

Лабораторная работа №5. Анализ файловой системы Linux.

2 May, 2022 Moscow, Russia

Лабораторная работа №5. Анализ файловой системы Linux.

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.

Задание

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 1. Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него.
 2. В домашнем каталоге создайте директорию `~/ski.places`.
 3. Переместите файл `equipment` в каталог `~/ski.places`.
 4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
 6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 8. Создайте и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`.
3. Определите опции команды **chmod**, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
 1. `drwxr-r- ... australia`
 2. `drwx-x-x ... play`

3. -r-xr-r- ... my_os
4. -rw-rw-r- ... feathers

При необходимости создайте нужные файлы.

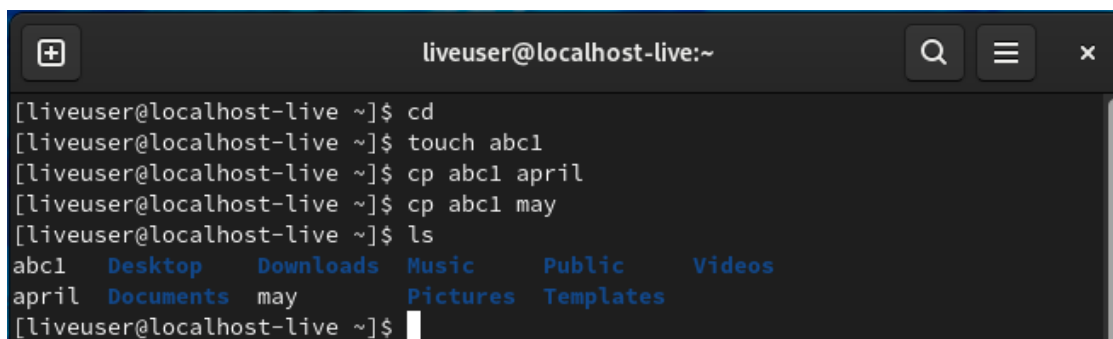
4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 1. Просмотрите содержимое файла/etc/password.
 2. Скопируйте файл~/feathersв файл~/file.old.
 3. Переместите файл~/file.oldв каталог~/play.
 4. Скопируйте каталог~/playв каталог~/fun.
 5. Переместите каталог~/funв каталог~/playи назовите егоgames.
 6. Лишите владельца файла~/feathersправа на чтение. 7.Что произойдёт,если вы попытаетесь просмотреть файл~/feathersкомандой cat?
 7. Что произойдёт,если вы попытаетесь скопировать файл~/feathers?
 8. Дайте владельцу файла~/feathersправо на чтение.
 9. Лишите владельца каталога~/playправа на выполнение.
 10. Перейдите в каталог~/play.Что произошло?
 11. Дайте владельцу каталога~/playправо на выполнение.
5. Прочитайте ман по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры.

Выполнение лабораторной работы

Шаг 1 - Выполните все примеры, приведённые в первой части описания лабораторной работы.

Пример 1

Копирование файла в текущем каталоге. Скопировать файл ~/abc1 в файл april и в файл may:



```
liveuser@localhost-live:~  
[liveuser@localhost-live ~]$ cd  
[liveuser@localhost-live ~]$ touch abc1  
[liveuser@localhost-live ~]$ cp abc1 april  
[liveuser@localhost-live ~]$ cp abc1 may  
[liveuser@localhost-live ~]$ ls  
abc1  Desktop  Downloads  Music      Public    Videos  
april Documents  may        Pictures   Templates  
[liveuser@localhost-live ~]$
```

Пример 1

Пример 2

Копирование нескольких файлов в каталог. Скопировать файлы april и may в каталог montly:

```
[liveuser@localhost-live ~]$ mkdir montly
[liveuser@localhost-live ~]$ cp april may montly
[liveuser@localhost-live ~]$ ls
abcl Desktop Downloads montly Pictures Templates
april Documents may Music Public Videos
[liveuser@localhost-live ~]$ cd montly
[liveuser@localhost-live montly]$ ls
april may
```

