

Отчет по лабораторной работе №5

Операционные системы

Морозова Ульяна Константиновна

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.

Выполнение лабораторной работы

1. Выполним все примеры, приведенные в первой части описания лабораторной работы (рис.1-15)

```
ukmorofova@dk4n65 ~ $ cd
ukmorofova@dk4n65 ~ $ touch abc1
ukmorofova@dk4n65 ~ $ cp abc1 april
ukmorofova@dk4n65 ~ $ cp abc1 may
```

{ #fig:001 width=70% }

```
ukmorofova@dk4n65 ~ $ mkdir monthly
ukmorofova@dk4n65 ~ $ cp april may monthly
ukmorofova@dk4n65 ~ $ ls
abc1          Hugo         public       Документы    'Рабочий стол'
april         lab01        public_html  Загрузки     Шаблоны
Architecture_PC may          tmp          Изображения
C_programs    monthly      work         Музыка
GNUstep       newdir       Видео        Общедоступные
ukmorofova@dk4n65 ~ $ cd monthly
ukmorofova@dk4n65 ~/monthly $ ls
april may
```

Рис. 0.1.: Пример 2

```
ukmorofova@dk4n65 ~ $ cp monthly/may monthly/june
ukmorofova@dk4n65 ~ $ cd monthly
ukmorofova@dk4n65 ~/monthly $ ls
april  june  may
```

Рис. 0.2.: Пример 3

```
ukmorofova@dk4n65 ~ $ cp -r monthly.00 tmp
ukmorofova@dk4n65 ~ $ cd tmp
ukmorofova@dk4n65 ~/tmp $ ls
monthly.00
```

Рис. 0.3.: Пример 4

```
ukmorofova@dk4n65 ~ $ mkdir mohthly.00
ukmorofova@dk4n65 ~ $ cp -r monthly monthly.00
ukmorofova@dk4n65 ~ $ ls
abc1      Hugo      monthly.00  work      Музыка
april     lab01     newdir      Видео     Общедоступные
Architecture_PC  may      public      Документы 'Рабочий стол'
C_programs mohthly.00 public_html Загрузки  Шаблоны
GNUstep   monthly   tmp         Изображения
ukmorofova@dk4n65 ~ $ cd monthly.00
ukmorofova@dk4n65 ~/monthly.00 $ ls
april  june  may
```

Рис. 0.4.: Пример 5

```
ukmorofova@dk4n65 ~/tmp $ cd
ukmorofova@dk4n65 ~ $ mv april july
ukmorofova@dk4n65 ~ $ ls
abc1      july      newdir      Видео      Общедоступные
Architecture_PC  lab01     public      Документы 'Рабочий стол'
C_programs    may      public_html Загрузки  Шаблоны
GNUstep       monthly   tmp         Изображения
Hugo          monthly.00 work      Музыка
```

{ #fig:001 width=70% }

```
ukmorofova@dk4n65 ~ $ mv july monthly.00
ukmorofova@dk4n65 ~ $ ls monthly.00
april  july  june  may
```

{ #fig:001 width=70% }

```
ukmorofova@dk4n65 ~ $ mv monthly.00 monthly.01
ukmorofova@dk4n65 ~ $ ls
abc1          lab01          public          Документы      'Рабочий стол'
Architecture_PC may             public_html     Загрузки       Шаблоны
C_programs    monthly        tmp            Изображения
GNUstep       monthly.01     work           Музыка
Hugo          newdir         Видео          Общедоступные
```

Рис. 0.5.: Пример 8

```
ukmorofova@dk4n65 ~ $ mkdir reports
ukmorofova@dk4n65 ~ $ mv monthly.01 reports
ukmorofova@dk4n65 ~ $ ls
abc1          lab01          public_html     Документы      'Рабочий стол'
Architecture_PC may             reports         Загрузки       Шаблоны
C_programs    monthly        tmp            Изображения
GNUstep       newdir         work           Музыка
Hugo          public         Видео          Общедоступные
ukmorofova@dk4n65 ~ $ ls reports
monthly.01
```

Рис. 0.6.: Пример 9

```
ukmorofova@dk4n65 ~ $ mv reports/monthly.01 reports/monthly
ukmorofova@dk4n65 ~ $ ls reports
monthly
```

