Лабораторная работа №5

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 проверке использования диска и обслуживанию файловой системы.

Копирование файлов и каталогов. Команда cp

Копируем файл /usr/include/sys/io.h в домашний каталог

```
nastarkov@dk6n53 ~ $ cp /usr/include/sys/io.h ~
nastarkov@dk6n53 ~ $ ls
 abc1   blog                            equipment   io.h   monthly   public_html   tmp    Видео
 bin    course-directory-student-template   GNUstep     may    public    reports       work   Документы
```

Figure 1: Копирование файла в домашний каталог

Создаем в домашнем каталоге файл abc1 и копируем его в каталог ~/ski.plases, называем его equiplist2.

```
nastarkov@dk6n53 ~ $ touch abc1
nastarkov@dk6n53 ~ $ cp abc1 ski.plases
nastarkov@dk6n53 ~ $ mv ski.plases/abc1 ski.plases/equiplist2
```

Figure 2: Копирование файла в каталог и изменение названия

Копируем файл ~/feathers в файл ~/file.old

```
nastarkov@dk6n53 ~ $ touch my_os
nastarkov@dk6n53 ~ $ touch feathers
```

Figure 3: Копирование одного файла в другой

Перемещение файлов и каталогов. Команда mv

Перемещаем файл equipment в каталог ~/ski.plases

```
nastarkov@dk6n53 ~ $ mkdir ski.plases
nastarkov@dk6n53 ~ $ mv equipment ski.plases
```

Figure 5: Перемещение директории в каталог

Перемещаем файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment

```
nastarkov@dk6n53 ~ $ mv ski.plases/equiplist ski.plases/equipment
nastarkov@dk6n53 ~ $ mv ski.plases/equiplist2 ski.plases/equipment
```

Figure 6: Перемещение файлов в каталог

Перемещаем файл ~/file.old в каталог ~/play

```
nastarkov@dk6n53 ~ $ mv file.old play
```

Figure 7: Перемещение файла в каталог

Изменение имени файла и каталога. Команда mv

Переименовываем файл ~/ski.plases/equipment в ~/ski.plases/equiplist

```
nastarkov@dk6n53 ~ $ mv ski.plases/equipment ski.plases/equiplist
```

Figure 8: Переименовывание файла

Перемещаем каталог ~/fun в каталог ~/play и называем его games.

```
nastarkov@dk6n53 ~ $ mv play/fun play/games
```

Figure 9: Переименовывание файла

Изменение прав доступа. Команда chmod

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

```
nastarkov@dk6n53 ~ $ chmod 744 australia
nastarkov@dk6n53 ~ $ chmod 711 play
nastarkov@dk6n53 ~ $ chmod 544 my_os
nastarkov@dk6n53 ~ $ chmod 664 feathers
```

