Лабораторная работа №6

Журавлев Георгий Иванович

Цель 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 про- верке использования диска и обслуживанию файловой системы.

Выполнение лабораторной работы

1. Выполняем все примеры, приведённые в первой части описания лабораторной работы.

1.1. Копирование файла в текущем каталоге. Скопировать файл ~/abc1 в файл april и в файл may

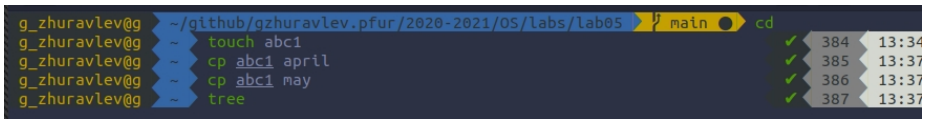


Рис. 1: Создание файлов и копирование

1.2. Копирование нескольких файлов в каталог. Скопировать файлы april и may в каталог monthly

1.3. Копирование файлов в произвольном каталоге. Скопировать файл monthly/may в файл с именем jun

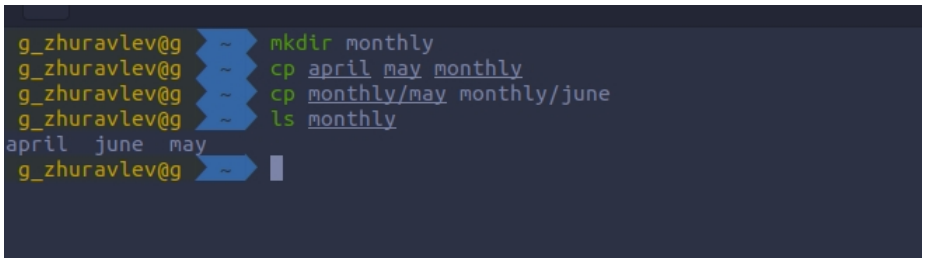


Рис. 2: Создание директории и копирование файлов

1.4. Копирование каталогов в текущем каталоге. Скопировать каталог monthly в каталог monthly.00
1.5. Копирование каталогов в произвольном каталоге. Скопировать каталог

Рис. 3: Копирование каталогов в новый каталог

1.6. Переименование файлов в текущем каталоге. Изменить название файла april на july в домашнем ка
1.7. Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00

1.8. Переименование каталогов в текущем каталоге. Переименовать каталог monthly.00 в monthly.01
1.9. Перемещение каталога в другой каталог. Переместить каталог monthly.01 в каталог reports
1.10. Переименование каталога, не являющегося текущим. Переименовать каталог reports/monthly.01 в

1.11. Требуется создать файл ~/may с правом выполнения для владельца
1.12. Требуется лишить владельца файла ~/may права на выполнение

1.13. Требуется создать каталог monthly с запретом на чтение для членов группы и всех остальных поль

1.14. Требуется создать файл ~/abc1 с правом записи для членов группы.

2. Выполним следующие действия.

2.1. Скопируем файл /usr/include/search.h в домашний каталог и назовём его equipment.

2.2. В домашнем каталоге создадим директорию ~/ski.plases.
2.3. Переместим файл equipment в каталог ~/ski.plases.
2.4. Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist.
2.5. Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.plases, назовём его equipli
2.6. Создадим каталог с именем equipment в каталоге ~/ski.plases.
2.7. Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment.

2.8. Создадим и переместим каталог ~/newdir в каталог ~/ski.plases и назовём его plans.

3. Определим опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет.

начальные значения: ---------
3.1. Создаем каталоги и файлы.
3.2. изменяем права командой chmod.

