Операционные системы

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

Кайнова Екатерина Андреевна

16 марта 2025

Российский университет дружбы народов, Москва, Россия

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

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

Рис. 1: Выполнение примеров

Выполнение примеров

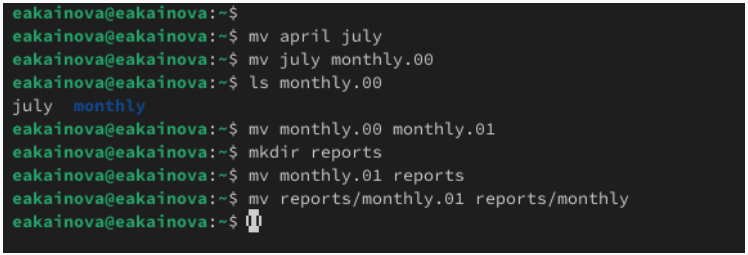


Рис. 2: Выполнение примеров

Выполнение примеров

```
eakainova@eakainova:~$
eakainova@eakainova:~$ touch may
eakainova@eakainova:~$ ls -l may
-rw-r--r--. 1 eakainova eakainova 0 мар 16 13:50 may
eakainova@eakainova:~$ chmod u+x may
eakainova@eakainova:~$ ls -l may
-rwxr--r--. 1 eakainova eakainova 0 мар 16 13:50 may
eakainova@eakainova:~$ chmod u-x may
eakainova@eakainova:~$ ls -l may
-rw-r--r--. 1 eakainova eakainova 0 мар 16 13:50 may
eakainova@eakainova:~$ chmod g-r,o-r monthly
eakainova@eakainova:~$ chmod g+w abc1
eakainova@eakainova:~$
```

Рис. 3: Выполнение примеров

Создание директорий и копирование файлов

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

Рис. 4: Работа с каталогами

Работа с командой chmod

```
eakainova@eakainova:~$ mkdir australia play
eakainova@eakainova:~$ touch my_os feathers
eakainova@eakainova:~$ chmod 744 australia/
eakainova@eakainova:~$ chmod 711 play/
eakainova@eakainova:~$ chmod 544 my_os
eakainova@eakainova:~$ chmod 664 feathers
eakainova@eakainova:~$ ls -l
итого 0
-rw-rw-r--. 1 eakainova eakainova  0 мар 16 13:51  abc1
drwxr--r--. 1 eakainova eakainova  0 мар 16 13:53  australia
-rw-rw-r--. 1 eakainova eakainova  0 мар 16 13:53  feathers
drwxr-xr-x. 1 eakainova eakainova 74 мар  5 16:22  git-extended
-rw-r--r--. 1 eakainova eakainova  0 мар 16 13:50  may
drwx--x--x. 1 eakainova eakainova 24 мар 16 13:50  monthly
-r-xr--r--. 1 eakainova eakainova  0 мар 16 13:53  my_os
drwx--x--x. 1 eakainova eakainova  0 мар 16 13:53  play
drwxr-xr-x. 1 eakainova eakainova 14 мар 16 13:50  reports
drwxr-xr-x. 1 eakainova eakainova 28 мар 16 13:51  ski.plases
drwxr-xr-x. 1 eakainova eakainova 10 мар  5 16:11  work
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35  Видео
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35  Документы
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35  Загрузки
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35  Изображения
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35  Музыка
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35  Общедоступные
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35 'Рабочий стол'
drwxr-xr-x. 1 eakainova eakainova  0 мар  5 15:35  Шаблоны
eakainova@eakainova:~$
```

