

Лабораторная работа №5: Презентация.

Анализ файловой системы Linux. Команды для работы с файлами и каталогами.

Евдокимов Максим Михайлович. Группа - НФИбд-01-20.¹

24 декабря, 2023, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи 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.

1. Работа с файловой системой и методы её определения и создания.
2. Настройка и работа с уровнями доступа к файлам и директориям.
3. Перемещение, удаление и переназначение файлов и директорий.

Указание к работе

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 “head” выводит по умолчанию первые 10 строк файла или указанное количество n.

Команда “tail” выводит по умолчанию 10 последних строк файла или указанное количество n.

Команда “cp” используется для копирования файлов и каталогов.

Команды “mv” и “mvdir” предназначены для перемещения и переименования файлов и каталогов.

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

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

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

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.

С помощью команды “fsck” можно проверить (а в ряде случаев восстановить) целостность файловой системы.

Процесс выполнения лабораторной работы

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

```
max@evdokimov:~$ sudo -i
[sudo] пароль для max:
[root@evdokimov ~]# touch abc1
[root@evdokimov ~]# cp abc1 april
[root@evdokimov ~]# cp abc1 may; ls
335728forGit 335728forGit.pub abc1 anaconda-ks.cfg april may saymyname saymyname.pub test.txt work
[root@evdokimov ~]#
```

Рис. 1: Пример копирования файлов

```
[root@evdokimov ~]# mkdir monthly  
[root@evdokimov ~]# cp april may monthly; ls  
335728forGit 335728forGit.pub abcl anaconda-ks.cfg april may monthly saymyname saymyname.pub test.txt work  
[root@evdokimov ~]#
```

Рис. 2: Пример копирование нескольких файлов

```
[root@evdokimov ~]# cp monthly/may monthly/june; ls monthly  
april  june  may  
[root@evdokimov ~]#
```

Рис. 3: Пример кокирование в произвольный каталог

```
[root@evdokimov ~]# mkdir monthly.00  
[root@evdokimov ~]# cp -r monthly monthly.00  
[root@evdokimov ~]#
```

Рис. 4: Пример копирования в текущий каталог

```
[root@evdokimov ~]# cp -r monthly.00 /tmp; ls /tmp
monthly.00
ssh-XXXXXXNBKxNZ
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-chrond.service-wSndg7
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-dbus-broker.service-cN7yh3
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-ModemManager.service-ygPDuc
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-polkit.service-VlosAJ
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-rtkit-daemon.service-DAjYT0
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-systemd-logind.service-d0vFlu
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-systemd-oomd.service-ugspie
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-systemd-resolved.service-zKwyGB
systemd-private-97920cdc28eb4c0d889ebcf4514af3f9-upower.service-QK8T05
[root@evdokimov ~]#
```

Рис. 5: Пример копирование в произвольный каталог 2

```
[root@evdokimov ~]# cd; mv april july  
[root@evdokimov ~]# mv july monthly.00; ls monthly.00  
july  monthly  
[root@evdokimov ~]#
```

Рис. 6: Пример переименования текущего каталога

```
[root@evdokimov ~]# mv monthly.00 monthly.01
[root@evdokimov ~]# mkdir reports
[root@evdokimov ~]# mv monthly.01 reports
[root@evdokimov ~]# mv reports/monthly.01 reports/monthly
[root@evdokimov ~]# ls
335728forGit 335728forGit.pub abc1 anaconda-ks.cfg may monthly reports saymyname saymyname.pub test.txt work
[root@evdokimov ~]#
```

Рис. 7: Пример переименования и перемещения каталогов и файлов


```
[root@evdokimov ~]# cd; touch may; ls -l may
-rw-r--r--. 1 root root 0 дек 25 15:09 may
[root@evdokimov ~]# chmod u+x may; ls -l may
-rwxr--r--. 1 root root 0 дек 25 15:09 may
[root@evdokimov ~]# chmod u-x may; ls -l may
-rw-r--r--. 1 root root 0 дек 25 15:09 may
[root@evdokimov ~]# cd; mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
[root@evdokimov ~]# chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
[root@evdokimov ~]# cd; touch abc1; chmod g+w abc1
[root@evdokimov ~]# ls -l abc1
-rw-rw-r--. 1 root root 0 дек 25 15:10 abc1
[root@evdokimov ~]#
```

