

# Операционные системы

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

---

Артём Арутюнян

14 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

---

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

Рис. 1: Выполнение примеров

```
abrutynyan@abrutynyan:~$  
abrutynyan@abrutynyan:~$ cd  
abrutynyan@abrutynyan:~$ mv april may monthly  
abrutynyan@abrutynyan:~$ touch april  
abrutynyan@abrutynyan:~$ touch may  
abrutynyan@abrutynyan:~$ mv april july  
abrutynyan@abrutynyan:~$ mv july monthly.00/  
abrutynyan@abrutynyan:~$ ls monthly.00/  
july  monthly  
abrutynyan@abrutynyan:~$ mv monthly.00/ monthly.01  
abrutynyan@abrutynyan:~$ mkdir reports  
abrutynyan@abrutynyan:~$ mv monthly.01/ reports/  
abrutynyan@abrutynyan:~$ mv reports/monthly.01/ reports/monthly  
abrutynyan@abrutynyan:~$
```

Рис. 2: Выполнение примеров

```
abarutynyan@abarutynyan:~$ touch may
abarutynyan@abarutynyan:~$ ls -l may
-rw-r--r--. 1 abarutynyan abarutynyan 0 map 14 10:33 may
abarutynyan@abarutynyan:~$ chmod u+x may
abarutynyan@abarutynyan:~$ ls -l may
-rwxr--r--. 1 abarutynyan abarutynyan 0 map 14 10:33 may
abarutynyan@abarutynyan:~$ chmod u-x may
abarutynyan@abarutynyan:~$ ls -l may
-rw-r--r--. 1 abarutynyan abarutynyan 0 map 14 10:33 may
abarutynyan@abarutynyan:~$ chmod g-r,o-r monthly/
abarutynyan@abarutynyan:~$ chmod g+w abc1
abarutynyan@abarutynyan:~$
```

Рис. 3: Выполнение примеров



## Создание директорий и копирование файлов

```
abarutynyan@abarutynyan:~$  
abarutynyan@abarutynyan:~$ cp /usr/include/linux/sysinfo.h ~  
abarutynyan@abarutynyan:~$ mv sysinfo.h equipment  
abarutynyan@abarutynyan:~$ mkdir ski.plases  
abarutynyan@abarutynyan:~$ mv equipment ski.plases/  
abarutynyan@abarutynyan:~$ mv ski.plases/equipment ski.plases/equiplist  
abarutynyan@abarutynyan:~$ touch abc1  
abarutynyan@abarutynyan:~$ cp abc1 ski.plases/equiplist2  
abarutynyan@abarutynyan:~$ cd s  
site/      ski.plases/ snap/  
abarutynyan@abarutynyan:~$ cd ski.plases/  
abarutynyan@abarutynyan:~/ski.plases$ mkdir equipment  
abarutynyan@abarutynyan:~/ski.plases$ mv equiplist equipment/  
abarutynyan@abarutynyan:~/ski.plases$ mv equiplist2 equipment/  
abarutynyan@abarutynyan:~/ski.plases$ cd  
abarutynyan@abarutynyan:~$ mkdir newdir  
abarutynyan@abarutynyan:~$ mv newdir/ ski.plases/  
abarutynyan@abarutynyan:~$ mv ski.plases/newdir/ ski.plases/plans  
abarutynyan@abarutynyan:~$
```

Рис. 4: Работа с каталогами

## Работа с командой chmod

```
abrutynyan@abrutynyan:~$ mkdir australia play
abrutynyan@abrutynyan:~$ touch my_os feathers
abrutynyan@abrutynyan:~$ chmod 744 australia/
abrutynyan@abrutynyan:~$ chmod 711 play/
abrutynyan@abrutynyan:~$ chmod 544 my_os
abrutynyan@abrutynyan:~$ chmod 664 feathers
abrutynyan@abrutynyan:~$ ls -l
итого 0
-rw-rw-r--. 1 abrutynyan abrutynyan 0 мар 14 10:37 abc1
drwxr--r--. 1 abrutynyan abrutynyan 0 мар 14 10:38 australia
-rw-rw-r--. 1 abrutynyan abrutynyan 0 мар 14 10:38 feathers
drwxr-xr-x. 1 abrutynyan abrutynyan 74 фев 26 11:05 git-extended
-rw-r--r--. 1 abrutynyan abrutynyan 0 мар 14 10:33 may
drwx--x--x. 1 abrutynyan abrutynyan 24 мар 14 10:31 monthly
-r-xr--r--. 1 abrutynyan abrutynyan 0 мар 14 10:38 my_os
drwx--x--x. 1 abrutynyan abrutynyan 0 мар 14 10:38 play
drwxr-xr-x. 1 abrutynyan abrutynyan 14 мар 14 10:32 reports
drwxr-xr-x. 1 abrutynyan abrutynyan 58 фев 26 11:14 site
drwxr-xr-x. 1 abrutynyan abrutynyan 28 мар 14 10:38 ski.places
drwx-----. 1 abrutynyan abrutynyan 8 фев 26 11:14 snap
drwxr-xr-x. 1 abrutynyan abrutynyan 10 фев 26 10:45 work
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 Видео
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 Документы
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 Загрузки
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 Изображения
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 Музыка
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 Общедоступные
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 'Рабочий стол'
drwxr-xr-x. 1 abrutynyan abrutynyan 0 фев 26 10:19 Шаблоны
abrutynyan@abrutynyan:~$
```

```
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/sbin:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/usr/sbin/nologin
operator:x:11:0:operator:/root:/usr/sbin/nologin
games:x:12:100:games:/usr/games:/usr/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/usr/sbin/nologin
dbus:x:81:81:System Message Bus:/:/usr/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/:/usr/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
geoclue:x:999:999>User for geoclue:/var/lib/geoclue:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
systemd-oom:x:998:998:systemd Userspace OOM Killer:/:/usr/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
polkitd:x:114:114>User for polkitd:/:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/:/sbin/nologin
```

Рис. 6: Файл /etc/passwd

## Работа с файлами и правами доступа

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

Рис. 7: Работа с файлами и правами доступа

```
abarutynyan@abarutynyan:~ — man mount
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

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

```
abaratynyan@abaratynyan:~ — man fsck
FSCK(8)                                     System Administration                                     FSCK(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)
```

```
abaratynyan@abaratynyan:~ — man mkfs
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)
```

```
abarutynyan@abarutynyan:~ — man kill
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.