

Операционные системы

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

Анастасия Гончарь

13 марта 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи 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.

- 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

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

Рис. 1: Выполнение примеров

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

Рис. 2: Выполнение примеров

```
aagonchar@aagonchar:~$  
aagonchar@aagonchar:~$ cd  
aagonchar@aagonchar:~$ touch may  
aagonchar@aagonchar:~$ ls -l may  
-rw-r--r--. 1 aagonchar aagonchar 0 map 13 10:51 may  
aagonchar@aagonchar:~$ chmod u+x may  
aagonchar@aagonchar:~$ ls -l may  
-rwxr--r--. 1 aagonchar aagonchar 0 map 13 10:51 may  
aagonchar@aagonchar:~$ chmod u-x may  
aagonchar@aagonchar:~$ ls -l may  
-rw-r--r--. 1 aagonchar aagonchar 0 map 13 10:51 may  
aagonchar@aagonchar:~$ chmod g-r,o-r monthly/  
aagonchar@aagonchar:~$ chmod g+w abc1  
aagonchar@aagonchar:~$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

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

Рис. 4: Работа с каталогами

Работа с командой chmod

```
aagonchar@aagonchar:~$  
aagonchar@aagonchar:~$ mkdir australia play  
aagonchar@aagonchar:~$ touch my_os feathers  
aagonchar@aagonchar:~$ chmod 744 australia/  
aagonchar@aagonchar:~$ chmod 711 play/  
aagonchar@aagonchar:~$ chmod 544 my_os  
aagonchar@aagonchar:~$ chmod 664 feathers  
aagonchar@aagonchar:~$ ls -l  
итого 0  
-rw-rw-r--. 1 aagonchar aagonchar 0 map 13 10:54 abc1  
drwxr--r--. 1 aagonchar aagonchar 0 map 13 10:55 australia  
-rw-rw-r--. 1 aagonchar aagonchar 0 map 13 10:55 feathers  
drwxr-xr-x. 1 aagonchar aagonchar 74 фев 25 11:39 git-extended  
-rw-r--r--. 1 aagonchar aagonchar 0 map 13 10:51 may  
drwx--x--x. 1 aagonchar aagonchar 24 map 13 10:49 monthly  
-r-xr--r--. 1 aagonchar aagonchar 0 map 13 10:55 my_os  
drwx--x--x. 1 aagonchar aagonchar 0 map 13 10:55 play  
drwxr-xr-x. 1 aagonchar aagonchar 14 map 13 10:50 reports  
drwxr-xr-x. 1 aagonchar aagonchar 28 map 13 10:55 ski.places  
drwxr-xr-x. 1 aagonchar aagonchar 10 фев 25 11:10 work  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 Видео  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 Документы  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 Загрузки  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 Изображения  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 Музыка  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 Общедоступные  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 'Рабочий стол'  
drwxr-xr-x. 1 aagonchar aagonchar 0 фев 25 10:56 Шаблоны  
aagonchar@aagonchar:~$
```

```
aagonchar@aagonchar:~ — 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
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
geoclue:x:999:999:User for geoclue:/var/lib/geoclue:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
systemd-oom:x:998:998:systemd Userspace OOM Killer:/:/usr/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
polkitd:x:114:114:User for polkitd:/:/sbin/nologin
```

Рис. 6: Файл /etc/passwd

Работа с файлами и правами доступа

```
aagonchar@aagonchar:~$  
aagonchar@aagonchar:~$ cp feathers file.old  
aagonchar@aagonchar:~$ mv file.old play  
aagonchar@aagonchar:~$ mkdir fun  
aagonchar@aagonchar:~$ cp -r play/fun  
cp: после 'play/fun' пропущен операнд, задающий целевой файл  
По команде «cp --help» можно получить дополнительную информацию.  
aagonchar@aagonchar:~$ cp -r play/ fun  
aagonchar@aagonchar:~$ mv fun/ play/games  
aagonchar@aagonchar:~$ chmod u-r feathers  
aagonchar@aagonchar:~$ cat feathers  
cat: feathers: Отказано в доступе  
aagonchar@aagonchar:~$ cp feathers feathers  
cp: 'feathers' и 'feathers' - один и тот же файл  
aagonchar@aagonchar:~$ cp feathers feathers2  
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе  
aagonchar@aagonchar:~$ chmod u+r feathers  
aagonchar@aagonchar:~$ chmod u-x play/  
aagonchar@aagonchar:~$ cd play/  
bash: cd: play/: Отказано в доступе  
aagonchar@aagonchar:~$ chmod u+x play/  
aagonchar@aagonchar:~$
```

Рис. 7: Работа с файлами и правами доступа

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

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-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.

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

```
aagonchar@aagonchar:~ — 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=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
        Usage or syntax error

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

```
aagonchar@aagonchar:~ — 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.

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

```
aagonchar@aagonchar:~ — 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
            All processes in the current process group are signaled.

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


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.