

Липецкий государственный технический университет

Факультет автоматизации и информатики

Автоматизированные системы управления

ЛАБОРАТОРНАЯ РАБОТА №2

по курсу ДПО Интаро - Linux

«Работа с файловой системой ОС Linux»

Вариант 6-2.

Студент

Митина М. В.

Группа ПИ-20-1

Руководитель
доц.

Кургасов В.В.

Липецк 2022 г.

Цель работы

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

Задание кафедры

1. Ознакомиться с краткими теоретическими сведениями.
2. Приобрести навыки работы в терминале Linux.
3. Научиться создавать новые файлы и каталоги, разобрать назначение прав доступа к файлам и папкам.
4. Подготовить отчет о выполнении работы.

Оглавление

Ход работы	4
Этап 1	4
Этап 2 – Вариант 6	9
Этап 3	12
Этап 4 – Вариант 2	14
Контрольные вопросы	17
Вывод	22
Приложение	23

Ход работы

Этап 1

Перечень основных каталогов Linux.

Теоретические сведения: после установки дистрибутива Linux, в файловом менеджере мы не увидим привычной многим структуры разделов Windows. Файловая система Linux организована иначе, она имеет древовидную структуру, которая, как и любое дерево начинается с корня (точка монтирования «/»). Посмотрим на основные каталоги системы:

/ - корень

/bin (*бинарные файлы пользователя*) - Этот каталог содержит в основном готовые к исполнению программы, большинство из которых необходимы во время старта системы (или в однопользовательском системном режиме, используемом для отладки). Здесь хранится значительное количество общеупотребительных команд Linux.

/boot (*файлы загрузчика*) - Содержит основные постоянные файлы для загрузки системы, в частности загружаемое ядро.

/dev (*файлы устройств*) - Каталог специальных файлов или файлов устройств.

/etc (*конфигурационные файлы*) - В этой папке содержатся конфигурационные файлы всех программ, установленных в системе.

/home (*домашняя папка*) - В этой папке хранятся домашние каталоги всех пользователей.

/lib (*системные библиотеки*) - Этот каталог содержит разделяемые библиотеки функций

/media (*съёмные носители*) - сюда монтируются компакт-диски, флэшки и виндовые разделы

/mnt (*монтирование*) - В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы.

/proc (*информация о процессах*) - Это точка монтирования для файловой системы `proc`, которая обеспечивает информацию о выполняющихся процессах, ядре, оборудовании вычислительной установки и т. д. Это псевдо-файловая система – подсистема, динамически создаваемая ядром.

/root - Это домашний каталог суперпользователя

/sbin (*системные исполняемые файлы*) - Также как и `/bin`, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не примонтирован каталог `/usr`, но здесь находятся программы, которые можно выполнять только с правами суперпользователя.

/tmp (*временные файлы*) - В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию. Файлы удаляются при каждой перезагрузке.

/usr (*программы пользователя*) - Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию.

/var (*переменные файлы*) - Этот каталог содержит файлы, в которых сохраняются различные переменные данные, определяющие конфигурацию некоторых программ при следующем запуске или временно сохраняемую информацию, которая будет использоваться позже в ходе текущего сеанса.

Каталоги в моей ОС Debian:

```
./var/tmp/systemd-private-a68c0eed9a0b43c3b7d041106ce2d60-systemd-timesyncd.service-1s
root@debian:/# dir
bin  etc      initrd.img.old  lib64      media  proc  sbin  tmp  vmlinuz
boot home    lib             libx32     mnt     root  srv   usr  vmlinuz.old
dev  initrd.img lib32           lost+found opt       run   sys  var
```

Выполнение заданий первой части лабораторной работы.

```
mitina_mv@debian:~$ su - root
Пароль:
root@debian:~# _
```

