

Отчёта по лабораторной работе №5 :

Анализ файловой системы Linux. Команды для работы с
файлами и каталогами

Сомсиков Даниил Сергеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	8
4	Выполнение лабораторной работы	10
5	Контрольные вопросы	24
6	Выводы	32
	Список литературы	33

Список иллюстраций

4.1 Копирование файла в текущем каталоге	10
4.2 Копирование нескольких файлов в каталог	10
4.3 Копирование файлов в произвольном каталоге	11
4.4 Копирование каталогов в текущем и произвольном каталогах	11
4.5 Переименование и перемещение файлов в другой каталог	12
4.6 Переименование и перемещение каталога	12
4.7 Установка прав	12
4.8 Копирование файла io.h	13
4.9 Создание директории, перемещение и переименование файла	13
4.10Создание файла и копирование его в каталог	13
4.11Перемещение файлов в новый каталог	14
4.12Новый каталог	14
4.13Изменение прав каталога	15
4.14Изменение прав каталога	16
4.15Изменение прав файла	16
4.16Изменение прав файла	17
4.17Содержимое файла /etc/passwd	17
4.18Копирование и перемещение файла	18
4.19Перемещение каталога	18
4.20Изменение прав владельца файла	18
4.21Изменение прав владельца каталога	19
4.22Команда mount	20
4.23Команда fsck	21
4.24Команда mkfs	22
4.25Команда kill	23

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 Задание

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 - Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него.
 - В домашнем каталоге создайте директорию `~/ski.plases`.
 - Переместите файл `equipment` в каталог `~/ski.plases`.
 - Переименуйте файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`.
 - Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.plases`, назовите его `equiplist2`.
 - Создайте каталог с именем `equipment` в каталоге `~/ski.plases`.
 - Переместите файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment`.
 - Создайте и переместите каталог `~/newdir` в каталог `~/ski.plases` и назовите его `plans`.

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

- `drwxr-r- ... australia`
- `drwx-x-x ... play`
- `-r-xr-r- ... my_os`
- `-rw-rw-r- ... feathers`

При необходимости создайте нужные файлы.

4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:

- Просмотрите содержимое файла `/etc/password`.
- Скопируйте файл `~/feathers` в файл `~/file.old`.
- Переместите файл `~/file.old` в каталог `~/play`.
- Скопируйте каталог `~/play` в каталог `~/fun`.
- Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`.
- Лишите владельца файла `~/feathers` права на чтение.
- Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`?
- Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`?
- Дайте владельцу файла `~/feathers` право на чтение.

- Лишите владельца каталога ~/play права на выполнение.
 - Перейдите в каталог ~/play. Что произошло?
 - Дайте владельцу каталога ~/play право на выполнение.
5. Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры.

3 Теоретическое введение

Приведём основные команды Unix, которые мы будем использовать в процессе выполнения лабораторной работы №4.

В табл. tbl. 3.1 приведен краткий справочник по командам терминала Unix-систем.

Таблица 3.1: Краткий справочник по командам терминала GNU Linux

Имя	
коман-	
ды	Описание команды
touch	Создание файла
имя_файла	
cat	Просмотр содержимого файлов небольшого размера
имя_файла	
less	Просмотр файла постранично
имя_файла	
head	Вывод по умолчанию 10 первых строк файла или n строк
[-n]	
имя_файла	
tail	Вывод последних n строк файла
[-n]	
имя_файла	

Имя	
коман-	
ды	Описание команды
ср	Копирование файлов и каталогов
исход-	
ная_директория	
целе-	
вая_директория	
mkdir	Создание директории
имя_директории	
mv	Перемещение (переименование) каталогов
ста-	
рый_файл	
но-	
вый_файл	

4 Выполнение лабораторной работы

1. Выполним все примеры, приведённые в первой части описания лабораторной работы.

