

Отчёт по лабораторной работе №5

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

Леденев Егор Олегович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	13

Список иллюстраций

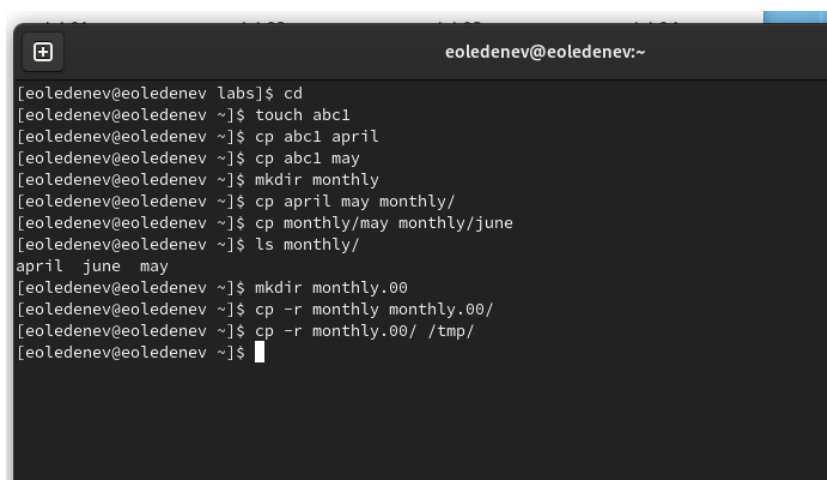
2.1	Выполнение примеров	5
2.2	Выполнение примеров	5
2.3	Выполнение примеров	6
2.4	Работа с каталогами	6
2.5	Настройка прав доступа	7
2.6	Файл /etc/passwd	8
2.7	Работа с файлами и правами доступа	8
2.8	Команда mount	9
2.9	Команда fsck	10
2.10	Команда mkfs	11
2.11	Команда kill	11

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.



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

Рис. 2.1: Выполнение примеров



```
[eoledenev@eoledenev ~]$  
[eoledenev@eoledenev ~]$ cd  
[eoledenev@eoledenev ~]$ mv april july  
[eoledenev@eoledenev ~]$ mv july monthly.00/  
[eoledenev@eoledenev ~]$ ls monthly.00/  
july  monthly  
[eoledenev@eoledenev ~]$ mv monthly.00/ monthly.01  
[eoledenev@eoledenev ~]$ mkdir reports  
[eoledenev@eoledenev ~]$ mv monthly.01/ reports/  
[eoledenev@eoledenev ~]$ mv reports/monthly.01/ reports/monthly  
[eoledenev@eoledenev ~]$
```

Рис. 2.2: Выполнение примеров

```

[eoledenev@eoledenev ~]$ cd
[eoledenev@eoledenev ~]$ touch may
[eoledenev@eoledenev ~]$ ls -l may
-rw-r--r--. 1 eoledenev eoledenev 0 апр 12 11:38 may
[eoledenev@eoledenev ~]$ chmod u+x may
[eoledenev@eoledenev ~]$ ls -l may
-rwxr--r--. 1 eoledenev eoledenev 0 апр 12 11:38 may
[eoledenev@eoledenev ~]$ chmod u-x may
[eoledenev@eoledenev ~]$ ls -l may
-rw-r--r--. 1 eoledenev eoledenev 0 апр 12 11:38 may
[eoledenev@eoledenev ~]$ cd
[eoledenev@eoledenev ~]$ mkdir monthly/
mkdir: невозможно создать каталог «monthly/»: Файл существует
[eoledenev@eoledenev ~]$ chmod g-r,o-r monthly/
[eoledenev@eoledenev ~]$ cd
[eoledenev@eoledenev ~]$ touch abc1
[eoledenev@eoledenev ~]$ chmod g+w abc1
[eoledenev@eoledenev ~]$