Рис. 0.7.: Пример 10

```
ukmorofova@dk4n65 ~ $ touch may
ukmorofova@dk4n65 ~ $ ls -l may
-rw-r--r-- 1 ukmorofova studsci 0 anp 27 19:29 may
ukmorofova@dk4n65 ~ $ chmod u+x may
ukmorofova@dk4n65 ~ $ ls -l may
-rwxr--r-- 1 ukmorofova studsci 0 anp 27 19:29 may
```

Рис. 0.8.: Пример 11

```
ukmorofova@dk4n65 ~ $ chmod u-x may
ukmorofova@dk4n65 ~ $ ls -l may
-rw-r--r-- 1 ukmorofova studsci 0 anp 27 19:29 may
```

Рис. 0.9.: Пример 12

```
ukmorofova@dk4n65 ~ $ chmod g-r monthly
ukmorofova@dk4n65 ~ $ chmod o-r monthly
ukmorofova@dk4n65 ~ $ ls -l monthly
```

Рис. 0.10.: Пример 13

```
drwx--x--x 2 ukmorofova studsci 2048 apr 27 19:20 monthly
```

Рис. 0.11.: Пример 14

```
ukmorofova@dk4n65 ~ $ touch abc1
ukmorofova@dk4n65 ~ $ chmod g+w abc1
ukmorofova@dk4n65 ~ $ ls -l abc1
-rw-rw-r-- 1 ukmorofova studsci 0 apr 27 19:34 abc1
```

Рис. 0.12.: Пример 15

2. Выполним следующие действия.
3. Скопируем файл /usr/include/sys/io.h в домашний каталог и назовите его equipment (рис.16).

```
ukmorofova@dk4n65 ~ $ cp /usr/include/sys/io.h equipment.h
ukmorofova@dk4n65 ~ $ ls
abc1          Hugo          public        Видео          Общедоступные
Architecture_PC lab01         public_html   Документы     'Рабочий стол'
C_programs    may           reports       Загрузки      Шаблоны
equipment.h   monthly      tmp           Изображения
GNUstep       newdir       work          Музыка
```

Рис. 0.13.: Копирование файла

2. В домашнем каталоге создам директорию ~/ski.places (рис.17). Переместим файл equipment в каталог ~/ski.places

```
ukmorofova@dk4n65 ~ $ mkdir ski.places
ukmorofova@dk4n65 ~ $ mv equipment.h ski.places
ukmorofova@dk4n65 ~ $ ls ski.places
equipment.h
```

Рис. 0.14.: Создание директории

3. Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist (рис.19).

```
ukmorofova@dk4n65 ~ $ mv sci.plases/equipment.h sci.plases/equiplist
ukmorofova@dk4n65 ~ $ ls sci.plases
equiplist
```

Рис. 0.15.: Переименование файла

4. Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.plases, назовем его equiplist2. Создадим каталог с именем equipment в каталоге ~/ski.plases (рис.19).

```
ukmorofova@dk4n65 ~ $ ls
abc1      lab01      public_html Видео      Общедоступные
Architecture_PC may        reports    Документы 'Рабочий стол'
C_programs monthly    sci.plases Загрузки  Шаблоны
GNUstep   newdir     tmp        Изображения
Hugo      public     work       Музыка

ukmorofova@dk4n65 ~ $ cp abc1 sci.plases/equiplist2
ukmorofova@dk4n65 ~ $ ls sci.plases
equiplist equiplist2
ukmorofova@dk4n65 ~ $ mkdir sci.plases/equipment
ukmorofova@dk4n65 ~ $ ls sci.plases
equiplist equiplist2 equipment
```

Рис. 0.16.: Создание файла и каталога

5. Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment (рис.20).

```
ukmorofova@dk4n65 ~ $ mv sci.plases/equiplist sci.plases/equiplist2 sci.plases/equipment
ukmorofova@dk4n65 ~ $ ls sci.plases
equipment
ukmorofova@dk4n65 ~ $ ls
abc1      lab01      public_html Видео      Общедоступные
Architecture_PC may        reports    Документы 'Рабочий стол'
C_programs monthly    sci.plases Загрузки  Шаблоны
GNUstep   newdir     tmp        Изображения
Hugo      public     work       Музыка
```

