

Лабораторная работа №5

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

Салькова Кристина Михайловна

Содержание

1	Цель работы.....	3
2	Задание	3
3	Выполнение лабораторной работы.....	4
4	Выводы	12
	Список литературы	12

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

3 Выполнение лабораторной работы

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

```
kmsaljkova@dk8n74 ~ $ ls /etc/passwd
ls: невозможно получить доступ к '/etc/passwd': Нет такого файла или каталога
kmsaljkova@dk8n74 ~ $ ls
abc1      file.old  my_os     reports   t.cpp.save  work      Загрузки      'Рабочий ст
ол'
australia file.old.  play      ski.plases temp        'без имени'  Изображения  Шаблоны
c++       may       public    t         tmp         Видео        Музыка        '
feathers  monthly  public_html t.cpp     t.o         Документы   Общедоступные
kmsaljkova@dk8n74 ~ $ ls /
afs  boot  dev  home  lib64  media  net  proc  run  srv  tmp  var
bin  com  etc  lib  lost+found  mnt  opt  root  sbin  sys  usr
kmsaljkova@dk8n74 ~ $ cp ~/feathers ~/file.old
kmsaljkova@dk8n74 ~ $ mv ~/file.old ~/play
kmsaljkova@dk8n74 ~ $ cp -r ~/play ~/fun
kmsaljkova@dk8n74 ~ $ mv ~/fun ~/play
kmsaljkova@dk8n74 ~ $ mv ~/fun/play ~/fun/games
mv: не удалось выполнить stat для '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/fun/play': Нет такого фай
ла или каталога
kmsaljkova@dk8n74 ~ $ mv ~/fun/play ~/fun/games
mv: не удалось выполнить stat для '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/fun/play': Нет такого фай
ла или каталога
kmsaljkova@dk8n74 ~ $ mv ~/play/fun ~/play/fun
mv: невозможно перенести '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/play/fun' в свой собственный подка
талог, '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/play/fun/fun'
kmsaljkova@dk8n74 ~ $ cd play
kmsaljkova@dk8n74 ~/play $ mv fun games
kmsaljkova@dk8n74 ~/play $ ls
file.old  games
```

Figure 1: выполнение команд

```

kmsaljkova@dk8n74 ~ $ ls /etc/passwd
ls: невозможно получить доступ к '/etc/passwd': Нет такого файла или каталога
kmsaljkova@dk8n74 ~ $ ls
abc1      file.old  my_os     reports   t.cpp.save  work      Загрузки      'Рабочий ст
ол'
australia file.old.  play      ski.plases temp        'без имени'  Изображения  Шаблоны
c++       may       public    t         tmp         Видео        Музыка        'Рабочий ст
ол'
feathers   monthly   public_html t.cpp      t.o        Документы    Общедоступные
kmsaljkova@dk8n74 ~ $ ls /
afs  boot  dev  home  lib64  media  net  proc  run  srv  tmp  var
bin  com   etc  lib   lost+found mnt  opt  root  sbin sys  usr
kmsaljkova@dk8n74 ~ $ cp ~/feathers ~/file.old
kmsaljkova@dk8n74 ~ $ mv ~/file.old ~/play
kmsaljkova@dk8n74 ~ $ cp -r ~/play ~/fun
kmsaljkova@dk8n74 ~ $ mv ~/fun ~/play
kmsaljkova@dk8n74 ~ $ mv ~/fun/play ~/fun/games
mv: не удалось выполнить stat для '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/fun/play': Нет такого фай
ла или каталога
kmsaljkova@dk8n74 ~ $ mv ~/fun/play ~/fun/games
mv: не удалось выполнить stat для '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/fun/play': Нет такого фай
ла или каталога
kmsaljkova@dk8n74 ~ $ mv ~/play/fun ~/play/fun
mv: невозможно перенести '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/play/fun' в свой собственный подка
талог, '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/play/fun/fun'
kmsaljkova@dk8n74 ~ $ cd play
kmsaljkova@dk8n74 ~/play $ mv fun games
kmsaljkova@dk8n74 ~/play $ ls
file.old  games

