

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

Мартынов Николай Алексеевич НБИбд-02-21¹

3 мая, 2022, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

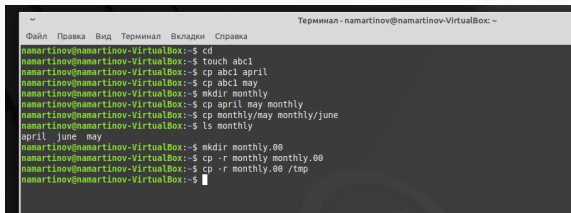
Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами, по проверке использования диска и обслуживанию файловой системы.

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

Figure 1: Выполнение примеров

Выполнение примеров

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

Figure 2: Выполнение примеров

Выполнение примеров

```
namartinov@namartinov-VirtualBox:~$  
namartinov@namartinov-VirtualBox:~$ cd  
namartinov@namartinov-VirtualBox:~$ touch may  
namartinov@namartinov-VirtualBox:~$ ls -l may  
-rw-rw-r-- 1 namartinov namartinov 0 мая 2 19:52 may  
namartinov@namartinov-VirtualBox:~$ chmod u+x may  
namartinov@namartinov-VirtualBox:~$ ls -l may  
-rwxrw-r-- 1 namartinov namartinov 0 мая 2 19:52 may  
namartinov@namartinov-VirtualBox:~$ chmod u-x may  
namartinov@namartinov-VirtualBox:~$ ls -l may  
-rw-rw-r-- 1 namartinov namartinov 0 мая 2 19:52 may  
namartinov@namartinov-VirtualBox:~$ cd  
namartinov@namartinov-VirtualBox:~$ mkdir monthly  
mkdir: невозможно создать каталог «monthly»: Файл существует  
namartinov@namartinov-VirtualBox:~$ chmod g-r,o-r monthly  
namartinov@namartinov-VirtualBox:~$ cd  
namartinov@namartinov-VirtualBox:~$ touch abc1  
namartinov@namartinov-VirtualBox:~$ chmod g+w abc1  
namartinov@namartinov-VirtualBox:~$
```

Figure 3: Выполнение примеров

Создание директорий и копирование файлов

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

Figure 4: Работа с каталогами

Работа с командой chmod

```
namartinov@namartinov-VirtualBox:~$  
namartinov@namartinov-VirtualBox:~$ mkdir australia play  
namartinov@namartinov-VirtualBox:~$ touch my os feathers  
namartinov@namartinov-VirtualBox:~$ chmod 744 australia/  
namartinov@namartinov-VirtualBox:~$ chmod 711 play/  
namartinov@namartinov-VirtualBox:~$ chmod 544 my os  
namartinov@namartinov-VirtualBox:~$ chmod 664 feathers  
namartinov@namartinov-VirtualBox:~$ ls -l  
итого 56  
-rw-rw-r-- 1 namartinov namartinov  0 мая 2 19:52 abcl  
drwxr--r-- 2 namartinov namartinov 4096 мая 2 19:52 australia  
-rw-rw-r-- 1 namartinov namartinov  0 мая 2 19:52 feathers  
-rw-rw-r-- 1 namartinov namartinov  0 мая 2 19:52 may  
drwx-wx--x 2 namartinov namartinov 4096 мая 2 19:51 monthly  
-r-xr--r-- 1 namartinov namartinov  0 мая 2 19:52 my_os  
drwx--x--x 2 namartinov namartinov 4096 мая 2 19:52 play  
drwxrwxr-x 3 namartinov namartinov 4096 мая 2 19:51 reports  
drwxrwxr-x 4 namartinov namartinov 4096 мая 2 19:52 ski.places  
drwxrwxr-x 3 namartinov namartinov 4096 апр 23 19:18 work  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 Видео  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 Документы  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 Загрузки  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 Изображения  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 Музыка  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 Общедоступные  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 'Рабочий стол'  
drwxr-xr-x 2 namartinov namartinov 4096 апр 23 18:59 Шаблоны  
namartinov@namartinov-VirtualBox:~$
```