Рисунок 1. Авторизация под root

```
root@debian:~# cd dev
root@debian:/dev# ls
autofs          hidraw0         pts             stderr          tty21           tty38           tty54           uhid            vcsa6
block           hpet            random          stdin           tty22           tty39           tty55           uinput          vcsu
bsg             hugepages       rfkill          stdout          tty23           tty4            tty56           urandom         vcsu1
btrfs-control   initctl         rtc             tty             tty24           tty40           tty57           vboxguest       vcsu2
bus             input           rtc0            tty0            tty25           tty41           tty58           vboxuser        vcsu3
cdrom           kmsg            sda             tty1            tty26           tty42           tty59           vcs             vcsu4
char            log             sda1            tty10           tty27           tty43           tty6            vcs1            vcsu5
console         loop-control    sda2            tty11           tty28           tty44           tty60           vcs2            vcsu6
core            mapper          sda5            tty12           tty29           tty45           tty61           vcs3            vfio
cpu_dma_latency mem             sda6            tty13           tty3            tty46           tty62           vcs4            vga_arbiter
cuse            mqueue         sda7            tty14           tty30           tty47           tty63           vcs5            vhci
disk            net             sda8            tty15           tty31           tty48           tty7            vcs6            vhost-net
dri             null            sg0             tty16           tty32           tty49           tty8            vcsa            vhost-vsock
dvd             nvram           sg1             tty17           tty33           tty5            tty9            vcsa1           zero
fb0             port            shm             tty18           tty34           tty50           ttyS0           vcsa2
fd              ppp             snapshot        tty19           tty35           tty51           ttyS1           vcsa3
full            psaux           snd             tty2            tty36           tty52           ttyS2           vcsa4
fuse            ptmx            sr0             tty20           tty37           tty53           ttyS3           vcsa5
```

Рисунок 2. просмотр папок

```
root@debian:/boot# more vmlinuz-5.10.0-19-amd64
M2++++++1++++@ #t      #»Iëz1LÍÍ'>Ú> #Use  boo | ro  der+

ReLoLe dist  ||td press ||t± res |o reboot+++
PED% K^I^  Ÿ^ ^
<^L+seHpL;L; P`+reroc L= L=@B+coLp|| Ÿ= Ÿ=@B+|e||ô^>mi> P`ÚÚŸÉÚÚÚ-ëJHdrS`8+@Q+++ ?%i+>+++i
+++++9+tt+@0+tt+$+s1+++u++++f+++h+++f>+;U+Z2u+<+Cof1+)++++f+f+f+f+++++8+tt+++f+++f`++ ,+++
                                     ff+fa+++
```

Рисунок 3. просмотр файла

```
root@debian /h/testUser [1]# sudo useradd testUser2
root@debian /home# ls
lost+found/  mitina_mv/  testUser/
root@debian /home# cd testUser/
```

Рисунок 4. создание пользователя

```
root@debian /h/testUser# touch 1.txt
root@debian /h/testUser# cat > 2.txt
строка 1
строка 2
И тебя разбудит опять
Отчаянный звонок посреди ночи
....^Z^J
```

Рисунок 5. создание файлов

```
GNU nano 5.4                               3.txt
Не выходи из комнаты,
Не совершай ошибку
Зачем тебе солнце,
Если ты куришь шипку?

За дверью бессмысленно все
Особенно возглас счастья
Только в уборную - и сразу же возвращайся.

Не выходи из комнаты. Считай, что тебя продуло
Что интереснее на свете стены и стула?

[ Wrote 11 lines ]
Help      Записать  Поиск    Cut       Execute   Location  M-U       Отмена
Выход     Чит.файл  Замена  Paste     Выводить  К строке  M-E       Повтор
```

Рисунок 6. создание файла через команду nano

```
root@debian /h/testUser# ls -al
итого 28
drwxr-xr-x 2 testUser testUser 4096 ноя 13 21:48 ./
drwxr-xr-x 5 root      root    4096 ноя 13 21:20 ../
-rw-r--r-- 1 root      root      0 ноя 13 21:49 1.txt
-rw-r--r-- 1 root      root    125 ноя 13 21:50 2.txt
-rw-r--r-- 1 root      root    476 ноя 13 21:48 3.txt
-rw-r--r-- 1 testUser testUser  220 мар 27  2022 .bash_logout
-rw-r--r-- 1 testUser testUser 3526 мар 27  2022 .bashrc
-rw-r--r-- 1 testUser testUser  807 мар 27  2022 .profile
root@debian /h/testUser# _
```

Рисунок 7. просмотр прав

```
root@debian /h/testUser [1]# chmod 777 1.txt
root@debian /h/testUser# ls -al
итого 28
drwxr-xr-x 2 testUser testUser 4096 ноя 13 21:48 ./
drwxr-xr-x 5 root      root    4096 ноя 13 21:20 ../
-rwxrwxrwx 1 root      root      0 ноя 13 21:49 1.txt*
-rw-r--r-- 1 root      root    125 ноя 13 21:50 2.txt
-rw-r--r-- 1 root      root    476 ноя 13 21:48 3.txt
-rw-r--r-- 1 testUser testUser  220 мар 27  2022 .bash_logout
-rw-r--r-- 1 testUser testUser 3526 мар 27  2022 .bashrc
-rw-r--r-- 1 testUser testUser  807 мар 27  2022 .profile
```

