

# **Отчёт по лабораторной работе №5**

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

Галиева Аделина Руслановна

# Содержание

|   |                                |    |
|---|--------------------------------|----|
| 1 | Цель работы                    | 5  |
| 2 | Выполнение лабораторной работы | 6  |
| 3 | Контрольные вопросы            | 16 |
| 4 | Выводы                         | 23 |
|   | Список литературы              | 24 |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 2.1  | Выполнение примеров . . . . .                | 7  |
| 2.2  | Выполнение примеров . . . . .                | 8  |
| 2.3  | Работа с каталогами . . . . .                | 9  |
| 2.4  | Настройка прав доступа . . . . .             | 10 |
| 2.5  | Файл /etc/passwd . . . . .                   | 11 |
| 2.6  | Работа с файлами и правами доступа . . . . . | 12 |
| 2.7  | Команда mount . . . . .                      | 12 |
| 2.8  | Команда mount . . . . .                      | 13 |
| 2.9  | Команда fsck . . . . .                       | 13 |
| 2.10 | Команда fsck . . . . .                       | 13 |
| 2.11 | Команда mkfs . . . . .                       | 14 |
| 2.12 | Команда mkfs . . . . .                       | 14 |
| 2.13 | Команда kill . . . . .                       | 14 |
| 2.14 | 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. Выполнение всех примеров, приведённых в первой части описания лабораторной работы. (рис. 2.1) (рис. 2.2)

```

argalieva@dk8n54 ~ $ cd
argalieva@dk8n54 ~ $ touch abc1
argalieva@dk8n54 ~ $ cp abc1 april
argalieva@dk8n54 ~ $ cp abc1 may
argalieva@dk8n54 ~ $ mkdir monthly
argalieva@dk8n54 ~ $ cp april may monthly
argalieva@dk8n54 ~ $ cp monthly/may monthly/june
argalieva@dk8n54 ~ $ ls monthly
april  june  may
argalieva@dk8n54 ~ $ mkdir monthly.00
argalieva@dk8n54 ~ $ cp -r monthly monthly.00
argalieva@dk8n54 ~ $ cp -r monthly.00 /tmp
argalieva@dk8n54 ~ $ mv april july
argalieva@dk8n54 ~ $ mv july monthly.00
argalieva@dk8n54 ~ $ ls monthly.00
july  monthly
argalieva@dk8n54 ~ $ mv monthly.00 monthly.01
argalieva@dk8n54 ~ $ mkdir reports
argalieva@dk8n54 ~ $ mv monthly.01 reports
argalieva@dk8n54 ~ $ mv reports/monthly.01 reports/monthly
argalieva@dk8n54 ~ $ touch may
argalieva@dk8n54 ~ $ ls -l may
-rw-r--r-- 1 argalieva studsci 0 map  9 15:39 may
argalieva@dk8n54 ~ $ chmod u+x may
argalieva@dk8n54 ~ $ ls -l may
-rwxr--r-- 1 argalieva studsci 0 map  9 15:39 may
argalieva@dk8n54 ~ $ chmod u-x may
argalieva@dk8n54 ~ $ ls -l may
-rw-r--r-- 1 argalieva studsci 0 map  9 15:39 may

```

Рис. 2.1: Выполнение примеров

```

argaliev@dk8n54 ~ $ ls -l may
-rw-r--r-- 1 argaliev studsci 0 may 9 15:39 may
argaliev@dk8n54 ~ $ mkdir monthly
Инструменты Возможно создать каталог «monthly»: Файл существует
argaliev@dk8n54 ~ $ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
argaliev@dk8n54 ~ $ chmod g - r, o - r monthly/
chmod: неверный режим: «g»
По команде «chmod --help» можно получить дополнительную информацию.
argaliev@dk8n54 ~ $ chmod g-r, o-r monthly/
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
argaliev@dk8n54 ~ $ chmod g-r,o-r monthly
argaliev@dk8n54 ~ $ touch abc1
argaliev@dk8n54 ~ $ chmod g+w abc1

