

Лабораторная работа №5

Операционные системы

Сячинова Ксения Ивановна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответы на контрольные вопросы	16
4	Вывод	21

Список иллюстраций

2.1	Примеры	7
2.2	Примеры	7
2.3	Примеры	8
2.4	Ход работы	9
2.5	Ход работы	10
2.6	Работа с <code>chmod</code>	10
2.7	<code>cat /etc/passwd</code>	12
2.8	4 пункт	12
2.9	<code>MOUNT</code>	13
2.10	<code>FSCK</code>	14
2.11	<code>MKFS</code>	14

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

- Скопируем файл `~/abc1` в файл `april` и в файл `may`. Для этого создадим файл `abc1`, используя команду `touch abc1`, далее осуществим копирование с помощью команд `ср abc1 april` и `ср abc1 may`.
- Скопируем файлы `april` и `may` в каталог `monthly`, используя команды `mkdir monthly` – для создания каталога `monthly` и `ср april may monthly` – для копирования.
- Скопируем файл `monthly/may` в файл с именем `june`. Выполним команды `ср monthly/may monthly/june` и `ls monthly` (для просмотра содержимого каталога).
- Скопируем каталог `monthly` в каталог `monthly.00`. Для этого создадим каталог `monthly.00` командой `mkdir monthly.00` и осуществим копирование, используя команду `ср -r monthly monthly.00` (команда `ср` с опцией `r` (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).
- Скопируем каталог `monthly.00` в каталог `/tmp`, используя команду `ср -r monthly.00 /tmp`.

```

kisyachinova@dk6n58 ~ $ cd
kisyachinova@dk6n58 ~ $ touch abc1
kisyachinova@dk6n58 ~ $ cp abc1 april
kisyachinova@dk6n58 ~ $ cp abc1 may
kisyachinova@dk6n58 ~ $ mkdir monthly
kisyachinova@dk6n58 ~ $ cp april may monthly
kisyachinova@dk6n58 ~ $ cp monthly/may monthly/ju
kisyachinova@dk6n58 ~ $ ls monthly
april  june  may
kisyachinova@dk6n58 ~ $ mkdir monthly.00
kisyachinova@dk6n58 ~ $ cp -r monthly monthly.00
kisyachinova@dk6n58 ~ $ cp -r ,onthly.00 /tmp
cp: не удалось выполнить stat для ',onthly.00': Н
kisyachinova@dk6n58 ~ $ cp -r monthly.00 /tmp

```

Рис. 2.1: Примеры

- Изменим название файла `april` на `july` в домашнем каталоге, используя команду «`mv april july`».
- Переместим файл `july` в каталог `monthly.00` с помощью команды «`mv july monthly.00`». Проверим результат командой «`ls monthly.00`».
- Переименуем каталог `monthly.00` в `monthly.01`, используя команду «`mv monthly.00 monthly.01`».
- Переместим каталог `monthly.01` в каталог `reports`. Для этого создадим каталог `reports` помощью команды «`mkdir reports`» и выполним перемещение командой «`mv monthly.01 reports`».
- Переименуем каталог `reports/monthly.01` в `reports/monthly` командой «`mv reports/monthly.01 reports/monthly`» (рис. 2.2)

```

kisyachinova@dk6n58 ~ $ cd
kisyachinova@dk6n58 ~ $ mv april july
kisyachinova@dk6n58 ~ $ mv july monthly.00
kisyachinova@dk6n58 ~ $ ls monthly.00
july  monthly
kisyachinova@dk6n58 ~ $ ls monthly
april  june  may
kisyachinova@dk6n58 ~ $ mv monthly.00 monthly.01
kisyachinova@dk6n58 ~ $ mkdir reports
kisyachinova@dk6n58 ~ $ mv monthly.01 reports
kisyachinova@dk6n58 ~ $ mv reports/monthly.01 reports/monthly

