

Лабораторная работа № 7

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

Лупупа Чилеше

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15
	Список литературы	16

Список иллюстраций

2.1	equipment	6
2.2	~/ski.places.	6
2.3	equipment	6
2.4	equiplist	7
2.5	каталоге ~/ski.places.	7
2.6	plans	7
2.7	chmod	8
2.8	/etc/password	9
2.9	~/file.old	9
2.10	~/play	9
2.11	fun	9
2.12	games	9
2.13	права на чтение	10
2.14	ср	10
2.15	Лишите владельца каталога ~/play	10
2.16	право на выполнение	10
2.17	fsck	11
2.18	mount	12
2.19	mkfs	13
2.20	kill	14

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 Выполнение лабораторной работы

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:
 - 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.

```
[lchileshe@lchileshe ~]$ cp /usr/include/sys/io.h equipment
[lchileshe@lchileshe ~]$ ls equipment
equipment
```

Рис. 2.1: equipment

- В домашнем каталоге создайте директорию `~/ski.places`.

```
[lchileshe@lchileshe ~]$ mkdir ski.places
```

Рис. 2.2: ~/ski.places.

- Переместите файл `equipment` в каталог `~/ski.places`.

```
[lchileshe@lchileshe ~]$ mv equipment ski.places/
[lchileshe@lchileshe ~]$ ls ski.places/
equipment
```

Рис. 2.3: equipment

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

```
[lchileshe@lchileshe ~]$ mv ski.plases/equipment ski.plases/equiplist
[lchileshe@lchileshe ~]$ ls ski.plases/
equiplist
```

Рис. 2.4: equiplist

- Создайте в домашнем каталоге файл abc1 и скопируйте его в каталог

```
[lchileshe@lchileshe ~]$ mkdir abc1
mkdir: cannot create directory 'abc1': File exists
[lchileshe@lchileshe ~]$ cp abc1 ski.plases/equiplist2
[lchileshe@lchileshe ~]$ ls ski.plases/
equiplist equiplist2
```

~/ski.plases, назовите его equiplist2.

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

```
[lchileshe@lchileshe ~]$ mkdir ski.plases/equipment
[lchileshe@lchileshe ~]$ ls ski.plases/
equiplist equiplist2 equipment
```

Рис. 2.5: каталоге ~/ski.plases.

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

~/ski.plases/equipment.

```
[lchileshe@lchileshe ~]$ mv ski.plases/equiplist ski.plases/equiplist2 ski.plases/equipment/
```

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

```
[lchileshe@lchileshe ~]$ ls newdir/
[lchileshe@lchileshe ~]$ mv newdir ski.plases/plans
[lchileshe@lchileshe ~]$ ls ski.plases/
equipment plans
```

Рис. 2.6: 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.

```
[lchileshe@lchileshe ~]$ mkdir australia
[lchileshe@lchileshe ~]$ mkdir play
[lchileshe@lchileshe ~]$ touch my_os
[lchileshe@lchileshe ~]$ touch feathers
[lchileshe@lchileshe ~]$ chmod 744 australia
[lchileshe@lchileshe ~]$ chmod 711 play
[lchileshe@lchileshe ~]$ chmod 544 my_os
[lchileshe@lchileshe ~]$ chmod 664 feathers
[lchileshe@lchileshe ~]$ ls -l
total 20
-rw-rw-r--. 1 lchileshe lchileshe  0 Mar 23 17:49 abc1
drwxr--r--. 1 lchileshe lchileshe  0 Mar 23 18:12 australia
drwxr-xr-x. 1 lchileshe lchileshe 14 Mar 16 01:14 bin
drwxr-xr-x. 1 lchileshe lchileshe  0 Feb 29 17:43 Desktop
drwxr-xr-x. 1 lchileshe lchileshe 136 Mar  1 23:52 Documents
drwxr-xr-x. 1 lchileshe lchileshe 370 Mar 16 23:21 Downloads
-rw-rw-r--. 1 lchileshe lchileshe  0 Mar 23 18:13 feathers
drwxr-xr-x. 1 lchileshe lchileshe  74 Mar  8 14:06 git-extended
-rw-r--r--. 1 lchileshe lchileshe 18657 Mar 15 23:34 LICENSE
-rw-r--r--. 1 lchileshe lchileshe  0 Mar 23 17:43 may
drwx--x--x. 1 lchileshe lchileshe  0 Mar 23 17:46 monthly
drwxr-xr-x. 1 lchileshe lchileshe  0 Feb 29 17:43 Music
-r-xr--r--. 1 lchileshe lchileshe  0 Mar 23 18:13 my_os
drwxr-xr-x. 1 lchileshe lchileshe  0 Mar 15 20:50 Pictures
drwx--x--x. 1 lchileshe lchileshe  0 Mar 23 18:12 play
drwxr-xr-x. 1 lchileshe lchileshe  0 Feb 29 17:43 Public
drwxr-xr-x. 1 lchileshe lchileshe 14 Mar 23 17:41 reports
drwxr-xr-x. 1 lchileshe lchileshe 28 Mar 23 18:11 ski.plases
drwxr-xr-x. 1 lchileshe lchileshe  0 Feb 29 17:43 Templates
drwxr-xr-x. 1 lchileshe lchileshe  0 Feb 29 17:43 Videos
drwxr-xr-x. 1 lchileshe lchileshe 66 Mar  5 23:14 work
```