1. for australia: mkdir australia(d) -> chmod u+r, u+w, u+x, g+r, o+r australia
2. for play: mkdir play(d)-> chmod u+r, u+w, u+x, g+x, o+x play
3. for my_os: touch my_os(-) -> chmod u+r, u+x, g+r, o+r my_os
4. for feathers: touch feathers -> chmod u+r, u+w, g+r, g+w, o+r feathers

Рис. 4: Переименовываем и перемещаем файл

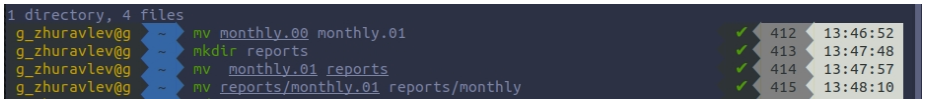


Рис. 5: Переименовываем с помощью mv

Рис. 6: Создание файла

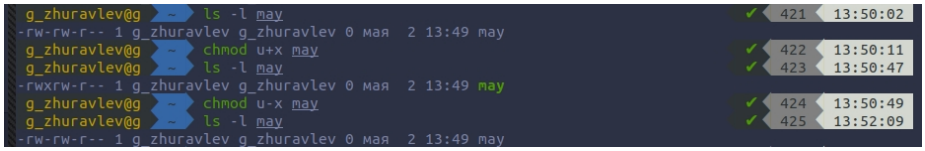


Рис. 7: Изменение прав

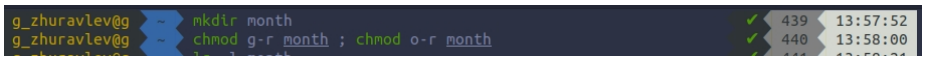


Рис. 8: Изменение прав на чтение каталога

Рис. 9: Изменение прав на запись файла

Рис. 10: Копирование каталога

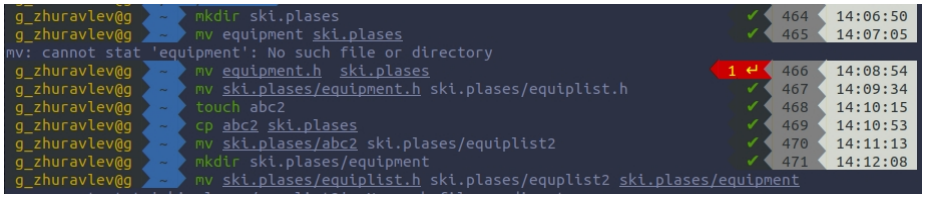


Рис. 11: Выполнение вышеописаных шагов

Рис. 12: подтверждение результата

Рис. 13: Создание и перемещение каталога

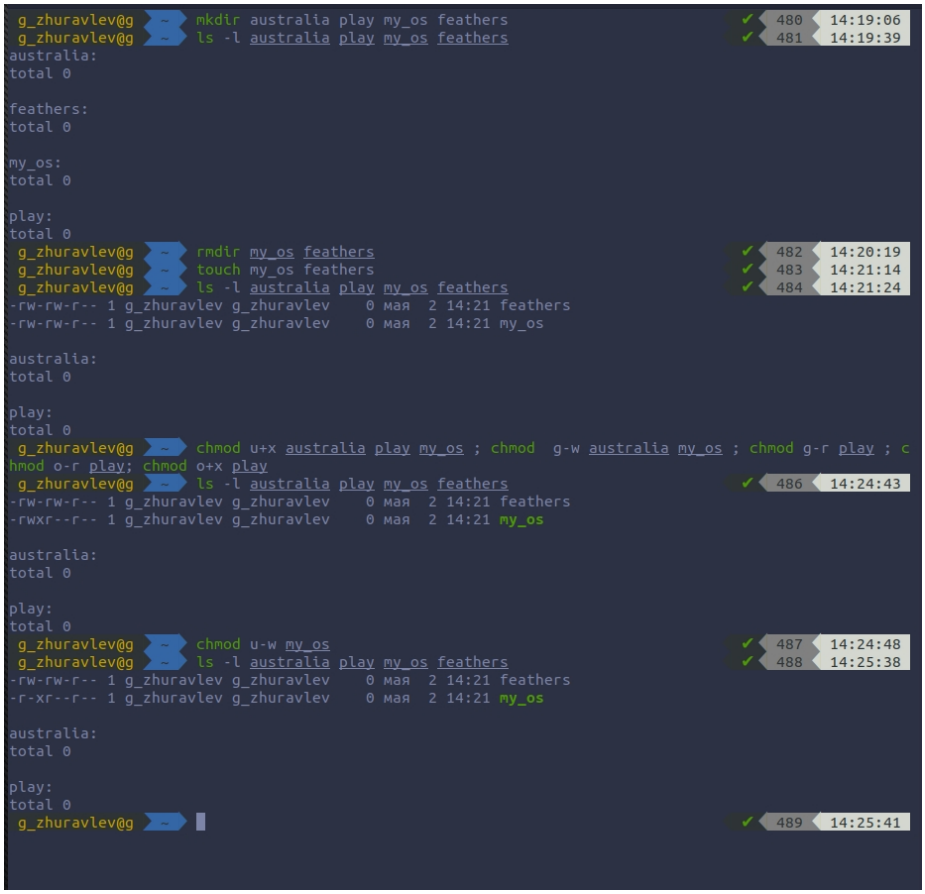


Рис. 14: Создание и изменение прав

4. Проделем приведённые упражнения.

