Лабораторная работа №7

Операционные системы

Башиянц А. К.

Российский университет дружбы народов, Москва, Россия

Вводная часть

Цели и задачи

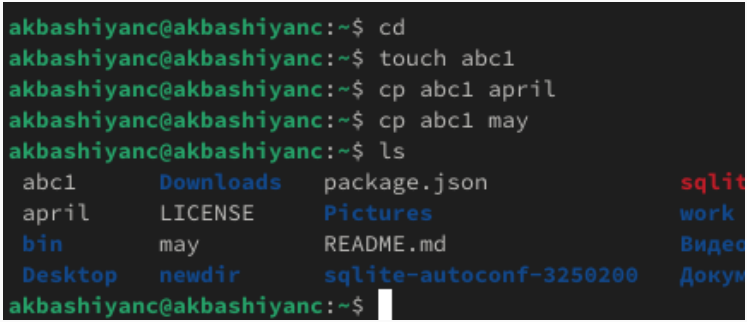
Цель данной работы — приобретение практических навыков работы с файловой системой.

- Создавать файлы;
- Копировать файлы и директории;
- Перемещать файлы и директории;
- Настраивать права доступа к файлам и директориям.

Выполнение лабораторной работы

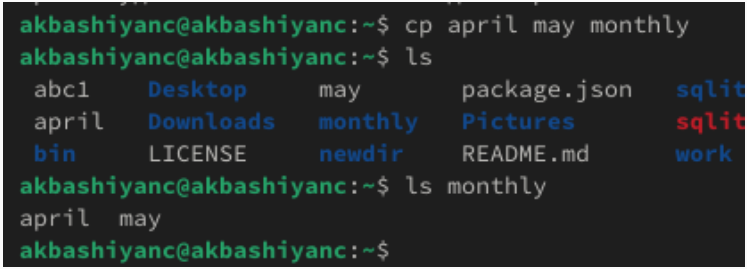
Копирование файла в текущем каталоге

Скопируем файл ~/abc1 в файл april и в файл may.



Копирование нескольких файлов в каталог

Скопируем файлы april и may в каталог monthly.



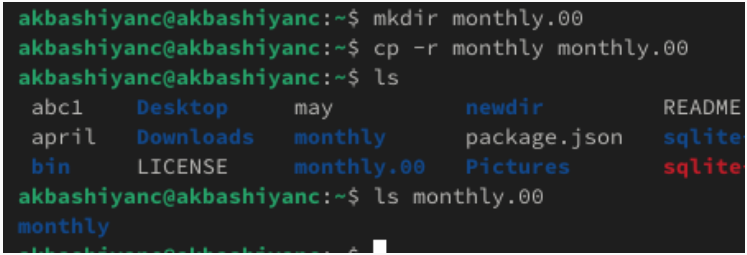
Копирование файлов в произвольном каталоге

Скопируем файл monthly/may в файл с именем june.

```
akbashiyanc@akbashiyanc:~$ cp monthly/may monthly/june
akbashiyanc@akbashiyanc:~$ ls monthly
april  june  may
akbashiyanc@akbashiyanc:~$
```

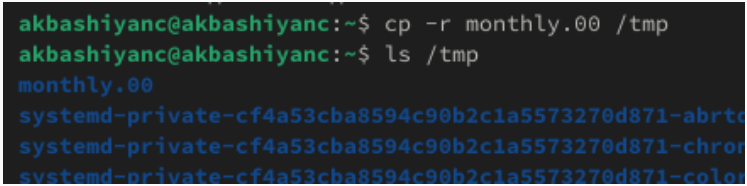
Копирование каталогов в текущем каталоге

Скопируем каталог monthly в каталог monthly.00.



