

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

---

Олейников Артём Игоревич<sup>1</sup>

16 июня, 2024, 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.

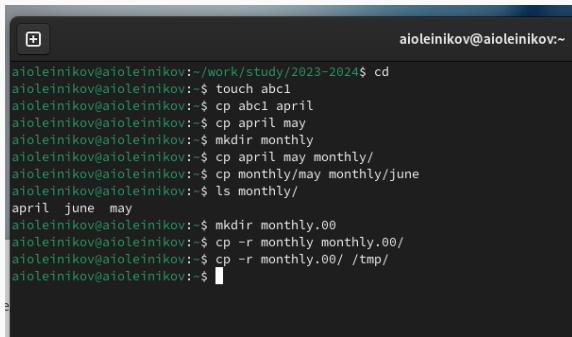
# Задачи лабораторной работы

- 1 Выполнить 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**

---

# Выполнение примеров



```
aioleinikov@aioleinikov:~  
aioleinikov@aioleinikov:~/work/study/2023-2024$ cd  
aioleinikov@aioleinikov:~$ touch abc1  
aioleinikov@aioleinikov:~$ cp abc1 april  
aioleinikov@aioleinikov:~$ cp april may  
aioleinikov@aioleinikov:~$ mkdir monthly  
aioleinikov@aioleinikov:~$ cp april may monthly/  
aioleinikov@aioleinikov:~$ cp monthly/may monthly/june  
aioleinikov@aioleinikov:~$ ls monthly/  
april  june  may  
aioleinikov@aioleinikov:~$ mkdir monthly.00  
aioleinikov@aioleinikov:~$ cp -r monthly monthly.00/  
aioleinikov@aioleinikov:~$ cp -r monthly.00/ /tmp/  
aioleinikov@aioleinikov:~$
```

Рис. 1: Выполнение примеров

# Выполнение примеров

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

Рис. 2: Выполнение примеров

# Выполнение примеров

```
aioleinikov@aioleinikov:~$  
aioleinikov@aioleinikov:~$ cd  
aioleinikov@aioleinikov:~$ touch may  
aioleinikov@aioleinikov:~$ ls -l may  
-rw-r--r--. 1 aioleinikov aioleinikov 0 июн 16 09:32 may  
aioleinikov@aioleinikov:~$ chmod u+x may  
aioleinikov@aioleinikov:~$ ls -l may  
-rwxr--r--. 1 aioleinikov aioleinikov 0 июн 16 09:32 may  
aioleinikov@aioleinikov:~$ chmod u-x may  
aioleinikov@aioleinikov:~$ ls -l may  
-rw-r--r--. 1 aioleinikov aioleinikov 0 июн 16 09:32 may  
aioleinikov@aioleinikov:~$ cd  
aioleinikov@aioleinikov:~$ chmod g-r,o-r monthly  
aioleinikov@aioleinikov:~$ chmod g+w abc1  
aioleinikov@aioleinikov:~$
```

Рис. 3: Выполнение примеров



# Создание директорий и копирование файлов

```
Из аioleinikov@aioleinikov:~$ cp /usr/include/linux/sysinfo.h ~
Му аioleinikov@aioleinikov:~$ mv sysinfo.h equipment
аioleinikov@aioleinikov:~$ mkdir ski.plases
Ко аioleinikov@aioleinikov:~$ mv equipment ski.plases/
аioleinikov@aioleinikov:~$ mv ski.plases/equipment ski.plases/equiplist
аioleinikov@aioleinikov:~$ touch abc1
Др аioleinikov@aioleinikov:~$ cp abc1 ski.plases/equiplist2
аioleinikov@aioleinikov:~$ cd ski.plases/
аioleinikov@aioleinikov:~/ski.plases$ mkdir equipment
аioleinikov@aioleinikov:~/ski.plases$ mv equiplist equipment/
аioleinikov@aioleinikov:~/ski.plases$ mv equiplist2 equipment/
аioleinikov@aioleinikov:~/ski.plases$ cd
аioleinikov@aioleinikov:~$ mkdir newdir
аioleinikov@aioleinikov:~$ mv newdir/ ski.plases/
аioleinikov@aioleinikov:~$ mv ski.plases/newdir/ ski.plases/plans
аioleinikov@aioleinikov:~$
```

