

**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

Осипов А.А.

Группа АИ-18

Руководитель

Кургасов В.В.

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

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

```
Ubuntu 20.04.1 LTS artemserver tty1
artemserver login: artem
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-42-generic x86_64)

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

System information as of Thu 29 Oct 2020 01:22:35 PM UTC

System load:  0.63           Processes:           102
Usage of /:   43.2% of 8.79GB Users logged in:          0
Memory usage: 17%           IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%

0 updates can be installed immediately.
0 of these updates are security updates.

Last login: Wed Oct 28 12:50:23 UTC 2020 on tty1
artem@artemserver:~$ sudo su
[sudo] password for artem:
root@artemserver:/home/artem# _
```

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@artemserver:/home/artem# cd /
root@artemserver:/# 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` – в каталоге `/bin` находятся основные двоичные пользовательские модули (программы), которые должны присутствовать, если система монтируется в однопользовательском режиме.

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 – в каталоге /dev находится ряд специальных файлов, с помощью которых представлены устройства.;

5. /etc – в каталоге /etc находятся конфигурационные файлы, которые обычно можно отредактировать вручную в текстовом редакторе. 6. /home – содержит домашние каталоги всех пользователей, зарегистрированных в системе;

7. /lib – в каталоге /lib находятся библиотеки, необходимые для основных двоичных файлов, находящихся в каталогах /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 – 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.

10. /media – в каталоге находятся подкаталоги, в которых монтируются съемные носители, вставляемые в компьютер. Например, когда вы в системе Linux вставляете диск CD, в каталоге /media будет автоматически создан соответствующий подкаталог

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 – в каталоге /opt находятся подкаталоги для дополнительных пакетов программного обеспечения.

13. /proc – каталог /proc похож на каталог /dev, поскольку он не содержит стандартных файлов. В нем находятся специальные файлы, в которых представлена информация о системе и о процессах.

14. /root – каталог /root является домашним каталогом пользователя root.

15. /run – каталог /run является достаточно новым каталогом, в котором приложениям предоставляется возможность стандартным образом хранить вспомогательные файлы, которые им требуются, например, сокеты и идентификаторы процессов.

16. /sbin – каталог /sbin похож на каталог /bin. В нем находятся важные двоичные файлы, которые, как правило, предназначены для их запуска пользователем при администрировании системы.

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

18. /srv – в каталоге /srv находятся данные для сервисов, предоставляемых системой.

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 – приложения хранят временные файлы в каталоге /tmp.

21. /usr – В каталоге /usr находятся приложения и файлы, используемые пользователями, в отличие от приложений и файлов, используемых системой. 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.

```

Last login: Wed Oct 28 12:50:23 UTC 2020 on tty1
artem@artemserver:~$ sudo su
[sudo] password for artem:
root@artemserver:/home/artem# cd /
root@artemserver:/# 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
root@artemserver:/#
/  cd  ls
root@artemserver:/# cd dev
root@artemserver:/dev# ls
autofs          hwmon          port            tty0            tty3            tty50           ttyS12          ttyS5           vcsa3
block           i2c-0          ppp             tty1            tty30           tty51           ttyS13          ttyS6           vcsa4
bsg             initctl        psaux          tty10           tty31           tty52           ttyS14          ttyS7           vcsa5
btrfs-control   input          ptmx           tty11           tty32           tty53           ttyS15          ttyS8           vcsa6
bus             kmsg          pts            tty12           tty33           tty54           ttyS16          ttyS9           vcsu
cdrom           lightnvm       random          tty13           tty34           tty55           ttyS17          ubuntu-vg       vcsu1
char            log            rfkill          tty14           tty35           tty56           ttyS18          udmabuf         vcsu2
console         loop0          rtc             tty15           tty36           tty57           ttyS19          uhid            vcsu3
core            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        vfi
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
hugepages       nvram          tty             tty29           tty5             ttyS11           ttyS4           vcsa2
root@artemserver:/dev#

