современных дистрибутивах Linux этот процесс обычно происходит автоматически. При этом в каталогах /mnt или /media создается подкаталог, имя которого совпадает с именем монтируемого тома;

12. /opt – дополнительное программное обеспечение, здесь обычно размещаются установленные программы, имеющие большой дисковый объем, или вспомогательные пакеты;

13. /proc – каталог псевдофайловой системы procfs, которая используется для предоставления информации о процессах (по-другому это

виртуальная файловая система, которая обеспечивает связь с ядром и монтируется в каталогу `/proc`). Он существует только во время работы системы в оперативной памяти компьютера. Каталог представляет интерес и с точки зрения безопасности;

14. `/root` – каталог пользователя `root`;

15. `/run` – хранение данных, которые были запущены приложениями;

16. `/sbin` – набор утилит для системного администрирования, содержит исполняемые файлы, необходимые для загрузки системы и ее восстановления в различных щекотливых ситуациях. Запускать эти утилиты имеет право только `root`;

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

18. `/srv` – параметры, которые специфичные для окружения системы, чаще всего данная директория пуста;

19. `/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;

20. `/tmp` – временные файлы. Linux, регулярно очищает этот каталог;

21. `/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;

22. `/var` – переменные файлы (*variable*), которые подвергаются наиболее частому изменению. Например, кэши различных программ; файлы блокировки для недопустимости одновременного использования одной программы несколькими пользователями; файлы системных журналов; временные файлы (при выключении компьютера содержимое очищается); информация о различных программах; общая информация о состоянии

системы с момента последней загрузки, входа в систему и т.д.; очередь печати, факсов, а также входящие почтовые ящики пользователей и т.д;

Просмотрим содержимое каталога файлов физических устройств (/dev).

На рисунке 3 приведён перечень файлов физических устройств.

```
root@annaserver:/# cd dev
root@annaserver:/dev# ls
autofs          hwrng           port            tty0            tty3            tty50           ttyS13          ttyS6           vcsa3
block           i2c-0           ppp             tty1            tty30           tty51           ttyS14          ttyS7           vcsa4
bsg             initctl         psaux           tty10           tty31           tty52           ttyS15          ttyS8           vcsa5
btrfs-control   input           ptmx            tty11           tty32           tty53           ttyS16          ttyS9           vcsa6
bus             kmsg            pts             tty12           tty33           tty54           ttyS17          ttyprintk       vcsu
cdrom           lightningvm     random          tty13           tty34           tty55           ttyS18          ubuntu-vg       vcsu1
char            log             rfkill          tty14           tty35           tty56           ttyS19          udmabuf         vcsu2
console         loop-control    rtc             tty15           tty36           tty57           ttyS2           uhid            vcsu3
core            loop0           rtc0            tty16           tty37           tty58           ttyS20          uinput         vcsu4
cpu_dma_latency loop1           sda             tty17           tty38           tty59           ttyS21          urandom         vcsu5
cuse            loop2           sda1            tty18           tty39           tty6             ttyS22          userio          vcsu6
disk            loop3           sda2            tty19           tty4             tty60           ttyS23          vboxguest       vfiio
dm-0            loop4           sda3            tty2             tty40           tty61           ttyS24          vboxuser        vga_arbiter
dri             loop5           sg0             tty20           tty41           tty62           ttyS25          vcs             vhci
dvd             loop6           sg1             tty21           tty42           tty63           ttyS26          vcs1            vhost-net
ecryptfs        loop7           shm             tty22           tty43           tty7             ttyS27          vcs2            vhost-vsock
fb0             mapper          snapshot        tty23           tty44           tty8             ttyS28          vcs3            zero
fd              mclog           snd             tty24           tty45           tty9             ttyS29          vcs4            zfs
full            mem             sr0            tty25           tty46           tty80           ttyS3           vcs5
fuse            mqqueue         stderr          tty26           tty47           tty81           ttyS30          vcs6
hidraw0         net             stdin          tty27           tty48           tty810          ttyS31          vcsa
hpet            null            stdout          tty28           tty49           tty811          ttyS4           vcsa1
hugepages       nvram           tty            tty29           tty5             tty812          ttyS5           vcsa2
root@annaserver:/dev# _
```