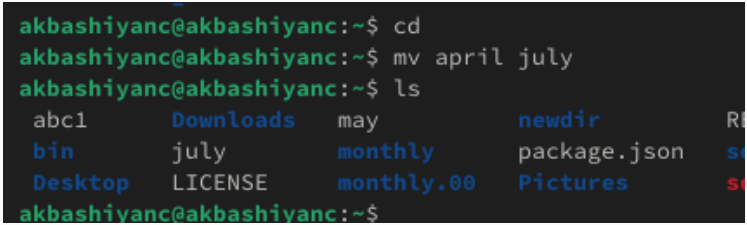
Копирование каталогов в произвольном каталоге

Скопируем каталог monthly.00 в каталог /tmp.



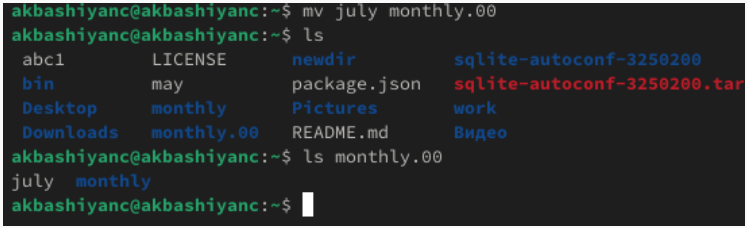
Переименование файлов в текущем каталоге

Изменим название файла april на july в домашнем каталоге.



Перемещение файлов в другой каталог

Переместим файл july в каталог monthly.00.



Переименование каталогов в текущем каталоге

Переименовать каталог monthly.00 в monthly.01.

```
akbashiyanc@akbashiyanc:~$ mv monthly.00 monthly.01
akbashiyanc@akbashiyanc:~$ ls
 abc1          LICENSE       newdir          sqlite-autoconf-3250200
 bin           may           package.json    sqlite-autoconf-3250200.tar
 Desktop       monthly       Pictures        work
 Downloads     monthly.01    README.md       Видео
akbashiyanc@akbashiyanc:~$
```

Перемещение каталога в другой каталог

Переместим кереименование каталогов в таталог monthly.01 в каталог reports.

```
akbashiyanc@akbashiyanc:~$ mkdir reports
akbashiyanc@akbashiyanc:~$ ^[[200~mv monthly.01 reports
bash: mv: команда не найдена...
akbashiyanc@akbashiyanc:~$ ~^C
akbashiyanc@akbashiyanc:~$ ^C
akbashiyanc@akbashiyanc:~$ mv monthly.01 reports
akbashiyanc@akbashiyanc:~$ ls
 abc1       Downloads   monthly         Pictures    sqlite-autoconf-32
 bin        LICENSE     newdir          README.md   sqlite-autoconf-32
 Desktop    may         package.json    reports     work
akbashiyanc@akbashiyanc:~$ ;s reports
bash: синтаксическая ошибка рядом с неожиданным маркером «;»
akbashiyanc@akbashiyanc:~$ ls reports
monthly.01
akbashiyanc@akbashiyanc:~$
```

Переименование каталога, не являющегося текущим

Переименовать каталог reports/monthly.01 в reports/monthly.

```
akbashiyanc@akbashiyanc:~$ mv reports/monthly.01 reports/monthly
akbashiyanc@akbashiyanc:~$ ls reports
monthly
akbashiyanc@akbashiyanc:~$
```

Право выполнения для владельца

Создадим файл ~/may с правом выполнения для владельца.

```
akbashiyanc@akbashiyanc:~$ cd
akbashiyanc@akbashiyanc:~$ touch may
akbashiyanc@akbashiyanc:~$ ls -l may
-rw-r--r--. 1 akbashiyanc akbashiyanc 0 мар 24 09:18 may
akbashiyanc@akbashiyanc:~$ chmod u+x may
akbashiyanc@akbashiyanc:~$ ls -l may
-rwxr--r--. 1 akbashiyanc akbashiyanc 0 мар 24 09:18 may
akbashiyanc@akbashiyanc:~$
```

Лишение права

Лишим владельца файла ~/may права на выполнение.

```
akbashiyanc@akbashiyanc:~$ chmod u-x may
akbashiyanc@akbashiyanc:~$ ls -l may
-rw-r--r--. 1 akbashiyanc akbashiyanc 0 мар 24 09:18 may
akbashiyanc@akbashiyanc:~$
```

