

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

---

Сальников Даниил Александрович НБИбд-02-21<sup>1</sup>

3 мая, 2022, 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.

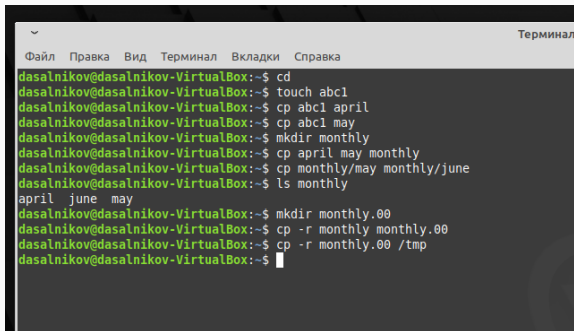
# Задачи лабораторной работы

- 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



```

Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
dasalnikov@dasalnikov-VirtualBox:~$ cd
dasalnikov@dasalnikov-VirtualBox:~$ touch abc1
dasalnikov@dasalnikov-VirtualBox:~$ cp abc1 april
dasalnikov@dasalnikov-VirtualBox:~$ cp abc1 may
dasalnikov@dasalnikov-VirtualBox:~$ mkdir monthly
dasalnikov@dasalnikov-VirtualBox:~$ cp april may monthly
dasalnikov@dasalnikov-VirtualBox:~$ cp monthly/may monthly/june
dasalnikov@dasalnikov-VirtualBox:~$ ls monthly
april  june  may
dasalnikov@dasalnikov-VirtualBox:~$ mkdir monthly.00
dasalnikov@dasalnikov-VirtualBox:~$ cp -r monthly monthly.00
dasalnikov@dasalnikov-VirtualBox:~$ cp -r monthly.00 /tmp
dasalnikov@dasalnikov-VirtualBox:~$
```

Figure 1: Выполнение примеров

# Выполнение примеров

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

Figure 2: Выполнение примеров

# Выполнение примеров

```
dasalnikov@dasalnikov-VirtualBox:~$ cd
dasalnikov@dasalnikov-VirtualBox:~$ touch may
dasalnikov@dasalnikov-VirtualBox:~$ ls -l may
-rw-rw-r-- 1 dasalnikov dasalnikov 0 мая 3 15:37 may
dasalnikov@dasalnikov-VirtualBox:~$ chmod u+x may
dasalnikov@dasalnikov-VirtualBox:~$ ls -l may
-rwxrw-r-- 1 dasalnikov dasalnikov 0 мая 3 15:37 may
dasalnikov@dasalnikov-VirtualBox:~$ chmod u-x may
dasalnikov@dasalnikov-VirtualBox:~$ ls -l may
-rw-rw-r-- 1 dasalnikov dasalnikov 0 мая 3 15:37 may
dasalnikov@dasalnikov-VirtualBox:~$ cd
dasalnikov@dasalnikov-VirtualBox:~$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
dasalnikov@dasalnikov-VirtualBox:~$ chmod g-r,o-r monthly
dasalnikov@dasalnikov-VirtualBox:~$ cd
dasalnikov@dasalnikov-VirtualBox:~$ touch abc1
dasalnikov@dasalnikov-VirtualBox:~$ chmod g+w abc1
dasalnikov@dasalnikov-VirtualBox:~$
```

Figure 3: Выполнение примеров



# Создание директорий и копирование файлов

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

Figure 4: Работа с каталогами

# Работа с командой chmod

```
dasalnikov@dasalnikov-VirtualBox:~$  
dasalnikov@dasalnikov-VirtualBox:~$  
dasalnikov@dasalnikov-VirtualBox:~$ mkdir australia play  
dasalnikov@dasalnikov-VirtualBox:~$ touch my_os feathers  
dasalnikov@dasalnikov-VirtualBox:~$ chmod 744 australia/  
dasalnikov@dasalnikov-VirtualBox:~$ chmod 711 play/  
dasalnikov@dasalnikov-VirtualBox:~$ chmod 544 my_os  
dasalnikov@dasalnikov-VirtualBox:~$ chmod 664 feathers  
dasalnikov@dasalnikov-VirtualBox:~$ ls -l  
итого 56  
-rw-rw-r-- 1 dasalnikov dasalnikov  0 мая 3 15:37 abc1  
drwxr--r-- 2 dasalnikov dasalnikov 4096 мая 3 15:37 australia  
-rw-rw-r-- 1 dasalnikov dasalnikov  0 мая 3 15:37 feathers  
-rw-rw-r-- 1 dasalnikov dasalnikov  0 мая 3 15:37 may  
drwx-wx--x 2 dasalnikov dasalnikov 4096 мая 3 15:36 monthly  
-r-xr--r-- 1 dasalnikov dasalnikov  0 мая 3 15:37 my_os  
drwx--x--x 2 dasalnikov dasalnikov 4096 мая 3 15:37 play  
drwxrwxr-x 3 dasalnikov dasalnikov 4096 мая 3 15:37 reports  
drwxrwxr-x 4 dasalnikov dasalnikov 4096 мая 3 15:37 ski.places  
drwxrwxr-x 3 dasalnikov dasalnikov 4096 апр 30 11:26 work  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 Видео  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 Документы  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 Загрузки  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 Изображения  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 Музыка  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 Общедоступные  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 'Рабочий стол'  
drwxr-xr-x 2 dasalnikov dasalnikov 4096 апр 30 11:15 Шаблоны  
dasalnikov@dasalnikov-VirtualBox:~$
```