Рис. 0.17.: Перемещение файлов

6. Создадим и переместим каталог ~/newdir в каталог ~/ski.plases и назовем его plans (рис.21).

```

ukmorofova@dk4n65 ~ $ mv newdir sci.plases
ukmorofova@dk4n65 ~ $ mv sci.plases/newdir plans
ukmorofova@dk4n65 ~ $ ls sci.plases
equipment
ukmorofova@dk4n65 ~ $ ls
abc1          lab01      public_html Видео      Общедоступные
Architecture_PC may        reports    Документы 'Рабочий стол'
C_programs    monthly   sci.plases Загрузки  Шаблоны
GNUstep       plans     tmp        Изображения
Hugo          public    work       Музыка
ukmorofova@dk4n65 ~ $ mv plans sci.plases
ukmorofova@dk4n65 ~ $ ls sci.plases
equipment plans

```

Рис. 0.18.: Перемещение каталога

3. Определим опции команды `chmod` для того, чтобы присвоить файлам и каталогам определенные права доступа:

4. Для каталога `australia` `drwxr-r-` (рис.22-25)

```

ukmorofova@dk4n58 ~ $ mkdir australia
ukmorofova@dk4n58 ~ $ ls -l australia

```

```

drwxr-xr-x 2 ukmorofova studsci 2048 anp 28 10:46 australia

```

```

ukmorofova@dk4n58 ~ $ chmod go-x australia
ukmorofova@dk4n58 ~ $ ls -l australia

```

```

drwxr--r-- 2 ukmorofova studsci 2048 anp 28 10:46 australia

```

2. Для каталога `play` `drwx-x-x` (рис. 26-29)

```

ukmorofova@dk4n58 ~ $ mkdir play
ukmorofova@dk4n58 ~ $ ls -l

```

```

drwxr-xr-x 2 ukmorofova studsci 2048 anp 28 10:49 play

```

```

ukmorofova@dk4n58 ~ $ chmod go-r play
ukmorofova@dk4n58 ~ $ ls -l play

```

```

drwx--x--x 2 ukmorofova studsci 2048 anp 28 10:49 play

```

3. Для файла `my_os` `-r-xr-r-` (рис. 30)

```

ukmorofova@dk4n58 ~ $ touch my_os
ukmorofova@dk4n58 ~ $ ls -l my_os

```

```

-rw-r--r-- 1 ukmorofova studsci 0 anp 28 10:50 my_os

```

```

ukmorofova@dk4n58 ~ $ chmod u-w my_os

```

```

ukmorofova@dk4n58 ~ $ chmod u+x my_os

```

```

ukmorofova@dk4n58 ~ $ ls -l my_os

```

```

-r-xr--r-- 1 ukmorofova studsci 0 anp 28 10:50 my_os

```

4. Для файла feathers -rw-rw-r- (рис.31)

```
ukmorozova@dk4n58 ~ $ touch feathers
ukmorozova@dk4n58 ~ $ ls -l feathers
-rw-r--r-- 1 ukmorozova studsci 0 anp 28 10:52 feathers
ukmorozova@dk4n58 ~ $ chmod g-w feathers
ukmorozova@dk4n58 ~ $ ls -l feathers
-rw-r--r-- 1 ukmorozova studsci 0 anp 28 10:52 feathers
ukmorozova@dk4n58 ~ $ chmod g+w feathers
ukmorozova@dk4n58 ~ $ ls -l feathers
-rw-rw-r-- 1 ukmorozova studsci 0 anp 28 10:52 feathers
```

4. Выполним следующие задания.

5. Просмотрим содержимое файла /etc/passwd (рис.32).

```
ukmorozova@dk4n58 /etc $ cat 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:cron:/var/spool/cron:/bin/false
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
```

Рис. 0.19.: Команда cat

2. Скопируем файл ~/feathers в файл ~/file.old (рис.33).

```
ukmorozova@dk3n56 ~ $ cp feathers file.old
ukmorozova@dk3n56 ~ $ ls
abcl          GNUstep      play         work         Общедоступные
Architecture_PC Hugo         public       Видео        'Рабочий стол'
australia     lab01       public_html  Документы    Шаблоны
C_programs    may         reports      Загрузки
feathers      monthly    sci.plases   Изображения
file.old      my_os      tmp          Музыка
```