Запрет на чтение

Создадим каталог monthly с запретом на чтение для членов группы и всех остальных пользователей.

```
akbashiyanc@akbashiyanc:~$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
akbashiyanc@akbashiyanc:~$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
akbashiyanc@akbashiyanc:~$ chmod g-r o-r monthly
chmod: невозможно получить доступ к 'o-r': Нет такого файла или ката
akbashiyanc@akbashiyanc:~$ chmod g-r monthly
akbashiyanc@akbashiyanc:~$ chmod o-r monthly
```

Право записи для группы

Создадим файл ~/abc1 с правом записи для членов группы.

```
akbashiyanc@akbashiyanc:~$ touch abc1
akbashiyanc@akbashiyanc:~$ chmod g+w abc1
akbashiyanc@akbashiyanc:~$ ls -l abc1
-rw-rw-r--. 1 akbashiyanc akbashiyanc 0 мар 24 09:22 abc1
akbashiyanc@akbashiyanc:~$
```

Скопируем файл /usr/include/sys/io.h в домашний каталог и назовём его equipment. Для этого используем команду cp.

```
akbashiyanc@akbashiyanc:~$ cp /usr/include/sys/io.h ~/equipment
akbashiyanc@akbashiyanc:~$ ls
 abc1       Downloads   may          package.json   reports
 bin        equipment   monthly      Pictures       sqlite-autoconf-3250
 Desktop    LICENSE     newdir       README.md      sqlite-autoconf-3250
akbashiyanc@akbashiyanc:~$
```

mv equipment

Переместим файл equipment в каталог ~/ski.plases. Для этого используем команду mv.

```
akbashiyanc@akbashiyanc:~$ mv equipment ski.plases
akbashiyanc@akbashiyanc:~$ ls
 abc1       Downloads   monthly         Pictures     ski.plases
 bin        LICENSE     newdir          README.md    sqlite-autoconf-32
 Desktop    may         package.json    reports      sqlite-autoconf-32
akbashiyanc@akbashiyanc:~$ ls ski.plases
equipment
akbashiyanc@akbashiyanc:~$
```

mv equipment

Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist. Для этого используем команду mv.

```
akbashiyanc@akbashiyanc:~$ mv ~/ski.plases/equipment ~/ski.plases/equiplist
akbashiyanc@akbashiyanc:~$ ls ski.plases
equiplist
akbashiyanc@akbashiyanc:~$
```

Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.plases, назовём его equiplist2. Для этого используем команду cp.

```
akbashiyanc@akbashiyanc:~$ touch abc1
akbashiyanc@akbashiyanc:~$ cp abc1 ski.plases/equiplist2
akbashiyanc@akbashiyanc:~$ ls ski.plases
equiplist  equiplist2
akbashiyanc@akbashiyanc:~$
```

mkdir equipment

Создадим каталог с именем equipment в каталоге ~/ski.plases. Для этого используем команду mkdir.

```
akbashiyanc@akbashiyanc:~$ mkdir ski.plases/equipment
akbashiyanc@akbashiyanc:~$ ls ski.plases
equiplist  equiplist2  equipment
akbashiyanc@akbashiyanc:~$
```

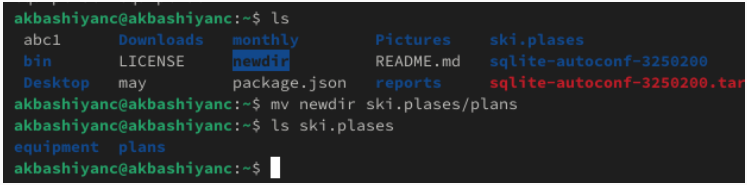
mv equipment

Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment. Для этого используем команду mv).

```
akbashiyanc@akbashiyanc:~$ mv ski.plases/equiplist ski.plases/equipment
akbashiyanc@akbashiyanc:~$ ls ski.plases
equiplist2  equipment
akbashiyanc@akbashiyanc:~$ mv ski.plases/equiplist2 ski.plases/equipment
akbashiyanc@akbashiyanc:~$ ls ski.plases
equipment
akbashiyanc@akbashiyanc:~$ ls ski.plases/equipment
equiplist  equiplist2
```

mv equipment

Создадим и переместим каталог ~/newdir в каталог ~/ski.plases и назовём его plans. Для этого используем команду mv.



