

Отчёт по лабораторной работе №7

Операционные системы

Андреева Софья Владимировна

Содержание

1	Цель работы	4
2	Выполнение работы	5
3	Контрольные вопросы	10
4	Выводы	18
	Список литературы	19

Список иллюстраций

2.1	Выполняем примеры, приведённые в первой части описания лабораторной работы	5
2.2	права файла ~/тау	6
2.3	Создадим и переместим каталог	7
2.4	Определим опции команды chmod	7
2.5	Выполняем все примеры	8
2.6	Прочитали man по командам mount, fsck, mkfs, kill	9
3.1	Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера	10

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 работы

Выполняем все примеры, приведённые в первой части описания лабораторной работы. Скопируем файл ~/abc1 в файл april и в файл may.г. Скопировать файлы april и may в каталог monthly одной командой. Скопируем файл monthly/may в файл с именем june. Скопируем каталог monthly в каталог monthly.00. Скопируем каталог monthly.00 в каталог /tmp. Проверяем все проведенные команды. (рис. fig. 2.1).

```
[svandreeva@fedora ~]$ touch abc1
[svandreeva@fedora ~]$ cp abc1 april
[svandreeva@fedora ~]$ cp abc1 may
[svandreeva@fedora ~]$ mkdir monthly
[svandreeva@fedora ~]$ cp april may monthly
[svandreeva@fedora ~]$ ls
abc1  bin  git-extended  may  passw.txt  work  Документы  Изображения
april  Downloads  LICENSE  monthly  project  Видео  Загрузки  Музыка
[svandreeva@fedora ~]$ cd ~/monthly
[svandreeva@fedora monthly]$ ls
april  may
[svandreeva@fedora monthly]$ cd
[svandreeva@fedora ~]$ cp monthly/may monthly/june
[svandreeva@fedora ~]$ ls monthly
april  june  may
[svandreeva@fedora ~]$ mkdir monthly.00
[svandreeva@fedora ~]$ cp -r monthly monthly.00
[svandreeva@fedora ~]$ ls monthly.00
monthly
[svandreeva@fedora ~]$ cp -r monthly.00 /tmp
[svandreeva@fedora ~]$ ls /tmp
monthly
```

Рис. 2.1: Выполняем примеры, приведённые в первой части описания лабораторной работы

Изменим название файла april на july в домашнем каталоге. Переместим файл july в каталог monthly.00. Переименуем каталог monthly.00 в monthly.01.г. Переместим каталог monthly.01 в каталог reports. Переименуем каталог reports/monthly.01 в reports/monthly. Проверяем результаты проведенных команд. (рис. fig. ??).

[Выполняем примеры, приведённые в первой части описания лабораторной работы]](image/2.jpg){#fig:002 width=70%}

Создаем файл ~/may с правом выполнения для владельца. Лишаем владельца файла ~/may права на выполнение. Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Требуется создать файл ~/abc1 с правом записи для членов группы. (рис. fig. 2.2).

```
[svandreeva@fedora ~]$ touch may
[svandreeva@fedora ~]$ ls -l may
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:45 may
[svandreeva@fedora ~]$ chmod u+x may
[svandreeva@fedora ~]$ ls -l may
-rwxr--r--. 1 svandreeva svandreeva 0 мар 18 14:45 may
[svandreeva@fedora ~]$ chmod u-x may
[svandreeva@fedora ~]$ ls -l may
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:45 may
[svandreeva@fedora ~]$ ls -l monthly
итого 0
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:35 april
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:36 june
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:35 may
[svandreeva@fedora ~]$ chmod g-r monthly
[svandreeva@fedora ~]$ chmod o-r monthly
[svandreeva@fedora ~]$ ls -l monthly
итого 0
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:35 april
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:36 june
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 14:35 may
[svandreeva@fedora ~]$ chmod g+w abc1
[svandreeva@fedora ~]$ ls -l abc1
-rw-rw-r--. 1 svandreeva svandreeva 0 мар 18 14:34 abc1
[svandreeva@fedora ~]$
```

