

Лабораторная работа №6

Отчет

Бондарь Алексей Олегович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	21

Список таблиц

Список иллюстраций

2.1	1	7
2.2	2	7
2.3	4	8
2.4	5	9
2.5	6	9
2.6	7	9
2.7	8	10
2.8	9	11
2.9	11	12
2.10	12	13
2.11	13	14

1 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.

2 Выполнение лабораторной работы

1) Для начала выполним примеры, описанные в первой части описания лабораторной работы. (рис. -fig. 2.1):

1. Скопируем файл `~/abc1` в файл `april` и в файл `may`. Для этого создадим файл `abc1`, используя команду `touch abc1`, далее осуществим копирование с помощью команд `ср abc1 april` и `ср abc1 may`.
2. Скопируем файлы `april` и `may` в каталог `monthly`, используя команды `mkdir monthly` – для создания каталога `monthly` и `ср april may monthly` – для копирования.
3. Скопируем файл `monthly/may` в файл с именем `june`. Выполним команды `ср monthly/may monthly/june` и `ls monthly` (для просмотра содержимого каталога).
4. Скопируем каталог `monthly` в каталог `monthly.00`. Для этого создадим каталог `monthly.00` командой `mkdir monthly.00` и осуществим копирование, используя команду `ср -r monthly monthly.00` (команда `ср` с опцией `r` (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).
5. Скопируем каталог `monthly.00` в каталог `/tmp`, используя команду `ср -r monthly.00 /tmp`.

```

aleksei@AlekseiBondar:~$ touch abc1
aleksei@AlekseiBondar:~$ cp abc1 april
aleksei@AlekseiBondar:~$ cp abc1 may
aleksei@AlekseiBondar:~$ mkdir monthly
aleksei@AlekseiBondar:~$ cp april may monthly
aleksei@AlekseiBondar:~$ cp monthly/may monthly/june
aleksei@AlekseiBondar:~$ ls monthly
april  june  may
aleksei@AlekseiBondar:~$ mkdir monthly.00
aleksei@AlekseiBondar:~$ cp -r monthly monthly.00
aleksei@AlekseiBondar:~$ cp -r monthly.00 /tmp
aleksei@AlekseiBondar:~$ clear

```

Рис. 2.1: 1

(рис. -fig. 2.2): 1. Изменим название файла april на july в домашнем каталоге, используя команду «mv april july». 2. Переместим файл july в каталог monthly.00 с помощью команды «mv july monthly.00». Проверим результат командой «ls monthly.00». 3. Переименуем каталог monthly.00 в monthly.01, используя команду «mv monthly.00 monthly.01». 4. Переместим каталог monthly.01 в каталог reports. Для этого создадим каталог reports с помощью команды «mkdir reports» и выполним перемещение командой «mv monthly.01 reports». 5. Переименуем каталог reports/monthly.01 в reports/monthly командой «mv reports/monthly.01 reports/monthly»

```

aleksei@AlekseiBondar:~$ cd
aleksei@AlekseiBondar:~$ mv april july
aleksei@AlekseiBondar:~$ mv july monthly.00
aleksei@AlekseiBondar:~$ ls monthly.00
july  monthly
aleksei@AlekseiBondar:~$ mv monthly.00 monthly.01
aleksei@AlekseiBondar:~$ mkdir reports
aleksei@AlekseiBondar:~$ mv monthly.01 reports
aleksei@AlekseiBondar:~$ mv reports/monthly.01 reports/monthly
aleksei@AlekseiBondar:~$ clear

```

Рис. 2.2: 2

(рис. -fig. ??) и (рис. -fig. 2.3) 1. Создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may»

(изменение прав), «ls -l may» 2. Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле). 3. Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly» (создание каталога), «chmod go-r monthly» (изменение прав). 4. Создаем файл ~/abc1 с правом записи для членов группы, используя команды: «touch abc1» (создание файла), «chmod g+w abc1» (изменение прав).

```
aleksei@alekseiBondar:~$ cd
aleksei@alekseiBondar:~$ touch may
aleksei@alekseiBondar:~$ ls -l may
-rw-rw-r-- 1 aleksei aleksei 0 май 15 20:29 may
aleksei@alekseiBondar:~$ chmod u-x may
aleksei@alekseiBondar:~$ ls -l may
-rwxrwx-r-- 1 aleksei aleksei 0 май 15 20:29 may
aleksei@alekseiBondar:~$ chmod u-x may
aleksei@alekseiBondar:~$ ls -l may
-rw-rw-r-- 1 aleksei aleksei 0 май 15 20:29 may
aleksei@alekseiBondar:~$ cd
aleksei@alekseiBondar:~$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: файл существует
```

```
aleksei@alekseiBondar:~$ chmod go-r monthly
aleksei@alekseiBondar:~$ cd
aleksei@alekseiBondar:~$ touch abc1
aleksei@alekseiBondar:~$ chmod g+w abc1
```

