

Операционные системы

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

Татьяна Соколова

11 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

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

Рис. 1: Выполнение примеров

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

Рис. 2: Выполнение примеров

```
tssokolova@tssokolova:~$  
tssokolova@tssokolova:~$ touch may  
tssokolova@tssokolova:~$ ls -l may  
-rw-r--r--. 1 tssokolova tssokolova 0 map 11 12:20 may  
tssokolova@tssokolova:~$ chmod u+x may  
tssokolova@tssokolova:~$ ls -l may  
-rwxr--r--. 1 tssokolova tssokolova 0 map 11 12:20 may  
tssokolova@tssokolova:~$ chmod u-x may  
tssokolova@tssokolova:~$ ls -l may  
-rw-r--r--. 1 tssokolova tssokolova 0 map 11 12:20 may  
tssokolova@tssokolova:~$ mkdir monthly  
mkdir: невозможно создать каталог «monthly»: Файл существует  
tssokolova@tssokolova:~$ chmod g-r,o-r monthly  
tssokolova@tssokolova:~$ touch abc1  
tssokolova@tssokolova:~$ chmod g+w abc1  
tssokolova@tssokolova:~$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

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

Рис. 4: Работа с каталогами

Работа с командой chmod

```
tssokolova@tssokolova:~$ mkdir australia play
mkdir: невозможно создать каталог «australia»: Файл существует
mkdir: невозможно создать каталог «play»: Файл существует
tssokolova@tssokolova:~$ touch my_os feathers
tssokolova@tssokolova:~$ chmod 744 australia/
tssokolova@tssokolova:~$ chmod 711 play/
tssokolova@tssokolova:~$ chmod 544 my_os
tssokolova@tssokolova:~$ chmod 664 feathers
tssokolova@tssokolova:~$ ls -l
итого 0
-rw-rw-r--. 1 tssokolova tssokolova 0 мар 11 12:21 abc1
drwxr--r--. 1 tssokolova tssokolova 0 мар 11 12:24 australia
-rw-rw-r--. 1 tssokolova tssokolova 0 мар 11 12:24 feathers
drwxr-xr-x. 1 tssokolova tssokolova 74 фев 11 16:22 git-extended
-rw-r--r--. 1 tssokolova tssokolova 0 мар 11 12:20 may
drwx--x--x. 1 tssokolova tssokolova 24 мар 11 12:18 monthly
-r-xr--r--. 1 tssokolova tssokolova 0 мар 11 12:24 my_os
drwx--x--x. 1 tssokolova tssokolova 0 мар 11 12:24 play
drwxr-xr-x. 1 tssokolova tssokolova 14 мар 11 12:20 reports
drwxr-xr-x. 1 tssokolova tssokolova 48 мар 6 11:20 site
drwxr-xr-x. 1 tssokolova tssokolova 28 мар 11 12:22 ski.plases
drwx-----. 1 tssokolova tssokolova 8 мар 6 11:34 snap
drwxr-xr-x. 1 tssokolova tssokolova 10 фев 11 15:19 work
drwxr-xr-x. 1 tssokolova tssokolova 0 фев 11 15:06 Видео
drwxr-xr-x. 1 tssokolova tssokolova 0 фев 11 15:06 Документы
drwxr-xr-x. 1 tssokolova tssokolova 54 фев 11 16:19 Загрузки
drwxr-xr-x. 1 tssokolova tssokolova 0 фев 11 15:06 Изображения
drwxr-xr-x. 1 tssokolova tssokolova 0 фев 11 15:06 Музыка
drwxr-xr-x. 1 tssokolova tssokolova 0 фев 11 15:06 Общедоступные
drwxr-xr-x. 1 tssokolova tssokolova 0 фев 11 15:06 'Рабочий стол'
drwxr-xr-x. 1 tssokolova tssokolova 0 фев 11 15:06 Шаблоны
tssokolova@tssokolova:~$
```

```
rtkit:x:172:172:RealtimeKit:/:sbin/nologin
chrony:x:997:994:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:996:993:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
gluster:x:995:992:GlusterFS daemons:/run/gluster:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
pipewire:x:994:991:PipeWire System Daemon:/run/pipewire:/usr/sbin/nologin
unbound:x:993:990:Unbound DNS resolver:/var/lib/unbound:/sbin/nologin
nm-openconnect:x:992:989:NetworkManager user for OpenConnect:/:sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
wsdd:x:991:988:Web Services Dynamic Discovery host daemon:/:sbin/nologin
sssd:x:990:986:User for sssd:/run/sss:/sbin/nologin
openvpn:x:989:985:OpenVPN:/etc/openvpn:/sbin/nologin
nm-openvpn:x:988:984:Default user for running openvpn spawned by NetworkManager:/:sbin/nologin
flatpak:x:987:983:Flatpak system helper:/:usr/sbin/nologin
colord:x:986:982:User for colord:/var/lib/colord:/sbin/nologin
abrt:x:173:173:/:etc/abrt:/sbin/nologin
gdm:x:42:42:GNOME Display Manager:/var/lib/gdm:/usr/sbin/nologin
gnome-initial-setup:x:985:981:/:run/gnome-initial-setup:/sbin/nologin
/etc/passwd
```

Рис. 6: Файл /etc/passwd

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

Рис. 7: Работа с файлами и правами доступа

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*.

If only the directory or the device is given, for example:

`mount /dir`

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

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
- 32 Checking canceled by user request

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

```
tssokolova@tssokolova:~ — 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.)
```

```
tssokolova@tssokolova:~ — 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] [--] pidname...

    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.

        -1
            All processes with a PID larger than 1 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.