

# **Отчёт по лабораторной работе №5**

**Отчёт по лабораторной работе 5**

Репина Ангелина Олеговна

# Цель работы

Цель лабораторной работы - ознакомиться с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрести практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию системы.

# Выполнение лабораторной работы

1. Выполняю примеры, приведенные в первой части описания лабораторной работы 1.1 Выполняю примеры из пункта 5.2.2 (рис. Пример1, Пример2)

```
aorepina@dk3n52 ~ $ cd
aorepina@dk3n52 ~ $ touch abc1
aorepina@dk3n52 ~ $ cp abc1 april
aorepina@dk3n52 ~ $ cp abc1 may
aorepina@dk3n52 ~ $ mkdir monthly
aorepina@dk3n52 ~ $ cp april may monthly
aorepina@dk3n52 ~ $ cp monthly/may monthly/june
aorepina@dk3n52 ~ $ ls monthly
april  june  may
```

Figure 0.1: Примеры1

```
aorepina@dk3n52 ~ $ mkdir monthly.00
aorepina@dk3n52 ~ $ cp -r monthly monthly.00
aorepina@dk3n52 ~ $ cp -r monthly.00 /tmp
```

Figure 0.2: Пример2

- 1.2 Выполняю примеры из пункта 5.2.3 (рис. Пример3)

```

aorepina@dk3n52 ~ $ cd
aorepina@dk3n52 ~ $ mv april july
bash: mv april: команда не найдена
aorepina@dk3n52 ~ $ mv april july
aorepina@dk3n52 ~ $ mv july monthly.00
aorepina@dk3n52 ~ $ ls monthly.00
july  monthly
aorepina@dk3n52 ~ $ ls monthly
april  june  may
aorepina@dk3n52 ~ $ mv monthly.00 monthly.01
aorepina@dk3n52 ~ $ mkdir reports
aorepina@dk3n52 ~ $ mv monthly.01 reports
aorepina@dk3n52 ~ $ mv reports/monthly.01 reports/monthly

```

Figure 0.3: Пример3

### 1.3 Выполняю примеры из пункта 5.2.5 (рис. Пример4)

```

aorepina@dk3n52 ~ $ cd
aorepina@dk3n52 ~ $ touch may
aorepina@dk3n52 ~ $ ls -l may
-rw-r--r-- 1 aorepina studsci 0 anp 28 10:48 may
aorepina@dk3n52 ~ $ chmod u+x may
aorepina@dk3n52 ~ $ ls -l may
-rwxr--r-- 1 aorepina studsci 0 anp 28 10:48 may
aorepina@dk3n52 ~ $ chmod u-x may
aorepina@dk3n52 ~ $ ls -l may
-rw-r--r-- 1 aorepina studsci 0 anp 28 10:48 may
aorepina@dk3n52 ~ $ cd
aorepina@dk3n52 ~ $ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
aorepina@dk3n52 ~ $ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
aorepina@dk3n52 ~ $ chmod g-r monthly
aorepina@dk3n52 ~ $ chmod o-r monthly
aorepina@dk3n52 ~ $ cd
aorepina@dk3n52 ~ $ touch abc1
aorepina@dk3n52 ~ $ chmod g+w abc1

```

Figure 0.4: Пример4

## 2. Продолжаю выполнение лабораторной работы.

2.1 Копирую файл /usr/include/sys/io.h в домашний каталог и называю его equipment. (рис. 2.1)

```

aorepina@dk3n52 ~ $ cp /usr/include/sys/io.h ~
aorepina@dk3n52 ~ $ mv io.h equipment
aorepina@dk3n52 ~ $ ls
-
'2022-04-21 10-37-53.mkv'
abc1
course-directory-student-template
data.txt
equipment
GNUstep
keygen
keygen.pub
may
monthly
public
public_html
reports
rez.txt
tmp
trash
work
Видео
Документы
Загрузки
Изображения
Музыка
'нужно использовать для просмотра содержимое не только указанного каталога, но и подкаталогов'
Общедоступные
'Рабочий стол'
Шаблоны

