

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

---

Саргсян Сурен Бабкенович<sup>1</sup>

29 августа, 2023, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и задачи работы

---

## Цель лабораторной 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.

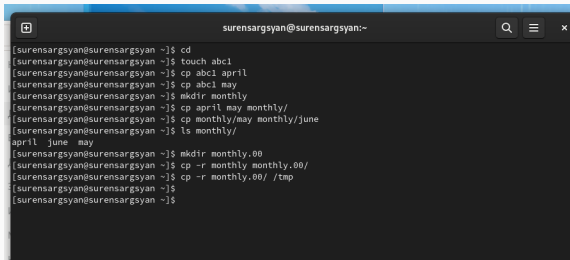
# Задачи лабораторной работы

- 1 Выполнить примеры
- 2 Выполнить действия 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**

---

# Выполнение примеров



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

Рис. 1: Выполнение примеров

# Выполнение примеров

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

Рис. 2: Выполнение примеров

# Выполнение примеров

```
[surensargsyan@surensargsyan ~]$  
[surensargsyan@surensargsyan ~]$ cd  
[surensargsyan@surensargsyan ~]$ touch may  
[surensargsyan@surensargsyan ~]$ ls -l may  
-rw-r--r--. 1 surensargsyan surensargsyan 0 апр 29 14:33 may  
[surensargsyan@surensargsyan ~]$ chmod u+x may  
[surensargsyan@surensargsyan ~]$ ls -l may  
-rwxr--r--. 1 surensargsyan surensargsyan 0 апр 29 14:33 may  
[surensargsyan@surensargsyan ~]$ chmod u-x may  
[surensargsyan@surensargsyan ~]$ ls -l may  
-rw-r--r--. 1 surensargsyan surensargsyan 0 апр 29 14:33 may  
[surensargsyan@surensargsyan ~]$ ccd  
bash: ccd: команда не найдена...  
Аналогичная команда: 'cd'  
[surensargsyan@surensargsyan ~]$ cd  
[surensargsyan@surensargsyan ~]$ mkdir monthly/  
mkdir: невозможно создать каталог «monthly/»: Файл существует  
[surensargsyan@surensargsyan ~]$ chmod g-r,o-r monthly/  
[surensargsyan@surensargsyan ~]$ cd  
[surensargsyan@surensargsyan ~]$ touch abc1  
[surensargsyan@surensargsyan ~]$ chmod g+w abc1  
[surensargsyan@surensargsyan ~]$
```

