

Отчет по лабораторной работе №5

Алмазова Елизавета Андреевна

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Цель работы и задание

Цель данной 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 (и работами), по проверке использования диска и обслуживанию файловой системы.

Задание:

Ход работы

Выполнение примеров

Я выполнила все примеры, приведенные в части описания лабораторной работы, в том числе создавала (`touch`, `mkdir`), копировала (`cp`), перемещала и переименовала (`mv`) файлы и каталоги `may`, `april`, `june`, `monthly`, `monthly.00`, `monthly.01`, `abc1`, а также изменяла (`chmod`) их права доступа (рис. 1).

```

[eaalmazova@fedora ~]$ cd
[eaalmazova@fedora ~]$ touch abc1
[eaalmazova@fedora ~]$ cp abc1 april
[eaalmazova@fedora ~]$ cp abc1 may
[eaalmazova@fedora ~]$ ls
abc1  april  bin  Desktop  Documents  Downloads  may  Music  Pictures  Public  Templates  Videos  work
[eaalmazova@fedora ~]$ mkdir monthly
[eaalmazova@fedora ~]$ cp april may monthly
[eaalmazova@fedora ~]$ cp monthly/may monthly/june
[eaalmazova@fedora ~]$ ls monthly
april  june  may
[eaalmazova@fedora ~]$ mkdir monthly.00
[eaalmazova@fedora ~]$ cp -r monthly monthly.00
[eaalmazova@fedora ~]$ cp -r monthly.00 /tmp
[eaalmazova@fedora ~]$ ls /tmp
monthly.00
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-chronyd.service-WnS0ZD
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-colord.service-nYffBu
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-dbus-broker.service-gj6auC
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-fwupd.service-RZeVlu
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-geoclue.service-CY8NDu
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-low-memory-monitor.service-LaoHG9
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-ModemManager.service-iJ2b5n
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-power-profiles-daemon.service-x1RxpA
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-rtkit-daemon.service-0Yqmie
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-switcheroo-control.service-L2IkjP
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-systemd-logind.service-hvEe6k
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-systemd-oomd.service-9z9SX5
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-systemd-resolved.service-l1qrNo
systemd-private-9b2ceb377cb14f85b617dce8b0eae181-upower.service-ksC1EA

```

Рисунок 1 - Создание abc1 и копирование в may и april, работа с каталогами monthly и monthly.00

Работа по созданию, перемещению и переименованию файлов и каталогов

Я скопировала файл `/usr/include/sys/io.h` в домашний каталог и назвала его `equipment`. В домашнем каталоге создала директорию `~/ski.places`, переместила туда файл `equipment` и переименовала в `equiplist`. Создала в домашнем каталоге файл `abc1` и скопировала его в каталог, назвав его `equiplist2`. Создала каталог с именем `equipment` в каталоге `~/ski.places`.

```

[eaalmazova@fedora ~]$ cp /usr/include/sys/io.h equipment
[eaalmazova@fedora ~]$ ls
abc1 bin Desktop Documents Downloads equipment may monthly Music Pictures Public reports Templates Videos work
[eaalmazova@fedora ~]$ mkdir /ski.places
mkdir: cannot create directory '/ski.places': Permission denied
[eaalmazova@fedora ~]$ mkdir ski.places
[eaalmazova@fedora ~]$ mv equipment ski.places
[eaalmazova@fedora ~]$ mv ski.places/equipment ski.places/equiplist
[eaalmazova@fedora ~]$ rm abc1
[eaalmazova@fedora ~]$ touch abc1
[eaalmazova@fedora ~]$ cp abc1 ski.places/equiplist2
[eaalmazova@fedora ~]$ mkdir ski.places/equipment
[eaalmazova@fedora ~]$ mv ski.places/equiplist ski.places/equiplist2 ski.places/equipment
[eaalmazova@fedora ~]$ ls
abc1 bin Desktop Documents Downloads may monthly Music Pictures Public reports ski.places Templates Videos work
[eaalmazova@fedora ~]$ ls ski.places
equipment
[eaalmazova@fedora ~]$ ls sli.places/equipment
ls: cannot access 'sli.places/equipment': No such file or directory
[eaalmazova@fedora ~]$ ls ski.places/equipment
equiplist equiplist2
[eaalmazova@fedora ~]$ mkdir newdir
[eaalmazova@fedora ~]$ mv newdir ski.places
[eaalmazova@fedora ~]$ ls ski.places
equipment newdir
[eaalmazova@fedora ~]$ mv ski.places/newdir ski.places/plans
[eaalmazova@fedora ~]$ ls ski.places
equipment plans

