

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

Голчин Задех Негин¹

21 июня, 2023, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

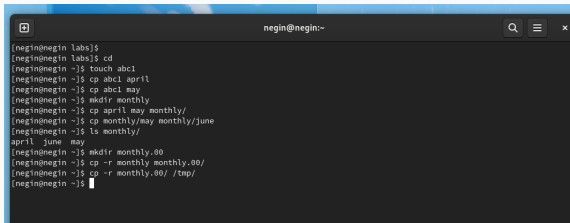
Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами, по проверке использования диска и обслуживанию файловой системы.

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



```
negin@negin:~  
(negin@negin labs)$  
(negin@negin labs)$ cd  
(negin@negin ~)$ touch abc1  
(negin@negin ~)$ cp abc1 april  
(negin@negin ~)$ cp abc1 may  
(negin@negin ~)$ mkdir monthly  
(negin@negin ~)$ cp april may monthly/  
(negin@negin ~)$ cp monthly/may monthly/june  
(negin@negin ~)$ ls monthly/  
april  june  may  
(negin@negin ~)$ mkdir monthly.00  
(negin@negin ~)$ cp -r monthly monthly.00/  
(negin@negin ~)$ cp -r monthly.00/ /tmp/  
(negin@negin ~)$
```

Рис. 1: Выполнение примеров

Выполнение примеров

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

Рис. 2: Выполнение примеров

Выполнение примеров

```
[negin@negin ~]$  
[negin@negin ~]$  
[negin@negin ~]$ cd  
[negin@negin ~]$ touch may  
[negin@negin ~]$ ls -l may  
-rw-r--r--. 1 negin negin 0 июн 21 22:04 may  
[negin@negin ~]$ chmod u+x may  
[negin@negin ~]$ ls -l may  
-rwxr--r--. 1 negin negin 0 июн 21 22:04 may  
[negin@negin ~]$ chmod u-x may  
[negin@negin ~]$ ls -l may  
-rw-r--r--. 1 negin negin 0 июн 21 22:04 may  
[negin@negin ~]$ cd  
[negin@negin ~]$ mkdir monthly/  
mkdir: невозможно создать каталог «monthly/»: Файл существует  
[negin@negin ~]$ chmod g-r,o-r monthly/  
[negin@negin ~]$ cd  
[negin@negin ~]$ touch abc1  
[negin@negin ~]$ chmod g+w abc1  
[negin@negin ~]$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

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

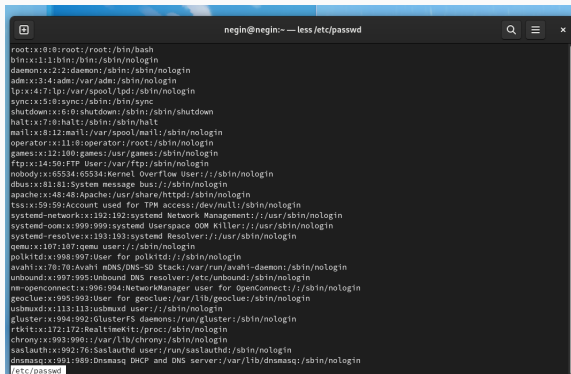
Рис. 4: Работа с каталогами

Работа с командой chmod