```

Figure 2: выполнение команд

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.plases. 2.3. Переместите файл equipment в каталог ~/ski.plases. 2.4. Переименуйте файл ~/ski.plases/equipment в ~/ski.plases/equiplist.

```

kmsaljkova@dk8n64 ~ $ ls -l abc1
-rw-rw-r-- 1 kmsaljkova studsci 0 map  9 14:55 abc1
kmsaljkova@dk8n64 ~ $ cp /usr/include/sys/io.h equipment
kmsaljkova@dk8n64 ~ $ ls
abc1      may       public_html  t.cpp      tmp        'без имени'  Загрузки      Общедоступные
c++       monthly   reports      t.cpp.save t.o        Видео        Изображения  'Рабочий стол'
equipment public    t            temp       work       Документы    Музыка        Шаблоны
kmsaljkova@dk8n64 ~ $ mkdir ~/ski.plases
kmsaljkova@dk8n64 ~ $ ls
abc1      may       public_html  t        temp       work       Документы    Музыка        Шаблоны
c++       monthly   reports      t.cpp    tmp        'без имени'  Загрузки      Общедоступные
equipment public    ski.plases   t.cpp.save t.o        Видео        Изображения  'Рабочий стол'
kmsaljkova@dk8n64 ~ $ mv equipment ^C
kmsaljkova@dk8n64 ~ $ mv equipment ski.plases
mv: не удалось выполнить stat для 'equiomenr': Нет такого файла или каталога
kmsaljkova@dk8n64 ~ $ mv equipment ski.plases
mv: не удалось выполнить stat для 'equioment': Нет такого файла или каталога
kmsaljkova@dk8n64 ~ $ mv ~/ski.plases/equipment ~/ski.plases/equiplist.
kmsaljkova@dk8n64 ~ $ ls ski.plases
equiplist.
kmsaljkova@dk8n64 ~ $ mv ~/ski.plases/equiplist. ~/ski.plases/equiplist
kmsaljkova@dk8n64 ~ $ ls ski.plases
equiplist

```

Figure 3: выполнение команд

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.

```
kmsaljkova@dk8n64 ~ $ touch abc1
kmsaljkova@dk8n64 ~ $ cp ~/ski.plases equiplist2
cp: не указан -r; пропускается каталог '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/ski.plases'
kmsaljkova@dk8n64 ~ $ cd ~/ski.plases
kmsaljkova@dk8n64 ~/ski.plases $ cp abc1 equiplist2
cp: не удалось выполнить stat для 'abc1': Нет такого файла или каталога
kmsaljkova@dk8n64 ~/ski.plases $ cp ~/abc1 equiplist2
kmsaljkova@dk8n64 ~/ski.plases $ ls
equiplist equiplist2
kmsaljkova@dk8n64 ~/ski.plases $ mkdir equipment
kmsaljkova@dk8n64 ~/ski.plases $ ls
equiplist equiplist2 equipment
kmsaljkova@dk8n64 ~/ski.plases $ mv equiplist equiplist2 equipment
kmsaljkova@dk8n64 ~/ski.plases $ ls ^C
kmsaljkova@dk8n64 ~/ski.plases $ ls equipment
equiplist equiplist2
kmsaljkova@dk8n64 ~/ski.plases $ mdir ~/newdir
bash: mdir: команда не найдена
kmsaljkova@dk8n64 ~/ski.plases $ mkdir ~/newdir
kmsaljkova@dk8n64 ~/ski.plases $ mv ^C
kmsaljkova@dk8n64 ~/ski.plases $ mv ^[[200~/newdir
mv: после '$'\033''[200~/newdir' пропущен операнд, заданный целевой файл
По команде «mv --help» можно получить дополнительную информацию.
kmsaljkova@dk8n64 ~/ski.plases $ mv ~/newdir plans
kmsaljkova@dk8n64 ~/ski.plases $ ls
equipment plans
kmsaljkova@dk8n64 ~/ski.plases $
```

Figure 4: выполнение команд

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

```
kmsaljkova@dk8n64 ~ $ chmod g-x australia
kmsaljkova@dk8n64 ~ $ chmod o-x australia
kmsaljkova@dk8n64 ~ $ ls -l australia
итого 0
kmsaljkova@dk8n64 ~ $ ls -l
итого 67
-rw-rw-r-- 1 kmsaljkova studsci 0 мар 9 15:06 abc1
drwxr--r-- 2 kmsaljkova studsci 2048 мар 9 15:27 australia
```