```

Рис. 2.2: Примеры

- создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may» (изменение прав), «ls -l may».
- лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле)
- Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly» (создание каталога), «chmod go-r monthly» (изменение прав).
- Создаем файл ~/abc1 с правом записи для членов группы, используя команды: «touch abc1» (создание файла), «chmod g+w abc1» (изменение прав). (рис. 2.3)

```
kisyachinova@dk6n58 ~ $ cd
kisyachinova@dk6n58 ~ $ touch may
kisyachinova@dk6n58 ~ $ ls -l may
-rw-r--r-- 1 kisyachinova studsci 0 anp 28 18:48 may
kisyachinova@dk6n58 ~ $ chmod u+x may
kisyachinova@dk6n58 ~ $ ls -l may
-rwxr--r-- 1 kisyachinova studsci 0 anp 28 18:48 may
kisyachinova@dk6n58 ~ $ chmod u-x may
kisyachinova@dk6n58 ~ $ ls -l may
-rw-r--r-- 1 kisyachinova studsci 0 anp 28 18:48 may
kisyachinova@dk6n58 ~ $ cd
kisyachinova@dk6n58 ~ $ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
kisyachinova@dk6n58 ~ $ chmod g-r monthly
kisyachinova@dk6n58 ~ $ chmod o-r monthly
kisyachinova@dk6n58 ~ $ cd
kisyachinova@dk6n58 ~ $ touch abc1
kisyachinova@dk6n58 ~ $ chmod g+w abc1
```

Рис. 2.3: Примеры

2.

- Копируем файл /usr/include/aio.h в домашний каталог (команда «cp /usr/include/aio.h ~») и называем 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/equipment equiplist2 ski.places/equipment»).
- Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~/ski.places (команда «mv newdir ski.places») и называем его plans (команда «mv ski.places/newdir ski.places/plans»).(рис. 2.4)

```
kisyachinova@dk6n58 ~ $ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
kisyachinova@dk6n58 ~ $ cp /usr/include/sys/io.h ~
kisyachinova@dk6n58 ~ $ mv io.h equipment
kisyachinova@dk6n58 ~ $ ls
abc1  equipment  may  monthly  my_os  reports  tmp  tutorial  work  Видео  Докуме
kisyachinova@dk6n58 ~ $ mkdir ski.places
kisyachinova@dk6n58 ~ $ mv equipment ski.places
kisyachinova@dk6n58 ~ $ mv ~/ski.places/equipment ~/ski.places/equiplist
kisyachinova@dk6n58 ~ $ ls ski.places
equiplist
```

Рис. 2.4: Ход работы

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. 2.5)

```

kisyachinova@dk6n58 ~ $ touch abc1
kisyachinova@dk6n58 ~ $ cp abc1 ski.places
kisyachinova@dk6n58 ~ $ mv ski.places/abc1 aki.places/equiplist2
mv: невозможно переместить 'ski.places/abc1' в 'aki.places/equiplist2': Нет такого файла или каталога
kisyachinova@dk6n58 ~ $ mv ski.places/abc1 ski.places/equiplist2
kisyachinova@dk6n58 ~ $ ls ski.places
equiplist equiplist2
kisyachinova@dk6n58 ~ $ mkdir newdir
kisyachinova@dk6n58 ~ $ mv newsir plans
mv: не удалось выполнить stat для 'newsir': Нет такого файла или каталога
kisyachinova@dk6n58 ~ $ mv newdir plans
kisyachinova@dk6n58 ~ $ mv plans ski.places
kisyachinova@dk6n58 ~ $ ls
abc1  may  monthly  my_os  reports  ski.places  tmp  tutorial  work  Видео  Документы  Загрузки  Изображения
kisyachinova@dk6n58 ~ $ mkdir australia
kisyachinova@dk6n58 ~ $ mkdir play
kisyachinova@dk6n58 ~ $ touch my_os
kisyachinova@dk6n58 ~ $ touch feathers

