

Отчет по лабораторной работе №7

Операционные системы

Шихалиева Зурият Арсеновна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	17
6	Ответы на контрольные вопросы	18

Список иллюстраций

4.1	Копирование файла	12
4.2	Создание директории	13

Список таблиц

1 Цель работы

Целью данной лабораторной работы является 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 Задание

1. Выполнить все примеры из лабораторной работы
2. Выполнить команды по копированию, созданию и перемещению файлов и каталогов
3. Определить опции команды `chmod`
4. Изменить права доступа к файлам
5. Прочитать документацию о командах `mount`, `fsck`, `mkfs`, `kill`

3 Теоретическое введение

Для создания текстового файла можно использовать команду `touch`. Для просмотра файлов небольшого размера можно использовать команду `cat`. Для просмотра файлов постранично удобнее использовать команду `less`. Команда `cp` используется для копирования файлов и каталогов. Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов.

Каждый файл или каталог имеет права доступа. В сведениях о файле или каталоге указываются:

- тип файла (символ (-) обозначает файл, а символ (d) — каталог);
- права для владельца файла (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует);
- права для членов группы (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует);
- права для всех остальных (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует).

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

Файловая система в Linux состоит из файлов и каталогов. Каждому физическому носителю соответствует своя файловая система. Существует несколько типов файловых систем. Перечислим наиболее часто встречающиеся типы:

- ext2fs (second extended filesystem);
- ext3fs (third extended file system);

- ext4 (fourth extended file system);
- ReiserFS;
- xfs;
- fat (file allocation table);
- ntfs (new technology file system).

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров.

4 Выполнение лабораторной работы

Создаю файл, дважды копирую его с новыми именами и проверяю, что все команды были выполнены корректно (рис. 1).

```
zashikhalieva@zashikhalieva:~$ touch abc1
zashikhalieva@zashikhalieva:~$ cp abc1 april
zashikhalieva@zashikhalieva:~$ cp abc1 may
zashikhalieva@zashikhalieva:~$ ls
abc1      file.txt      may          repo
april     git-extended  newdir       repository
```

{#fig:001

width=70%, рис.1}

Создаю директорию, копирую в нее два файла, созданных на прошлом этапе, проверяю, что все скопировалось (рис. 2).

```
zashikhalieva@zashikhalieva:~$ mkdir monthly
zashikhalieva@zashikhalieva:~$ cp april may monthly
zashikhalieva@zashikhalieva:~$ ls /monthly
ls: невозможно получить доступ к '/monthly': Нет так
zashikhalieva@zashikhalieva:~$ ls monthly
april  may
```

{#fig:002

width=70%, рис. 2}

Копирую файл, находящийся не в текущей директории в файл с новым именем тоже не текущей директории (рис. 3).

```
zashikhalieva@zashikhalieva:~$ cp monthly/may monthly/june
zashikhalieva@zashikhalieva:~$ ls monthly
april  june  may
```

{#fig:003

width=70%, рис.3}

Создаю новую директорию. Копирую предыдущую созданную директорию в новую созданную. Затем копирую предыдущую созданную директорию вместе со всем содержимым в каталог /tmp (рис. 4).

```
zashikhalieva@zashikhalieva:~$ mkdir monthly.00
zashikhalieva@zashikhalieva:~$ cp -r monthly monthly.00
zashikhalieva@zashikhalieva:~$ ls monthly.00
monthly
zashikhalieva@zashikhalieva:~$ cp -r monthly.00 /tmp
```

{#fig:004

width=70% рис.4}

Переименовываю файл, затем перемещаю его в каталог (рис. 5).

```
zashikhalieva@zashikhalieva:~$ mv april july
zashikhalieva@zashikhalieva:~$ mv july monthly.00/
zashikhalieva@zashikhalieva:~$ ls monthly.00
july  monthly
```

{#fig:005

width=70% рис. 5}

Переименовываю monthly.00 в monthly.01, создаю новую директорию, перемещаю директорию в директорию reports, переименовываю эту директорию, убираю из названия 01 (рис. 6).

```
zashikhalieva@zashikhalieva:~$ mv monthly.00 monthly.01
zashikhalieva@zashikhalieva:~$ mkdir reports
zashikhalieva@zashikhalieva:~$ mv monthly.01 reports
zashikhalieva@zashikhalieva:~$ mv reports/monthly.01 reports/monthly
```

