

Отчёт по лабораторной работе №7

Анализ файловой структуры UNIX. Команды для работы с файлами и каталогами

Умирзоков Иброхим Шомилжон угли

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	12
4	Контрольные вопросы	13

List of Figures

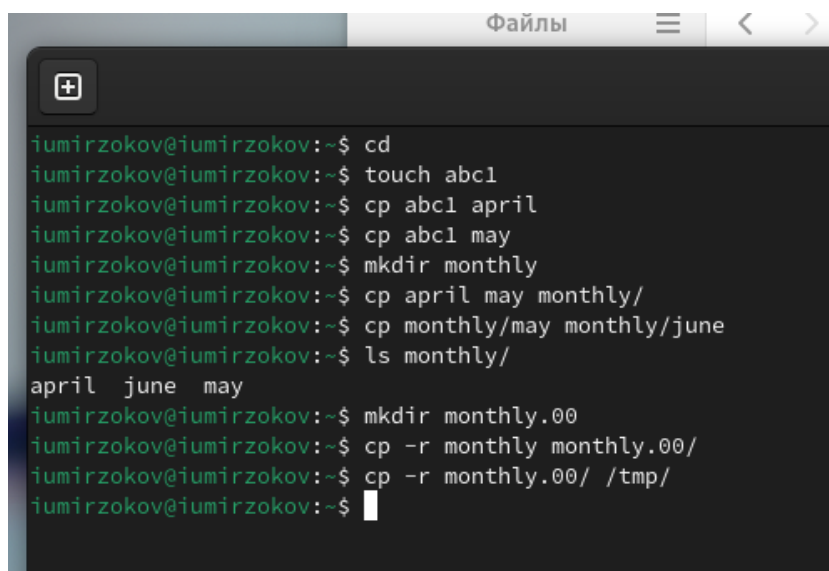
2.1	Выполнение примеров	5
2.2	Выполнение примеров	5
2.3	Выполнение примеров	6
2.4	Работа с каталогами	6
2.5	Настройка прав доступа	7
2.6	Файл /etc/passwd	8
2.7	Работа с файлами и правами доступа	9
2.8	Команда mount	9
2.9	Команда fsck	10
2.10	Команда mkfs	11
2.11	Команда kill	11

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, по проверке использования диска и обслуживанию файловой системы.

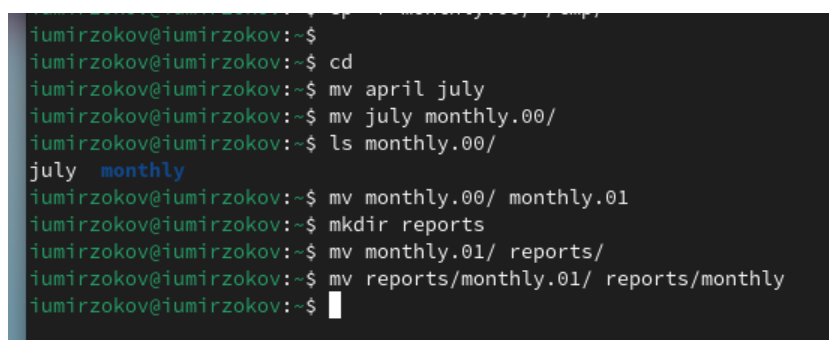
2 Выполнение лабораторной работы

1. Выполним примеры, приведённые в первой части описания лабораторной работы.



```
iumirzokov@iumirzokov:~$ cd
iumirzokov@iumirzokov:~$ touch abc1
iumirzokov@iumirzokov:~$ cp abc1 april
iumirzokov@iumirzokov:~$ cp abc1 may
iumirzokov@iumirzokov:~$ mkdir monthly
iumirzokov@iumirzokov:~$ cp april may monthly/
iumirzokov@iumirzokov:~$ cp monthly/may monthly/june
iumirzokov@iumirzokov:~$ ls monthly/
april  june  may
iumirzokov@iumirzokov:~$ mkdir monthly.00
iumirzokov@iumirzokov:~$ cp -r monthly monthly.00/
iumirzokov@iumirzokov:~$ cp -r monthly.00/ /tmp/
iumirzokov@iumirzokov:~$
```