Пример 2

Пример 3

Копирование файлов в произвольном каталоге. Скопировать файл montly/may в файл с именем june:

```
[liveuser@localhost-live ~]$ cp montly/may montly/june
[liveuser@localhost-live ~]$ cd montly
[liveuser@localhost-live montly]$ ls
april june may
```

Пример 3

Пример 4

Копирование каталогов в текущем каталоге. Скопировать каталог montly в каталог montly.00:

```
[liveuser@localhost-live ~]$ mkdir montly.00
[liveuser@localhost-live ~]$ cp -r montly montly.00
[liveuser@localhost-live ~]$ ls
abcl Desktop Downloads montly Music Public Videos
april Documents may montly.00 Pictures Templates
```

Пример 4

Пример 5

Копирование каталогов в произвольном каталоге. Скопировать каталог montly.00 в каталог /tmp

```

[liveuser@localhost-live ~]$ cp -r montly.00 /tmp
[liveuser@localhost-live ~]$ tree
.
├── abc1
├── april
├── Desktop
├── Documents
├── Downloads
├── may
├── montly
│   ├── april
│   ├── june
│   └── may
├── montly.00
│   └── montly
│       ├── april
│       ├── june
│       └── may
├── Music
├── Pictures
├── Public
├── Templates
└── Videos

11 directories, 9 files
[liveuser@localhost-live ~]$ cd /tmp
[liveuser@localhost-live tmp]$ ls
montly.00
systemd-private-e481671f4ead42138f9f4ca4d5d44368-chronyd.service-mTrebh
systemd-private-e481671f4ead42138f9f4ca4d5d44368-colord.service-NbYuwG

```

Пример 5

Пример 6

ереименование файлов в текущем каталоге. Изменить название файла april на july в домашнем каталоге:

```

[liveuser@localhost-live tmp]$ cd
[liveuser@localhost-live ~]$ mv april july
[liveuser@localhost-live ~]$ ls
abc1      Documents  july  montly  Music  Public  Videos
Desktop  Downloads  may  montly.00  Pictures  Templates
[liveuser@localhost-live ~]$

```

Пример 6

Пример 7

Перемещение файлов в другой каталог. Переместить файл july в каталог montly.00:

```

[liveuser@localhost-live ~]$ mv july montly.00
[liveuser@localhost-live ~]$ ls montly.00
july  montly

```

Пример 7

Пример 8

Переименование каталогов в текущем каталоге. Переименовать каталог monthly.00 в monthly.01

```
[liveuser@localhost-live ~]$ mv montly.00 montly.01
[liveuser@localhost-live ~]$ ls
abcl  Documents  may  montly.01  Pictures  Templates
Desktop  Downloads  montly  Music  Public  Videos
[liveuser@localhost-live ~]$
```

Пример 8

Пример 9

Перемещение каталога в другой каталог. Переместить каталог monthly.01 в каталог reports:

```
[liveuser@localhost-live ~]$ mkdir reports
[liveuser@localhost-live ~]$ mv montly.01 reports
[liveuser@localhost-live ~]$ ls
abcl  Documents  may  Music  Public  Templates
Desktop  Downloads  montly  Pictures  reports  Videos
[liveuser@localhost-live ~]$ ls reports
montly.01
[liveuser@localhost-live ~]$
```

Пример 9

Пример 10

Переименование каталога, не являющегося текущим. Переименовать каталог reports/monthly.01 в reports/monthly:

```
[liveuser@localhost-live ~]$ mv reports/monthly.01 reports/monthly
[liveuser@localhost-live ~]$ ls reports
montly
[liveuser@localhost-live ~]$
```

Пример 10

Пример 11

Требуется создать файл ~/may с правом выполнения для владельца:

```
[liveuser@localhost-live ~]$ cd
[liveuser@localhost-live ~]$ touch may
[liveuser@localhost-live ~]$ ls -l may
-rw-rw-r--. 1 liveuser liveuser 0 May  2 09:38 may
[liveuser@localhost-live ~]$ chmod u+x may
[liveuser@localhost-live ~]$ ls -l may
-rwxrw-r--. 1 liveuser liveuser 0 May  2 09:38 may
[liveuser@localhost-live ~]$
```

