

# Операционные системы

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

---

Сарханов Рамиэль

21 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

---

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

Рис. 1: Выполнение примеров

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

Рис. 2: Выполнение примеров

```
rsarhanov@rsarhanov:~$  
rsarhanov@rsarhanov:~$ touch may  
rsarhanov@rsarhanov:~$ ls -l may  
-rw-r--r--. 1 rsarhanov rsarhanov 0 map 21 14:50 may  
rsarhanov@rsarhanov:~$ chmod u+x may  
rsarhanov@rsarhanov:~$ ls -l may  
-rwxr--r--. 1 rsarhanov rsarhanov 0 map 21 14:50 may  
rsarhanov@rsarhanov:~$ chmod u-x may  
rsarhanov@rsarhanov:~$ ls -l may  
-rw-r--r--. 1 rsarhanov rsarhanov 0 map 21 14:50 may  
rsarhanov@rsarhanov:~$ chmod g-r,o-r monthly  
rsarhanov@rsarhanov:~$ chmod g+w abc1  
rsarhanov@rsarhanov:~$
```

Рис. 3: Выполнение примеров



## Создание директорий и копирование файлов

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

Рис. 4: Работа с каталогами

## Работа с командой chmod

```
rsarhanov@rsarhanov:~$ mkdir australia play
rsarhanov@rsarhanov:~$ touch my_os feathers
rsarhanov@rsarhanov:~$ chmod 744 australia/
rsarhanov@rsarhanov:~$ chmod 711 play/
rsarhanov@rsarhanov:~$ chmod 544 my_os
rsarhanov@rsarhanov:~$ chmod 664 feathers
rsarhanov@rsarhanov:~$ ls -l
итого 0
-rw-rw-r--. 1 rsarhanov rsarhanov  0 map 21 14:50 abc1
drwxr--r--. 1 rsarhanov rsarhanov  0 map 21 14:51 australia
-rw-rw-r--. 1 rsarhanov rsarhanov  0 map 21 14:51 feathers
-rw-r--r--. 1 rsarhanov rsarhanov  0 map 21 14:50 may
drwx--x--x. 1 rsarhanov rsarhanov 24 map 21 14:49 monthly
-r-xr--r--. 1 rsarhanov rsarhanov  0 map 21 14:51 my_os
drwx--x--x. 1 rsarhanov rsarhanov  0 map 21 14:51 play
drwxr-xr-x. 1 rsarhanov rsarhanov 14 map 21 14:50 reports
drwxr-xr-x. 1 rsarhanov rsarhanov 56 map 21 14:32 site
drwxr-xr-x. 1 rsarhanov rsarhanov 28 map 21 14:51 ski.plases
drwx-----. 1 rsarhanov rsarhanov  8 map 21 14:32 snap
drwxr-xr-x. 1 rsarhanov rsarhanov 10 map 21 14:25 work
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:21 Видео
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:21 Документы
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:49 Загрузки
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:21 Изображения
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:21 Музыка
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:21 Общедоступные
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:21 'Рабочий стол'
drwxr-xr-x. 1 rsarhanov rsarhanov  0 map 21 14:21 Шаблоны
rsarhanov@rsarhanov:~$
```

```
rsarhanov@rsarhanov:~ — less /etc/passwd
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
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
/etc/passwd
```

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

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`.

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

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

```
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
        Usage or syntax error

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

```

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.

    -V, --version
        Display version information 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

All processes in the current process group are signaled.

-1

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

Рис. 11: Команда kill



## 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.