Рис. 8: Пример изменения уровня доступа

```
[root@evdokinov ~]# mount
/dev/sda3 on / type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvolid=257,subvol=/root)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=377862,mode=755,inode64)
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)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
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)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=612832k,nr_inodes=819200,mode=755,inode64)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=34,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=4045)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,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)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,nr_inodes=1048576,inode64)
/dev/sda2 on /boot type ext4 (rw,relatime,seclabel)
/dev/sda3 on /home type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvolid=256,subvol=/home)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=306412k,nr_inodes=76603,mode=700,uid=1000,gid=1000,inode64)
[root@evdokinov ~]#
```

Рис. 9: Пример применения команды mount

```
[root@evdokimov ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Fri Dec 22 09:05:32 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=33692ece-0306-424b-8ac7-4217df7cfbfa / btrfs subvol=root,compress=zstd:1 0 0
UUID=90037961-4104-4b24-88c0-8ed289bcf2de /boot ext4 defaults 1 2
UUID=33692ece-0306-424b-8ac7-4217df7cfbfa /home btrfs subvol=home,compress=zstd:1 0 0
```

Рис. 10: Пример применения команды cat /etc/fstab

```
[root@evdokimov ~]# df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
/dev/sda3          82834432      3469884  77519668          5% /
devtmpfs           4096           0     4096           0% /dev
tmpfs             1532072           0  1532072          0% /dev/shm
tmpfs             612832          1084   611748          1% /run
tmpfs             1532072           4  1532068          1% /tmp
/dev/sda2          996780          270148   657820         30% /boot
/dev/sda3          82834432      3469884  77519668          5% /home
tmpfs             306412           80   306332          1% /run/user/1000
```

Рис. 11: Пример применения команды df

```
[root@evdokimov ~]# fsck /dev/sda1
fsck from util-linux 2.39.2
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>
```

Рис. 12: Пример применения команды fsck к каталогу /dev/sda1

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

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.

```
max@evdokimov:~$ pwd
/home/max
max@evdokimov:~$ cp /usr/include/sys/io.h /home/max/equipment
max@evdokimov:~$ ls
equipment  new  times  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
max@evdokimov:~$
```

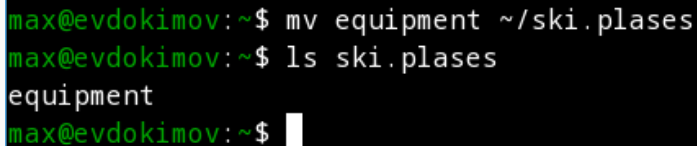
Рис. 13: Копирование `io.h`

2.2. В домашнем каталоге создайте директорию ~/ski.plases.

```
max@evdokimov:~$ mkdir ski.plases
max@evdokimov:~$ ls | grep "ski"; ls -F | grep "ski"
ski.plases
ski.plases/
max@evdokimov:~$
```

Рис. 14: Создание ski.plases

2.3. Переместите файл equipment в каталог ~/ski.places.



```
max@evdokimov:~$ mv equipment ~/ski.places
max@evdokimov:~$ ls ski.places
equipment
max@evdokimov:~$
```

Рис. 15: Перемещение equipment

2.4. Переименуйте файл ~/ski.plases/equipment в ~/ski.plases/equiplist.

```
max@evdokimov:~$ mv ~/ski.plases/equipment ~/ski.plases/equiplist
max@evdokimov:~$ ls ski.plases
equiplist
max@evdokimov:~$
```

Рис. 16: Переименование файла equipment

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

```
max@evdokimov:~$ touch abc1; ls
abc1  new  ski.places  times  Видео  Документы  Загрузки  Изображения  Музыка  Общедоступные  'Рабочий стол'  Шаблоны
max@evdokimov:~$ cp abc1 ~/ski.places; ls ski.places
abc1  equiplist
max@evdokimov:~$
```

