

Операционные системы

Анализ файловой структуры UNIX. Команды для работы с файлами и каталогами

Данил Артёмов

11 марта 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи 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, по проверке использования диска и обслуживанию файловой системы.

- 1 Выполнить приимеры
- 2 Выполнить дествия по работе с каталогами и файлами
- 3 Выполнить действия с правами доступа
- 4 Получить дополнительные сведения при помощи справки по командам.

Процесс выполнения лабораторной работы

```
dsartemov@dsartemov:~$  
dsartemov@dsartemov:~$ touch abc1  
dsartemov@dsartemov:~$ cp abc1 april  
dsartemov@dsartemov:~$ cp abc1 may  
dsartemov@dsartemov:~$ mkdir monthly  
dsartemov@dsartemov:~$ cp april may monthly  
dsartemov@dsartemov:~$ cp monthly/may monthly/june  
dsartemov@dsartemov:~$ ls monthly  
april  june  may  
dsartemov@dsartemov:~$ mkdir monthly.00  
dsartemov@dsartemov:~$ cp -r monthly monthly.00  
dsartemov@dsartemov:~$ cp -r monthly.00 /tmp  
dsartemov@dsartemov:~$
```

Рис. 1: Выполнение примеров

```
dsartemov@dsartemov:~$  
dsartemov@dsartemov:~$ mv april july  
dsartemov@dsartemov:~$ mv july monthly.00  
dsartemov@dsartemov:~$ ls monthly.00  
july  monthly  
dsartemov@dsartemov:~$ mv monthly.00 monthly.01  
dsartemov@dsartemov:~$ mkdir reports  
dsartemov@dsartemov:~$ mv monthly.01 reports  
dsartemov@dsartemov:~$ mv reports/monthly.01 reports/monthly  
dsartemov@dsartemov:~$
```

Рис. 2: Выполнение примеров

Выполнение примеров

```
dsartemov@dsartemov:~$ touch may
dsartemov@dsartemov:~$ ls -l may
-rw-r--r--. 1 dsartemov dsartemov 0 map 11 13:10 may
dsartemov@dsartemov:~$ chmod u+x may
dsartemov@dsartemov:~$ ls -l may
-rwxr--r--. 1 dsartemov dsartemov 0 map 11 13:10 may
dsartemov@dsartemov:~$ chmod u-x may
dsartemov@dsartemov:~$ ls -l may
-rw-r--r--. 1 dsartemov dsartemov 0 map 11 13:10 may
dsartemov@dsartemov:~$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
dsartemov@dsartemov:~$ chmod g-r,o-r monthly
dsartemov@dsartemov:~$ touch abc1
dsartemov@dsartemov:~$ chmod g+w abc1
dsartemov@dsartemov:~$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

```
dsartemov@dsartemov:~$ cp /usr/include/linux/sysinfo.h ~
dsartemov@dsartemov:~$ mv sysinfo.h equipment
dsartemov@dsartemov:~$ mkdir ski.plases
dsartemov@dsartemov:~$ mv equipment ski.plases/
dsartemov@dsartemov:~$ mv ski.plases/equipment ski.plases/equiplist
dsartemov@dsartemov:~$ touch abc1
dsartemov@dsartemov:~$ cp abc1 ski.plases/equiplist2
dsartemov@dsartemov:~$ cd ski.plases/
dsartemov@dsartemov:~/ski.plases$ mkdir equipment
dsartemov@dsartemov:~/ski.plases$ mv equiplist equipment/
dsartemov@dsartemov:~/ski.plases$ mv equiplist2 equipment/
dsartemov@dsartemov:~/ski.plases$ mkdir newdir
dsartemov@dsartemov:~/ski.plases$ mv newdir ski.plases/
dsartemov@dsartemov:~/ski.plases$ mv ski.plases/newdir/ ski.plases/plans
mv: не удалось выполнить stat для 'ski.plases/newdir/': Нет такого файла или каталога
dsartemov@dsartemov:~/ski.plases$
```

Рис. 4: Работа с каталогами