Пример 11

Пример 12

Требуется лишить владельца файла ~/may права на выполнение:

```
[liveuser@localhost-live ~]$ chmod u-x may
[liveuser@localhost-live ~]$ ls -l may
-rw-rw-r--. 1 liveuser liveuser 0 May  2 09:38 may
[liveuser@localhost-live ~]$
```

Пример 12

Пример 13

Требуется создать каталог `monthly` с запретом на чтение для членов группы и всех остальных пользователей:

```
[liveuser@localhost-live ~]$ mkdir monthly
[liveuser@localhost-live ~]$ chmod g-r, o-r monthly
chmod: invalid mode: 'g-r,'
Try 'chmod --help' for more information.
```

Пример 13

Пример 14

Требуется создать файл `~/abc1` с правом записи для членов группы:

```
[liveuser@localhost-live ~]$ cd
[liveuser@localhost-live ~]$ touch abc1
[liveuser@localhost-live ~]$ chmod g+w abc1
[liveuser@localhost-live ~]$ ls -l abc1
-rw-rw-r--. 1 liveuser liveuser 0 May  2 09:45 abc1
[liveuser@localhost-live ~]$
```

Пример 14

Пример 15

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров. В результате её применения можно получить примерно следующее

```
[liveuser@localhost-live ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=1970628k,nr_inodes=492657,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=801656k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/sr0 on /run/initramfs/live type iso9660 (ro,relatime,nojoliet,check=s,map=n,blocksize=2048,iocharset=utf8)
/dev/mapper/live-rw on / type ext4 (rw,relatime,seclabel)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=30,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18275)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,size=2004140k,nr_inodes=409600,inode64)
vartmp on /var/tmp type tmpfs (rw,relatime,seclabel,inode64)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=400824k,nr_inodes=100206,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sr1 on /run/media/liveuser/VBox_GAs_6.1.34 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,iocharset=utf8,uhelper=udisks2)
[liveuser@localhost-live ~]$
```

Пример 15

Пример 16

Другой способ определения смонтированных в операционной системе файловых систем — просмотр файла `/etc/fstab`. Сделать это можно например с помощью команды `cat`:

```
[liveuser@localhost-live ~]$ cat /etc/fstab
vartmp /var/tmp tmpfs defaults 0 0
[liveuser@localhost-live ~]$
```

Пример 16

Пример 17

Для определения объёма свободного пространства на файловой системе можно воспользоваться командой `df`, которая выведет на экран список всех файловых систем в соответствии с именами устройств, с указанием размера и точки монтирования. Например:

```
[liveuser@localhost-live ~]$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
devtmpfs         1970628        0   1970628   0% /dev
tmpfs            2004136        0   2004136   0% /dev/shm
tmpfs            801656     9552    792104   2% /run
/dev/sr0         1962240 1962240        0 100% /run/initramfs/live
/dev/mapper/live-rw 7640588 5501212   2122992  73% /
tmpfs            2004140        52   2004088   1% /tmp
vartmp           2004136        0   2004136   0% /var/tmp
tmpfs            400824       136    400688   1% /run/user/1000
/dev/sr1          60096     60096        0 100% /run/media/liveuser/VBox_GAs_6.1.34
[liveuser@localhost-live ~]$
```

Пример 17

Пример 18

С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы:

```
[liveuser@localhost-live ~]$ fsck /dev/sda1
fsck from util-linux 2.37.2
e2fsck 1.46.3 (27-Jul-2021)
fsck.ext2: No such file or directory while trying to open /dev/sda1
Possibly non-existent device?
[liveuser@localhost-live ~]$
```

Пример 18

Шаг 2 - Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:

Шаг 2.1

Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него.

