

Отчет по лабораторной работе №5

дисциплина: Операционные системы

Старков Никита Алексеевич

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Контрольные вопросы	14
4	Вывод	21

Список иллюстраций

2.1	Копирование файла в файл	5
2.2	Копирование файлов в каталог	5
2.3	Копирование каталогов	5
2.4	Копирование каталогов	6
2.5	Меняем название файла	6
2.6	Перемещение файла в каталог	6
2.7	Переименовывание каталога	6
2.8	Перемещение одного каталога в другой	6
2.9	Меняем названия каталога	6
2.10	Изменение прав доступа	7
2.11	Изменение прав доступа	7
2.12	Изменение прав доступа	7
2.13	Копирование файла в домашний каталог	7
2.14	Изменение названия	7
2.15	Создание директории и её перемещение в каталог	8
2.16	Переименовывание файла	8
2.17	Копирование файла в каталог и изменение названия	8
2.18	Создание каталога	8
2.19	Перемещение файлов в каталог	8
2.20	Создание и перемещение одного каталога в другой со сменой имени	8
2.21	Создание файлов	9
2.22	Создание каталогов	9
2.23	Присвоение определенных прав доступа	9
2.24	Результат	9
2.25	Просмотр содержимого с помощью команды cat	9
2.26	Копирование одного файла в другой	9
2.27	Перемещение файла в каталог	10
2.28	Копирование одного каталога в другой	10
2.29	Изменение прав доступа. Проверка	10
2.30	Изменение прав доступа. Проверка	10
2.31	Описание команды mount	11
2.32	Описание команды fsck	12
2.33	Описание команды kill	12
2.34	Описание команды mkfs	13

1 Цель работы

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

1) Выполняем все примеры, приведенные в первой части описания лабораторной работы:

1.1) Копируем в файл в текущий каталог. Копируем файл abc1 в файл april

```
nastarkov@dk6n53 ~ $ cd
nastarkov@dk6n53 ~ $ touch abc1
nastarkov@dk6n53 ~ $ cp abc1 april
nastarkov@dk6n53 ~ $ cp abc1 may
```

Рис. 2.1: Копирование файла в файл

1.2) Копируем несколько файлов в каталог. Копируем файлы april и may в каталог monthly.

```
nastarkov@dk6n53 ~ $ mkdir monthly
nastarkov@dk6n53 ~ $ cp april may monthly
```

Рис. 2.2: Копирование файлов в каталог

1.3) Копируем файлы в произвольном каталоге. Копируем файл monthly/may в файл с именем june

Копирование файлов в каталог и меняем имя

1.4) Копируем каталог monthly в каталог monthly.00

```
nastarkov@dk6n53 ~ $ mkdir monthly.00
nastarkov@dk6n53 ~ $ cp -r cp -r monthly monthly.00
```

Рис. 2.3: Копирование каталогов

1.5) Копируем каталог monthly.00 в каталог /tmp

```
nastarkov@dk6n53 ~ $ cp -r monthly.00 /tmp
```

Рис. 2.4: Копирование каталогов

1.6) Изменяем название файла april на july в домашнем каталоге

```
nastarkov@dk6n53 ~ $ cd  
nastarkov@dk6n53 ~ $ mv april july
```

Рис. 2.5: Меняем название файла

1.7) Перемещаем файл july в каталог monthly.00

```
nastarkov@dk6n53 ~ $ mv july monthly.00  
nastarkov@dk6n53 ~ $ ls monthly.00
```

Рис. 2.6: Перемещение файла в каталог

1.8) Переименовываем каталог monthly.00 в каталог в monthly.01

```
nastarkov@dk6n53 ~ $ mv monthly.00 monthly.01
```

Рис. 2.7: Переименовывание каталога

1.9) Перемещаем каталог monthly.01 в каталог reports

```
nastarkov@dk6n53 ~ $ mkdir reports  
nastarkov@dk6n53 ~ $ mv monthly.01 reports
```

