

Лабораторная работа №5

Операционные системы

Оширова Юлия Николаевна, НКАбд-02-22

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	8
4 Выполнение лабораторной работы	9
5 Выводы	18
6 Контрольные вопросы	19
Список литературы	23

Список иллюстраций

4.1 Команда cp	9
4.2 Команда cp	9
4.3 Команда mv	9
4.4 Команда mv	10
4.5 Команда chmod	10
4.6 Команда chmod	10
4.7 Изменение имени файла io.h на equipment	11
4.8 Создание каталога, перемещение файла в каталог, изменение имени файла	11
4.9 Создание и копирование файла, перемещение каталога	12
4.10 Создание, перемещение и изменение имени каталога	12
4.11 Команда chmod	12
4.12 Команда chmod	13
4.13 Просмотр содержимого файла	13
4.14 Копирование и перемещение файла, копирование каталога в другой каталог	14
4.15 Перемещение и изменение имени каталога	14
4.16 Команда cat и chmod	14
4.17 Переход в каталог play, возвращение права на выполнение владельцу каталога	15
4.18 Команда man mount	15
4.19 Команда mount	16
4.20 Команда man fsck	16
4.21 Команда man mkfs	17
4.22 Команда man kill	17
4.23 Команда kill -l	17

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

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

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

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

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

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

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

2.8. Создайте и переместите каталог ~/newdir в каталог ~/ski.plases и назовите его plans. 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. Проделайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:

4.1. Просмотрите содержимое файла /etc/password.

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

таетесь 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. Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры.

3 Теоретическое введение

Файловая система (ФС) — архитектура хранения данных, которые могут находиться в разделах жесткого диска и ОП. Выдает пользователю доступ к конфигурации ядра. Определяет, какую структуру принимают файлы в каждом из разделов, создает правила для их генерации, а также управляет файлами в соответствии с особенностями каждой конкретной ФС [1]. Основные файловые системы, используемые в дистрибутивах Linux: Ext2; Ext3; Ext4; JFS; ReiserFS; XFS; Btrfs; ZFS. Ext2, Ext3, Ext4 или Extended Filesystem – стандартная файловая система, первоначально разработанная еще для Minix [2].

4 Выполнение лабораторной работы

№1 Выполним все примеры, приведённые в первой части описания лабораторной работы:

Рис. 4.1: Команда сп

```
yoshirova@yoshirova-VirtualBox:~/monthly$ cd  
yoshirova@yoshirova-VirtualBox: $ cp monthly/may monthly/june  
yoshirova@yoshirova-VirtualBox: $ ls monthly  
april june may  
yoshirova@yoshirova-VirtualBox: $ mkdir monthly.00  
yoshirova@yoshirova-VirtualBox: $ cp -r monthly monthly.00  
yoshirova@yoshirova-VirtualBox: $ cp -r monthly.00 /tmp  
yoshirova@yoshirova-VirtualBox: $
```

Рис. 4.2: Команда сп

```
ynoshirova@ynoshirova-VirtualBox:~$ cd  
ynoshirova@ynoshirova-VirtualBox:~$ mv april july  
ynoshirova@ynoshirova-VirtualBox:~$ mv july monthly.00  
ynoshirova@ynoshirova-VirtualBox:~$ ls monthly.00  
july monthly  
ynoshirova@ynoshirova-VirtualBox:~$
```

Рис. 4.3: Команда mv

```
ynoshirova@ynoshirova-VirtualBox: $ mv monthly.00 monthly.01
ynoshirova@ynoshirova-VirtualBox: $ mkdir reports
ynoshirova@ynoshirova-VirtualBox: $ mv monthly.01 reports
ynoshirova@ynoshirova-VirtualBox: $ mv reports/monthly.01 reports/monthly
ynoshirova@ynoshirova-VirtualBox: $ cd
ynoshirova@ynoshirova-VirtualBox: $ touch may
```

Рис. 4.4: Команда mv

```
ynoshirova@ynoshirova-VirtualBox: ~
ynoshirova@ynoshirova-VirtualBox: $ touch may
ynoshirova@ynoshirova-VirtualBox: $ ls -l may
-rw-rw-r-- 1 ynoshirova ynoshirova 0 мар 9 15:19 may
ynoshirova@ynoshirova-VirtualBox: $ chmod u+x may
ynoshirova@ynoshirova-VirtualBox: $ ls -l may
-rwxrwxr-- 1 ynoshirova ynoshirova 0 мар 9 15:19 may
ynoshirova@ynoshirova-VirtualBox: $ chmod u-x may
ynoshirova@ynoshirova-VirtualBox: $ ls -l may
-rw-rw-r-- 1 ynoshirova ynoshirova 0 мар 9 15:19 may
ynoshirova@ynoshirova-VirtualBox: $ rm
```

