

Отчёт по лабораторной работе №5

Операционные системы

Никулина Ксения Ильинична

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Вывод	16
4	Ответы на контрольные вопросы	17

Список иллюстраций

2.1	Первая часть	6
2.2	Вторая часть	6
2.3	Третья часть	7
2.4	Выполнение второго пункта	8
2.5	Выполнение второго пункта	8
2.6	Выполнение второго пункта	8
2.7	Выполнение второго пункта	8
2.8	Выполнение третьего пункта	9
2.9	Изменение прав	9
2.10	Просмотр содержимого файла	10
2.11	Выполнение четвёртого пункта	11
2.12	Команда mount	12
2.13	Команда fsck	13
2.14	Команда mkfs	14
2.15	Команда 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

1. Выполним действия, которые описаны в первой части лабораторной работы.
(рис. 2.1), (рис. 2.2), (рис. 2.3)

```
kinikulina@dk8n52 /usr/include/sys $ cp io.h /afs/.dk.sci.pfu.edu.ru/home/k/i/kinikulina
kinikulina@dk8n52 /usr/include/sys $ cd
kinikulina@dk8n52 ~ $ mv io.h equipment
kinikulina@dk8n52 ~ $ ls
bin          kinikulina.github.io  tmp          Изображения  Шаблоны
blog         ls1                   work         лабы
'cd ~/.pub'  public               Видео       Музыка
equipment    public_html          Документы   Общедоступные
hello        study_2022-2023_arh-pc Загрузки    'Рабочий стол'
```

Рис. 2.1: Первая часть

```
kinikulina@dk8n52 ~ $ head [-2] hello
head: невозможно открыть '[-2]' для чтения: Нет такого файла или каталога
==> hello <==
kinikulina@dk8n52 ~ $
```

Рис. 2.2: Вторая часть

```

kinikulina@dk8n52 ~ $ touch hello
kinikulina@dk8n52 ~ $ ls
bin                ls1                work              лабы
blog               public            Видео             Музыка
'cd ~/.pub'        public_html       Документы        Общедоступные
hello              study_2022-2023_arh-pc  Загрузки        'Рабочий стол'
kinikulina.github.io tmp              Изображения      Шаблоны
kinikulina@dk8n52 ~ $ cat hello
kinikulina@dk8n52 ~ $ less hello
kinikulina@dk8n52 ~ $ head [-2] hello
head: невозможно открыть '[-2]' для чтения: Нет такого файла или каталога
==> hello <==
kinikulina@dk8n52 ~ $ tail [-1] hello
tail: невозможно открыть '[-1]' для чтения: Нет такого файла или каталога
==> hello <==

```

Рис. 2.3: Третья часть

2. 2.1-2.8. (рис. 2.4), (рис. 2.8), (рис. 2.9), (рис. 2.10)

- Копируем файл `/usr/include/aio.h` в домашний каталог (команда «`cp /usr/include/aio.h ~`») и назоваем его `equipment` (команда «`mv io.h equipment`»).
- В домашнем каталоге создаем директорию `~/ski.places` (команда «`mkdir ski.places`»). Перемещаем файл `equipment` в каталог `~/ski.places` (команда «`mv equipment ski.places`»).
- Переименовываем файл `~/ski.places/equipment` в `~/ski.places/equiplist` (команда «`mv ski.places/equipment ski.places/equiplist`»).
- Создаем в домашнем каталоге файл `abc1` (команда «`touch abc1`») и копируем его в каталог `~/ski.places` (команда «`cp abc1 ski.places`»), называем его `equiplist2` (команда «`mv ski.places/abc1 ski.places/equiplist2`»).
- Создаем каталог с именем `equipment` в каталоге `~/ski.places` (команда «`mkdir ski.places/equipment`»).
- Перемещаем файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment` (команда «`mv ski.places/equiplist ski.places/equiplist2 ski.places/equipment`»).

- Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~/ski.plases (команда «mv newdir ski.plases») и называем его plans (команда «mv ski.plases/newdir ski.plases/plans»)

```
kinikulina@dk8n52 ~ $ mkdir ~/ski.plases
kinikulina@dk8n52 ~ $ ls
bin                ls1
blog               public
'cd ~/.pub'        public_html
equipment          ski.plases
hello              study_2022-2023_arh-pc
kinikulina.github.io tmp
```

Рис. 2.4: Выполнение второго пункта

```
kinikulina@dk8n52 ~ $ mv equipment ~/ski.plases
kinikulina@dk8n52 ~ $ ls ~/ski.plases
equipment
```

Рис. 2.5: Выполнение второго пункта

```
kinikulina@dk8n52 ~/ski.plases $ mv equipment equiplist
kinikulina@dk8n52 ~/ski.plases $ ls
equiplist
```

Рис. 2.6: Выполнение второго пункта

```
kinikulina@dk8n52 ~ $ touch abc1
kinikulina@dk8n52 ~ $ cp abc1 ~/ski.plases
kinikulina@dk8n52 ~ $ cd ~/ski.plases
kinikulina@dk8n52 ~/ski.plases $ mv abc1 equiplist2
kinikulina@dk8n52 ~/ski.plases $ ls
equiplist equiplist2
```

Рис. 2.7: Выполнение второго пункта

3. Определяем опции команды `chmod`, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «`mkdir australia`», «`mkdir play`», «`touch my_os`», «`touch feathers`» (рис. 2.8)

```
chmod u+r+x+w australia
chmod g+r-w-x australia
chmod o+r-w-x australia
```

Рис. 2.8: Выполнение третьего пункта

Затем используем команды (рис. 2.9)

`drwxr-r- ... australia`: команда «`chmod 744 australia`»,

`drwx-x-x ... play`: команда «`chmod 711 play`»,

`-r-xr-r- ... my_os`: команды «`chmod 544 my_os`»,

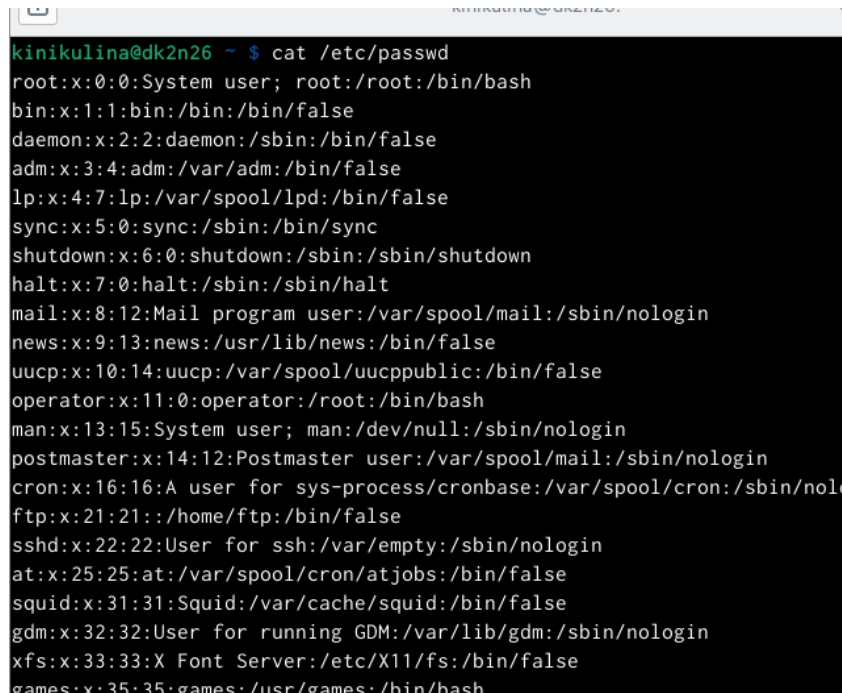
`-rw-rw-r- ... feathers`: команды «`chmod 664 feathers`».