Figure 2.1: Выполнение примеров



```
iumirzokov@iumirzokov:~$ cd
iumirzokov@iumirzokov:~$ mv april july
iumirzokov@iumirzokov:~$ mv july monthly.00/
iumirzokov@iumirzokov:~$ ls monthly.00/
july  monthly
iumirzokov@iumirzokov:~$ mv monthly.00/ monthly.01
iumirzokov@iumirzokov:~$ mkdir reports
iumirzokov@iumirzokov:~$ mv monthly.01/ reports/
iumirzokov@iumirzokov:~$ mv reports/monthly.01/ reports/monthly
iumirzokov@iumirzokov:~$
```

Figure 2.2: Выполнение примеров

```
iumirzokov@iumirzokov:~$  
iumirzokov@iumirzokov:~$ cd  
iumirzokov@iumirzokov:~$ touch may  
iumirzokov@iumirzokov:~$ ls -l may  
-rw-r--r--. 1 iumirzokov iumirzokov 0 июн 22 10:50 may  
iumirzokov@iumirzokov:~$ chmod u+x may  
iumirzokov@iumirzokov:~$ ls -l may  
-rwxr--r--. 1 iumirzokov iumirzokov 0 июн 22 10:50 may  
iumirzokov@iumirzokov:~$ chmod u-x may  
iumirzokov@iumirzokov:~$ ls -l may  
-rw-r--r--. 1 iumirzokov iumirzokov 0 июн 22 10:50 may  
iumirzokov@iumirzokov:~$ cd  
iumirzokov@iumirzokov:~$ chmod g-r,o-r monthly/  
iumirzokov@iumirzokov:~$ chmod g+w abc1  
iumirzokov@iumirzokov:~$
```

Figure 2.3: Выполнение примеров

2.1. Скопируем файл /usr/include/sys/io.h в домашний каталог и переименуем его equipment. Такого нет, взяли другой файл.

2.2. - 2.5. В домашнем каталоге создаем директорию ski.places. и перемещаем в него файл equipment. Переименовываем файл equipment в equiplist. После этого создаем в домашнем каталоге файл abc1 и копируем его в каталог ski.places. и переименовываем в equiplist2. 2.6. - 2.7. Создаем каталог с именем equipment в каталоге ski.places. Перемещаем файлы equiplist и equiplist2 в каталог equipment. 2.8. Создаем и перемещаем каталог newdir в каталог ski.places и называем его plans.

```
iumirzokov@iumirzokov:~$  
iumirzokov@iumirzokov:~$ cp /usr/include/linux/sysinfo.h ~  
iumirzokov@iumirzokov:~$ mv sysinfo.h equipment  
iumirzokov@iumirzokov:~$ mkdir ski.places  
iumirzokov@iumirzokov:~$ mv equipment ski.places/  
iumirzokov@iumirzokov:~$ mv ski.places/equipment ski.places/equiplist  
iumirzokov@iumirzokov:~$ touch abc1  
iumirzokov@iumirzokov:~$ cp abc1 ski.places/equiplist2  
iumirzokov@iumirzokov:~$ cd ski.places/  
iumirzokov@iumirzokov:~/ski.places$ mkdir equipment  
iumirzokov@iumirzokov:~/ski.places$ mv equiplist equipment/  
iumirzokov@iumirzokov:~/ski.places$ mv equiplist2 equipment/  
iumirzokov@iumirzokov:~/ski.places$ cd  
iumirzokov@iumirzokov:~$ mkdir newdir  
iumirzokov@iumirzokov:~$ mv newdir/ ski.places/  
iumirzokov@iumirzokov:~$ mv ski.places/newdir/ ski.places/plans  
iumirzokov@iumirzokov:~$
```

Figure 2.4: Работа с каталогами

3. Определим опции команды chmod, необходимые для того, чтобы присво-

ить файлам из хода работы нужные права доступа.