Рис. 4.5: Команда chmod

```
по команде chmod - нет, что можно полу чить дополнительную информацию:
ynoshirova@ynoshirova-VirtualBox: ~$ chmod g-r monthly
ynoshirova@ynoshirova-VirtualBox: ~$ chmod o-r monthly
ynoshirova@ynoshirova-VirtualBox: ~$ ls -l monthly
ИТОГО 0
-rw-rw-r-- 1 ynoshirova ynoshirova 0 мар 9 15:02 april
-rw-rw-r-- 1 ynoshirova ynoshirova 0 мар 9 15:06 june
-rw-rw-r-- 1 ynoshirova ynoshirova 0 мар 9 15:02 may
ynoshirova@ynoshirova-VirtualBox: ~$ touch abc1
ynoshirova@ynoshirova-VirtualBox: ~$ chmod g+rw abc1
ynoshirova@ynoshirova-VirtualBox: ~$ ls -l abc1
-rw-rw-r-- 1 ynoshirova ynoshirova 0 мар 9 15:33 abc1
```

Рис. 4.6: Команда chmod

№2 Выполним 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:

Рис. 4.7: Изменение имени файла io.h на equipment

2.2. В домашнем каталоге создадим директорию `~/ski.plases`. 2.3. Переместим файл `equipment` в каталог `~/ski.plases`. 2.4. Переименуем файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`:

Рис. 4.8: Создание каталога, перемещение файла в каталог, изменение имени файла

2.5. Создадим в домашнем каталоге файл abc1 и скопируйте его в каталог ~ski.plases, назовите его equiplist2. 2.6. Создадим каталог с именем equipment в каталоге ~ski.plases. 2.7. Переместим файлы ~ski.plases/equiplist и equiplist2 в каталог ~ski.plases/equipment:

```
Рабочий стол
WhatsApp
yoshirova@yoshirova-VirtualBox: ~$ mv equipment ski.plases
yoshirova@yoshirova-VirtualBox: ~$ cd ski.plases
yoshirova@yoshirova-VirtualBox: /ski.plases$ ls
equipment
yoshirova@yoshirova-VirtualBox: /ski.plases$ mv ski.plases/equipment ski.plases/equipment
mv: не удалось выполнить stat для "ski.plases/equipment": Нет такого файла или каталога
yoshirova@yoshirova-VirtualBox: /ski.plases$ cd
yoshirova@yoshirova-VirtualBox: ~$ mv ski.plases/equipment ski.plases/equiplist
yoshirova@yoshirova-VirtualBox: ~$ cd ski.plases
yoshirova@yoshirova-VirtualBox: /ski.plases$ ls
equiplist
yoshirova@yoshirova-VirtualBox: /ski.plases$ cd
yoshirova@yoshirova-VirtualBox: $ mkdir abc1
mkdir: невозможно создать каталог «abc1»: О的那种 существует
yoshirova@yoshirova-VirtualBox: $ mv abc1 ski.plases
yoshirova@yoshirova-VirtualBox: $ cd ski.plases
yoshirova@yoshirova-VirtualBox: /ski.plases$ ls
abc1 equiplist
yoshirova@yoshirova-VirtualBox: /ski.plases$ mv abc1 equiplist2
yoshirova@yoshirova-VirtualBox: /ski.plases$ ls
equiplist equiplist2
yoshirova@yoshirova-VirtualBox: /ski.plases$ mkdir equipment
yoshirova@yoshirova-VirtualBox: /ski.plases$ ls
equiplist equiplist2 equipment
yoshirova@yoshirova-VirtualBox: /ski.plases$ cd
yoshirova@yoshirova-VirtualBox: $ mv ski.plases/equiplist ski.plases/equipment
yoshirova@yoshirova-VirtualBox: $ mv ski.plases/equiplist2 ski.plases/equipment
yoshirova@yoshirova-VirtualBox: $ cd ski.plases/equipment
yoshirova@yoshirova-VirtualBox: /ski.plases/equipment$ ls
equiplist equiplist2
yoshirova@yoshirova-VirtualBox: /ski.plases/equipment$
```



