

Лабораторная работа №7

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

Бунин Арсений Викторович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	Первичная работа с файловой системой	8
3.2	Перемещение объектов файловой системы	8
3.3	Изменяем доступ к файлам и папкам	9
3.4	Проверка целостности файловой системы	10
3.5	Копируем, создаем и читаем файлы и папки	11
3.6	Редактирование прав доступа	11
3.7	Отказ в доступе	12
3.8	Получение справочных сведений о команде mount	12
3.9	Получение справочных сведений о команде fsck	12
3.10	Получение справочных сведений о команде mkfs	13
3.11	Получение справочных сведений о команде kill	13

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

- Выполнить все примеры, приведенные в тексте лабораторной работы
- Выполнить задание по работе в файловой системе через терминал
- Выполнить задание по работе с правами доступа в файловой системе через терминал
- Проверить целостность файловой системы

3 Теоретическое введение

Для создания текстового файла можно использовать команду `touch`. Для просмотра файлов небольшого размера можно использовать команду `cat`. Команда `cp` используется для копирования файлов и каталогов. Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов. Каждый файл или каталог имеет права доступа (табл. 5.1). В сведениях о файле или каталоге указываются: – тип файла (символ `-` обозначает файл, а символ `d` – каталог); – права для владельца файла (`r` – разрешено чтение, `w` – разрешена запись, `x` – разрешено выполнение, `-` – право доступа отсутствует); – права для членов группы (`r` – разрешено чтение, `w` – разрешена запись, `x` – разрешено выполнение, `-` – право доступа отсутствует); – права для всех остальных (`r` – разрешено чтение, `w` – разрешена запись, `x` – разрешено выполнение, `-` – право доступа отсутствует). Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров. Другой способ определения смонтированных в операционной системе файловых систем – просмотр файла `/etc/fstab`. Сделать это можно например с помощью команды `cat`. С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы.

Выполнение лабораторной работы

Создание, копирование и чтение папок и файлов(рис. 3.1).

```

[arsenii@fedora ~]$ cd
[arsenii@fedora ~]$ touch abc
[arsenii@fedora ~]$ mkdir monthly
[arsenii@fedora ~]$ touch april
[arsenii@fedora ~]$ touch may
[arsenii@fedora ~]$ cp abc april
[arsenii@fedora ~]$ cp abc may
[arsenii@fedora ~]$ cp abc monthly
[arsenii@fedora ~]$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
[arsenii@fedora ~]$ cp april may monthly
[arsenii@fedora ~]$ cd monthly
[arsenii@fedora monthly]$ touch june
[arsenii@fedora monthly]$ cd
[arsenii@fedora ~]$ cp monthly/may monthly/june
[arsenii@fedora ~]$ ls monthly
abc  april  june  may
[arsenii@fedora ~]$ mkdir monthly.00
[arsenii@fedora ~]$ cp -r monthly monthly.00
[arsenii@fedora ~]$ cp -r monthly.00 /tmp
[arsenii@fedora ~]$

```

Рис. 3.1: Первичная работа с файловой системой

Перемещение файлов и папок

```

[arsenii@fedora ~]$ cd
[arsenii@fedora ~]$ mv april july
[arsenii@fedora ~]$ mv july monthly.00
[arsenii@fedora ~]$ ls monthly.00
july  monthly
[arsenii@fedora ~]$ mv monthly.00 monthly.01
[arsenii@fedora ~]$ mkdir reports
[arsenii@fedora ~]$ mv monthly.01 reports
[arsenii@fedora ~]$ mv reports/monthly.01 reports/monthly
[arsenii@fedora ~]$ ls reports
monthly
[arsenii@fedora ~]$

```

Рис. 3.2: Перемещение объектов файловой системы

Управление правами доступа в файловой системе (рис. 3.3).


```

[arsenii@fedora ~]$ cd
[arsenii@fedora ~]$ touch may
[arsenii@fedora ~]$ ls -l may
-rw-r--r--. 1 arsenii arsenii 0 map 20 13:06 may
[arsenii@fedora ~]$ chmod u+x may
[arsenii@fedora ~]$ ls -l may
-rwxr--r--. 1 arsenii arsenii 0 map 20 13:06 may
[arsenii@fedora ~]$ chmod u-x may
[arsenii@fedora ~]$ ls -l may
-rw-r--r--. 1 arsenii arsenii 0 map 20 13:06 may
[arsenii@fedora ~]$ cd
[arsenii@fedora ~]$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
[arsenii@fedora ~]$ chmod g-r monthly
[arsenii@fedora ~]$ chmod o-r monthly
[arsenii@fedora ~]$ ls -l monthly
итого 0
-rw-r--r--. 1 arsenii arsenii 0 map 20 12:45 abc
-rw-r--r--. 1 arsenii arsenii 0 map 20 12:46 april
-rw-r--r--. 1 arsenii arsenii 0 map 20 12:48 june
-rw-r--r--. 1 arsenii arsenii 0 map 20 12:46 may
[arsenii@fedora ~]$ cd
[arsenii@fedora ~]$ touch abc1
[arsenii@fedora ~]$ chmod g+w abc1
[arsenii@fedora ~]$ ls -l abc1
-rw-rw-r--. 1 arsenii arsenii 0 map 20 13:12 abc1
[arsenii@fedora ~]$

