

# Отчёт по лабораторной работе №7

## операционные системы

Ведьмина Александра Сергеевна

### Содержание

1	Цель работы .....	1
2	Теоретическое введение .....	1
3	Выполнение лабораторной работы .....	2
4	Выводы.....	8
5	Ответы на контрольные вопросы .....	8

## 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 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`.

Каждый файл или каталог имеет права доступа (табл. 5.1). 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.

### 3 Выполнение лабораторной работы

Выполняю все примеры из первой части работы.

```
asvedjmina@fedora:~$ cd
asvedjmina@fedora:~$ touch abc1
asvedjmina@fedora:~$ cp abc1 april
asvedjmina@fedora:~$ cp abc1 may
asvedjmina@fedora:~$ mkdir monthly
asvedjmina@fedora:~$ cp april may monthly
asvedjmina@fedora:~$ cp monthly/may monthly/june
asvedjmina@fedora:~$ ls monthly
april  june  may
asvedjmina@fedora:~$
```

Рис. 1: Выполнение примеров - 1

```
asvedjmina@fedora:~$ mkdir monthly.00
asvedjmina@fedora:~$ cp -r monthly monthly.00
asvedjmina@fedora:~$ cp -r monthly.00 /tmp
asvedjmina@fedora:~$
```

Рис. 2: Выполнение примеров - 2

```
asvedjmina@fedora:~$ cd
asvedjmina@fedora:~$ mv april july
asvedjmina@fedora:~$ mv july monthly.00
asvedjmina@fedora:~$ ls monthly.00
july  monthly
asvedjmina@fedora:~$
```

Рис. 3: Выполнение примеров - 3

```
asvedjmina@fedora:~$ touch may
asvedjmina@fedora:~$ ls -l may
-rw-r--r--. 1 asvedjmina asvedjmina 0 Mar 19 20:00 may
asvedjmina@fedora:~$ chmod u+x may
asvedjmina@fedora:~$ ls -l may
-rwxr--r--. 1 asvedjmina asvedjmina 0 Mar 19 20:00 may
asvedjmina@fedora:~$ chmod u-x may
asvedjmina@fedora:~$ ls -l may
-rw-r--r--. 1 asvedjmina asvedjmina 0 Mar 19 20:00 may
asvedjmina@fedora:~$
```

Рис. 4: Выполнение примеров - 4

```
asvedjmina@fedora:~$ chmod g-r monthly
asvedjmina@fedora:~$ chmod o-r monthly
asvedjmina@fedora:~$ touch abc1
asvedjmina@fedora:~$ chmod g+w abc1
asvedjmina@fedora:~$
```

Рис. 5: Выполнение примеров - 5

Скопирую файл /usr/include/sys/io.h в домашний каталог и называю его equipment. Создаю директорию ~/ski.places, перемещаю в неё файл equipment.

```
asvedjmina@fedora:~$ cp /usr/include/sys/io.h ~/equipment
asvedjmina@fedora:~$ mkdir ~/ski.places
asvedjmina@fedora:~$ mv ~/equipment ~/ski.places
asvedjmina@fedora:~$
```

Рис. 6: Выполнение пунктов 2.1-2.3

Переименовываю файл ~/ski.places/equipment в ~/ski.places/equiplist. Создаю в домашнем каталоге файл abc1 и скопируйте его в каталог ~/ski.places, называя equiplist2. Создаю каталог с именем equipment в каталоге ~/ski.places. Перемещаю файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment.

```
asvedjmina@fedora:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
asvedjmina@fedora:~$ touch ~/abc1
asvedjmina@fedora:~$ cp ~/abc1 ~/ski.places/equiplist2
asvedjmina@fedora:~$ mkdir ~/ski.places/equipment
asvedjmina@fedora:~$ mv ~/ski.places/equiplist ~/ski.places/equipment
asvedjmina@fedora:~$ mv ~/ski.places/equiplist2 ~/ski.places/equipment
asvedjmina@fedora:~$
```

Рис. 7: Выполнение пунктов 2.4-2.7

Создаю каталог ~/newdir и перемещаю его в ~/ski.places, называя plans.

```
asvedjmina@fedora:~$ mkdir ~/newdir
asvedjmina@fedora:~$ mv ~/newdir ~/ski.places/plans
asvedjmina@fedora:~$
```

Рис. 8: Выполнение пункта 2.8

Создаю файлы australia, play, my\_os, feathers и определяю опции команды chmod, которые нужны, чтобы присвоить этим файлам указанные права доступа.

```
asvedjmina@fedora:~$ mkdir australia
asvedjmina@fedora:~$ mkdir play
asvedjmina@fedora:~$ mkdir my_os
asvedjmina@fedora:~$ mkdir feathers
asvedjmina@fedora:~$ chmod 744 australia
asvedjmina@fedora:~$ chmod 711 play
asvedjmina@fedora:~$ chmod 554 my_os
asvedjmina@fedora:~$ chmod 664 feathers
asvedjmina@fedora:~$
```