Рис. 2.2: права файла ~/may

Скопируем файл /usr/include/sys/io.h в домашний каталог и назовем его equipment. В домашнем каталоге создадим директорию ~/ski.places, переместим туда файл equipment. Переименуем файл ~/ski.places/equipment в ~/ski.places/equiplist. Файл abc1 скопируем в каталог ~/ski.places, назовем его equiplist2. Создадим каталог с именем equipment в каталоге ~/ski.places, переместим туда файлы ~/ski.places/equiplist и equiplist2. Создадим и переместим каталог ~/newdir в каталог ~/ski.places и назовем его plans (рис. fig. 2.3).

```
[svandreeva@fedora ~]$ cp /usr/include/sys/io.h ~/equipment
[svandreeva@fedora ~]$ ls
abc1  Downloads  git-extended  may  passwd.txt  reports  Видео  Загрузки  Мусор
bin  equipment  LICENSE  monthly  Project  work  Документы  Изображения  Обычные файлы
[svandreeva@fedora ~]$ mkdir ~/ski.places
[svandreeva@fedora ~]$ mv equipment ski.places/
[svandreeva@fedora ~]$ ls ~/ski.places
equipment
[svandreeva@fedora ~]$ mv ~/ski.places/equipment ~/ski.places/equiplist
[svandreeva@fedora ~]$ ls ~/ski.places
equiplist
[svandreeva@fedora ~]$ cp abc1 ~/ski.places/equiplist2
cp: невозможно создать обычный файл '/home/svandreeva/ski.places/equiplist2': Нет такого файла или каталога
[svandreeva@fedora ~]$ cp abc1 ~/ski.places/equiplist2
[svandreeva@fedora ~]$ ls ~/ski.places
equiplist  equiplist2
[svandreeva@fedora ~]$ mkdir ski.places/equipment
[svandreeva@fedora ~]$ mv ~/ski.places/equiplist ~/ski.places/equiplist2/ski.places/equipment
[svandreeva@fedora ~]$ ls ~/ski.places/equipment
equiplist  equiplist2
[svandreeva@fedora ~]$ mkdir ~/newdir
[svandreeva@fedora ~]$ mv ~/newdir ~/ski.places/plans
[svandreeva@fedora ~]$ ls ~/ski.places/
equipment  plans
```

Рис. 2.3: Создадим и переместим каталог

Определим опции команды `chmod`, необходимые для того, чтобы присвоить предварительно созданным файлам заданные нам права доступа. (рис. fig. 2.4).

```
[svandreeva@fedora LAB7TEST]$ chmod g-x australia
[svandreeva@fedora LAB7TEST]$ chmod o-x australia
[svandreeva@fedora LAB7TEST]$ chmod g-r play
[svandreeva@fedora LAB7TEST]$ chmod o-r play
[svandreeva@fedora LAB7TEST]$ chmod u-w my_os
[svandreeva@fedora LAB7TEST]$ chmod u+x my_os
[svandreeva@fedora LAB7TEST]$ chmod g+w feathers
[svandreeva@fedora LAB7TEST]$ ls -l
total 0
drwxr--r--. 1 svandreeva svandreeva 0 map 18 15:23 australia
-rw-rw-r--. 1 svandreeva svandreeva 0 map 18 15:19 feathers
-r-xr--r--. 1 svandreeva svandreeva 0 map 18 15:19 my_os
drwx--x--x. 1 svandreeva svandreeva 0 map 18 15:23 play
[svandreeva@fedora LAB7TEST]$
```

Рис. 2.4: Определим опции команды `chmod`

Выполняем все примеры (рис. fig. 2.5): - Просмотрели содержимое файла `/etc/passwd` с помощью команды `cat`. - Скопируем файл `~/feathers` в файл `~/file.old`. Переместите файл `~/file.old` в каталог `~/play`. Скопируем каталог `~/play` в каталог `~/fun`. Переместим каталог `~/fun` в каталог `~/play` и назовем его `games`. - Лишим владельца файла `~/feathers` права на чтение. Попытавшись просмотреть файл `~/feathers` командой `cat`, мы получили отказ в доступе. Попытавшись

скопировать файл ~/feathers, мы получили отказ в доступе. Дадим владельцу файла ~/feathers право на чтение. - Лишим владельца каталога ~/play права на выполнение. Перейдя в каталог ~/play, получили отказ в доступе, вся информация скрыта вопросительными знаками. Дадим владельцу каталога ~/play право на выполнение.

```
[svandreeva@fedora ~]$ cp ~/LAB7TEST/feathers ~/file.old
[svandreeva@fedora ~]$ mv ~/file.old play
[svandreeva@fedora ~]$ cp -r play fun
[svandreeva@fedora ~]$ mv fun play/games
mv: не удалось выполнить stat для 'play/games': Это не каталог
[svandreeva@fedora ~]$ mv fun ~/LAB7TEST/play/games
[svandreeva@fedora ~]$ ls ~/LAB7TEST/play/games
/home/svandreeva/LAB7TEST/play/games
[svandreeva@fedora ~]$ cd ~/LAB7TEST/play/games
bash: cd: /home/svandreeva/LAB7TEST/play/games: Это не каталог
[svandreeva@fedora ~]$ ls
bin          LAB7TEST    passwd.txt  reports     Видео       Изв
Downloads    LICENSE    play        ski.plases  Документы   Муз
git-extended monthly    project     work        Загрузки     Обс
[svandreeva@fedora ~]$ ls ~/LAB7TEST/play/
games
[svandreeva@fedora ~]$ ls ~/file.old
ls: невозможно получить доступ к '/home/svandreeva/file.old': Нет
[svandreeva@fedora ~]$ cd ~/LAB7TEST
[svandreeva@fedora LAB7TEST]$ ls
australia  feathers  my_os  play
[svandreeva@fedora LAB7TEST]$ chmod u-r feathers
[svandreeva@fedora LAB7TEST]$ cat feathers
cat: feathers: Отказано в доступе
[svandreeva@fedora LAB7TEST]$ chmod u+r feathers
[svandreeva@fedora LAB7TEST]$ chmod u-x play
[svandreeva@fedora LAB7TEST]$ ls -l play
ls: невозможно получить доступ к 'play/games': Отказано в доступе
итого 0
-????????? ? ? ? ?      ? games
[svandreeva@fedora LAB7TEST]$ chmod u+x play
[svandreeva@fedora LAB7TEST]$ ls -l play
итого 0
-rw-r--r--. 1 svandreeva svandreeva 0 мар 18 15:45 games
[svandreeva@fedora LAB7TEST]$
```