```

Рис. 2.2: Выполнение примеров

2. Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовём его `equipment`. Такого нет, используем другой. В домашнем каталоге создаём директорию `~/ski.plases` и перемещаем файл `equipment` в каталог `~/ski.plases`. Переименовываем файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`. Создаём в домашнем каталоге файл `abc1` и копируем его в каталог `~/ski.plases`, назовём его `equiplist2`. Создадим каталог с именем `equipment` в каталоге `~/ski.plases`. Переместим файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment`. Создаём и перемещаем каталог `~/newdir` в каталог `~/ski.plases` и называем его `plans`. (рис. 2.3)



```

argalieva@dk8n54 ~ $ cp /usr/include/linux/sysinfo.h ~
argalieva@dk8n54 ~ $ mv sysinfo.h equipment
argalieva@dk8n54 ~ $ mkdir ski.plases
argalieva@dk8n54 ~ $ mv equipment ski.plases/
argalieva@dk8n54 ~ $ mv ski.plases/equipment ski.plases/equiplist
argalieva@dk8n54 ~ $ touch abc1
argalieva@dk8n54 ~ $ cp abc1
cp: после 'abc1' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
argalieva@dk8n54 ~ $ cp abc1 ski.plases/equiplist2
argalieva@dk8n54 ~ $ cd ski.plases/
argalieva@dk8n54 ~/ski.plases $ mkdir equipment
argalieva@dk8n54 ~/ski.plases $ mv equiplist equipment/
argalieva@dk8n54 ~/ski.plases $ mv equiplist2 equipment/
argalieva@dk8n54 ~/ski.plases $ cd
argalieva@dk8n54 ~ $ mkdir newdir
argalieva@dk8n54 ~ $ mv newdir/ ski.plases/
argalieva@dk8n54 ~ $ mv ski.plases/newdir/ ski.plases/plans

```

Рис. 2.3: Работа с каталогами

3. Определяем опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным файлам из хода работы нужные права доступа. (рис. 2.4)

```

argalieva@dk8n54 ~ $ cp abc1 ski.places/equiplist2
argalieva@dk8n54 ~ $ cd ski.places/
argalieva@dk8n54 ~/ski.places $ mkdir equipment
mkdir: невозможно создать каталог «equipment»: Файл существует
argalieva@dk8n54 ~/ski.places $ mv equiplist equipment/
mv: не удалось выполнить stat для 'equiplist': Нет такого файла или каталога
argalieva@dk8n54 ~/ski.places $ mv equiplist2 equipment/
argalieva@dk8n54 ~/ski.places $ mv equiplist equipment/
mv: не удалось выполнить stat для 'equiplist': Нет такого файла или каталога
argalieva@dk8n54 ~/ski.places $ cd
argalieva@dk8n54 ~ $ mkdir newdir
argalieva@dk8n54 ~ $ mv newdir/ ski.places/
argalieva@dk8n54 ~ $ mv ski.places/newdir/ ski.places/plans
argalieva@dk8n54 ~ $ mkdir australia play
argalieva@dk8n54 ~ $ touch my_os feathers
argalieva@dk8n54 ~ $ chmod 744 australia/
argalieva@dk8n54 ~ $ chmod 711 play/
argalieva@dk8n54 ~ $ chmod 544 my_os
argalieva@dk8n54 ~ $ chmod 664 feathers
argalieva@dk8n54 ~ $ ls -l
итого 190
drwxr-xr-x 2 argalieva studsci 6144 фев  4 19:26 21.09
-rw-rw-r-- 1 argalieva studsci   0 мар  9 15:56 abc1
drwxr--r-- 2 argalieva studsci 2048 мар  9 16:09 australia
drwxr-xr-x 2 argalieva studsci 2048 мар  9 16:55 k1a

```

Рис. 2.4: Настройка прав доступа

#### 4. Просматриваем содержимое файла /etc/passwd. (рис. 2.5)

```

argalieva@dk8n54 ~ $ 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
named:x:40:40:bind:/var/bind:/bin/false
mysql:x:60:60:MySQL program user:/dev/null:/sbin/nologin
postgres:x:70:70:PostgreSQL program user:/var/lib/postgresql:/bin/sh
nut:x:84:84:nut:/var/state/nut:/bin/false
cyrus:x:85:12:./usr/cyrus:/bin/false
vpopmail:x:89:89:./var/vpopmail:/bin/false
alias:x:200:200:./var/qmail/alias:/bin/false
qmaild:x:201:200:./var/qmail:/bin/false