```

Рисунок 3 – Содержимое каталога файлов физических устройств

Укажем назначения файлов

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

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 – 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 – тип таймера, 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 – программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86;

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

38. `tty` – виртуальная консоль;

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

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

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

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

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

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

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

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

47. vga\_arbiter – сканирует все устройства PCI и добавляет в арбитраж VGA. Затем арбитр включает / отключает декодирование на разных устройствах устаревших инструкций VGA;

48. vhci – виртуальный драйвер HCI Bluetooth;

49. vhost-net – ускоритель ядра хоста для virtio net;

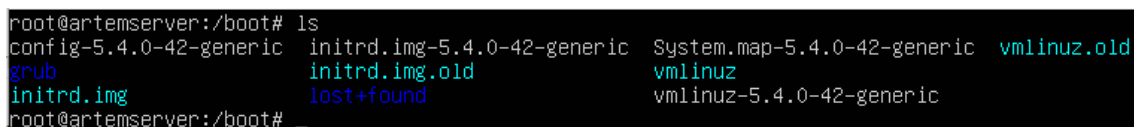
50. vhost-vsock – программное устройство, поэтому нет пробного вызова, который вызывает драйвер, чтобы зарегистрировать его узел устройства misc char. Это создает проблема с курицей и яйцом: приложения в пользовательском пространстве должны открываться / dev / vhost-vsock, чтобы использовать драйвер,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

51. video – устройство видеозахвата / наложения;

52. zero – специальный файл в UNIX-подобных системах, представляющий собой источник нулевых байтов;

53. zfs – файловая система, разработанная компанией Sun Microsystems и обладающая такими характеристиками как возможность хранения больших объёмов данных, управления томами и множеством других.

Перейдём к директории пользователя root и посмотрим содержимое каталога на рисунке 4.



```
root@artemserver:/boot# ls
config-5.4.0-42-generic  initrd.img-5.4.0-42-generic  System.map-5.4.0-42-generic  vmlinuz.old
grub                   initrd.img.old              vmlinuz
initrd.img             lost+found                  vmlinuz-5.4.0-42-generic
root@artemserver:/boot# _
```

Рисунок 4 – Содержимое каталога

Посмотрим содержимое файла vmlinuz с помощью команды cat на рисунке 5.