Рисунок 3 – Содержимое каталога файлов физических устройств

Укажем назначения файлов

1. `acpi_thermal_rel` – обеспечивает функции управления температурой модуля ACPI;
2. `autofs` – система управления автоматическим монтированием (и отмонтированием).
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` – реализация символьных устройств (char devices) в Linux

7. `drm_dp_aux` – канал DisplayPort AUX;
8. `ecryptfs` – POSIX- совместимая многоуровневая криптографическая файловая система в ядре Linux;
9. `fb` – устройство обеспечивает абстракцию для графического оборудования;
10. `freefall` – это решение для управления брандмауэром для многих дистрибутивов Linux, включая Ubuntu, Debian, CentOS, RHEL и Fedora;
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` – тип таймера, используемый в персональных компьютерах
13. `hwrng` – генератор случайных чисел;
14. `i2c` – шина предлагает различные преимущества, такие как экономия места на плате, уменьшение общей стоимости оборудования, а также предлагает средства упрощённой отладки;
15. `kmsg` – узел символьного устройства обеспечивает доступ пользователя к буферу `printk` ядра;
16. `kvm` – программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86;
17. `loop` – это блочное устройство, которое отображает блоки данных обычного файла в файловой системе или другое блочное устройство;
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` – программа `mcelog` декодирует машинные события (аппаратных ошибок) на x86-64, работающих под управлением 64-разрядной Linux

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

21. `mem` – это символьный файл устройства, в котором отображается главная память компьютера. Он может использоваться, например, для проверки (и даже исправления) системы;

22. `null` – специальный файл в системах класса UNIX, представляющий собой так называемое «пустое устройство»;

23. `nvrnm` – она же энергонезависимая память, применяется в современных UEFI BIOS, в отличие от старых BIOS, где для хранения использовали CMOS SRAM + батарейка;

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

25. `ppp` – это механизм для создания и запуска IP (Internet Protocol) и других сетевых протоколов;

26. `psaux` – устройство мыши PS / 2;

27. `ptmx` – является символьным файлом с основным номером, равным 5 и вторичным номером 2, обычно имеет права доступа 0666, владелец и группа равны `root`. Используется для создания пары основного и подчиненного псевдотерминала;

28. `random` – специальные символьные псевдоустройства в некоторых UNIX-подобных системах, впервые появившиеся в ядре Linux версии ;

29. `rftkill` – это подсистема в ядре Linux, предоставляющая интерфейс, через который можно запрашивать, активировать и деактивировать радиопередатчики в компьютерной 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. tpm – разрешает доступ к устройству Trusted Platform Module (tpm);

38. tty – виртуальная консоль;

39. ttyprintk – драйвер псевдо TTY, который позволяет пользователям создавать сообщения printk через вывод на устройство ttyprintk;

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

41. uinput – поддержка драйвера уровня пользователя для ввода;

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

43. userio – призван упростить жизнь разработчикам драйверов ввода, позволяя им тестировать различные устройства Serio (в основном, различные сенсорные панели на ноутбуках), не имея физического устройства перед ними;

44. vcs – 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

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

Создадим нового пользователя `user`, для этого воспользуемся командой `useradd`. Пример выполнения представлен на рисунке 6

```
root@annaserver:/# useradd -m user
root@annaserver:/# cd /home
root@annaserver:/home# ls
anna user
root@annaserver:/home# ls -l
total 8
drwxr-xr-x 3 anna anna 4096 Oct 23 08:01 anna
drwxr-xr-x 2 user user 4096 Oct 28 14:42 user
root@annaserver:/home# _
```

Рисунок 6 – Пример создания нового пользователя `user`

Создадим в директории пользователя `user` три файла `1.txt`, `2.txt` и `3.txt`, используя команды `touch`, `cat` и текстовый редактор `nano`. Пример выполнения представлен на рисунках 7 и 8.

```
root@annaserver:/home/user# ls -l
total 4
-rw-r--r-- 1 root root 0 Oct 28 16:15 1.txt
-rw-r--r-- 1 root root 3 Oct 28 16:18 2.txt
-rw-r--r-- 1 root root 0 Oct 28 16:27 3.txt
root@annaserver:/home/user# _
```

Рисунок 7 – Пример работы

Просмотрим и поясним права доступа к файлам.