Рис. 17: Перенос и переименование файла abc1

2.6. Создайте каталог с именем equipment в каталоге ~/ski.plases.

```
max@evdokimov:~$ mkdir ~/ski.plases/equipment; ls ski.plases
abc1  equiplist  equipment
max@evdokimov:~$
```

Рис. 18: Создание каталога equipment

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

```
max@evdokimov:~$ mv ~/ski.plases/equiplist ~/ski.plases/equipment
max@evdokimov:~$ mv ~/ski.plases/equiplist2 ~/ski.plases/equipment
mv: не удалось выполнить stat для '/home/max/ski.plases/equiplist2': Нет такого файла или каталога
max@evdokimov:~$ ls ski.plases; ls ski.plases/equipment/
abc1  equipment
equiplist
max@evdokimov:~$
```

Рис. 19: Перемещение файлов из каталога в каталог

2.8. Создайте и переместите каталог ~/newdir в каталог ~/ski.places и назовите его plans.

```
max@evdokimov:~$ mkdir ~/newdir; ls | grep new
new
newdir
max@evdokimov:~$ mv newdir ~/ski.places/plans; ls ski.places
abc1  equipment  plans
max@evdokimov:~$
```

Рис. 20: Перемещение и переименование каталога

3. Определите опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет (При необходимости создайте нужные файлы): 3.1. `drwxr-r- ... australia` 3.2. `drwx-x-x ... play` 3.3. `-r-xr-r- ... my_os` 3.4. `-rw-rw-r- ... feathers`

```
max@evdokimov:~$ mkdir australia play my_os feathers
max@evdokimov:~$ ls -ld
drwx-----, 1 max max 1306 дек 25 15:33 .
max@evdokimov:~$ ls
abc1      feathers  new      ski.places  Видео      Загрузки    Музыка      'Рабочий стол'
australia my_os     play     times       Документы  Изображения  Общедоступные  Шаблоны
max@evdokimov:~$ chmod 744 australia; chmod 711 play; chmod 544 my_os; chmod 664 feathers
max@evdokimov:~$ ls -l australia play my_os feathers
australia:
итого 0

feathers:
итого 0

my_os:
итого 0

play:
итого 0
```

Рис. 21: Работа с уровнем доступа 4 файлов

```
max@evdokimov:~$ ls -l
итого 0
-rw-r--r--. 1 max max 0 дек 25 15:30 abc1
drwxr--r--. 1 max max 0 дек 25 15:33 australia
drw-rw-r--. 1 max max 0 дек 25 15:33 feathers
dr-xr--r--. 1 max max 0 дек 25 15:33 my_os
drwxr-xr-x. 1 max max 0 дек 25 01:51 new
drwx--x--x. 1 max max 0 дек 25 15:33 play
drwxr-xr-x. 1 max max 36 дек 25 15:32 ski.places
drwxr-xr-x. 1 max max 0 дек 25 01:51 times
drwxr-xr-x. 1 max max 0 дек 22 18:06 Видео
drwxr-xr-x. 1 max max 0 дек 22 18:06 Документы
drwxr-xr-x. 1 max max 0 дек 22 18:06 Загрузки
drwxr-xr-x. 1 max max 0 дек 22 18:06 Изображения
drwxr-xr-x. 1 max max 0 дек 22 18:06 Музыка
drwxr-xr-x. 1 max max 0 дек 22 18:06 Общедоступные
drwxr-xr-x. 1 max max 0 дек 22 18:06 'Рабочий стол'
drwxr-xr-x. 1 max max 0 дек 22 18:06 Шаблоны
```