```

Рис. 3.3: Изменяем доступ к файлам и папкам

Проверяем целостность файловой системы (рис. 3.4).

```

[arsenii@fedora ~]$ cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Sat Sep 23 07:05:42 2023
#
# Accessible filesystems, by reference, are maintained under '/d
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) fo
#
# After editing this file, run 'systemctl daemon-reload' to upda
# units generated from this file.
#
UUID=c2687151-354f-4b6d-9422-2cea07e0a6d3 /
UUID=0abf8be7-75fe-4302-9683-26017a0049c6 /boot
UUID=c2687151-354f-4b6d-9422-2cea07e0a6d3 /home
[arsenii@fedora ~]$ fsck arsenii
fsck из util-linux 2.38.1
Usage: fsck.ext4 [-panyrcdfktvDFV] [-b superblock] [-B blocksize
[-l|-L bad_blocks_file] [-C fd] [-j external_jou
[-E extended-options] [-z undo_file] device

Emergency help:
-p                Automatic repair (no questions)
-n                Make no changes to the filesystem
-y                Assume "yes" to all questions
-c                Check for bad blocks and add them to the b
-f                Force checking even if filesystem is marke
-v                Be verbose
-b superblock     Use alternative superblock
-B blocksize      Force blocksize when looking for superblo
-j external_journal Set location of the external journal
-l bad_blocks_file Add to badblocks list
-L bad_blocks_file Set badblocks list
-z undo_file      Create an undo file
[arsenii@fedora ~]$

```

Рис. 3.4: Проверка целостности файловой системы

Самостоятельная работа по созданию, копированию и чтению папок (рис. 3.5)

```

[arsenii@fedora home]$ cd
[arsenii@fedora ~]$ cp /usr/include/gnumake.h ~
[arsenii@fedora ~]$ mv gnumake.h equipment
[arsenii@fedora ~]$ mkdir ski.places
[arsenii@fedora ~]$ mv gnumake.h ski.places
mv: не удалось выполнить stat для 'gnumake.h': Нет такого файла или каталога
[arsenii@fedora ~]$ mv equipment ski.places
[arsenii@fedora ~]$ mv /ski.places/equipment /ski.places/equiplist
mv: не удалось выполнить stat для '/ski.places/equipment': Нет такого файла или каталога
[arsenii@fedora ~]$ mv ~/ski.places/equipment ~/ski.places/equiplist
[arsenii@fedora ~]$ touch abc1
[arsenii@fedora ~]$ cp abc1 ski.places
[arsenii@fedora ~]$ mv ~/ski.places/abc1 ~/ski.places/equiplist2
[arsenii@fedora ~]$ mkdir ~/ski.places/equipment
[arsenii@fedora ~]$ mv ~/ski.places/equiplist2 ~/ski.places/equipment
[arsenii@fedora ~]$ mv ~/ski.places/equiplist1 ~/ski.places/equipment
mv: не удалось выполнить stat для '/home/arsenii/ski.places/equiplist1': Нет такого файла или каталога
[arsenii@fedora ~]$ mv ~/ski.places/equiplist ~/ski.places/equipment
[arsenii@fedora ~]$ mkdir newdir
[arsenii@fedora ~]$ mv ~/newdir ~/ski.places
[arsenii@fedora ~]$ mv ~/ski.places/newdir ~/ski.places/plans
[arsenii@fedora ~]$

```

Рис. 3.5: Копируем, создаем и читаем файлы и папки

Самостоятельная работа по редактированию прав доступа(рис. 3.6)

```

[arsenii@fedora ~]$ chmod u=rwx australia
[arsenii@fedora ~]$ chmod g=r-- australia
[arsenii@fedora ~]$ chmod o=r-- australia
[arsenii@fedora ~]$ mkdir play
[arsenii@fedora ~]$ chmod u=rwx play
[arsenii@fedora ~]$ chmod g=--x play
[arsenii@fedora ~]$ chmod o=--x play
[arsenii@fedora ~]$ touch my_os
[arsenii@fedora ~]$ chmod u=r- my_os
[arsenii@fedora ~]$ chmod u=r-x my_os
chmod: неверный режим: «u=r_x»
По команде «chmod --help» можно получить дополнительную информацию.
[arsenii@fedora ~]$ chmod u=r-x my_os
[arsenii@fedora ~]$ chmod g=r-- my_os
[arsenii@fedora ~]$ chmod o=r-- my_os
[arsenii@fedora ~]$ touch feathers
[arsenii@fedora ~]$ chmod u=rw- feathers
[arsenii@fedora ~]$ chmod g=rw- feathers
[arsenii@fedora ~]$ chmod o=r-- feathers
[arsenii@fedora ~]$