4.1. Просмотрим содержимое файла /etc/password.(команда sudo cat)

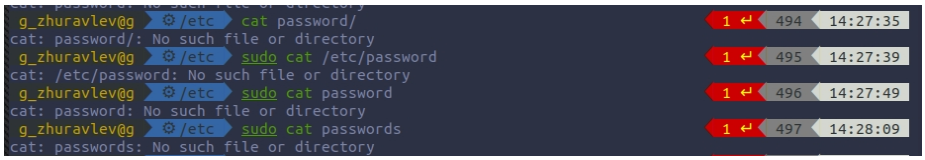


Рис. 15: Просмотр содержимого

4.2. Скопируем файл ~/feathers в файл ~/file.old.
4.3. Переместим файл ~/file.old в каталог ~/play.
4.4. Скопируем каталог ~/play в каталог ~/fun.
4.5. Переместим каталог ~/fun в каталог ~/play и назовём его games.

Рис. 16: Выполнение действий

4.6. Лишим владельца файла ~/feathers права на чтение.

Рис. 17: Лишение прав

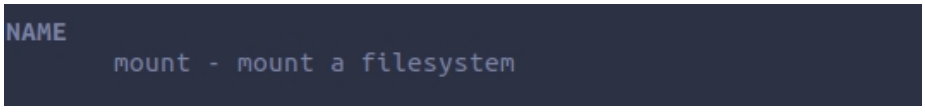
4.7. Просмотрим этот файл с помощью cat и 4.8. Попробуем скопировать

Ответ: нехватка прав.

4.9. Дадим владельцу файла ~/feathers право на чтение.
4.10. Лишим владельца каталога ~/play права на выполнение.
4.11. Перейдём в каталог ~/play -> нехватка прав.
4.12. Дадим владельцу каталога ~/play право на выполнение.

5. Прочитаем man по командам mount, fsck, mkfs, kill и кратко их охарактеризуем, приведя примеры.