{#fig:006

width=70% рис. 6}

Создаю пустой файл, проверяю права доступа у него, изменяю права доступа, добавляя пользователю (создателю) можно выполнять файл (рис. 7).

```

zashikhalieva@zashikhalieva:~$ touch may
zashikhalieva@zashikhalieva:~$ ls -l may
-rw-r--r--. 1 zashikhalieva zashikhalieva 0 сен  3 10:26 may
zashikhalieva@zashikhalieva:~$ chmod u+x may
zashikhalieva@zashikhalieva:~$ ls -l may
-rwxr--r--. 1 zashikhalieva zashikhalieva 0 сен  3 10:26 may
zashikhalieva@zashikhalieva:~$ chmod u-x may
zashikhalieva@zashikhalieva:~$ ls -l may
-rw-r--r--. 1 zashikhalieva zashikhalieva 0 сен  3 10:26 may

```

{#fig:007

width=70% рис.7}

Меняю права доступа у директории: группы и остальные пользователи не смогут ее прочесть (рис. 8).

```

zashikhalieva@zashikhalieva:~$ chmod g-r monthly
zashikhalieva@zashikhalieva:~$ chmod o-r monthly

```

{#fig:008

width=70% рис. 8}

Создаю новый пустой файл, даю ему права доступа: группы могут в этом файле писать содержимое (рис. 9).

```

zashikhalieva@zashikhalieva:~$ chmod g+w abc1
zashikhalieva@zashikhalieva:~$ ls -l abc1
-rw-rw-r--. 1 zashikhalieva zashikhalieva 0 сен  3 10:40 abc1

```

{#fig:009

width=70% рис.9}

Проверяю файловую систему (рис. 10).

```

zashikhalieva@zashikhalieva:~$ sudo fsck /dev/sda1
[sudo] пароль для zashikhalieva:
fsck from util-linux 2.40.1
e2fsck 1.47.0 (5-Feb-2023)
ext2fs_open2: Bad magic number in super-block
fsck.ext2: Superblock invalid, trying backup blocks...
fsck.ext2: Bad magic number in super-block while trying to open /dev/sda1

The superblock could not be read or does not describe a valid ext2/ext3/ext4
filesystem. If the device is valid and it really contains an ext2/ext3/ext4
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>
or
    e2fsck -b 32768 <device>

```

{#fig:010}

width=70% рис. 10}

Копирую файл в домашний каталог с новым именем, создаю новую пустую директорию, перемещаю файл в эту директорию, переименовываю файл (рис. 11).

```

zashikhalieva@zashikhalieva:~$ cp /usr/include/sys/io.h ~/equipment
zashikhalieva@zashikhalieva:~$ cp /usr/include/sys/io.h equipment
zashikhalieva@zashikhalieva:~$ ls
abc1      file.txt  may       newdir    reports   work      Документы
equipment git-extended monthly   repo      repository Видео     Загрузки
zashikhalieva@zashikhalieva:~$ mkdir ~/ski.plases
zashikhalieva@zashikhalieva:~$ mv equipment ~/ski.plases
zashikhalieva@zashikhalieva:~$ ls
abc1      git-extended monthly   repo      repository work      Документы
file.txt  may         newdir    reports   ski.plases Видео     Загрузки
zashikhalieva@zashikhalieva:~$ mv ~/ski.plases/equipment ~/ski.plases/equiplist
zashikhalieva@zashikhalieva:~$ ls ~/ski.plases/
equiplist

```

Рис. 4.1: Копирование файла

Создаю новый файл, копирую его в новую директорию, но уже сразу с новым именем. Создаю внутри этого каталога подкаталог, перемещаю файлы в подкаталог (рис. 12).

```

zashikhalieva@zashikhalieva:~$ touch abc1
zashikhalieva@zashikhalieva:~$ cp abc1 ~/ski.plases/equiplist2
zashikhalieva@zashikhalieva:~$ mkdir ~/ski.plases/equipment
zashikhalieva@zashikhalieva:~$ mv ~/ski.plases/equiplist2 ~/ski.plases/equipment/
zashikhalieva@zashikhalieva:~$ ls ~/ski.plases/
equipment
zashikhalieva@zashikhalieva:~$ ls ~/ski.plases/equipment/
equiplist equiplist2