```

Рис. 3.6: Редактирование прав доступа

Отказано в доступе к папке по причине закрытия исполнения функций папки, то есть ее открытия(рис. 3.7)

```
arsenii@fedora ~]$ touch file.old
arsenii@fedora ~]$ cp file.old file.old
arsenii@fedora ~]$ mv file.old ~/play
arsenii@fedora ~]$ mkdir fun
arsenii@fedora ~]$ cp play ~/fun
arsenii@fedora ~]$ cp -r play ~/fun
arsenii@fedora ~]$ cp -r play ~/play
arsenii@fedora ~]$ cp -r play ~/play
arsenii@fedora ~]$ mv fun ~/play
arsenii@fedora ~]$ mv ~/play/fun ~/play/games
arsenii@fedora ~]$ chmod u-r file.old
arsenii@fedora ~]$ cat file.old
cat: file.old: Отказано в доступе
arsenii@fedora ~]$ chmod u+r file.old
arsenii@fedora ~]$ chmod u-x play
arsenii@fedora ~]$ cd play
bash: cd: play: Отказано в доступе
arsenii@fedora ~]$ chmod u+x play
arsenii@fedora ~]$
```

Рис. 3.7: Отказ в доступе

Получение справочных сведений о четырех командах, работающих с монтированием и целостностью файловой системы

```
arsenii@fedora ~$ man mount
NAME
    mount - mount a filesystem
SYNOPSIS
    mount [-h-v]
    mount [-t] [-t filesystem]
    mount -a [-f filesystem] [-t filesystem] [-o options]
    mount [-f filesystem] [-o options] device mountpoint
    mount [-f filesystem] [-o options] device mountpoint
    mount --bind --rbind --move olddir newdir
    mount --make-shared|-make-private|-make-unbindable|-make-relaxed|-make-private|-make-unbindable mountpoint
DESCRIPTION
    All files accessible to 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 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 context (if any) and user and mode of dir become restrictive, 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 device
    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 which case already mounted filesystems are ignored (see --all below for more details).
    Manual page mount(8) line 1 (press h for help or q to quit)
```

Рис. 3.8: Получение справочных сведений о команде mount

```
arsenii@fedora ~$ man fsck
NAME
    fsck - check and repair a Linux filesystem
SYNOPSIS
    fsck [-L|-A|-W|-T] [-v] [-t filesystem] [-o options]
DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name (e.g., /dev/hda1, /dev/cdrom), a mount point (e.g., /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=6a892b2f-4d30-4020-b000-000000000000 or LABEL=). 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 -a option.
    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_* programs) available under Linux. The filesystem-specific checker is searched for in the PATH environment variable. If the PATH is undefined then fsck falls back to /sbin.
    Manual page fsck(8) line 1 (press h for help or q to quit)
```

Рис. 3.9: Получение справочных сведений о команде fsck

```
arsenii@fedora:~$ man mkfs
mkfs(8)
System Administration

NAME
    mkfs - build a Linux filesystem

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

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific mkfs.type utilities.

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

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

    In actuality, mkfs is simply a front-end for the various filesystem builders (mkfs.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.

    -F, --filesystem filesystem
        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.)

NOTES
    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 <dauid@pan.com>, <ask to see kernel mailinglist at l.magnet.org>, <see kernel at kernel.org>, <see kernel at kernel.org>.

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

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

Рис. 3.10: Получение справочных сведений о команде mkfs

```
arsenii@fedora:~$ man kill
kill(1)
User Commands

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal=-signal+] [-pid] [-w] [-timeout milliseconds signal] [-o] [pidname...]
    kill -l [number] | -t

DESCRIPTION
    The command kill sends the specified signal to the specified processes or process groups.

    If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

    Many modern shells have a built-in kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --queue options, and the possibility to specify processes by command name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

    pid
        Each pid can be expressed in one of the following ways:
        0
            where g is larger than 0. The process with PID g is signaled.
        *
            All processes in the current process group are signaled.
        -1
            All processes with a PID larger than 1 are signaled.
        -0
            where g is larger than 1. All processes in process group g are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either a signal must be specified first, or the argument must be preceded by a '+' option, otherwise it will be taken as the signal to send.

    name
        All processes invoked using this name will be signaled.
```

Рис. 3.11: Получение справочных сведений о команде kill

4 Выводы

Научились работать с файловой системой Linux.

Список литературы