

Отчёт по лабораторной работе № 5

Анализ файловой системы Linux. Команды для работы с файлами и каталогами

Толстых Максим Алексеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Команды для работы с файлами и каталогами	7
3.2	Копирование, перемещение и переименование файлов и каталогов	9
3.3	Права доступа	13
3.4	Выполнение упражнений	14
3.5	Команда <code>map</code>	17
4	Выводы	20
5	Ответы на контрольные вопросы	21

Список иллюстраций

3.1	touch	7
3.2	cat	7
3.3	less	7
3.4	less	8
3.5	head	8
3.6	tail	9
3.7	equipment	9
3.8	equipment	9
3.9	ski.places.	10
3.10	ski.places.	10
3.11	Файл equipment	10
3.12	Файл equipment	10
3.13	equiplist	11
3.14	equiplist	11
3.15	equiplist2	11
3.16	equiplist2	11
3.17	equipment	12
3.18	equipment	12
3.19	~/ski.places/equipment	12
3.20	~/ski.places/equipment	12
3.21	newdir	13
3.22	newdir	13
3.23	plans	13
3.24	chmod	14
3.25	Содержимое файла /etc/passwd	14
3.26	file.old	15
3.27	play	15
3.28	fun	15
3.29	games	15
3.30	feather	15
3.31	feather	16
3.32	feather	16
3.33	feather	16
3.34	play	16
3.35	play	16
3.36	play	17
3.37	mount	17

3.38 fsck	18
3.39 mkfs	18
3.40 kill	19
5.1 kill	21

1 Цель работы

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

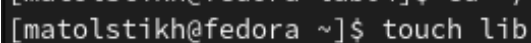
2 Задание

– Ознакомиться и разобрать на практике основные команды для работы с файловой системой Linux, её структурой, именами и содержанием каталогов -
Выполнить упражнения - Ответить на контрольные вопросы

3 Выполнение лабораторной работы

3.1 Команды для работы с файлами и каталогами

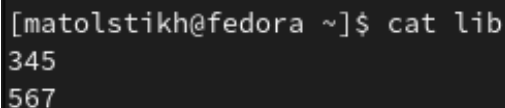
Для создания текстового файла воспользовались командой touch (рис. [3.1])



```
[matolstikh@fedora ~]$ touch lib
```

Рис. 3.1: touch

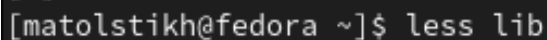
Для просмотра файла небольшого размера можно воспользовались командой cat.(рис. [3.2])



```
[matolstikh@fedora ~]$ cat lib
345
567
```

Рис. 3.2: cat

Для просмотра файла постранично воспользовались командой less. (рис. [3.3]), (рис. [3.4])



```
[matolstikh@fedora ~]$ less lib
```

Рис. 3.3: less



Рис. 3.4: less

Для просмотра первых 10 страниц файла воспользовались командой head. (рис. [3.5])

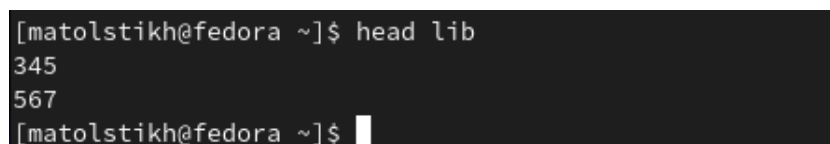


Рис. 3.5: head

Для просмотра последних 10 страниц файла воспользовались командой tail. (рис. [3.6])


```
[matolstikh@fedora ~]$ tail lib
345
567
[matolstikh@fedora ~]$
```

Рис. 3.6: tail

3.2 Копирование, перемещение и переименование файлов и каталогов

Выполнили следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:

1. Скопировали файл `/usr/include/sys/io.h` в домашний каталог и назвали его `equipment`. (рис. [3.7]), (рис. [3.8])