Рис. 22: Просмотр результата

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

4.1. Просмотрите содержимое файла /etc/passwd.

```
max@evdokiMOV:~$ cat /etc/passwd; ls /etc | grep pass
cat: /etc/passwd: Нет такого файла или каталога
passwd
passwd-
passwdqc.conf
max@evdokiMOV:~$ 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
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
polkitd:x:114:114:User for polkitd:/usr/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/usr/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/usr/sbin/nologin
pipewire:x:995:995:PipeWire System Daemon:/run/pipewire:/usr/sbin/nologin
chrony:x:994:994:chrony system user:/var/lib/chrony:/usr/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/usr/sbin/nologin
colord:x:993:993:User for colord:/var/lib/colord:/usr/sbin/nologin
unbound:x:992:992:Unbound DNS resolver:/var/lib/unbound:/usr/sbin/nologin
geoclue:x:991:991:User for geoclue:/var/lib/geoclue:/usr/sbin/nologin
setroubleshoot:x:990:990:SELinux troubleshoot server:/var/lib/setroubleshoot:/usr/sbin/nologin
```

Рис. 23: Просмотр файла passwd

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

```
max@evdokimov:~$ cp ~/feathers ~/file.old; ls file.old
cp: не указан -r; пропускается каталог '/home/max/feathers'
ls: невозможно получить доступ к 'file.old': Нет такого файла или каталога
max@evdokimov:~$ mkdir file.old
max@evdokimov:~$ cp ~/feathers ~/file.old; ls file.old
cp: не указан -r; пропускается каталог '/home/max/feathers'
max@evdokimov:~$ cp -r ~/feathers ~/file.old; ls file.old
feathers
max@evdokimov:~$
```

Рис. 24: Копирование файлов и содержания

4.3. Переместите файл ~/file.old в каталог ~/play.

```
max@evdokimov:~$ mv file.old play; ls play  
file.old  
max@evdokimov:~$
```

Рис. 25: Перемещение файлов

4.4. Скопируйте каталог ~/play в каталог ~/fun.

```
max@evdokimov:~$ cp play fun; ls; cp -r play fun; ls fun
cp: не указан -r; пропускается каталог 'play'
abc1      feathers  new      ski.plases  Видео      Загрузки    Музыка      'Рабочий стол'
australia my_os     play     times      Документы  Изображения  Общедоступные  Шаблоны
file.old
max@evdokimov:~$
```

Рис. 26: Проводим копирование каталога play

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

```
max@evdokimov:~$ mv fun play; ls
abc1      feathers  new      ski.places  Видео      Загрузки    Музыка      'Рабочий стол'
australia my_os     play     times      Документы  Изображения  Общедоступные  Шаблоны
max@evdokimov:~$ cd play; mv fun games; ls
file.old  games
max@evdokimov:~/play$
```

Рис. 27: Проводим копирование каталога fun

4.6. Лишите владельца файла ~/feathers права на чтение.

```
max@evdokimov:~/play$ chmod u-r feathers; ls -ld feathers
chmod: невозможно получить доступ к 'feathers': Нет такого файла или каталога
ls: невозможно получить доступ к 'feathers': Нет такого файла или каталога
max@evdokimov:~/play2$ cd
max@evdokimov:~$ chmod u-r feathers; ls -ld feathers
d-w-rw-r--. 1 max max 0 дек 25 15:33 feathers
```

Рис. 28: Ограничение на чтение

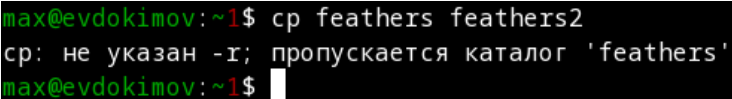
4.7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?

A terminal window with a black background and green text. The first line shows the prompt 'max@evdokimov:~\$' followed by the command 'cat feathers'. The second line shows the output 'cat: feathers: Отказано в доступе'. The third line shows the prompt 'max@evdokimov:~1\$' followed by a white cursor block.

```
max@evdokimov:~$ cat feathers
cat: feathers: Отказано в доступе
max@evdokimov:~1$
```

Рис. 29: Попытка чтение файла

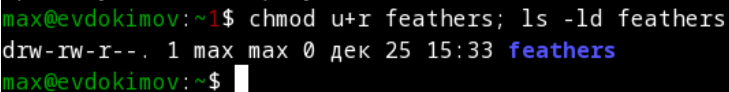
4.8. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?