Рис. 4: Работа с каталогами

# Работа с командой chmod

```
airoleinikov@airoleinikov:~$  
airoleinikov@airoleinikov:~$  
airoleinikov@airoleinikov:~$ mkdir australia play  
airoleinikov@airoleinikov:~$ touch my_os feathers  
airoleinikov@airoleinikov:~$ chmod 744 feathers  
airoleinikov@airoleinikov:~$ chmod 711 play/  
airoleinikov@airoleinikov:~$ chmod 544 my_os  
airoleinikov@airoleinikov:~$ chmod 644 feathers  
airoleinikov@airoleinikov:~$ ls -l  
  
итого 0  
-rw-rw-r--. 1 aioleinikov aioleinikov 0 июн 16 09:33 abc1  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 16 09:35 australia  
-rw-rw-r--. 1 aioleinikov aioleinikov 0 июн 16 09:35 feathers  
drwxr-xr-x. 1 aioleinikov aioleinikov 74 июн 16 08:44 git-extended  
-rw-rw-r--. 1 aioleinikov aioleinikov 0 июн 16 09:32 may  
drwx--x--x. 1 aioleinikov aioleinikov 24 июн 16 09:29 monthly  
-r-xr--r--. 1 aioleinikov aioleinikov 0 июн 16 09:35 my_os  
drwx--x--x. 1 aioleinikov aioleinikov 0 июн 16 09:35 play  
drwxr-xr-x. 1 aioleinikov aioleinikov 14 июн 16 09:31 reports  
drwxr-xr-x. 1 aioleinikov aioleinikov 28 июн 16 09:34 ski.plases  
drwxr-xr-x. 1 aioleinikov aioleinikov 10 июн 15 17:50 work  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 Видео  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 Документы  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 Загрузки  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 Изображения  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 Музыка  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 Общедоступные  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 'Рабочий стол'  
drwxr-xr-x. 1 aioleinikov aioleinikov 0 июн 15 17:22 Шаблоны  
airoleinikov@airoleinikov:~$
```

Рис. 5: Настройка прав доступа

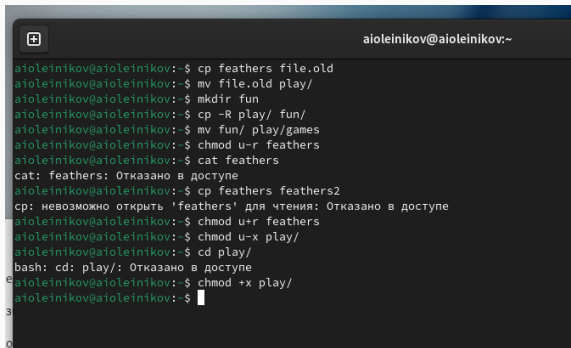
# Файл /etc/passwd

```

aiolenikov@aiolenikov:~ — 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
^
systemd-coredump:x:998:998:systemd Core Dumper:/usr/sbin/nologin
^
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
^
systemd-oom:x:997:997:systemd Userspace OOM Killer:/usr/sbin/nologin
^
systemd-resolve:x:193:193:systemd Resolver:/usr/sbin/nologin
^
systemd-timesync:x:996:996:systemd Time Synchronization:/usr/sbin/nologin
qemu:x:107:107:qemu user:/usr/sbin/nologin
^
polkitd:x:114:114:User for polkitd:/usr/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/usr/sbin/nologin
^
geoclue:x:995:994:User for geoclue:/usr/lib/geoclue:/usr/sbin/nologin
^
nm-openconnect:x:994:993:NetworkManager user for OpenConnect:/usr/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/usr/sbin/nologin
^
gluster:x:993:992:GlusterFS daemons:/usr/lib/glusterd:/usr/sbin/nologin
rtkit:x:172:172:RealtimeKit:/usr/lib/rtkit:/usr/sbin/nologin
^
pipewire:x:992:990:Pipewire System Daemon:/usr/lib/pipewire:/usr/sbin/nologin
^
saslauthd:x:991:76:Saslauthd user:/usr/lib/saslauthd:/usr/sbin/nologin
^
chrony:x:990:989:chrony system user:/usr/lib/chrony:/usr/sbin/nologin
dnsmasq:x:989:988:Dnsmasq DHCP and DNS server:/usr/lib/dnsmasq:/usr/sbin/nologin
^
rpc:x:32:32:Rpcbind Daemon:/usr/lib/rpcbind:/usr/sbin/nologin
^
rpcuser:x:29:29:RPC Service User:/usr/lib/nfs:/usr/sbin/nologin
openvpn:x:988:987:OpenVPN:/etc/openvpn:/usr/sbin/nologin
nm-openvpn:x:987:986:Default user for running openvpn spawned by NetworkManager:/usr/sbin/nologin
colord:x:986:985:User for colord:/usr/lib/colord:/usr/sbin/nologin
unbound:x:985:984:Unbound DNS resolver:/usr/lib/unbound:/usr/sbin/nologin
abrt:x:173:173:/usr/lib/abrt:/usr/sbin/nologin
flatpak:x:984:982:Flatpak system helper:/usr/lib/flatpak:/usr/sbin/nologin
gdm:x:42:42:GNOME Display Manager:/usr/lib/gdm:/usr/sbin/nologin
/etc/passwd
```