1. mount



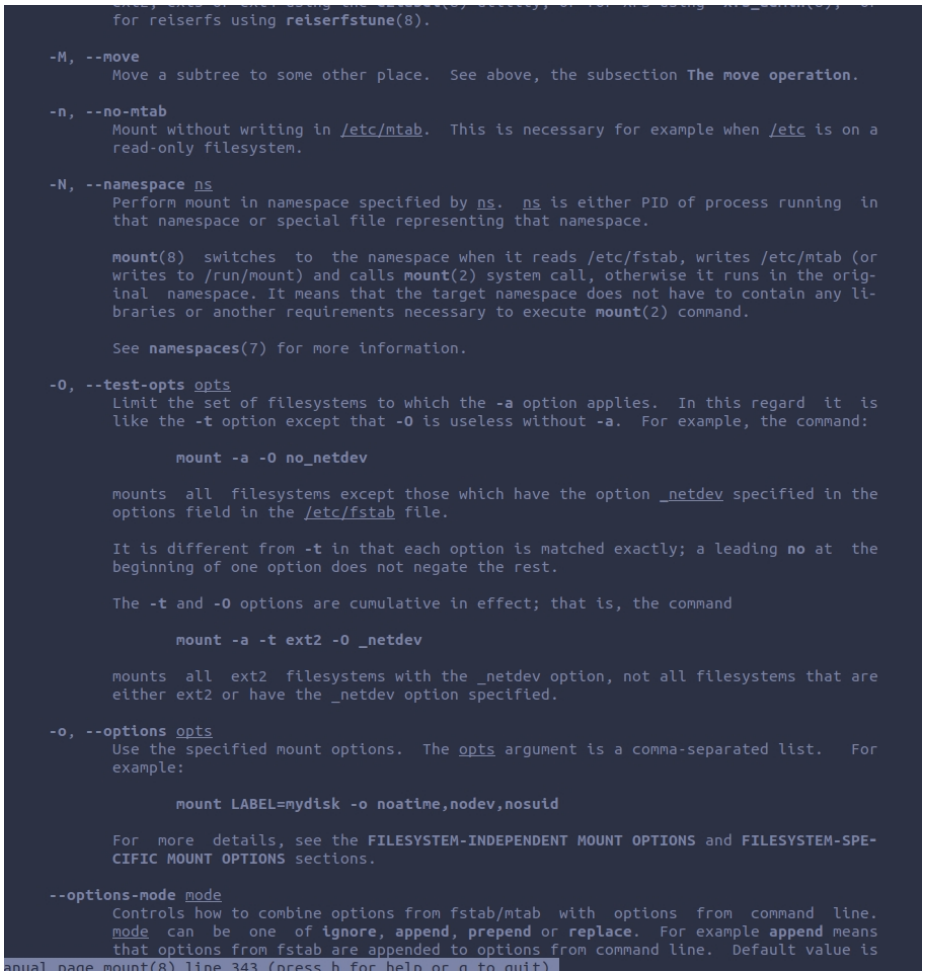
```
g_zhuravlev@g ~ cat feathers                                    ✓ 509 14:32:41
cat: feathers: Permission denied
g_zhuravlev@g ~ cp feathers copy                                1 ↵ 510 14:33:42
cp: cannot open 'feathers' for reading: Permission denied
g_zhuravlev@g ~ chmod u+r feathers                              1 ↵ 511 14:34:20
g_zhuravlev@g ~ chmod u-x play                                  ✓ 512 14:34:37
g_zhuravlev@g ~ cd play                                         ✓ 513 14:35:06
cd: permission denied: play
g_zhuravlev@g ~ chmod u+x play                                  1 ↵ 514 14:35:13
g_zhuravlev@g ~                                                 ✓ 515 14:35:32
```

Рис. 18: Просмотр файла и проба копирования

```
g_zhuravlev@g ~ cat feathers                                    ✓ 509 14:32:41
cat: feathers: Permission denied
g_zhuravlev@g ~ cp feathers copy                                1 ↵ 510 14:33:42
cp: cannot open 'feathers' for reading: Permission denied
g_zhuravlev@g ~ chmod u+r feathers                              1 ↵ 511 14:34:20
g_zhuravlev@g ~ chmod u-x play                                  ✓ 512 14:34:37
g_zhuravlev@g ~ cd play                                         ✓ 513 14:35:06
cd: permission denied: play
g_zhuravlev@g ~ chmod u+x play                                  1 ↵ 514 14:35:13
g_zhuravlev@g ~                                                 ✓ 515 14:35:32
```