chmod для директорий

Создадим директории australia и play, и с помощью chmod изменим права.

```
akbashiyanc@akbashiyanc:~$ mkdir australia
akbashiyanc@akbashiyanc:~$ chmod -R u=rwx,g=r,o=r australia
akbashiyanc@akbashiyanc:~$ ls -l  australia
итого 0
akbashiyanc@akbashiyanc:~$ ls -l australia
итого 0
akbashiyanc@akbashiyanc:~$ getfalc australia
bash: getfalc: команда не найдена...
Аналогичная команда: 'getfacl'
akbashiyanc@akbashiyanc:~$ mkdir play
akbashiyanc@akbashiyanc:~$ chmod -R u=rwx,g=x,o=x play
akbashiyanc@akbashiyanc:~$ touch my_os
```

chmod для файлов

Создадим файлы my_os и feathers, и с помощью chmod изменим права.

```
akbashiyanc@akbashiyanc:~$ chmod u=rx,g=r,o=r my_os
akbashiyanc@akbashiyanc:~$ chmod u=rw,g=rw,o=r features
```

cp feathers

Скопируем файл ~/feathers в файл ~/files.old. Для этого используем команду cp.

```
akbashiyanc@akbashiyanc:~$ cp features files.old
akbashiyanc@akbashiyanc:~$ ls
 abc1        Downloads   may             Pictures
 australia   features    monthly         play
 bin         files.old   my_os           README.md
 Desktop     LICENSE     package.json    reports
akbashiyanc@akbashiyanc:~$
```

mv files.old

Переместим файл ~/files.old в каталог ~/play. Для этого используем команду mv.

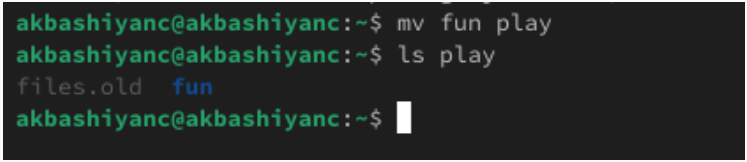
```
akbashiyanc@akbashiyanc:~$ mv files.old play
akbashiyanc@akbashiyanc:~$ ls play
files.old
akbashiyanc@akbashiyanc:~$
```

Скопируем каталог ~/play в каталог ~/fun. Для этого используем команду cp.

```
akbashiyanc@akbashiyanc:~$ cp -r play fun
akbashiyanc@akbashiyanc:~$ ls
 abc1        Downloads   may           Pictures
 australia   features    monthly       play
 bin         fun         my_os         README.md
 Desktop     LICENSE     package.json  reports
akbashiyanc@akbashiyanc:~$
```

mv fun

Переместим каталог ~/fun в каталог ~/play и назовём его games. Для этого используем команду mv.

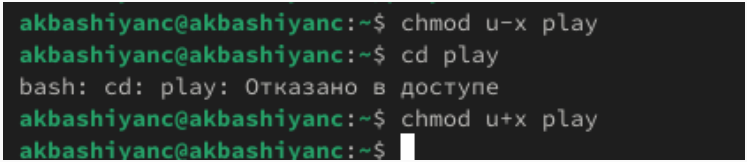


Лишим владельца файла ~/feathers права на чтение. Заметим, что теперь мы не можем читать файл. Вернем все обратно /