Рис. 4.9: Создание и копирование файла, перемещение каталога

2.8. Создадим и переместим каталог `~/newdir` в каталог `~/ski.plases` и назовите его `plans`:

```
ynoshirova@ynoshirova-VirtualBox: $ mv newdir ski.plases
ynoshirova@ynoshirova-VirtualBox: $ cd ski.plases
ynoshirova@ynoshirova-VirtualBox: ~/ski.plases$ ls
equipment newdir
ynoshirova@ynoshirova-VirtualBox: ~/ski.plases$ mv newdir plans
ynoshirova@ynoshirova-VirtualBox: ~/ski.plases$ ls
equipment plans
```



Рис. 4.10: Создание, перемещение и изменение имени каталога

№3 Определим 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 создадим нужные файлы:



Рис. 4.11: Команда chmod

```
drwxr-xr-x 3 yoshirova yoshirova 4096 фев 25 18:14 Загрузки
drwxr-xr-x 2 yoshirova yoshirova 4096 ноя 6 01:18 Изображения
drwxr-xr-x 2 yoshirova yoshirova 4096 ноя 6 01:18 Музыка
drwxr-xr-x 2 yoshirova yoshirova 4096 ноя 6 01:18 Общедоступные
drwxr-xr-x 2 yoshirova yoshirova 4096 ноя 6 01:18 'Рабочий стол'
drwxr-xr-x 2 yoshirova yoshirova 4096 ноя 6 01:18 Маблони
yoshirova@yoshirova-VirtualBox: ~ chmod g+r,or australia
yoshirova@yoshirova-VirtualBox: ~ ls -l
 всего 112
drwxr-xr-- 2 yoshirova yoshirova 4096 мар 9 16:19 australia
drwxrwxr-x 2 yoshirova yoshirova 4096 фев 25 00:31 bin
-rw-rw-r-- 1 yoshirova yoshirova 0 мар 9 16:20 feathers
drwx-wx--- 2 yoshirova yoshirova 0 мар 9 15:19 may
drwxr-xr-- 1 yoshirova yoshirova 0 мар 9 16:20 monthly
drwxr-xr-- 2 yoshirova yoshirova 0 мар 9 16:20 my_os
drwxr-xr-- 2 yoshirova yoshirova 4096 мар 9 16:20 play
drwxrwxr-x 3 yoshirova yoshirova 4096 мар 9 15:16 reports
drwxrwxr-x 4 yoshirova yoshirova 4096 мар 9 16:09 skiplases
drwx----- 5 yoshirova yoshirova 4096 фев 16 11:47 snap
-rw-r--r-- 1 yoshirova yoshirova 46493 фев 15 21:01 'udo sysctl kernel.dmesg_restrict=0 kernel.dmesg_restrict = 0$udo s
el,dmesg_restrict=0 kernel,dmesg_restrict = 0'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0
B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0B'$\033'0
drwxrwxr-x 5 yoshirova yoshirova 4096 фев 25 14:14 work
drwxr-xr- 2 yoshirova yoshirova 4096 ноя 6 01:18 Видео
drwxr-xr- 2 yoshirova yoshirova 4096 ноя 6 01:18 Документы
drwxr-xr- 2 yoshirova yoshirova 4096 ноя 6 01:18 Загрузки
drwxr-xr- 2 yoshirova yoshirova 4096 фев 25 18:14 Изображения
drwxr-xr- 2 yoshirova yoshirova 4096 ноя 6 01:18 Музыка
drwxr-xr- 2 yoshirova yoshirova 4096 ноя 6 01:18 Общедоступные
drwxr-xr- 2 yoshirova yoshirova 4096 ноя 6 01:18 'Рабочий стол'
drwxr-xr- 2 yoshirova yoshirova 4096 ноя 6 01:18 Маблони
yoshirova@yoshirova-VirtualBox: ~ chmod g+r,or australia
yoshirova@yoshirova-VirtualBox: ~ chmod u+w,g+r,o+r play
yoshirova@yoshirova-VirtualBox: ~ chmod u+w,u+x,g-
```