```

Рис. 2.5: Файл /etc/passwd

Выполняем все указанные действия по перемещению файлов и каталогов. (рис. 2.6)

```
argalievadk8n54 ~ $ cp feathers file.old
argalievadk8n54 ~ $ mv file.old play/
argalievadk8n54 ~ $ mkdir fun
argalievadk8n54 ~ $ vp -R play/ fun/
bash: vp: команда не найдена
argalievadk8n54 ~ $ mv fun/ play/games
argalievadk8n54 ~ $ chmod u-r feathers
argalievadk8n54 ~ $ cat feathers
cat: feathers: Отказано в доступе
argalievadk8n54 ~ $ cp feathers feathers2
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
argalievadk8n54 ~ $ chmod u+r feathers
argalievadk8n54 ~ $ chmod u-x play/
argalievadk8n54 ~ $ chmod +x play/
```

Рис. 2.6: Работа с файлами и правами доступа

Если мы попытаемся просмотреть файл `feathers` командой `cat`, то нам будет отказано.

Если мы попытаемся скопировать файл `feathers`, то у нас не получится сделать это, поскольку мы ограничили себя в доступе для чтения.

5. Прочитаем ман по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.(рис. 2.7) (рис. 2.8) (рис. 2.9) (рис. 2.10) (рис. 2.11) (рис. 2.12) (рис. 2.13) (рис. 2.14)

```
argalievadk8n54 ~ $ man mount
```

Рис. 2.7: Команда `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|runbindable] mountpoint
```

Рис. 2.8: Команда mount

```
argaliev@dk8n54 ~ $ man fsck
```

Рис. 2.9: Команда 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.

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

Рис. 2.10: Команда fsck

```
argaliev@dk8n54 ~ $ man mkfs
```

Рис. 2.11: Команда 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

```

Рис. 2.12: Команда mkfs

```
argaliev@dk8n54 ~ $ man kill
argaliev@dk8n54 ~ $
```

Рис. 2.13: Команда kill

|   |               |         |
|---|---------------|---------|
| KILL(1)   | User Commands | KILL(1) |
| <b>NAME</b>   |               |         |
| kill - send a signal to a process   |               |         |
| <b>SYNOPSIS</b>   |               |         |
| kill [options] <pid> [...]  |               |         |
| <b>DESCRIPTION</b>  |               |         |
| <p>The default signal for kill is TERM. Use <b>-l</b> or <b>-L</b> to list available signals. Particularly useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: <b>-9</b>, <b>-SIGKILL</b> or <b>-KILL</b>. Negative PID values may be used to choose whole process groups; see the PGID column in ps command output. A PID of <b>-1</b> is special; it indicates all processes except the kill process itself and init.</p> |               |         |
| <b>OPTIONS</b>  |               |         |
| <p>&lt;pid&gt; [...]<br/> Send signal to every &lt;pid&gt; listed.</p>  |               |         |
| <p><b>-&lt;signal&gt;</b></p>   |               |         |

Рис. 2.14: Команда kill

### 3 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу. Ext2FS (расширенная файловая система номер два). Многие годы ext2 была файловой системой по умолчанию в GNU/Linux. Ext2 заменила собой Extended File System (вот откуда появилось “Second” в названии). В “новой” файловой системе были исправлены некоторые проблемы, а также убраны ограничения. Отличная стабильность, комплексные инструментальные средства для спасения удаленных файлов, очень долгое время перезагрузки после аварии, есть вероятность частичной или полной потери данных после аварии. Одним из главных недостатков “традиционных” файловых систем, подобных Ext2FS, является низкая сопротивляемость к резким системным сбоям (сбой питания или авария программного обеспечения)

Ext3 (Расширенная файловая система номер три) - является наследником файловой системы Ext2FS. Ext3 совместима с Ext2, но обладает одной новой и очень интересной особенностью – запись. Процесс сохранения объекта происходит прежде чем запись в журнал. В результате мы получаем всегда последовательную файловую систему. Это приводит к тому, что при появлении проблем, проверка и восстановление происходят очень быстро. Время, потраченное на то, чтобы проверить файловую систему таким образом, пропорционально его фактическому использованию и не больше его размера.

ReiserFS (Это тоже журналируемая файловая система подобно Ext3FS, но их внутренняя структура радикально отличается. В ReiserFS используется концепция



бинарных деревьев (binary-tree), позаимствованная из программного обеспечения баз данных.