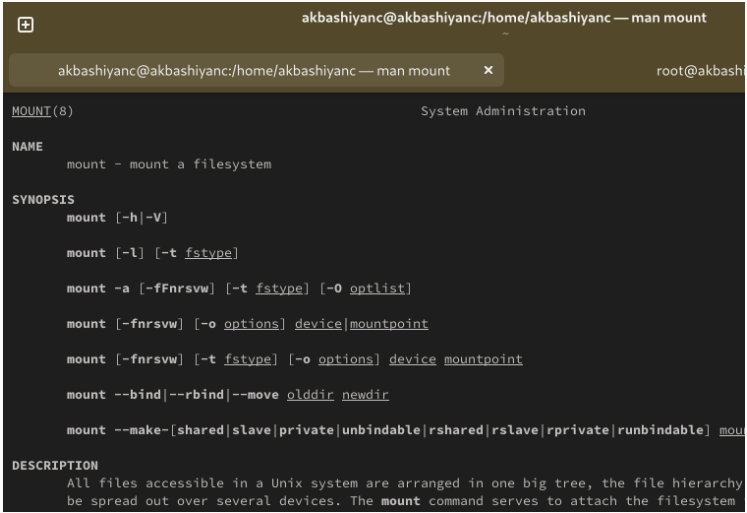
```
^[[A^C
akbashiyanc@akbashiyanc:~$ chmod u-r features
akbashiyanc@akbashiyanc:~$ cat features
cat: features: Отказано в доступе
akbashiyanc@akbashiyanc:~$ chmod u+r features
```

chmod play

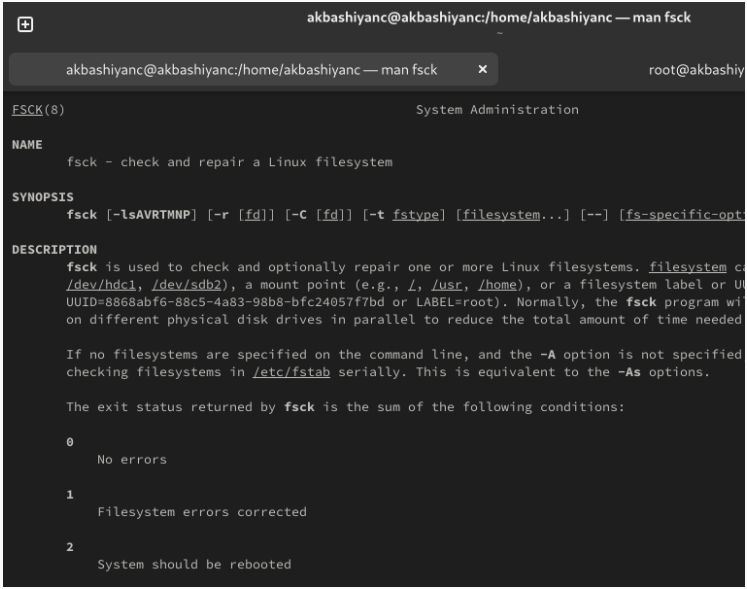
Лишим владельца каталога ~/play права на выполнение. Заметим, что теперь мы не можем выполнять команды с каталогом и файлами внутри. Вернем все обратно.