Figure 10: Присвоение определенных прав доступа

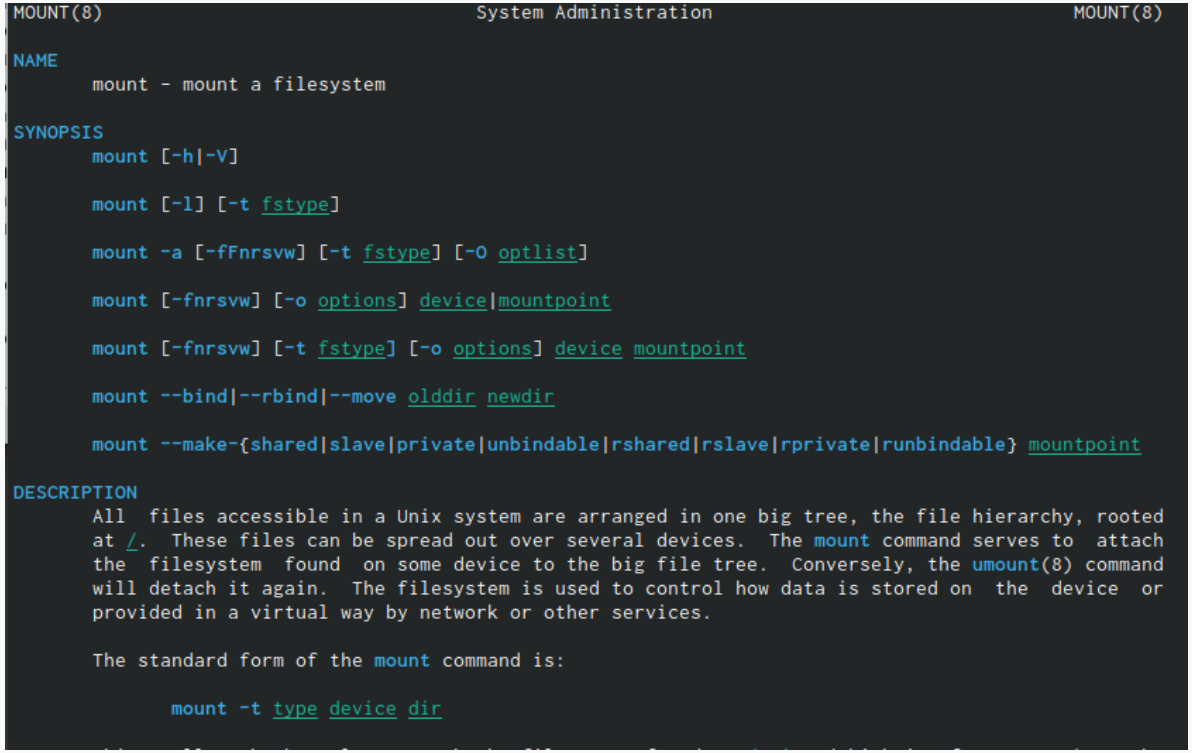
Лишаем владельца файла ~/feathers права на чтение. Просматриваем файл командой cat. Видим надпись “Отказано в доступе”. При копировании видим надпись “Отказано в доступе”

```
nastarkov@dk6n53 ~ $ mv play/fun play/games
nastarkov@dk6n53 ~ $ chmod u-r feathers
nastarkov@dk6n53 ~ $ cat feathers
cat: feathers: Отказано в доступе
nastarkov@dk6n53 ~ $ cp feathers ski.plases
```

Figure 11: Изменение прав доступа. Проверка

Команда man

По команде man читаем описание команд mount, mkfs, fsck, kill



```
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
```

Figure 12: Описание команды mount

Команда man

```
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.  filesys can be a
       device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an  filesys-
       tem  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 op-
       tions.

       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.
```