```
[negin@negin ~]$  
[negin@negin ~]$ mkdir australia play  
[negin@negin ~]$ touch my_os feathers  
[negin@negin ~]$ chmod 744 australia/  
[negin@negin ~]$ chmod 711 play/  
[negin@negin ~]$ chmod 544 my_os  
[negin@negin ~]$ chmod 664 feathers  
[negin@negin ~]$ ls -l  
итого 0  
-rw-rw-r--. 1 negin negin 0 июн 21 22:06 abc1  
drwxr--r--. 1 negin negin 0 июн 21 22:07 australia  
-rw-rw-r--. 1 negin negin 0 июн 21 22:08 feathers  
-rw-r--r--. 1 negin negin 0 июн 21 22:04 may  
drwx--x--x. 1 negin negin 24 июн 21 22:08 monthly  
-r-xr--r--. 1 negin negin 0 июн 21 22:08 my_os  
drwx--x--x. 1 negin negin 0 июн 21 22:07 play  
drwxr-xr-x. 1 negin negin 14 июн 21 22:03 reports  
drwxr-xr-x. 1 negin negin 28 июн 21 22:07 ski.places  
drwxr-xr-x. 1 negin negin 10 июн 21 20:01 work  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 видео  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 документы  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 загрузки  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 изображения  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 музыка  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 Общедоступные  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 'Рабочий стол'  
drwxr-xr-x. 1 negin negin 0 июн 21 19:50 Шаблоны  
[negin@negin ~]$
```

Рис. 5: Настройка прав доступа

Файл /etc/passwd



The image shows a terminal window with the title bar "negin@negin:~ — less /etc/passwd". The terminal displays the contents of the /etc/passwd file, which lists system and user accounts. Each line represents an account with fields for username, password placeholder (x), UID, GID, username, home directory, and shell. The accounts listed are: root, bin, daemon, adm, lp, sync, shutdown, halt, mail, operator, games, ftp, nobody, dbus, apache, tss, systemd-network, systemd-oom, systemd-resolve, qemu, polkitd, avahi, unbound, na-openconnect, geoclue, usbmuxd, gluster, rtkit, chrony, saslauthd, and dnsmasq. The terminal window has a dark background and a light-colored text.

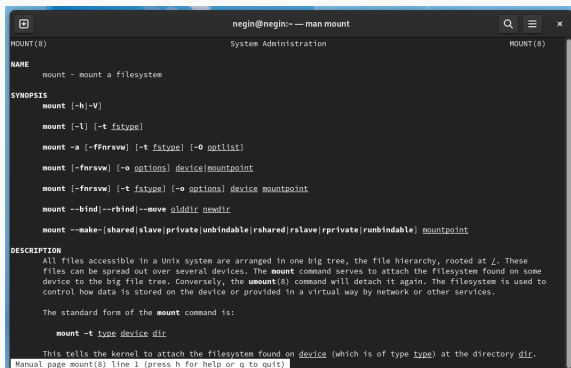
```
negin@negin:~ — less /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
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
na-openconnect:x:396:904:NetworkManager user for OpenConnect:/usr/sbin/nologin
geoclue:x:995:992: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
saslauthd:x:992:76:Saslauthd user:/run/saslauthd:/sbin/nologin
dnsmasq:x:991:989:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
/etc/passwd
```

Рис. 6: Файл /etc/passwd

Работа с файлами и правами доступа

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

Рис. 7: Работа с файлами и правами доступа



```
negin@negin:~ — man mount
MOUNT(8)                               System Administration                               MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-L] [-t fstype]

    mount -a [-fFnrsw] [-t fstype] [-o optlist]

    mount [-fnrsw] [-o options] device mountpoint

    mount [-fnrsw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable] mountpoint

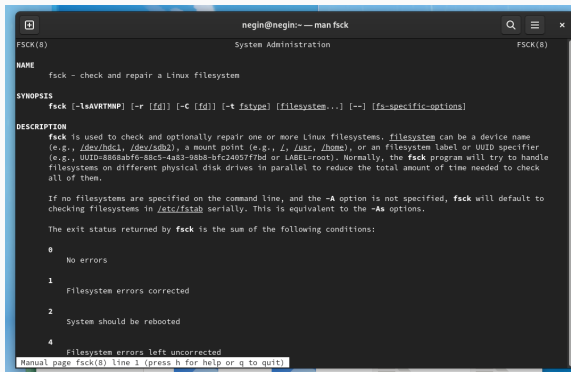
DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the filesystem found on some device to the big file tree. Conversely, the umount(8) command will 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.
    Manual page mount(8) line 1 (press h for help or q to quit)
```

Рис. 8: Команда mount



```
negin@negin:~$ 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 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

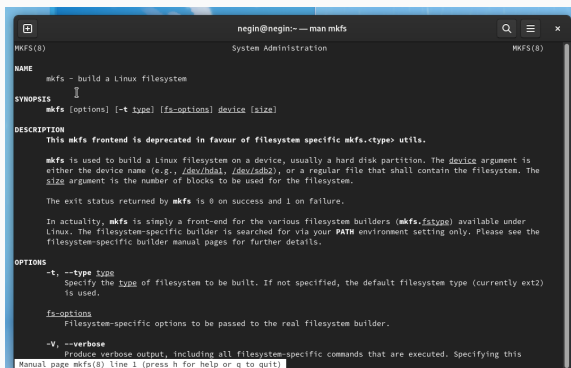
    1      Filesystem errors corrected

    2      System should be rebooted

    4      Filesystem errors left uncorrected

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

Рис. 9: Команда fsck



```
negin@negin:~ -- 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/ndb2), 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.

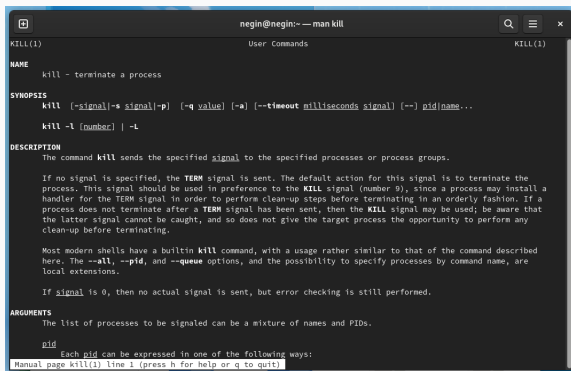
OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.

    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

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

Рис. 10: Команда mkfs



```
negin@negin:~ -- 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:
        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.