

Отчёт по лабораторной работе №7

дисциплина: Операционные системы

Маслова Анна Павловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16
4	Ответы на контрольные вопросы	17
	Список литературы	19

Список иллюстраций

2.1	Первая часть лр	6
2.2	Файл equipment	7
2.3	Создание директории ~/ski.plases	7
2.4	Перемещение файла equipment	7
2.5	Переименование файла equipment	8
2.6	Создание abc1	8
2.7	Создание каталога с именем equipment в каталоге ~/ski.plases. . .	8
2.8	Перемещение файлов ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment	8
2.9	Создайте и переместите каталог ~/newdir в каталог ~/ski.plases и назовите его plans.	8
2.10	Создание файлов	9
2.11	Определение прав доступа	9
2.12	Просмотр содержимого password	10
2.13	Скопируем файл ~/feathers в файл ~/file.old	10
2.14	Переместим файл ~/file.old в каталог ~/play	10
2.15	Скопируем каталог ~/play в каталог ~/fun	10
2.16	Перемещение каталога ~/fun в каталог ~/play	10
2.17	Лишение владельца файла ~/feathers права на чтение	10
2.18	Команда cat и sr	11
2.19	Дадим владельцу файла ~/feathers право на чтение	11
2.20	Лишение владельца каталога ~/play права на выполнение	11
2.21	Переход в каталог ~/play.	11
2.22	man mount	12
2.23	man fsck	13
2.24	man mkfs	14
2.25	man kill	15

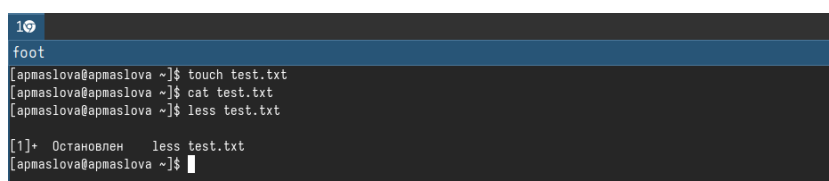
Список таблиц

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 Выполнение лабораторной работы

Выполните все примеры, приведённые в первой части описания лабораторной работы (рис. 2.1).



```
10
foot
[apmaslova@apmaslova ~]$ touch test.txt
[apmaslova@apmaslova ~]$ cat test.txt
[apmaslova@apmaslova ~]$ less test.txt

[1]+ Остановлен less test.txt
[apmaslova@apmaslova ~]$
```

Рис. 2.1: Первая часть лр

Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него (рис. 2.2).

```

foot
insw (unsigned short int __port, void *__addr, unsigned long int __count)
{
    __asm__ __volatile__ ("cld ; rep ; insw" : "=D" (__addr), "=c" (__count)
                        : "d" (__port), "0" (__addr), "1" (__count));
}

static __inline void
insl (unsigned short int __port, void *__addr, unsigned long int __count)
{
    __asm__ __volatile__ ("cld ; rep ; insl" : "=D" (__addr), "=c" (__count)
                        : "d" (__port), "0" (__addr), "1" (__count));
}

static __inline void
outsb (unsigned short int __port, const void *__addr,
       unsigned long int __count)
{
    __asm__ __volatile__ ("cld ; rep ; outsb" : "=S" (__addr), "=c" (__count)
                        : "d" (__port), "0" (__addr), "1" (__count));
}

static __inline void
outsw (unsigned short int __port, const void *__addr,
       unsigned long int __count)
{
    __asm__ __volatile__ ("cld ; rep ; outsw" : "=S" (__addr), "=c" (__count)
                        : "d" (__port), "0" (__addr), "1" (__count));
}

static __inline void
outsl (unsigned short int __port, const void *__addr,
       unsigned long int __count)
{
    __asm__ __volatile__ ("cld ; rep ; outsl" : "=S" (__addr), "=c" (__count)
                        : "d" (__port), "0" (__addr), "1" (__count));
}

#endif /* GNU C */

__END_DECLS
#endif /* _SYS_IO_H */
[apmaslova@apmaslova ~]$ ls
Desktop Downloads equipment git-extended LICENSE pass.txt test.txt work Видео Документы Загрузки
[apmaslova@apmaslova ~]$