```

Figure 0.5: 2.1

2.2 В домашнем каталоге создаю директорию ~/ski.plases. Перемещаю файл equipment в каталог ~/ski.plases. Переименовываю файл ~/ski.plases/equipment в ~/ski.plases/equiplist. (рис. 2.2-2.4)

```

aorepina@dk3n52 ~ $ mkdir ski.plases
aorepina@dk3n52 ~ $ mv equipment ski.plases
aorepina@dk3n52 ~ $ mv ~/ski.plases/equipment ~/ski.plases/equiplist
aorepina@dk3n52 ~ $ ls ski.plases
equiplist

```

Figure 0.6: 2.2-2.4

2.3 Создаю в домашнем каталоге файл abc1 и копирую его в каталог ~/ski.plases, называю его equiplist2. (рис. 2.5)

```

aorepina@dk3n52 ~ $ touch abc1
aorepina@dk3n52 ~ $ cd abc1 ski.plases
bash: cd: слишком много аргументов
aorepina@dk3n52 ~ $ cp abc1 ski.plases
aorepina@dk3n52 ~ $ mv ski.plases/abc1 ski.plases/equiplist2
aorepina@dk3n52 ~ $ ls ski.plases
equiplist  equiplist2

```

Figure 0.7: 2.5

2.4 Создаю каталог с именем equipment в каталоге ~/ski.plases. Перемещаю файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment. (рис. 2.6-7)

```
aorepina@dk3n52 ~ $ mkdir ski.plases/equipment
aorepina@dk3n52 ~ $ mv ski.plases/equiplist ski.plases/equipment
aorepina@dk3n52 ~ $ mv ski.plases/equiplist2 ski.plases/equipment
aorepina@dk3n52 ~ $ ls equipment
ls: невозможно получить доступ к 'equipment': Нет такого файла или каталога
aorepina@dk3n52 ~ $ ls ski.plases/equipment
equiplist  equiplist2
```

Figure 0.8: 2.6-7

2.5 Создаю и перемещаю каталог ~/newdir в каталог ~/ski.plases и называю его plans. (рис. 2.8)

```
aorepina@dk3n52 ~ $ mkdir newdir
aorepina@dk3n52 ~ $ mv newdir plans
aorepina@dk3n52 ~ $ mv plans ski.plases
aorepina@dk3n52 ~ $ ls
-
'2022-04-21 10-37-53.mkv'
abc1
course-directory-student-template
data.txt
GNUstep
keygen
keygen.pub
may
monthly
public
public_html
reports
rez.txt
ski.plases
tmp
trash
work
Видео
Документы
Загрузки
Изображения
Музыка
'нужно использовать для просмотра содержимое не только указанного каталога, но и подкаталогов'
Общедоступные
'Рабочий стол'
Шаблоны
aorepina@dk3n52 ~ $ ls ski.plases
equipment  plans
```

Figure 0.9: 2.8

3. Создаю необходимые файлы и каталоги. Определяю опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:

3.1. drwxr-r- ... australia 3.2. drwx-x-x ... play 3.3. -r-xr-r- ... my\_os 3.4. -rw-rw-r- ... feathers Вижу, что с помощью показанных на скриншоте

