

Лабораторная работа № 5

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

Андреева Яна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Контрольные вопросы	20
6	Выводы	27
7	Список литературы	28

Список иллюстраций

Список таблиц

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 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:
 - 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.
 - 2.2. В домашнем каталоге создайте директорию `~/ski.places`.
 - 2.3. Переместите файл `equipment` в каталог `~/ski.places`.
 - 2.4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 - 2.5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
 - 2.6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 - 2.7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 - 2.8. 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:
 - 3.1. `drwxr-r- ... australia`

3.2. `drwx-x-x ... play`

3.3. `-r-xr-r- ... my_os`

3.4. `-rw-rw-r- ... feathers`

При необходимости создайте нужные файлы.

4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 - 4.1. Просмотрите содержимое файла `/etc/passwd`.
 - 4.2. Скопируйте файл `~/feathers` в файл `~/file.old`.
 - 4.3. Переместите файл `~/file.old` в каталог `~/play`.
 - 4.4. Скопируйте каталог `~/play` в каталог `~/fun`.
 - 4.5. Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`.
 - 4.6. Лишите владельца файла `~/feathers` права на чтение.
 - 4.7. Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`?
 - 4.8. Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`?
 - 4.9. Дайте владельцу файла `~/feathers` право на чтение.
 - 4.10. Лишите владельца каталога `~/play` права на выполнение.
 - 4.11. Перейдите в каталог `~/play`. Что произошло?
 - 4.12. Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте ман по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.

3 Теоретическое введение

Для создания текстового файла можно использовать команду touch.

Формат команды:

touch имя-файла

Для просмотра файлов небольшого размера можно использовать команду cat.

Формат команды:

cat имя-файла

Для просмотра файлов постранично удобнее использовать команду less.

Формат команды:

less имя-файла

4 Выполнение лабораторной работы

Выполним все примеры, приведённые в первой части описания лабораторной

работы.

```
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ touch newfile
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cat newfile
str1
str2
str3
str4
str5
str6
str7
str8
str9
str10
str11
str12
str13
str14
str15
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ less newfile
```

```
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ head -5 newfile
str1
str2
str3
str4
str5
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ tail -5 newfile
str11
str12
str13
str14
str15
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ touch newfile2
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cp newfile newfile2
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mkdir newfiles
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cp newfile newfile2 newfiles
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cp newfiles/newfile newfiles/newfile2
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mkdir newfiles.00
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cp -r newfiles newfiles.00
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cp -r newfiles.00 /tmp
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mv newfile oldfile
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mv oldfile newfiles.00
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mv newfiles.00 newfiles.01
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mkdir reports
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mv newfiles.01 reports
```

```
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report
chmod: невозможно получить доступ к 'newfile': Нет такого файла или каталога
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ ls
bib image Makefile newfile2 newfiles newile2 pandoc report.md reports
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ chmod u+x newfile2
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ ls -l newfile2
-rwxr--r--. 1 yaandreeva yaandreeva 81 июн  2 22:30 newfile2
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ chmod u-x newfile2
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ ls -l newfile2
-rw-r--r--. 1 yaandreeva yaandreeva 81 июн  2 22:30 newfile2
```

```
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ mount
/dev/sda3 on / type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvol=257,subvol=/root)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=245735,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=400768k,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=37,prp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=6218)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,relatime,seclabel,pagesize=2M)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/credentials/systemd-journald.service type tmpfs (ro,nosuid,nodev,noexec,relatime,nosymfollow,seclabel,size=1024k,nr_inodes=1024,mode=700,inode64,noswap)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,size=1001920k,nr_inodes=1048576,inode64)
/dev/sda3 on /home type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvol=256,subvol=/home)
/dev/sda2 on /boot type ext4 (rw,relatime,seclabel)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run/credentials/systemd-resolved.service type tmpfs (ro,nosuid,nodev,noexec,relatime,nosymfollow,seclabel,size=1024k,nr_inodes=1024,mode=700,inode64,noswap)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
work on /media/sf_work type vboxsf (rw,nodev,relatime,icharset=utf8,uid=0,gid=982,dmode=0770,fmode=0770,tag=VBoxAutomounte
r)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=200380k,nr_inodes=50095,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/sr0 on /run/media/yaandreeva/VBox_GAs_7.1.8 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,icharset=utf8,uhelper=udisks2)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
```

```
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Fri May 23 07:34:36 2025
#
# 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=8cb16f2b-63fd-45e0-98e0-a65f5c2827dd / btrfs subvol=root,compress=zstd:1 0 0
UUID=9045a02e-da85-4f98-8b0f-ea5cdfac1f8 /boot ext4 defaults 1 2
UUID=8cb16f2b-63fd-45e0-98e0-a65f5c2827dd /home btrfs subvol=home,compress=zstd:1 0 0
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ df
Файловая система 1К-блоков Использовано Доступно Использовано% Смонтировано в
/dev/sda3 82834432 12757700 69313292 16% /
devtmpfs 4096 0 4096 0% /dev
tmpfs 1001916 88 1001828 1% /dev/shm
tmpfs 400768 1448 399320 1% /run
tmpfs 1024 0 1024 0% /run/credentials/systemd-journald.service
tmpfs 1001920 340 1001580 1% /tmp
/dev/sda3 82834432 12757700 69313292 16% /home
/dev/sda2 996780 337896 590072 37% /boot
tmpfs 1024 0 1024 0% /run/credentials/systemd-resolved.service
work 487755772 141445860 346309912 29% /media/sf_work
tmpfs 200380 236 200144 1% /run/user/1000
/dev/sr0 59942 59942 0 100% /run/media/yaandreeva/VBox_GAs_7.1.8
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$
```

Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовем его `equipment`. В домашнем каталоге создадим директорию `~/ski.places`.

```
yaandreeva@yaandreeva:~$ cp /usr/include/sys/io.h ~/work/equipment
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ ls ~/work
equipment study
yaandreeva@yaandreeva:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab05/report$ cd
yaandreeva@yaandreeva:~$ mkdir ~/ski.places
yaandreeva@yaandreeva:~$ ls
Desktop Documents Downloads Music Pictures Public ski.places study_2024-2025_os-intro
```

Переместим файл `equipment` в каталог `~/ski.places`. Переименуем файл `~/ski.places/equipment` в `~/ski.places/equiplist`. Создадим в домашнем каталоге

```
yaandreeva@yaandreeva:~  
yaandreeva@yaandreeva:~$ mv ~/work/equipment ~/ski.places  
yaandreeva@yaandreeva:~$ ls ~/ski.places  
equipment  
yaandreeva@yaandreeva:~$ cd ski.places  
yaandreeva@yaandreeva:~/ski.places$ mv equipment equiplist  
yaandreeva@yaandreeva:~/ski.places$ ls  
equiplist  
yaandreeva@yaandreeva:~/ski.places$ cd  
yaandreeva@yaandreeva:~$ touch abc1  
yaandreeva@yaandreeva:~$ ls  
abc1 Documents Music Public ski.places
```

файл abc1.

Скопируем его в каталог ~/ski.places, назовем его equiplist2. Создадим каталог с именем equipment в каталоге ~/ski.places. Переместим файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment. Создадим и переместим каталог ~/newdir в каталог ~/ski.places и назовем его plans.

```
yaandreeva@yaandreeva:~  
yaandreeva@yaandreeva:~$ mv abc1 ~/ski.places/equiplist2  
yaandreeva@yaandreeva:~$ ls ~/ski.places  
equiplist equiplist2  
yaandreeva@yaandreeva:~$ cd ski.places  
yaandreeva@yaandreeva:~/ski.places$ mkdir equipment  
yaandreeva@yaandreeva:~/ski.places$ ls  
equiplist equiplist2 equipment  
yaandreeva@yaandreeva:~/ski.places$ mv equiplist equipment  
yaandreeva@yaandreeva:~/ski.places$ mv equiplist2 equipment  
yaandreeva@yaandreeva:~/ski.places$ ls equipment  
equiplist equiplist2  
yaandreeva@yaandreeva:~/ski.places$ cd  
yaandreeva@yaandreeva:~$ mkdir newdir  
yaandreeva@yaandreeva:~$ mv newdir ~/ski.places/plans  
yaandreeva@yaandreeva:~$ ls ~/ski.places  
equipment plans  
yaandreeva@yaandreeva:~$
```

Определим опции команды chmod, необходимые для того, чтобы присвоить

перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет: drwxr-r- ... australia drwx-x-x ... play -r-xr-r- ... my_os -rw-rw-r- ...

```
yaandreeva@yaandreeva:~/ski.places
~/ski.places

yaandreeva@yaandreeva:~$ cd ski.places
yaandreeva@yaandreeva:~/ski.places$ touch australia play my_os feathers
yaandreeva@yaandreeva:~/ski.places$ ls -l australia play my_os feathers
-rw-r--r--. 1 yaandreeva yaandreeva 0 июн  2 23:36 australia
-rw-r--r--. 1 yaandreeva yaandreeva 0 июн  2 23:36 feathers
-rw-r--r--. 1 yaandreeva yaandreeva 0 июн  2 23:36 my_os
-rw-r--r--. 1 yaandreeva yaandreeva 0 июн  2 23:36 play
yaandreeva@yaandreeva:~/ski.places$ chmod 544 my_os
yaandreeva@yaandreeva:~/ski.places$ chmod 664 feathers
yaandreeva@yaandreeva:~/ski.places$ chmod 744 australia
yaandreeva@yaandreeva:~/ski.places$ chmod 711 play
yaandreeva@yaandreeva:~/ski.places$ ls -l australia play my_os feathers
-rwxr--r--. 1 yaandreeva yaandreeva 0 июн  2 23:36 australia
-rw-rw-r--. 1 yaandreeva yaandreeva 0 июн  2 23:36 feathers
-r-xr--r--. 1 yaandreeva yaandreeva 0 июн  2 23:36 my_os
-rwx--x--x. 1 yaandreeva yaandreeva 0 июн  2 23:36 play
yaandreeva@yaandreeva:~/ski.places$
```