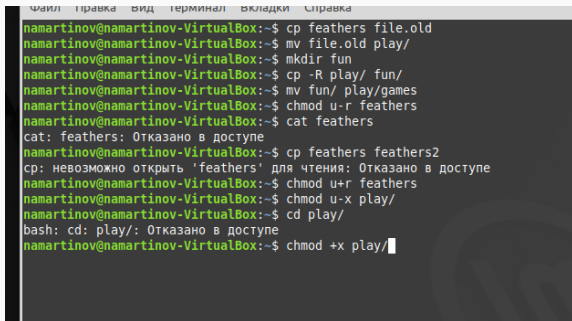
Figure 5: Настройка прав доступа

Файл /etc/passwd

```
nasartimov@nasartimov-VirtualBox:~$  
nasartimov@nasartimov-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:Mail 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  
system-network:x:100:100:system Network Management,,,:/run/systemd:/usr/sbin/nologin  
system-resolve:x:101:101:system Resolver,,,:/run/systemd:/usr/sbin/nologin  
system-timesync:x:102:102:system Time Synchronization,,,:/run/systemd:/usr/sbin/nologin  
messagebus:x:103:100:/:/nonexistent:/usr/sbin/nologin  
syslog:x:104:110:/:/home/syslog:/usr/sbin/nologin  
apt:x:109:65534:/:/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

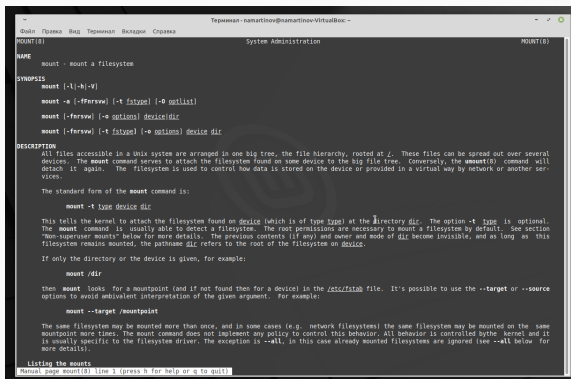
Работа с файлами и правами доступа



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

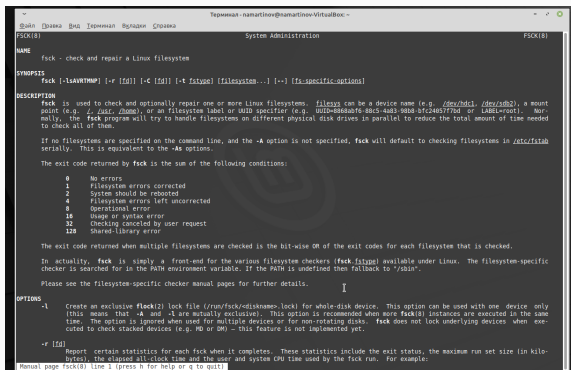
Figure 7: Работа с файлами и правами доступа

Справка по командам



```
Терминал - namartinov@namartinov-VirtualBox: ~
[Файл Правка Вид Терминал Вкладки Справка]
System Administration
MOUNT(8)
NAME
mount - mount a filesystem
SYNOPSIS
mount [-l|-h|-V]
mount -e [-f|-fsrvw] [-t fstype] [-o optlist]
mount [-fsrvw] [-o options] device|dir
mount [-fsrvw] [-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 univalent 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
Manual page mount(8) line 3 (press h for help or q to quit)
```

Figure 8: Команда mount



```
Терминал - namartinov@namartinov-VirtualBox ~
@YAN Давид Вид Терминал Виртуал Давид
FSCK(8) System Administration FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-laAVTWP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystems 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=880a8a76-8d53-4a43-98ba-bf42465777da or LABEL=root1). 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 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.

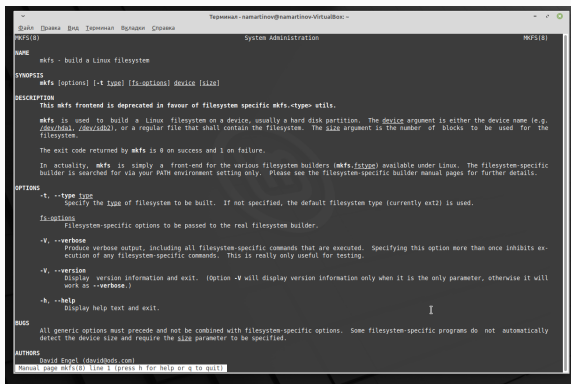
    Please see the filesystem-specific checker manual pages for further details.

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 [fd] 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 time and the user and system CPU time used by the fsck run. For example:

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

Figure 9: Команда fsck



```
Терминал - namarinov@namartnov-VirtualBox: ~
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/sdb1), /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.<type>) 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.

    -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.)

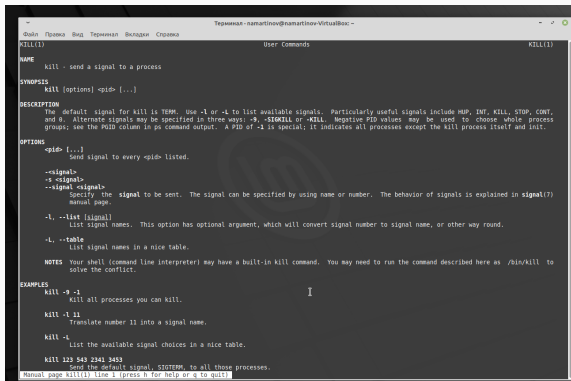
    -h, --help
        Display help text and exit.

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 Engel (david@dods.com)

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

Figure 10: Команда mkfs



```
Терминал: namartinov@namartinov-VirtualBox -
Файл Правка Вид Терминал Вкладки Справка
kill(1) User Commands KILL(1)

NAME
  kill - send a signal to a process

SYNOPSIS
  kill [options] <pid> [...]

DESCRIPTION
  The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process groups; see the PID column in ps command output. A PID of -1 is special; it indicates all processes except the kill process itself and init.

OPTIONS
  <pid> [...]
    Send signal to every <pid> listed.

  -s <signal>
  -s <signal>
  --signal <signal>
    Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.

  -l, --list [<signal>]
    List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.

  -L, --table
    List signal names in a nice table.

NOTES
  Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as ./bin/kill to solve the conflict.

EXAMPLES
  kill -9 -1
    Kill all processes you can kill.

  kill -l 11
    Translate number 11 into a signal name.

  kill -l
    List the available signal choices in a nice table.

  kill 123 543 2341 3453
    Send the default signal, SIGTERM, to all those processes.

Manual page kill(1) http://linux.die.net/man/1/kill & press h for help or q to quit
```

Figure 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.