A terminal window with a black background and green text. The prompt is 'max@evdokimov:~1\$'. The command 'cp feathers feathers2' is entered. The output is 'cp: не указан -r; пропускается каталог 'feathers''. The prompt 'max@evdokimov:~1\$' is shown again with a white cursor block.

```
max@evdokimov:~1$ cp feathers feathers2
cp: не указан -r; пропускается каталог 'feathers'
max@evdokimov:~1$
```

Рис. 30: Попытка копирования

4.9. Дайте владельцу файла ~/feathers право на чтение.



```
max@evdokimov:~$ chmod u+r feathers; ls -ld feathers
drw-rw-r-- 1 max max 0 дек 25 15:33 feathers
max@evdokimov:~$
```

Рис. 31: Возвращения прав feathers

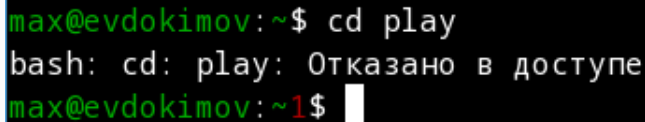
4.10. Лишите владельца каталога ~/play права на выполнение.

A terminal window with a black background and green text. The prompt is 'max@evdokimov:~\$'. The command 'chmod u-x play' is entered on the first line. On the second line, the prompt 'max@evdokimov:~\$' is followed by a white cursor block.

```
max@evdokimov:~$ chmod u-x play
max@evdokimov:~$
```

Рис. 32: Ограничения выполнения для play

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

A terminal window with a black background. The prompt 'max@evdokimov:~\$' is in green. The command 'cd play' is entered in white. The response 'bash: cd: play: Отказано в доступе' is shown in white. The prompt 'max@evdokimov:~1\$' is shown in green, with a white cursor block at the end.

```
max@evdokimov:~$ cd play
bash: cd: play: Отказано в доступе
max@evdokimov:~1$
```

Рис. 33: Проверка на доступ

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

```
max@evdokimov:~1$ chmod u+x play
max@evdokimov:~$ ls -l
итого 0
-rw-r--r--. 1 max max  0 дек 25 15:30 abc1
drwxr--r--. 1 max max  0 дек 25 15:33 australia
drw-rw-r--. 1 max max  0 дек 25 15:33 feathers
dr-xr--r--. 1 max max  0 дек 25 15:33 my_os
drwxr-xr-x. 1 max max  0 дек 25 01:51 new
drwx--x--x. 1 max max 26 дек 25 15:41 play
drwxr-xr-x. 1 max max 36 дек 25 15:32 ski.places
drwxr-xr-x. 1 max max  0 дек 25 01:51 times
drwxr-xr-x. 1 max max  0 дек 22 18:06 Видео
drwxr-xr-x. 1 max max  0 дек 22 18:06 Документы
drwxr-xr-x. 1 max max  0 дек 22 18:06 Загрузки
drwxr-xr-x. 1 max max  0 дек 22 18:06 Изображения
drwxr-xr-x. 1 max max  0 дек 22 18:06 Музыка
drwxr-xr-x. 1 max max  0 дек 22 18:06 Общедоступные
drwxr-xr-x. 1 max max  0 дек 22 18:06 'Рабочий стол'
drwxr-xr-x. 1 max max  0 дек 22 18:06 Шаблоны
max@evdokimov:~$
```

Рис. 34: Возвращение прав выполнение

5. Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, 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|xprivate|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)
```

Рис. 35: Анализ командой man команды mount

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

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTNMP] [-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/sdb1), 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 conditions:

    0      No errors

    1      Filesystem errors corrected

    2      System should be rebooted

    4      Filesystem errors left uncorrected

    8      Operational error

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

Рис. 36: Анализ командой man команды fsck

```
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> utilities.

    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
        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.

    -h, --help
        Display help text and exit.