Для копирования файла я воспользовался командой `cp`, после чего убедился в успешности операции с помощью команд `ls` и `cat`.


```

[liveuser@localhost-live ~]$ cp /usr/include/sys/io.h ~/equipment
[liveuser@localhost-live ~]$ ls
abcl    Documents  equipment  monthly  Music    Public  Templates
Desktop Downloads  may       montly   Pictures  reports  Videos
[liveuser@localhost-live ~]$ cat equipment
/* Copyright (C) 1996-2021 Free Software Foundation, Inc.
   This file is part of the GNU C Library.

   The GNU C Library is free software; you can redistribute it and/or
   modify it under the terms of the GNU Lesser General Public
   License as published by the Free Software Foundation; either
   version 2.1 of the License, or (at your option) any later version.

   The GNU C Library is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   Lesser General Public License for more details.

   You should have received a copy of the GNU Lesser General Public
   License along with the GNU C Library; if not, see
   <https://www.gnu.org/licenses/>.  */

#ifndef _SYS_IO_H
#define _SYS_IO_H      1

#include <features.h>

__BEGIN_DECLS

/* If TURN_ON is TRUE, request for permission to do direct i/o on the
   port numbers in the range [FROM, FROM+NUM-1].  Otherwise, turn I/O
   permission off for that range.  This call requires root privileges.

   Portability note: not all Linux platforms support this call.  Most
   platforms based on the PC I/O architecture probably will, however.
   E.g., Linux/Alpha for Alpha PCs supports this.  */
extern int ioperm (unsigned long int __from, unsigned long int __num,
                  int __turn_on) __THROW;

/* Set the I/O privilege level to LEVEL.  If LEVEL>3, permission to
   access any I/O port is granted.  This call requires root

```

Задание 2.1

Шаг 2.2

В домашнем каталоге создайте директорию ~/ski.plases.

Я создал нужную директорию с помощью команды mkdir и убедился в результате с помощью команды ls.

```

[liveuser@localhost-live ~]$ mkdir ski.plases
[liveuser@localhost-live ~]$ ls
abcl    Documents  equipment  monthly  Music    Public  ski.plases  Videos
Desktop Downloads  may       montly   Pictures  reports  Templates
[liveuser@localhost-live ~]$

```

Задание 2.2

Шаг 2.3

Переместите файл equipment в каталог ~/ski.plases.

Для перемещения файла я воспользовался командой `mv`, а затем проверил результат с помощью `ls`.

```
[liveuser@localhost-live ~]$ mv equipment ski.places
[liveuser@localhost-live ~]$ ls
abc1      Documents  may        montly  Pictures  reports   Templates
Desktop  Downloads  monthly    Music   Public    ski.places Videos
[liveuser@localhost-live ~]$ ls ski.places
equipment
[liveuser@localhost-live ~]$
```

Задание 2.3

Шаг 2.4

Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.

Для переименования файла я воспользовался командой `mv`, а затем проверил результат с помощью `ls`.

```
[liveuser@localhost-live ~]$ mv ski.places/equipment ski.places/equiplist
[liveuser@localhost-live ~]$ ls ski.places
equiplist
[liveuser@localhost-live ~]$
```

Задание 2.4

Шаг 2.5

Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.

С помощью команды `touch` я создал файл `abc1`, затем с помощью команды `cp` скопировал его в каталог `~/ski.places` под новым именем `equiplist2`. С помощью команды `ls` проверил результат.

```
[liveuser@localhost-live ~]$ touch abc1
[liveuser@localhost-live ~]$ cp abc1 ski.places/equiplist2
[liveuser@localhost-live ~]$ ls ski.places
equiplist  equiplist2
[liveuser@localhost-live ~]$
```

Задание 2.5

Шаг 2.6

Создайте каталог с именем `equipment` в каталоге `~/ski.places`.

С помощью команды `cd` я перешел в `~/ski.places`, где с помощью `mkdir` создал каталог `equipment`.

```
[liveuser@localhost-live ski.plases]$ mkdir equipment
[liveuser@localhost-live ski.plases]$ mv equiplist equipment
[liveuser@localhost-live ski.plases]$ mv equiplist2 equipment
[liveuser@localhost-live ski.plases]$ tree
.
└─ equipment
   ├── equiplist
   └── equiplist2

1 directory, 2 files
[liveuser@localhost-live ski.plases]$
```

Задание 2.6

Шаг 2.7

Переместите файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment

С данным заданием справляется двухкратный вызов команды mv, а спомощью tree легко увидеть результат.

```
[liveuser@localhost-live ski.plases]$ mkdir equipment
[liveuser@localhost-live ski.plases]$ mv equiplist equipment
[liveuser@localhost-live ski.plases]$ mv equiplist2 equipment
[liveuser@localhost-live ski.plases]$ tree
.
└─ equipment
   ├── equiplist
   └── equiplist2

1 directory, 2 files
[liveuser@localhost-live ski.plases]$
```

Задание 2.7

Шаг 2.8

8. Создайте и переместите каталог ~/newdir в каталог ~/ski.plases и назовите его plans.

