

Лабораторная работа №5

Дисциплина: Операционные системы

Старикова Евгения Дмитриевна

Содержание

Цель работы	1
Выполнение лабораторной работы	1
Выводы	8
Контрольные вопросы.....	9

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. Выполнить все примеры, приведённые в верхней части ЛР.
 - изучаем команды для работы с файлами и каталогами (touch, cat, less, head, tail) (рис.1 [-@fig:001])

```
Обзор  Терминал
[edstarikova@fedora ~]$ cd laboratory/
[edstarikova@fedora laboratory]$ touch 123
[edstarikova@fedora laboratory]$ ls
123  Makefile  README.docx  README.md  лаба2  лаба3  лаба4
[edstarikova@fedora laboratory]$ cat 123
[edstarikova@fedora laboratory]$ less 123
[edstarikova@fedora laboratory]$ less README.docx
"README.docx" may be a binary file.  See it anyway?
[edstarikova@fedora laboratory]$ head -5 123
dtvhjsecfhrdskbvftb
dkhbfvjvksd
dsujvbkdsbvuuj
dhnvidsrhmv
dvkigtd
[edstarikova@fedora laboratory]$ tail -3 123
uihdjih
sjdedfhbd
[edstarikova@fedora laboratory]$ cp 123 bin
[edstarikova@fedora laboratory]$ touch 321
[edstarikova@fedora laboratory]$ cp 123 321
[edstarikova@fedora laboratory]$ mv 123 old
```

рис. 1

1. Копируем файл /usr/include/sys/io.h в домашний каталоги назовите его equipment. (рис.2 [-@fig:002])

```
[edstarikova@fedora laboratory]$ mv 123 old
[edstarikova@fedora laboratory]$ cd
[edstarikova@fedora ~]$ cp /usr/include/sys/io.h home
[edstarikova@fedora ~]$ cp /usr/include/sys/io.h ~/home
[edstarikova@fedora ~]$ cp /usr/include/sys/io.h ~/home
[edstarikova@fedora ~]$ mv home equipment
```

2. В домашнем каталоге создаём директорию ~/ski.plases. Перемещаем файл equipment в каталог ~/ski.plases. Переименовываем файл ~/ski.plases/equipment в ~/ski.plases/equiplist. (рис.3 [-@fig:003])

```
[edstarikova@fedora ~]$ mkdir ~/ski.plases
[edstarikova@fedora ~]$ ls
bin  equipment  gitflow  gitflow-installer.sh  homework  laboratory
[edstarikova@fedora ~]$ mv equipment ~/ski.plases
[edstarikova@fedora ~]$ mv ~/ski.plases/equipment ~/ski.plases/equiplist
[edstarikova@fedora ~]$ touch abc1
```

3. Создаём в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.plases, называем его equiplist2 (рис.4 [-@fig:004])

```
[edstarikova@fedora ~]$ touch abc1
[edstarikova@fedora ~]$ cp abc1 ~/ski.plases/
[edstarikova@fedora ~]$ cd ski.plases/
[edstarikova@fedora ski.plases]$ mv abc1 equiplist2
[edstarikova@fedora ski.plases]$ mkdir equipment
[edstarikova@fedora ski.plases]$ mv equiplist equipment
[edstarikova@fedora ski.plases]$ mv equiplist2 equipment
```

4. Создаём каталог с именем equipment в каталоге ~/ski.places Переместим файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment. (рис.5 [-@fig:005])

```
[edstarikova@fedora ~]$ touch abc1
[edstarikova@fedora ~]$ cp abc1 ~/ski.places/
[edstarikova@fedora ~]$ cd ski.places/
[edstarikova@fedora ski.places]$ mv abc1 equiplist2
[edstarikova@fedora ski.places]$ mkdir equipment
[edstarikova@fedora ski.places]$ mv equiplist equipment
[edstarikova@fedora ski.places]$ mv equiplist2 equipment
```