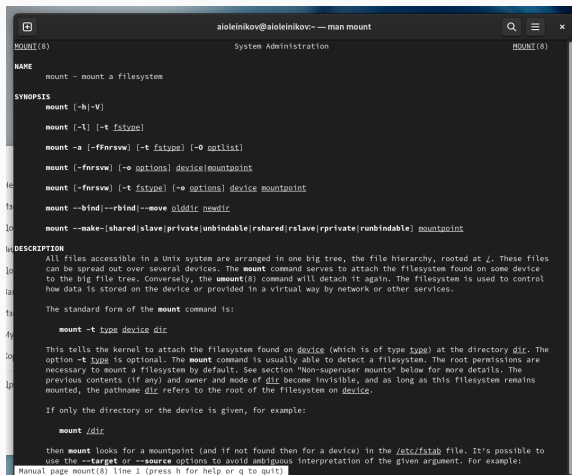
# Работа с файлами и правами доступа



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

Рис. 7: Работа с файлами и правами доступа

# Справка по командам



```
aiolelnikov@aiolelnikov:~ — man mount
NAME
mount - mount a filesystem

SYNOPSIS
mount [-h|-V]

mount [-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

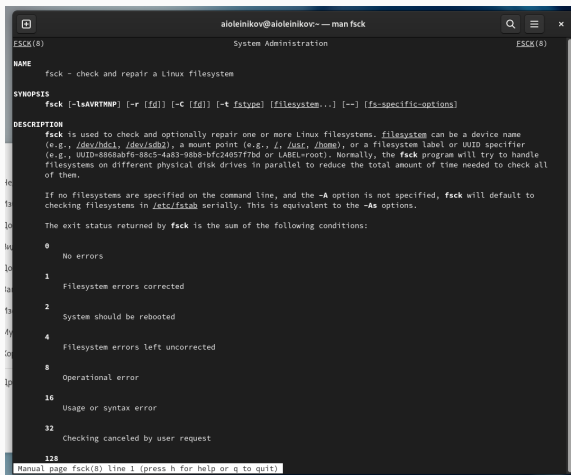
This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. The
option -t type is optional. The mount command is usually able to detect a filesystem. The root permissions are
necessary to mount a filesystem by default. See section "Non-superuser mounts"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

    mount /dir

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

Рис. 8: Команда mount



```
aioleinikov@aioleinikov:~ -- 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=8869a1fe-88c5-4a83-9808-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.

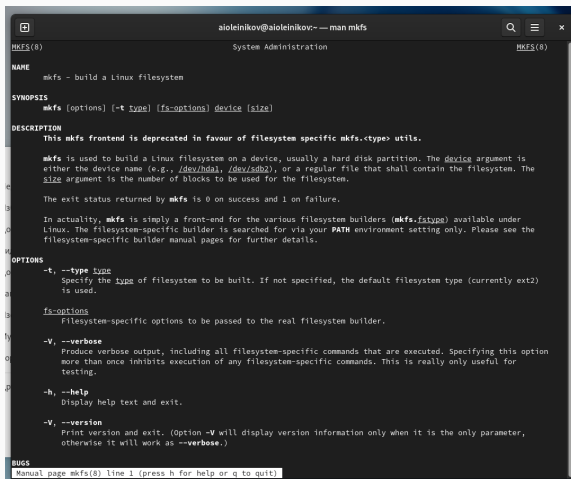
    The exit status returned by fsck is the sum of the following conditions:

    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
    32     Checking canceled by user request

    128

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

Рис. 9: Команда fsck



```
aioleinikov@aioleinikov:~ -- 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.

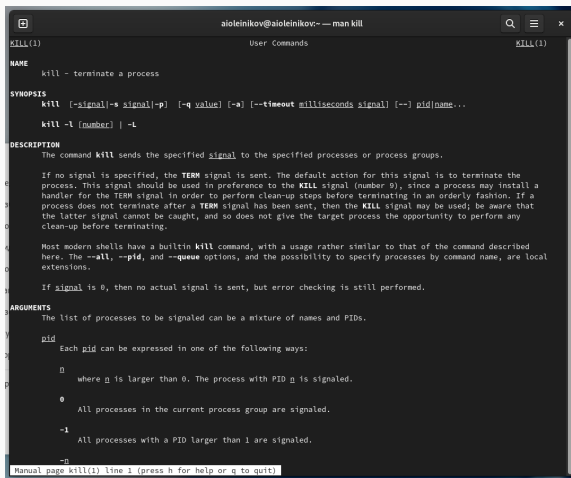
  -h, --help
    Display help text and exit.

  -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.)

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

Рис. 10: Команда mkfs

# Справка по командам



```
aioleinikov@aioleinikov:~ -- 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:

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

    0
      All processes in the current process group are signaled.

    -1
      All processes with a PID larger than 1 are signaled.

    -p
      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.