

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра Автоматизированных систем управления**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**по дисциплине «Операционная система Linux»**

**Работа с файловой системой ОС Linux**

**Вариант 4**

Студентка

Закиров Р.Р.

Группа АС-20-1

Руководитель

Кургасов В.В.

К.П.Н.

Липецк 2022

## Оглавление

Цель работы .....	3
Задание кафедры.....	4
Ход работы .....	6
Вывод.....	34
Контрольные вопросы.....	34

## 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 Debian.
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.

## Ход работы

### ЭТАП 1

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

По умолчанию `sudo` не установлена в Debian, но вы можете установить её.

Вначале включите режим `su`: Установите `sudo` запустив:

```
rolan@debian:/$ su -
Пароль:
root@debian:~# apt install sudo
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  sudo
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не об
новлено.
Необходимо скачать 1 059 кВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 4 699 кВ.
Пол:1 http://deb.debian.org/debian bullseye/main amd64 sudo amd64 1.9.5p2-3 [1 059 кВ]
Получено 1 059 кВ за 0с (2 575 кВ/с)
Выбор ранее не выбранного пакета sudo.
(Чтение базы данных ... на данный момент установлено 38684 файла и каталога.)
Подготовка к распаковке .../sudo_1.9.5p2-3_amd64.deb ...
Распаковывается sudo (1.9.5p2-3) ...
Настраивается пакет sudo (1.9.5p2-3) ...
Обрабатываются триггеры для man-db (2.9.4-2) ...
root@debian:~# _
```

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@debian:/# ls
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
```

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. /dev – содержит файлы устройств;
4. /etc – содержит конфигурационные файлы операционной системы и всех сетевых служб;
5. /home – содержит домашние каталоги всех пользователей, зарегистрированных в системе;
6. /initrd.img.old - Временная файловая система, используемая ядром Linux при начальной загрузке. initrd обычно используется для начальной инициализации перед монтированием «настоящих» файловых систем.
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 FHS2.3);
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. /srv – параметры, которые специфичные для окружения системы, чаще всего данная директория пуста;

18. /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;

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

20. /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;

21. /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;

22. /vmlinuz - это сжатое ядро Linux, и оно является загрузочным.

Загрузочный означает, что он способен загружать операционную систему в память, чтобы компьютер стал пригодным для использования и можно было запускать прикладные программы.

Посмотрим содержимое каталога файлов физических устройств (/dev). На рисунке 3 приведён перечень файлов физических устройств.

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

Рисунок 3 – Содержимое каталога файлов физических устройств

Укажем назначения файлов

1. aspi\_thermal\_rel – обеспечивает функции управления температурой модуля ACPI;

2. `autofs` – система управления автоматическим монтированием (и отмонтированием).
3. `btrfs-control` – устройство 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. nvram – 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. rfkill – 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 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@debian:/# ls
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
root@debian:/# cat vmlinuz_
```

Рисунок 4 – Содержимое каталог

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

```

3ERROR: Failed to load initrd!
3ERROR: Failed to allocate memory for 'rom'
3ERROR: Failed to read rom->vendor
3ERROR: Failed to read rom->devid
3ERROR: Unsupported properties proto version
3ERROR: Failed to allocate memory for 'properties'
3ERROR: efi_main() failed!
3ERROR: exit_boot() failed!
3ERROR: Failed to allocate usable memory for kernel.
6UEFI Secure Boot is enabled.
3ERROR: Could not determine UEFI Secure Boot status.
Spec ID Event033ERROR: Unable to allocate memory for event log
***0**10`He0\*****,*0** *H**
+*7*NOL0
+*70 *H****eld**r+P**b**A}*B*&*u****kr*G0*C0**+2*(+o****e*:B&C0
3208151732372011/0-Uoot CA0
*0* *H** &Debian Secure Boot Signer 2022 - linux0**0
***~*
$S**'+2*6+~67*(Wx***mU*P~w*pI*****
*****}***('7h+/.s+|~*~&+p**|**x )** B0+'=
*zaY.*6**U*+t{**A**o\**4*H7"***t*IA* *Q***<*cJ***H***7} *_1***4**}
H*J>***;**B*j%****|* TK*d0b0
U*0U%
0
*Š5ô±■ÄN#iKЯ H`KЯ€+•üг#£L»s`Z*+Ji%£Г°»ВлӨЦπ±s%3CkX*,,%Я/ᄂᄃᄄ-B»·ь±л^iГLó"NHÜ+ГгÜ≥ċ`L&)μӨГ±·X±JŮр%†°'GĚ
`%He πċ100**Üгi*H**1†ŷ*ÜB 1UDebi|+ Secre Boo CA2 (ľ•oċĀŷeĀ:μ3B&C
D102207*H**} **;CEs**W*F**Y{***D<8eqg***0
*`*=h**n**}
7***7*Y^*b*_**-*a*I**{*****}`PTi)*c(*#nrR*<
*:***1h*>[c***U *M**
$&1**H**}***X**D+gV*!c*$*vD** fhJ-v**R**1* * **}Dn***1**4***&*'*de***J***z*****F\1**S$**xK!* **}
qLJ*/m**/μ*g*#3**