```

Рис. 2.2: Файл equipment

В домашнем каталоге создайте директорию `~/ski.places` (рис. 2.3).

```

[apmaslova@apmaslova ~]$ mkdir ~/ski.places
[apmaslova@apmaslova ~]$ ls
Desktop Downloads equipment git-extended LICENSE pass.txt ski.places test.txt work Видео Документы Загрузки
[apmaslova@apmaslova ~]$

```

Рис. 2.3: Создание директории `~/ski.places`

Переместите файл `equipment` в каталог `~/ski.places` (рис. 2.4).

```

[apmaslova@apmaslova ~]$ mv equipment ski.places
[apmaslova@apmaslova ~]$ ls
Desktop Downloads git-extended LICENSE pass.txt ski.places test.txt work Видео Документы Загрузки
[apmaslova@apmaslova ~]$ ls ski.places
equipment
[apmaslova@apmaslova ~]$

```

Рис. 2.4: Перемещение файла `equipment`

Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist` (рис. 2.5).

```
[apmaslova@apmaslova ~]$ mv ski.places/equipment ski.places/equiplist
[apmaslova@apmaslova ~]$ ls ski.places
equiplist
[apmaslova@apmaslova ~]$
```

Рис. 2.5: Переименование файла equipment

Создайте в домашнем каталоге файл abc1 и скопируйте его в каталог ~/ski.places, назовите его equiplist2. (рис. 2.6)

```
[apmaslova@apmaslova ~]$ touch abc1
[apmaslova@apmaslova ~]$ ls
abc1 Desktop Downloads git-extended LICENSE pass.txt ski.places test.txt work Видео Документы
[apmaslova@apmaslova ~]$ cp abc1 ski.places/equiplist2
[apmaslova@apmaslova ~]$ ls ski.places
equiplist equiplist2
[apmaslova@apmaslova ~]$
```

Рис. 2.6: Создание abc1

Создайте каталог с именем equipment в каталоге ~/ski.places. (рис. 2.7)

```
[apmaslova@apmaslova ~]$ mkdir ski.places/equipment
[apmaslova@apmaslova ~]$ ls ski.places
equiplist equiplist2 equipment
[apmaslova@apmaslova ~]$
```

Рис. 2.7: Создание каталога с именем equipment в каталоге ~/ski.places.

Переместите файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment. (рис. 2.8)

```
[apmaslova@apmaslova ~]$ ls ski.places
equiplist equiplist2 equipment
[apmaslova@apmaslova ~]$ mv ski.places/equiplist ski.places/equipment
[apmaslova@apmaslova ~]$ ls ski.places
equiplist2 equipment
[apmaslova@apmaslova ~]$ mv ski.places/equiplist2 ski.places/equipment
[apmaslova@apmaslova ~]$ ls ski.places/equipment
ls: невозможно получить доступ к 'ski.places/equipment': Нет такого файла или каталога
[apmaslova@apmaslova ~]$ ls ski.places/equipment
equiplist equiplist2
[apmaslova@apmaslova ~]$
```

Рис. 2.8: Перемещение файлов ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment

Создайте и переместите каталог ~/newdir в каталог ~/ski.places и назовите его plans. (рис. 2.9)

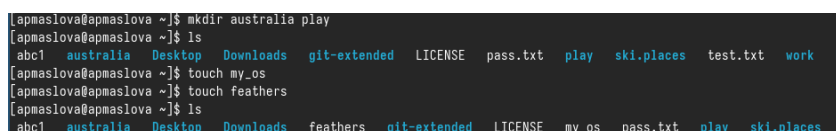
```
[apmaslova@apmaslova ~]$ mkdir newdir
[apmaslova@apmaslova ~]$ mv newdir ski.places/plans
[apmaslova@apmaslova ~]$ ls
abc1 Desktop Downloads git-extended LICENSE pass.txt ski.places test.txt work Видео Документы
[apmaslova@apmaslova ~]$ ls ski.places
equipment plans
[apmaslova@apmaslova ~]$
```

Рис. 2.9: Создайте и переместите каталог ~/newdir в каталог ~/ski.places и назовите его plans.