Рисунок 8. изменение прав для файла 1.txt

```

root@debian /h/testUser# ln -s 2.txt 1_2.txt
root@debian /h/testUser# ls -al
итого 28
drwxr-xr-x 2 testUser testUser 4096 ноя 13 22:42 ./
drwxr-xr-x 5 root      root      4096 ноя 13 21:20 ../
-rwxrwxrwx 1 root      root        0 ноя 13 21:49 1.txt*
-rw-r--r-- 1 root      root      125 ноя 13 21:50 2.txt
-rw-r--r-- 1 root      root      476 ноя 13 21:48 3.txt
-rw-r--r-- 1 testUser testUser  220 мар 27  2022 .bash_logout
-rw-r--r-- 1 testUser testUser 3526 мар 27  2022 .bashrc
lrwxrwxrwx 1 root      root        5 ноя 13 22:42 1_2.txt -> 2.txt
-rw-r--r-- 1 testUser testUser  807 мар 27  2022 .profile

```

Рисунок 9. Создание символической ссылки на файл 2.txt

```

root@debian /h/testUser# mkdir newDir
root@debian /h/testUser# ls -al
итого 32
drwxr-xr-x 3 testUser testUser 4096 ноя 13 22:43 ./
drwxr-xr-x 5 root      root      4096 ноя 13 21:20 ../
-rwxrwxrwx 1 root      root        0 ноя 13 21:49 1.txt*
-rw-r--r-- 1 root      root      125 ноя 13 21:50 2.txt
-rw-r--r-- 1 root      root      476 ноя 13 21:48 3.txt
-rw-r--r-- 1 testUser testUser  220 мар 27  2022 .bash_logout
-rw-r--r-- 1 testUser testUser 3526 мар 27  2022 .bashrc
lrwxrwxrwx 1 root      root        5 ноя 13 22:42 1_2.txt -> 2.txt
drwxr-xr-x 2 root      root      4096 ноя 13 22:43 newDir/
-rw-r--r-- 1 testUser testUser  807 мар 27  2022 .profile

```

Рисунок 10. Создание каталога new в директории пользователя testUser.

```

root@debian /h/testUser# cat 3.txt > copy_3.txt
root@debian /h/testUser# mv 2.txt mv_2.txt
root@debian /h/testUser# ls -al
итого 36
drwxr-xr-x 3 testUser testUser 4096 ноя 13 22:45 ./
drwxr-xr-x 5 root      root      4096 ноя 13 21:20 ../
-rwxrwxrwx 1 root      root        0 ноя 13 21:49 1.txt*
-rw-r--r-- 1 root      root      476 ноя 13 21:48 3.txt
-rw-r--r-- 1 testUser testUser  220 мар 27  2022 .bash_logout
-rw-r--r-- 1 testUser testUser 3526 мар 27  2022 .bashrc
-rw-r--r-- 1 root      root      476 ноя 13 22:44 copy_3.txt
lrwxrwxrwx 1 root      root        5 ноя 13 22:42 1_2.txt -> 2.txt
-rw-r--r-- 1 root      root      125 ноя 13 21:50 mv_2.txt
drwxr-xr-x 2 root      root      4096 ноя 13 22:43 newDir/
-rw-r--r-- 1 testUser testUser  807 мар 27  2022 .profile

```

Рисунок 11. Копирование файла 3.txt, перенос файла 2.txt в mv_2.txt

На этом рисунке видно, что симлинк сломался после выполнения команды mv. Сама же команда mv предназначена для изменения расположения файла, однако в данном случае получилось так, что 2.txt был переименован.

```
root@debian /h/testUser# chgrp mitina_mv copy_3.txt
root@debian /h/testUser# chown mitina_mv newDir/
root@debian /h/testUser# ls -al
итого 32
drwxr-xr-x 3 testUser testUser 4096 ноя 13 22:46 ./
drwxr-xr-x 5 root      root      4096 ноя 13 21:20 ../
-rwxrwxrwx 1 root      root        0 ноя 13 21:49 1.txt*
-rw-r--r-- 1 root      root      476 ноя 13 21:48 3.txt
-rw-r--r-- 1 testUser testUser  220 мар 27  2022 .bash_logout
-rw-r--r-- 1 testUser testUser 3526 мар 27  2022 .bashrc
-rw-r--r-- 1 root      mitina_mv  476 ноя 13 22:44 copy_3.txt
lrwxrwxrwx 1 root      root         5 ноя 13 22:42 1_2.txt -> 2.txt
drwxr-xr-x 2 mitina_mv root      4096 ноя 13 22:46 newDir/
-rw-r--r-- 1 testUser testUser  807 мар 27  2022 .profile
```