Рис. 4.12: Команда chmod

№4 Проделаем приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды: 4.1. Просмотрим содержимое файла /etc/password:

```
gnashirova@gnashirova-VirtualBox: ~ $ cd /etc  
gnashirova@gnashirova-VirtualBox: ~ $ ls  
acpi      debconf.conf    hosts      magic.mime    profile.d      sudoers  
adduser.conf  debian_version  hosts.allow  mailcap        protocols  sudoers.d  
alsa       default        hosts.deny   mailcap.order  pulse      sudo.logrwd.conf  
alternatives  deluser.conf   hp          nanopath.config  python3  sysctl.conf  
anacrontab  depmod.d      ifplugd     nc          python3.10  sysctl.d  
apache2    dhcp          init       mime.types    rc0.d      systemd  
app.conf   dictionary-common intd      nke2fs.rcf    rc1.d      terminal  
apm       dplg          intransf-tools  nodemanager  rc2.d      thermalid  
apparmor  e2fsckub.conf  iptrafnets-tools  nodeadm     rc3.d      thunderbird  
armor     ftrace        iuserv.conf.d  modules      rc4.d      timezone  
apport    environment  ipi-usb     modules-load.d  rc5.d      tmpfiles.d  
appstream.conf  environment.d  ionroute2   ntbtab      rc6.d      ubuntu-advantage  
apt       ethertypes    issue      nororc      rc7.d      ucf.conf  
avahi     firefox       issue.net   netconfig    resolv.conf  udev  
bash      fonts         kernel     netplan      rmt      udisks2  
bash_completion  fprintd.conf  kernel-ing.conf  network     rpc      ufw  
bash_completion.d  fstab       kernelloops.conf  networkd-dispatcher  rsyslog.conf  
bindsproxy.blacklist  fuse.conf  ldap       NetworkManager  rsyslog.d  update-manager  
bindmnt.d     fwupd       ld.so.cache  networks      rygel.conf  update-motd.d  
blueooth   gat.conf     ld.so.conf   neutrino     sane.d      update-notifier  
brlapi.key  git          legal      nf_tables.conf  nftables.conf  UPower  
bzr       gdm3         libao.conf  nftunnel.conf  nslookup.conf  sensors  
britty.conf  geoclue      libbaudit  opt          nftunnel     sensors3  
ca-certificates  ghostscript  libblkdev   os-release   nslookup     services  
ca-certificates.conf  glvnd      liblbbadit  pam.conf    sgml        shadow  
ca-certificates.conf.dpkg-old  gnome     liblbbadit  pam.d      shadow-        shadow  
chatscripts  graff       libpaper.d  PackageKit  sgml        shadow-  
console-setup  orolin      libreoffice  pam.conf    shadow-  shadow
```

Рис. 4.13: Просмотр содержимого файла

4.2. Скопируем файл ~/feathers в файл ~/file.old. 4.3. Переместим файл ~/file.old в каталог ~/play. 4.4. Скопируем каталог ~/play в каталог ~/fun:

```
еро не указаны, пропускается каталог feathers
ynoshirova@ynoshirova-VirtualBox: $ cp -r feathers file.old
ynoshirova@ynoshirova-VirtualBox: $ mv file.old play
ynoshirova@ynoshirova-VirtualBox: $ ls play
file.old
ynoshirova@ynoshirova-VirtualBox: $ cp -r play fun
ynoshirova@ynoshirova-VirtualBox: $ ls fun
file.old
ynoshirova@ynoshirova-VirtualBox: $ cp -r play fun
ynoshirova@ynoshirova-VirtualBox: $ ls fun
file.old play
ynoshirova@ynoshirova-VirtualBox: $
```

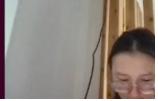


Рис. 4.14: Копирование и перемещение файла, копирование каталога в другой каталог

4.5. Переместим каталог ~/fun в каталог ~/play и назовем его games:

```
ynoshirova@ynoshirova-VirtualBox:~/play$ mv fun games
ynoshirova@ynoshirova-VirtualBox:~/play$ ls
file.old games
ynoshirova@ynoshirova-VirtualBox:~/play$
```

Рис. 4.15: Перемещение и изменение имени каталога

4.6. Лишим 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. Дадим владельцу файла ~/feathers право на чтение. 4.10. Лишим владельца каталога ~/play права на выполнение:

```
ynoshirova@ynoshirova-VirtualBox:~$ cat feathers
cat: feathers: Отказано в доступе
ynoshirova@ynoshirova-VirtualBox:~$ cp feathers
cp: после 'feathers' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
ynoshirova@ynoshirova-VirtualBox:~$
```

Рис. 4.16: Команда cat и chmod

4.11. Перейдем в каталог ~/play. Что произошло? 4.12. Дадим владельцу каталога ~/play право на выполнение:

Рис. 4.17: Переход в каталог play, возвращение права на выполнение владельцу каталога

№5 Прочитаем тап по командам mount, fsck, mkfs, kill:

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 t
-t type is optional. The **mount** command is usually able to detect a filesystem. The root per
mount a filesystem by default. See section "Non-superuser mounts" below for more details. T