Определите опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет (рис. 2.11):

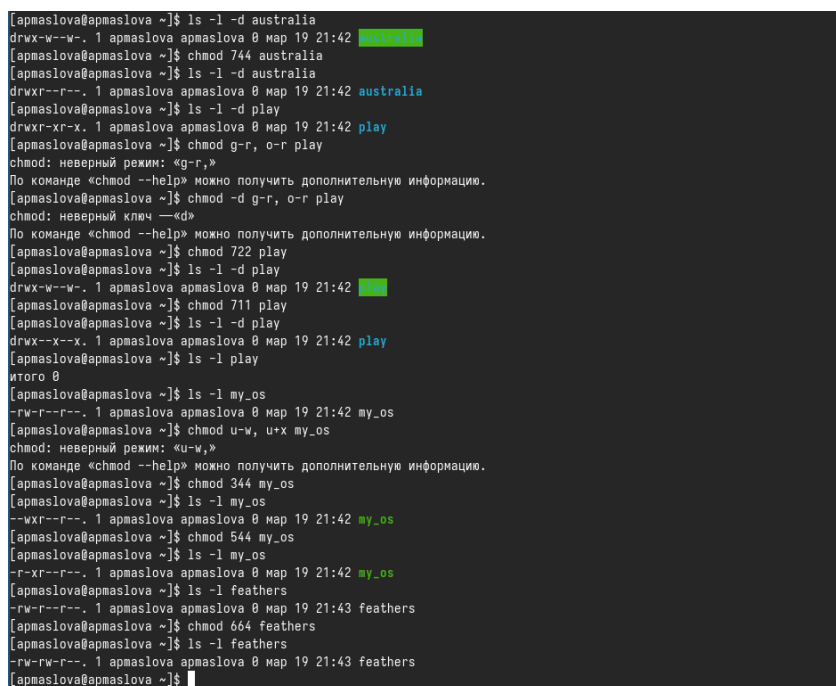
```
drwxr--r--  australia
drwx--x--x   play
-r-xr--r--   my_os
-rw-rw-r--   feathers
```

При необходимости создайте нужные файлы.(рис. 2.10)



```
[apmaslova@apmaslova ~]$ mkdir australia play
[apmaslova@apmaslova ~]$ ls
abc1  australia  Desktop  Downloads  git-extended  LICENSE  pass.txt  play  ski.places  test.txt  work  B
[apmaslova@apmaslova ~]$ touch my_os
[apmaslova@apmaslova ~]$ touch feathers
[apmaslova@apmaslova ~]$ ls
abc1  australia  Desktop  Downloads  feathers  git-extended  LICENSE  my_os  pass.txt  play  ski.places
```

Рис. 2.10: Создание файлов



```
[apmaslova@apmaslova ~]$ ls -l -d australia
drwx-w--w-. 1 apmaslova apmaslova 0 map 19 21:42 australia
[apmaslova@apmaslova ~]$ chmod 744 australia
[apmaslova@apmaslova ~]$ ls -l -d australia
drwxr--r--. 1 apmaslova apmaslova 0 map 19 21:42 australia
[apmaslova@apmaslova ~]$ ls -l -d play
drwx-w--w-. 1 apmaslova apmaslova 0 map 19 21:42 play
[apmaslova@apmaslova ~]$ chmod g-r, o-r play
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
[apmaslova@apmaslova ~]$ chmod -d g-r, o-r play
chmod: неверный ключ «-d»
По команде «chmod --help» можно получить дополнительную информацию.
[apmaslova@apmaslova ~]$ chmod 722 play
[apmaslova@apmaslova ~]$ ls -l -d play
drwx-w--w-. 1 apmaslova apmaslova 0 map 19 21:42 play
[apmaslova@apmaslova ~]$ chmod 711 play
[apmaslova@apmaslova ~]$ ls -l -d play
drwx--x--x. 1 apmaslova apmaslova 0 map 19 21:42 play
[apmaslova@apmaslova ~]$ ls -l play
итого 0
[apmaslova@apmaslova ~]$ ls -l my_os
-rw-r--r--. 1 apmaslova apmaslova 0 map 19 21:42 my_os
[apmaslova@apmaslova ~]$ chmod u-w, u+x my_os
chmod: неверный режим: «u-w,»
По команде «chmod --help» можно получить дополнительную информацию.
[apmaslova@apmaslova ~]$ chmod 344 my_os
[apmaslova@apmaslova ~]$ ls -l my_os
--wxr--r--. 1 apmaslova apmaslova 0 map 19 21:42 my_os
[apmaslova@apmaslova ~]$ chmod 544 my_os
[apmaslova@apmaslova ~]$ ls -l my_os
-r-xr--r--. 1 apmaslova apmaslova 0 map 19 21:42 my_os
[apmaslova@apmaslova ~]$ ls -l feathers
-rw-r--r--. 1 apmaslova apmaslova 0 map 19 21:43 feathers
[apmaslova@apmaslova ~]$ chmod 664 feathers
[apmaslova@apmaslova ~]$ ls -l feathers
-rw-rw-r--. 1 apmaslova apmaslova 0 map 19 21:43 feathers
[apmaslova@apmaslova ~]$
```