- a) Australia (drwxr-r-)
- b) play (drwx-x-x)
- c) My_ос (-r-xr-r-)
- d) feathers (-rw-rw-r-)

```
iumirzokov@iumirzokov:~$  
iumirzokov@iumirzokov:~$ mkdir australia play  
iumirzokov@iumirzokov:~$ touch my_os feathers  
iumirzokov@iumirzokov:~$ chmod 744 australia/  
iumirzokov@iumirzokov:~$ chmod 711 play  
iumirzokov@iumirzokov:~$ chmod 544 my_os  
iumirzokov@iumirzokov:~$ chmod 664 feathers  
iumirzokov@iumirzokov:~$ ls -l  
итого 0  
-rw-rw-r--. 1 iumirzokov iumirzokov 0 июн 22 10:52 abc1  
drwxr--r--. 1 iumirzokov iumirzokov 0 июн 22 10:53 australia  
-rw-rw-r--. 1 iumirzokov iumirzokov 0 июн 22 10:53 feathers  
drwxr-xr-x. 1 iumirzokov iumirzokov 74 июн 21 21:55 git-extended  
-rw-r--r--. 1 iumirzokov iumirzokov 0 июн 22 10:50 may  
drwx--x--x. 1 iumirzokov iumirzokov 24 июн 22 10:45 monthly  
-r-xr--r--. 1 iumirzokov iumirzokov 0 июн 22 10:53 my_os  
drwx--x--x. 1 iumirzokov iumirzokov 0 июн 22 10:53 play  
drwxr-xr-x. 1 iumirzokov iumirzokov 14 июн 22 10:48 reports  
drwxr-xr-x. 1 iumirzokov iumirzokov 28 июн 22 10:53 ski.plases  
drwxr-xr-x. 1 iumirzokov iumirzokov 10 июн 21 21:20 work  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 Видео  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 Документы  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 Загрузки  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 Изображения  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 Музыка  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 Общедоступные  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 'Рабочий стол'  
drwxr-xr-x. 1 iumirzokov iumirzokov 0 июн 21 21:08 Шаблоны  
iumirzokov@iumirzokov:~$
```