Рис. 2.5: Выполняем все примеры

Прочитали ман по командам mount, fsck, mkfs, kill (рис. fig. 2.6).


```
KILL(1) User Commands KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s 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.
```

Рис. 2.6: Прочитали man по командам mount, fsck, mkfs, kill

3 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу (рис. fig. 3.1).

Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера

Рис. 3.1: Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера

- `proc`. Файловая система `proc` является важным источником информации о вашей Linux-системе, который попросту нельзя игнорировать. Вообще, `proc` является псевдо- или виртуальной файловой системой, которая предоставляет пользователям доступ к внутренним структурам ядра Linux. Другими словами, `proc` не является реальной файловой системой в обычном смысле; она располагается исключительно в оперативной памяти, а не на диске. При этом она автоматически монтируется системой.
- `Sysfs` - отправляет данные в пространство пользователя с помощью виртуальных файлов. Эти данные содержат данные о различных подсистемах ядра, аппаратных устройствах и связанных с ними драйверах устройств.
- `tmpfs` и `devtmpfs` - они относятся к энергозависимой памяти.
- `devpts` - обеспечивает доступ к терминалам `pseudo` (PTY).

- `cgroup2` - Неверно ведущий себя процесс может создавать тонны процессов через ветвления, запуская некую бомбу ветвлений и сокрушая своё ядро. Это означает, что нам требуется ввести некий способ контроля ресурсов для процессов в пределах заданного пространства имён. Это достигается через механизм, носящий название групп контроля (`control groups`), обычно именуемых `cgroups`. `cgroups` работают под понятием контроллеров `cgroup` и представляются в файловой системе с названием `cgroupfs` в самом ядре Linux. В настоящее время применяется `cgroup v2` версия `cgroups`.
- `pstore` - был введен в Linux для записи информации (например, `dmesg tail`) при выключении. `Pstore` не зависит от `kdump` и может запускаться до него. В определенных сценариях (например, хосты/гости с корневыми файловыми системами на NFS/iSCSI, где произошел сбой сетевого программного и/или аппаратного обеспечения), `pstore` может содержать информацию, доступную для посмертной отладки, которая не может быть получена иным образом.
- `bpf` - это псевдо-файловая система, существующая только в памяти, которая позволяет создавать файлы, ссылающиеся на объекты BPF.
- `btrfs` - файловая система для Linux, основанная на структурах B-деревьев и работающая по принципу «копирование при записи» (`copy-on-write`). Опубликована корпорацией Oracle в 2007 году под лицензией GNU General Public License.
- `selinux` - Как и файловая система `/proc`, `/selinux` является псевдофайловой системой. Новая реализация SE Linux использует расширенные атрибуты для хранения контекста безопасности.
- `tracefs` - файловая система для задач трассировки Linux
- `debugfs` - `DebugFS` является самой известной утилитой, предназначенной для работы с файловыми системами `EXT2FS` и `EXT3FS`.

- `hugetlb` - использует страницы большого размера, что позволяет кэшировать больше адресов за раз.
- `mqueue` - обеспечивает необходимую поддержку ядра для библиотеки пользовательского пространства, которая реализует интерфейсы очереди сообщений POSIX.
- `fusectl` - это простой интерфейс для программ пользовательского пространства для экспорта виртуальной файловой системы в ядро Linux.
- `configfs` - Виртуальная файловая система, представляющая состояние ядра операционной системы и запущенных процессов в виде файлов.
- `ext4` - журналируемая файловая система, используемая преимущественно в операционных системах с ядром Linux, созданная на базе `ext3` в 2006 году.
- `fuse.gv`, `fuse.po`, `fuse.je` - FUSE (файловая система в пользовательском пространстве) — это интерфейс для программ пользовательского пространства для экспорта файловой системы в ядро Linux.