```

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



```
root@debian:/# useradd -m user
root@debian:/# cd home
root@debian:/home# ls
rolan user
root@debian:/home# ls -l
итого 8
drwxr-xr-x 2 rolan rolan 4096 ноя 12 23:54 rolan
drwxr-xr-x 2 user user 4096 ноя 13 21:23 user
```

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@debian:/home# cd user/
root@debian:/home/user# ls -l
итого 0
root@debian:/home/user# touch 1.txt
root@debian:/home/user# touch 2.txt
root@debian:/home/user# touch 3.txt
root@debian:/home/user# ls -l
итого 0
-rw-r--r-- 1 root root 0 ноя 13 21:28 1.txt
-rw-r--r-- 1 root root 0 ноя 13 21:28 2.txt
-rw-r--r-- 1 root root 0 ноя 13 21:28 3.txt
```

Рисунок 7 – Пример работы

Просмотрим и поясним права доступа к файлам. Владелец 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@debian:/home/user# cd /
root@debian:/# cd root
root@debian:~# ls
root@debian:~# ls -a
.  ..  .bashrc  .local  .profile
root@debian:~#
```

Рисунок 9 – Каталог root

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

```
root@debian:/# chmod 777 /home/user/1.txt
root@debian:/# cd home/user/
root@debian:/home/user# ls -l
итого 0
-rwxrwxrwx 1 root root 0 ноя 13 21:28 1.txt
-rw-r--r-- 1 root root 0 ноя 13 21:28 2.txt
-rw-r--r-- 1 root root 0 ноя 13 21:28 3.txt
```

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. Пример

выполнения представлен на рисунка 11 и 12.

```
root@debian:/home/user# ln 2.txt hardlink
root@debian:/home/user# cd /
root@debian:/# ls -l
итого 60
lrwxrwxrwx 1 root root 7 ноя 12 23:45 bin -> usr/bin
drwxr-xr-x 3 root root 4096 ноя 12 23:55 boot
drwxr-xr-x 17 root root 3140 ноя 13 00:00 dev
drwxr-xr-x 72 root root 4096 ноя 13 21:23 etc
drwxr-xr-x 4 root root 4096 ноя 13 21:23 home
lrwxrwxrwx 1 root root 31 ноя 12 23:49 initrd.img -> boot/initrd.img-5.10.0-19-amd64
lrwxrwxrwx 1 root root 31 ноя 12 23:47 initrd.img.old -> boot/initrd.img-5.10.0-18-amd64
lrwxrwxrwx 1 root root 7 ноя 12 23:45 lib -> usr/lib
lrwxrwxrwx 1 root root 9 ноя 12 23:45 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 ноя 12 23:45 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 ноя 12 23:45 libx32 -> usr/libx32
drwx----- 2 root root 16384 ноя 12 23:45 lost+found
drwxr-xr-x 3 root root 4096 ноя 12 23:45 media
drwxr-xr-x 2 root root 4096 ноя 12 23:45 mnt
drwxr-xr-x 2 root root 4096 ноя 12 23:45 opt
dr-xr-xr-x 140 root root 0 ноя 12 23:56 proc
drwx----- 3 root root 4096 ноя 13 21:29 root
drwxr-xr-x 17 root root 500 ноя 13 20:33 run
lrwxrwxrwx 1 root root 8 ноя 12 23:45 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 ноя 12 23:45 srv
dr-xr-xr-x 13 root root 0 ноя 12 23:56 sys
drwxrwxrwt 10 root root 4096 ноя 12 23:56 tmp
drwxr-xr-x 14 root root 4096 ноя 12 23:45 usr
drwxr-xr-x 12 root root 4096 ноя 12 23:53 var
lrwxrwxrwx 1 root root 28 ноя 12 23:49 vmlinuz -> boot/vmlinuz-5.10.0-19-amd64
lrwxrwxrwx 1 root root 28 ноя 12 23:47 vmlinuz.old -> boot/vmlinuz-5.10.0-18-amd64
```