```

{#fig:012}

width=70% рис. 12}

Создаю новую директорию, в этой же строчке перемещаю ее с новым именем в директорию, созданную в прошлый раз (рис. fig. 4.2).

```
zashikhalieva@zashikhalieva:~$ rmdir newdir
zashikhalieva@zashikhalieva:~$ mkdir newdir ; mv newdir ~/ski.places/plans
```

Рис. 4.2: Создание директории

Проверяю, какие права нужно поменять и как, чтобы у новой директории были нужные по заданию права (рис. 14).

```
zashikhalieva@zashikhalieva:~$ mkdir australia
zashikhalieva@zashikhalieva:~$ chmod u+x australia
zashikhalieva@zashikhalieva:~$ ls -l
итого 0
-rw-rw-r--. 1 zashikhalieva zashikhalieva 0 сен 3 10:52 abcl
drwxr-xr-x. 1 zashikhalieva zashikhalieva 0 сен 3 11:01 australia
```

width=70% рис. 14}

Проверяю, какие права нужно поменять и как, чтобы у новых файлов были нужные по заданию права (рис. 15).

```
zashikhalieva@zashikhalieva:~$ chmod o-r+x play
zashikhalieva@zashikhalieva:~$ ls -l
итого 0
-rw-rw-r--. 1 zashikhalieva zashikhalieva 0 сен 3 10:52 abcl
drwxr-xr-x. 1 zashikhalieva zashikhalieva 0 сен 3 11:01 australia
-rw-r--r--. 1 zashikhalieva zashikhalieva 0 сен 2 20:52 file.txt
drwxr-xr-x. 1 zashikhalieva zashikhalieva 90 сен 2 11:11 git-extended
-rw-r--r--. 1 zashikhalieva zashikhalieva 0 сен 3 10:26 may
drwx--x--x. 1 zashikhalieva zashikhalieva 24 сен 3 09:57 monthly
drwx--x--x. 1 zashikhalieva zashikhalieva 0 сен 3 11:05 play
drwxr-xr-x. 1 zashikhalieva zashikhalieva 0 сен 3 11:06
```

width=70% рис. 15}

Создаю файл, добавляю в правах доступа право но исполнение и убираю право на запись для владельца, затем создаю следующий файл, ему в правах доступа добавляю право на запись для группы (рис. 16).

```

zashikhalieva@zashikhalieva:~$ touch my_os
zashikhalieva@zashikhalieva:~$ chmod u+x-w my_os
zashikhalieva@zashikhalieva:~$ ls -l m
may      monthly/ my_os
zashikhalieva@zashikhalieva:~$ ls -l my_os
-r-xr--r--. 1 zashikhalieva zashikhalieva 0 сен  3 12:13 my_os
zashikhalieva@zashikhalieva:~$ touch feathers
zashikhalieva@zashikhalieva:~$ chmod g+w feathers
zashikhalieva@zashikhalieva:~$ ls -l feathers
-rw-rw-r--. 1 zashikhalieva zashikhalieva 0 сен  3 12:14 feathers

```

{#fig:016

width=70% рис. 16}

Читаю содержимое файла (рис. 17).

```

zashikhalieva@zashikhalieva:~$ cat /etc/passwd
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/sbin:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/sbin/nologin
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:/var/spool/mail:/usr/sbin/nologin

```

{#fig:017

width=70%, рис. 17}

Копирую файл с новым именем, перемещаю его в ранее созданную директорию, рекурсивно ее копирую с новым именем, рекурсивно копирую в нее скопированную до этого папку (рис. 18).

```

zashikhalieva@zashikhalieva:~$ cp ~/feathers ~/files.old
zashikhalieva@zashikhalieva:~$ mv ~/files.old ~/play/
zashikhalieva@zashikhalieva:~$ cp -r ~/play/ ~/fun/
zashikhalieva@zashikhalieva:~$ cp -r ~/fun/ ~/play/games
zashikhalieva@zashikhalieva:~$ ls plat
ls: невозможно получить доступ к 'plat': Нет такого файла
каталога
zashikhalieva@zashikhalieva:~$ ls play
files.old  games