Командой `ls` проверяем правильность действий.

```
-rw-r--r-- 1 kinikulina studsci  0 мар  9 09:38 abc1
-rwxr--r-- 1 kinikulina studsci  0 мар  9 09:48 australia
drwxr-xr-x 2 kinikulina studsci 2048 фев 22 15:22 bin
drwxr-xr-x 12 kinikulina studsci 2048 фев 22 16:02 blog
-rw-r--r-- 1 kinikulina studsci 590 окт 13 11:13 'cd ~/.pub'
-rw-rw-r-- 1 kinikulina studsci  0 мар  9 09:57 feather
-rw-r--r-- 1 kinikulina studsci  0 мар  9 09:11 hello
drwxr-xr-x 3 kinikulina studsci 2048 фев 22 15:49 kinikulina.git
-rw-r--r-- 1 kinikulina studsci 10239 мар  1 15:56 ls1
-r-xr--r-- 1 kinikulina studsci  0 мар  9 09:57 my_os
-rwx--x--x 1 kinikulina studsci  0 мар  9 09:56 play
drwxr-xr-x 2 kinikulina studsci 2048 мар  2 17:54 public
```

Рис. 2.9: Изменение прав

4. Затем посмотрим содержимое файла `/etc/passwd` (команда «`cat /etc/passwd`»).(рис. 2.10).



```
kinikulina@dk2n26 ~ $ cat /etc/passwd
root:x:0:0:System user; root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
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 program user:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
man:x:13:15:System user; man:/dev/null:/sbin/nologin
postmaster:x:14:12:Postmaster user:/var/spool/mail:/sbin/nologin
cron:x:16:16:A user for sys-process/cronbase:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/home/ftp:/bin/false
sshd:x:22:22:User for ssh:/var/empty:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/bin/false
squid:x:31:31:Squid:/var/cache/squid:/bin/false
gdm:x:32:32:User for running GDM:/var/lib/gdm:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/bin/false
games:x:35:35:games:/usr/games:/bin/bash
```

Рис. 2.10: Просмотр содержимого файла

- Копируем файл `~/feathers` в файл `~/file.old` (команда «`cp feathers file.old`»).
- Переместим файл `~/file.old` в каталог `~/play` (команда «`mv file.ord play`»).
- Скопируем каталог `~/play` в каталог `~/fun` (команда «`cp -r play fun`»).
- Переместим каталог `~/fun` в каталог `~/play` (команда «`mv fun play`») и назовем его `games` (команда «`mv play/fun play/games`»).
- Лишим владельца файла `~/feathers` права на чтение (команда «`chmod u-r feathers`»).
- Если мы попытаемся просмотреть файл `~/feathers` командой `cat`, то получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на чтение данного файла.

- Если мы попытаемся скопировать файл ~/feathers, например, в каталог monthly, то получим отказ в доступе, по причине, описанной в предыдущем пункте.
- Дадим владельцу файла ~/feathers право на чтение (команда «chmod u+r feathers»).
- Лишим владельца каталога ~/play права на выполнение (команда «chmod u-x play»).
- Перейдем в каталог ~/play (команда «cd play»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога.
- Дадим владельцу каталога ~/play право на выполнение (команда «chmod u+x play»)(рис. 2.11).

```
drw-r-xr-x  2 kinikulina studsci 2048 мар 10 13:43 play
```

Рис. 2.11: Выполнение четвёртого пункта

- Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах:
 - 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. (рис. 2.12)

```
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

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

Рис. 2.12: Команда 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`» (рис. 2.13)

```
FCK(8)                               System Administration                               FCK(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.

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

Рис. 2.13: Команда fsck

- Команда mkfs: создаёт новую файловую систему Linux. Имеет следующий синтаксис: `mkfs [-V] [-t fstype] [fs-options] filesys [blocks]` mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента filesys для файловой системы может выступать или название устройства (например, /dev/hda1, /dev/sdb2) или точка монтирования (например, /, /usr, /home). (рис. 2.14)

```

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

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

Рис. 2.14: Команда mkfs

- 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 (рис. 2.15)

```
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>

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

Рис. 2.15: Команда kill

3 Вывод

В ходе выполнения данной лабораторной работы я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов, получила 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. Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th» (Рисунок 15). 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 CD- ROM. 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
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 систе-

мы 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: 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 Команда `nl` действует аналогично команде `cat`, но выводит еще и номера строк в столбце слева. 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 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

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: `-с` – выводить 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