Рисунок 11 – Создание жёсткой ссылки

```
root@debian:/# ln -s /home/user/2.txt slink
root@debian:/# ls -l
итого 60
lrwxrwxrwx 1 root root 7 ноя 12 23:45 bin -> usr/bin
drwxr-xr-x 3 root root 4096 ноя 12 23:55 boot
drwxr-xr-x 17 root root 3140 ноя 13 00:00 dev
drwxr-xr-x 72 root root 4096 ноя 13 21:23 etc
drwxr-xr-x 4 root root 4096 ноя 13 21:23 home
lrwxrwxrwx 1 root root 31 ноя 12 23:49 initrd.img -> boot/initrd.img-5.10.0-19-amd64
lrwxrwxrwx 1 root root 31 ноя 12 23:47 initrd.img.old -> boot/initrd.img-5.10.0-18-amd64
lrwxrwxrwx 1 root root 7 ноя 12 23:45 lib -> usr/lib
lrwxrwxrwx 1 root root 9 ноя 12 23:45 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 ноя 12 23:45 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 ноя 12 23:45 libx32 -> usr/libx32
drwx----- 2 root root 16384 ноя 12 23:45 lost+found
drwxr-xr-x 3 root root 4096 ноя 12 23:45 media
drwxr-xr-x 2 root root 4096 ноя 12 23:45 mnt
drwxr-xr-x 2 root root 4096 ноя 12 23:45 opt
dr-xr-xr-x 140 root root 0 ноя 12 23:56 proc
drwx----- 3 root root 4096 ноя 13 21:29 root
drwxr-xr-x 17 root root 500 ноя 13 20:33 run
lrwxrwxrwx 1 root root 8 ноя 12 23:45 sbin -> usr/sbin
lrwxrwxrwx 1 root root 16 ноя 13 21:57 slink -> /home/user/2.txt
drwxr-xr-x 2 root root 4096 ноя 12 23:45 srv
dr-xr-xr-x 13 root root 0 ноя 12 23:56 sys
drwxrwxrwt 10 root root 4096 ноя 12 23:56 tmp
drwxr-xr-x 14 root root 4096 ноя 12 23:45 usr
drwxr-xr-x 12 root root 4096 ноя 12 23:53 var
lrwxrwxrwx 1 root root 28 ноя 12 23:49 vmlinuz -> boot/vmlinuz-5.10.0-19-amd64
lrwxrwxrwx 1 root root 28 ноя 12 23:47 vmlinuz.old -> boot/vmlinuz-5.10.0-18-amd64
```

Рисунок 12 – Создание символической ссылки

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

```
root@debian:/# cd /
root@debian:/# mkdir /home/user/new
root@debian:/# cd /home/user/
root@debian:/home/user# ls -l
итого 4
-rwxrwxrwx 1 root root    0 ноя 13 21:28 1.txt
-rw-r--r-- 2 root root    0 ноя 13 21:28 2.txt
-rw-r--r-- 1 root root    0 ноя 13 21:28 3.txt
-rw-r--r-- 2 root root    0 ноя 13 21:28 hardlink
drwxr-xr-x 2 root root 4096 ноя 13 22:18 new
```