```

{#fig:018}

width=70% рис.18}

Убираю право на чтение у файла для создателя, поэтому не могу его прочесть, также не могу его скопировать, потому что отказано в доступе на чтение, возвращаю все права (рис. 19).

```

zashikhalieva@zashikhalieva:~$ chmod u-r feathers
zashikhalieva@zashikhalieva:~$ cat feathers
cat: feathers: Отказано в доступе
zashikhalieva@zashikhalieva:~$ cp feathers feathers1
cp: невозможно открыть 'feathers' для чтения: Отказано в досту
пе
zashikhalieva@zashikhalieva:~$ chmod u+r feathers

```

{#fig:019}

width=70% рис. 19}

Убираю у директории право на исполнение для пользователя, пытаюсь в нее войти - отказано в доступе, возвращаю все права (рис. 20).

```

zashikhalieva@zashikhalieva:~$ chmod u-x play
zashikhalieva@zashikhalieva:~$ cd ~/play
bash: cd: /home/zashikhalieva/play: Отказано в доступе
zashikhalieva@zashikhalieva:~$ chmod u+x play

```

{#fig:020}

width=70% рис. 20}

Я прочитала описание каждой из четырех команд с помощью man (рис. 21). - mount — утилита командной строки в UNIX-подобных операционных системах. Применяется для монтирования файловых систем. Пример: mount /dev/sda1 - fsck (проверка файловой системы) - это утилита командной строки,

которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет. - 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. Пример: mkfs ext4. - Команда Kill посылает указанный сигнал указанному процессу. Если не указано ни одного сигнала, посылается сигнал SIGTERM. Сигнал SIGTERM завершает лишь те процессы, которые не обрабатывают его приход. Для других процессов может быть необходимым послать сигнал SIGKILL, поскольку этот сигнал перехватить невозможно. kill 279.

```
zashikhalieva@zashikhalieva:~/play$ man mount  
zashikhalieva@zashikhalieva:~/play$ man fsck  
zashikhalieva@zashikhalieva:~/play$ man mkfs  
zashikhalieva@zashikhalieva:~/play$ man kill
```

{#fig:021

width=70%, рис. 21}

5 Выводы

При выполнении данной лабораторной работы 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

6 Ответы на контрольные вопросы

1. Ext2, Ext3, Ext4 или Extended Filesystem - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзбайта. Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера на лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

2. Общая структура файловой системы и её характеристика:

/ — 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 суперполь-

зователя;

/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. Чтобы содержимое некоторой файловой системы было доступно операционной системе нужно выполнить монтирование тома.
4. Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

Один блок адресуется несколькими mode (принадлежит нескольким файлам). Блок помечен как свободный, но в то же время занят (на него ссылается onode). Блок помечен как занятый, но в то же время свободен (ни один inode на него не

ссылается). Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах). Несовпадение между размером файла и суммарным размером адресуемых inode блоков. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы). “Потерянные” файлы (правильные inode, на которые не ссылаются записи каталогов). Недопустимые или неразмещенные номера inode в записях каталогов.

5. `mkfs` - позволяет создать файловую систему Linux.
6. Характеристика командам для просмотра текстовых файлов: `Cat` - выводит содержимое файла на стандартное устройство вывода. Выполнение команды `head` выведет первые 10 строк текстового файла. Выполнение команды `tail` выведет последние 10 строк текстового файла. Команда `tac` - это тоже самое, что и `cat`, только отображает строки в обратном порядке. Для того, чтобы просмотреть огромный текстовый файл применяются команды для постраничного просмотра. Такие как `more` и `less`.
7. Основные возможности: `cp` – копирует или перемещает директорию, файлы.
8. Основные возможности: `mv` - переименовать или переместить файл или директорию
9. Что такое права доступа? Как они могут быть изменены? Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Может быть право на чтение, выполнение, и запись.