Для копирования файлов и каталогов будем использовать команду `cp [-опции] исходный_файл целевой_файл`, а для их перемещения и переименования будем использовать команду `mv [-опции] старый_файл новый_файл`:

Копирование файла в текущем каталоге - скопировать файл `~/abc1` в файл `april` и в файл `may` (рис. 4.1).

```
dssomsikov@dssomsikov:~$ touch abc1
dssomsikov@dssomsikov:~$ cp abc1 april
dssomsikov@dssomsikov:~$ cp abc1 may
dssomsikov@dssomsikov:~$ ls
abc1  may  work  Документы  Изображения  Общедоступные  Шаблоны
april  pandoc-crossref  Видео  Загрузки  Музыка  'Рабочий стол'
```

Рис. 4.1: Копирование файла в текущем каталоге

Копирование нескольких файлов в каталог - скопировать файлы `april` и `may` в каталог `monthly` (рис. 4.2).

```
dssomsikov@dssomsikov:~$ mkdir monthly
dssomsikov@dssomsikov:~$ cp april may monthly
dssomsikov@dssomsikov:~$ ls monthly/
april  may
```

Рис. 4.2: Копирование нескольких файлов в каталог

Копирование файлов в произвольном каталоге - скопировать файл `monthly/may` в файл с именем `june` (рис. 4.3).

```
dssomsikov@dssomsikov:~$ cp monthly/may monthly/june
dssomsikov@dssomsikov:~$ ls monthly/
april june may
dssomsikov@dssomsikov:~$
```

Рис. 4.3: Копирование файлов в произвольном каталоге

Копирование каталогов в текущем и произвольном каталогах - скопировать каталог `monthly` в каталог `monthly.00`, скопировать каталог `monthly.00` в каталог `/tmp` (рис. 4.4).

```
dssomsikov@dssomsikov:~$ mkdir monthly.00
dssomsikov@dssomsikov:~$ cp -r month
monthly.00/ monthly/
dssomsikov@dssomsikov:~$ cp -r monthly monthly.00/
dssomsikov@dssomsikov:~$ ls monthly
april june may
dssomsikov@dssomsikov:~$ ls monthly.00/
monthly
dssomsikov@dssomsikov:~$ cp -r monthly.00/ /tmp/
dssomsikov@dssomsikov:~$ ls /tmp/
dvipdfmx.mnp3zk
monthly.00
systemd-private-1a5e174afdc34d5596925da4d5e395ee-abrtd.service-UAsubj
systemd-private-1a5e174afdc34d5596925da4d5e395ee-chronyd.service-8wcp63
systemd-private-1a5e174afdc34d5596925da4d5e395ee-colord.service-PiemuV
systemd-private-1a5e174afdc34d5596925da4d5e395ee-dbus-broker.service-1PZ2o0
systemd-private-1a5e174afdc34d5596925da4d5e395ee-low-memory-monitor.service-8J7m8z
systemd-private-1a5e174afdc34d5596925da4d5e395ee-ModemManager.service-EDl7RW
systemd-private-1a5e174afdc34d5596925da4d5e395ee-passim.service-UUmXa7
systemd-private-1a5e174afdc34d5596925da4d5e395ee-polkit.service-vdRC6K
systemd-private-1a5e174afdc34d5596925da4d5e395ee-power-profiles-daemon.service-hcu4Mq
systemd-private-1a5e174afdc34d5596925da4d5e395ee-rtkit-daemon.service-7Bd6Mw
systemd-private-1a5e174afdc34d5596925da4d5e395ee-switcheroo-control.service-hiw3xf
systemd-private-1a5e174afdc34d5596925da4d5e395ee-systemd-logind.service-ULPRn5
systemd-private-1a5e174afdc34d5596925da4d5e395ee-systemd-oond.service-Tl46xG
systemd-private-1a5e174afdc34d5596925da4d5e395ee-systemd-resolved.service-NsF4p2
systemd-private-1a5e174afdc34d5596925da4d5e395ee-upower.service-29GIQm
dssomsikov@dssomsikov:~$
```

Рис. 4.4: Копирование каталогов в текущем и произвольном каталогах

Переименование файлов в текущем каталоге - изменить название файла `april` на `july` в домашнем каталоге. Перемещение файлов в другой каталог - переместить файл `july` в каталог `monthly.00` (рис. 4.5).

```
dssomsikov@dssomsikov:~$ mv april july
dssomsikov@dssomsikov:~$ mv july monthly.00/
dssomsikov@dssomsikov:~$ ls monthly.00/
july  monthly
dssomsikov@dssomsikov:~$
```