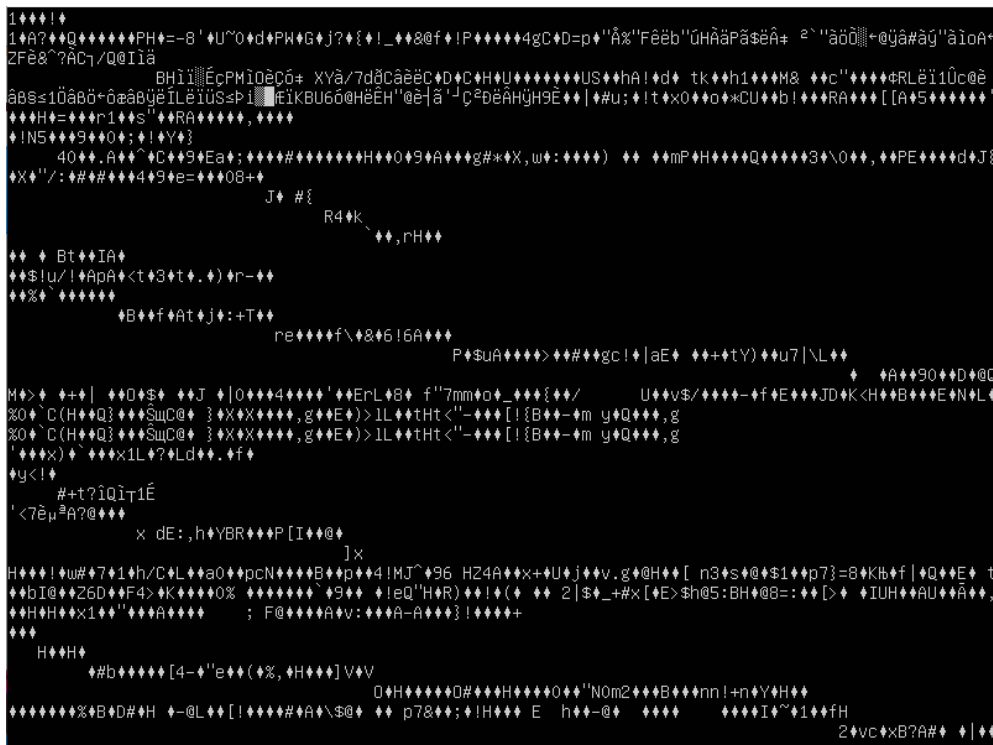


Рисунок 5 – Содержимое файла vmlinuz

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

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

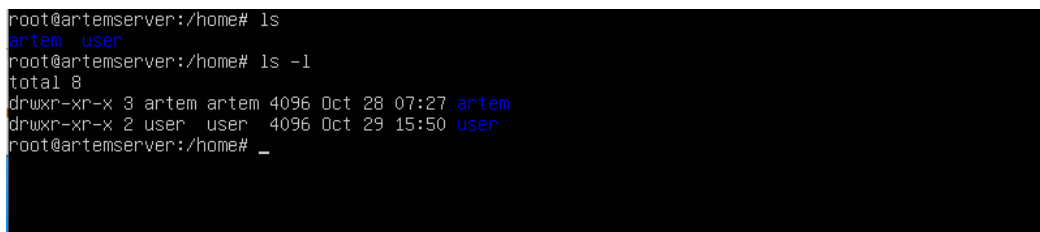


Рисунок 6 – Пример создания нового пользователя user

Создадим в директории пользователя user три файла 1.txt, 2.txt и 3.txt, используя команды touch, cat и текстовый редактор nano.

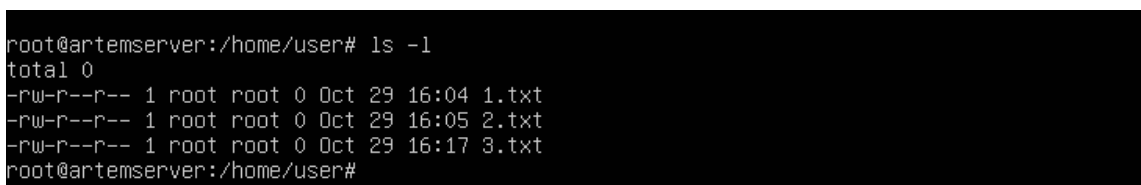


Рисунок 7 – Пример работы

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.