Рис. 2.8: Перемещение одного каталога в другой

1.10) Переименовываем каталог reports/monthly.01 в reports/monthly

```
nastarkov@dk6n53 ~ $ mv reports/monthly.01 reports/monthly
```

Рис. 2.9: Меняем названия каталога

1.11) Создаем файл ~/mau с правом выполнения для владельца. Лишаем владельца файла ~/mau права на выполнение

```
nastarkov@dk6n53 ~ $ cd
nastarkov@dk6n53 ~ $ touch may
nastarkov@dk6n53 ~ $ ls -l may
-rw-r--r-- 1 nastarkov studsci 0 мая  4 16:25 may
nastarkov@dk6n53 ~ $ chmod u+x may
nastarkov@dk6n53 ~ $ ls -l may
-rwxr--r-- 1 nastarkov studsci 0 мая  4 16:25 may
nastarkov@dk6n53 ~ $ chmod u-x may
nastarkov@dk6n53 ~ $ ls -l may
-rw-r--r-- 1 nastarkov studsci 0 мая  4 16:25 may
```

Рис. 2.10: Изменение прав доступа

1.12) Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей

```
nastarkov@dk6n53 ~ $ cd
nastarkov@dk6n53 ~ $ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
nastarkov@dk6n53 ~ $ chmod g-r, o-r monthly
```

Рис. 2.11: Изменение прав доступа

1.13) Создаем файл ~/abc1 с правом записи для членов группы

```
nastarkov@dk6n53 ~ $ cd
nastarkov@dk6n53 ~ $ touch abc1
nastarkov@dk6n53 ~ $ chmod g+w abc1
```

Рис. 2.12: Изменение прав доступа

2) Копируем файл /usr/include/sys/io.h в домашний каталог и называем его equipment.

```
nastarkov@dk6n53 ~ $ cp /usr/include/sys/io.h ~
nastarkov@dk6n53 ~ $ ls
abc1  blog  equipment  io.h  monthly  public_html  tmp  Видео
bin   course-directory-student-template  GNUstep  may   public   reports  work  Документы
```

Рис. 2.13: Копирование файла в домашний каталог

```
nastarkov@dk6n53 ~ $ mv ~/io.h ~/equipment
```

Рис. 2.14: Изменение названия

2.1) Создаем директорию ~/ski.places. Перемещаем файл equipment в каталог ~/ski.places

```
nastarkov@dk6n53 ~ $ mkdir ski.places
nastarkov@dk6n53 ~ $ mv equipment ski.places
```

Рис. 2.15: Создание директории и её перемещение в каталог

2.2)Переименовываем файл ~/ski.places/equipment в ~/ski.places/equiplist

```
nastarkov@dk6n53 ~ $ mv ski.places/equipment ski.places/equiplist
```

Рис. 2.16: Переименовывание файла

2.3)Создаем в домашнем каталоге файл abc1 и копируем его в каталог ~/ski.places, называем его equiplist2.

```
nastarkov@dk6n53 ~ $ touch abc1
nastarkov@dk6n53 ~ $ cp abc1 ski.places
nastarkov@dk6n53 ~ $ mv ski.places/abc1 ski.places/equiplist2
```

Рис. 2.17: Копирование файла в каталог и изменение названия

2.4)Создаем каталог с именем equipment в каталоге ~/ski.places

```
nastarkov@dk6n53 ~ $ mkdir ski.places/equipment
```

Рис. 2.18: Создание каталога

2.5)Перемещаем файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment

```
nastarkov@dk6n53 ~ $ mv ski.places/equiplist ski.places/equipment
nastarkov@dk6n53 ~ $ mv ski.places/equiplist2 ski.places/equipment
```

Рис. 2.19: Перемещение файлов в каталог

2.6)Создаем и перемещаем каталог ~/newdir в каталог ~/ski.places и называем его plans

```
nastarkov@dk6n53 ~ $ mkdir newdir
nastarkov@dk6n53 ~ $ mv newdir plans
nastarkov@dk6n53 ~ $ mv plans ski.places
```