Рис. 4.5: Переименование и перемещение файлов в другой каталог

Переименование каталогов в текущем каталоге - переименовать каталог `monthly.00` в `monthly.01`. Перемещение каталога в другой каталог - переместить каталог `monthly.01` в каталог `reports`. Переименование каталога, не являющегося текущим - переименовать каталог `reports/monthly.01` в `reports/monthly` (рис. 4.6).

```
dssomsikov@dssomsikov:~$ mv monthly.00 monthly.01
dssomsikov@dssomsikov:~$ mkdir reports
dssomsikov@dssomsikov:~$
dssomsikov@dssomsikov:~$ mv monthly.01 reports
dssomsikov@dssomsikov:~$ mv reports/monthly.01/ reports/monthly
dssomsikov@dssomsikov:~$ ls reports/
monthly
dssomsikov@dssomsikov:~$
```

Рис. 4.6: Переименование и перемещение каталога

Создадим файл `~/may` с правом выполнения для владельца, а затем лишим владельца прав на выполнение (рис. 4.7).

```
dssomsikov@dssomsikov:~$ touch may
dssomsikov@dssomsikov:~$ ls -l may
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн  4 22:57 may
dssomsikov@dssomsikov:~$ chmod u+x may
dssomsikov@dssomsikov:~$ ls -l may
-rwxr--r--. 1 dssomsikov dssomsikov 0 июн  4 22:57 may
dssomsikov@dssomsikov:~$ chmod u-x may
dssomsikov@dssomsikov:~$ ls -l may
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн  4 22:57 may
dssomsikov@dssomsikov:~$
```

Рис. 4.7: Установка прав

2. Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовем его `equipment`. Если файла `io.h` нет, то используем любой другой файл в каталоге `/usr/include/sys/` вместо него (рис. 4.8).

```
dssomsikov@dssomsikov:~$ cp /usr/include/sys/io.h equipment
dssomsikov@dssomsikov:~$ ls
abc1      may      pandoc-crossref  work  Документы  Изображения  Общедоступные  Шаблоны
equipment  monthly  reports          Видео  Загрузки   Музыка        'Рабочий стол'
```

Рис. 4.8: Копирование файла io.h

В домашнем каталоге создадим директорию ~/ski.places и переместим файл equipment в новый каталог. После переименуем файл equiplist (рис. 4.9).

```
dssomsikov@dssomsikov:~$ mkdir ski.places
dssomsikov@dssomsikov:~$ mv equipment ski.places
dssomsikov@dssomsikov:~$ ls ski.places/
equipment
dssomsikov@dssomsikov:~$ mv ski.places/equipment ski.places/equiplist
dssomsikov@dssomsikov:~$ ls ski.places/
equiplist
dssomsikov@dssomsikov:~$
```

Рис. 4.9: Создание директории, перемещение и переименование файла

Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.places и назовем его equiplist2 (рис. 4.10).

```
dssomsikov@dssomsikov:~$ touch abc1
dssomsikov@dssomsikov:~$ cp abc1 ski.places/
dssomsikov@dssomsikov:~$ ls ski.places/
abc1  equiplist
dssomsikov@dssomsikov:~$ mv ski.places/abc1 ski.places/equiplist2
dssomsikov@dssomsikov:~$ ls ski.places/
equiplist  equiplist2
dssomsikov@dssomsikov:~$
```

Рис. 4.10: Создание файла и копирование его в каталог

Создадим каталог с именем equipment в каталоге ~/ski.places. Затем переместим файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment (рис. 4.11).

```

dssomsikov@dssomsikov:~$ mkdir ~/ski.places/equipment
dssomsikov@dssomsikov:~$ ls ski.places/
equiplist  equiplist2  equipment
dssomsikov@dssomsikov:~$ mv ski.places/equiplist ski.places/equipment
dssomsikov@dssomsikov:~$ mv ski.places/equiplist2 ski.places/equipment
dssomsikov@dssomsikov:~$ ls ski.places/equipment/
equiplist  equiplist2
dssomsikov@dssomsikov:~$