JFS (журналируемая файловая система). JFS была разработана и использовалась IBM. Вначале JFS была закрытой системой, но недавно IBM решила открыть доступ для движения свободного программного обеспечения. Внутренняя структура JFS близка к ReiserFS. Средняя стабильность, нет комплексных инструментальных средств для спасения удаленных файлов, очень быстрая перезагрузка после аварии, очень хорошее восстановление данных после аварии.

2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

- Загрузочный блок занимает первый блок файловой системы. Только корневая файловая система имеет активный загрузочный блок, хотя место для него резервируется в каждой файловой системе.
- Суперблок располагается непосредственно за загрузочным блоком и содержит самую общую информацию о ФС (размер ФС, размер области индексных дескрипторов, их число, список свободных блоков, свободные индексные дескрипторы и т. д.). Суперблок всегда находится в оперативной памяти. Различные версии ОС Unix способны поддерживать разные типы файловых систем. Поэтому у структуры суперблока могут быть варианты (сведения о свободных блоках, например, часто хранятся не как список, а как шкала бит), но суперблок всегда располагается за загрузочным блоком. При монтировании файловой системы в оперативной памяти создается копия ее суперблока. Все последующие операции по созданию и удалению файлов влекут изменения копии суперблока в оперативной памяти. Эта копия периодически записывается на магнитный диск. Обычно причиной повреждения файловой системы является отключение электропитания (или зависание ОС) в тот момент, когда система производит копирование суперблока из оперативной памяти на магнитный диск.

- Область индексных дескрипторов содержит описатели файлов (inode). С каждым файлом связан один inode, но одному inode может соответствовать несколько файлов. Binode хранится вся информация о файле, кроме его имени. Область индексных дескрипторов имеет фиксированный формат и располагается непосредственно за суперблоком. Общее число описателей и, следовательно, максимальное число файлов задается в момент создания файловой системы. Описатели нумеруются натуральными числами. Первый описатель используется ОС для описания специального файла (файла «Плохих блоков»). То есть поврежденные блоки раздела рассматриваются ОС как принадлежащие к специальному файлу и поэтому считаются «занятыми». Вторым описывает корневой каталог файловой системы.
- В области данных расположены как обычные файлы, так и файлы каталогов (в том числе корневой каталог). Специальные файлы представлены в ФС только записями в соответствующих каталогах и индексными дескрипторами специального формата, т. е. места в области памяти не занимают.

3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе? Команда `cat` - позволяет вывести на экран содержимое любого файла, однако в таком виде эта команда практически не используется. Если файл слишком большой, то его содержимое пролистается на экране, а Вы увидите только последние строки файла. С помощью этой команды можно комбинировать и объединять копии файлов, а также создавать новые файлы. Если набрать просто в командной строке `cat` и нажать `Enter`, то можно вводить (и соответственно видеть) текст на экране. Повторное нажатие клавиши `Enter` удвоит строку и позволит начать следующую. Когда текст набран, следует одновременно нажать клавиши `Ctrl` и `d`.

4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы? Некорректность файловой системы может возникать:

- В результате насильственного прерывания операций ввода-вывода, выполняемых непосредственно с диском.
- В результате нарушения работы дискового кэша. Кэширование данных с диска предполагает, что в течение некоторого времени результаты операций ввода-вывода никак не сказываются на содержимом диска — все изменения происходят с копиями блоков диска, временно хранящихся в буферах оперативной памяти (в этих буферах оседают данные из пользовательских файлов и служебная информация файловой системы, такая как каталоги, индексные дескрипторы, списки свободных, занятых и поврежденных блоков и т. п.)

5. Как создаётся файловая система? Общее дерево файлов и каталогов системы Linux формируется из отдельных “ветвей”, соответствующих различным физическим носителям. В UNIX нет понятия “форматирования диска” (и команды форматирования), а используется понятие “создание файловой системы”. Когда мы получаем новый носитель, например, жесткий диск, мы должны создать на нем файловую систему. То есть каждому носителю ставится в соответствие отдельная файловая система. Чтобы эту файловую систему использовать для записи в нее файлов, надо ее вначале подключить в общее дерево каталогов (“смонтировать”). Вот и получается, что можно говорить о монтировании файловых систем или о монтировании носителей (с созданными на них файловыми системами). Например, создается файловая система типа ext2fs. Создание файловой системы типа ext2fs подразумевает создание в данном разделе на диске суперблока, таблицы индексных дескрипторов и совокупности блоков данных. Делается все это все с помощью команды mkfs. В простейшем случае достаточно дать эту команду в следующем формате:

[root]# mkfs -t ext2 /dev/hda5, где /dev/hda5 надо заменить указанием на соответствующее устройство или раздел. Например, если вы хотите создать файловую систему на дискете, то команда примет вид:

```
[root]# mkfs -t ext2 /dev/fd0
```

После выполнения команды `mkfs` в указанном разделе будет создана файловая система `ext2fs`. В новой файловой системе автоматически создается один каталог с именем `lost+found`. Он используется в экстренных случаях программой `fsck`, поэтому не удаляйте его. Для того, чтобы начать работать с новой файловой системой, необходимо подключить ее в общее дерево каталогов, что делается с помощью команды `mount`. В качестве параметров команде `mount` надо, как минимум, указать устройство и “точку монтирования”. Точкой монтирования называется тот каталог в уже существующем и известном системе дереве каталогов, который будет теперь служить корневым каталогом для подключаемой файловой системы. После монтирования файловой системы в каталог `/mnt/disk2` прежнее содержимое этого каталога станет для вас недоступно до тех пор, пока вы не размонтируете вновь подключенную файловую систему. Прежнее содержимое не уничтожается, а просто становится временно недоступным. Поэтому в качестве точек монтирования лучше использовать пустые каталоги (заранее заготовленные).

6. Дайте характеристику командам, которые позволяют просмотреть текстовые файлы. Для просмотра небольших файлов удобно пользоваться командой `cat`. Формат команды: `cat имя-файла`

Для просмотра больших файлов используйте команду `less` — она позволяет осуществлять постраничный просмотр файлов (длина страницы соответствует размеру экрана). Формат команды: `less имя-файла`

Для управления процессом просмотра можно использовать следующие управляющие клавиши: - `Space` — переход на следующую страницу, - `ENTER` — сдвиг вперед на одну строку, - `b` — возврат на предыдущую страницу, - `h` — обращение

за подсказкой, - q — выход в режим командной строки.

Для просмотра начала файла можно воспользоваться командой `head`. По умолчанию она выводит первые 10 строк файла. Формат команды: `head [-n] имя-файла`, где `n` — количество выводимых строк.

Команда `tail` выводит несколько (по умолчанию 10) последних строк файла. Формат команды: `tail [-n] имя-файла`, где `n` — количество выводимых строк.

7. Приведите основные возможности команды `cp` в Linux. Копирование отдельных файлов Для копирования файла следует использовать утилиту `cp` с аргументами, представленными путями к исходному и целевому файлам.

Копирование файлов в другую директорию В том случае, если в качестве пути к целевому файлу используется путь к директории, исходные файлы будут скопированы в эту целевую директорию.

Команда `cp -r` Для копирования директорий целиком следует использовать команду `cp -r` (параметр `-r` позволяет осуществлять рекурсивное копирование всех файлов из всех поддиректорий).

Копирование множества файлов в директорию Вы также можете использовать утилиту `cp` для копирования множества файлов в одну директорию. В этом случае последний аргумент (аргумент, указывающий на цель) должен быть представлен путем к директории.

Команда `cp -i` Для предотвращения перезаписи существующих файлов в ходе использования утилиты `cp` следует использовать параметр `-i` (для активации интерактивного режима копирования).

8. Назовите и дайте характеристику командам перемещения и переименования файлов и каталогов. Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов. Формат команды `mv`: `mv [-опции] старый_файл новый_файл` Примеры:

- Переименование файлов в текущем каталоге. Изменить название файла `april` на `july` в домашнем каталоге: `cd mv april july`

- Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00: `mv july monthly.00` `ls monthly.00` Результат: april july june may. Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию `i`.
  - Переименование каталогов в текущем каталоге. Переименовать каталог monthly.00 в monthly.01 `mv monthly.00 monthly.01`
  - Перемещение каталога в другой каталог. Переместить каталог monthly.01 в каталог reports: `mkdir reports` `mv monthly.01 reports`
  - Переименование каталога, не являющегося текущим. Переименовать каталог reports/monthly.01 в reports/monthly: `mv reports/monthly.01 reports/monthly`
9. Что такое права доступа? Как они могут быть изменены? Права доступа — совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам). Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Формат команды: `chmod режим имя_файла` Режим (в формате команды) 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 В работе с правами доступа можно использовать их цифровую запись (восьмеричное значение) вместо символьной

## 4 Выводы

Я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

## **Список литературы**