Рис. 2.11: Определение прав доступа

Просмотрите содержимое файла `/etc/passwd`. (рис. 2.12)

```
[apmaslova@apmaslova ~]$ cat /etc/passwd
cat: /etc/passwd: Нет такого файла или каталога
[apmaslova@apmaslova ~]$
```

Рис. 2.12: Просмотр содержимого password

Скопируйте файл ~/feathers в файл ~/file.old. (рис. 2.13)

```
[apmaslova@apmaslova ~]$ cp feathers file.old
[apmaslova@apmaslova ~]$ ls
abc1 Desktop feathers git-extended my_os play test.txt Видео Загрузки Музыка
australia Downloads file.old LICENSE pass.txt ski.places work Документы Изображения Общед
```

Рис. 2.13: Скопируем файл ~/feathers в файл ~/file.old

Переместите файл ~/file.old в каталог ~/play. (рис. 2.14)

```
[apmaslova@apmaslova ~]$ mv file.old play
[apmaslova@apmaslova ~]$ ls play
file.old
[apmaslova@apmaslova ~]$
```

Рис. 2.14: Переместим файл ~/file.old в каталог ~/play

Скопируйте каталог ~/play в каталог ~/fun. (рис. 2.15)

```
[apmaslova@apmaslova ~]$ cp -r play fun
[apmaslova@apmaslova ~]$ ls
abc1 australia Desktop Downloads feathers fun git-extended LICENSE my_os pass.txt play ski.pla
[apmaslova@apmaslova ~]$
```

Рис. 2.15: Скопируем каталог ~/play в каталог ~/fun

Переместите каталог ~/fun в каталог ~/play и назовите его games. (рис. 2.16)

```
[apmaslova@apmaslova ~]$ mv fun play/games
[apmaslova@apmaslova ~]$ ls
abc1 australia Desktop Downloads feathers git-extended LICENSE my_os pass.txt play ski.places
[apmaslova@apmaslova ~]$ ls play
file.old games
[apmaslova@apmaslova ~]$
```

Рис. 2.16: Перемещение каталога ~/fun в каталог ~/play

Лишите владельца файла ~/feathers права на чтение. (рис. 2.17)

```
[apmaslova@apmaslova ~]$ ls -l feathers
-rw-rw-r--. 1 apmaslova apmaslova 0 map 19 21:43 feathers
[apmaslova@apmaslova ~]$ chmod u-r feathers
[apmaslova@apmaslova ~]$ ls -l feathers
--w-rw-r--. 1 apmaslova apmaslova 0 map 19 21:43 feathers
[apmaslova@apmaslova ~]$
```

Рис. 2.17: Лишение владельца файла ~/feathers права на чтение

Попытаемся просмотреть файл ~/feathers командой cat и скопировать файл (рис. 2.18)

```
[aromaslova@aromaslova ~]$ cat feathers
cat: feathers: Отказано в доступе
[aromaslova@aromaslova ~]$ cp feathers feathers01
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[aromaslova@aromaslova ~]$
```