Рисунок 12. Изменение владельца файла copy_3.txt и каталога newDir на mitina_mv.

```
root@debian /h/testUser# rm -r newDir/
root@debian /h/testUser# ls -al
итого 28
drwxr-xr-x 2 testUser testUser 4096 ноя 13 22:54 ./
drwxr-xr-x 5 root      root      4096 ноя 13 21:20 ../
-rwxrwxrwx 1 root      root        0 ноя 13 21:49 1.txt*
-rw-r--r-- 1 root      root      476 ноя 13 21:48 3.txt
-rw-r--r-- 1 testUser testUser  220 мар 27  2022 .bash_logout
-rw-r--r-- 1 testUser testUser 3526 мар 27  2022 .bashrc
-rw-r--r-- 1 root      mitina_mv  476 ноя 13 22:44 copy_3.txt
lrwxrwxrwx 1 root      root         5 ноя 13 22:42 1_2.txt -> 2.txt
-rw-r--r-- 1 testUser testUser  807 мар 27  2022 .profile
```

Рисунок 13. Удаление каталога newDir/

Этап 2 – Вариант 6

```
root@debian /h/t/2step# tee 2.txt
cdfhksdj
cdfhksdj
jdfh
jdfh
sdjfl
sdjfl
kjslalwei
kjslalwei
kdjalksdj
kdjalksdj
^Z^J
```

Рисунок 14. Создание файла командой tee

```

root@debian /h/t/2step# ls -al
итого 16
drwxr-xr-x 2 root      root      4096 ноя 13 23:04 ./
drwxr-xr-x 3 testUser testUser 4096 ноя 13 22:58 ../
-rw-r--r-- 1 root      root       151 ноя 13 22:59 1.txt
-rw-r--r-- 1 root      root        40 ноя 13 23:01 2.txt
-rw-r--r-- 1 root      root         0 ноя 13 23:03 3.txt

```

Рисунок 15. Созданные файлы для задания

```

root@debian /h/t/2/1/2lv1# mkdir 3lv1
root@debian /h/t/2/1/2lv1# cd 3lv1/
root@debian /h/t/2/1/2/3lv1# mkdir 4_1lv1 4_2lv1
root@debian /h/t/2/1/2/3lv1# ls
4_1lv1/ 4_2lv1/

```

Рисунок 16. созданная структура

```

root@debian /h/t/2step# ls -R
.:
1lv1/

./1lv1:
1.txt 2lv1/ 2.txt 3.txt

./1lv1/2lv1:
3lv1/

./1lv1/2lv1/3lv1:
4_1lv1/ 4_2lv1/

./1lv1/2lv1/3lv1/4_1lv1:

./1lv1/2lv1/3lv1/4_2lv1:
root@debian /h/t/2step# cd 1lv1/
root@debian /h/t/2/1lv1# mv 2.txt 2lv1/
root@debian /h/t/2/1lv1# mv 3.txt 2lv1/3lv1/4_1lv1/
root@debian /h/t/2/1lv1# ls -R
.:
1.txt 2lv1/

./2lv1:
2.txt 3lv1/

./2lv1/3lv1:
4_1lv1/ 4_2lv1/

./2lv1/3lv1/4_1lv1:
3.txt

./2lv1/3lv1/4_2lv1:

```

Рисунок 17. полный обзор созданной структуры

```

root@debian /h/t/2/1lv1 [1]# ln -P 1.txt 2lv1/3lv1/4_2lv1/ln_1_p.txt
root@debian /h/t/2/1lv1# ls -al
итого 16
drwxr-xr-x 3 root root 4096 ноя 14 09:11 ./
drwxr-xr-x 3 root root 4096 ноя 13 23:13 ../
-rw-r--r-- 4 root root 151 ноя 13 22:59 1.txt
drwxr-xr-x 3 root root 4096 ноя 13 23:14 2lv1/
root@debian /h/t/2/1lv1# ls -R
.:
1.txt 2lv1/

./2lv1:
2.txt 3lv1/

./2lv1/3lv1:
4_1lv1/ 4_2lv1/

./2lv1/3lv1/4_1lv1:
3.txt copy_1.txt

./2lv1/3lv1/4_2lv1:
ln_1_1.txt ln_1_p.txt ln_1.txt