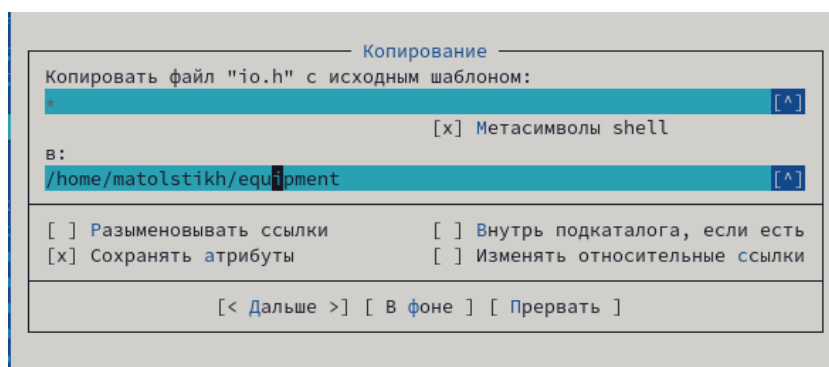


Рис. 3.7: equipment

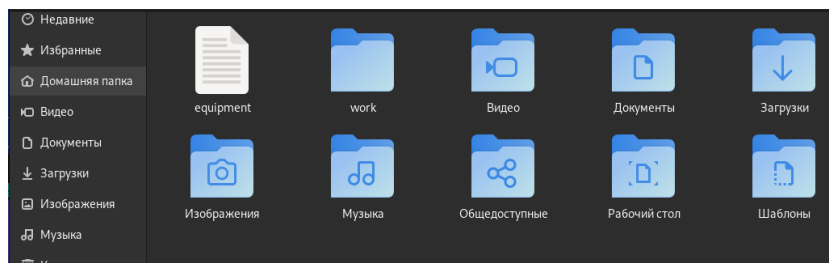


Рис. 3.8: equipment

2. В домашнем каталоге создали директорию ~/ski.places. (рис. [3.9]), (рис. [3.10])

```
[matolstikh@fedora sys]$ mkdir ~/ski.places.
```

Рис. 3.9: ski.places.

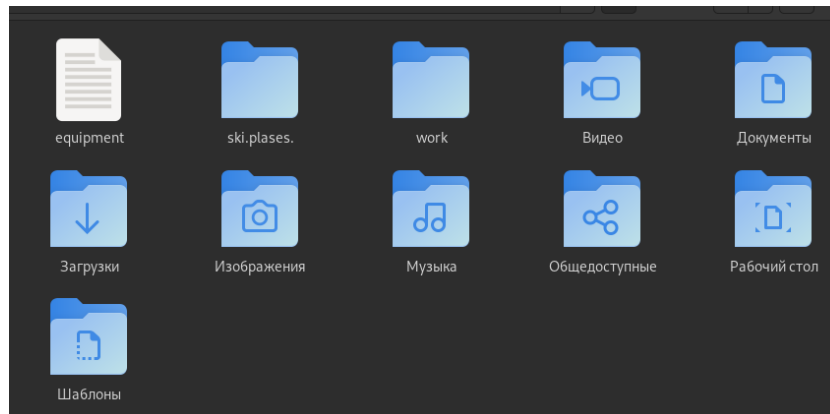


Рис. 3.10: ski.places.

3. Переместили файл equipment в каталог ~/ski.places.(рис. [3.11]), (рис. [3.12])

```
[matolstikh@fedora ~]$ mv equipment ski.places.
```

Рис. 3.11: Файл equipment

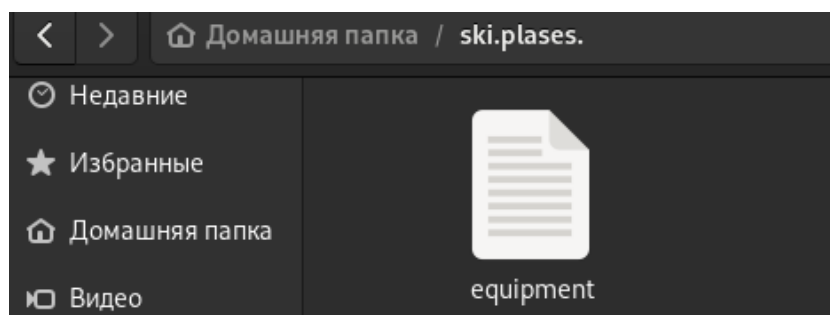


Рис. 3.12: Файл equipment

4. Переименовали файл ~/ski.plases/equipment в ~/ski.plases/equiplist. (рис. [3.13]), (рис. [3.14])

```
[matolstikh@fedora ~]$ mv ski.plases./equipment ski.plases./equiplist
```