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 рисунке 14.

```
root@debian:/home/user# ls -l
итого 4
-rwxrwxrwx 1 root root    0 ноя 13 21:28 1.txt
-rw-r--r-- 2 root root    0 ноя 13 21:28 2.txt
-rw-r--r-- 1 root root    0 ноя 13 21:28 3.txt
-rw-r--r-- 2 root root    0 ноя 13 21:28 hardlink
drwxr-xr-x 2 root root 4096 ноя 13 22:18 new
root@debian:/home/user# cp 1.txt new
root@debian:/home/user# cd new
root@debian:/home/user/new# ls
1.txt
root@debian:/home/user/new# cd /
root@debian:/# cd home/user/
root@debian:/home/user# ls
1.txt 2.txt 3.txt hardlink new
root@debian:/home/user# mv 2.txt new
root@debian:/home/user# ls
1.txt 3.txt hardlink new
root@debian:/home/user# cd new/
root@debian:/home/user/new# ls
1.txt 2.txt
root@debian:/home/user/new# _
```

Рисунок 14 – Копирование файла и перемещение файла



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

```
-rwxrwxrwx 1 root root    0 ноя 13 21:28 1.txt
-rw-r--r-- 1 root root    0 ноя 13 21:28 3.txt
drwxr-xr-x 2 root root 4096 ноя 13 22:27 new
root@debian:/home/user# chown user /home/user/3.txt
root@debian:/home/user# ls -l
итого 4
-rwxrwxrwx 1 root root    0 ноя 13 21:28 1.txt
-rw-r--r-- 1 user root    0 ноя 13 21:28 3.txt
drwxr-xr-x 2 root root 4096 ноя 13 22:27 new
root@debian:/home/user# chown user /home/user/new
root@debian:/home/user# ls -l
итого 4
-rwxrwxrwx 1 root root    0 ноя 13 21:28 1.txt
-rw-r--r-- 1 user root    0 ноя 13 21:28 3.txt
drwxr-xr-x 2 user root 4096 ноя 13 22:27 new
```

Рисунок 15 – 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@debian:/home/user# rm new/1.txt
root@debian:/home/user# ls new
2.txt
root@debian:/home/user# rm new
rm: невозможно удалить 'new': Это каталог
root@debian:/home/user# rm -R new
root@debian:/home/user# ls
1.txt 3.txt
```

Рисунок 16 – Удаление файла и директории

## ЭТАП 2

Необходимо создать 3 текстовых файла разными способами. Пример выполнения на рисунке 17.

```
"1.txt" [New File] 3 lines, 83 bytes written
root@debian:/home/user# ls
1.txt
root@debian:/home/user# cat 1.txt
I love winter! Winter is cold.
Is's snowy hills with sled.
It's warm snuggly beds.
root@debian:/home/user# _
```

Рисунок 17 – Создание файла при помощи vi редактора

```
root@debian:/home/user# tee 2.txt
All kinds of food I like to eat -
All kinds of food I like to eat -
Fruits, vegetables, fish and meet...
Fruits, vegetables, fish and meet...
I'm fond of sausages and cheese
I'm fond of sausages and cheese
^C
root@debian:/home/user# cat 2.txt
All kinds of food I like to eat -
Fruits, vegetables, fish and meet...
I'm fond of sausages and cheese
root@debian:/home/user# _
```

Рисунок 18 – Создание файла при помощи tee редактора

Далее, создаем структуру каталогов в соответствии с вариантом(4).