Рис. 2.3: 4

2) Выполняем следующие действия, отображенные на (рис. -fig. 2.4) и (рис. -fig. 2.5) : 1. Копируем файл /usr/include/aio.h (т.к. у меня нет каталога /usr/include/sys/, то беру произвольный файл из каталога /usr/include/) в домашний каталог (команда «cp /usr/include/aio.h ~») и называем его equipment (команда «mv aio.h equipment»). 2. В домашнем каталоге создаем директорию ~/ski.places (команда «mkdir ski.places»). 3. Перемещаем файл equipment в каталог ~/ski.places (команда «mv equipment ski.places»). 4. Переименовываем файл /ski.places/equipment в /ski.places/equiplist (команда «mv ski.places/equipment ski.places/equiplist»). 5. Создаем в домашнем каталоге файл abc1 (команда «touch abc1») и копируем его в каталог ~/ski.places (команда «cp abc1 ski.places»), называем его equiplist2 (команда «mv ski.places/abc1 ski.places/equiplist2»). 6. Создаем каталог с именем equipment в каталоге ~/ski.places (ко-

манда«mkdir ski.plases/equipment»). 7. Перемещаем файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment (команда «mv ski.plases/equiplist ski.plases/equipment»). 8. Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~/ski.plases (команда «mv newdir ski.plases») и называем его plans (команда «mv ski.plases/newdir ski.plases/plans»).

```
aleksei@AlekseiBondar:~$ cp /usr/include/stdio.h -
aleksei@AlekseiBondar:~$ mv stdio.h equipment
aleksei@AlekseiBondar:~$ mkdir ski.plases
aleksei@AlekseiBondar:~$ mv equipment ski.plases
aleksei@AlekseiBondar:~$ mv ski.plases/equipment ski.plases/equiplist
mv: невозможно переместить 'ski.plases/equipment' в 'ski.plases/equiplist'
такого файла или каталога
aleksei@AlekseiBondar:~$ mv ski.plases/equipment ski.plases/equiplist
aleksei@AlekseiBondar:~$ touch abc1
aleksei@AlekseiBondar:~$ cp abc1 ski.plases
aleksei@AlekseiBondar:~$ mv ski.plases/abc1 ski.plases/equiplist2
aleksei@AlekseiBondar:~$ mkdir ski.plases/equipment
```

Рис. 2.4: 5

```
aleksei@AlekseiBondar:~$ mv ski.plases/equiplist ski.plases/equiplist2 ski.plases/equipment
aleksei@AlekseiBondar:~$ mkdir newdir
aleksei@AlekseiBondar:~$ mv newdir ski.plases
aleksei@AlekseiBondar:~$ mv ski.plases/newdir ski.plases/plans
aleksei@AlekseiBondar:~$
```

Рис. 2.5: 6

3) Определяем опции команды `chmod`, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «`mkdir australia`», «`mkdir play`», «`touch my_os`», «`touch feathers`». (рис. -fig. 2.6)

```
aleksei@AlekseiBondar:~$ mkdir australia
aleksei@AlekseiBondar:~$ mkdir play
aleksei@AlekseiBondar:~$ touch my_os
aleksei@AlekseiBondar:~$ touch feathers
```

Рис. 2.6: 7

(рис. -fig. 2.7): `drwxr-r-... australia`: команда «`chmod 744 australia`» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение). `drwx-x-x ... play`: команда «`chmod 711 play`» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и