```

Рисунок 18. создание жесткой ссылки на файл 1.txt в каталог
2lv1/3lv1/4_2lv1/

```

root@debian /h/t/2/1/2lv1# ln -s /home/testUser/2step/1lv1/2lv1/3lv1/4_1lv1/3.txt 3lv1/4_2lv1/ln_3_s
.txt
root@debian /h/t/2/1/2lv1# ls -liR
.:
итого 8
890156 -rw-r--r-- 1 root root 40 ноя 13 23:01 2.txt
890162 drwxr-xr-x 4 root root 4096 ноя 13 23:07 3lv1/

./3lv1:
итого 8
890163 drwxr-xr-x 2 root root 4096 ноя 13 23:16 4_1lv1/
890164 drwxr-xr-x 2 root root 4096 ноя 14 09:23 4_2lv1/

./3lv1/4_1lv1:
итого 4
890159 -rw-r--r-- 1 root root 0 ноя 13 23:03 3.txt
890165 -rw-r--r-- 1 root root 151 ноя 13 23:16 copy_1.txt

./3lv1/4_2lv1:
итого 12
890157 -rw-r--r-- 4 root root 151 ноя 13 22:59 ln_1_1.txt
890157 -rw-r--r-- 4 root root 151 ноя 13 22:59 ln_1_p.txt
890157 -rw-r--r-- 4 root root 151 ноя 13 22:59 ln_1.txt
890169 lrwxrwxrwx 1 root root 36 ноя 14 09:21 ln_2_3_s.txt -> /home/testUser/2step/1lv1/2lv1/2.txt
890170 lrwxrwxrwx 1 root root 48 ноя 14 09:23 ln_3_s.txt -> /home/testUser/2step/1lv1/2lv1/3lv1/4_1
lv1/3.txt
890168 lrwxrwxrwx 1 root root 5 ноя 14 09:16 ln_s_2_2.txt -> 2.txt
890167 lrwxrwxrwx 1 root root 5 ноя 14 09:17 ln_s_2_3.txt -> 2.txt
890166 lrwxrwxrwx 1 root root 10 ноя 14 09:13 ln_s_2.txt -> 2lv1/2.txt

```

Рисунок 19. создание символических ссылок и результирующая
структура каталогов согласно варианта 6.

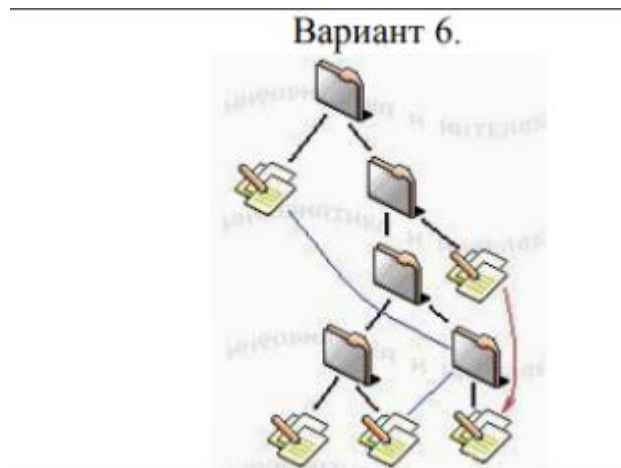


Рисунок 20. ожидаемый результат по варианту 6

```
root@debian /h/t/2step# rm -r 1lv1/
root@debian /h/t/2step# ls -liR
.:
итого 0
```

Рисунок 21. рекурсивное удаление созданной структуры.

Этап 3

```
root@debian ~# useradd -m MariaPI201
root@debian ~# passwd
Новый пароль:
Повторите ввод нового пароля:
Пароль не указан.
Новый пароль:
Повторите ввод нового пароля:
Пароль не указан.
Новый пароль:
Повторите ввод нового пароля:
Извините, но пароли не совпадают.
passwd: Ошибка при операциях с маркером проверки подлинности
passwd: пароль не изменён
root@debian ~ [10]# pwd
/root
root@debian ~# passwd MariaPI201
Новый пароль:
Повторите ввод нового пароля:
Извините, но пароли не совпадают.
passwd: Ошибка при операциях с маркером проверки подлинности
passwd: пароль не изменён
root@debian ~ [10]# passwd MariaPI201
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
```