Рис. 2.18: Команда cat и cp

Дайте владельцу файла ~/feathers право на чтение. (рис. 2.19)

```
[aromaslova@aromaslova ~]$ chmod u+r feathers
[aromaslova@aromaslova ~]$ ls -l feathers
-rw-rw-r--. 1 aromaslova aromaslova 0 мар 19 21:43 feathers
[aromaslova@aromaslova ~]$
```

Рис. 2.19: Дадим владельцу файла ~/feathers право на чтение

Лишите владельца каталога ~/play права на выполнение (рис. 2.20)

```
[aromaslova@aromaslova ~]$ chmod u-x play
[aromaslova@aromaslova ~]$ ls -l play
ls: невозможно получить доступ к 'play/file.old': Отказано в доступе
ls: невозможно получить доступ к 'play/games': Отказано в доступе
итого 0
-???????? ? ? ? ? ? ? file.old
d???????? ? ? ? ? ? ? games
[aromaslova@aromaslova ~]$ ls -l -d play
drwx--x--x. 1 aromaslova aromaslova 26 мар 19 22:09 play
[aromaslova@aromaslova ~]$
```

Рис. 2.20: Лишение владельца каталога ~/play права на выполнение

Перейдите в каталог ~/play. Дайте владельцу каталога ~/play право на выполнение. (рис. 2.21)

```
[aromaslova@aromaslova ~]$ cd play
bash: cd: play: Отказано в доступе
[aromaslova@aromaslova ~]$ chmod u+x play
[aromaslova@aromaslova ~]$ ls -l -d play
drwx--x--x. 1 aromaslova aromaslova 26 мар 19 22:09 play
[aromaslova@aromaslova ~]$
```

Рис. 2.21: Переход в каталог ~/play.

Прочитайте man по командам mount, fsck, mkfs, kill. (рис. 2.22, 2.23, 2.24, 2.25)

```
foot
MOUNT(8) System Admini

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|runbindable] mountpoint

DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files
    the big file tree. Conversely, the umount(8) command will detach it again. The filesystem is used to control h

    The standard form of the mount command is:

        mount -t type device dir

    This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. T
    necessary to mount a filesystem by default. See section "Non-superuser mounts" below for more details. The pre
    the pathname dir refers to the root of the filesystem on device.

    If only the directory or the device is given, for example:

        mount /dir

    then mount looks for a mountpoint (and if not found then for a device) in the /etc/fstab file. It's possible t

        mount --target /mountpoint

    The same filesystem may be mounted more than once, and in some cases (e.g., network filesystems) the same file
    control this behavior. All behavior is controlled by the kernel and it is usually specific to the filesystem d
    details).

Manual page mount(8) line 1 (press h for help or q to quit)
```

Рис. 2.22: man mount

```
foot
FSCK(8) System Admini

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.
    UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck program will try to handle filesy

    If no filesystems are specified on the command line, and the -A option is not specified, fsck will default to

    The exit status returned by fsck is the sum of the following conditions:

    0      No errors

    1      Filesystem errors corrected

    2      System should be rebooted

    4      Filesystem errors left uncorrected

    8      Operational error

    16     Usage or syntax error

    32     Checking canceled by user request

    128    Shared-library error

    The exit status returned when multiple filesystems are checked is the bit-wise OR of the exit statuses for eac

    In actuality, fsck is simply a front-end for the various filesystem checkers (fsck.fstype) available under Lin

Manual page fsck(8) line 1 (press h for help or q to quit)
```

Рис. 2.23: man fsck

```

foot
MKFS(8) System Admini

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 a device name or a file name. 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. See the
    filesystem-specific builder manual pages for further details.

OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) will be used.

    fs-options
        Filesystem-specific options to be passed to the real filesystem builder.

    -V, --verbose
        Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option will also
        display the progress of the build.

    -h, --help
        Display help text and exit.

    -V, --version
        Print version and exit. (Option -V will display version information only when it is the only parameter, otherwise it will
        also display the progress of the build.)

BUGS
    All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific options may
    be incompatible with generic options.

AUTHORS
    David Engel <david@ods.com>, Fred N. van Kempen <waltje@u.walt.nl.mugnet.org>, Ron Sommeling <sommel@sci.kun.nl>

    The manual page was shamelessly adapted from Remy Card's version for the ext2 filesystem.

Manual page mkfs(8) line 1 (press h for help or q to quit)