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

`/` — root каталог. Содержит в себе всю иерархию системы;

`/bin` — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: `pwd`, `ls`, `cat`, `ps`);

`/boot` — тут расположены файлы, используемые для загрузки системы (образ `initrd`, ядро `vmlinuz`);

`/dev` — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;

/etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;

/home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;

/lib — содержит системные библиотеки, с которыми работают программы и модули ядра;

/lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;

/media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;

/mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;

/opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);

/proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;

/root — директория, которая содержит файлы и личные настройки суперпользователя;

/run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;

/sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;

/srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);

`/sys` — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;

`/tmp` — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;

`/usr` — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме `root`). Каталог имеет вторичную иерархию и похож на корневой;

`/var` — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в `/var/log`, кэш в `/var/cache`, очереди заданий в `/var/spool/` и так далее.

3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?

Монтирование тома.

4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?

Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

1. Один блок адресуется несколькими `inode` (принадлежит нескольким файлам).
2. Блок помечен как свободный, но в то же время занят (на него ссылается `inode`).
3. Блок помечен как занятый, но в то же время свободен (ни один `inode` на него не ссылается).
4. Неправильное число ссылок в `inode` (недостаток или избыток ссылающихся записей в файловой системе).

5. Несовпадение между размером файла и суммарным размером адресуемых inode блоков
6. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы)
7. "Потерянные" файлы (правильные inode, на которые не ссылаются записи каталогов)
8. Недопустимые или неразмещенные номера inode в записях каталогов.

5. Как создаётся файловая система?

- `mkfs` - позволяет создать файловую систему Linux.

Создать файловую систему linux, семейства ext, на устройстве можно с помощью команды `mkfs`. Ее синтаксис выглядит следующим образом:

```
sudo mkfs -t тип устройства
```

Доступны дополнительные параметры:

- `c` - проверить устройство на наличие битых секторов
- `b` - размер блока файловой системы
- `j` - использовать журналирование для ext3
- `L` - задать метку раздела
- `v` - показать подробную информацию о процессе работы
- `V` - версия программы

6. Дайте характеристику командам для просмотра текстовых файлов.

- `cat <имя_файла>` - Это самая простая и, пожалуй, самая популярная команда для просмотра файла в Linux. Cat просто печатает содержимое файла на стандартном экране, т.е. на экране. В основном используется для небольших файлов.
- `Less <имя_файла>`. Команда Less просматривает файл по одной странице за раз.

- `Head <имя_файла>`. Команда `Head` — это еще один способ просмотра текстового файла, но с небольшой разницей. Команда `head` отображает первые 10 строк текстового файла по умолчанию. Вы можете изменить это поведение, используя опции с командой `head`, но основной принцип остается тем же: команда `head` начинает работать с заголовка (начала) файла.
- `Tail <имя_файла>`. Команда `Tail` в Linux аналогична и все же противоположна команде `head`. В то время как команда `head` отображает файл с начала, команда `tail` отображает файл с конца. По умолчанию команда `tail` отображает последние 10 строк файла. Команды `Head` и `Tail` могут быть объединены для отображения выбранных строк из файла. Вы также можете использовать команду `tail` для просмотра изменений, внесенных в файл в режиме реального времени.

7. Приведите основные возможности команды `cp` в Linux.

Это сокращение от `copy`, и она делает именно то, что предполагает ее название: она копирует. `cp` используется для копирования файлов из одного местоположения в другое. `cp` также можно использовать для копирования всех каталогов в новое место. Можно использовать эту команду для копирования нескольких файлов и каталогов.

8. Приведите основные возможности команды `mv` в Linux.

Команда `mv` используется для перемещения файлов из одного каталога в другой. Также команда `mv` используется для переименования файла в системах Linux.

9. Что такое права доступа? Как они могут быть изменены?

Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами.

Каждый файл можно изменять по трём параметрам доступа. Вот они:

- Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;
- Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;
- Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Чтобы получить доступ к файлам в Linux, используются разрешения. Эти разрешения назначаются трем объектам: файлу, группе и другому объекту (то есть всем остальным). Изменить права доступа можно при помощи команды **chmod**:

```
chmod <параметры изменения> <имя_файла/каталога>
```

4 Выводы

В ходе выполнения этой лабораторной 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

1. Руководство к выполнению лабораторной работы №5
2. Статья “Chmod (777, 755, 444) — настройка прав доступа к файлам и папкам”