

Операционные системы

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

Останин Владислав Александрович

29 мая 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

Выполнение примеров

```
vlad@worker-node2:~$ touch abc1
vlad@worker-node2:~$ cp abc1 april
vlad@worker-node2:~$ cp abc1 may
vlad@worker-node2:~$ mkdir monthly
vlad@worker-node2:~$ cp april may monthly
vlad@worker-node2:~$ cp monthly/may monthly/june
vlad@worker-node2:~$ ls monthly
april  june  may
vlad@worker-node2:~$ mkdir monthly.00
vlad@worker-node2:~$ cp -r monthly monthly.00
vlad@worker-node2:~$ cp -r monthly.00 /tmp
vlad@worker-node2:~$
```

Выполнение примеров

```
vlad@worker-node2:~$ mv april july
vlad@worker-node2:~$ mv july monthly.00
vlad@worker-node2:~$ ls monthly.00
july  monthly
vlad@worker-node2:~$ mv monthly.00 monthly.01
vlad@worker-node2:~$ mkdir reports
vlad@worker-node2:~$ mv monthly.01 reports
vlad@worker-node2:~$ mv reports/monthly.01 reports/monthly
vlad@worker-node2:~$
```

Рис. 2: Выполнение примеров

Выполнение примеров

```
@worker-node2:~$ touch may
@worker-node2:~$ ls -l may
-rw-r-- 1 vlad vlad 0 июн 16 18:40 may
@worker-node2:~$ chmod u+x may
@worker-node2:~$ ls -l may
-rw-r-- 1 vlad vlad 0 июн 16 18:40 may
@worker-node2:~$ chmod u-x may
@worker-node2:~$ ls -l may
-rw-r-- 1 vlad vlad 0 июн 16 18:40 may
@worker-node2:~$ chmod g-r,o-r monthly
@worker-node2:~$ chmod g+w abc1
@worker-node2:~$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

```
vlad@worker-node2:~$ cp /usr/include/linux/sysinfo.h ~
vlad@worker-node2:~$ mv sysinfo.h equipment
vlad@worker-node2:~$ mkdir ski.places
vlad@worker-node2:~$ mv equipment ski.places/
vlad@worker-node2:~$ mv ski.places/equipment ski.places/equiplist
vlad@worker-node2:~$ touch abc1
vlad@worker-node2:~$ cp abc1 ski.places/equiplist2
vlad@worker-node2:~$ cd ski.places/
vlad@worker-node2:~/ski.places$ mkdir equipment
vlad@worker-node2:~/ski.places$ mv equiplist equipment/
vlad@worker-node2:~/ski.places$ mv equiplist2 equipment/
vlad@worker-node2:~/ski.places$ cd
vlad@worker-node2:~$ mkdir newdir
vlad@worker-node2:~$ mv newdir ski.places/
vlad@worker-node2:~$ mv ski.places/newdir/ ski.places/plans
vlad@worker-node2:~$ █
```

Работа с командой chmod

```
vlad@worker-node2:~$ mkdir australia play
vlad@worker-node2:~$ touch my_os feathers
vlad@worker-node2:~$ chmod 744 australia/
vlad@worker-node2:~$ chmod 711 play/
vlad@worker-node2:~$ chmod 544 my_os
vlad@worker-node2:~$ chmod 664 feathers

vlad@worker-node2:~$ ls -l
total 28
-rw-rw-r-- 1 vlad vlad    0 июн 16 18:41 abc1
drwxr--r-- 2 vlad vlad 4096 июн 16 18:43 australia
-rw-rw-r-- 1 vlad vlad    0 июн 16 18:43 feathers
-rw-rw-r-- 1 vlad vlad    0 июн 16 18:40 may
drwx-wx--x 2 vlad vlad 4096 июн 16 18:37 monthly
-r-xr--r-- 1 vlad vlad    0 июн 16 18:43 my_os
drwx--x--x 2 vlad vlad 4096 июн 16 18:43 play
drwxrwxr-x 5 vlad vlad 4096 июн 16 16:27 Pycharm
drwxrwxr-x 3 vlad vlad 4096 июн 16 18:39 reports
drwxrwxr-x 4 vlad vlad 4096 июн 16 18:42 ski.places
drwx----- 3 vlad vlad 4096 июн 15 21:01 snap

vlad@worker-node2:~$
```

```
yusufsubanov@yusufsubanov:~ — 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
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
```

Работа с файлами и правами доступа

```
vlad@worker-node2:~$ cp feathers file.old
vlad@worker-node2:~$ mv file.old play/
vlad@worker-node2:~$ mkdir fun
vlad@worker-node2:~$ v
v: command not found
vlad@worker-node2:~$ cp -R play/ fun/
vlad@worker-node2:~$ mv fun/ play/games
vlad@worker-node2:~$ cat feathers
vlad@worker-node2:~$ chmod u-r feathers
vlad@worker-node2:~$ cp feathers feathers2
cp: cannot open 'feathers' for reading: Permission denied
vlad@worker-node2:~$ chmod u+r feathers
vlad@worker-node2:~$ chmod u-x play/
vlad@worker-node2:~$ cd play/
bash: cd: play/: Permission denied
vlad@worker-node2:~$ chmod +x play/
vlad@worker-node2:~$ █
```

```
yusufsubanov@yusufsubanov:~ — 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

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

```
yusufsubanov@yusufsubanov:~ — 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

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

```
yusufsubanov@yusufsubanov:~ — 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.

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

```
yusufsubanov@yusufsubanov:~ -- 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

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.