Figure 5: результат

3.2. drwx-x-x ... play

```

kmsaljkova@dk8n64 ~ $ chmod o-r play
kmsaljkova@dk8n64 ~ $ ls -l
итого 69
-rw-rw-r-- 1 kmsaljkova studsci  0 мар 9 15:06 abc1
drwx---r-- 2 kmsaljkova studsci 2048 мар 9 15:27 australia
drwxr-xr-x 2 kmsaljkova studsci 2048 окт 6 16:12 c++
-rw-r--r-- 1 kmsaljkova studsci  0 мар 9 14:50 may
drwxr-xr-x 2 kmsaljkova studsci 2048 мар 9 14:46 monthly
drwx--x--x 2 kmsaljkova studsci 2048 мар 9 15:31 play

```

Figure 6: результат

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

```

kmsaljkova@dk8n64 ~ $ chmod u-w my_os
kmsaljkova@dk8n64 ~ $ chmod u+x my_os
kmsaljkova@dk8n64 ~ $ ls -l
итого 69
-rw-rw-r-- 1 kmsaljkova studsci  0 мар 9 15:06 abc1
drwx---r-- 2 kmsaljkova studsci 2048 мар 9 15:27 australia
drwxr-xr-x 2 kmsaljkova studsci 2048 окт 6 16:12 c++
-rw-r--r-- 1 kmsaljkova studsci  0 мар 9 14:50 may
drwxr-xr-x 2 kmsaljkova studsci 2048 мар 9 14:46 monthly
-r-xr--r-- 1 kmsaljkova studsci  0 мар 9 15:38 my_os

```

Figure 7: результат

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

```

kmsaljkova@dk8n64 ~ $ chmod g+w feathers
kmsaljkova@dk8n64 ~ $ ls -l
итого 69
-rw-rw-r-- 1 kmsaljkova studsci  0 мар 9 15:06 abc1
drwx---r-- 2 kmsaljkova studsci 2048 мар 9 15:27 australia
drwxr-xr-x 2 kmsaljkova studsci 2048 окт 6 16:12 c++
-rw-rw-r-- 1 kmsaljkova studsci  0 мар 9 15:40 feathers

```

Figure 8: результат

4. Прodelайте 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.


```

kmsaljkova@dk8n74 ~ $ ls /etc/password
ls: невозможно получить доступ к '/etc/password': Нет такого файла или каталога
kmsaljkova@dk8n74 ~ $ ls
abc1      file.old  my_os     reports   t.cpp.save  work      Загрузки      'Рабочий ст
ол'
australia file.old.  play      ski.plases temp        'без имени'  Изображения  Шаблоны
c++       may       public    t         tmp         Видео        Музыка        '
feathers   monthly   public_html t.cpp     t.o         Документы   Общедоступные
kmsaljkova@dk8n74 ~ $ ls /
afs boot dev home lib64 media net proc run srv tmp var
bin com etc lib lost+found mnt opt root sbin sys usr
kmsaljkova@dk8n74 ~ $ cp ~/feathers ~/file.old
kmsaljkova@dk8n74 ~ $ mv ~/file.old ~/play
kmsaljkova@dk8n74 ~ $ cp -r ~/play ~/fun
kmsaljkova@dk8n74 ~ $ mv ~/fun ~/play
kmsaljkova@dk8n74 ~ $ mv ~/fun/play ~/fun/games
mv: не удалось выполнить stat для '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/fun/play': Нет такого фай
ла или каталога
kmsaljkova@dk8n74 ~ $ mv ~/fun/play ~/fun/games
mv: не удалось выполнить stat для '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/fun/play': Нет такого фай
ла или каталога
kmsaljkova@dk8n74 ~ $ mv ~/play/fun ~/play/fun
mv: невозможно перенести '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/play/fun' в свой собственный подка
талог, '/afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/play/fun/fun'
kmsaljkova@dk8n74 ~ $ cd play
kmsaljkova@dk8n74 ~/play $ mv fun games
kmsaljkova@dk8n74 ~/play $ ls
file.old games

```

Figure 9: выполнение команд

4.6. Лишите владельца файла ~/feathers права на чтение. 4.7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?

```

kmsaljkova@dk8n74 ~ $ cat ~/feathers
cat: /afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/feathers: Отказано в доступе

```

Figure 10: выполнение команд

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

```

kmsaljkova@dk8n74 ~ $ cp feathers
cp: после 'feathers' пропущен операнд, задающий целевой файл