```

Рис. 4.11: Перемещение файлов в новый каталог

Создадим и переместим каталог ~/newdir в каталог ~/ski.places и назовем его plans. (рис. 4.12).

```

dssomsikov@dssomsikov:~$ mkdir newdir
dssomsikov@dssomsikov:~$ mv newdir ski.places/plans
dssomsikov@dssomsikov:~$ ls ski.places/
equipment  plans
dssomsikov@dssomsikov:~$

```

Рис. 4.12: Новый каталог

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

- `drwxr-r- ... australia`: создадим новый каталог `australia`, проверим какие у него права, а после, в моем случае, используем `chmod g-x,o-x australia`, чтобы изменить на нужные права (рис. 4.13).

```

dssomsikov@dssomsikov:~$ mkdir australia
dssomsikov@dssomsikov:~$ ls -l
итого 0
-rw-r--r--. 1 dssomsikov dssomsikov  0 июн  4 23:02 abcl
drwxr-xr-x. 1 dssomsikov dssomsikov  0 июн  4 23:06 australia
-rw-r--r--. 1 dssomsikov dssomsikov  0 июн  4 22:57 may
drwxr-xr-x. 1 dssomsikov dssomsikov 24 июн  4 22:42 monthy
drwxr-xr-x. 1 dssomsikov dssomsikov 612 июн  3 13:36 pandoc-crossref
drwxr-xr-x. 1 dssomsikov dssomsikov  14 июн  4 22:56 reports
drwxr-xr-x. 1 dssomsikov dssomsikov 28 июн  4 23:05 ski.places
drwxr-xr-x. 1 dssomsikov dssomsikov  10 июн  1 19:57 work
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Видео
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Документы
drwxr-xr-x. 1 dssomsikov dssomsikov 130 июн  4 12:46 Загрузки
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Изображения
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Музыка
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Общедоступные
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 'Рабочий стол'
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Шаблоны
dssomsikov@dssomsikov:~$ chmod g-x,o-x australia
dssomsikov@dssomsikov:~$ ls -l
итого 0
-rw-r--r--. 1 dssomsikov dssomsikov  0 июн  4 23:02 abcl
drwxr--r--. 1 dssomsikov dssomsikov  0 июн  4 23:06 australia
-rw-r--r--. 1 dssomsikov dssomsikov  0 июн  4 22:57 may
drwxr-xr-x. 1 dssomsikov dssomsikov 24 июн  4 22:42 monthy
drwxr-xr-x. 1 dssomsikov dssomsikov 612 июн  3 13:36 pandoc-crossref
drwxr-xr-x. 1 dssomsikov dssomsikov  14 июн  4 22:56 reports
drwxr-xr-x. 1 dssomsikov dssomsikov 28 июн  4 23:05 ski.places
drwxr-xr-x. 1 dssomsikov dssomsikov  10 июн  1 19:57 work
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Видео
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Документы
drwxr-xr-x. 1 dssomsikov dssomsikov 130 июн  4 12:46 Загрузки
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Изображения
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Музыка
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Общедоступные
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 'Рабочий стол'
drwxr-xr-x. 1 dssomsikov dssomsikov  0 мая  5 19:12 Шаблоны
dssomsikov@dssomsikov:~$

