

Операционные системы

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

Бельчуг Александр Константинович

14 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

```
[akbeljchug@vbox ~]$ touch abc1
[akbeljchug@vbox ~]$ cp abc1 april
[akbeljchug@vbox ~]$ cp abc1 may
[akbeljchug@vbox ~]$ mkdir monthly
[akbeljchug@vbox ~]$ cp april may monthly
[akbeljchug@vbox ~]$ cp monthly/may monthly/june
[akbeljchug@vbox ~]$ ls monthly
april  june  may
[akbeljchug@vbox ~]$ mkdir monthly.00
[akbeljchug@vbox ~]$ cp -r monthly monthly.00
[akbeljchug@vbox ~]$ cp -r monthly.00 /tmp
[akbeljchug@vbox ~]$
```

Рис. 1: Выполнение примеров

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

Рис. 2: Выполнение примеров

```
[akbeljchug@vbox ~]$ touch may
[akbeljchug@vbox ~]$ ls -l may
-rw-r--r--. 1 akbeljchug akbeljchug 0 map 14 15:19 may
[akbeljchug@vbox ~]$ chmod u+x may
[akbeljchug@vbox ~]$ ls -l may
-rwxr--r--. 1 akbeljchug akbeljchug 0 map 14 15:19 may
[akbeljchug@vbox ~]$ chmod u-x may
[akbeljchug@vbox ~]$ ls -l may
-rw-r--r--. 1 akbeljchug akbeljchug 0 map 14 15:19 may
[akbeljchug@vbox ~]$ chmod g-r,o-r monthly
[akbeljchug@vbox ~]$ chmod g+w abc1
[akbeljchug@vbox ~]$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

```
[akbeljchug@vbox ~]$ cp /usr/include/gdb/jit-reader.h -
[akbeljchug@vbox ~]$ mv jit-reader.h equipment
mv: не удалось выполнить stat для 'jit-reader.h': Нет такого файла или каталога
[akbeljchug@vbox ~]$ mv - equipment
[akbeljchug@vbox ~]$ mkdir ski.plases
[akbeljchug@vbox ~]$ mv equipment ski.plases/
[akbeljchug@vbox ~]$ mv ski.plases/equipment ski.plases/equiplist
[akbeljchug@vbox ~]$ touch abc1
[akbeljchug@vbox ~]$ cp abc1 ski.plases/equiplist2
[akbeljchug@vbox ~]$ cd ski.plases/
[akbeljchug@vbox ski.plases]$ mkdir equipment
[akbeljchug@vbox ski.plases]$ mv equiplist equipment/
[akbeljchug@vbox ski.plases]$ mv equiplist2 equipment/
mv: невозможно переместить 'equiplist2' в 'equipment/': Это не каталог
[akbeljchug@vbox ski.plases]$ mv equiplist2 equipment/
mv: невозможно переместить 'equiplist2' в 'equipment/': Это не каталог
[akbeljchug@vbox ski.plases]$ mkdir newdir
[akbeljchug@vbox ski.plases]$ mv newdir ski.plases/
[akbeljchug@vbox ski.plases]$ mv ski.plases/newdir/ ski.plases/plans
mv: не удалось выполнить stat для 'ski.plases/newdir/': Нет такого файла или каталога
[akbeljchug@vbox ski.plases]$ mkdir newdir
[akbeljchug@vbox ski.plases]$ mv ski.plases/newdir/ ski.plases/plans
mv: не удалось выполнить stat для 'ski.plases/newdir/': Нет такого файла или каталога
[akbeljchug@vbox ski.plases]$ mv ski.plases/newdir/ ski.plases/plans
[akbeljchug@vbox ski.plases]$
```

Рис. 4: Работа с каталогами