```

Figure 11: выполнение команд

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

```

kmsaljkova@dk8n74 ~/play $ cd
kmsaljkova@dk8n74 ~ $ chmod u-r feathers
kmsaljkova@dk8n74 ~ $ cp feathers
cp: после 'feathers' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
kmsaljkova@dk8n74 ~ $ cat ~/feathers
cat: /afs/.dk.sci.pfu.edu.ru/home/k/m/kmsaljkova/feathers: Отказано в доступе
kmsaljkova@dk8n74 ~ $ chmod u+r feathers

```

Figure 12: выполнение команд

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

4.11. Перейдите в каталог ~/play. Что произошло? ничего не произошло(

```
kmsaljkova@dk8n74 ~ $ chmod u-x play
kmsaljkova@dk8n74 ~ $ cd play
kmsaljkova@dk8n74 ~/play $ cd
kmsaljkova@dk8n74 ~ $ chmod u-x ~/play
kmsaljkova@dk8n74 ~ $ cd ~/play
kmsaljkova@dk8n74 ~/play $ cd
kmsaljkova@dk8n74 ~ $ ls -l
итого 69
-rw-rw-r-- 1 kmsaljkova studsci    0 map  9 15:06  abc1
drwx---r-- 2 kmsaljkova studsci 2048 map  9 15:27  australia
drwxr-xr-x 2 kmsaljkova studsci 2048 окт  6 16:12  c++
-rw-rw-r-- 1 kmsaljkova studsci    0 map  9 15:40  feathers
-rw-r--r-- 1 kmsaljkova studsci    0 map 10 15:52  file.old.
-rw-r--r-- 1 kmsaljkova studsci    0 map  9 14:50  may
drwxr-xr-x 2 kmsaljkova studsci 2048 map  9 14:46  monthly
-r-xr--r-- 1 kmsaljkova studsci    0 map  9 15:38  my_os
drw---x--x 3 kmsaljkova studsci 2048 map 10 17:10  play
```

Figure 13: выполнение команд

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

```
kmsaljkova@dk8n74 ~ $ chmod u+x ~/play
kmsaljkova@dk8n74 ~ $ ls -l
итого 69
-rw-rw-r-- 1 kmsaljkova studsci    0 map  9 15:06  abc1
drwx---r-- 2 kmsaljkova studsci 2048 map  9 15:27  australia
drwxr-xr-x 2 kmsaljkova studsci 2048 окт  6 16:12  c++
-rw-rw-r-- 1 kmsaljkova studsci    0 map  9 15:40  feathers
-rw-r--r-- 1 kmsaljkova studsci    0 map 10 15:52  file.old.
-rw-r--r-- 1 kmsaljkova studsci    0 map  9 14:50  may
drwxr-xr-x 2 kmsaljkova studsci 2048 map  9 14:46  monthly
-r-xr--r-- 1 kmsaljkova studsci    0 map  9 15:38  my_os
drwx--x--x 3 kmsaljkova studsci 2048 map 10 17:10  play
```

Figure 14: выполнение команды

5. Прочитаем man по командам mount, fsck, mkfs, kill и кратко их охарактеризуем, приведя примеры.

```
kmsaljkova@dk8n74 ~ $ man mount
kmsaljkova@dk8n74 ~ $ man fsck
kmsaljkova@dk8n74 ~ $ man mkfs
kmsaljkova@dk8n74 ~ $ man kill
```