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

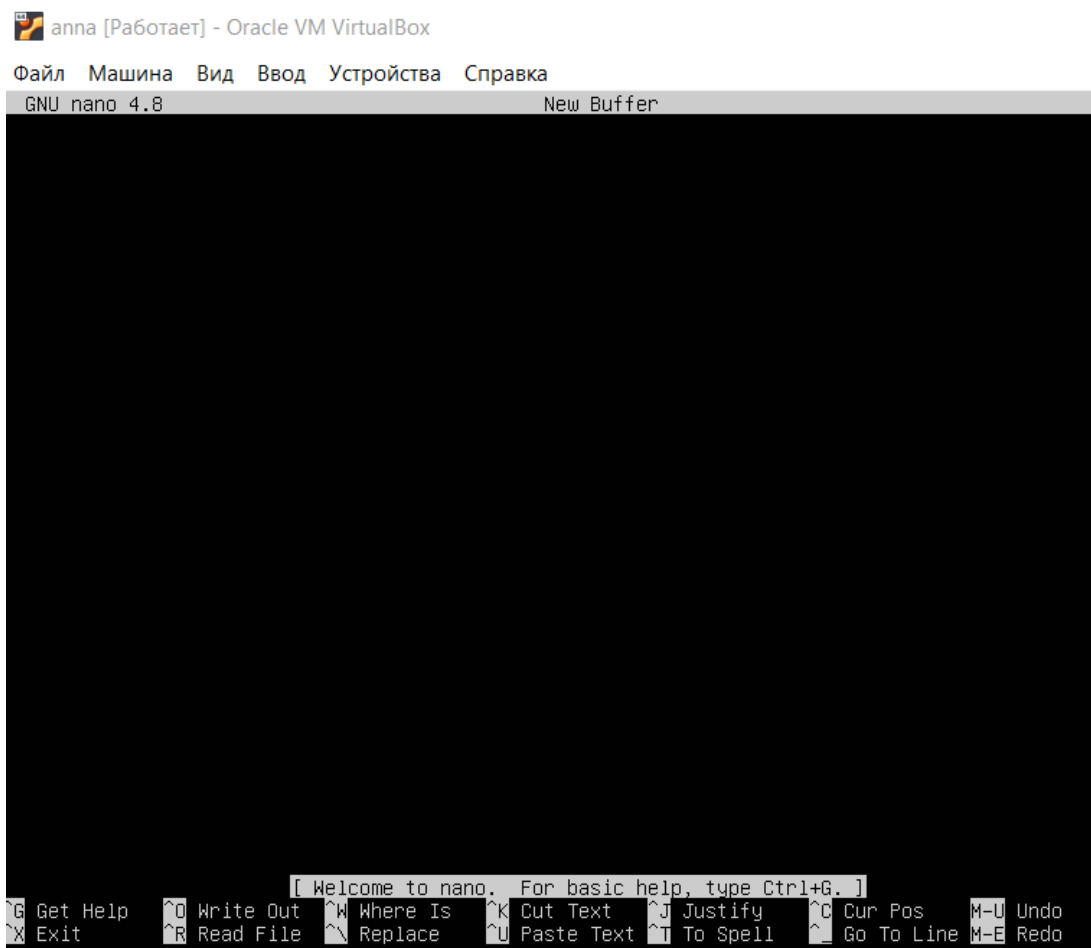


Рисунок 8 – Пример создания файла используя nano

Перейдём в директорию пользователя root. Пример выполнения представлен на рисунке 9

```
root@annaserver:/home/anna# cd /
root@annaserver:/# cd root
root@annaserver:~# ls
snap
root@annaserver:~# ls -a
. . . .bash_history .bashrc .local .profile .ssh snap
root@annaserver:~# _
```

Рисунок 9 – Каталог root

Выполним следующее задание, изменим права доступа на файл 1.txt в директории пользователя user с помощью команды chmod.

```
root@annaserver:~# chmod 777 /home/user/1.txt
root@annaserver:~# ls-l
ls-l: command not found
root@annaserver:~# ls -l
total 4
drwxr-xr-x 3 root root 4096 Oct 23 07:58 snap
root@annaserver:~#
```

Рисунок 10 – Пример изменения прав доступа с помощью chmod

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`.