Рис. 3.13: equiplist

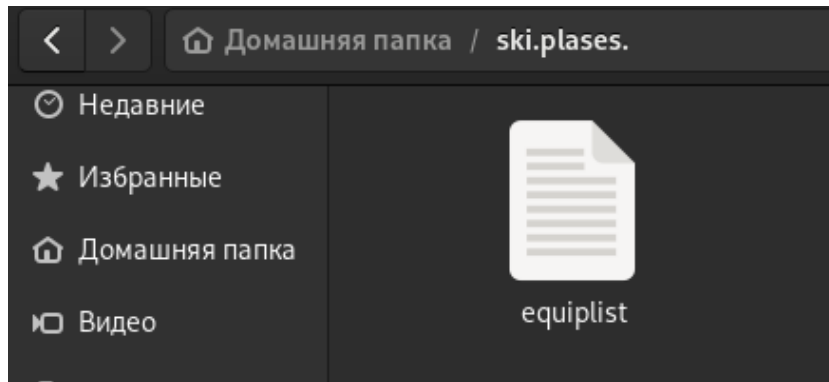


Рис. 3.14: equiplist

5. Создали в домашнем каталоге файл abc1 и скопировали его в каталог ~/ski.plases, назвали его equiplist2. (рис. [3.15]), (рис. [3.16])

```
[matolstikh@fedora ~]$ touch abc1  
[matolstikh@fedora ~]$ cp abc1 ski.plases./equiplist2
```

Рис. 3.15: equiplist2

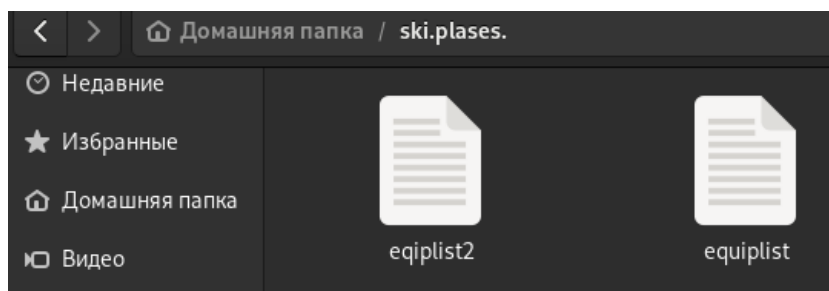


Рис. 3.16: equiplist2

6. Создали каталог с именем equipment в каталоге ~/ski.plases. (рис. [3.17]), (рис. [3.18])

```
[matolstikh@fedora ski.plases.]$ mkdir equipment
```

Рис. 3.17: equipment

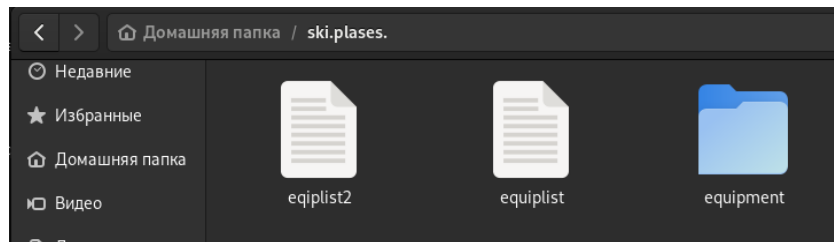


Рис. 3.18: equipment

7. Переместите файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment. (рис. [3.19]), (рис. [3.20])

```
[matolstikh@fedora ski.plases.]$ mv equiplist equipment  
[matolstikh@fedora ski.plases.]$ mv equiplist2 equipment
```