Figure 15: man

```
MOUNT(8)                                     System Administration                                     MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-ffnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable] mountpoint

DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the filesystem found on some device to the big file tree. Conversely, the umount(8) command will detach it again. The filesystem is used to control how data is stored on the device or provided in a virtual way by network or other services.

    The standard form of the mount command is:

        mount -t type device dir

    This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. The option -t type is optional. The mount command is usually able to detect a filesystem. The root permissions are necessary to mount a filesystem by default. See section "Non-superuser mounts" below for more details. The previous contents (if any) and owner and mode of dir become invisible, and as long as this filesystem remains mounted, the pathname dir refers to the root of the filesystem on device.

    If only the directory or the device is given, for example:

        mount /dir

    then mount looks for a mountpoint (and if not found then for a device) in the /etc/fstab file. It's possible to use the --target or --source options to avoid ambiguous interpretation of the given argument. For example:

        mount --target /mountpoint

    The same filesystem may be mounted more than once, and in some cases (e.g., network filesystems) the same filesystem may be mounted on the same mountpoint multiple times. The mount command does not implement any policy to control this behavior. All behavior is controlled by the kernel and it is usually specific to the filesystem driver. The exception is --all, in this case already mounted filesystems are ignored (see --all below for more details).

    Listing the mounts
    The listing mode is maintained for backward compatibility only.

    For more robust and customizable output use findmnt(8), especially in your scripts. Note that control characters in the mountpoint name are replaced with '?'.

    The following command lists all mounted filesystems (of type type):

        mount [-l] [-t type]

    The option -l adds labels to this listing. See below.

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

Figure 16: `man mount`

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

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-laAVRTMNP] [-r [fs]] [-C [fs]] [-t fstype] [filesystem...] [--] [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck program will try to handle filesystems on different physical disk drives in parallel to reduce the total amount of time needed to check all of them.

    If no filesystems are specified on the command line, and the -A option is not specified, fsck will default to checking filesystems in /etc/fstab serially. This is equivalent to the -As options.

    The exit status returned by fsck is the sum of the following conditions:

    0
    No errors

    1
    Filesystem errors corrected

    2
    System should be rebooted

    4
    Filesystem errors left uncorrected

    8
    Operational error

    16
    Usage or syntax error

    32
    Checking canceled by user request

    128
    Shared-library error

    The exit status returned when multiple filesystems are checked is the bit-wise OR of the exit statuses for each filesystem that is checked.

    In actuality, fsck is simply a front-end for the various filesystem checkers (fsck.fstype) available under Linux. The filesystem-specific checker is searched for in the PATH environment variable. If the PATH is undefined then fallback to /sbin.

    Please see the filesystem-specific checker manual pages for further details.

OPTIONS
    -l
    Create an exclusive fsck(2) lock file (/run/fsck/filesystemname.lock) for whole-disk device. This option can be used with one device only (this means that -A and -l are mutually exclusive). This option is recommended when more fsck instances are executed in the same time. The option is ignored when used for multiple devices or for non-rotating disks. fsck does not lock underlying devices when executed to check stacked devices (e.g. MD or DM) - this feature is not implemented yet.

    -r [fs]

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

Figure 17: `man 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> 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.<type>) available under Linux. The filesystem-specific builder is searched for via your PATH environment setting only. Please see the filesystem-specific builder manual pages for further details.

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

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

    -V, --verbose
        Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option more than once inhibits execution of any filesystem-specific commands. This is really only useful for testing.

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

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

BUGS
    All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific programs do not automatically detect the device size and require the size parameter to be specified.

AUTHORS
    David Engel <david@bds.com>, Fred N. van Kempen <waltje@u.walt.nl.mugnet.org>, Ron Sommeling <sommel@sci.kun.nl>.

    The manual page was shamelessly adapted from Remy Card's version for the ext2 filesystem.

SEE ALSO
    fs(5), badblocks(8), fsck(8), mkdosfs(8), mke2fs(8), mkfs.bfs(8), mkfs.ext2(8), mkfs.ext3(8), mkfs.ext4(8), mkfs.minix(8), mkfs.msdos(8), mkfs.vfat(8), mkfs.xfs(8)

REPORTING BUGS
    For bug reports, use the issue tracker at https://github.com/util-linux/util-linux/issues.

AVAILABILITY
    The mkfs command is part of the util-linux package which can be downloaded from Linux Kernel Archive <https://www.kernel.org/pub/linux/utils/util-linux/>.

util-linux 2.38.1                                2022-05-11                                MKFS(8)
Manual page mkfs(8) line 1/55 (END) (press h for help or q to quit)

```

Figure 18: *man mkfs*

```

KILL(1)                                     User Commands                                KILL(1)

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.

    -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 number to signal name, or other way round.

    -L, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.

    kill -1 11
        Translate number 11 into a signal name.

    kill -L
        List the available signal choices in a nice table.

    kill 123 543 2341 3453
        Send the default signal, SIGTERM, to all those processes.

SEE ALSO
    kill(2), killall(1), nice(1), pkill(1), renice(1), signal(7), sigqueue(3), skkill(1)

STANDARDS
    This command meets appropriate standards. The -L flag is Linux-specific.

AUTHOR
    Albert Cahalan (albert@users.sf.net) wrote kill in 1999 to replace a bsduutils one that was not standards compliant. The util-linux one might also work correctly.

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

```

Figure 19: *man kill*

4 Выводы

Мы 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.

Список литературы