```

Рис. 4.13: Изменение прав каталога

- drwx-x-x ... play: делаем то же самое, но используем `chmod g-r,o-r play` (рис. 4.14).


```

dssomsikov@dssomsikov:~$ mkdir play
dssomsikov@dssomsikov:~$ ls -l
итого 0
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн 4 23:02 abc1
drwxr--r--. 1 dssomsikov dssomsikov 0 июн 4 23:06 australia
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн 4 22:57 may
drwxr-xr-x. 1 dssomsikov dssomsikov 24 июн 4 22:42 monthy
drwxr-xr-x. 1 dssomsikov dssomsikov 612 июн 3 13:36 pandoc-crossref
drwxr-xr-x. 1 dssomsikov dssomsikov 0 июн 4 23:07 play
drwxr-xr-x. 1 dssomsikov dssomsikov 14 июн 4 22:56 reports
drwxr-xr-x. 1 dssomsikov dssomsikov 28 июн 4 23:05 ski.places
drwxr-xr-x. 1 dssomsikov dssomsikov 10 июн 1 19:57 work
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Видео
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Документы
drwxr-xr-x. 1 dssomsikov dssomsikov 130 июн 4 12:46 Загрузки
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Изображения
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Музыка
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Общедоступные
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 'Рабочий стол'
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Шаблоны
dssomsikov@dssomsikov:~$ chmod g-r,o-r play
dssomsikov@dssomsikov:~$ ls -l
итого 0
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн 4 23:02 abc1
drwxr--r--. 1 dssomsikov dssomsikov 0 июн 4 23:06 australia
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн 4 22:57 may
drwxr-xr-x. 1 dssomsikov dssomsikov 24 июн 4 22:42 monthy
drwx--x--x. 1 dssomsikov dssomsikov 0 июн 4 23:07 play
drwxr-xr-x. 1 dssomsikov dssomsikov 14 июн 4 22:56 reports
drwxr-xr-x. 1 dssomsikov dssomsikov 28 июн 4 23:05 ski.places
drwxr-xr-x. 1 dssomsikov dssomsikov 10 июн 1 19:57 work
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Видео
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Документы
drwxr-xr-x. 1 dssomsikov dssomsikov 130 июн 4 12:46 Загрузки
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Изображения
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Музыка
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Общедоступные
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 'Рабочий стол'
drwxr-xr-x. 1 dssomsikov dssomsikov 0 мая 5 19:12 Шаблоны
dssomsikov@dssomsikov:~$

```

Рис. 4.14: Изменение прав каталога

- `-r-xr-r-` ... `my_os`: создаем в этот раз уже файл `my_os` и пишем команду `chmod u-w,u+x my-os` (рис. 4.15).

```

dssomsikov@dssomsikov:~$ touch my_os
dssomsikov@dssomsikov:~$ ls -l my_os
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн 4 23:12 my_os
dssomsikov@dssomsikov:~$ chmod u-w,u+x my_os
dssomsikov@dssomsikov:~$ ls -l my_os
-r-xr--r--. 1 dssomsikov dssomsikov 0 июн 4 23:12 my_os
dssomsikov@dssomsikov:~$

```

Рис. 4.15: Изменение прав файла

- `-rw-rw-r-` ... `feathers`: здесь уже используем `chmod g+w feathers` (рис. 4.16).


```
dssomsikov@dssomsikov:~$ touch feathers
dssomsikov@dssomsikov:~$ ls -l feathers
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн  4 23:13 feathers
dssomsikov@dssomsikov:~$ chmod g+w feathers
dssomsikov@dssomsikov:~$ ls -l feathers
-rw-rw-r--. 1 dssomsikov dssomsikov 0 июн  4 23:13 feathers
dssomsikov@dssomsikov:~$
```

Рис. 4.16: Изменение прав файла

4. Просмотрим содержимое файла `/etc/passwd`, используя команду `cat` (рис. 4.17).

```
dssomsikov@dssomsikov:~$ cat /etc/passwd
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/sbin:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/sbin/nologin
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:/var/spool/mail:/usr/sbin/nologin
operator:x:11:0:operator:/root:/usr/sbin/nologin
games:x:12:100:games:/usr/games:/usr/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/usr/sbin/nologin
dbus:x:81:81:System Message Bus:/:/usr/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/usr/sbin/nologin
systemd-oom:x:998:998:systemd Userspace OOM Killer:/:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
polkitd:x:114:114:User for polkitd:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
unbound:x:996:996:Unbound DNS resolver:/var/lib/unbound:/sbin/nologin
geoclue:x:995:994:User for geoclue:/var/lib/geoclue:/sbin/nologin
nm-openconnect:x:994:992:NetworkManager user for OpenConnect:/:/sbin/nologin
gluster:x:993:990:GlusterFS daemons:/run/gluster:/sbin/nologin
saslauthd:x:992:76:Saslauthd user:/run/saslauthd:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/:/sbin/nologin
pipewire:x:991:989:PipeWire System Daemon:/run/pipewire:/usr/sbin/nologin
chrony:x:990:988:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:989:987:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
openvpn:x:988:986:OpenVPN:/etc/openvpn:/sbin/nologin
nm-openvpn:x:987:985:Default user for running openvpn spawned by NetworkManager:/:/sbin/nologin
colord:x:986:984:User for colord:/var/lib/colord:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
abrt:x:173:173:/:/etc/abrt:/sbin/nologin
wsdd:x:985:982:Web Services Dynamic Discovery host daemon:/:/sbin/nologin
flatpak:x:984:981:Flatpak system helper:/:/usr/sbin/nologin
```