команд мы выполнили пункт. (рис.3)

```
aorepina@dk3n52 ~$ mkdir australia
aorepina@dk3n52 ~$ mkdir play
aorepina@dk3n52 ~$ touch my_os
aorepina@dk3n52 ~$ touch fearhers
aorepina@dk3n52 ~$ chmod 744 australia
aorepina@dk3n52 ~$ chmod 711 play
aorepina@dk3n52 ~$ chmod 544 my_os
aorepina@dk3n52 ~$ chmod 664 fearhers
chmod: невозможно получить доступ к 'fearhers': Нет такого файла или каталога
aorepina@dk3n52 ~$ chmod 664 fearhers
chmod: невозможно получить доступ к 'fearhers': Нет такого файла или каталога
aorepina@dk3n52 ~$ chmod 664 fearhers
aorepina@dk3n52 ~$ ls -l
итого 72
-rw-r--r-- 1 aorepina studsci 5886 апр 28 10:56 -
-rw-r--r-- 1 aorepina studsci 0 апр 21 10:37 '2022-04-21 10-37-53.mkv'
-rw-r--r-- 1 aorepina studsci 0 апр 28 11:01 abc1
drwxr-xr-x 2 aorepina studsci 2048 апр 28 11:09 australia
drwxr-xr-x 5 aorepina studsci 2048 апр 22 13:35 course-directory-student-template
-rw-r--r-- 1 aorepina studsci 38 дек 1 14:23 data.txt
-rw-r--r-- 1 aorepina studsci 0 апр 28 11:10 fearhers
drwxr-xr-x 3 aorepina studsci 2048 сен 2 2021 gdata
-rw-r--r-- 1 aorepina studsci 411 апр 21 15:44 keygen
-rw-r--r-- 1 aorepina studsci 97 апр 21 15:44 keygen.pub
-rw-r--r-- 1 aorepina studsci 0 апр 28 10:40 may
drwxr-xr-x 2 aorepina studsci 2048 апр 28 10:41 monthly
-r-xr--r-- 1 aorepina studsci 0 апр 28 11:10 my_os
drwxr-xr-x 2 aorepina studsci 2048 апр 28 11:10 play
drwxr-xr-x 3 aorepina studsci 2048 сен 2 2021 public
drwxr-xr-x 1 aorepina root 18 апр 14 19:42 public_html -> public/public_html
drwxr-xr-x 3 aorepina studsci 2048 апр 28 10:45 reports
-rw-r--r-- 1 aorepina studsci 40 дек 1 14:23 res.txt
drwxr-xr-x 4 aorepina studsci 2048 апр 28 11:00 ski_places
drwxr-xr-x 2 aorepina studsci 2048 окт 20 2021 tmp
drwxr-xr-x 4 aorepina studsci 2048 сен 2 2021 trash
drwxr-xr-x 3 aorepina studsci 2048 апр 21 18:24 work
drwxr-xr-x 2 aorepina studsci 2048 сен 2 2021 видео
drwxr-xr-x 9 aorepina studsci 2048 апр 22 16:50 Документы
drwxr-xr-x 2 aorepina studsci 4096 апр 27 19:24 Загрузки
drwxr-xr-x 2 aorepina studsci 14336 апр 21 17:08 Изображения
drwxr-xr-x 3 aorepina studsci 2048 сен 5 2021 Музыка
-rw-r--r-- 1 aorepina studsci 10800 апр 22 17:32 'нужно использовать для просмотра содержимое не только указанного каталога, но и подкаталогов.'
drwxr-xr-x 2 aorepina studsci 2048 сен 2 2021 Общедоступные
drwxr-xr-x 2 aorepina studsci 2048 мар 2 18:45 'Рабочий стол'
drwxr-xr-x 2 aorepina studsci 2048 сен 2 2021 Шаблоны
```

Figure 0.10: 3

4. Прodelываю приведенные ниже упражнения.

4.1 . Просматриваю содержимое файла /etc/passwd. (рис. 4.1)

```
aorepina@dk3n52 ~$ cat /etc/passwd
root:x:0:0:System user; root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:Mail program user:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
man:x:13:15:System user; man:/dev/null:/sbin/nologin
postmaster:x:14:12:Postmaster user:/var/spool/mail:/sbin/nologin
cron:x:16:16:A user for sys-process/cronbase:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/home/ftp:/bin/false
sshd:x:22:22:User for ssh:/var/empty:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/bin/false
squid:x:31:31:Squid:/var/cache/squid:/bin/false
gdm:x:32:32:User for running GDM:/var/lib/gdm:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/bin/false
games:x:35:35:games:/usr/games:/bin/bash
named:x:40:40:bind:/var/bind:/bin/false
mysql:x:60:60:MySQL program user:/dev/null:/sbin/nologin
postgres:x:70:70:PostgreSQL program user:/var/lib/postgresql:/bin/sh
nut:x:84:84:nut:/var/state/nut:/bin/false
cyrus:x:85:12:/usr/cyrus:/bin/false
vpormail:x:89:89:/var/vpormail:/bin/false
alias:x:200:200:/var/qmail/alias:/bin/false
qmaild:x:201:200:/var/qmail:/bin/false
qmail1:x:202:200:/var/qmail:/bin/false
qmailp:x:203:200:/var/qmail:/bin/false
qmailq:x:204:201:/var/qmail:/bin/false
qmailr:x:205:201:/var/qmail:/bin/false
qmails:x:206:201:/var/qmail:/bin/false
postfix:x:207:207:postfix:/var/spool/postfix:/bin/false
```