Пример выполнения представлен на рисунка 11 и 12.

```
root@annaserver:/home/user# cd -  
/home  
root@annaserver:/home# cd /  
root@annaserver:/# ln /home/user/2.txt hardlink  
root@annaserver:/# ls -l  
total 1808460  
lrwxrwxrwx 1 root root 7 Jul 31 16:28 bin -> usr/bin  
drwxr-xr-x 4 root root 4096 Oct 23 07:55 boot  
drwxr-xr-x 2 root root 4096 Oct 23 07:36 cdrom  
drwxr-xr-x 19 root root 4080 Oct 29 12:52 dev  
drwxr-xr-x 93 root root 4096 Oct 28 14:42 etc  
-rw-r--r-- 2 root root 3 Oct 28 16:18 hardlink  
drwxr-xr-x 4 root root 4096 Oct 28 14:42 home  
lrwxrwxrwx 1 root root 7 Jul 31 16:28 lib -> usr/lib  
lrwxrwxrwx 1 root root 9 Jul 31 16:28 lib32 -> usr/lib32  
lrwxrwxrwx 1 root root 9 Jul 31 16:28 lib64 -> usr/lib64  
lrwxrwxrwx 1 root root 10 Jul 31 16:28 libx32 -> usr/libx32  
drwx----- 2 root root 16384 Oct 23 07:35 lost+found  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 media  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 mnt  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 opt  
dr-xr-xr-x 107 root root 0 Oct 29 12:52 proc  
drwx----- 5 root root 4096 Oct 28 16:22 root  
drwxr-xr-x 25 root root 760 Oct 29 12:55 run  
lrwxrwxrwx 1 root root 8 Jul 31 16:28/sbin -> usr/sbin  
drwxr-xr-x 6 root root 4096 Oct 23 07:57 snap  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 srv  
-rw----- 1 root root 1851785216 Oct 23 07:45 swap.img  
dr-xr-xr-x 13 root root 0 Oct 29 12:52 sys  
drwxrwxrwt 11 root root 4096 Oct 29 12:53 tmp  
drwxr-xr-x 14 root root 4096 Jul 31 16:29 usr  
drwxr-xr-x 13 root root 4096 Jul 31 16:30 var  
root@annaserver:/#
```

Рисунок 11 – Создание жёсткой ссылки (hardlink)

```
root@annaserver:/home/anna# cd /  
root@annaserver:/# ls -l  
total 1808460  
lrwxrwxrwx 1 root root 7 Jul 31 16:28 bin -> usr/bin  
drwxr-xr-x 4 root root 4096 Oct 23 07:55 boot  
drwxr-xr-x 2 root root 4096 Oct 23 07:36 cdrom  
drwxr-xr-x 19 root root 4080 Oct 30 06:23 dev  
drwxr-xr-x 93 root root 4096 Oct 28 14:42 etc  
-rw-r--r-- 1 root root 3 Oct 28 16:18 hardlink  
drwxr-xr-x 4 root root 4096 Oct 28 14:42 home  
lrwxrwxrwx 1 root root 7 Jul 31 16:28 lib -> usr/lib  
lrwxrwxrwx 1 root root 9 Jul 31 16:28 lib32 -> usr/lib32  
lrwxrwxrwx 1 root root 9 Jul 31 16:28 lib64 -> usr/lib64  
lrwxrwxrwx 1 root root 10 Jul 31 16:28 libx32 -> usr/libx32  
drwx----- 2 root root 16384 Oct 23 07:35 lost+found  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 media  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 mnt  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 opt  
dr-xr-xr-x 112 root root 0 Oct 30 06:23 proc  
drwx----- 5 root root 4096 Oct 28 16:22 root  
drwxr-xr-x 25 root root 760 Oct 30 06:23 run  
lrwxrwxrwx 1 root root 8 Jul 31 16:28/sbin -> usr/sbin  
drwxr-xr-x 6 root root 4096 Oct 23 07:57 snap  
lrwxrwxrwx 1 root root 16 Oct 29 13:05 softlink -> /home/user/2.txt  
drwxr-xr-x 2 root root 4096 Jul 31 16:28 srv  
-rw----- 1 root root 1851785216 Oct 23 07:45 swap.img  
dr-xr-xr-x 13 root root 0 Oct 30 06:22 sys  
drwxrwxrwt 11 root root 4096 Oct 30 06:23 tmp  
drwxr-xr-x 14 root root 4096 Jul 31 16:29 usr  
drwxr-xr-x 13 root root 4096 Jul 31 16:30 var  
root@annaserver:/# _
```