остальные –только выполнение). -r-xr-r-... my_os: команды «chmod 544 my_os»(это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные –только чтение). -rw-rw-r-... feathers: команды «chmod 664 feathers»(это файл, владелец и группа владельца имеют право на чтение и запись, остальные –только чтение). Командой «ls -l» проверяем правильность выполненных действий.

```

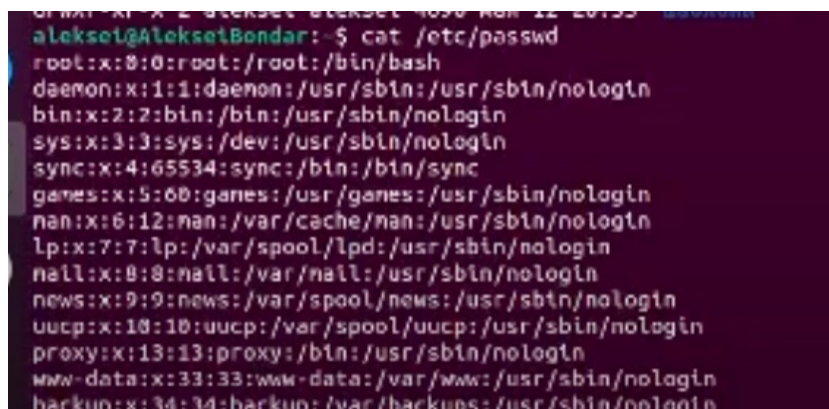
aleksei@alekseiBondar:~$ chmod 744 australia
aleksei@alekseiBondar:~$ chmod 711 play
aleksei@alekseiBondar:~$ chmod 544 my_os
aleksei@alekseiBondar:~$ chmod 664 feathers
aleksei@alekseiBondar:~$ ls -l
итого 68
-rw-rw-r-- 1 aleksei aleksei  0 мая 15 20:36 abc1
drwxr--r-- 2 aleksei aleksei 4096 мая 15 20:42 australia
-rw-rw-r-- 1 aleksei aleksei  0 мая 15 20:42 feathers
drwxrwxr-x 3 aleksei aleksei 4096 мая 13 22:36 lab3
-rw-rw-r-- 1 aleksei aleksei  0 мая 15 20:29 may
drwx-mx--x 2 aleksei aleksei 4096 мая 15 20:23 monthly
-r-xr--r-- 1 aleksei aleksei  0 мая 15 20:42 my_os
drwxrwxr-x 2 aleksei aleksei 4096 мая 14 12:54 pandoc-crossref
drwx--x--x 2 aleksei aleksei 4096 мая 15 20:42 play
drwxrwxr-x 3 aleksei aleksei 4096 мая 15 20:28 reports
drwxrwxr-x 4 aleksei aleksei 4096 мая 15 20:41 ski.places
drwxr-xr-x 3 aleksei aleksei 4096 мая 14 01:18 snap
drwxrwxr-x 4 aleksei aleksei 4096 мая 14 00:04 work
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Видео
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Документы
drwxr-xr-x 3 aleksei aleksei 4096 мая 14 13:26 Загрузки
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Изображения
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Музыка
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Общедоступные
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 'Рабочий стол'
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Шаблоны

```

Рис. 2.7: 8

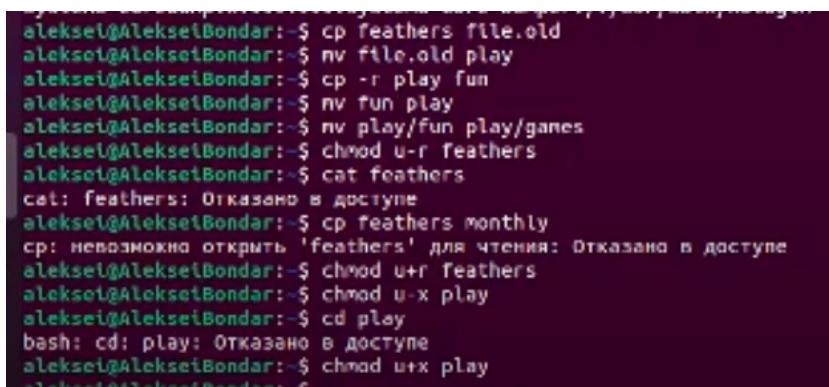
4)Выполняем следующие действия, отображенные на (рис. -fig. 2.8) и (рис. -fig. ??): 1. Просмотрим содержимое файла /etc/passwd (команда «cat/etc/passwd»). 2. Копируем файл ~/feathers в файл ~/file.old(команда «cpfeathersfile.old»). 3. Переместим файл ~/file.old в каталог ~/play(команда «mvfile.ordplay»). 4. Скопируем каталог ~/play в каталог ~/fun(команда «cp-rplayfun»). 5. Переместим каталог ~/fun в каталог ~/play(команда «mvfunplay»)и назовем его games(команда «mvplay/funplay/games»). 6. Лишим владельца файла ~/feathers права на чтение(команда «chmod u-rfeathers»). 7. Если мы попытаемся просмотреть файл ~/feathers командой cat, то получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на чтение данного файла. 8. Если мы попытаемся скопировать файл ~/feathers, например, в каталог monthly, то получим отказ в доступе, по