```
[akbeljchug@vbox ski.plases]$ mkdir australia play
[akbeljchug@vbox ski.plases]$ touch my_os feathers
[akbeljchug@vbox ski.plases]$ chmod 744 australia/
[akbeljchug@vbox ski.plases]$ chmod 744 play/
[akbeljchug@vbox ski.plases]$ chmod 711 play/
[akbeljchug@vbox ski.plases]$ chmod 544 my_os
[akbeljchug@vbox ski.plases]$ chmod 664 feathers
[akbeljchug@vbox ski.plases]$ ls -l
итого 0
drwxr--r--. 1 akbeljchug akbeljchug  0 map 14 16:19 australia
drwxr-xr-x. 1 akbeljchug akbeljchug 38 map 14 16:14 equipment
-rw-rw-r--. 1 akbeljchug akbeljchug  0 map 14 16:20 feathers
-r-xr--r--. 1 akbeljchug akbeljchug  0 map 14 16:20 my_os
drwx--x--x. 1 akbeljchug akbeljchug  0 map 14 16:19 play
drwxr-xr-x. 1 akbeljchug akbeljchug 10 map 14 16:14 ski.plases
[akbeljchug@vbox ski.plases]$
```

Рис. 5: Настройка прав доступа

```
[akbel]chug@vbox ski.plases]$ cat /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
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
systemd-oom:x:998:998:systemd Userspace OOM Killer:/:/usr/sbin/nologin
polkitd:x:114:114>User for polkitd:/:/sbin/nologin
sstpc:x:997:995:Secure Socket Tunneling Protocol (SSTP) Client:/var/run/sstpc:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/:/sbin/nologin
chrony:x:996:994:chrony system user:/var/lib/chrony:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
pipewire:x:995:993:PipeWire System Daemon:/run/pipewire:/usr/sbin/nologin
sssd:x:994:992>User for sssd:/run/sss:/sbin/nologin
unbound:x:993:991:Unbound DNS resolver:/var/lib/unbound:/sbin/nologin
nm-openconnect:x:992:990:NetworkManager user for OpenConnect:/:/sbin/nologin
wssd:x:991:989:Web Services Dynamic Discovery host daemon:/:/sbin/nologin
openvpn:x:990:988:OpenVPN:/etc/openvpn:/sbin/nologin
nm-openvpn:x:989:987:Default user for running openvpn spawned by NetworkManager:/:/sbin/nologin
colord:x:988:986>User for colord:/var/lib/colord:/sbin/nologin
abrt:x:173:173:/:etc/abrt:/sbin/nologin
setroubleshoot:x:987:985:SELinux troubleshoot server:/var/lib/setroubleshoot:/usr/sbin/nologin
sddm:x:986:984:SDDM Greeter Account:/var/lib/sddm:/usr/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
vboxadd:x:985:1:/:var/run/vboxadd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
dnsmasq:x:984:982:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
tcpdump:x:72:72:tcpdump:/:/usr/sbin/nologin
passim:x:980:980:Local Caching Server:/usr/share/empty:/usr/sbin/nologin
```

Рис. 6: Файл /etc/passwd

Работа с файлами и правами доступа

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

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 optional] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o optional] device mountpoint

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

    mount --make-[shared|slave|private|unbindable|rshared|rslave|sprivate|unbindable] 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

    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 ambiguous interpretation of the given argument. For example:

        mount --target /mountpoint

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

Рис. 8: Команда mount

```
fsck(8)                                     System Administration                               fsck(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRtMNP] [-x [fs]] [-C [fs]] [-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/sda1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or a filesystem label or UUID specifier (e.g., UUID=8866bab6-88c5-4a83-9b88-bfc2405777bd 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
    128    Shared-library error

    The exit status returned when multiple filesystems are checked is the bit-wise OR of the exit statuses for each filesystem that is checked.

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.<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.

    -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.)

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@ods.com>, Fred N. van Kempen <waltje@uwalnt.nl.mugnet.org>, Ron Sommeling <sommel@sci.kun.nl>.

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

Рис. 10: Команда mkfs

```
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:
        0
            where p is larger than 0. The process with PID p is signaled.
        0
            All processes in the current process group are signaled.
        -1
            All processes with a PID larger than 1 are signaled.
        -0
            where p is larger than 1. All processes in process group p are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either a signal must be specified first, or the argument must be preceded by a '--' option, otherwise it will be taken as the signal to send.
NAME
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.