Рис. 2.20: Создание и перемещение одного каталога в другой со сменой имени

3)Определяем опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем файлы


```
nastarkov@dk6n53 ~ $ chmod 744 australia
nastarkov@dk6n53 ~ $ chmod 711 play
```

Рис. 2.21: Создание файлов

```
nastarkov@dk6n53 ~ $ mkdir australia play
```

Рис. 2.22: Создание каталогов

```
nastarkov@dk6n53 ~ $ chmod 744 australia
nastarkov@dk6n53 ~ $ chmod 711 play
nastarkov@dk6n53 ~ $ chmod 544 my_os
nastarkov@dk6n53 ~ $ chmod 664 feathers
```

Рис. 2.23: Присвоение определенных прав доступа

```
-rw-rw-r-- 1 nastarkov studsci  0 мая 4 16:46 abc1
drwxr--r-- 2 nastarkov studsci 2048 мая 4 16:58 australia
drwxr-xr-x 2 nastarkov studsci 2048 апр 30 12:55 bin
drwxr-xr-x 10 nastarkov studsci 2048 апр 30 12:59 blog
drwxr-xr-x 5 nastarkov studsci 2048 апр 22 18:14 course-directory-student-template
-rw-rw-r-- 1 nastarkov studsci  0 мая 4 16:54 feathers
drwxr-xr-x 3 nastarkov studsci 2048 ноя 10 17:39 GNUstep
-rw-r--r-- 1 nastarkov studsci 5086 мая 4 16:40 io.h
-rw-r--r-- 1 nastarkov studsci  0 мая 4 16:25 may
drwxr-xr-x 2 nastarkov studsci 2048 мая 4 16:21 monthly
-r-xr--r-- 1 nastarkov studsci  0 мая 4 16:54 my_os
drwxr-xr-x 2 nastarkov studsci 2048 мая 4 16:58 play
drwxr-xr-x 3 nastarkov studsci 2048 сен 2 2021 public
lrwxr-xr-x 1 nastarkov root 18 апр 14 22:20 public_html -> public/public_html
drwxr-xr-x 3 nastarkov studsci 2048 мая 4 16:25 reports
drwxr-xr-x 4 nastarkov studsci 2048 мая 4 16:52 ski_places
drwxr-xr-x 2 nastarkov studsci 2048 апр 29 18:23 tmp
drwxr-xr-x 5 nastarkov studsci 2048 апр 30 14:25 work
drwxr-xr-x 2 nastarkov studsci 2048 сен 16 2021 Видео
drwxr-xr-x 2 nastarkov studsci 4096 апр 29 18:23 Документы
drwxr-xr-x 4 nastarkov studsci 2048 апр 30 13:33 Загрузки
drwxr-xr-x 2 nastarkov studsci 4096 мая 4 16:55 Изображения
drwxr-xr-x 2 nastarkov studsci 2048 сен 2 2021 Музыка
drwxr-xr-x 2 nastarkov studsci 2048 сен 2 2021 Общедоступные
drwxr-xr-x 2 nastarkov studsci 2048 апр 22 19:51 'Рабочий стол'
drwxr-xr-x 2 nastarkov studsci 2048 сен 2 2021 Шаблоны
```

Рис. 2.24: Результат

4)Просматриваем содержимое файла /etc/passwd

```
nastarkov@dk3n59 ~ $ cat /etc/passwd
cat: /etc/passwd: Нет такого файла или каталога
```

Рис. 2.25: Просмотр содержимого с помощью команды cat

4.1)Копируем файл ~/feathers в файл ~/file.old

```
nastarkov@dk6n53 ~ $ touch my_os
nastarkov@dk6n53 ~ $ touch feathers
```

Рис. 2.26: Копирование одного файла в другой

4.2)Перемещаем файл ~/file.old в каталог ~/play

```
nastarkov@dk6n53 ~ $ mv file.old play
```

Рис. 2.27: Перемещение файла в каталог

4.3)Копируем каталог ~/play в каталог ~/fun

```
nastarkov@dk6n53 ~ $ cp -r play fun
```