Рис. 2.7: `chmod`

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

- Просмотрите содержимое файла `/etc/passwd`.

```
[lchileshe@lchileshe ~]$ ls etc/passwd  
ls: cannot access 'etc/passwd': No such file or directory
```

Рис. 2.8: `/etc/passwd`

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

```
[lchileshe@lchileshe ~]$ cp feathers file.old
```

Рис. 2.9: `~/file.old`

- Переместите файл `~/file.old` в каталог `~/play`.

```
[lchileshe@lchileshe ~]$ mv file.old play
```

Рис. 2.10: `~/play`

- Скопируйте каталог `~/play` в каталог `~/fun`.

```
[lchileshe@lchileshe ~]$ cp -r play fun
```

Рис. 2.11: `fun`

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

```
[lchileshe@lchileshe ~]$ mv fun play/games
```

Рис. 2.12: `games`

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

```
[lchileshe@lchileshe ~]$ chmod u-r feathers
[lchileshe@lchileshe ~]$ cat feathers
cat: feathers: Permission denied
```

Рис. 2.13: права на чтение

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

```
[lchileshe@lchileshe ~]$ cp feathers play
cp: cannot open 'feathers' for reading: Permission denied
```

Рис. 2.14: cp

- Дайте владельцу файла ~/feathers право на чтение.

```
[lchileshe@lchileshe ~]$ chmod u+r fe
```
- Лишите владельца каталога ~/play права на выполнение.
- Перейдите в каталог ~/play. Что произошло?

```
[lchileshe@lchileshe ~]$ chmod u-x play/
[lchileshe@lchileshe ~]$ cd play/
bash: cd: play/: Permission denied
```

Рис. 2.15: Лишите владельца каталога ~/play

Дайте владельцу каталога ~/play право на выполнение.

```
[lchileshe@lchileshe ~]$ chmod u+x play/
```

Рис. 2.16: право на выполнение

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

```
1 foot 100% 100% 10.0.2.15/24 0% 26% us 95% 20:26
foot Pictures - Thunar 005-lab_files.pdf — Mozilla Firefox [Browser]

OPTIONS
-l
  Create an exclusive flock(2) lock file (/run/fsck/<diskname>.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 [fd]
  Report certain statistics for each fsck when it completes. These statistics include the exit status, the maximum
  run set size (in kilobytes), the elapsed all-clock time and the user and system CPU time used by the fsck run.
  For example:

  /dev/sda1: status 0, rss 92828, real 4.002804, user 2.677592, sys 0.86186

  GUI front-ends may specify a file descriptor fd, in which case the progress bar information will be sent to that
  file descriptor in a machine parsable format. For example:

  /dev/sda1 0 92828 4.002804 2.677592 0.86186

-s
  Serialize fsck operations. This is a good idea if you are checking multiple filesystems and the checkers are in
  an interactive mode. (Note: e2fsck(8) runs in an interactive mode by default. To make e2fsck(8) run in a
  non-interactive mode, you must either specify the -p or -a option, if you wish for errors to be corrected
  automatically, or the -n option if you do not.)

-t fslist
  Specifies the type(s) of filesystem to be checked. When the -A flag is specified, only filesystems that match
  fslist are checked. The fslist parameter is a comma-separated list of filesystems and options specifiers. All of
  the filesystems in this comma-separated list may be prefixed by a negation operator 'no' or '!', which requests
  that only those filesystems not listed in fslist will be checked. If none of the filesystems in fslist is
  prefixed by a negation operator, then only those listed filesystems will be checked.

Manual page fsck(8) line 52 (press h for help or q to quit)
```