Figure 13: Описание команды fsck

Команда man

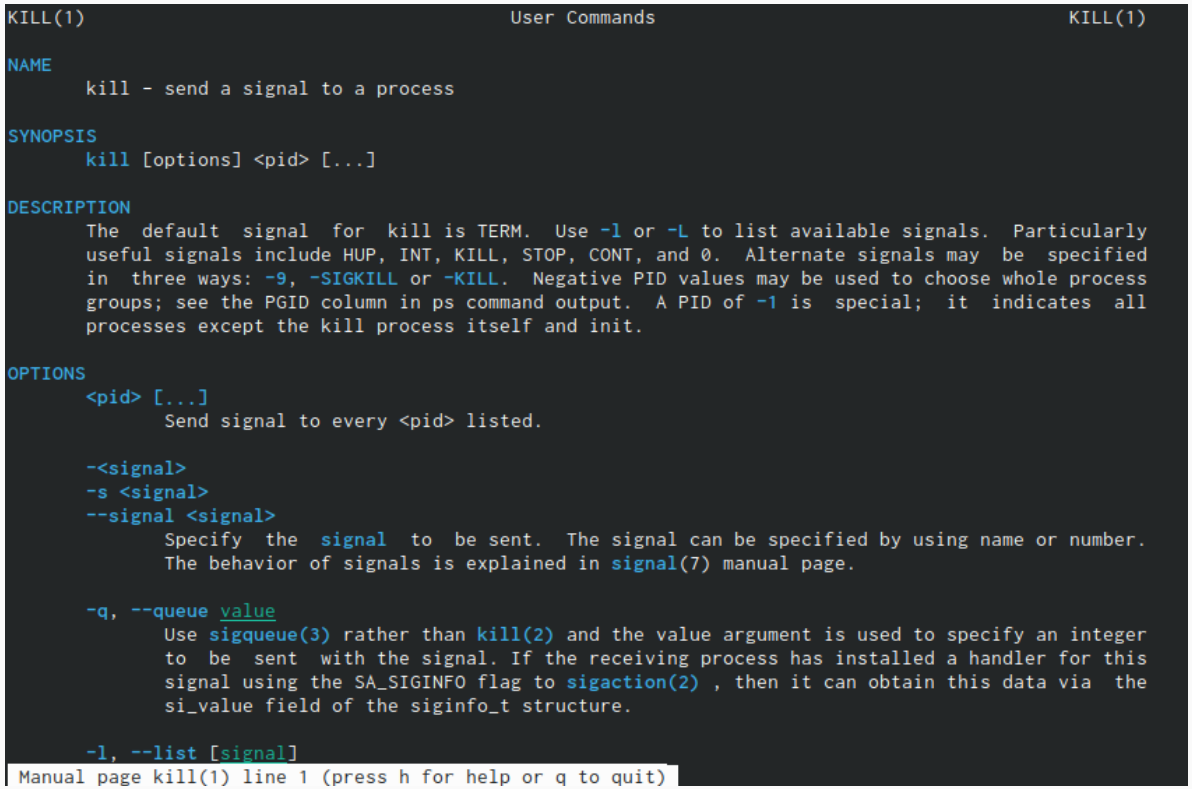
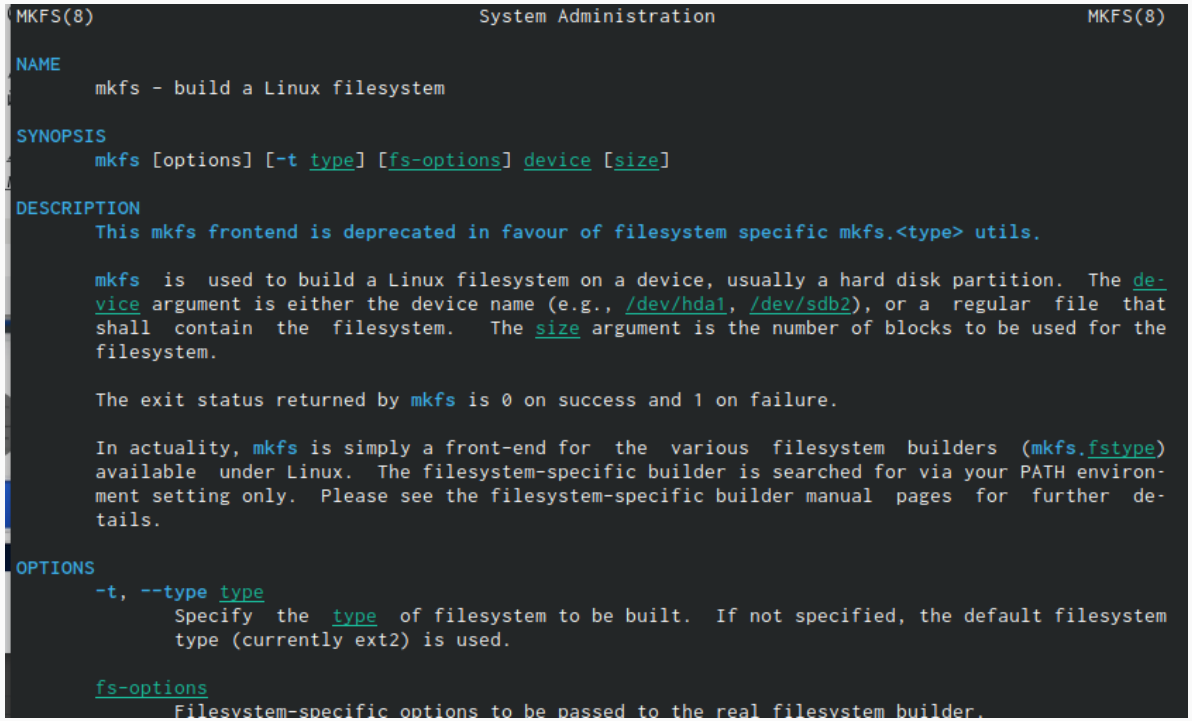


Figure 14: Описание команды kill

Команда man



```
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 de-
       vice 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 environ-
       ment setting only.  Please see the filesystem-specific builder manual  pages  for  further  de-
       tails.

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.
```

Figure 15: Описание команды mkfs