С помощью команды mkdir я создал каталог newdir, после чего переместил его в ~/ski.plases с помощью команды mv. С помощью ls увидел результат.

```
[liveuser@localhost-live ~]$ mkdir newdir
[liveuser@localhost-live ~]$ mv newdir ski.plases/plans
[liveuser@localhost-live ~]$ ls
abcl  Documents  may  montly  Pictures  reports  Templates
Desktop  Downloads  monthly  Music  Public  ski.plases  Videos
[liveuser@localhost-live ~]$ ls ski.plases
equipment  plans
[liveuser@localhost-live ~]$
```

Задание 2.8

Шаг 3 - Определите опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет.

На данном этапе я задавал права доступа к файлам в численном виде.

Шаг 3.1 - drwxr--r-- ... australia

```
[liveuser@localhost-live ~]$ touch australia
[liveuser@localhost-live ~]$ chmod 744 australia
[liveuser@localhost-live ~]$ ls -l australia
-rwxr--r--. 1 liveuser liveuser 0 May  2 10:23 australia
[liveuser@localhost-live ~]$
```

Задание 3.1

Шаг 3.2 - drwx--x--x ... play

```
[liveuser@localhost-live ~]$ touch play
[liveuser@localhost-live ~]$ chmod 711 play
[liveuser@localhost-live ~]$ ls -l play
-rwx--x--x. 1 liveuser liveuser 0 May  2 10:24 play
[liveuser@localhost-live ~]$
```

Задание 3.2

Шаг 3.3 - -r-xr--r-- ... my_os

```
[liveuser@localhost-live ~]$ chmod 544 my_os
[liveuser@localhost-live ~]$ ls -l my_os
-r-xr--r--. 1 liveuser liveuser 0 May  2 10:25 my_os
[liveuser@localhost-live ~]$
```

Задание 3.3

Шаг 3.4 - -rw-rw-r-- ... feathers

```
[liveuser@localhost-live ~]$ chmod 664 feathers
[liveuser@localhost-live ~]$ ls -l feathers
-rw-rw-r--. 1 liveuser liveuser 0 May  2 10:21 feathers
[liveuser@localhost-live ~]$
```

Задание 3.4

Шаг 4 - Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:

Шаг 4.1

Просмотрите содержимое файла /etc/passwd.

Я ввел cat /etc/passwd, но искомого файла не оказалось.

```
[liveuser@localhost-live ~]$ cat /etc/passwd
cat: /etc/passwd: No such file or directory
[liveuser@localhost-live ~]$
```

Задание 4.1

Шаг 4.2

Скопируйте файл ~/feathers в файл ~/file.old.

Я использовал команду `cp`.

```
[liveuser@localhost-live ~]$ cp feathers file.old
[liveuser@localhost-live ~]$ ls
abcl      Documents  file.old  montly  Pictures  reports  Videos
australia Downloads  may      Music   play     ski.plases
Desktop   feathers  monthly  my_os   Public   Templates
[liveuser@localhost-live ~]$
```

Задание 4.2

Шаг 4.3

Скопируйте файл ~/file.old в каталог ~/play.

Я использовал команду `cp`.

```
[liveuser@localhost-live ~]$ cp file.old play/
```

Задание 4.3

Шаг 4.4

Скопируйте каталог ~/play в каталог ~/fun.

Я использовал команду `cp` с опцией `-r`.

```
[liveuser@localhost-live ~]$ cp -r ~/play ~/fun
[liveuser@localhost-live ~]$ ls
abcl      Documents  file.old  monthly  my_os   Public  Templates
australia Downloads  fun       montly   Pictures reports  Videos
Desktop   feathers  may      Music    play     ski.plases
[liveuser@localhost-live ~]$
```

Задание 4.4

Шаг 4.5

Переместите каталог ~/fun в каталог ~/play и назовите его `games`.