Рисунок 22. создание пользователя, изменение его пароля

```
root@debian:~# su - MariaPI201
$ _
```

Рисунок 23. изменение пользователя с root на нового.

```
$ ls
1.txt 2.txt
$ mv 1.txt /root/
mv: не удалось выполнить stat для '/root/1.txt': Отказано в доступе
```

Рисунок 24. созданные в каталоге пользователя файлы и попытка переноса одного из них в директорию рута.

```
$ su - root
Пароль:
root@debian:~# su - MariaPI201
$ su - mitina_mv
Пароль:
mitina_mv@debian:~$ sudo -s /usr/bin/fish
[sudo] пароль для mitina_mv:
Welcome to fish, the friendly interactive shell
Type `help` for instructions on how to use fish
root@debian /h/mitina_mv# touch newfile.txt
root@debian /h/mitina_mv# ls
1.txt newfile.txt
root@debian /h/mitina_mv# mv newfile.txt /home/MariaPI201/
root@debian /h/mitina_mv# cd ../MariaPI201/
root@debian /h/MariaPI201# ls
1.txt 2.txt newfile.txt
root@debian /h/MariaPI201# _
```

Рисунок 25. изменение пользователя на пользователя, входящего в sudoers, создание файла в директории другого пользователя.

```
GNU nano 5.4 newfile.txt *
#!/bin/bash
echo "This is page Митина Мария Валерьевна"
echo "Редактировал пользователь mitina_mvЭ"
```

Рисунок 26. создание bash скрипта средствами nano.

```
root@debian /h/MariaPI201# chmod 4777 newfile
root@debian /h/MariaPI201# ls -li
итого 12
130057 -rw-r--r-- 1 MariaPI201 MariaPI201 73 ноя 14 14:22 1.txt
130058 -rw-r--r-- 1 MariaPI201 MariaPI201 116 ноя 14 14:23 2.txt
130059 -rwsrwxrwx 1 root root 146 ноя 14 14:30 newfile*
520198 -rw-r--r-- 1 root root 0 ноя 14 14:26 newfile.txt
```

Рисунок 27. установка исполнимости для созданного скрипта.

```
root@debian /h/MariaPI201# sh newfile
This is page Митина Мария Валерьевна
Редактировал пользователь mitina_mvЭ
```

Рисунок 28. Выполнение исполняемого файла.

Этап 4 – Вариант 2

1) Создать архив arh2.tar.gz, состоящий из нескольких файлов. Вывести список файлов архива. Вывести содержимое файлов архива, без его распаковки. Распаковать архив.

2) В указанном каталоге (без обработки подкаталогов) найти все обычные файлы, имеющие расширение.

```
root@debian /h/MariaPI201# ls -li
итого 12
130057 -rw-r--r-- 1 MariaPI201 MariaPI201  73 ноя 14 14:22 1.txt
130058 -rw-r--r-- 1 MariaPI201 MariaPI201 116 ноя 14 14:23 2.txt
130059 -rwsrwxrwx 1 root      root      146 ноя 14 14:30 newfile*
520198 -rw-r--r-- 1 root      root       0 ноя 14 14:26 newfile.txt
root@debian /h/MariaPI201# tar -cvf ../MariaPI201/
tar: Пробный отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
root@debian /h/MariaPI201 [2]# cd ../
root@debian /home# tar -cvfz ~/MariaPI201/
tar: Удаляется начальный '/' из имен объектов
tar: /root/MariaPI201: функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: Завершение работы с состоянием неисправности из-за возникших ошибок
```

Рисунок 29. структура каталога, который подлежит архивации.

Попытка архивации и ошибки.

```
root@debian /home# tar -zcvf arh2.tar.gz ~/MariaPI201/
tar: Удаляется начальный '/' из имен объектов
/home/MariaPI201/
/home/MariaPI201/newfile
/home/MariaPI201/.local/
/home/MariaPI201/.local/share/
/home/MariaPI201/.local/share/nano/
/home/MariaPI201/.bash_logout
/home/MariaPI201/2.txt
/home/MariaPI201/1.txt
/home/MariaPI201/newfile.txt
/home/MariaPI201/.bashrc
/home/MariaPI201/.profile
root@debian /home# ls
arh2.tar.gz  lost+found/  MariaPI201/  mitina_mv/  z
```