причине, описанной в предыдущем пункте. 9. Дадим владельцу файла ~/feathers право на чтение (команда «chmod+rfeathers»). 10. Лишим владельца каталога ~/play права на выполнение (команда «chmod-xplay»). 11. Перейдем в каталог ~/play (команда «cd play»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога. 12. Дадим владельцу каталога ~/play право на выполнение (команда «chmod+xplay»)



```
aleksei@AleksaiBondar:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
nail:x:6:12:nail:/var/cache/nail:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

Рис. 2.8: 9



```
aleksei@AleksaiBondar:~$ cp feathers file.old
aleksei@AleksaiBondar:~$ mv file.old play
aleksei@AleksaiBondar:~$ cp -r play fun
aleksei@AleksaiBondar:~$ mv fun play
aleksei@AleksaiBondar:~$ mv play/fun play/games
aleksei@AleksaiBondar:~$ chmod u-r feathers
aleksei@AleksaiBondar:~$ cat feathers
cat: feathers: Отказано в доступе
aleksei@AleksaiBondar:~$ cp feathers monthly
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
aleksei@AleksaiBondar:~$ chmod u+r feathers
aleksei@AleksaiBondar:~$ chmod u-x play
aleksei@AleksaiBondar:~$ cd play
bash: cd: play: Отказано в доступе
aleksei@AleksaiBondar:~$ chmod u+x play
aleksei@AleksaiBondar:~$
```

5) Используя коман-

ды «manmount», «manfsck», «manmkfs», «mankill», получим информацию о соответствующих командах ((рис. -fig. 2.9), (рис. -fig. 2.10), (рис. -fig. 2.11), (рис. -fig. ??)). - Команда mount: предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается как /. 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 fstype device dir`». Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа `fstype`, расположенную на устройстве `device`, к заданному каталогу `dir`, который часто называют точкой монтирования.

```

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
  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

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

```

Рис. 2.9: 11

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


```
F5CK(8)                                System Administration                                F5CK(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.
    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=8968abf6-88c5-4a93-98b8-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 conditions:

        0      No errors
        1      Filesystem errors corrected
        2      System should be rebooted

Manual page fsck(8). Line 1 (press h for help, q to quit)
```

Рис. 2.10: 12

- Команда `mkfs`: создаёт новую файловую систему Linux. Имеет следующий синтаксис: `mkfs[-V] [-t fstype] [fs-options] filesystem[blocks]`. `mkfs` используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента `filesystem` для файловой системы может выступать или название устройства (например, `/dev/hda1`, `/dev/sdb2`) или точка монтирования (например, `/usr`, `/home`). Аргументом `blocks` указывается количество блоков, которые выделяются для использования этой файловой системой. По окончании работы `mkfs` возвращает 0 - в случае успеха, а 1 - при неудачной операции. Например, команда «`mkfs -t ext2 /dev/hdb1`» создаёт файловую систему типа `ext2` в разделе `/dev/hdb1` (второй жёсткий диск).

```

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

```

Рис. 2.11: 13

- Команда kill:посылает сигнал процессу или выводит список допустимых сигналов.Имеет следующий синтаксис: kill[опции] PID, где PID–это PID (числовой идентификатор) процесса или несколько PIDпроцессов, если требуется послать сигнал сразу нескольким процессам.Например, команда «kill -KILL 3121» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс.

```

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

```

3.Контрольные

вопросы: 1)Чтобы узнать, какие файловые системы существуют на жёстком

диске моего компьютера, использую команду «df-Th» (Рисунок 15). Из рисунка видно, что 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 ТБ; максимальный размер раздела: 16 ТБ; 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 утилит `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`) – сим-

волны 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 изменен `-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 командыКоманда renamетакже предназначена,
 чтобы переименовать файл.Синтаксис:rename [опции] старое_имя новое_имя
 файлыОсновные опции:-v -вывести список обработанных файлов-n -тестовый
 режим, на самом деле никакие действия выполнены не будут-f -принудительно
 перезаписывать существующие файлы9)Права доступа- совокупность правил,
 регламентирующих порядок и условия доступа субъекта к объектам информа-
 ционной системы (информации, её носителям, процессам и другим ресурсам)
 установленных правовыми документами или собственником, владельцем
 информации.Права доступа к файлу или каталогу можно изменить, восполь-
 зовавшись командой chmod. Сделать это может владелец файла (или каталога)
 или пользователь с правами администратора.Синтаксискоманды:chmod режим
 имя_файлаРежим имеет следующие компоненты структуры и способзаписи:=
 установить право-лишить права+ датьправог чтениеw записьх выполнениеu
 (user) владелец файлаg (group) группа, к которой принадлежит владелец файлаo
 (others) все оста

3 Выводы

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