Рис. 3: Выполнение примеров



# Создание директорий и копирование файлов

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

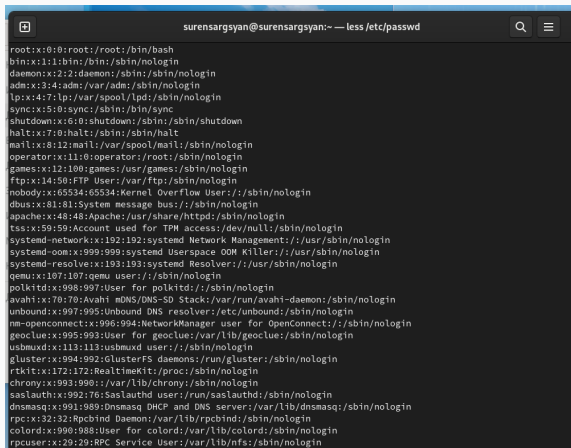
Рис. 4: Работа с каталогами

# Работа с командой chmod

```
[surensargsyan@surensargsyan ~]$  
[surensargsyan@surensargsyan ~]$ mkdir australia play  
[surensargsyan@surensargsyan ~]$ touch my_os fethers  
[surensargsyan@surensargsyan ~]$ chmod 744 australia  
[surensargsyan@surensargsyan ~]$ chmod 711 play  
[surensargsyan@surensargsyan ~]$ chmod 544 my_os  
[surensargsyan@surensargsyan ~]$ chmod 664 fethers  
[surensargsyan@surensargsyan ~]$ ls -l  
итого 0  
-rw-rw-r--. 1 surensargsyan surensargsyan 0 авг 29 14:38 abcl  
drwxr--r--. 1 surensargsyan surensargsyan 0 авг 29 14:44 australia  
-rw-rw-r--. 1 surensargsyan surensargsyan 0 авг 29 14:44 fethers  
-rw-r--r--. 1 surensargsyan surensargsyan 0 авг 29 14:33 may  
drwx--x--x. 1 surensargsyan surensargsyan 24 авг 29 14:29 monthly  
-r-xr--r--. 1 surensargsyan surensargsyan 0 авг 29 14:44 my_os  
drwx--x--x. 1 surensargsyan surensargsyan 0 авг 29 14:44 play  
drwxr-xr-x. 1 surensargsyan surensargsyan 14 авг 29 14:33 reports  
drwxr-xr-x. 1 surensargsyan surensargsyan 28 авг 29 14:39 ski.places  
drwxr-xr-x. 1 surensargsyan surensargsyan 10 авг 29 13:19 work  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 Видео  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 Документы  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 Загрузки  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 Изображения  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 Музыка  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 Общедоступные  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 'Рабочий стол'  
drwxr-xr-x. 1 surensargsyan surensargsyan 0 авг 29 12:49 Шаблоны  
[surensargsyan@surensargsyan ~]$
```

Рис. 5: Настройка прав доступа

# Файл /etc/passwd

A terminal window with a dark background and light text. The title bar shows the user 'surensargsyan' and the command 'less /etc/passwd'. The terminal displays the contents of the /etc/passwd file, listing system and regular users with their IDs, names, shells, and home directories.

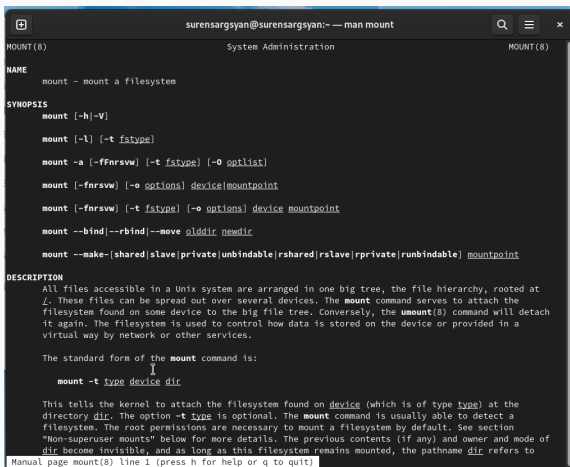
```
surensargsyan@surensargsyan:~ — less /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
systemd-oom:x:999:999:systemd Userspace OOM Killer:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/usr/sbin/nologin
qemu:x:107:107:qemu user:/sbin/nologin
polkitd:x:998:997:User for polkitd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
unbound:x:997:995:Unbound DNS resolver:/etc/unbound:/sbin/nologin
nm-openconnect:x:996:994:NetworkManager user for OpenConnect:/sbin/nologin
geoclue:x:995:993:User for geoclue:/var/lib/geoclue:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/sbin/nologin
gluster:x:994:992:GlusterFS daemons:/run/gluster:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
chrony:x:993:990:/var/lib/chrony:/sbin/nologin
sasauthd:x:992:76:Saslauthd user:/run/saslauthd:/sbin/nologin
dnsmasq:x:991:989:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
colord:x:990:988:User for colord:/var/lib/colord:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
```

Рис. 6: Файл /etc/passwd

# Работа с файлами и правами доступа

```
[surensargsyan@surensargsyan ~]$  
+ [surensargsyan@surensargsyan ~]$ cp fethers file.old  
[surensargsyan@surensargsyan ~]$ mv file.old play/  
[surensargsyan@surensargsyan ~]$ mkdir fun  
[surensargsyan@surensargsyan ~]$ cp -R play/ fun/  
[surensargsyan@surensargsyan ~]$ mv fun/ play/games  
[surensargsyan@surensargsyan ~]$ chmod u-r fethers  
[surensargsyan@surensargsyan ~]$ cat fethers  
cat: fethers: Отказано в доступе  
[surensargsyan@surensargsyan ~]$ cp fethers fethers2  
cp: невозможно открыть 'fethers' для чтения: Отказано в доступе  
[surensargsyan@surensargsyan ~]$ chmod u+r fethers  
[surensargsyan@surensargsyan ~]$ chmod u-x play/  
[surensargsyan@surensargsyan ~]$ cd play/  
bash: cd: play/: Отказано в доступе  
[surensargsyan@surensargsyan ~]$ chmod +x play/  
[surensargsyan@surensargsyan ~]$
```