feathers

```
yaandreeva@yaandreeva:/etc$ cat 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/bin/nologin
tss:x:59:59:Account used for TPM access:/usr/bin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
geoclue:x:999:999>User for geoclue:/var/lib/geoclue:/sbin/nologin
systemd-oom:x:998:998:systemd Userspace OOM Killer:/usr/bin/nologin
polkitd:x:114:114>User for polkitd:/sbin/nologin
ssnptc:x:997:996:Secure Socket Tunneling Protocol (SSTP) Client:/usr/bin/nologin
rtkit:x:172:172:RealtimeKit:/sbin/nologin
chrony:x:995:995:chrony system user:/var/lib/chrony:/sbin/nologin
systemd-coredump:x:994:994:systemd Core Dumper:/usr/bin/nologin
systemd-timesync:x:993:993:systemd Time Synchronization:/usr/bin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/usr/bin/nologin
systemd-resolve:x:193:193:systemd Resolver:/usr/bin/nologin
pipewire:x:992:992:PipeWire System Daemon:/run/pipewire:/usr/bin/nologin
sssd:x:991:991>User for sssd:/run/sss:/sbin/nologin
unbound:x:990:990:Unbound DNS resolver:/var/lib/unbound:/sbin/nologin
nm-openconnect:x:989:989:NetworkManager user for OpenConnect:/usr/bin/nologin
wsdd:x:988:988:Web Services Dynamic Discovery host daemon:/usr/bin/nologin
openvpn:x:987:987:OpenVPN:/etc/openvpn:/sbin/nologin
nm-openvpn:x:986:986:Default user for running openvpn spawned by NetworkManager:/usr/bin/nologin
colord:x:985:985>User for colord:/var/lib/colord:/sbin/nologin
abrt:x:173:173:/etc/abrt:/sbin/nologin
setroubleshoot:x:984:984:SELinux troubleshoot server:/var/lib/abrt:/usr/bin/nologin
sddm:x:983:983:SDDM Greeter Account:/var/lib/sddm:/usr/bin/nologin
```

Просмотрим содержимое файла/etc/passwd.

Скопируем файл ~/feathers в файл ~/file.old. Переместим файл ~/file.old в каталог ~/play. Скопируем каталог ~/play в каталог ~/fun. Переместим каталог ~/fun в каталог ~/play и назовем его games. Лишим владельца файла ~/feathers права на чтение. Если мы попытаемся просмотреть файл ~/feathers командой cat, будет отказано в доступе. Если мы попытаемся скопировать файл ~/feathers, будет отказано в доступе. Дадим владельцу файла ~/feathers право на чтение. Лишим владельца каталога ~/play права на выполнение. Перейдем в каталог ~/play. Отказано в доступе Дадим владельцу каталога ~/play право на выполнение.

```
yaandreeva@yaandreeva:~  
yaandreeva@yaandreeva:/etc  
yaandreeva@yaandreeva:~$ cp feathers file.old  
yaandreeva@yaandreeva:~$ mv file.old play  
yaandreeva@yaandreeva:~$ ls play  
file.old  
yaandreeva@yaandreeva:~$ cp play fun  
cp: не указан -r; пропускается каталог 'play'  
yaandreeva@yaandreeva:~$ cp -r play fun  
yaandreeva@yaandreeva:~$ ls fun  
play  
yaandreeva@yaandreeva:~$ mv fun play  
yaandreeva@yaandreeva:~$ cd play  
yaandreeva@yaandreeva:~/play$ ls  
file.old  fun  
yaandreeva@yaandreeva:~/play$ mv fun games  
yaandreeva@yaandreeva:~/play$ ls  
file.old  games  
yaandreeva@yaandreeva:~/play$ cd  
yaandreeva@yaandreeva:~$ chmod u-r feathers  
yaandreeva@yaandreeva:~$ cat feathers  
cat: feathers: Отказано в доступе  
yaandreeva@yaandreeva:~$ cp feathers  
cp: после 'feathers' пропущен операнд, задающий целевой файл  
По команде «cp --help» можно получить дополнительную информацию.  
yaandreeva@yaandreeva:~$ cp feathers play  
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе  
yaandreeva@yaandreeva:~$ chmod u+r feathers  
yaandreeva@yaandreeva:~$ chmod u-x play  
yaandreeva@yaandreeva:~$ cd play  
bash: cd: play: Отказано в доступе  
yaandreeva@yaandreeva:~$ chmod u+x play  
yaandreeva@yaandreeva:~$
```

```
yaandreeva@yaandreeva:/etc
MOUNT(8) System Admini
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 m
mount --bind|--rbind|--move olddir newdir
mount --make-[shared|slave|private|unbindable|rs
DESCRIPTION
All files accessible in a Unix system are arrange
/.. These files can be spread out over several dev
filesystem found on some device to the big file t
detach it again. The filesystem is used to contro
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 c
```