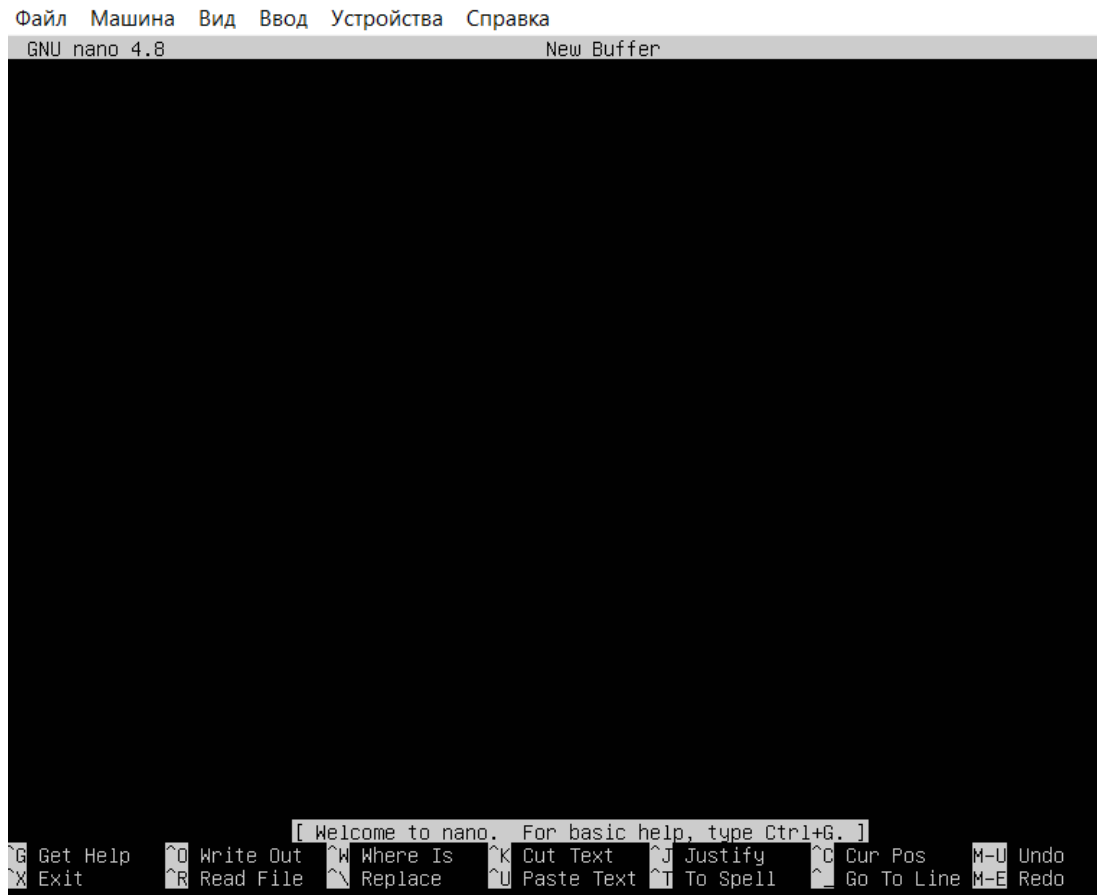


Рисунок 8 – Пример создания файла используя nano

Перейдём в директорию пользователя root.

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

Рисунок 9 – Каталог root

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

```
root@artemserver:~# chmod 777 /home/user/1.txt
root@artemserver:~# ls -l
total 4
drwxr-xr-x 3 root root 4096 Oct 28 07:25 snap
root@artemserver:~#
```

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`.

```
root@artemserver:~# ln /home/user/2.txt hardlink
root@artemserver:~# ls -l
total 4
-rw-r--r-- 2 root root    0 Oct 29 16:05 hardlink
drwxr-xr-x 3 root root 4096 Oct 28 07:25 snap
root@artemserver:~# _
```

Рисунок 11 – Создание жёсткой ссылки (hardlink)

```
root@artemserver:~# ln -s /home/user/2.txt softlink
root@artemserver:~# ls -l
total 4
-rw-r--r-- 2 root root    0 Oct 29 16:05 hardlink
drwxr-xr-x 3 root root 4096 Oct 28 07:25 snap
lrwxrwxrwx 1 root root   16 Oct 29 16:41 softlink -> /home/user/2.txt
root@artemserver:~# _
```

Рисунок 12 – Создание символической ссылки (softlink)

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

```
root@artemserver:~# mkdir /home/user/new
root@artemserver:~# ls -l /home
total 8
drwxr-xr-x 3 artem artem 4096 Oct 28 07:27 artem
drwxr-xr-x 3 user  user  4096 Oct 29 16:43 user
root@artemserver:~# ls -l /home/user/
total 4
-rwxrwxrwx 1 root root    0 Oct 29 16:04 1.txt
-rw-r--r-- 2 root root    0 Oct 29 16:05 2.txt
-rw-r--r-- 1 root root    0 Oct 29 16:17 3.txt
drwxr-xr-x 2 root root 4096 Oct 29 16:43 new
root@artemserver:~# _
```