```
dsartemov@dsartemov:~/ski.plases$  
dsartemov@dsartemov:~/ski.plases$ mkdir australia play  
dsartemov@dsartemov:~/ski.plases$ touch my_os feathers  
dsartemov@dsartemov:~/ski.plases$ chmod 744 australia/  
dsartemov@dsartemov:~/ski.plases$ chmod 711 play/  
dsartemov@dsartemov:~/ski.plases$ chmod 544 my_os  
dsartemov@dsartemov:~/ski.plases$ chmod 664 feathers  
dsartemov@dsartemov:~/ski.plases$ ls -l  
итого 0  
drwxr--r--. 1 dsartemov dsartemov  0 map 11 13:11 australia  
drwxr-xr-x. 1 dsartemov dsartemov 38 map 11 13:11 equipment  
-rw-rw-r--. 1 dsartemov dsartemov  0 map 11 13:11 feathers  
-r-xr--r--. 1 dsartemov dsartemov  0 map 11 13:11 my_os  
drwx--x--x. 1 dsartemov dsartemov  0 map 11 13:11 play  
drwxr-xr-x. 1 dsartemov dsartemov  0 map 11 13:11 ski.plases  
dsartemov@dsartemov:~/ski.plases$
```

Рис. 5: Настройка прав доступа

```
qemu:x:107:107:qemu user:/:/sbin/nologin
polkitd:x:114:114:User for polkitd:/:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/:/sbin/nologin
chrony:x:997:994:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:996:993:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
gluster:x:995:992:GlusterFS daemons:/run/gluster:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
pipewire:x:994:991:PipeWire System Daemon:/run/pipewire:/usr/sbin/nologin
unbound:x:993:990:Unbound DNS resolver:/var/lib/unbound:/sbin/nologin
nm-openconnect:x:992:989:NetworkManager user for OpenConnect:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
wsdd:x:991:988:Web Services Dynamic Discovery host daemon:/:/sbin/nologin
sssd:x:990:986:User for sssd:/run/sss:/sbin/nologin
openvpn:x:989:985:OpenVPN:/etc/openvpn:/sbin/nologin
nm-openvpn:x:988:984:Default user for running openvpn spawned by NetworkManager:/:/sbin/nologin
/etc/passwd
```

Рис. 6: Файл /etc/passwd

Работа с файлами и правами доступа

```
dsartemov@dsartemov:~$  
dsartemov@dsartemov:~$ cp feathers file.old  
dsartemov@dsartemov:~$ mv file.old play/  
mv: невозможно переместить 'file.old' в 'play/': Это не каталог  
dsartemov@dsartemov:~$ mkdir australia play  
dsartemov@dsartemov:~$ mv file.old play/  
dsartemov@dsartemov:~$ mkdir fun  
dsartemov@dsartemov:~$ cp -R play/ fun/  
dsartemov@dsartemov:~$ mv fun/ play/games  
dsartemov@dsartemov:~$ chmod u-r feathers  
dsartemov@dsartemov:~$ cat feathers  
cat: feathers: Отказано в доступе  
dsartemov@dsartemov:~$ cp feathers feathers2  
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе  
dsartemov@dsartemov:~$ chmod u+r feathers  
dsartemov@dsartemov:~$ chmod u-x play/  
dsartemov@dsartemov:~$ cd play/  
bash: cd: play/: Отказано в доступе  
dsartemov@dsartemov:~$ chmod +x play/  
dsartemov@dsartemov:~$
```

Рис. 7: Работа с файлами и правами доступа

```
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.

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

```
FCK(8)                                     System Administration                                     FCK(8)

NAME
    fsck - check and repair a Linux filesystem

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

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

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

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

    0
        No errors

    1
        Filesystem errors corrected

    2
        System should be rebooted

    4
        Filesystem errors left uncorrected

    8
        Operational error

    16

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

Рис 9: Команда 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.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. 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.

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

Рис. 10: Команда mkfs

```
KILL(1)                                User Commands                                KILL(1)
```

NAME

kill - terminate a process

SYNOPSIS

```
kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...
```

```
kill -l [number] | -L
```

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.

Most modern shells have a builtin **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:

n

where n is larger than 0. The process with PID n is signaled.

0

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

Выводы по проделанной работе

В ходе данной работы мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Научились совершать базовые операции с файлами, управлять правами их доступа для пользователя и групп. Ознакомились с Анализом файловой системы. А также получили базовые навыки по проверке использования диска и обслуживанию файловой системы.