Рис. 2.28: Копирование одного каталога в другой

4.4)Перемещаем каталог ~/fun в каталог ~/play и называем его games. Лишаем владельца файла ~/feathers права на чтение. Просматриваем файл командой cat. Видим надпись “Отказано в доступе”. При копировании видим надпись “Отказано в доступе”

```
nastarkov@dk6n53 ~ $ mv play/fun play/games
nastarkov@dk6n53 ~ $ chmod u-r feathers
nastarkov@dk6n53 ~ $ cat feathers
cat: feathers: Отказано в доступе
nastarkov@dk6n53 ~ $ cp feathers ski.places
```

Рис. 2.29: Изменение прав доступа. Проверка

4.5)Даем владельцу право на чтение. Лишаем владельца каталога ~/play право на выполнение. Заходим в каталог. Возвращаем право на выполнение

```
nastarkov@dk6n53 ~ $ chmod u+r feathers
nastarkov@dk6n53 ~ $ cat feathers
nastarkov@dk6n53 ~ $ chmod u-x play
nastarkov@dk6n53 ~ $ cd play
nastarkov@dk6n53 ~/play $ chmod u+x play
chmod: невозможно получить доступ к 'play': Нет такого файла или каталога
nastarkov@dk6n53 ~/play $ cd ~
nastarkov@dk6n53 ~ $ chmod u+x play
```

Рис. 2.30: Изменение прав доступа. Проверка

5)По команде man читаем описание команд mount, mkfs, fsck, kill

Команда mount:

предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается

как /. Физически файлы могут располагаться на различных устройствах. Команда `mount` служит для подключения файловых систем разных устройств к этому большому дереву. Наиболее часто встречающаяся форма команды `mount` выглядит следующим образом: «`mount -t vfstype device dir`» Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа `vfstype`, расположенную на устройстве `device`, к заданному каталогу `dir`, который часто называют точкой монтирования

```

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|runbindable} mountpoint

DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the filesystem found on some device to the big file tree. Conversely, the umount(8) command will detach it again. The filesystem is used to control how data is stored on the device or provided in a virtual way by network or other services.

    The standard form of the mount command is:

    mount -t type device dir
  
```

Рис. 2.31: Описание команды `mount`

Команда `fsck`:

это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет. У команды `fsck` следующий синтаксис: `fsck [параметр] – [параметры ФС] [...]` Например, если нужно восстановить («починить») файловую систему на некотором устройстве `/dev/sdb2`, следует воспользоваться командой: «`sudo fsck -y /dev/sdb2`» Опция `-y` необходима, т. к. при её отсутствии придётся слишком часто давать подтверждение.

```

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. filesystems 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 op-
    tions.

    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 each filesystem that is checked.

```

Рис. 2.32: Описание команды fsck

Команда kill:

посылает сигнал процессу или выводит список допустимых сигналов. Имеет следующий синтаксис: kill [опции] PID, где PID – это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам. Например, команда «kill -KILL 3121» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс

```

KILL(1)                                User Commands                                KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly
    useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified
    in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process
    groups; see the PGID column in ps command output. A PID of -1 is special; it indicates all
    processes except the kill process itself and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number.
        The behavior of signals is explained in signal(7) manual page.

    -q, --queue value
        Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer
        to be sent with the signal. If the receiving process has installed a handler for this
        signal using the SA_SIGINFO flag to sigaction(2), then it can obtain this data via the
        si_value field of the siginfo_t structure.

    -l, --list [signal]
        List available signals.

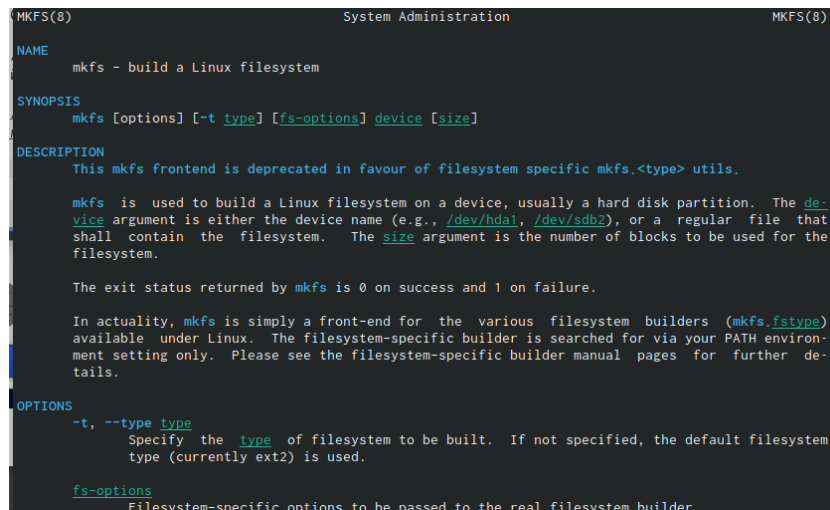
Manual page kill(1) line 1 (press h for help or q to quit)