Figure 0.11: 4.1

4.2 Копирую файл ~/feathers в файл ~/file.old. Перемещаю файл ~/file.old в каталог ~/play. Копирую каталог ~/play в каталог ~/fun. Перемещаю каталог ~/fun в каталог ~/play и называю его games. (рис. 4.2-5)

```
aorepina@dk3n52 ~ $ cp feathers file.old
aorepina@dk3n52 ~ $ mv file.old play
aorepina@dk3n52 ~ $ cp -r play fun
aorepina@dk3n52 ~ $ mv fun play
aorepina@dk3n52 ~ $ mv play/fun games
aorepina@dk3n52 ~ $ ls play
file.old
aorepina@dk3n52 ~ $ ls file.old
ls: невозможно получить доступ к 'file.old': Нет такого файла или каталога
aorepina@dk3n52 ~ $ mv fun play
mv: не удалось выполнить stat для 'fun': Нет такого файла или каталога
aorepina@dk3n52 ~ $ cp -r play fun
```

Figure 0.12: 4.2-5

4.3 Лишаю владельца файла ~/feathers права на чтение. Вижу, что при попытке просмотреть файл командой cat нам отказано в доступе. (рис. 4.6-7)

```
aorepina@dk3n52 ~ $ mv fun play
aorepina@dk3n52 ~ $ mv play/fun play/games
aorepina@dk3n52 ~ $ chmod u-r feathers
aorepina@dk3n52 ~ $ cat feathers
cat: feathers: Отказано в доступе
```

Figure 0.13: 4.6-7

4.4 При попытке скопировать файл feathers нам отказано в доступе. Даю владельцу файла право на чтение. (рис. 4.8-9)

```
aorepina@dk3n52 ~ $ cp feathers monthly
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
aorepina@dk3n52 ~ $ chmod u+r feathers
```

Figure 0.14: 4.8-9

4.5 Лишила владельца каталога ~/play права на выполнение. Попыталась перейти в каталог, у меня получилось. Предоставила владельцу права на выполнение. (рис. 4.10-12)



```
aorepina@dk3n52 ~ $ chmod u-x play
aorepina@dk3n52 ~ $ cd play
aorepina@dk3n52 ~/play $ cd ~
aorepina@dk3n52 ~ $ chmod u+x play
```

Figure 0.15: 4.10-12

5. Прочитала man по командам mount, fsck, mkfs, kill. (рис. 5)

```
aorepina@dk3n52 ~ $ man mount
aorepina@dk3n52 ~ $ man fsck
aorepina@dk3n52 ~ $ man mkfs
aorepina@dk3n52 ~ $ man kill
aorepina@dk3n52 ~ $
```

Figure 0.16: 5

5.1 Команда mount предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки и листья. Корень этого дерева обозначается как слеш. Физически файлы могут располагаться на различных устройствах. Команда mount служит для подключения файловых систем разных устройств к этому большому дереву. Наиболее часто встречающаяся форма команды mount выглядит следующим образом: “mount -t vfstype device dir”. Такая команда предлагает ядру смонтировать файловую систему указанного типа, расположенную на устройстве к заданному каталогу, который часто называют точкой монтирования (рис. mount)

```
~: man — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
[иконка] Новая вкладка  [иконка] Разделить окно по вертикали  [иконка] Разделить окно по горизонтали  >> >>

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|rlslave|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.

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

Figure 0.17: mount

5.2 Команда fsck: это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах. Он используется программы, специфичные для типа файловой системы, которую он проверяет. У этой команды следующий синтаксис `fsck [параметр]-[параметр ФС][...]`. Например, если нужно восстановить файловую систему на некотором устройстве `/dev/sdb2`, следует воспользоваться `sudo fsck -y /dev/sdb2`. Опция `-y` необходима, при её отсутствии придется часто давать подтверждение (рис. fsck)