5. Создаём и переместим каталог ~/newdir в каталог ~/ski.places и назовите его plans. (рис.6 [-@fig:006])

```
[edstarikova@fedora ~]$ mkdir newdir
mkdir: невозможно создать каталог «newdir»: Файл существует
[edstarikova@fedora ~]$ mv newdir ski.places/
[edstarikova@fedora ~]$ cd ski.places/
[edstarikova@fedora ski.places]$ mv newdir plans
[edstarikova@fedora ski.places]$ cd plans/
```

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

- drwxr-r- ... australia (рис.7 [-@fig:007])

```
[edstarikova@fedora plans]$ chmod u+x australia
[edstarikova@fedora plans]$ chmod u+w australia
[edstarikova@fedora plans]$ chmod u+r australia
[edstarikova@fedora plans]$ chmod g+r australia
[edstarikova@fedora plans]$ chmod o+r australia
[edstarikova@fedora plans]$ ls =l
ls: невозможно получить доступ к '=l': Нет такого файла или каталога
[edstarikova@fedora plans]$ ls -l
итого 0
-rwxrw-r--. 1 edstarikova edstarikova 0 мая  2 20:55 australia
[edstarikova@fedora plans]$ ls -l australia
-rwxrw-r--. 1 edstarikova edstarikova 0 мая  2 20:55 australia
[edstarikova@fedora plans]$ chmod g-w australia
[edstarikova@fedora plans]$ ls -l australia
-rwxr--r--. 1 edstarikova edstarikova 0 мая  2 20:55 australia
```

- drwx-x-x ... play (рис.8 [-@fig:008])

```
[edstarikova@fedora plans]$ touch play
[edstarikova@fedora plans]$ chmod u+r play
[edstarikova@fedora plans]$ chmod u+w play
[edstarikova@fedora plans]$ chmod u+x play
[edstarikova@fedora plans]$ chmod g+x play
[edstarikova@fedora plans]$ chmod o+x play
[edstarikova@fedora plans]$ ls -l play
-rwxrwxr-x. 1 edstarikova edstarikova 0 мая  2 21:05 play
[edstarikova@fedora plans]$ chmod g-r play
[edstarikova@fedora plans]$ chmod g-w play
[edstarikova@fedora plans]$ chmod o-r play
[edstarikova@fedora plans]$ ls -l play
-rwx--x--x. 1 edstarikova edstarikova 0 мая  2 21:05 play
```

- -r-xr-r- ... my_os (рис.9 [-@fig:009])

```
[edstarikova@fedora plans]$ touch my_os
[edstarikova@fedora plans]$ chmod u+r my_os
[edstarikova@fedora plans]$ chmod u+x my_os
[edstarikova@fedora plans]$ chmod u-w my_os
[edstarikova@fedora plans]$ chmod g+r my_os
[edstarikova@fedora plans]$ chmod o+r my_os
[edstarikova@fedora plans]$ ls -l my_os
-r-xrw-r--. 1 edstarikova edstarikova 0 мая  2 21:07 my_os
[edstarikova@fedora plans]$ chmod g-w my_os
[edstarikova@fedora plans]$ ls -l my_os
-r-xr--r--. 1 edstarikova edstarikova 0 мая  2 21:07 my_os
```

- -rw-rw-r- ... feathers (рис.10 [-@fig:010])

Обзор ☒ Терминал

```
[edstarikova@fedora plans]$ chmod g+r feathers
[edstarikova@fedora plans]$ chmod o+r feathers
[edstarikova@fedora plans]$ ls -l feathers
-rw-rw-r--. 1 edstarikova edstarikova 0 мая  2 21:09 feathers
```

1. Чтобы просмотреть содержимое файла /etc/passwd используйте команду cat (рис.11 [-@fig:011])

```
[edstarikova@fedora plans]$ cat /etc/passwd
cat: /etc/passwd: Нет такого файла или каталога
[edstarikova@fedora plans]$ less /etc/passwd
/etc/passwd: Нет такого файла или каталога
[edstarikova@fedora plans]$ cd
[edstarikova@fedora ~]$ cat /etc/passwd
cat: /etc/passwd: Нет такого файла или каталога
```