Файл /etc/passwd

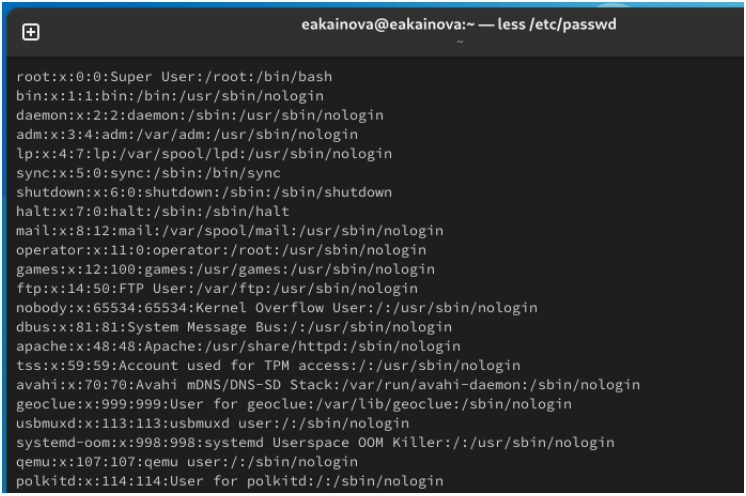


Рис. 6: Файл /etc/passwd

Работа с файлами и правами доступа

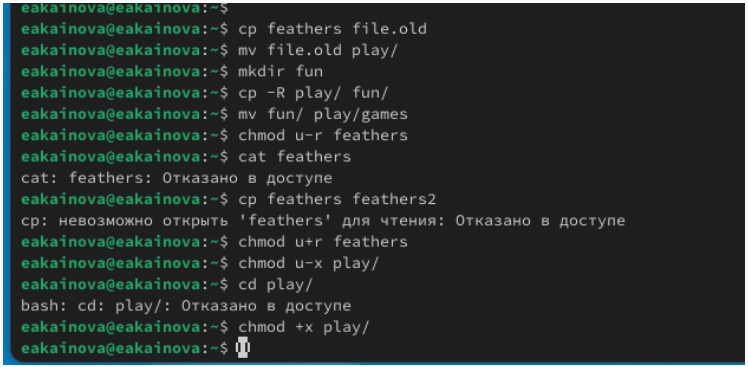
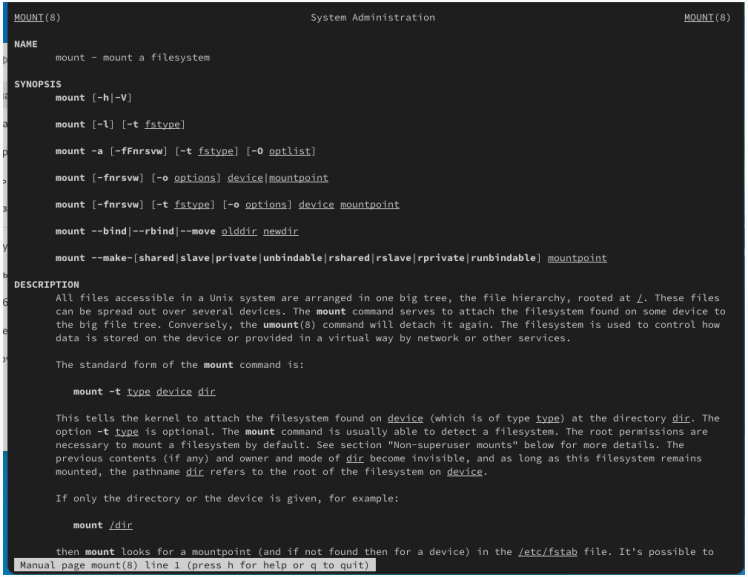


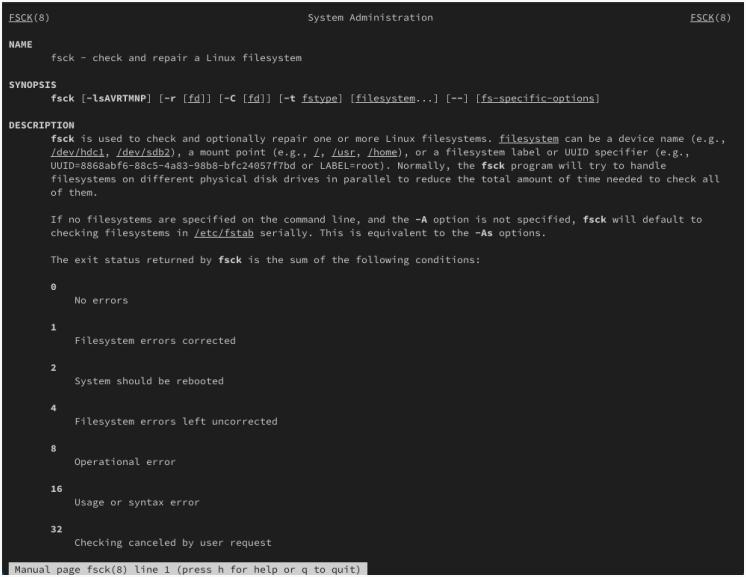
Рис. 7: Работа с файлами и правами доступа

Справка по командам



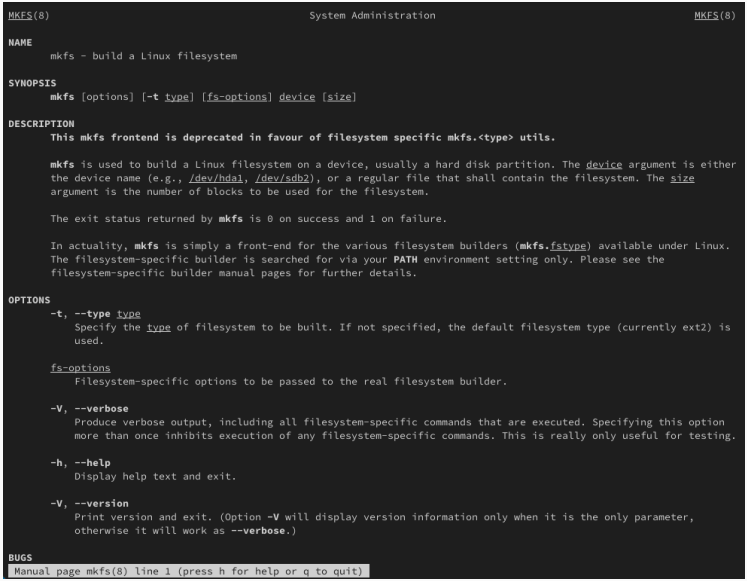
```
MOUNT(8)                        System Administration                        MOUNT(8)

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

Справка по командам



```
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)
```

Справка по командам



```
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.)

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

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.

           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

Вывод

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.