

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

Илья Колонтырский¹

16 марта, 2024, 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 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

Выполнение примеров

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

Рис. 1: Выполнение примеров

Выполнение примеров

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

Рис. 2: Выполнение примеров

Выполнение примеров

```
irkolontirskiy@irkolontirskiy:~$  
irkolontirskiy@irkolontirskiy:~$ ls -l may  
-rw-r--r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:21 may  
irkolontirskiy@irkolontirskiy:~$ chmod u+x may  
irkolontirskiy@irkolontirskiy:~$ ls -l may  
-rwxr--r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:21 may  
irkolontirskiy@irkolontirskiy:~$ chmod u-x may  
irkolontirskiy@irkolontirskiy:~$ ls -l may  
-rw-r--r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:21 may  
irkolontirskiy@irkolontirskiy:~$ chmod g-r,o-r monthly/  
irkolontirskiy@irkolontirskiy:~$ chmod g+w abc1  
irkolontirskiy@irkolontirskiy:~$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

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

Рис. 4: Работа с каталогами

Работа с командой chmod

```
irkolontirskiy@irkolontirskiy:~$  
irkolontirskiy@irkolontirskiy:~$ mkdir australia play  
irkolontirskiy@irkolontirskiy:~$ touch my_os feathers  
irkolontirskiy@irkolontirskiy:~$ chmod 744 australia/  
irkolontirskiy@irkolontirskiy:~$ chmod 711 play/  
irkolontirskiy@irkolontirskiy:~$ chmod 544 my_os  
irkolontirskiy@irkolontirskiy:~$ chmod 664 feathers  
irkolontirskiy@irkolontirskiy:~$ ls -l  
итого 0  
-rw-rw-r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:25 abc1  
drwxr--r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:26 australia  
-rw-rw-r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:26 feathers  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 74 map 6 13:05 git-extended  
-rw-r--r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:21 may  
drwx--x--x. 1 irkolontirskiy irkolontirskiy 24 map 16 17:21 monthly  
-r-xr--r--. 1 irkolontirskiy irkolontirskiy 0 map 16 17:26 my_os  
drwx--x--x. 1 irkolontirskiy irkolontirskiy 0 map 16 17:26 play  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 14 map 16 17:23 reports  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 28 map 16 17:26 ski.places  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 10 map 6 12:56 work  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 Видео  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 Документы  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 Загрузки  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 Изображения  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 Музыка  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 Общедоступные  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 'Рабочий стол'  
drwxr-xr-x. 1 irkolontirskiy irkolontirskiy 0 map 6 12:27 Шаблоны  
irkolontirskiy@irkolontirskiy:~$
```

Рис. 5: Настройка прав доступа

```
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
systemd-coredump:x:998:998:systemd Core Dumper:/usr/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
```

Рис. 6: Файл /etc/passwd

Работа с файлами и правами доступа

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

Рис. 7: Работа с файлами и правами доступа

Справка по командам

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

Рис. 8: Команда mount

```
ESCK(8)                                     System Administration                                     ESCK(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

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

Рис. 9: Команда fsck

```
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.

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

Рис. 10: Команда mkfs

Справка по командам

```
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.

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.