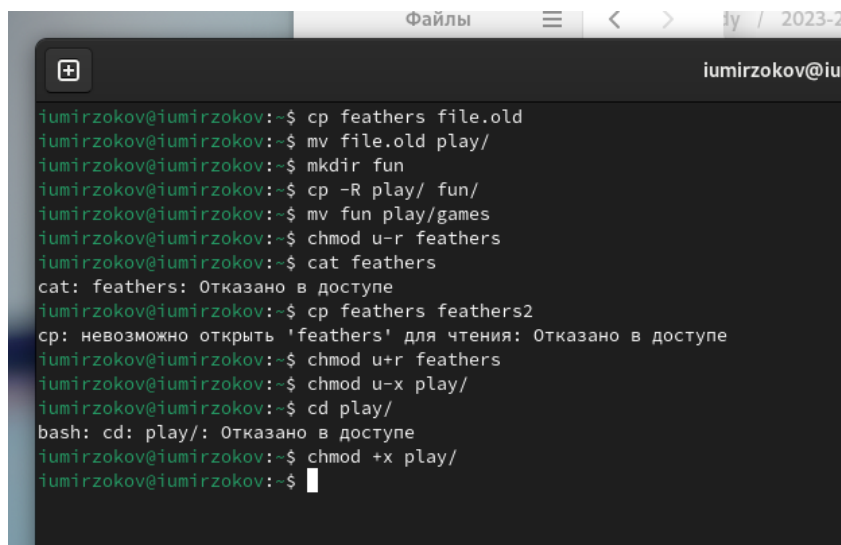
Figure 2.5: Настройка прав доступа

4.1. Просмотрим содержимое файла /etc/passwd.

```
Файлы
iumirzokov@iumirzokov:~
итого 1484
drwxr-xr-x. 1 root root 4750 июн 21 21:06 ./
dr-xr-xr-x. 1 root root 158 мар 8 19:41 ../
drwxr-xr-x. 1 root root 126 ноя 1 2023 abrt/
-rw-r--r--. 1 root root 16 фев 8 15:51 adjtime
-rw-r--r--. 1 root root 1529 июл 25 2023 aliases
drwxr-xr-x. 1 root root 70 янв 29 03:00 alsa/
drwxr-xr-x. 1 root root 1584 апр 5 10:26 alternatives/
drwxr-xr-x. 1 root root 56 ноя 1 2023 anaconda/
-rw-r--r--. 1 root root 269 июл 19 2023 anthy-unicode.conf
-rw-r--r--. 1 root root 833 фев 10 2023 appstream.conf
-rw-r--r--. 1 root root 55 янв 29 03:00 asound.conf
drwxr-x--. 1 root root 108 янв 24 03:00 audit/
drwxr-xr-x. 1 root root 232 фев 8 15:51 authselect/
drwxr-xr-x. 1 root root 66 ноя 1 2023 avahi/
drwxr-xr-x. 1 root root 84 ноя 1 2023 bash_completion.d/
-rw-r--r--. 1 root root 2638 июл 25 2023 bashrc
-rw-r--r--. 1 root root 535 янв 5 03:00 bindresvport.blacklist
drwxr-xr-x. 1 root root 0 мар 1 03:00 binfmt.d/
dr-xr-xr-x. 1 root root 62 мар 8 19:44 bluetooth/
-rw-r-----. 1 root brlapi 33 ноя 1 2023 brlapi.key
drwxr-xr-x. 1 root root 76 дек 20 2023 brltty/
-rw-r--r--. 1 root root 30583 дек 20 2023 brltty.conf
drwxr-xr-x. 1 root root 0 янв 10 03:00 ceph/
drwxr-xr-x. 1 root root 0 янв 17 03:00 chkconfig.d/
drwxr-xr-x. 1 root root 44 ноя 1 2023 chromium/
-rw-r--r--. 1 root root 1372 дек 5 2023 chrony.conf
drwxr-xr-x. 1 root root 24 янв 12 03:00 cifs-utils/
drwxr-xr-x. 1 root root 168 янв 29 03:00 containers/
drwx-----. 1 root root 0 мар 1 03:00 credstore/
drwx-----. 1 root root 0 мар 1 03:00 credstore.encrypted/
drwxr-xr-x. 1 root root 70 дек 4 2023 crypto-policies/
-rw-----. 1 root root 0 фев 8 15:49 crypttab
-rw-r--r--. 1 root root 1401 июл 25 2023 csh.cshrc
-rw-r--r--. 1 root root 1544 июл 25 2023 csh.login
drwxr-xr-x. 1 root lp 396 июн 22 10:23 cups/
/etc
```

Figure 2.6: Файл /etc/passwd

4.2 - 4.12. Выполним все указанные действия по перемещению файлов и каталогов



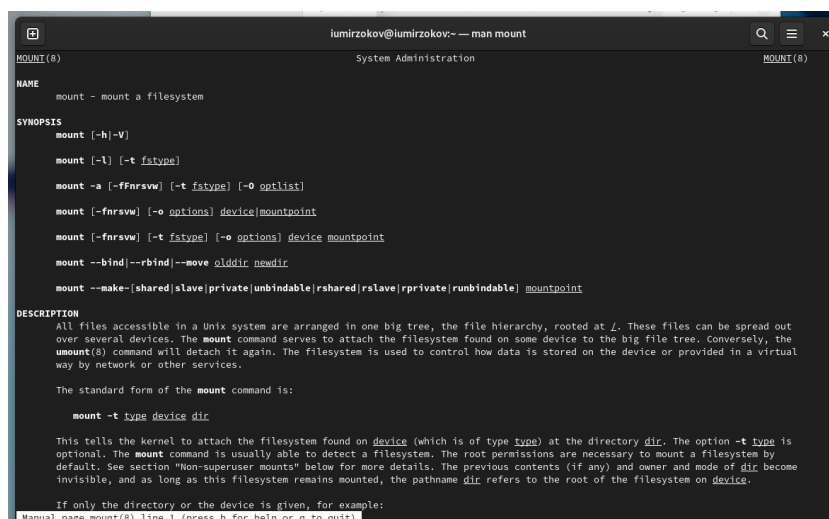
```
Файлы  2023-2
iumirzokov@iu
iumirzokov@iumirzokov:~$ cp feathers file.old
iumirzokov@iumirzokov:~$ mv file.old play/
iumirzokov@iumirzokov:~$ mkdir fun
iumirzokov@iumirzokov:~$ cp -R play/ fun/
iumirzokov@iumirzokov:~$ mv fun play/games
iumirzokov@iumirzokov:~$ chmod u-r feathers
iumirzokov@iumirzokov:~$ cat feathers
cat: feathers: Отказано в доступе
iumirzokov@iumirzokov:~$ cp feathers feathers2
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
iumirzokov@iumirzokov:~$ chmod u+r feathers
iumirzokov@iumirzokov:~$ chmod u-x play/
iumirzokov@iumirzokov:~$ cd play/
bash: cd: play/: Отказано в доступе
iumirzokov@iumirzokov:~$ chmod +x play/
iumirzokov@iumirzokov:~$
```

Figure 2.7: Работа с файлами и правами доступа

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

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

5. Прочитаем `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуем, приведя примеры.



```
iumirzokov@iumirzokov:~$ man mount
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(s) 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
    This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. The option -t type is
    optional. The mount command is usually able to detect a filesystem. The root permissions are necessary to mount a filesystem by
    default. See section "Non-superuser mounts" below for more details. The previous contents (if any) and owner and mode of dir become
    invisible, and as long as this filesystem remains mounted, the pathname dir refers to the root of the filesystem on device.
    If only the directory or the device is given, for example:
    Manual page mount(8) line 1 (press h for help or q to quit)
```

Figure 2.8: Команда mount

Монтирование файловой системы к общему дереву каталогов. Для размонтирования используется команда `umount`.