```

Рис. 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`.

```

[eoledenev@eoledenev ~]$
[eoledenev@eoledenev ~]$
[eoledenev@eoledenev ~]$ cp /usr/include/linux/sysinfo.h ~
[eoledenev@eoledenev ~]$ mv sysinfo.h equipment
[eoledenev@eoledenev ~]$ mkdir ski.places
[eoledenev@eoledenev ~]$ mv equipment ski.places/
[eoledenev@eoledenev ~]$ mv ski.places/equipment ski.places/equiplist
[eoledenev@eoledenev ~]$ touch abc1
[eoledenev@eoledenev ~]$ cp abc1 ski.places/equiplist2
[eoledenev@eoledenev ~]$ cd ski.places/
[eoledenev@eoledenev ski.places]$ mkdir equipment
[eoledenev@eoledenev ski.places]$ mv equiplist equipment/
[eoledenev@eoledenev ski.places]$ mv equiplist2 equipment/
[eoledenev@eoledenev ski.places]$ cd
[eoledenev@eoledenev ~]$ mkdir newdir
[eoledenev@eoledenev ~]$ mv newdir/ ski.places/
[eoledenev@eoledenev ~]$ mv ski.places/newdir/ ski.places/plans
[eoledenev@eoledenev ~]$

```

Рис. 2.4: Работа с каталогами

3. Определим опции команды `chmod`, необходимые для того, чтобы присвоить

файлам из хода работы нужные права доступа.

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

```
[eolenedev@eolenedev ~]$ mkdir australia
[eolenedev@eolenedev ~]$ mkdir play
[eolenedev@eolenedev ~]$ touch my_os feathers
[eolenedev@eolenedev ~]$ chmod 744 australia/
[eolenedev@eolenedev ~]$ chmod 711 play
[eolenedev@eolenedev ~]$ chmod 544 my_os
[eolenedev@eolenedev ~]$ chmod 664 feathers
[eolenedev@eolenedev ~]$ ls -l
итого 0
-rw-rw-r--. 1 eolenedev eolenedev 0 авг 12 11:41 abc1
drwxr--r--. 1 eolenedev eolenedev 0 авг 12 11:43 australia
-rw-rw-r--. 1 eolenedev eolenedev 0 авг 12 11:44 feathers
-rw-r--r--. 1 eolenedev eolenedev 0 авг 12 11:38 may
drwx--x--x. 1 eolenedev eolenedev 24 авг 12 11:35 monthly
-r-xr--r--. 1 eolenedev eolenedev 0 авг 12 11:44 my_os
drwx--x--x. 1 eolenedev eolenedev 0 авг 12 11:43 play
drwxr-xr-x. 1 eolenedev eolenedev 14 авг 12 11:38 reports
drwxr-xr-x. 1 eolenedev eolenedev 28 авг 12 11:43 ski.places
drwxr-xr-x. 1 eolenedev eolenedev 10 авг 11 13:04 work
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 Видео
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 Документы
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 Загрузки
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 Изображения
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 Музыка
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 Общедоступные
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 'Рабочий стол'
drwxr-xr-x. 1 eolenedev eolenedev 0 авг 11 12:04 Шаблоны
[eolenedev@eolenedev ~]$
```

Рис. 2.5: Настройка прав доступа

4.1. Просмотрим содержимое файла /etc/passwd.

```
eoledenev@eoledenev:~ — less /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/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:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
systemd-oom:x:999:999:systemd Userspace OOM Killer:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/usr/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
polkitd:x:998:997:User for polkitd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
unbound:x:997:995:Unbound DNS resolver:/etc/unbound:/sbin/nologin
nm-openconnect:x:996:994:NetworkManager user for OpenConnect:/sbin/nologin
geoclue:x:995:993:User for geoclue:/var/lib/geoclue:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/sbin/nologin
gluster:x:994:992:GlusterFS daemons:/run/gluster:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
chrony:x:993:990:/var/lib/chrony:/sbin/nologin
sasauthd:x:992:76:Sasauthd user:/run/sasauthd:/sbin/nologin
/etc/passwd
```