Я использовал команду `mv`, результат проверил с помощью команды `tree`.

```
[liveuser@localhost-live ~]$ mv ~/fun ~/play/games
[liveuser@localhost-live ~]$ cd play
[liveuser@localhost-live play]$ tree
.
├── file.old
└── games
    └── file.old

1 directory, 2 files
[liveuser@localhost-live play]$
```

Задание 4.5

Шаг 4.6

Лишите владельца файла ~/feathers права на чтение.

Я использовал команду chmod с аргументом u-r.

```
[liveuser@localhost-live ~]$ chmod u-r feathers
```

Задание 4.6

Шаг 4.7

Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?

Я получил ошибку “cat: feathers: Permission denied”, что означает “отказано в доступе”.

```
[liveuser@localhost-live ~]$ chmod u-r feathers
[liveuser@localhost-live ~]$ cat feathers
cat: feathers: Permission denied
[liveuser@localhost-live ~]$
```

Задание 4.7

Шаг 4.8

Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?

Я получил ошибку “cp: cannot open ‘feathers’ for reading: Permission denied”, - программа cp не смогла прочитать файл, так как у пользователя отсутствует на это право.

```
[liveuser@localhost-live ~]$ cp feathers feathers_copy
cp: cannot open 'feathers' for reading: Permission denied
[liveuser@localhost-live ~]$
```

Задание 4.8

Шаг 4.9

Дайте владельцу файла ~/feathers право на чтение.

Я использовал команду chmod с аргументом u+r. В изменении прав убедился с помощью ls -l.

```
[liveuser@localhost-live ~]$ chmod u+r feathers
[liveuser@localhost-live ~]$ ls -l feathers
-rw-rw-r--. 1 liveuser liveuser 0 May  2 10:21 feathers
[liveuser@localhost-live ~]$
```

Задание 4.9

Шаг 4.10

Лишите владельца каталога ~/play права на выполнение.

Я использовал команду `chmod` с аргументом `u-x`.

```
[liveuser@localhost-live ~]$ chmod u-x play
```

Задание 4.10

Шаг 4.11

Перейдите в каталог `~/play`. Что произошло?

При попытке перейти в каталог мне было отказано в доступе.

```
[liveuser@localhost-live ~]$ cd play
bash: cd: play: Permission denied
[liveuser@localhost-live ~]$
```

Задание 4.11

Шаг 4.12

Дайте владельцу каталога `~/play` право на выполнение.

Я использовал команду `chmod` с аргументом `u+x`.

```
[liveuser@localhost-live ~]$ chmod u+x play
```

Задание 4.12

Шаг 5 - Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте,приведя примеры.

`mount`

Команда **`mount`** позволяет объединить несколько файловых систем в единое дерево каталогов. Для подмонтирования нового устройства нужно написать `mount` файл_устройства пака_назначения.

```
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

    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.

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

man mount

fsck

fsck - это утилита для проверки и восстановления файловых систем Linux. Обычно команда fsck автоматически запускается по возможности в параллельном режиме при загрузке ОС. По этой причине обычно нет необходимости запускать ее через командную строку.


```
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
        Filesystem errors left uncorrected

    8
        Operational error

    16
        Usage or syntax error

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

man fsck

mkfs

Команда **mkfs** используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жесткого диска. В качестве аргумента может выступать название устройства (например /dev/sda1) или точка монтирования (например /, /home).

```

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.fs<type>) 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.

    -V, --version
        Display version information and exit. (Option -V will display
        version information only when it is the only parameter, otherwise it
        will work as

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

```

man mkfs

kill

Команда **kill** позволяет отправить сигнал процессу, принимая на вход его PID идентификатор. Например, можно принудительно завершить процесс с PID=2000

набрав `kill -KILL 2000`

```
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.

        0
```

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

Вывод

Я познакомился с файловой системой Linux, ее структурой, именами и содержанием основных каталогов. Также приобрел практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами и по проверке диска и обслуживанию файловых систем.