```

root@debian:/home/user# ls
1.txt 2.txt 3.txt 4.txt level
root@debian:/home/user# cd level/
root@debian:/home/user/level# ls -l
итого 8
drwxr-xr-x 2 root root 4096 ноя 15 21:14 level2
drwxr-xr-x 4 root root 4096 ноя 15 21:04 level3
root@debian:/home/user/level# cd level2
root@debian:/home/user/level/level2# ls -l
итого 4
-rw-r--r-- 1 root root 83 ноя 15 21:14 1.txt
root@debian:/home/user/level/level2# cd ..
root@debian:/home/user/level# cd level3
root@debian:/home/user/level/level3# ls -l
итого 8
drwxr-xr-x 2 root root 4096 ноя 15 21:15 level4
drwxr-xr-x 2 root root 4096 ноя 15 22:30 level5
root@debian:/home/user/level/level3# cd level4
root@debian:/home/user/level/level3/level4# ls -l
итого 8
-rw-r--r-- 1 root root 104 ноя 15 21:14 2.txt
-rw-r--r-- 1 root root 181 ноя 15 21:15 3.txt
root@debian:/home/user/level/level3/level4# cd ..
root@debian:/home/user/level/level3# cd level5
root@debian:/home/user/level/level3/level5# ls -l
итого 4
-rw-r--r-- 1 root root 5 ноя 15 22:30 4.txt
root@debian:/home/user/level/level3/level5# _

```

Рисунок 19 – Созданная структура каталогов

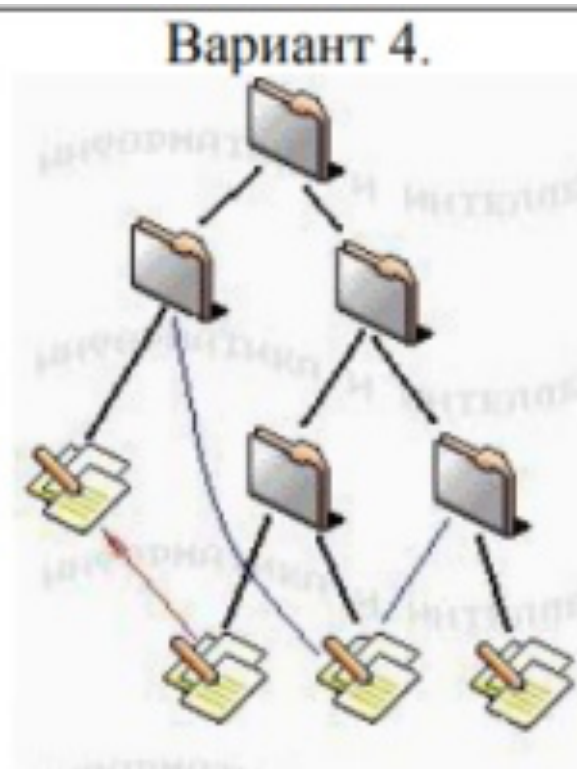


Рисунок 20 – Вариант для выполнения задания

```

root@debian:/home/user/level# cd level2
root@debian:/home/user/level/level2# ln /home/user/level/level3/level4/3.txt thirdhardlink
root@debian:/home/user/level/level2# cd ..
root@debian:/home/user/level# cd level3
root@debian:/home/user/level/level3# cd level5
root@debian:/home/user/level/level3/level5# ln /home/user/level/level3/level4/3.txt secondhardlink

```

Рисунок 21 – Создание ссылок(синие линии)

```

root@debian:/home/user/level/level3/level4# ln -s /home/user/level/1.txt thirdlinkone
root@debian:/home/user/level/level3/level4# cd ..
root@debian:/home/user/level/level3# cd ..
root@debian:/home/user/level# ls -l
итого 8
drwxr-xr-x 2 root root 4096 ноя 15 22:51 level2
drwxr-xr-x 4 root root 4096 ноя 15 21:04 level3
root@debian:/home/user/level# cd ..
root@debian:/home/user# ls -l
итого 20
-rw-r--r-- 1 root root 83 ноя 14 21:31 1.txt
-rw-r--r-- 1 root root 104 ноя 14 21:41 2.txt
-rw-r--r-- 1 root root 181 ноя 15 20:43 3.txt
-rw-r--r-- 1 root root 5 ноя 15 22:30 4.txt
drwxr-xr-x 4 root root 4096 ноя 15 22:31 level
root@debian:/home/user# cd level/
root@debian:/home/user/level# cd level3
root@debian:/home/user/level/level3# cd level4
root@debian:/home/user/level/level3/level4# ls -l
итого 8
-rw-r--r-- 1 root root 104 ноя 15 21:14 2.txt
-rw-r--r-- 3 root root 181 ноя 15 21:15 3.txt
lrwxrwxrwx 1 root root 23 ноя 15 22:41 thirdlink -> /home/user/level1/1.txt
lrwxrwxrwx 1 root root 22 ноя 15 22:55 thirdlinkone -> /home/user/level/1.txt

```

Рисунок 22 – Создание символической ссылки(красная линия)

```

lrwxrwxrwx 1 root root 29 ноя 15 23:06 thirdlink.txt -> /home/user/level/level2/1.txt
root@debian:/home/user/level/level3/level4# cat thirdlink.txt
I love winter! Winter is cold.
Is's snowy hills with sled.
It's warm snuggly beds.
root@debian:/home/user/level/level3/level4#

```

Рисунок 23 – Переход по символической ссылке