```

Рис. 2.24: man mkfs

```
foot
kill(1) User Co
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 for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before termination.
    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here.
    If signal is 0, then no actual signal is sent, but error checking is still performed.
ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.
    pid
        Each pid can be expressed in one of the following ways:
        n
            where n is larger than 0. The process with PID n is signaled.
        0
            All processes in the current process group are signaled.
        -1
            All processes with a PID larger than 1 are signaled.
        -n
            where n is larger than 1. All processes in process group n are signaled. When an argument of the form -n must be preceded by a '--' option, otherwise it will be taken as the signal to send.
    name
        All processes invoked using this name will be signaled.
Manual page kill(1) line 1 (press h for help or q to quit)
```

Рис. 2.25: man kill

3 Выводы

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.

4 Ответы на контрольные вопросы

1. На жёстком диске компьютера, на котором я выполняла лабораторную работу, были следующие файловые системы: ext4, NTFS.
2. Общая структура файловой системы включает в себя корневую директорию /, в которой находятся поддиректории и файлы. Директории первого уровня этой структуры имеют следующие характеристики:

/bin - содержит исполняемые файлы, необходимые для запуска системы и основных утилит; /etc - содержит конфигурационные файлы системы; /home - домашние директории пользователей; /var - содержит переменные данные, такие как логи и временные файлы; /dev - содержит файлы устройств.

3. Для доступности содержимого файловой системы операционной системе необходимо выполнить монтирование (mount).
4. Основные причины нарушения целостности файловой системы могут быть сбои в работе жёсткого диска, некорректное завершение работы операционной системы, вирусы и многие другие. Для устранения повреждений файловой системы может потребоваться использование инструментов для восстановления данных, проверки и исправления ошибок (например, fsck в Linux).
5. Файловая система создаётся при форматировании диска или раздела. При этом создаются необходимые структуры данных для хранения файлов и каталогов.

6. Для просмотра текстовых файлов в Linux можно использовать команды `cat`, `less`, `more`.
7. Основные возможности команды `cp` в Linux включают копирование файлов и директорий с сохранением атрибутов, возможность рекурсивного копирования директорий, указание целевого пути для копирования.
8. Основные возможности команды `mv` в Linux включают перемещение файлов и директорий с сохранением атрибутов, переименование файлов и директорий, перемещение в другую директорию.
9. Права доступа определяют, кто и как может читать, записывать или исполнять файлы. Их можно изменить с помощью команды `chmod` в Linux, указывая нужные права для владельца, группы и других пользователей.

Список литературы

1. Dash, P. Getting Started with Oracle VM VirtualBox / P. Dash. – Packt Publishing Ltd, 2013. – 86 сс.
2. Colvin, H. VirtualBox: An Ultimate Guide Book on Virtualization with VirtualBox. VirtualBox / H. Colvin. – CreateSpace Independent Publishing Platform, 2015. – 70 сс.
3. Vugt, S. van. Red Hat RHCSA/RHCE 7 cert guide : Red Hat Enterprise Linux 7 (EX200 and EX300) : Certification Guide. Red Hat RHCSA/RHCE 7 cert guide / S. van Vugt. – Pearson IT Certification, 2016. – 1008 сс.
4. Робачевский, А. Операционная система UNIX / А. Робачевский, С. Немнюгин, О. Стесик. – 2-е изд. – Санкт-Петербург : БХВ-Петербург, 2010. – 656 сс.
5. Немец, Э. Unix и Linux: руководство системного администратора. Unix и Linux / Э. Немец, Г. Снайдер, Т.Р. Хейн, Б. Уэйли. – 4-е изд. – Вильямс, 2014. – 1312 сс.
6. Колисниченко, Д.Н. Самоучитель системного администратора Linux : Системный администратор / Д.Н. Колисниченко. – Санкт-Петербург : БХВ-Петербург, 2011. – 544 сс.
7. Robbins, A. Bash Pocket Reference / A. Robbins. – O'Reilly Media, 2016. – 156 сс.