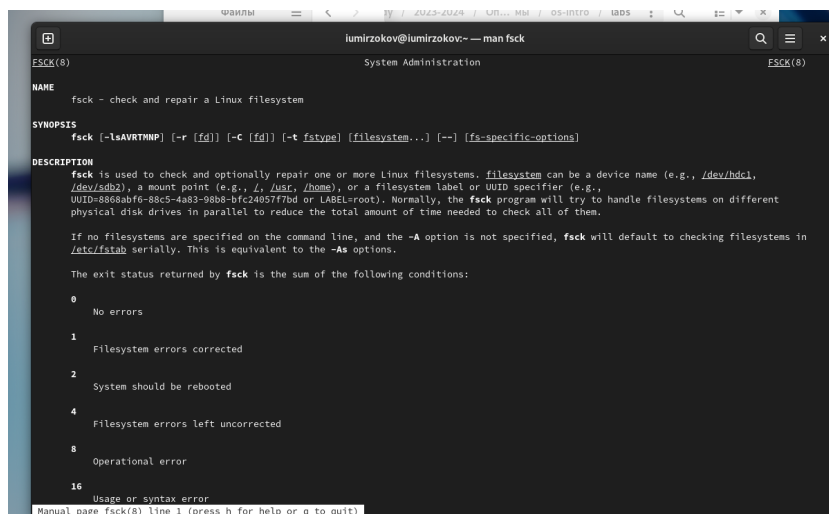


Figure 2.9: Команда `fsck`

`fsck` (проверка файловой системы) – это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Она использует программы, специфичные для типа файловой системы, которую она проверяет. Вы можете использовать команду `fsck` для восстановления поврежденных файловых систем в ситуациях, когда система не загружается или раздел не может быть смонтирован.

```
mkfs(8)
System Administration
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.
-V, --verbose
Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option more than once
inhibits execution of any filesystem-specific commands. This is really only useful for testing.
-h, --help
Display help text and exit.
-V, --version
Manual page mkfs(8) line 1 (press h for help or q to quit)
```

Figure 2.10: Команда mkfs

Буквы в mkfs значке означают “make file system” (создать файловую систему). Команда обычно используется для управления устройствами хранения в Linux. Вы можете рассматривать mkfs как инструмент командной строки для форматирования диска в определенной файловой системе.

```
kill(1)
User Commands
NAME
kill - terminate a process
SYNOPSIS
kill [-signal|--signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...
kill -l [number] | -L
DESCRIPTION
The command kill sends the specified signal to the specified processes or process groups.
If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal
should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to
perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent,
then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the
opportunity to perform any clean-up before terminating.
Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid,
and --queue options, and the possibility to specify processes by command name, are local extensions.
If signal is 0, then no actual signal is sent, but error checking is still performed.
ARGUMENTS
The list of processes to be signaled can be a mixture of names and PIDs.
pid
Each pid can be expressed in one of the following ways:
n
where n is larger than 0. The process with PID n is signaled.
0
All processes in the current process group are signaled.
Manual page kill(1) line 1 (press h for help or q to quit)
```

Figure 2.11: Команда kill

Системный вызов kill может быть использован для посылки какого-либо сигнала какому-либо процессу или группе процесса.

3 Вывод

В ходе данной работы мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Научились совершать базовые операции с файлами, управлять правами их доступа для пользователя и групп. Ознакомились с Анализом файловой системы. А также получили базовые навыки по проверке использования диска и обслуживанию файловой системы.

4 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

Ответ: 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. Ответ:

- 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. В inode 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». Второй – 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? Ответ: 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 одно-

временно нажать клавиши Ctrl и d.

4. Назовите основные причины нарушения целостности файловой системы.

Как устранить повреждения файловой системы? Ответ: 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? Ответ: 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. Делает-

ся 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. Ответ: 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 вперёд 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. Ответ: 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. Ответ: Команды mv и mvdir 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? Ответ: 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