Рис. 3.19: ~/ski.plases/equipment

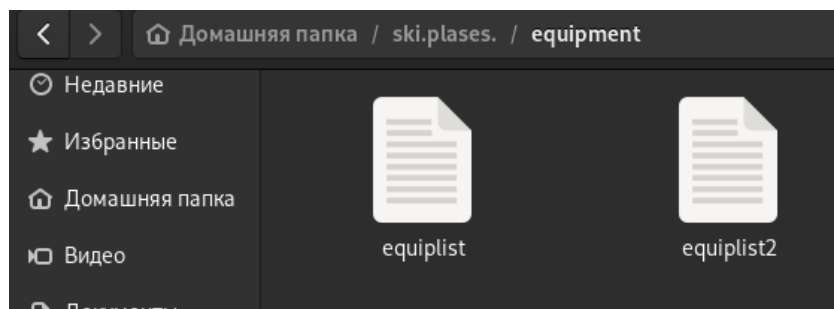


Рис. 3.20: ~/ski.plases/equipment

8. Создали и переместили каталог ~/newdir в каталог ~/ski.places и назвали его plans. (рис. [3.21]), (рис. [3.22]), (рис. [3.23])

```
[matolstikh@fedora ~]$ ls
abcl      ski.places.  Видео      Загрузки    Музыка      'Рабочий стол'
newdir    work        Документы  Изображения  Общедоступные  Шаблоны
```

Рис. 3.21: newdir

```
[matolstikh@fedora ~]$ mv newdir ski.places./plans
```

Рис. 3.22: newdir

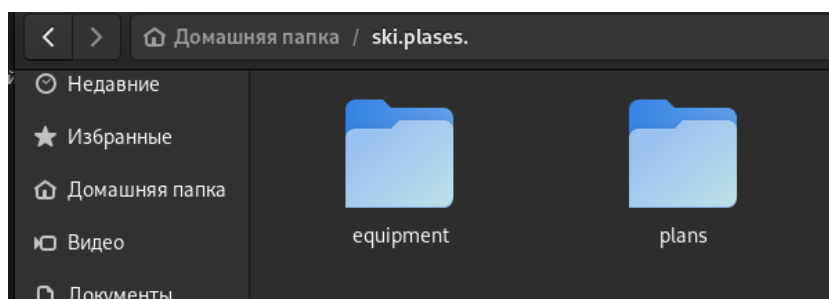


Рис. 3.23: plans

3.3 Права доступа

Создали нужные файлы. Определили опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет. (рис. [3.24]) 1. `drwxr-r- ... australia` 2. `drwx-x-x ... play` 3. `-r-xr-r- ... my_os` 4. `-rw-rw-r- ... feathers`

```

[matolstikh@fedora ~]$ touch australia play my_os feather
[matolstikh@fedora ~]$ ls
australia  play          Видео          Изображения  'Рабочий стол'
feather    ski.places.   Документы     Музыка       Шаблоны
my_os      work         Загрузки      Общедоступные
[matolstikh@fedora ~]$ chmod 744 australia
[matolstikh@fedora ~]$ chmod 711 play
[matolstikh@fedora ~]$ chmod 544 my_os
[matolstikh@fedora ~]$ chmod 664 feather

```

Рис. 3.24: chmod

3.4 Выполнение упражнений

Проделали приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:

1. Просмотрели содержимое файла /etc/passwd. (рис. [3.25])

```

[matolstikh@fedora ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
systemd-oom:x:999:999:systemd Userspace OOM Killer:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/usr/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
polkitd:x:998:997:User for polkitd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin

```

Рис. 3.25: Содержимое файла /etc/passwd

2. Скопировали файл ~/feathers в файл ~/file.old. (рис. [3.26])

```
[matolstikh@fedora ~]$ cp feather ~/file.old
[matolstikh@fedora ~]$ ls
australia  my_os      work      Загрузки      Общедоступные
feather    play      Видео     Изображения   'Рабочий стол'
file.old   ski.places Документы Музыка         Шаблоны
[matolstikh@fedora ~]$
```