```

Рис. 2.33: Описание команды kill

Команда `mkfs`:

создаёт новую файловую систему Linux. Имеет следующий синтаксис: `mkfs [-V] [-t fstype] [fs-options] filesystem [blocks]` `mkfs` используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента `filesystem` для файловой системы может выступать или название устройства (например, `/dev/hda1`, `/dev/sdb2`) или точка монтирования (например, `/`, `/usr`, `/home`). Аргументом `blocks` указывается количество блоков, которые выделяются для использования этой файловой системой. По окончании работы `mkfs` возвращает 0 - в случае успеха, а 1 - при неудачной операции. Например, команда «`mkfs -t ext2 /dev/hdb1`» создаёт файловую систему типа `ext2` в разделе `/dev/hdb1` (второй жёсткий диск)



```

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 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) is used.

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

Рис. 2.34: Описание команды `mkfs`

3 Контрольные вопросы

- 1) Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th». Из рисунка видно, что на моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. devtmpfs позволяет ядру создать экземпляр tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в devtmpfs. devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств. tmpfs – временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной памяти. Файловая система tmpfs предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. ext4 – имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extent file system», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность

файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных. Характеристики:

- максимальный размер файла: 16 ТБ;
- максимальный размер раздела: 16 ТБ;
- максимальный размер имени файла: 255 символов. Рекомендации по использованию:
- наилучший выбор для SSD;
- наилучшая производительность по сравнению с предыдущими Ext-системами;
- она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3. ISO 9660 – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CDROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

2) Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел – / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам.(рис.2.20)

/ – корень Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы,

флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге. - /BIN – бинарные файлы пользователя Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления. - /SBIN – системные исполняемые файлы Так же 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. - /ETC – конфигурационные файлы В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации Init Scripts, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ. - /DEV – файлы устройств В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры – это просто файлы в каталоге /dev/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. - /PROC – информация о процессах По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов. - /VAR – переменные файлы Название каталога /var говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кешы, базы данных и так далее. - /TMP – временные файлы В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию. - /USR – программы пользователя Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию. - /HOME – домашняя папка В этой

папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т. д. - /BOOT – файлы загрузчика Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящиеся в каталоге /boot/grub. - /LIB – системные библиотеки Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах /bin и /sbin. - /OPT – дополнительные программы В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями. - /MNT – монтирование В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы. - /MEDIA – съемные носители В этот каталог система монтирует все подключаемые внешние накопители –USB флешки, оптические диски и другие носители информации. - /SRV – сервер В этом каталоге содержатся файлы серверов и сервисов. - /RUN - процессы Каталог, содержащий PID файлы процессов, похожий на /var/run, но в отличие от него, он размещен в TMPFS, а поэтому после перезагрузки все файлы теряются. 3) Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount. 4) Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду fsck. 5) Файловую систему можно создать, используя команду mkfs. Ее краткое описание дано в пункте 5) в ходе выполнения заданий лабораторной работы. 6) Для просмотра текстовых файлов существуют следующие команды: - cat Задача команды cat очень проста – она читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис утилиты: cat [опции] файл1 файл2 ... Основные опции: -b – нумеровать только непустые строки -E – показывать символ \$ в конце каждой строки -n – нумеровать все строки -s – удалять пустые повторяющиеся строки -T – отображать табуляции в виде ^I -h – отобразить справку -v – версия утилиты - nl Команда nl действует