Рисунок 13 – Создание директории `new`

Копируем файл `1.txt` и переместим файл `2.txt` в созданную директорию `new`

```
root@artemserver:~# cp /home/user/1.txt /home/user/new/
root@artemserver:~# ls -l /home/user/new
total 0
-rwxr-xr-x 1 root root 0 Oct 29 19:42 1.txt
root@artemserver:~#
```

Рисунок 14 – Копирование файла

```

root@artemserver:/# mv /home/user/2.txt /home/user/new/
root@artemserver:/# ls -l /home/user/new/
total 0
-rwxr-xr-x 1 root root 0 Oct 30 08:51 1.txt
-rw-r--r-- 1 root root 0 Oct 30 08:46 2.txt
root@artemserver:/# ls -l /home/user/
total 4
-rwxrwxrwx 1 root root 0 Oct 29 16:04 1.txt
-rw-r--r-- 1 user root 0 Oct 29 16:17 3.txt
drwxr-xr-x 2 root root 4096 Oct 30 08:53 new
root@artemserver:/# _

```

Рисунок 15 – Перемещение файла

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

```

root@artemserver:/# ls -l /home/user/
total 4
-rwxrwxrwx 1 root root 0 Oct 29 16:04 1.txt
-rw-r--r-- 1 root root 0 Oct 29 16:17 3.txt
drwxr-xr-x 2 root root 4096 Oct 29 19:46 new
root@artemserver:/# chown user /home/user/3.txt
root@artemserver:/# chown user /home/user/new
root@artemserver:/# ls -l /home/user
total 4
-rwxrwxrwx 1 root root 0 Oct 29 16:04 1.txt
-rw-r--r-- 1 user root 0 Oct 29 16:17 3.txt
drwxr-xr-x 2 user root 4096 Oct 29 19:46 new
root@artemserver:/# _

```

Рисунок 16 – Изменение владельцев файла и каталога

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

```

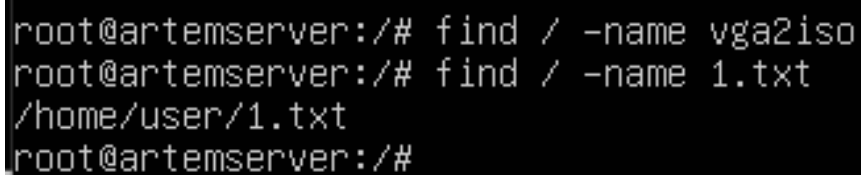
root@artemserver:/# rm /home/user/new/1.txt
root@artemserver:/# ls -l /home/user/new
total 0
-rw-r--r-- 2 root root 0 Oct 29 16:05 2.txt
root@artemserver:/# rm -R /home/user/new/
root@artemserver:/# ls -l /home/user
total 0
-rwxrwxrwx 1 root root 0 Oct 29 16:04 1.txt
-rw-r--r-- 1 user root 0 Oct 29 16:17 3.txt
root@artemserver:/#

```

Рисунок 17 – Удаление файла и директории



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

A terminal window with a black background and white text. The prompt is root@artemserver:/#. The first command is find / -name vga2iso, which returns no results. The second command is find / -name 1.txt, which returns /home/user/1.txt. The prompt returns to root@artemserver:/#.

```
root@artemserver:/# find / -name vga2iso
root@artemserver:/# find / -name 1.txt
/home/user/1.txt
root@artemserver:/#
```

Рисунок 18 – Нахождение файлов

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.

## Вывод

В процессе выполнения лабораторной работы был приобретен опыт работы с файлами и каталогами в ОС Linux, настройки прав на доступ к файлам и каталогам. Были выполнены 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.