Рис. 3.26: file.old

3. Переместили файл ~/file.old в каталог ~/play. (рис. [3.27])

```
[matolstikh@fedora ~]$ mkdir play
[matolstikh@fedora ~]$ mv file.old ~/play
[matolstikh@fedora ~]$ ls ~/play
file.old
[matolstikh@fedora ~]$
```

Рис. 3.27: play

4. Скопировали каталог ~/play в каталог ~/fun. (рис. [3.28])

```
[matolstikh@fedora ~]$ cp -r ~/play ~/fun
[matolstikh@fedora ~]$ ls ~/fun
play
[matolstikh@fedora ~]$
```

Рис. 3.28: fun

5. Переместили каталог ~/fun в каталог ~/play и назвали его games. (рис. [3.29])

```
[matolstikh@fedora ~]$ mv ~/fun ~/play/games
[matolstikh@fedora ~]$ ls play
file.old  games
[matolstikh@fedora ~]$
```

Рис. 3.29: games

6. Лишили владельца файла ~/feather права на чтение. (рис. [3.30])

```
[matolstikh@fedora ~]$ chmod u-r feather
```

Рис. 3.30: feather

7. Попытались просмотреть файл ~/feather командой cat, получили отказ в доступе. (рис. [3.31])

```
[matolstikh@fedora ~]$ cat feather  
cat: feather: Отказано в доступе
```

Рис. 3.31: feather

8. Попытались скопировать файл ~/feather, получили отказ в доступе. (рис. [3.32])

```
[matolstikh@fedora ~]$ cp feather ~/row  
cp: невозможно открыть 'feather' для чтения: Отказано в доступе
```

Рис. 3.32: feather

9. Дали владельцу файла ~/feathers право на чтение. (рис. [3.33])

```
[matolstikh@fedora ~]$ chmod u+r feather
```

Рис. 3.33: feather

10. Лишили владельца каталога ~/play права на выполнение. (рис. [3.34])

```
[matolstikh@fedora ~]$ chmod u-x play
```

Рис. 3.34: play

11. Попытались перейти в каталог ~/play, получили отказ в доступе. (рис. [3.35])

```
[matolstikh@fedora ~]$ cd play/  
bash: cd: play/: Отказано в доступе
```

Рис. 3.35: play

12. Дали владельцу каталога ~/play право на выполнение. (рис. [3.36])

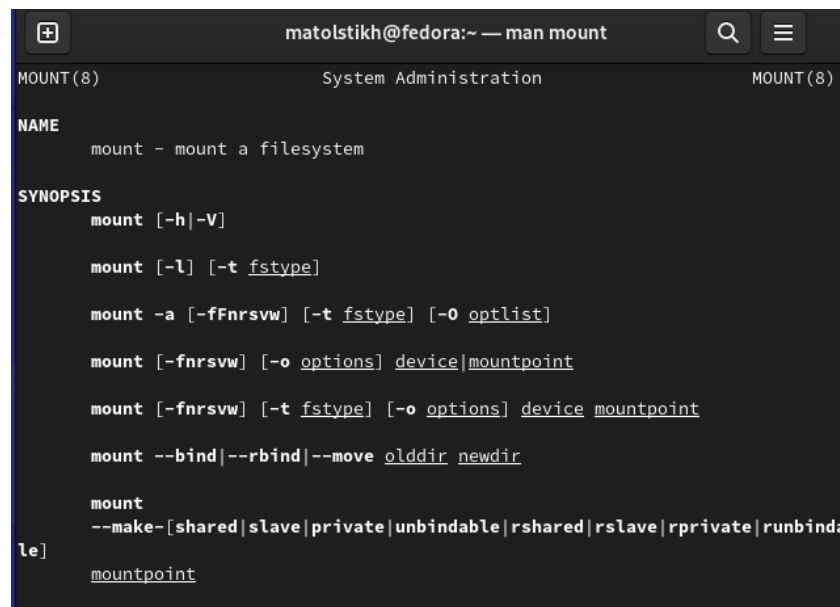
```
[matolstikh@fedora ~]$ chmod u+x play
```

Рис. 3.36: play

3.5 Команда man

Прочитали man по командам mount, fsck, mkfs, kill и кратко их охарактеризовали.