Рис. 4.17: Содержимое файла `/etc/passwd`

Скопируем файл `~/feathers` в файл `~/file.old`, а затем переместим файл `~/file.old` в каталог `~/play` (рис. 4.18).

```

dssomsikov@dssomsikov:~$ cp feathers file.old
dssomsikov@dssomsikov:~$ ls
abcl      file.old  my_os      reports    Видео      Изображения  'Рабочий стол'
australia may      pandoc-crossref ski.places  Документы    Музыка        Шаблоны
feathers  monthly  play       work       Загрузки     Общедоступные
dssomsikov@dssomsikov:~$ mv file.old play
dssomsikov@dssomsikov:~$ ls play/
file.old
dssomsikov@dssomsikov:~$

```

Рис. 4.18: Копирование и перемещение файла

Скопируем каталог ~/play в каталог ~/fun. После переместим каталог ~/fun в каталог ~/play и назовите его games (рис. 4.19).

```

dssomsikov@dssomsikov:~$ cp -r play/ fun
dssomsikov@dssomsikov:~$ ls fun
file.old  play
dssomsikov@dssomsikov:~$ mv fun/ play/
dssomsikov@dssomsikov:~$ ls pla
ls: невозможно получить доступ к 'pla': Нет такого файла или каталога
dssomsikov@dssomsikov:~$ ls play/
file.old  fun
dssomsikov@dssomsikov:~$ mv fun/ play/fun play/games
mv: цель 'play/games': Нет такого файла или каталога
dssomsikov@dssomsikov:~$ mv play/fun play/games
dssomsikov@dssomsikov:~$ ls play/
file.old  games
dssomsikov@dssomsikov:~$

```

Рис. 4.19: Перемещение каталога

Лишим владельца файла ~/feathers права на чтение, используя команду `chmod u-r feathers`. Если мы попробуем просмотреть файл ~/feathers командой `cat`, то нам будет отказано в доступе, то же самое произойдет, если захотим скопировать данный файл. После вернем все обратно командой `chmod u+r feathers` (рис. 4.20).

```

dssomsikov@dssomsikov:~$ ls -l feathers
-rw-rw-r--. 1 dssomsikov dssomsikov 0 июн  4 23:13 feathers
dssomsikov@dssomsikov:~$ chmod u-r feathers
dssomsikov@dssomsikov:~$ ls -l feathers
--w-rw-r--. 1 dssomsikov dssomsikov 0 июн  4 23:13 feathers
dssomsikov@dssomsikov:~$ cat feathers
cat: feathers: Отказано в доступе
dssomsikov@dssomsikov:~$ cp feathers play/
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
dssomsikov@dssomsikov:~$ chmod u+r feathers
dssomsikov@dssomsikov:~$ ls -l feathers
-rw-rw-r--. 1 dssomsikov dssomsikov 0 июн  4 23:13 feathers
dssomsikov@dssomsikov:~$

```

Рис. 4.20: Изменение прав владельца файла

Лишим теперь владельца каталога ~/play права на выполнение, командой `chmod u-x play`. Попробуем перейти в каталог ~/play, то нам будет отказано в доступе. Вернем владельцу каталога ~/play право на выполнение, командой `chmod u+x play` (рис. 4.21).

```
dssomsikov@dssomsikov:~$ chmod u-x play
dssomsikov@dssomsikov:~$ ls -l play
ls: невозможно получить доступ к 'play/file.old': Отказано в доступе
ls: невозможно получить доступ к 'play/games': Отказано в доступе
итого 0
-???????? ? ? ? ? ? ? file.old
d???????? ? ? ? ? ? ? games
dssomsikov@dssomsikov:~$ cd play/
bash: cd: play/: Отказано в доступе
dssomsikov@dssomsikov:~$ chmod u+x play
dssomsikov@dssomsikov:~$ ls -l play
итого 0
-rw-r--r--. 1 dssomsikov dssomsikov 0 июн  4 23:16 file.old
drwx--x--x. 1 dssomsikov dssomsikov 24 июн  4 23:18 games
dssomsikov@dssomsikov:~$
```