Рисунок 30. успешное создание архива.

```

root@debian /home# tar -t arh2.tar.gz
tar: Отказ чтения содержимого архива с терминала (пропущен параметр -f?)
tar: Error is not recoverable: exiting now
root@debian /home [2]# tar -tf arh2.tar.gz
home/MariaPI201/
home/MariaPI201/newfile
home/MariaPI201/.local/
home/MariaPI201/.local/share/
home/MariaPI201/.local/share/nano/
home/MariaPI201/.bash_logout
home/MariaPI201/2.txt
home/MariaPI201/1.txt
home/MariaPI201/newfile.txt
home/MariaPI201/.bashrc
home/MariaPI201/.profile

```

Рисунок 31. просмотр содержимого архива.

```

root@debian /home [2]# tar -xCf arh2.tar.gz arh2/
tar: Отказ чтения содержимого архива с терминала (пропущен параметр -f?)
tar: Error is not recoverable: exiting now
root@debian /home [2]# tar -xvf arh2.tar.gz arh2/
tar: C: Функция open завершилась с ошибкой: Нет такого файла или каталога
tar: Error is not recoverable: exiting now
root@debian /home [2]# tar -xvf arh2.tar.gz
home/MariaPI201/
home/MariaPI201/newfile
home/MariaPI201/.local/
home/MariaPI201/.local/share/
home/MariaPI201/.local/share/nano/
home/MariaPI201/.bash_logout
home/MariaPI201/2.txt
home/MariaPI201/1.txt
home/MariaPI201/newfile.txt
home/MariaPI201/.bashrc
home/MariaPI201/.profile
root@debian /home# ls
arh2.tar.gz  home/  lost+found/  MariaPI201/  mitina_mv/  z
root@debian /home# cd home/
root@debian /h/home# ls
MariaPI201/

```

Рисунок 32. разархивация архива (создалась директория home, в ней директория MariaPI201)

```

root@debian /home# mkdir arh2
root@debian /home# tar -xvf arh2.tar.gz -C ach2
tar: ach2: функция open завершилась с ошибкой: Нет такого файла или каталога
tar: Error is not recoverable: exiting now
root@debian /home [2]# ls
arh2/  arh2.tar.gz  home/  lost+found/  MariaPI201/  mitina_mv/  z
root@debian /home# tar -xvf arh2.tar.gz -C arh2
home/MariaPI201/
home/MariaPI201/newfile
home/MariaPI201/.local/
home/MariaPI201/.local/share/
home/MariaPI201/.local/share/nano/
home/MariaPI201/.bash_logout
home/MariaPI201/2.txt
home/MariaPI201/1.txt
home/MariaPI201/newfile.txt
home/MariaPI201/.bashrc
home/MariaPI201/.profile
root@debian /home# ls
arh2/  arh2.tar.gz  home/  lost+found/  MariaPI201/  mitina_mv/  z

```

Рисунок 33. распаковка архива в указанную директорию.

```

root@debian /home# cd arh2/
root@debian /h/arh2# ls
home/
root@debian /h/arh2# cd home/MariaPI201/
root@debian /h/a/h/MariaPI201# ls
1.txt 2.txt newfile* newfile.txt
root@debian /h/a/h/MariaPI201# ls -l --color *.*
-rw-r--r-- 1 MariaPI201 MariaPI201 73 ноя 14 14:22 1.txt
-rw-r--r-- 1 MariaPI201 MariaPI201 116 ноя 14 14:23 2.txt
-rw-r--r-- 1 root      root      0 ноя 14 14:26 newfile.txt

```

Рисунок 34. Поиск среди распакованных файлов обычных (тех. которые имеют расширение) через ls.

```

root@debian /h/a/h/MariaPI201# find -maxdepth 1 -perm /u=r-x -name "*"
.
./.local
./.bash_logout
./2.txt
./1.txt
./newfile.txt
./.bashrc
./.profile

```

Рисунок 35. Результат по команде find по регулярному выражению.

Контрольные вопросы

1. Что такое «файловая система»?

Файловая система – это организованный порядок, определяющий набор правил для безопасного расположения, хранения и последующего доступа к разнообразным данным на запоминающих хранилищах информации в компьютерных и других устройствах, содержащих цифровой накопитель.

2. Права доступа к файлам, назначение прав доступа.

Изначально каждый файл имел три параметра доступа:

- 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.

Чтобы изменить права на файл в linux можно использовать команду `chmod`. Она позволяет менять все флаги, включая специальные. Команда может применяться с категориями, выраженными буквами (**u** - владелец файла; **g** - группа файла; **o** - другие пользователи) или в числовом значении.

3. Что такое «символическая ссылка»?