Рис. 9: Выполнение пункта 3

Просматриваю содержимое файла /etc/passwd.

```

asvedjmina@fedora:~$ 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
operator:x:11:0:operator:/root:/usr/sbin/nologin
games:x:12:100:games:/usr/games:/usr/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/usr/sbin/nologin
dbus:x:81:81:System Message Bus:/:/usr/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/:/usr/sbin/nologin
systemd-coredump:x:998:998:systemd Core Dumper:/:/usr/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/usr/sbin/nologin
systemd-oom:x:997:997:systemd Userspace OOM Killer:/:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/:/usr/sbin/nologin

```

Рис. 10: Выполнение пункта 4.1

Скопирую файл ~/feathers в файл ~/file.old, перемещаю файл ~/file.old в каталог ~/play. Скопирую каталог ~/play в каталог ~/fun, перемещаю каталог ~/fun в каталог ~/play и называю его games.

```

asvedjmina@fedora:~$ cp -r ~/feathers ~/file.old
asvedjmina@fedora:~$ mv ~/file.old ~/play
asvedjmina@fedora:~$ cp -r ~/play ~/fun
asvedjmina@fedora:~$ mv ~/fun ~/play/games
asvedjmina@fedora:~$

```

Рис. 11: Выполнение пунктов 4.2-4.5

Лишаю владельца ~/feathers прав на чтение. Пытаюсь посмотреть этот файл, получаю ошибку, как и при попытке его копирования. Возвращаю права.

```

asvedjmina@fedora:~$ chmod u-r ~/feathers
asvedjmina@fedora:~$ cat ~/feathers
cat: /home/asvedjmina/feathers: Permission denied
asvedjmina@fedora:~$ chmod u+r ~/feathers
asvedjmina@fedora:~$

```

Рис. 12: Выполнение пунктов 4.6-4.9

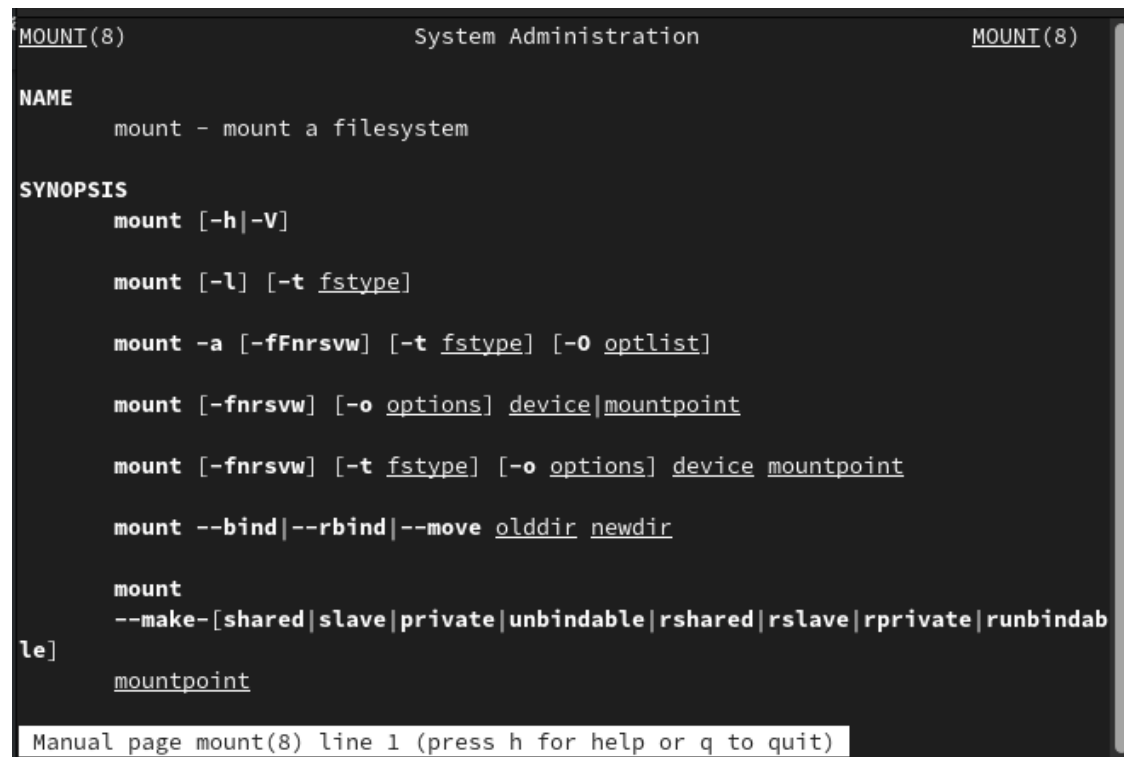
Лишаю владельца ~/play права на выполнение. При попытке перехода в этот каталог теперь получаю сообщение, что на это действие нет прав. Возвращаю права.

```
asvedjmina@fedora:~$ chmod u-x ~/play
asvedjmina@fedora:~$ cd ~/play
bash: cd: /home/asvedjmina/play: Permission denied
asvedjmina@fedora:~$ chmod u+x ~/play
asvedjmina@fedora:~$
```

Рис. 13: Выполнение пунктов 4.10-4.12

Читаю man по командам mount, fsck, mkfs, kill.

mount используется для подключения файловых систем к директории.