```

Рис. 2.5: Ход работы

Затем используем команды `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.(рис. 2.6)

```

kisyachinova@dk6n58 ~ $ mkdir australia
kisyachinova@dk6n58 ~ $ mkdir play
kisyachinova@dk6n58 ~ $ touch my_os
kisyachinova@dk6n58 ~ $ touch feathers
kisyachinova@dk6n58 ~ $ chmod 744 australia
kisyachinova@dk6n58 ~ $ chmod 711 play
kisyachinova@dk6n58 ~ $ chmod 544 my_os
kisyachinova@dk6n58 ~ $ chmod 644 feathers
kisyachinova@dk6n58 ~ $ ls -l
итого 34
-rw-rw-r-- 1 kisyachinova studsci 0 апр 28 18:53 abc1
drwxr--r-- 2 kisyachinova studsci 2048 апр 28 18:56 australia
-rw-r--r-- 1 kisyachinova studsci 0 апр 28 18:57 feathers
-rw-r--r-- 1 kisyachinova studsci 0 апр 28 18:48 may
drwx--x--x 2 kisyachinova studsci 2048 апр 28 18:45 monthly
dr-xr--r-- 2 kisyachinova studsci 2048 апр 28 18:57 my_os
drwx--x--x 2 kisyachinova studsci 2048 апр 28 18:57 play
drwxr-xr-x 3 kisyachinova studsci 2048 апр 28 18:48 reports
drwxr-xr-x 3 kisyachinova studsci 2048 апр 28 18:55 ski.places
drwxr-xr-x 2 kisyachinova studsci 2048 апр 27 19:49 tmp
drwxr-xr-x 3 kisyachinova studsci 2048 апр 21 18:00 tutorial
drwxr-xr-x 3 kisyachinova studsci 2048 апр 27 18:37 work
drwxr-xr-x 2 kisyachinova studsci 2048 сен 2 2021 Видео
drwxr-xr-x 2 kisyachinova studsci 2048 апр 28 14:11 Документы
drwxr-xr-x 2 kisyachinova studsci 2048 окт 21 2021 Загрузки
drwxr-xr-x 2 kisyachinova studsci 4096 апр 28 18:24 Изображения
drwxr-xr-x 2 kisyachinova studsci 2048 апр 21 13:45 Музыка
drwxr-xr-x 2 kisyachinova studsci 2048 сен 2 2021 Общедоступные
drwxr-xr-x 3 kisyachinova studsci 2048 апр 21 13:46 Программирование

```

Рис. 2.6: Работа с chmod

4. После этого:

- Просмотрим содержимое файла `/etc/passwd` (команда «`cat /etc/passwd`»).

- 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. 2.7) (рис. 2.8)

```
kisyachinova@dk6n58 ~ $ 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:cron:/var/spool/cron:/bin/false
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:named:/var/bind:/bin/false
```

Рис. 2.7: cat /etc/passwd

```
kisyachinova@dk6n58 ~ $ cp feathers file.old
kisyachinova@dk6n58 ~ $ mv file.old play
kisyachinova@dk6n58 ~ $ cp -r play fun
kisyachinova@dk6n58 ~ $ mv fun play
kisyachinova@dk6n58 ~ $ mv play/fun play/games
kisyachinova@dk6n58 ~ $ chmod u-r feathers
kisyachinova@dk6n58 ~ $ cat reathers
cat: reathers: Нет такого файла или каталога
kisyachinova@dk6n58 ~ $ cat feathers
cat: feathers: Отказано в доступе
kisyachinova@dk6n58 ~ $ cp feathers monthly
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
kisyachinova@dk6n58 ~ $ chmod u+r feathers
kisyachinova@dk6n58 ~ $ chmod u-x play
kisyachinova@dk6n58 ~ $ cd play
```