Рис. 7: Работа с файлами и правами доступа



```

suresargsyan@suresargsyan:~ -- man mount

MOUNT(8)                                     System Administration                                     MOUNT(8)

NAME
     mount - mount a filesystem

SYNOPSIS
     mount [-h|-V]

     mount [-l] [-t fstype]

     mount -a [-fnrsvw] [-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

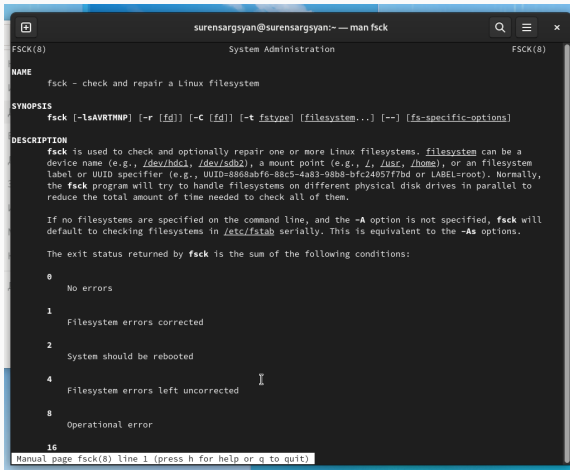
     This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. The option -t type is optional. The mount command is usually able to detect a filesystem. The root permissions are necessary to mount a filesystem by default. See section "Non-superuser mounts" 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

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

```

Рис. 8: Команда mount

# Справка по командам



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

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTnmp] [-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 an 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.

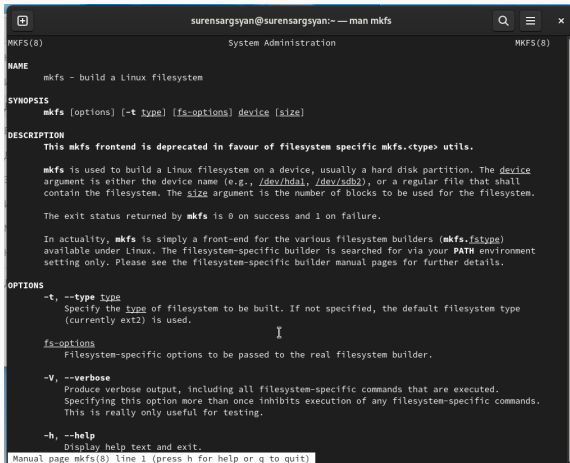
    If no filesystems are specified on the command line, and the -A option is not specified, fsck will
    default to checking filesystems in /etc/fstab serially. This is equivalent to the -As options.

    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



```
surensargsyan@surensargsyan:~ — 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/hdal, /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.

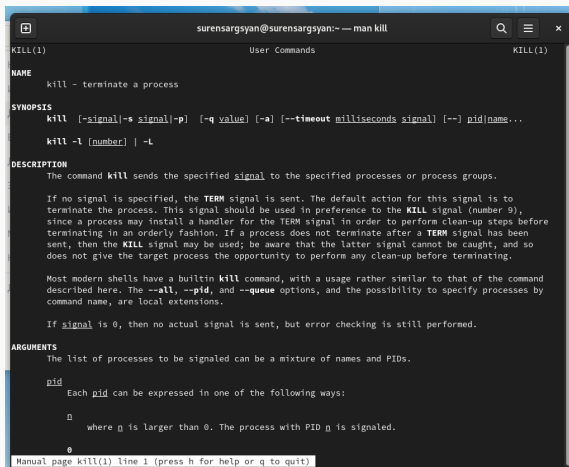
    fs-options
        Filesystem-specific options to be passed to the real filesystem builder.

    -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



```
surensargsyan@surensargsyan:~ -- 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)
```

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.