Рис. 4.21: Изменение прав владельца каталога

Прочитаем man по командам `mount`, `fsck`, `mkfs`, `kill`:

- `mount`: команда `mount` используется для подключения файловых систем к каталогу в иерархии файловой системы (рис. 4.22).

```
MOUNT(8)                                     System Administration                                     MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-fFnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable] mountpoint

DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the filesystem found on some device to the big file tree. Conversely, the umount(8) command will detach it again. The filesystem is used to control how data is stored on the device or provided in a virtual way by network or other services.

    The standard form of the mount command is:

        mount -t type device dir

    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. The previous contents (if any) and owner and mode of dir become invisible, and as long as this filesystem remains mounted, the pathname dir refers to the root of the filesystem on device.

    If only the directory or the device is given, for example:

        mount /dir

    then mount looks for a mountpoint (and if not found then for a device) in the /etc/fstab file. It's possible to use the --target or --source options to avoid ambiguous interpretation of the given argument. For example:
```

Рис. 4.22: Команда mount

- **fsck**: команда **fsck** (file system check) предназначена для проверки и исправления целостности файловых систем (рис. 4.23).

```
FCK(8)                                     System Administration                                     FCK(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 a 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

    16
        Usage or syntax error

    32
        Checking canceled by user request

    128
        Shared-library error
```

Рис. 4.23: Команда fsck

- mkfs: команда mkfs (make file system) используется для создания новой файловой системы на указанном устройстве (рис. 4.24).

```

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.

    -h, --help
        Display help text and exit.

    -V, --version
        Print version and exit. (Option -V will display version information only when it is the only parameter, otherwise it will work as --verbose.)

BUGS
    All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific programs do not automatically detect the device size and require the size parameter to

```

Рис. 4.24: Команда mkfs

- **kill**: команда **kill** используется для отправки сигнала процессу или группе процессов, что может привести к их завершению (рис. 4.25).

KILL(1)	User Commands	KILL(1)
NAME		
kill - terminate a process		
SYNOPSIS		
kill [-signal -s <u>signal</u> -p] [-q <u>value</u>] [-a] [--timeout <u>milliseconds</u> <u>signal</u>] [--] <u>pid</u> <u>name</u> ...		
kill -l [<u>number</u>] -L		
DESCRIPTION		
The command kill sends the specified <u>signal</u> to the specified processes or process groups.		
If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.		
Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all , --pid , and --queue options, and the possibility to specify processes by command name, are local extensions.		
If <u>signal</u> is 0, then no actual signal is sent, but error checking is still performed.		
ARGUMENTS		
The list of processes to be signaled can be a mixture of names and PIDs.		
<u>pid</u>		
Each <u>pid</u> can be expressed in one of the following ways:		
<u>n</u>		
where <u>n</u> is larger than 0. The process with PID <u>n</u> is signaled.		
0		
All processes in the current process group are signaled.		
-1		
All processes with a PID larger than 1 are signaled.		
- <u>n</u>		
where <u>n</u> is larger than 1. All processes in process group <u>n</u> are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either a signal must be specified		

Рис. 4.25: Команда kill

5 Контрольные вопросы

1. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

Общая структура файловой системы в большинстве операционных систем может быть представлена следующим образом:

/ - корневая директория

/bin - содержит основные исполняемые файлы, необходимые для загрузки и базовых операций в системе.

/boot - содержит файлы, необходимые для загрузки операционной системы.

/dev - содержит файлы, представляющие устройства в системе.

/etc - содержит конфигурационные файлы для системы и установленных программ.

/home - директории пользователей.

/lib - содержит библиотеки, необходимые для запуска исполняемых файлов в /bin и /sbin.

/media - место для монтирования временных носителей, таких как USB-накопители.

/mnt - место для временного монтирования файловых систем.

/opt - используется для установки “дополнительных” программ.

/proc - виртуальная файловая система, предоставляющая информацию о процессах в системе.