The image shows a terminal window displaying the man page for the 'mount' command. The window title is 'MOUNT(8) System Administration MOUNT(8)'. The content includes the command name 'mount - mount a filesystem', a synopsis section with various options like [-h|-V], [-l], [-t fstype], [-a], [-fnrsvw], [-o options], and [--bind|--rbind|--move], and a list of options for the --make- flag. The bottom of the window shows the footer 'Manual page mount(8) line 1 (press h for help or q to quit)'.

```
MOUNT(8)                                System Administration                                MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-fFnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount
    --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable]
mountpoint

Manual page mount(8) line 1 (press h for help or q to quit)
```

Рис. 14: Просмотр команды mount

fsck проверяет и исправляет целостность файловой системы.

```
FSCK(8)                                     System Administration                                     FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--]
    [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux
    filesystems. filesystem can be a device name (e.g., /dev/hdc1,
/dev/sdb2), a mount point (e.g., /, /usr, /home), or a filesystem label
    or UUID specifier (e.g., UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or
    LABEL=root). Normally, the fsck program will try to handle filesystems
    on different physical disk drives in parallel to reduce the total
    amount of time needed to check all of them.

    If no filesystems are specified on the command line, and the -A option
    is not specified, fsck will default to checking filesystems in
/etc/fstab serially. This is equivalent to the -As options.

    The exit status returned by fsck is the sum of the following

Manual page fsck(8) line 1 (press h for help or q to quit)
```

Рис. 15: Просмотр команды *fsck*

mkfs создаёт новую файловую систему на устройстве.

```
MKFS(8)                                     System Administration                                     MKFS(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [fs-options] device [size]

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific
    mkfs.<type> utils.

    mkfs is used to build a Linux filesystem on a device, usually a hard
    disk partition. The device argument is either the device name (e.g.,
/dev/hda1, /dev/sdb2), or a regular file that shall contain the
    filesystem. The size argument is the number of blocks to be used for
    the filesystem.

    The exit status returned by mkfs is 0 on success and 1 on failure.

    In actuality, mkfs is simply a front-end for the various filesystem
    builders (mkfs.fstype) available under Linux. The filesystem-specific
    builder is searched for via your PATH environment setting only. Please

Manual page mkfs(8) line 1 (press h for help or q to quit)
```

Рис. 16: Просмотр команды *mkfs*

kill завершает работу процессов.

```
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

Manual page kill(1) line 1 (press h for help or q to quit)
```

Рис. 17: Просмотр команды kill

## 4 Выводы

В ходе лабораторной работы я познакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов, а также приобрела практические навыки по применению команд для работы с файлами и каталогами.

## 5 Ответы на контрольные вопросы

1. Характеристика файловых систем на жёстком диске: ext4: Это распространенная файловая система в Linux, обеспечивающая хорошую производительность и надежность. Поддерживает файлы до 16 терабайт и объемы до 1 экзбайта.
2. swar: Файловая система swar используется для обмена данными между оперативной памятью и жестким диском, когда оперативная память заполнена. /home: Это директория, где хранятся пользовательские файлы и настройки. Общая структура файловой системы и директории первого уровня: /: Корневая директория, содержащая все остальные директории и файлы. /bin: Хранит основные исполняемые файлы. /etc: Содержит конфигурационные файлы. /home: Директория пользователей. /var: Хранит переменные данные, такие как журналы и временные файлы.



3. Доступность файловой системы операционной системе: Для доступности файловой системы операционной системе необходимо её смонтировать, что позволит системе видеть и работать с содержимым этой файловой системы.
4. Причины нарушения целостности файловой системы и способы восстановления: Причины: сбой в питании, отключение системы без корректного выключения, ошибки ввода-вывода. Восстановление: использование инструментов проверки целостности файловой системы, таких как fsck в Linux.
5. Создание файловой системы: Файловая система создается с помощью утилиты mkfs, которая форматирует устройство для использования определенной файловой системой.
6. Просмотр текстовых файлов: Для просмотра текстовых файлов в Linux используются команды cat, less, more, head, tail.
7. Основные возможности команды cp в Linux: Команда cp используется для копирования файлов и директорий в Linux. Основные возможности включают копирование файлов, сохранение атрибутов, рекурсивное копирование директорий.
8. Основные возможности команды mv в Linux: Команда mv используется для перемещения файлов и директорий в Linux. Основные возможности включают перемещение файлов, переименование файлов, перемещение с сохранением атрибутов.
9. Права доступа и их изменение: Права доступа определяют, кто может читать, записывать или выполнять файлы. Изменить их можно с помощью команды chmod в Linux, где можно установить разрешения для владельца, группы и остальных пользователей.