Рис. 19: Выполнение действий

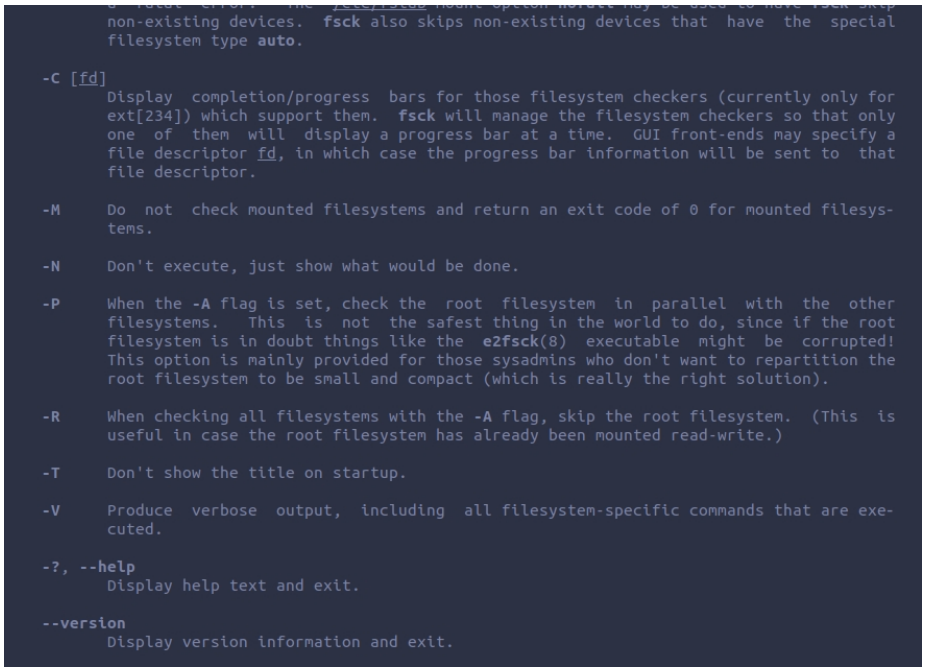
```
    -a, --all
           Mount  all  filesystems  (of  the  given types) mentioned in fstab (except for those
           whose line contains the noauto keyword).  The  filesystems  are  mounted  following
           their  order in fstab.  The mount command compares filesystem source, target (and fs
           root for bind mount or btrfs) to detect already mounted filesystems. The kernel  ta-
           ble with already mounted filesystems is cached during mount --all. It means that all
           duplicated fstab entries will be mounted.

           The option --all is possible to use for remount operation too. In this case all fil-
           ters (-t and -O) are applied to the table of already mounted filesystems.

           Note  that  it is a bad practice to use mount -a for fstab checking. The recommended
           solution is findmnt --verify.

    -B, --bind
           Remount a subtree somewhere else  (so  that  its  contents  are  available  in  both
           places).  See above, under Bind mounts.

    -c, --no-canonicalize
           Don't  canonicalize  paths.  The mount command canonicalizes all paths (from command
           line or fstab) by default.  This option can be used together with the  -f  flag  for
           already  canonicalized  absolute  paths.   The  option is designed for mount helpers
           which call mount -i.  It is strongly recommended to not use this command-line option
           for normal mount operations.

           Note that mount(8) does not pass this option to the /sbin/mount.type helpers.

    -F, --fork
           (Used in conjunction with -a.)  Fork off a new incarnation of mount for each device.
           This will do the mounts on different devices or different NFS servers  in  parallel.
           This  has the advantage that it is faster; also NFS timeouts go in parallel.  A dis-
           advantage is that the mounts are done in undefined order.  Thus, you cannot use this
           option if you want to mount both /usr and /usr/spool.

    -f, --fake
           Causes everything to be done except for the actual system call; if it's not obvious,
           this ``fakes'' mounting the filesystem.  This option is useful in  conjunction  with
           the  -v  flag  to  determine what the mount command is trying to do.  It can also be
           used to add entries for devices that were mounted earlier with the -n  option.   The
           -f  option  checks for an existing record in /etc/mtab and fails when the record al-
           ready exists (with a regular non-fake mount, this check is done by the kernel).

    -i, --internal-only
           Don't call the /sbin/mount.filesystem helper even if it exists.

    -L, --label label
           Mount the partition that has the specified label.

    -l, --show-labels
           Add the labels in the mount output.  mount must have permission to read the disk de-
```