2. Чтобы скопировать файл ~/feathers в файл ~/file.old используем команду cp (рис.12 [-@fig:012])

```
[edstarikova@fedora ~]$ cd ski.places/plans/
[edstarikova@fedora plans]$ cp feathers file.old
[edstarikova@fedora plans]$ mv feathers ~/play
[edstarikova@fedora plans]$ mkdir play
mkdir: невозможно создать каталог «play»: Файл существует
[edstarikova@fedora plans]$ mkdir playy
[edstarikova@fedora plans]$ mkdir fun
```

3. Чтобы переместить файл ~/file.old в каталог ~/play используем команду mv (рис.13 [-@fig:013])

```
[edstarikova@fedora ~]$ cd ski.places/plans/
[edstarikova@fedora plans]$ cp feathers file.old
[edstarikova@fedora plans]$ mv feathers ~/play
[edstarikova@fedora plans]$ mkdir play
mkdir: невозможно создать каталог «play»: Файл существует
[edstarikova@fedora plans]$ mkdir playy
[edstarikova@fedora plans]$ mkdir fun
```

4. Чтобы скопировать каталог ~/play в каталог ~/fun mvdir (рис.14 [-@fig:014])

```
[edstarikova@fedora plans]$ mkdir playy
[edstarikova@fedora plans]$ mkdir fun
[edstarikova@fedora plans]$ cp playy fun
cp: не указан -r; пропускается каталог 'playy'
[edstarikova@fedora plans]$ cp -r playy fun
[edstarikova@fedora plans]$ mv fun playy
[edstarikova@fedora plans]$ mv fun game
mv: не удалось выполнить stat для 'fun': Нет такого файла или каталога
[edstarikova@fedora plans]$ cd playy
[edstarikova@fedora playy]$ mv fun game
```

5. Чтобы переместить каталог ~/fun в каталог ~/play и назовите его games используем команду mv (рис.15 [-@fig:015])

```
[edstarikova@fedora plans]$ mkdir playy
[edstarikova@fedora plans]$ mkdir fun
[edstarikova@fedora plans]$ cp playy fun
cp: не указан -r; пропускается каталог 'playy'
[edstarikova@fedora plans]$ cp -r playy fun
[edstarikova@fedora plans]$ mv fun playy
[edstarikova@fedora plans]$ mv fun game
mv: не удалось выполнить stat для 'fun': Нет такого файла или каталога
[edstarikova@fedora plans]$ cd playy
[edstarikova@fedora playy]$ mv fun game
```


6. Чтобы лишить владельца файла ~/feathers права на чтение используем u-r (рис.16 [-@fig:016])

```
[edstarikova@fedora playy]$ cd ~/ski.plases/plans/  
[edstarikova@fedora plans]$ touch feachers  
[edstarikova@fedora plans]$ chmod u-w feachers  
[edstarikova@fedora plans]$ cat feachers  
[edstarikova@fedora plans]$ chmod u-r feachers  
[edstarikova@fedora plans]$ cat feachers  
cat: feachers: Отказано в доступе  
[edstarikova@fedora plans]$ chmod u+r feachers
```

7. Что произойдёт,если вы попытаетесь просмотреть файл ~/feathers командой cat? (рис.17 [-@fig:017])

```
[edstarikova@fedora playy]$ cd ~/ski.plases/plans/  
[edstarikova@fedora plans]$ touch feachers  
[edstarikova@fedora plans]$ chmod u-w feachers  
[edstarikova@fedora plans]$ cat feachers  
[edstarikova@fedora plans]$ chmod u-r feachers  
[edstarikova@fedora plans]$ cat feachers  
cat: feachers: Отказано в доступе  
[edstarikova@fedora plans]$ chmod u+r feachers
```

#fig:017 width=70% }

8. Чтобы дать владельцу файла ~/feathers право на чтение используем u+r (рис.18 [-@fig:018])

```
[edstarikova@fedora playy]$ cd ~/ski.plases/plans/  
[edstarikova@fedora plans]$ touch feachers  
[edstarikova@fedora plans]$ chmod u-w feachers  
[edstarikova@fedora plans]$ cat feachers  
[edstarikova@fedora plans]$ chmod u-r feachers  
[edstarikova@fedora plans]$ cat feachers  
cat: feachers: Отказано в доступе  
[edstarikova@fedora plans]$ chmod u+r feachers
```