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

Подключает файловую систему к определенной точке в файловом дереве.


```
yaandreeva@yaandreeva:~ — man fsck
yaandreeva@yaandreeva:/etc yaandreeva@yaandreeva:~ — man fsck
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 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)
```

Проверяет и восстанавливает целостность файловой системы.

```
yaandreeva@yaandreeva:/etc
MKFS(8)
NAME
    mkfs - build a Linux filesystem
SYNOPSIS
    mkfs [options] [-t type] [fs-options]
DESCRIPTION
    This mkfs frontend is deprecated in favor of mkfs.type.

    mkfs is used to build a Linux file system. The first argument is either the device name or the size of the filesystem. The size argument can be in the form of size or size units.

    The exit status returned by mkfs is 0 if successful, non-zero otherwise.

    In actuality, mkfs is simply a front-end to the mkfs.type programs available under Linux. The filesystem type is specified by the -t option setting only. Please see the files mkfs.type for more details.
OPTIONS
    -t, --type type
        Specify the type of filesystem to be created. (currently ext2) is used.

    fs-options
        Filesystem-specific options to the mkfs.type program.

    -V, --verbose
        Verbose output.

Manual page mkfs(8) line 1 (press h for help)
```

Создает новую файловую систему на разделе или диске.
Завершает процессы по их идентификатору.

```
yaandreeva@yaandreeva:/etc
KILL(1) User O

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a]
    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to
    the process. If no signal is specified, the TERM signal is
    used to terminate the process. This signal should be
    used since a process may install a handler for the signal
    before terminating in an orderly fashion. If the signal
    has already been sent, then the KILL signal may be used;
    however, it does not give the target process the opportunity
    to clean up.

    Most modern shells have a builtin kill command
    described here. The --all, --pid, and --queue
    options, and the command name, are local extensions.

    If signal is 0, then no actual signal is sent.

ARGUMENTS
    The list of processes to be signaled can be a
    list of pids.

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

5 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера.

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).

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.

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

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.

configfs - 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 простран-

стве) — 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. / — root каталог. Содержит в себе всю иерархию системы;

/bin — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps);

/boot — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz);

/dev — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;

/etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;

/home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;

/lib — содержит системные библиотеки, с которыми работают программы и модули ядра;

/lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;

/media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;

/mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;

/opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);

/proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;

/root — директория, которая содержит файлы и личные настройки 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? Монтирование тома.

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:

1. Один блок адресуется несколькими `inode` (принадлежит нескольким файлам).
2. Блок помечен как свободный, но в то же время занят (на него ссылается `inode`).
3. Блок помечен как занятый, но в то же время свободен (ни один `inode` на него не ссылается).
4. Неправильное число ссылок в `inode` (недостаток или избыток ссылающихся записей в каталогах).
5. Несовпадение между размером файла и суммарным размером адресуемых `inode` блоков.
6. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
7. “Потерянные” файлы (правильные `inode`, на которые не ссылаются записи каталогов).
8. Недопустимые или неразмещенные номера `inode` в записях каталогов.

5. Как создаётся файловая система?

`mkfs` - позволяет создать файловую систему Linux. Создать файловую систему linux, семейства ext, на устройстве можно с помощью команды `mkfs`. Ее синтаксис выглядит следующим образом:

```
sudo mkfs -t тип устройства
```

Доступны дополнительные параметры:

c - проверить устройство на наличие битых секторов b - размер блока файловой системы j - использовать журналирование для ext3 L - задать метку раздела v - показать подробную информацию о процессе работы V - версия программы

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 сору, и она делает именно то, что предполагает ее название: она копирует. cp используется для копирования файлов из одного местоположения в другое. cp также можно использовать для копирования всех каталогов в новое место. Можно использовать эту команду 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? Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами.

Каждый файл можно изменять по трём параметра доступа. Вот они:

Чтение - разрешает получать содержимое файла, но на запись нет. 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 - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу. Чтобы получить доступ к файлам в Linux, используются разрешения. Эти разрешения назначаются трем объектам: файлу, группе и другому объекту (то есть всем остальным). Изменить права доступа можно при помощи команды `chmod`:

`chmod`

6 Выводы

Ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрели практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

7 Список литературы