Описание: Команда mount монтирует устройство и позволяет присоединить хранящиеся на нем файлы к
Структура: mount [device_name] [mount_point]

2. fsck

```
NAME
       fsck - check and repair a Linux filesystem

SYNOPSIS
       fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-op-
       tions]

DESCRIPTION
       fsck is used to check and optionally repair one or more Linux filesystems. filesys 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 LA-
       BEL=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 code 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 code returned when multiple filesystems are checked is the bit-wise OR of the exit
       codes for each filesystem that is checked.

       In actuality, fsck is simply a front-end for the various filesystem checkers (fsck.fstype)
       available under Linux. The filesystem-specific checker is searched for in the PATH envi-
       ronment variable. If the PATH is undefined then fallback to "/sbin".

       Please see the filesystem-specific checker manual pages for further details.

OPTIONS
       -l     Create an exclusive flock(2) lock file (/run/fsck/<diskname>.lock) for whole-disk
              device. This option can be used with one device only (this means that -A and -l are
              mutually exclusive). This option is recommended when more fsck(8) instances are ex-
              ecuted in the same time. The option is ignored when used for multiple devices or
              for non-rotating disks. fsck does not lock underlying devices when executed to
              check stacked devices (e.g. MD or DM) - this feature is not implemented yet.

       -r [fd]
              Report certain statistics for each fsck when it completes. These statistics include
              the exit status, the maximum run set size (in kilobytes), the elapsed all-clock time
              and the user and system CPU time used by the fsck run. For example:
```



Описание:Fsck команда взаимодействует с соответствующей файловой системой конкретных FSCK кома

1. Проверка на наличие ошибок и подсказывает пользователю интерактивное решение, как решить инди
2. Проверка на наличие ошибок и постарается автоматически исправить все ошибки;
3. Проверка на наличие ошибок без возможности восстановить их, но тогда выдаст ошибки на стандартн

Структура: fsck

3. mkfs

Описание: “make file system” (создать файловую систему). Команда обычно используется для управлени

Структура: mkfs -t [fs type] [target device]

4. kill

Описание: Когда вы выполняете команду "kill", то фактически вы посылаете системе сигнал, чтобы заст

Вы можете просмотреть все сигналы с помощью команды: $ kill -l

Структура: kill [SIGKILL] PID

Вывод

Благодаря этой лабораторной работе, я научился: пользоваться некоторыми командами; копировать каталоги и удалять их; перемещать файлы и каталоги; копировать файлы и каталоги; изменять права доступа к файлам и каталогам.

```
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 code 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 envi-
       ronment 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 filesys-
              tem 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  exe-
              cuted.   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@uwalt.nl.mugnet.org)
       Ron Sommeling (sommel@sci.kun.nl)
```