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

Рис. 37: Анализ командой `man` команды `mkfs`

```

Kill(1)                                     User Commands                                     Kill(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-o] [--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
            All processes in the current process group are signaled.
Manual page Kill(1) line 1 (press h for help or q to quit)

```

Рис. 38: Анализ командой man команды kill

Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

```
max@evdokimov:~$ findmnt -l
TARGET                SOURCE                FSTYPE    OPTIONS
/                     /dev/sda3[/root]     btrfs     rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvol=1,subvol=/root
/dev                 devtmpfs              devtmpfs   rw,nosuid,seclabel,size=4096k,nr_inodes=377862,mode=755,inode64
/dev/shm              tmpfs                 tmpfs      rw,nosuid,nodev,seclabel,inode64
/dev/pts              devpts                devpts     rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000
/sys                  sysfs                 sysfs      rw,nosuid,nodev,noexec,relatime,seclabel
/sys/kernel/security  securityfs            securityfs rw,nosuid,nodev,noexec,relatime
/sys/fs/cgroup         cgroup2               cgroup2    rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot
/sys/fs/pstore         pstore                pstore     rw,nosuid,nodev,noexec,relatime,seclabel
/sys/fs/bpf            bpf                   bpf        rw,nosuid,nodev,noexec,relatime,mode=700
/sys/kernel/config     configfs              configfs    rw,nosuid,nodev,noexec,relatime
/proc                 proc                  proc        rw,nosuid,nodev,noexec,relatime
/run                  tmpfs                 tmpfs      rw,nosuid,nodev,seclabel,size=612832k,nr_inodes=819200,mode=755,inode64
/sys/fs/selinux        selinuxfs             selinuxfs   rw,nosuid,noexec,relatime
/proc/sys/fs/binfmt_misc system-1              autofs     rw,relatime,fd=34,prgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=4045
/dev/hugepages         hugetlbfs             hugetlbfs   rw,nosuid,nodev,relatime,seclabel,pagesize=2M
/dev/mqueue            mqueue               mqueue     rw,nosuid,nodev,noexec,relatime,seclabel
/sys/kernel/debug      debugfs               debugfs     rw,nosuid,nodev,noexec,relatime,seclabel
/sys/kernel/tracing     tracefs               tracefs     rw,nosuid,nodev,noexec,relatime,seclabel
/sys/fs/fuse/connections fusectl               fusectl     rw,nosuid,nodev,noexec,relatime
/tmp                   tmpfs                 tmpfs      rw,nosuid,nodev,seclabel,nr_inodes=1048576,inode64
/boot                  ext4                  ext4        rw,relatime,seclabel
/home                  /dev/sda3[/home]     btrfs      rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvol=1,subvol=/home
/var/lib/nfs/rpc_pipefs sunrpc                rpc_pipefs  rw,relatime
/run/user/1000         tmpfs                 tmpfs      rw,nosuid,nodev,relatime,seclabel,size=306412k,nr_inodes=76603,mode=700,uid=1000,gid=1000,inode64
```

Рис. 39: Просмотр файловых систем командой findmnt -l

- `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).
- `selinux` - Как и файловая система `/proc`, `/selinux` является псевдофайловой системой. Новая реализация SE Linux использует расширенные атрибуты для хранения контекста безопасности.
- `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.

- 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.

- `configf` - 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.

Верхний уровень / => root каталог, содержит в себе всю иерархию системы и от него идут все остольные:

/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?

Команда `mount` то есть монтирование тома (монтирование дисков).

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:

- Один блок адресуется несколькими `inode` (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).

5. Как создаётся файловая система?

mkfs - позволяет создать файловую систему Linux, семейства ext: “sudo mkfs -t тип устройство”.

6. Дайте характеристику командам для просмотра текстовых файлов.

- `cat` - 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` . Команда `Less` просматривает файл по одной странице за раз.
- `Head` . 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` . 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, также используется для копирования файлов из одного местоположения в другое, используется для копирования всех каталогов в новое место или 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? При ответах на вопросы используйте дополнительные источники информации по теме.

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:

- Чтение - разрешает получать содержимое файла. 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 - разрешает запускать программу, если у нее есть флаг выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл является запускаемым.

Для изменение этих данных (разделённых на 3 типа пользователей u - user, g - group, o - others) используется команда `chmod`.

Выводы по проделанной работе

В ходе выполнения лабораторной работы были изучены файловые системы linux а также повторены основы работы с файлами и каталогами и их уровнями доступа.