Рис. 2.8: 4 пункт

5. 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 распола-

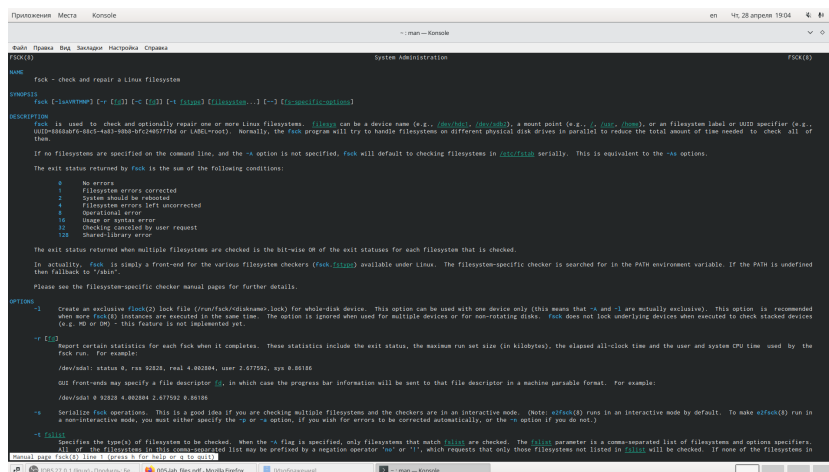


Рис. 2.10: FSCK

- **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. 2.11)

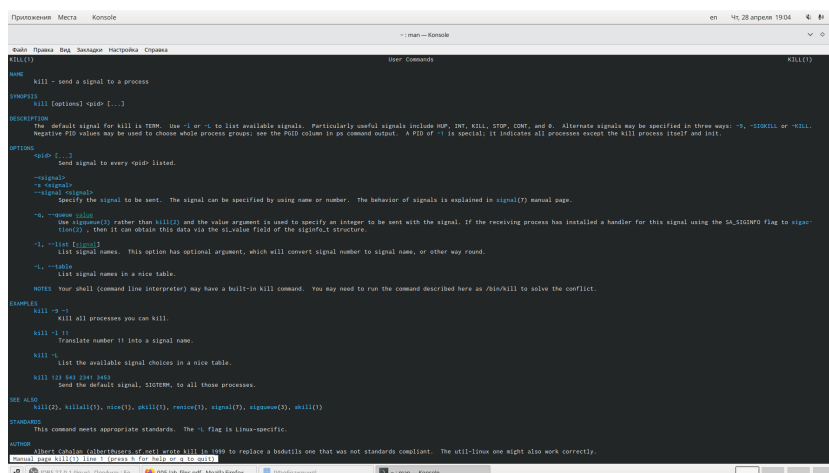
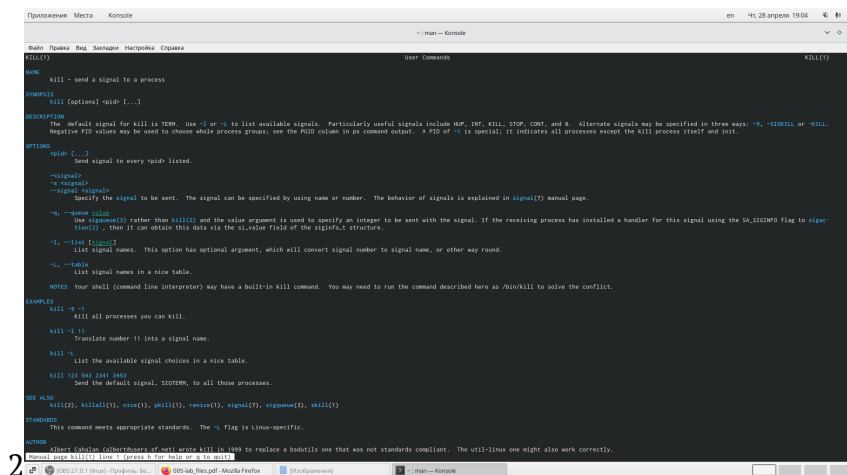


Рис. 2.11: MKFS

- **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. ??)



3 Ответы на контрольные вопросы

- 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 TB; • максимальный размер раздела: 16 TB; • 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.

- 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
- Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.
- 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`.

- Файловую систему можно создать, используя команду `mkfs`. Ее краткое описание дано в пункте 5) в ходе выполнения заданий лабораторной работы.
- 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 - 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 - 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 переа-

пись 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 - 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

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.