Символическая ссылка (symbolic link) — это специальный файл, который является ссылкой на другой файл или каталог.

4. Что такое «жесткая ссылка»?

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.

5. Команда поиска в Linux. Основные сведения.

`Find` – 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. Команда выглядит следующим образом:

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

Папка – каталог в котором будем искать.

Параметры – дополнительные параметры для поиска.

Критерий – критерий поиска (например, дата или права).

Шаблон – фильтрующее значение файлов.

Рассмотрим основные критерии:

- `name` - поиск файлов по имени;
- `perm` - поиск файлов в Linux по режиму доступа;
- `user` - поиск файлов по владельцу;
- `group` - поиск по группе;
- `mtime` - поиск по времени модификации файла;
- `atime` - поиск файлов по дате последнего чтения;
- `nogroup` - поиск файлов, не принадлежащих ни одной группе;
- `nouser` - поиск файлов без владельцев;
- `newer` - найти файлы новее чем указанный;
- `size` - поиск файлов в Linux по их размеру.

6. Перечислите основные команды для работы с каталогами.

Для того чтобы узнать, где располагается каталог, в котором мы сейчас находимся, используется команда: `pwd`. Просмотреть содержимое любой директории можно следующим способом: `ls /путь_к_директории`. Для перехода между каталогами используется команда `cd`. Создание каталогов выполняется с помощью команды вида: `mkdir`. Для удаления директорий используется команда `rmdir`. Для удаления директории с именем `dir1` со всеми подкаталогами и файлами используется: `rm -r dir1`. Для перемещения и переименования файлов и каталогов используется команда `mv`. Для копирования используется команда `cp`.

7. Чем отличается вывод команд `ls -F` от `ls -al`?

`-F` - показывать тип объекта, к каждому объекту будет добавлен один из специализированных символов `*/=>@|;`;

`-l` - выводить подробный список, в котором будет отображаться владелец, группа, дата создания, размер и другие параметры;

`-a` - отображать все файлы, включая скрытые, это те, перед именем которых стоит точка;

8. С помощью какой команды можно переместить файл в другой каталог?

Это можно осуществить с помощью команды `mv`.

9. Куда вы переходите, выполнив команду `cd` без параметров?

В домашнюю папку пользователя.

10. Как осуществить просмотр подкаталогов и их содержимого?

С помощью команды `ls -R`, где `R` отвечает за рекурсивный вывод.

11. Как осуществить создание нового каталога и необходимых подкаталогов рекурсивно?

Для этого нужно при вызове команды `mkdir` указать параметр `-p`.

Команда будет выглядеть так: `mkdir -p /путь_к_директории/`.

12. Как осуществить рекурсивное копирование всех файлов из одного каталога в другой?

С помощью комбинации команд `find` и `cp` и работы с потоком.

13. Как осуществить рекурсивное копирование всех файлов и подкаталогов из одного каталога в другой?

С помощью команды `cp -r /что_копируем/ /куда_копируем/`.

14. Как рекурсивно удалить все файлы и подкаталоги в определенном каталоге?

Чтобы это сделать, нам нужно перейти в директорию, которую планируется очистить и ввести команду `rm -r *`.

15. Перечислите основные ключи команды `ls` с их назначением.

Основные параметры:

- `a` – Показывает все файлы, в том числе скрытые с точкой.
- `l` – Отображает подробный список, в котором показываются права доступа, владелец и т.д.

- d – Отображает только каталоги, без содержимого, не переходить по символьным ссылкам.
- f. Аргумент отключит сортировку и покажет скрытые документы.
- r. Сортировка в обратном порядке.
- R - рекурсивно отображать содержимое поддиректорий;
- X. Сортировка по алфавиту.
- --help. Справка.
- L - для символических ссылок отображать информацию о файле, на который они ссылаются;

16. Команды tee и cat. Назначение и применение. Чем отличается cat от more и less?

Команда tee в Linux нужна для записи вывода любой команды в один или несколько файлов. С помощью команды cat можно очень просто посмотреть содержимое небольшого файла, склеить несколько файлов.

И cat, и more, и less выводят текст, но cat больше подходит для малого вывода, а more и less – для многостраничных текстов. More – листает только в одну сторону, а less позволяет перемещаться между страницами во всех направлениях.

Вывод

В ходе выполнения работы я ознакомилась с краткими теоретическими сведениями, приобрела навыки работы в терминале, научилась создавать файлы и каталоги в терминале, разобралась в назначении прав доступа.

Приложение

Несколько скриншотов установки Debian.