Рис. 2.6: Файл /etc/passwd

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

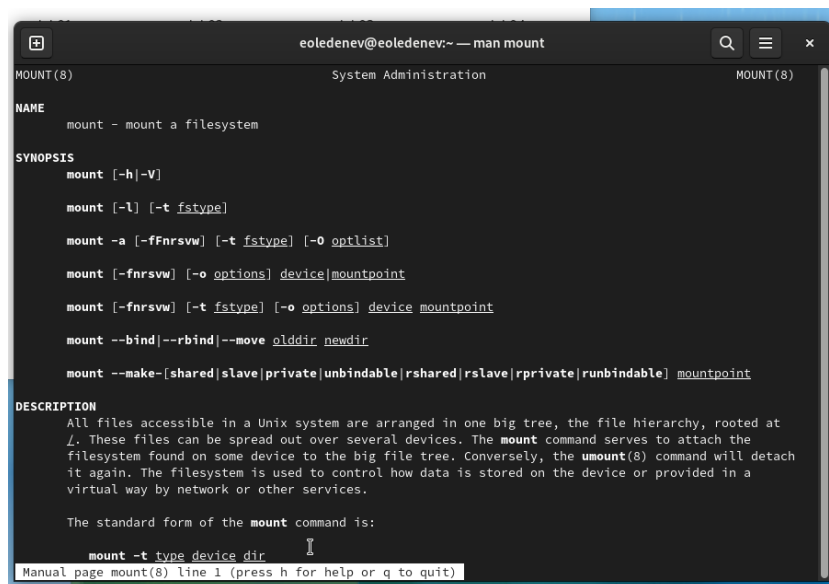
```
[eoledenev@eoledenev ~]$
[eoledenev@eoledenev ~]$ cp feathers file.old
[eoledenev@eoledenev ~]$ mv file.old play/
[eoledenev@eoledenev ~]$ mkdir fun
[eoledenev@eoledenev ~]$ cp -R play/ fun/
[eoledenev@eoledenev ~]$ mv fun/ play/games
[eoledenev@eoledenev ~]$ chmod u-r feathers
[eoledenev@eoledenev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

Рис. 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 ман по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуем, приведя примеры.



```
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

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(8) 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.

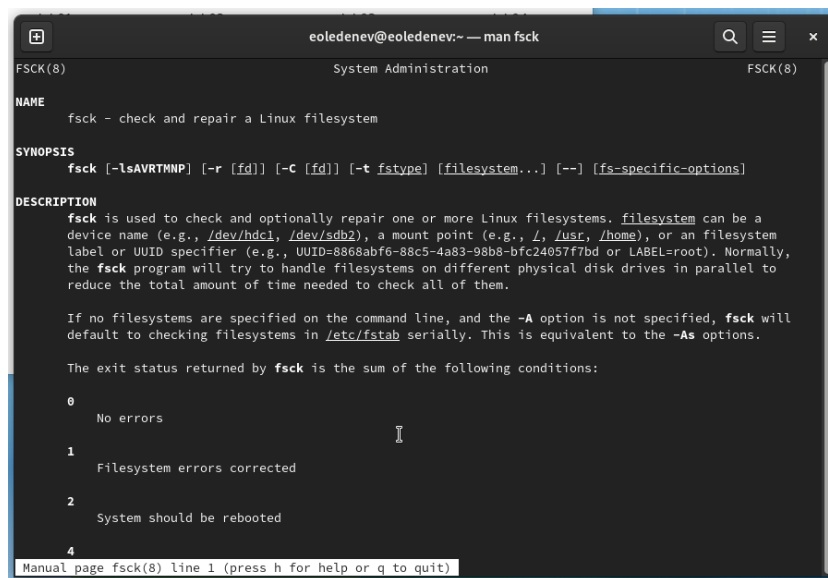
    The standard form of the mount command is:

    mount -t type device dir

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

Рис. 2.8: Команда `mount`

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

A screenshot of a terminal window titled "eolenedev@eolenedev:~ — man fsck". The window displays the manual page for the `fsck` command. The content is as follows:

```
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 an 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 conditions:

    0      No errors
    1      Filesystem errors corrected
    2      System should be rebooted
    4

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

Рис. 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                                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 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

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

```

Рис. 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                                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

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

```

Рис. 2.11: Команда kill

Системный вызов `kill` может быть использован для отправки 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.