#fig:018 width=70% }

9. Чтобы лишить владельца каталога ~/play права на выполнение используем u-x (рис.19 [-@fig:019])

```
[edstarikova@fedora plans]$ chmod u-x play  
[edstarikova@fedora plans]$ chmod u-x playy  
[edstarikova@fedora plans]$ chmod u+x playy  
[edstarikova@fedora plans]$ man mount
```

#fig:019 width=70% }

10. Чтобы дать владельцу каталога ~/play право на выполнение используем u+x (рис.20 [-@fig:020])

```
[edstarikova@fedora plans]$ chmod u-x play
[edstarikova@fedora plans]$ chmod u-x playy
[edstarikova@fedora plans]$ chmod u+x playy
[edstarikova@fedora plans]$ man mount
```

#fig:020 width=70% }

11. Прочитайте man по командам mount,fsck,mkfs,kill и кратко их охарактеризуйте, приведя примеры. (рис.21 [-@fig:021]) (рис.22 [-@fig:022]) (рис.23 [-@fig:023]) (рис.24 [-@fig:024])

```
Обзор Терминал Пн, 2 мая 22:30
edstarikova@fedora:~/ski.plases/plans

[edstarikova@fedora plans]$ chmod u+x playy
[edstarikova@fedora plans]$ man mount
[edstarikova@fedora plans]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=2188088k,nr_inodes=547022,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=883372k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/sda2 on / type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache,subvolid=258,subvol=/root)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=31,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=15325)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,size=2208428k,nr_inodes=409600,inode64)
/dev/sda2 on /home type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache,subvolid=256,subvol=/home)
/dev/sda1 on /boot type ext4 (rw,relatime,seclabel)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=441684k,nr_inodes=110421,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
[edstarikova@fedora plans]$ man fsck
[edstarikova@fedora plans]$ fsck
fsck из util-linux 2.37.2
e2fsck 1.46.3 (27-Jul-2021)
/dev/sda1 is mounted.
```

#fig:021 width=70% }

```
[edstarikova@fedora plans]$ fsck /dev/sda1
fsck из util-linux 2.37.2
e2fsck 1.46.3 (27-Jul-2021)
fsck.ext2: Нет такого файла или каталога while trying to open /dev/sda1
Possibly non-existent device?
[edstarikova@fedora plans]$ fsck
fsck из util-linux 2.37.2
e2fsck 1.46.3 (27-Jul-2021)
/dev/sda1 is mounted.
```

WARNING!!! The filesystem is mounted. If you continue you ***WILL*** cause ***SEVERE*** filesystem damage.

```
Do you really want to continue<n>? yes
fsck.ext4: Отказано в доступе while trying to open /dev/sda1
You must have r/w access to the filesystem or be root
```

#fig:022 width=70% }

```
Обзор  Терминал

+

<size>      число используемых блоков устройства
-V, --verbose  подробная информация о выполняемых действиях;
                указание параметра -V более одного раза приведет к тестовому запуску
-h, --help    display this help
-V, --version  display version

Для более детальной информации смотрите mkfs(8).
[edstarikova@fedora plans]$ mkfs edstarikova
mke2fs 1.46.3 (27-Jul-2021)
mkfs.ext2: Нет такого файла или каталога while trying to determine filesystem size
```

#fig:023 width=70% }

```
[edstarikova@fedora plans]$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS      8) SIGFPE     9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT    19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG     24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
[edstarikova@fedora plans]$
```

#fig:024 width=70% }

Выводы

Я ознакомилась с файловой системой, её структурой, именами и содержанием каталогов. Приобрела навыки по выполнению команд.

Контрольные вопросы

1. Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th». На моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. devtmpfs 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 «extent file 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 и 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
 - `r` чтение
 - `w` запись
 - `x` выполнение
 - `u` (user) владелец файла
 - `g` (group) группа, к которой принадлежит владелец файла
 - `o` (others) все остальные