Рис. 2.17: fsck

Проверяет и восстанавливает файловую систему Linux

```
1 foot 100% 100% 10.0.2.15/24 1% 26% us 95% 20:26
foot Pictures - Thunar 005-lab_files.pdf — Mozilla Firefox [Browser]

mount --make-unbindable /foo

COMMAND-LINE OPTIONS
The full set of mount options used by an invocation of mount is determined by first extracting the mount options for the filesystem from the fstab table, then applying any options specified by the -o argument, and finally applying a -r or -w option, when present.

The mount command does not pass all command-line options to the /sbin/mount.suffix mount helpers. The interface between mount and the mount helpers is described below in the EXTERNAL HELPERS section.

Command-line options available for the mount command are:

-a, --all
Mount all filesystems (of the given types) mentioned in fstab (except for those whose line contains the noauto keyword). The filesystems are mounted following their order in fstab. The mount command compares filesystem source, target (and fs root for bind mount or btrfs) to detect already mounted filesystems. The kernel table with already mounted filesystems is cached during mount --all. This means that all duplicated fstab entries will be mounted.

The correct functionality depends on /proc (to detect already mounted filesystems) and on /sys (to evaluate filesystem tags like UUID= or LABEL=). It's strongly recommended to mount /proc and /sys filesystems before mount -a is executed, or keep /proc and /sys at the beginning of fstab.

The option --all is possible to use for remount operation too. In this case all filters (-t and -O) are applied to the table of already mounted filesystems.

Since version 2.35 it is possible to use the command line option -o to alter mount options from fstab (see also --options-mode).

Note that it is a bad practice to use mount -a for fstab checking. The recommended solution is findmnt --verify.

-B, --bind
Remount a subtree somewhere else (so that its contents are available in both places). See above, under Bind mount operation.

Manual page mount(8) line 292 (press h for help or q to quit)
```

Рис. 2.18: mount

Команда монтирования в Linux используется для присоединения файловой системы к дереву каталогов в указанной точке монтирования.

```
1 foot 100% 100% 10.0.2.15/24 2% 26% us 95% 20:27
foot Pictures - Thunar 005-lab_files.pdf — Mozilla Firefox [Browser]
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@ods.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.
    Manual page mkfs(8) line 22 (press h for help or q to quit)
```

Рис. 2.19: mkfs

Команда `mkfs` в Linux используется для создания файловой системы на устройстве хранения данных, например жестком диске (HDD) или USB-накопителе.

```
1 foot 100% 100% 10.0.2.15/24 1% 26% us 95% 20:27
foot Pictures - Thunar 005-lab_files.pdf — Mozilla Firefox [Browser]

OPTIONS
-s, --signal signal
    The signal to send. It may be given as a name or a number.

-l, --list [number]
    Print a list of signal names, or convert the given signal number to a name. The signals can be found in
    /usr/include/linux/signal.h.

-L, --table
    Similar to -l, but it will print signal names and their corresponding numbers.

-a, --all
    Do not restrict the command-name-to-PID conversion to processes with the same UID as the present process.

-p, --pid
    Only print the process ID (PID) of the named processes, do not send any signals.

-r, --require-handler
    Do not send the signal if it is not caught in userspace by the signalled process.

--verbose
    Print PID(s) that will be signaled with kill along with the signal.

-q, --queue value
    Send the signal using sigqueue(3) rather than kill(2). The value argument is an integer that is sent along 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_sigval field of the siginfo_t structure.

--timeout milliseconds signal
    Send a signal defined in the usual way to a process, followed by an additional signal after a specified delay.
    The --timeout option causes kill to wait for a period defined in milliseconds before sending a follow-up signal
    to the process. This feature is implemented using the Linux kernel PID file descriptor feature in order to
    guarantee that the follow-up signal is sent to the same process or not sent if the process no longer exists.

Manual page kill(1) line 49 (press h for help or q to quit)
```

Рис. 2.20: kill

Отправляет сигнал процессу, обычно для его завершения.

3 Выводы

Познакомившись с файловой системой Linux, ее структурой, именами и содержанием, я получил четкое представление о том, как Linux организует файлы и каталоги и управляет ими.

Список литературы