```

Рисунок 2 - Работа с ski.places, equipment, equiplist, equiplist2 и plans

Изменение прав доступа

Я создала каталоги `australia` и `play`, файлы `my_os` и `feathers`, а затем изменила с помощью команды `chmod` и соответствующих опций права доступа на требуемые: `chmod 744 australia`, `chmod 711 play`, `chmod 544 my_os`, `chmod 664 feathers` - и проверила результат с помощью `ls -l` (рис. 3).


```

[eaalmazova@fedora ~]$ chmod 744 australia
[eaalmazova@fedora ~]$ chmod 711 play
[eaalmazova@fedora ~]$ chmod 544 my_os
[eaalmazova@fedora ~]$ chmod 664 feathers
[eaalmazova@fedora ~]$ ls -l
total 0
-rw-rw-r--. 1 eaalmazova eaalmazova  0 May  6 18:37 abc1
drwxr--r--. 1 eaalmazova eaalmazova  0 May  6 18:41 australia
drwxr-xr-x. 1 eaalmazova eaalmazova  8 Apr 30 14:15 bin
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Desktop
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Documents
drwxr-xr-x. 1 eaalmazova eaalmazova 142 Apr 30 14:14 Downloads
-rw-rw-r--. 1 eaalmazova eaalmazova  0 May  6 18:41 feathers
-rw-rw-r--. 1 eaalmazova eaalmazova  0 May  6 18:28 may
drwx-wx--x. 1 eaalmazova eaalmazova  0 May  6 18:29 monthly
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Music
-r-xr--r--. 1 eaalmazova eaalmazova  0 May  6 18:41 my_os
drwxr-xr-x. 1 eaalmazova eaalmazova 234 Apr 22 23:35 Pictures
drwx--x--x. 1 eaalmazova eaalmazova  0 May  6 18:41 play
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Public
drwxrwxr-x. 1 eaalmazova eaalmazova 14 May  6 18:25 reports
drwxrwxr-x. 1 eaalmazova eaalmazova 28 May  6 18:40 ski.places
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Templates
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Videos
drwxrwxr-x. 1 eaalmazova eaalmazova 58 Apr 30 14:31 work

```

Рисунок 3 - Изменение прав доступа каталогов australia и play, файлов my_os и feathers

Работа по созданию, перемещению и переименованию файлов и каталогов

Я посмотрела содержимое файла `/etc/passwd`. Затем я скопировала `~/feathers` в файл `~/file.old`, переместила файл `~/file.old` в каталог `~/play`. Скопировала каталог `~/play` в каталог `~/fun`, переместила каталог `~/fun` в каталог `~/play` и назвала его `games`.

Работа с файлами и каталогами с измененными правами доступа

```
[eaalmazova@fedora ~]$ chmod u-r feathers
[eaalmazova@fedora ~]$ ls -l
total 0
-rw-rw-r--. 1 eaalmazova eaalmazova  0 May  6 18:37 abc1
drwxr--r--. 1 eaalmazova eaalmazova  0 May  6 18:41 australia
drwxr-xr-x. 1 eaalmazova eaalmazova  8 Apr 30 14:15 bin
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Desktop
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Documents
drwxr-xr-x. 1 eaalmazova eaalmazova 142 Apr 30 14:14 Downloads
--w-rw-r--. 1 eaalmazova eaalmazova  0 May  6 18:41 feathers
-rw-rw-r--. 1 eaalmazova eaalmazova  0 May  6 18:28 may
drwx-wx--x. 1 eaalmazova eaalmazova  0 May  6 18:29 monthly
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Music
-r-xr--r--. 1 eaalmazova eaalmazova  0 May  6 18:41 my_os
drwxr-xr-x. 1 eaalmazova eaalmazova 234 Apr 22 23:35 Pictures
drwx--x--x. 1 eaalmazova eaalmazova  26 May  6 18:50 play
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Public
drwxrwxr-x. 1 eaalmazova eaalmazova  14 May  6 18:25 reports
drwxrwxr-x. 1 eaalmazova eaalmazova  28 May  6 18:40 ski.places
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Templates
drwxr-xr-x. 1 eaalmazova eaalmazova  0 Apr 22 15:45 Videos
drwxrwxr-x. 1 eaalmazova eaalmazova  58 Apr 30 14:31 work
[eaalmazova@fedora ~]$ cat feathers
cat: feathers: Permission denied
[eaalmazova@fedora ~]$ cp feathers try
cp: cannot open 'feathers' for reading: Permission denied
```

Рисунок 4 - Отказ в доступе к файлу feathers

Команды mount, fsck, mkfs, kill

Прочитала с помощью команды man справки по командам mount, fsck, mkfs, kill (рис. 5,6,7,8). mount монтирует файловую систему, fsck проверяет и восстанавливает файловую систему Linux, mkfs создает файловую систему Linux, а kill передает процессу сигнал завершения.

```
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

    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:
```

Рисунок 5 - Man mount

```

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 an 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

    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.

```

Рисунок 6 - Man 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.

    -V, --version
        Display version information and exit. (Option -V will display version information only when it is the only parameter, otherwise it will work as --verbose.)

    -h, --help
        Display help text and exit.

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@uwal.nl.mugnet.org>, Ron Sommeling <sommel@sci.kun.nl>.

    The manual page was shamelessly adapted from Remy Card's version for the ext2 filesystem.

```

Рисунок 7 - Man 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:

        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.

        -n
            where n is larger than 1. All processes in process group n 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
        All processes invoked using this name will be signaled.

OPTIONS
    -s, --signal signal
        The signal to send. It may be given as a name or a number.

    -l, --list [number]
        Print a list of signal names, or convert the given signal number to a name. The signals can be found in /usr/include/linux/signal.h.

```

Рисунок 8 - Man kill

Выводы

В ходе выполнения данной лабораторной работы я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов, приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

Спасибо за внимание!