Рисунок 12 – Создание символической ссылки (softlink)

Далее нужно создать директорию new в каталоге пользователя user. Для этого используем команду mkdir.

```
root@annaserver:/home/anna# cd /
root@annaserver:/# mkdir /home/user/new
root@annaserver:/# ls -l /home
total 8
drwxr-xr-x 3 anna anna 4096 Oct 23 08:01 anna
drwxr-xr-x 3 user user 4096 Oct 29 13:13 user
root@annaserver:/# ls -l /home/user/
total 8
-rwxrwxrwx 1 root root 0 Oct 28 16:15 1.txt
-rw-r--r-- 2 root root 3 Oct 28 16:18 2.txt
-rw-r--r-- 1 root root 0 Oct 28 16:27 3.txt
drwxr-xr-x 2 root root 4096 Oct 29 13:13 new
root@annaserver:/# _
```

Рисунок 13 – Создание директории new

Следующим заданием нужно скопировать файл 1.txt и переместить файл 2.txt в созданную директорию new. Пример выполнения представлен на рисунках 14 и 15.

```
root@annaserver:/# cp /home/user/1.txt /home/user/new/
root@annaserver:/# ls -l home/user/new
total 0
-rwxr-xr-x 1 root root 0 Oct 29 13:58 1.txt
root@annaserver:/# _
```

Рисунок 14 – Копирование файла


```

root@annaserver:/# mv /home/user/2.txt /home/user/new/
root@annaserver:/# ls -l /home/user/new
total 4
-rwxr-xr-x 1 root root 0 Oct 29 13:58 1.txt
-rw-r--r-- 2 root root 3 Oct 28 16:18 2.txt
root@annaserver:/# ls -l /home/user
ls: invalid option -- '/'
Try 'ls --help' for more information.
root@annaserver:/# ls -l /home/user
total 4
-rwxrwxrwx 1 root root 0 Oct 28 16:15 1.txt
-rw-r--r-- 1 root root 0 Oct 28 16:27 3.txt
drwxr-xr-x 2 root root 4096 Oct 29 14:06 new
root@annaserver:/# _

```

Рисунок 15 – Перемещение файла

Далее нужно поменять владельцев файла 3.txt и каталога new реализуем это с помощью chown

```

root@annaserver:/# ls -l /home/user/
total 4
-rwxrwxrwx 1 root root 0 Oct 28 16:15 1.txt
-rw-r--r-- 1 root root 0 Oct 28 16:27 3.txt
drwxr-xr-x 2 root root 4096 Oct 29 14:06 new
root@annaserver:/# chown user /home/user/3.txt
root@annaserver:/# chown user /home/user/new
root@annaserver:/# ls -l /home/user
total 4
-rwxrwxrwx 1 root root 0 Oct 28 16:15 1.txt
-rw-r--r-- 1 user root 0 Oct 28 16:27 3.txt
drwxr-xr-x 2 user root 4096 Oct 29 14:06 new
root@annaserver:/#

```

Рисунок 16 – Изменение владельцев файла и каталога

По заданию нужно удалить файл 1.txt из директории new, а затем удалить директорию new. Используем для этого команду rm:

```
root@annaserver:/# rm /home/user/new/1.txt
root@annaserver:/# ls -l /home/user/new
total 4
-rw-r--r-- 2 root root 3 Oct 28 16:18 2.txt
root@annaserver:/# rm -R /home/user/new/
root@annaserver:/# ls -l /home/user
total 0
-rwxrwxrwx 1 root root 0 Oct 28 16:15 1.txt
-rw-r--r-- 1 user root 0 Oct 28 16:27 3.txt
root@annaserver:/# _
```

Рисунок 17 – Удаление файла и директории

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

```
root@annaserver:/home/anna# cd /
root@annaserver:/# find / -name vga2iso
_
```

Рисунок 18 – Нахождение файла vga2iso

Так как работа производится в более новой версии Ubuntu Server файла vga2iso не существует. Поэтому приведём пример работы программы поиска другого файла, а именно 1.txt.

```
root@annaserver:/home/anna# find /-name boot
find: '/-name': No such file or directory
find: 'boot': No such file or directory
root@annaserver:/home/anna# find / -name 1.txt
/home/user/1.txt
```

Рисунок 19 – Нахождение файла 1.txt

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.

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.