Рис. 20: мануал команды mkfs

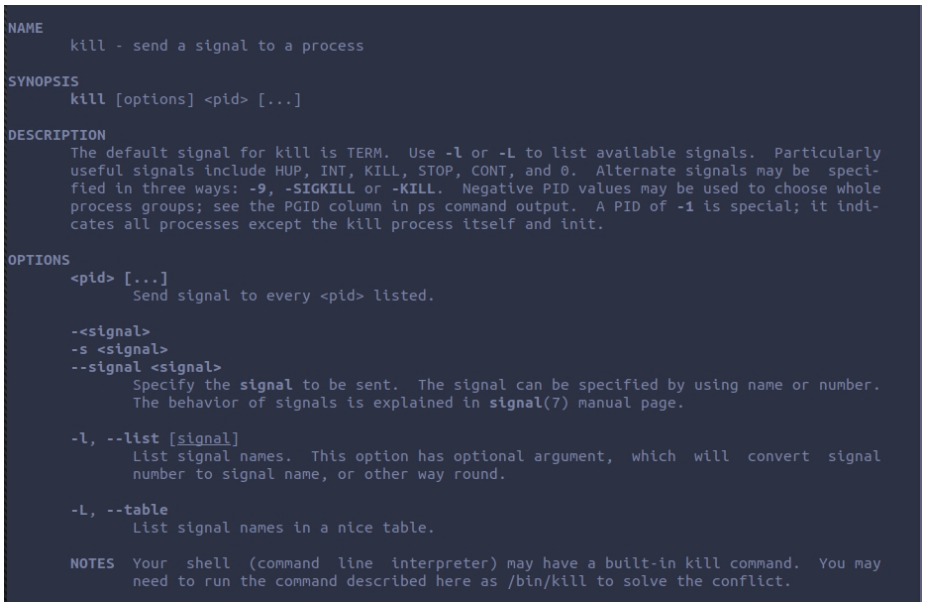


Рис. 21: мануал команды kill

Контрольные вопросы:

1 и 2. Файловая система в дисплейном классе содержит следующие каталоги первого уровня:
/bin - Основные программы, необходимые для работы в системе: командные оболочки shell, основные утил
/boot - Каталог, который содержит ядро системы— главную программу, загружающую и исполняющую в

/dev - Каталог, в котором содержатся псевдофайлы устройств. С точки зрения Linux все физические устро

/etc - В этом каталоге содержатся системные конфигурационные файлы — текстовые файлы, которые счит

/home - В структуре файловой системы Linux каждый пользователь имеет отдельный личный каталог для

/mnt - Каталоги для монтирования файловых систем сменных устройств и внешних файловых систем.

/proc - Файловая система на виртуальном устройстве, её файлы содержат информацию о текущем состоян

/root - Каталог администратора системы.

/sbin - Системные утилиты.

/usr - Программы и библиотеки, доступные пользователю.

/var - Рабочие файлы программ, различные временные данные: очереди (письма на отправку, файлы на пе

/tmp - Временные файлы.

3. Чтобы содержимое некоторой файловой системы было доступно операционной системе должно быть вы

4. Основные причины нарушения целостности файловой системы:

- Один блок адресуется несколькими mode (принадлежит нескольким файлам).

- Блок помечен как свободный, но в то же время занят (на него ссылается onode).

- Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).

- Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).

- Несовпадение между размером файла и суммарным размером адресуемых inode блоков.

- Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).

- "Потерянные" файлы (правильные inode, на которые не ссылаются записи каталогов).

- Недопустимые или неразмещенные номера inode в записях каталогов.

Чтобы устранить повреждения файловой системы используется команда fsck.

5. mkfs создаёт новую файловую систему.

6. Характеристика команд, которые позволяют просмотреть текстовые файлы:

- для просмотра небольших файлов - cat.

- для просмотра больших файлов - less — она позволяет осуществлять постраничный просмотр файлов.

- для просмотра начала файла - head[-n], по умолчанию она выводит первые 10 строк файла.

- команда tail[-n] выводит несколько (по умолчанию 10) последних строк файла.

7. Основные возможности команды cp:

- копирование файла в текущем каталоге.

- копирование нескольких файлов в каталог.

- копирование файлов в произвольном каталоге.

-i в команде cp выведет на экран запрос подтверждения о перезаписи файла, если на место целевого файла

Команда cp с опцией r (recursive) позволяет рекурсивно копировать каталоги вместе с входящими в них фа

8. Команды mv и mvdir предназначены для перемещения и переименования файлов и каталогов.
Формат команды: mv [-option] старый_файл новый_файл.
Для получения предупреждения перед переписыванием файла стоит использовать опцию i.

9. Права доступа определяют, кто и что может делать с содержимым файла. Существуют три группы прав
= установить право;

- лишить права;

+ дать право;

r чтение;

w запись;

x выполнение;

u (user) владелец файла;

g (group) группа, к которой принадлежит владелец файла;

o (others) все остальные.