```

root@debian:/home/user/level/level2# ls
1.txt thirdhardlink
root@debian:/home/user/level/level2# rm thirdhardlink
root@debian:/home/user/level/level2# ls
1.txt
root@debian:/home/user/level/level2# rm 1.txt
root@debian:/home/user/level/level2# ls
root@debian:/home/user/level/level2# cd ..
root@debian:/home/user/level# cd level3
root@debian:/home/user/level/level3# cd level4
root@debian:/home/user/level/level3/level4# ls -l
итого 8
-rw-r--r-- 1 root root 104 ноя 15 21:14 2.txt
-rw-r--r-- 2 root root 181 ноя 15 21:15 3.txt
lrwxrwxrwx 1 root root 23 ноя 15 22:41 thirdlink -> /home/user/level1/1.txt
lrwxrwxrwx 1 root root 22 ноя 15 22:55 thirdlinkone -> /home/user/level/1.txt
lrwxrwxrwx 1 root root 22 ноя 15 23:02 thirdlinkone.txt -> /home/user/level/1.txt
lrwxrwxrwx 1 root root 22 ноя 15 23:05 thirdlinkt.txt -> /home/user/level/1.txt
lrwxrwxrwx 1 root root 29 ноя 15 23:06 thirdlink.txt -> /home/user/level/level2/1.txt
root@debian:/home/user/level/level3/level4# rm thirdlink
root@debian:/home/user/level/level3/level4# ls -l
итого 8
-rw-r--r-- 1 root root 104 ноя 15 21:14 2.txt
-rw-r--r-- 2 root root 181 ноя 15 21:15 3.txt
lrwxrwxrwx 1 root root 22 ноя 15 22:55 thirdlinkone -> /home/user/level/1.txt
lrwxrwxrwx 1 root root 22 ноя 15 23:02 thirdlinkone.txt -> /home/user/level/1.txt
lrwxrwxrwx 1 root root 22 ноя 15 23:05 thirdlinkt.txt -> /home/user/level/1.txt
lrwxrwxrwx 1 root root 29 ноя 15 23:06 thirdlink.txt -> /home/user/level/level2/1.txt

```

Рисунок 24 – Реакция системы на удаление файла, на который имеются ссылки и файла, на который имеются символическая ссылка

```

root@debian:/home/user# ls -la
итого 52
drwxr-xr-x 3 user user 4096 ноя 15 22:30 .
drwxr-xr-x 4 root root 4096 ноя 13 21:23 ..
-rw-r--r-- 1 root root 83 ноя 14 21:31 1.txt
-rw----- 1 root root 12288 ноя 14 20:39 .1.txt.swp
-rw-r--r-- 1 root root 104 ноя 14 21:41 2.txt
-rw-r--r-- 1 root root 181 ноя 15 20:43 3.txt
-rw-r--r-- 1 root root 5 ноя 15 22:30 4.txt
-rw-r--r-- 1 user user 220 мар 27 2022 .bash_logout
-rw-r--r-- 1 user user 3526 мар 27 2022 .bashrc
drwxr-xr-x 4 root root 4096 ноя 15 22:31 level
-rw-r--r-- 1 user user 807 мар 27 2022 .profile
root@debian:/home/user#

```

Рисунок 25 – Уничтожены файлы и подкаталоги и оставлены исходные документы

## ЭТАП 3

```
root@debian:/# sudo adduser rolanac20
Добавляется пользователь «rolanac20» ...
Добавляется новая группа «rolanac20» (1002) ...
Добавляется новый пользователь «rolanac20» (1002) в группу «rolanac20» ...
Создаётся домашний каталог «/home/rolanac20» ...
Копирование файлов из «/etc/skel» ...
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
Изменение информации о пользователе rolanac20
Введите новое значение или нажмите ENTER для выбора значения по умолчанию
    Полное имя []: rolan
    Номер комнаты []: 7
    Рабочий телефон []: 88005553535
    Домашний телефон []: 222222
    Другое []: not
Данная информация корректна? [Y/n] y
root@debian:/# sudo adduser zakirovruslan
Добавляется пользователь «zakirovruslan» ...
Добавляется новая группа «zakirovruslan» (1003) ...
Добавляется новый пользователь «zakirovruslan» (1003) в группу «zakirovruslan» ...
Создаётся домашний каталог «/home/zakirovruslan» ...
Копирование файлов из «/etc/skel» ...
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
Изменение информации о пользователе zakirovruslan
Введите новое значение или нажмите ENTER для выбора значения по умолчанию
    Полное имя []: ruslan
    Номер комнаты []: -
    Рабочий телефон []: -
    Домашний телефон []: -
    Другое []: -
Данная информация корректна? [Y/n] y
root@debian:/# _
```

Рисунок 26 – Создание новых пользователей