Команда mount используется для монтирования файловой системы в Linux, её также можно использовать для монтирования образа ISO, монтирования удалённой файловой системы Linux и многого другого. (рис. [3.37])



```
matolstikh@fedora:~ — man mount
MOUNT(8) System Administration MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-fFnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount
    --make-=[shared|slave|private|unbindable|rshared|rslave|rprivate|runbinda
le]
    mountpoint
```

Рис. 3.37: mount

fsck используется для проверки и, при необходимости, для восстановления файловых систем Linux, она также может печатать тип файловой системы на указанных разделах диска. (рис. [3.38])

```
matolstikh@fedora:/etc/fwupd — man fsck
FSCK(8) System Administration FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--]
    [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux
    filesystems. filesystem can be a device name (e.g., /dev/hdc1,
/dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem
    label or UUID specifier (e.g.,
    UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the
    fsck program will try to handle filesystems on different physical disk
    drives in parallel to reduce the total amount of time needed to check
    all of them.

    If no filesystems are specified on the command line, and the -A option
    is not specified, fsck will default to checking filesystems in
    /etc/fstab serially. This is equivalent to the -As options.
```

Рис. 3.38: fsck

Буквы в mkfs означают «make file system» (создать файловую систему). Команда обычно используется для управления устройствами хранения в Linux. Отвечает за создание файловых систем. (рис. [3.39])

```
matolstikh@fedora:/etc/fwupd — man mkfs
MKFS(8) System Administration MKFS(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [fs-options] device [size]

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific
    mkfs.<type> utils.

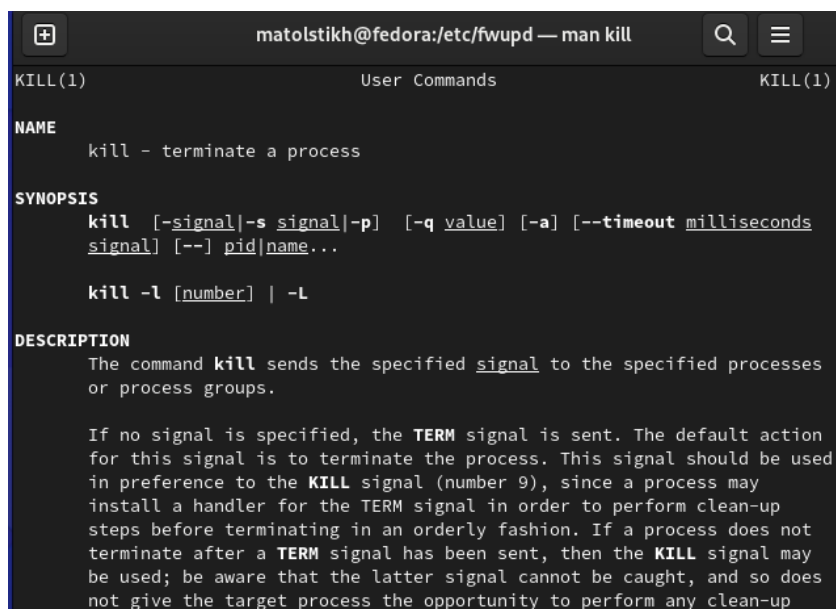
    mkfs is used to build a Linux filesystem on a device, usually a hard
    disk partition. The device argument is either the device name (e.g.,
/dev/hda1, /dev/sdb2), or a regular file that shall contain the
    filesystem. The size argument is the number of blocks to be used for
    the filesystem.

    The exit status returned by mkfs is 0 on success and 1 on failure.

    In actuality, mkfs is simply a front-end for the various filesystem
    builders (mkfs.fstype) available under Linux. The filesystem-specific
    builder is searched for via your PATH environment setting only. Please
```