```
~: man — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
[иконка] Новая вкладка  [иконка] Разделить окно по вертикали  [иконка] Разделить окно по горизонтали  >>

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

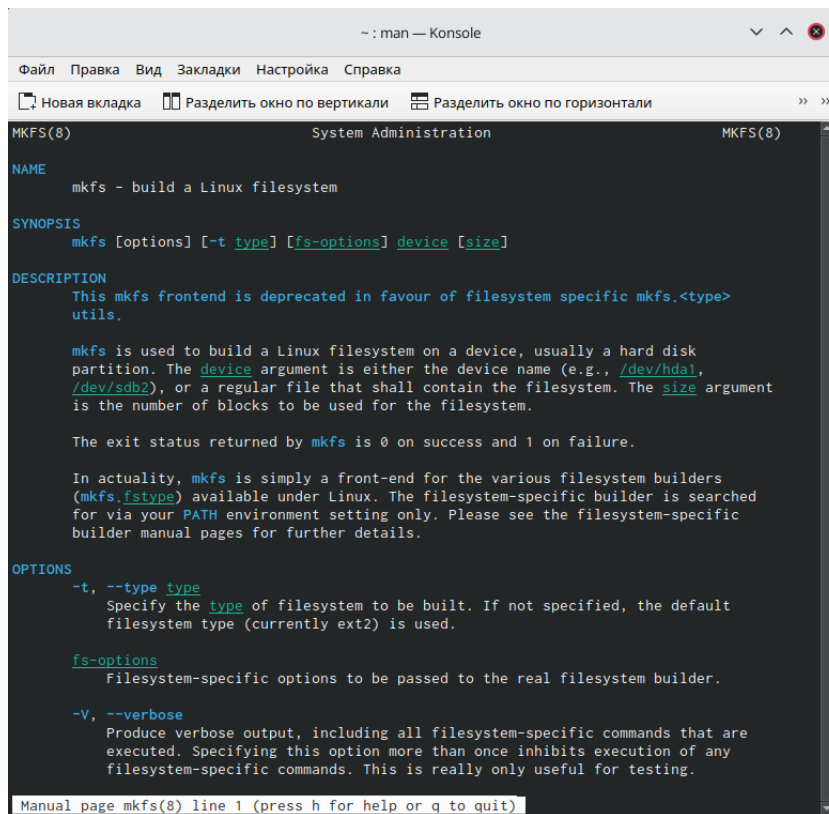
    4      Filesystem errors left uncorrected

    8      Operational error

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

Figure 0.18: fsck

5.3 Команда mkfs создает новую файловую систему Linux. Синтаксис mkfs [-V] [-tfstype] [fs-options] filesys [blocks] mkfs. Например, команда mkfs -t ext2/dev/hdb1 создает файловую систему ext2 в разделе /dev/hdb1



```
~ : man — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
Новая вкладка  Разделить окно по вертикали  Разделить окно по горизонтали  >> >>

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.

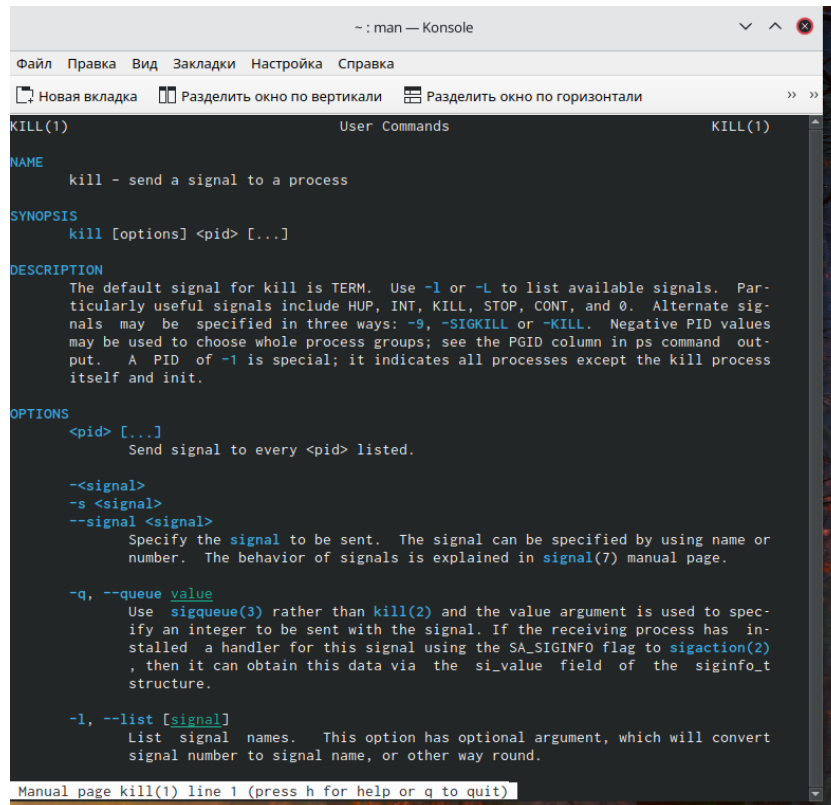
    fs-options
        Filesystem-specific options to be passed to the real filesystem builder.

    -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.

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

Figure 0.19: mkfs

5.4 Команда kill посылает сигнал процессору или выводит список допустимых сигналов. Имеет синтаксис kill [опции] PID, PID - числовой идентификатор процесса или несколько процессов, если требуется посылать сигнал сразу нескольким процессам. Например, kill -KILL 3121 посылает сигнал KILL процессору с PID 3121, чтобы принудительно завершить процесс (рис. kill)