```
root@debian:/# su - rolanac20
rolanac20@debian:~$ ls -l
итого 0
rolanac20@debian:~$ touch rolanac20.txt
rolanac20@debian:~$ ls -l
итого 0
-rw-r--r-- 1 rolanac20 rolanac20 0 ноя 15 23:21 rolanac20.txt
rolanac20@debian:~$ _
```

Рисунок 27 – Создание файла с именем

```
GNU nano 5.4 rolanac20.txt *
This is test page Zakirov Rolan Ruslanovich
```

Рисунок 28 – Внесение текста в файл при помощи редактора NANO и сохранение изменений

```
rolanac20@debian:~$ su
Пароль:
root@debian:/home/rolanac20# mv rolanac20.txt /home/zakirovruslan/
root@debian:/home/rolanac20# su - zakirovruslan
zakirovruslan@debian:~$ ls -l
итого 4
-rw-r--r-- 1 rolanac20 rolanac20 44 ноя 15 23:30 rolanac20.txt
```

Рисунок 29 – Результат переноса файла в другой каталог при помощи команды `mv`

```
GNU nano 5.4 rolanac20.txt
echo "This is test page Zakirov Rolan Ruslanovich"
echo "This page edited by user Ruslan Zakirov"
```

Рисунок 30 – Добавление строки в файл с помощью NANO

```
root@debian:/home/zakirovruslan# mv rolanac20.txt /home/rolanac20/
root@debian:/home/zakirovruslan# su - rolanac20
rolanac20@debian:~$ su
Пароль:
root@debian:/home/rolanac20# ls -l
итого 4
-rw-r--r-- 1 rolanac20 rolanac20 98 ноя 15 23:43 rolanac20.txt
```

Рисунок 31 – Результат переноса файла в другой каталог при помощи команды `mv`

```
GNU nano 5.4 rolanac20.txt
#!/bin/bash
echo "This is test page Zakirov Rolan Ruslanovich"
echo "This page edited by user Ruslan Zakirov"
```

Рисунок 32 – Добавление строки `#!/bin/bash` для запуска скрипта

```
root@debian:/home/rolanac20# chmod u+x rolanac20.txt
root@debian:/home/rolanac20# sh rolanac20.txt
This is test page Zakirov Rolan Ruslanovich
This page edited by user Ruslan Zakirov
root@debian:/home/rolanac20# _
```

Рисунок 33 – Результат после запуска исполняемого файла “rolanac20”



## ЭТАП 4

```
root@debian:/home/user# tar -cvf arh4.gz 1.txt 2.txt 3.txt
1.txt
2.txt
3.txt
root@debian:/home/user# _
```

Рисунок 34 – Создание архива arh4.gz

```
root@debian:/home/user# tar tf arh4.gz
1.txt
2.txt
3.txt
root@debian:/home/user# _
```

Рисунок 34 – Просмотр содержимого архива arh4.gz

```
root@debian:/home/user# gzip -6 arh4.gz > ah
gzip: arh4.gz already has .gz suffix -- unchanged
root@debian:/home/user# ls
1.txt 2.txt 3.txt ah arh4.gz
root@debian:/home/user# _
```

Рисунок 35 – Сжатие файла. Степень сжатия 6(0 – минимальный уровень; 9-максимальный уровень сжатия)

```
root@debian:/home/user# tar -xvf arh4.gz
1.txt
2.txt
3.txt
root@debian:/home/user# _
```

Рисунок 36 – Распаковка архива