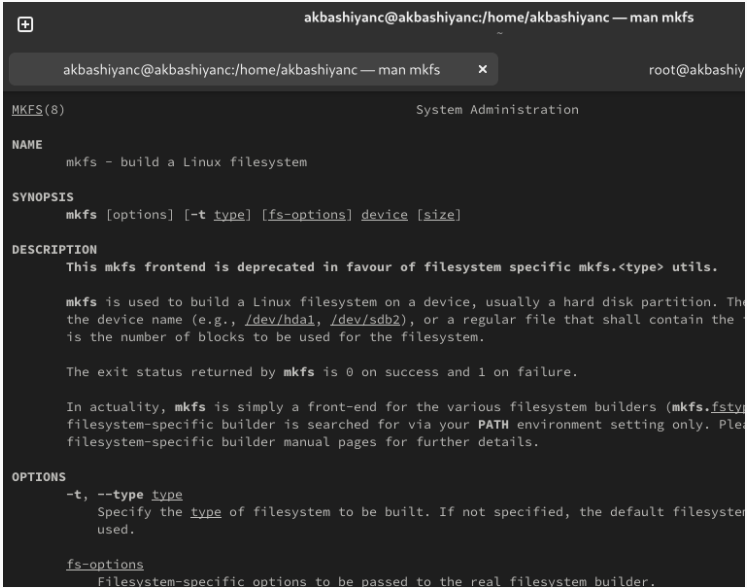
man mount


akbashiyanc@akbashiyanc:/home/akbashiyanc — man mount
akbashiyanc@akbashiyanc:/home/akbashiyanc — man mount
root@akbashi
MOUNT(8) System Administration
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]
DESCRIPTION
All files accessible in a Unix system are arranged in one big tree, the file hierarchy
be spread out over several devices. The mount command serves to attach the filesystem

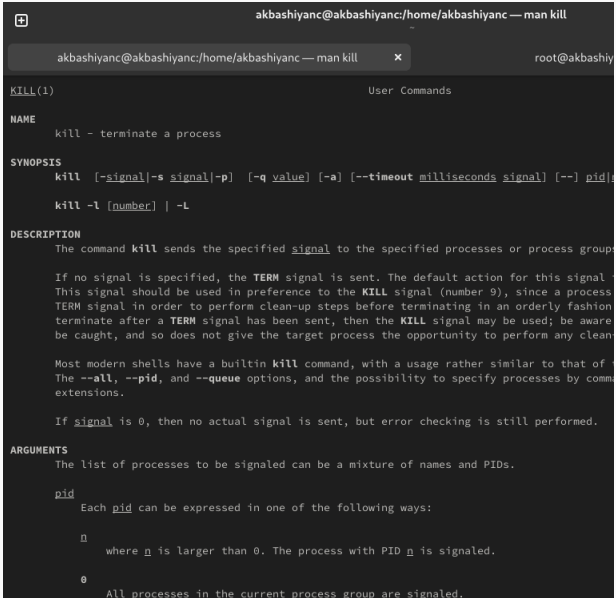
man fsck


akbashiyanc@akbashiyanc:/home/akbashiyanc — man fsck
akbashiyanc@akbashiyanc:/home/akbashiyanc — man fsck
root@akbashiy
FSCK(8) System Administration
NAME
fsck - check and repair a Linux filesystem
SYNOPSIS
fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-opti
DESCRIPTION
fsck is used to check and optionally repair one or more Linux filesystems. filesystem ca
/dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or a filesystem label or UU
UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck program wi
on different physical disk drives in parallel to reduce the total amount of time needed
If no filesystems are specified on the command line, and the -A option is not specified
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

man mkfs


akbashiyanc@akbashiyanc:/home/akbashiyanc — man mkfs
akbashiyanc@akbashiyanc:/home/akbashiyanc — man mkfs
root@akbashiy
MKFS(8) System Administration
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
the device name (e.g., /dev/hda1, /dev/sdb2), or a regular file that shall contain the
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.fsty
filesystem-specific builder is searched for via your PATH environment setting only. Ple
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 filesyste
used.
fs-options
Filesystem-specific options to be passed to the real filesystem builder.

man kill


akbashiyanc@akbashiyanc:/home/akbashiyanc — man kill
akbashiyanc@akbashiyanc:/home/akbashiyanc — man kill
root@akbashiy
KILL(1) User Commands
NAME
kill - terminate a process
SYNOPSIS
kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|
kill -l [number] | -L
DESCRIPTION
The command kill sends the specified signal to the specified processes or process group
If no signal is specified, the TERM signal is sent. The default action for this signal
This signal should be used in preference to the KILL signal (number 9), since a process
TERM signal in order to perform clean-up steps before terminating in an orderly fashion
terminate after a TERM signal has been sent, then the KILL signal may be used; be aware
be caught, and so does not give the target process the opportunity to perform any clean
Most modern shells have a builtin kill command, with a usage rather similar to that of
The --all, --pid, and --queue options, and the possibility to specify processes by comm
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.

Выводы

Выводы

- В этой лабораторной работе мы изучили работу с файловой системой.