Рис. 3.39: mkfs

Команда `kill` предназначена для посылки сигнала процессу. По умолчанию, если мы не указываем какой сигнал посылать, посылается сигнал `SIGTERM` (от слова *termination* — завершение). `SIGTERM` указывает процессу на то, что необходимо завершиться. (рис. [3.40])



```
matolstikh@fedora:/etc/fwupd — man kill
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds
    signal] [--] pid|name...

    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to the specified processes
    or process groups.

    If no signal is specified, the TERM signal is sent. The default action
    for this signal is to terminate the process. This signal should be used
    in preference to the KILL signal (number 9), since a process may
    install a handler for the TERM signal in order to perform clean-up
    steps before terminating in an orderly fashion. If a process does not
    terminate after a TERM signal has been sent, then the KILL signal may
    be used; be aware that the latter signal cannot be caught, and so does
    not give the target process the opportunity to perform any clean-up
```

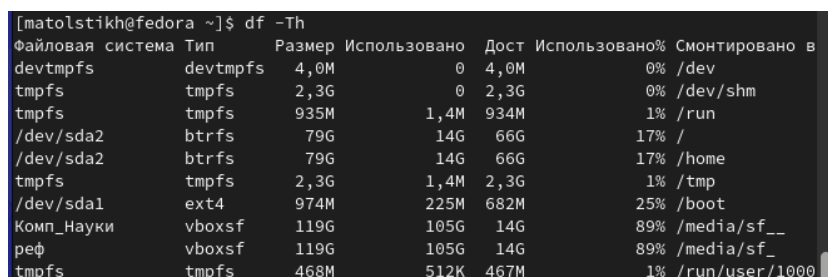
Рис. 3.40: `kill`

4 Выводы

В ходе выполнения лабораторной работы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрели практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

5 Ответы на контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.
(рис. [5.1])



Файловая система	Тип	Размер	Использовано	Дост	Использовано%	Смонтировано в
devtmpfs	devtmpfs	4,0M	0	4,0M	0%	/dev
tmpfs	tmpfs	2,3G	0	2,3G	0%	/dev/shm
tmpfs	tmpfs	935M	1,4M	934M	1%	/run
/dev/sda2	btrfs	79G	14G	66G	17%	/
/dev/sda2	btrfs	79G	14G	66G	17%	/home
tmpfs	tmpfs	2,3G	1,4M	2,3G	1%	/tmp
/dev/sda1	ext4	974M	225M	682M	25%	/boot
Комп_Науки	vboxsf	119G	105G	14G	89%	/media/sf_
реф	vboxsf	119G	105G	14G	89%	/media/sf_
tmpfs	tmpfs	468M	512K	467M	1%	/run/user/1000

Рис. 5.1: kill

2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.
 - / — root каталог. Содержит в себе всю иерархию системы;
 - /bin — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps);
 - /boot — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz);

- /dev — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;
- /etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;
- /home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;
- /lib — содержит системные библиотеки, с которыми работают программы и модули ядра;
- /lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;
- /media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;
- /mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;
- /opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);

- /proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;
 - /root — директория, которая содержит файлы и личные настройки суперпользователя;
 - /run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;
 - /sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;
 - /srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);
 - /sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;
 - /tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;
 - /usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;
 - /var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.
3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе? (Монтирование тома)

4. Назовите основные причины нарушения целостности файловой системы.
Как устранить повреждения файловой системы?

Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

- Один блок адресуется несколькими mode (принадлежит нескольким файлам).
- Блок помечен как свободный, но в то же время занят (на него ссылается inode).
- Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
- Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
- Несовпадение между размером файла и суммарным размером адресуемых inode блоков.
- Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
- “Потерянные” файлы (правильные inode, на которые не ссылаются записи каталогов).
- Недопустимые или неразмещенные номера inode в записях каталогов.

5. Как создаётся файловая система?

mkfs - позволяет создать файловую систему Linux.

6. Дайте характеристику командам для просмотра текстовых файлов.

Cat - выводит содержимое файла на стандартное устройство вывода

7. Приведите основные возможности команды `cp` в Linux.

`cp` – копирует или перемещает директорию, файлы.

8. Приведите основные возможности команды `mv` в Linux.

`Mv` - переименовать или переместить файл или директорию

9. Что такое права доступа? Как они могут быть изменены?

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.