Рис. 0.20.: Копирование

3. Переместим файл ~/file.old в каталог ~/play (рис.34).

```
ukmorofova@dk3n56 ~ $ cp file.old play
ukmorofova@dk3n56 ~ $ ls plpay
ls: невозможно получить доступ к 'plpay': Нет такого файла или каталога
ukmorofova@dk3n56 ~ $ ls play
file.old game
```

Рис. 0.21.: Перемещение

4. Скопируем каталог ~/play в каталог ~/fun (рис.35).

```
ukmorofova@dk4n58 ~ $ mkdir fun
ukmorofova@dk4n58 ~ $ cp play fun
cp: не указан -r; пропускается каталог 'play'
ukmorofova@dk4n58 ~ $ cp -r play fun
ukmorofova@dk4n58 ~ $ ls fun
play
```

Рис. 0.22.: Копирование каталогов

5. Переместим каталог ~/fun в каталог ~/play и назовем его games (рис.36).

```
ukmorofova@dk4n58 ~ $ mv fun play
ukmorofova@dk4n58 ~ $ mv fun game
mv: не удалось выполнить stat для 'fun': Нет такого файла или каталога
ukmorofova@dk4n58 ~ $ mv play/fun game
ukmorofova@dk4n58 ~ $ ls play
file.old
ukmorofova@dk4n58 ~ $ ls fun
ls: невозможно получить доступ к 'fun': Нет такого файла или каталога
ukmorofova@dk4n58 ~ $ ls
abcl          GNUstep      play          work          Общедоступные
Architecture_PC Hugo         public        Видео         'Рабочий стол'
australia     lab01        public_html   Документы     Шаблоны
C_programs    may          reports       Загрузки
feathers       monthly      sci.places    Изображения
game          my_os        tmp           Музыка
ukmorofova@dk4n58 ~ $ ls game
play
ukmorofova@dk4n58 ~ $ mv game play
ukmorofova@dk4n58 ~ $ ls play
file.old game
```

Рис. 0.23.: Переименование каталога

6. Лишим владельца файла ~/feathers права на чтение (рис.37). При попытке открыть этот файл консоль выдала ошибку, так как у нас нет доступа для чтения этого файла, тоже самое произошло, когда мы попытались скопировать файл. Вернем владельцу файла право на чтение (рис.38)

```
ukmorofova@dk4n58 ~ $ chmod u-r feathers
ukmorofova@dk4n58 ~ $ cat feathers
cat: feathers: Отказано в доступе
```

Рис. 0.24.: Изменение прав доступа

```
ukmorofova@dk4n58 ~ $ chmod u+r feathers
ukmorofova@dk4n58 ~ $ ls -l feathers
-rw-rw-r-- 1 ukmorofova studsci 0 anp 28 10:52 feathers
```

Рис. 0.25.: Изменение прав доступа

7. Лишим владельца каталога ~/play права на выполнение (рис.39) и попробуем перейти в этот каталог, что успешно получилось. Вернем права на выполнение (рис.40).

```
ukmorofova@dk4n58 ~ $ chmod u-x play
ukmorofova@dk4n58 ~ $ cd play
ukmorofova@dk4n58 ~/play $ cd
```

Рис. 0.26.: Изменение прав доступа

```
ukmorofova@dk4n58 ~ $ chmod u+x play
```

Рис. 0.27.: Изменение прав доступа

5. С помощью команды man прочитаем о командах mount, fsck, mkfs, kill (рис.41-44).

```

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|rpri-
    vate|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 sev-

```

Рис. 0.28.: mount

```

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 filesys-
    tems. filesystems 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 status returned by fsck is the sum of the following condi-
    tions:

```

Рис. 0.29.: fsck

```

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.

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

```

Рис. 0.30.: mkfs

```

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. Particularly useful signals include HUP, INT, KILL, STOP,
    CONT, and 0. Alternate signals 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 output. 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>

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

```

Рис. 0.31.: kill

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой mount без параметров.

С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы.

Команда `mkfs` нужна для того, чтобы создавать файловые системы Linux.

Команда `kill` “убивает” запущенные рабочие процессы.

Выводы

Я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов, приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

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: 16TB; 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 вы-

водить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды `cat`. Основные опции: `-с` – 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 `rsync` позволяет полностью копировать файлы и директории. Синтаксис: `rsync [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 `-r` – 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 r чтение w запись x выполнение u (user) владелец файла g (group) группа, к которой принадлежит владелец файла o (others) все остальные.