```
root@debian:/home/user# sudo du -sh /home/rolan
16K    /home/rolan
root@debian:/home/user# sudo du -sh /home/user
44K    /home/user
root@debian:/home/user# sudo du -sh /home/
124K   /home/
root@debian:/home/user# sudo du -h /home/user
44K    /home/user
```

Рисунок 37 – Действительный размер всех текущих каталогов

## 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 Debian.

## Контрольные вопросы

1. Файловая система - это структура, с помощью которой ядро операционной системы предоставляет пользователям (и процессам) ресурсы долговременной памяти системы, т. е. памяти на различного вида долговременных носителях информации - жестких дисках, магнитных лентах, CD-ROM и т. п.

2. Права доступа состоят из 9 символов (3 группы по 3 символа). Первая группа символов отвечает за права владельца файла, вторая – за права его группы пользователей, третья – за права остальных пользователей.

Символы, обозначающие права:

r – право на чтение файла

w – право на запись в файл

x – право на запуск файла на исполнение

Символы идут в порядке rwx, а прочерк означает, что соответствующее право не предоставлено.

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

4. Жесткая ссылка является просто другим именем исходного файла.

5. Поиск в linux производится командой find.

Основные 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

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. cd /Way/to/directory – переход в указанную директорию

ls – показывает список файлов в директории

mkdir – создаёт новую директорию

rmdir – удаляет выбранную директорию

7. ls -al – показывает все файлы в директории с атрибутами, включая скрытые файлы.

ls -F – выводит список подкаталогов данного каталога

8. mv /War/to/directory

9. В домашнюю директорию пользователя, от имени которого была выполнена данная команда.

10. С помощью команды ls -R

11. С помощью команды mkdir -p

12.

13. С помощью команды `cp -r`

14. С помощью команды `rm -r`

15. Ключи команды `ls`:

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

-A - не отображать ссылку на текущую папку и корневую папку `.` и `..`;

--author - выводить создателя файла в режиме подробного списка;

-b - выводить Escape последовательности вместо непечатаемых символов;

--block-size - выводить размер каталога или файла в определенной единице измерения, например, мегабайтах, гигабайтах или килобайтах;

-B - не выводить резервные копии, их имена начинаются с `~`;

-c - сортировать файлы по времени модификации или создания, сначала будут выведены новые файлы;

-C - выводить колонками;

--color - включить цветной режим вывода, автоматически активирована во многих дистрибутивах;

-d - выводить только директории, без их содержимого, полезно при рекурсивном выводе;

-D - использовать режим вывода, совместимый с Emacs;

-f - не сортировать;

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

--full-time - показывать подробную информацию, плюс вся информация о времени в формате ISO;

-g - показывать подробную информацию, но кроме владельца файла;

--group-directories-first - сначала отображать директории, а уже потом файлы;

-G - не выводить имена групп;

-h - выводить размеры папок в удобном для чтения формате;

-H - открывать символические ссылки при рекурсивном использовании;

--hide - не отображать файлы, которые начинаются с указанного символа;

-i - отображать номер индекса `inode`, в которой хранится этот файл;

- l - выводить подробный список, в котором будет отображаться владелец, группа, дата создания, размер и другие параметры;
- L - для символических ссылок отображать информацию о файле, на который они ссылаются;
- m - разделять элементы списка запятой;
- n - выводить UID и GID вместо имени и группы пользователя;
- N - выводить имена как есть, не обрабатывать контролирующие последовательности;
- Q - брать имена папок и файлов в кавычки;
- r - обратный порядок сортировки;
- R - рекурсивно отображать содержимое поддиректорий;
- s - выводить размер файла в блоках;
- S - сортировать по размеру, сначала большие;
- t - сортировать по времени последней модификации;
- u - сортировать по времени последнего доступа;
- U - не сортировать;
- X - сортировать по алфавиту;
- Z - отображать информацию о расширениях SELinux;
- 1 - отображать один файл на одну строку.

#### 16. Команда cat:

Команда cat используется для создания файлов. По команде cat на стандартный вывод (т. е. на экран) выводится содержимое указанного файла (или нескольких файлов, если их имена последовательно задать в качестве аргументов команды). Если вывод команды cat перенаправить в файл, то можно получить копию какого-то файла.

#### Команда tee:

Команда tee в Linux нужна для записи вывода любой команды в один или несколько файлов.

Команда cat отличается от команд less и more тем, что команда cat выводит на экран сразу весь файл, а команды less и more позволяют открывать большие файлы с возможностью перемещения по ним.