аналогично команде `cat`, но выводит еще и номера строк в столбце слева. - `less` Существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста. Синтаксис аналогичный синтаксису команды `cat`. Некоторые опции: `-g` – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения) `-N` – показывать номера строк - `head` Команда `head` выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита. Синтаксис аналогичный синтаксису команды `cat`. Основные опции: `-c` (`-bytes`) – позволяет задавать количество текста не в строках, а в байтах `-n` (`-lines`) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию `-q` (`-quiet`, `-silent`) – выводит только текст, не добавляя к нему название файла `-v` (`-verbose`) – перед текстом выводит название файла `-z` (`-zero-terminated`) – символы перехода на новую строку заменяет символами завершения строк - `tail` Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды `cat`. Основные опции: `-c` – вывести указанное количество байт с конца файла `-f` – обновлять информацию по мере появления новых строк в файле `-n` – выводить указанное количество строк из конца файла `-pid` – используется с опцией `-f`, позволяет завершить работу утилиты, когда завершится указанный процесс `-q` – не выводить имена файлов `-retry` – повторять попытки открыть файл, если он недоступен `-v` – выводить подробную информацию о файле 7) Утилита `cp` позволяет полностью копировать файлы и директории. Синтаксис: `cp [опции] файл-источник файл-приемник` После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем. Основные опции: `-attributes-only` – не копировать содержимое файла, а только флаги доступа и владельца `-f`, `-force` – перезаписывать

существующие файлы -i, -interactive – спрашивать, нужно ли перезаписывать существующие файлы -L – копировать не символические ссылки, а то, на что они указывают -n – не перезаписывать существующие файлы -P – не следовать символическим ссылкам -r – копировать папку Linux рекурсивно -s – не выполнять копирование файлов в Linux, а создавать символические ссылки -u – скопировать файл, только если он был изменён -x – не выходить за пределы этой файловой системы -p – сохранять владельца, временные метки и флаги доступа при копировании -t – считать файл-приемник директорией и копировать файл-источник в эту директорию 8) Команда mv используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий. Синтаксис: mv [-опции] старый_файл новый_файл Основные опции: -help – выводит на экран официальную документацию об утилите -version – отображает версию mv -b – создает копию файлов, которые были перемещены или перезаписаны -f – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла -i – наоборот, будет спрашивать разрешение у владельца -n – отключает перезапись уже существующих объектов -strip-trailing-slashes — удаляет завершающий символ / у файла при его наличии -t [директория] — перемещает все файлы в указанную директорию -u – осуществляет перемещение только в том случае, если исходный файл новее объекта назначения -v – отображает сведения о каждом элементе во время обработки команды Команда rename также предназначена, чтобы переименовать файл. Синтаксис: rename [опции] старое_имя новое_имя файлы Основные опции: -v – вывести список обработанных файлов -n – тестовый режим, на самом деле никакие действия выполнены не будут -f – принудительно перезаписывать существующие файлы 9) Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации. Права доступа к файлу или каталогу можно изменить,

воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Синтаксис команды: `chmod режим имя_файла` Режим имеет следующие компоненты структуры и способ записи: `=` установить право `-` лишить права `+` дать право `r` чтение `w` запись `x` выполнение `u` (user) владелец файла `g` (group) группа, к которой принадлежит владелец файла `o` (others) все остальные

4 Вывод

Вывод: в ходе выполнения лабораторной работы я ознакомился с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрел практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.