Figure 5: Настройка прав доступа

# Файл /etc/passwd

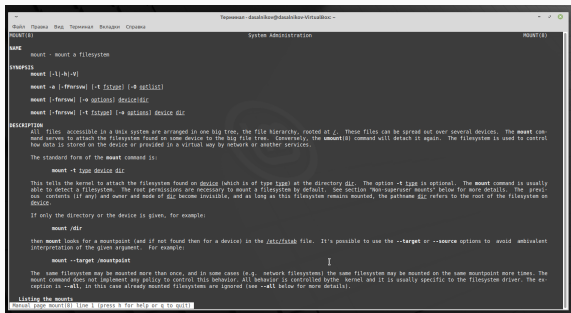
```
dasalnikov@dasalnikov-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/:nonexistent:/usr/sbin/nologin
syslog:x:104:110:/:home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:nonexistent:/usr/sbin/nologin
ntp:x:106:111:/:nonexistent:/usr/sbin/nologin
tss:x:107:112:TPM software stack,,:/var/lib/tpm:/bin/false
rtkit:x:108:113:RealtimeKit,,:/proc:/usr/sbin/nologin
systemd-coredump:x:109:114:systemd Core Dumper,,:/run/systemd:/usr/sbin/nologin
kernoops:x:110:65534:Kernel Oops Tracking Daemon,,:/usr/sbin/nologin
```

Figure 6: Файл /etc/passwd

# Работа с файлами и правами доступа

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

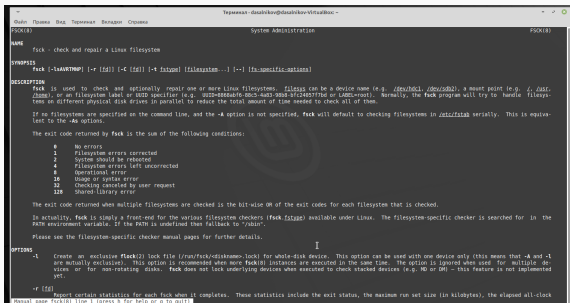
Figure 7: Работа с файлами и правами доступа



```
Tepovman - daniel@kali:~/VirtualBox -
man [options] [command] [section]
NAME
    mount - mount a filesystem
SYNOPSIS
    mount [-l|-h|-V]
    mount -a [-f|noauto] [-t fstype] [-o options]
    mount [-f|noauto] [-o options] device:dir
    mount [-f|noauto] [-t fstype] [-o options] device dir
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 another 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:
        mount --target /mountpoint
    The same filesystem may be mounted more than once, and in some cases (e.g. network filesystems) the same filesystem may be mounted on the same mountpoint more times. The mount command does not implement any policy to control this behavior. All behavior is controlled by the kernel and it is usually specific to the filesystem driver. The exception is --all, in this case already mounted filesystems are ignored (see --all below for more details).
    Listing the mounts
    Run the following command to see the help for the mount command.
```

Figure 8: Команда mount

# Справка по командам



```
Teplovskan-dasabikov@dasabikov-VirtualBox -
dash: /bin/dash: Bad interpreter: No such file or directory
fsck(8)                                System Administration              fsck(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-laAknrnp] [-r [fs]] [-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/sda1, /dev/sdb2), a mount point (e.g. /, /usr, /home), or an filesystem label or UUID specifier (e.g. UUID=8866b1f6-80c3-4a03-980d-bf240577b6 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 -Aa options.

    The exit code 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    Shared-library error

    The exit code returned when multiple filesystems are checked is the bit-wise OR of the exit codes for each filesystem that is checked.

    In actuality, fsck is simply a front-end for the various filesystem checkers (fsck.fstype) available under Linux.  The filesystem-specific checker is searched for in the PATH environment variable.  If the PATH is undefined then fallback to /sbin/.

    Please see the filesystem-specific checker manual pages for further details.

OPTIONS
    -l      Create an exclusive fsck(2) lock file (/run/fsck-diskname.lock) for whole-disk device.  This option can be used with one device only (this means that -A and -l are mutually exclusive).  This option is recommended when more fsck(8) instances are executed in the same time.  The option is ignored when used for multiple devices or for non-rotating disks.  fsck does not lock underlying devices when executed to check stacked devices (e.g. MD or DM) - this feature is not implemented yet.
    -r [fs] Report certain statistics for each fsck when it completes.  These statistics include the exit status, the maximum run set size (in kilobytes), the elapsed all-clock
```

Figure 9: Команда fsck

```
Terminal - david@bork@davidbork-virtualbox ~
david@bork: ~$ man mkfs
mkfs(8)                                System Administration                                mkfs(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [-f fs-options] device [size]

DESCRIPTION
    This mkfs frontend is deprecated in favor 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/sda), /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 code 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.

    -f, --fs-options 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.

    -V, --version
        Display version information and exit. (Option -V will display version information only when it is the only parameter, otherwise it will work as --verbose.)

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

BUGS
    All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific programs do not automatically detect the device size and require the size parameter to be specified.

AUTHORS
    David Forzi (david@bork.com)

Manual page mkfs(8) Line 2 (press h for help or q to quit)
```

Figure 10: Команда mkfs





## **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.