```
~: man — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
Новая вкладка  Разделить окно по вертикали  Разделить окно по горизонтали  » »
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Partic-
    ularly useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate sig-
    nals may be specified in three ways: -9, -SIGKILL or -KILL. Negative PID values
    may be used to choose whole process groups; see the PGID column in ps command out-
    put. A PID of -1 is special; it indicates all processes except the kill process
    itself and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or
        number. The behavior of signals is explained in signal(7) manual page.

    -q, --queue value
        Use sigqueue(3) rather than kill(2) and the value argument is used to spec-
        ify an integer to be sent with the signal. If the receiving process has in-
        stalled a handler for this signal using the SA_SIGINFO flag to sigaction(2)
        , then it can obtain this data via the si_value field of the siginfo_t
        structure.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert
        signal number to signal name, or other way round.

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

Figure 0.20: kill

## Выводы

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

### #Итоговые вопросы

Ответы на контрольные вопросы: 1).Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df-Th». Из рисунка видно, что на моем компьютере есть следующие файловые системы: dev tmpfs,tmpfs,ext4,iso9660. dev tmpfs позволяет ядру создать экземпляр tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в devtmpfs.devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств.tmpfs-временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов.Предполагает наличие достаточного объёма виртуальной памяти.Файловая система tmpfs предназначена для того, чтобы использовать часть физической памяти сервера как обычный

дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. ext4 – имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extentfile system», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных. Характеристики: максимальный размер файла: 16 TB; максимальный размер раздела: 16 TB; максимальный размер имени файла: 255 символов. Рекомендации по использованию: наилучший выбор для SSD; наилучшая производительность по сравнению с предыдущими Ext-системами; она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3. ISO 9660 – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

2). Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел – / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам. - “/” – корень. Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов

начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге. - “/BIN” – бинарные файлы пользователя. Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления. - “/SBIN” – системные исполняемые файлы. Так же как и “/bin”, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не примонтирован каталог /usr. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя. - “/ETC” – конфигурационные файлы В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации Init Scripts, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ. - “/DEV” – файлы устройств В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры – это просто файлы в каталоге /dev/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. - “/PROC” – информация о процессах По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов. - “/VAR” – переменные файлы. Название каталога “/var” говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кешы, базы данных и так далее. - “/TMP” – временные файлы В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию. - “/USR” – программы пользователя Это самый большой



каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию. - “/HOME” – домашняя папка. В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т.д. - “/BOOT” – файлы загрузчика. Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящиеся в каталоге /boot/grub. - “/LIB” – системные библиотеки. Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах /bin и /sbin. - “/OPT” – дополнительные программы. В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями. - “/MNT” – монтирование. В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы. - “/MEDIA” – съемные носители. В этот каталог система монтирует все подключаемые внешние накопители – USB флешки, оптические диски и другие носители информации. - “/SRV” – сервер. В этом каталоге содержатся файлы серверов и сервисов. - “/RUN” – процессы. Каталог, содержащий PID файлы процессов, похожий на “/var/run”, но в отличие от него, он размещен в TMPFS, а поэтому после перезагрузки все файлы теряются. 3). Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount. 4). Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду fsck. 5). Файловую систему можно создать, используя команду mkfs. Ее краткое описание дано в пункте 5) в ходе выполнения заданий лабораторной работы. 6). Для просмотра текстовых файлов существуют следующие команды: - cat. Задача команды cat очень проста – она читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис

утилиты: cat [опции] файл1 файл2 ... Основные опции: -b – нумеровать только непустые строки -E – показывать символ \$ в конце каждой строки -n – нумеровать все строки -s – удалять пустые повторяющиеся строки -T – отображать табуляции в виде ^I -h – отобразить справку -v – версия утилиты - nl. Команда nl действует аналогично команде cat, но выводит еще номера строк в столбце слева. - less. Существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста. Синтаксис аналогичный синтаксису команды cat. Некоторые опции: -g – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения) -N – показывать номера строк - head. Команда head выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита. Синтаксис аналогичный синтаксису команды cat. Основные опции: -c (–bytes) – позволяет задавать количество текста не в строках, а в байтах -n (–lines) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию -q (–quiet, –silent) – выводит только текст, не добавляя к нему название файла -v (–verbose) – перед текстом выводит название файла -z (–zero-terminated) – символы перехода на новую строку заменяет символами завершения строк - tail Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды cat. Основные опции: -c – выводить указанное количество байт с конца файла -f – обновлять информацию по мере появления новых строк в файле -n – выводить указанное количество строк из конца файла -pid – используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс -q – не выводить имена файлов -retry – повторять попытки открыть файл, если он недоступен -v – выводить подробную информацию о файле. 7). Утилита cp позволяет полностью копировать файлы и директории. Синтаксис: cp

[опции] файл-источник файл-приемник После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем. Основные опции: -attributes-only -не копировать содержимое файла, а только флаги доступа и владельца -f, -force -перезаписывать существующие файлы -i, -interactive -спрашивать, нужно ли перезаписывать существующие файлы -L -копировать не символические ссылки, а то, на что они указывают -n -не перезаписывать существующие файлы -P -не следовать символическим ссылкам -r -копировать папку Linux рекурсивно -s -не выполнять копирование файлов в Linux, а создавать символические ссылки -u -скопировать файл, только если он был изменён -x -не выходить за пределы этой файловой системы -p -сохранять владельца, временные метки и флаги доступа при копировании -t -считать файл-приемник директорией и копировать файл-источник в эту директорию. 8). Команда mv используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий. Синтаксис: mv [-опции] старый\_файл новый\_файл Основные опции: -help -выводит на экран официальную документацию об утилите -version -отображает версию mv -b -создает копию файлов, которые были перемещены или перезаписаны -f -при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла -i -наоборот, будет спрашивать разрешение у владельца -n -отключает перезапись уже существующих объектов -strip-trailing-slashes -удаляет завершающий символ / у файла при его наличии -t [директория] -перемещает все файлы в указанную директорию -u -осуществляет перемещение только в том случае, если исходный файл новее объекта назначения -v -отображает сведения о каждом элементе во время обработки команды Команда rename также предназначена, чтобы переименовать файл. Синтаксис: rename [опции] старое\_имя новое\_имя файлы. Основные опции: -v -вывести список обработанных файлов -n -тестовый режим,

на самом деле никакие действия выполнены не будут -f –принудительно перезаписывать существующие файлы. 9). Права доступа– совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Синтаксис команды: `chmod режим имя_файла` Режим имеет следующие компоненты структуры и способ записи: `=` установить право-лишить права `+` дать право `чтение` `w` записи `x` выполнение `u` (user) владелец файла `g` (group) группа, к которой принадлежит владелец файла `o` (others) все остальные. # Список литературы {unnumbered}