Рис. 4.18: Команда man mount

```
yoshiro@yoshiro-VirtualBox: ~
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relative_size=1797189,nr_inodes=492547,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relative,gid=5,mode=620,ptmxnode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relative,size=401792k,mode=755,inode64)
/dev/sda3 on / type ext4 (rw,relative,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
firmware on /lib/firmware type firmware (ro)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relative,size=128k,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relative,_nsdelegate,_memory_recursiveprot)
pscstore on /sys/fs/pscstore type pstore (rw,nosuid,nodev,noexec,relative)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relative,mode=700)
systemd-1 on /proc/sys/vb/inlmt_mis type autos (rw,relative,fd=29,grpr=0,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=15948)
nqueue on /dev/nqueue type mqueue (rw,nosuid,nodev,noexec,relative)
budgetfs on /dev/hugepages type hugepages (rw,relative,pagesize=2M)
tracefs on /sys/kernel/debug/tracing type tracefs (rw,nosuid,nodev,noexec,relative)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relative)
fusectl on /sys/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relative)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relative)
ramfs on /run/credentials/systemd-susers.service type ramfs (ro,nosuid,nodev,noexec,relative,mode=700)
[...]
/var/lib/snaps/snapd-base_5.snap on /snap/base/5 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
/var/lib/snaps/snapc_2697.snap on /snap/core18/2697 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
/var/lib/snaps/snapd-core_2697.snap on /snap/core18/2697 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
[...]
/var/lib/snaps/snapf_2356.snap on /snap/firefox/2356 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
[...]
/var/lib/snaps/snapf_2391.snap on /snap/firefox/2391 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
[...]
/var/lib/snaps/snapf_2356.snap on /snap/firefox/2356 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
[...]
/var/lib/snaps/snapf_592.snap on /snap/gb/592 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
[...]
/var/lib/snaps/snapn_gnome-3-38-2004_112.snap on /snap/gnome-3-38-2004/112 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
[...]
/var/lib/snaps/snapn_gnome-3-38-2004_119.snap on /snap/gnome-3-38-2004/119 type squashfs (ro,nodev,relative,errors=continue,x-gdu.hide)
[...]
/var/lib/snaps/snaps/atk-common-themes_1535.snap on /snap/atk-common-themes/1535 type squashfs (ro,relative,errors=continue,x-gdu.hide)
```

Рис. 4.19: Команда mount

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=8866abf6-86c5-4a83-98b8-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)

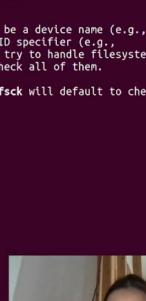


Рис. 4.20: Команда man fsck

```

NAME
    mkfs - build a Linux filesystem

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

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

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

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

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

OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.

    fs-options
        Filesystem-specific options to be passed to the real filesystem builder.

    -v, --verbose
        Produce verbose output, including all filesystem-specific commands that are executed. S
        than once inhibits execution of any filesystem-specific commands. This is really only u

    -V, --version
        Display version information and exit. (Option -V will display version information only

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

```

Рис. 4.21: Команда man mkfs

```

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals include HUP,
    INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL. Negative PID
    values may be used to choose whole process groups; see the PGID column in ps command output. A PID of -1 is special; it
    indicates all processes except the kill process itself and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is
        explained in signal(7) manual page.

    -q, --queue value
        Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer to be sent with the signal.
        If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(2),
        then it can obtain this data via the si_value field of the siginfo_t structure.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal n
        way round.

    -L, --table
Manual page kill(1) line 1 (press h for help or q to quit)

```

Рис. 4.22: Команда man kill

```

ynoshirova@ynoshirova-VirtualBox:~$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGKCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX

```

Рис. 4.23: Команда kill -l

5 Выводы

Ознакомилась с файловой системой Linux и с ее структурой. Научилась использовать различные команды в терминале для работы с файлами и каталогами.

6 Контрольные вопросы

Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу. Ext2, Ext3, Ext4 или Extended Filesystem - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзабайта. Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера не лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux. 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. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе? Монтирование тома. 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 mode (при надлежит 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). Недопустимые или неразмещенные номера inode в записях каталогов. Как создаётся файловая система? mkfs - позволяет создать файловую систему Linux. Дайте характеристику командам для просмотра текстовых файлов. Cat - выводит содержимое файла на стандартное устройство вывода Приведите основные возможности команды cp в Linux. Cp – копирует или перемещает директорию, файлы. Приведите основные возможности команды mv в Linux. Mv - переименовать или переместить файл или директорию Что такое права доступа? Как они могут быть изменены? Права доступа к файлу или каталогу можно из-

менить, воспользовавшись командой chmod. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

Список литературы

1. Структура и типы файловых систем в Linux Электронный ресурс
2. Типы файловых систем, их предназначение и отличия Электронный ресурс