/root - домашняя директория суперпользователя (root).

/run - временная файловая система, хранящая данные, специфичные для текущего запуска системы.

/sbin - содержит основные системные исполняемые файлы.

/srv - содержит данные, используемые локальными сервисами.

/sys - виртуальная файловая система, предоставляющая информацию о ядре Linux.

/tmp - временная директория, доступная всем пользователям для временных файлов.

/usr - вторичная иерархия, содержащая все несистемные программы и данные.

/var - содержит изменяемые файлы, такие как журналы, временные файлы и кэши.

Каждая из этих директорий первого уровня имеет свою уникальную роль в файловой системе операционной системы.

2. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?

Чтобы содержимое некоторой файловой системы стало доступным операционной системе, необходимо выполнить операцию монтирования (mount). Монтирование - это процесс подключения файловой системы к определенной точке в иерархии файловой системы операционной системы. Это позволяет операционной системе видеть и обращаться к содержимому этой файловой системы.

3. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?

Основные причины нарушения целостности файловой системы включают:

- Сбои в питании: Внезапное отключение питания может привести к повреждению файлов, так как операции записи могут быть прерваны, что приводит к неправильному состоянию файловой системы.
- Сбои в работе жёсткого диска или других устройств хранения данных: Неисправности жестких дисков или других устройств хранения данных могут вызвать повреждение файлов и структуры файловой системы.
- Вирусы и вредоносное ПО: Атаки вирусов и вредоносного программного обеспечения могут изменять или удалять файлы, что приводит к нарушению целостности файловой системы.
- Неправильное завершение работы программ или операционной системы: Если программа или операционная система завершается некорректно, это может привести к повреждению файлов и структуры файловой системы.
- Физические повреждения устройств хранения данных: Механические повреждения жестких дисков или других устройств хранения данных могут привести к потере данных и повреждению файловой системы.

Для устранения повреждений файловой системы можно предпринять следующие шаги:

- Проверка и восстановление файловой системы: Использование специальных инструментов, таких как `fsck` в Linux, для проверки и восстановления целостности файловой системы.
- Использование резервных копий: Восстановление поврежденных файлов из резервных копий, если они доступны.

- Использование специализированных программ: Использование специализированных программ для восстановления данных с поврежденных устройств хранения.
- Замена неисправных устройств хранения данных: В случае физических повреждений устройств хранения данных требуется их замена и восстановление данных из резервных копий, если таковые имеются.
- Установка антивирусного ПО и обновление системы: Защита от вирусов и вредоносного ПО путем установки антивирусного программного обеспечения и регулярное обновление операционной системы для устранения уязвимостей.

Кроме того, важно регулярно создавать резервные копии данных, чтобы в случае повреждения файловой системы можно было восстановить данные из последней рабочей копии.

4. Как создаётся файловая система?

- Использовать команду `mkfs`: В большинстве дистрибутивов Linux для создания файловой системы используется команда `mkfs`, например, для создания ext4-файловой системы: `mkfs.ext4 /dev/sdX`.
- Настройка параметров файловой системы: Вы можете указать различные параметры, такие как размер блока, метку тома и другие опции в зависимости от вашего выбора файловой системы.
- Монтирование файловой системы (опционально): После создания файловой системы вы можете выполнить операцию монтирования, чтобы сделать её доступной для использования операционной системой.

5. Дайте характеристику командам для просмотра текстовых файлов.

Команды для просмотра текстовых файлов предоставляют пользователю возможность просматривать содержимое текстовых файлов непосредственно в терминале или командной строке. Вот несколько основных команд для просмотра текстовых файлов:

- `cat`: Команда `cat` (concatenate) используется для отображения содержимого одного или нескольких файлов в терминале. Она также может использоваться для объединения нескольких файлов и вывода содержимого на экран.
- `less`: Команда `less` позволяет просматривать содержимое файла постранично. Это удобно для просмотра больших файлов, так как позволяет легко прокручивать вперед и назад, а также искать в тексте.
- `more`: Подобно команде `less`, команда `more` также используется для просмотра содержимого файла постранично. Однако `more` менее гибкая по сравнению с `less`.
- `head`: Команда `head` выводит начальные строки текстового файла. По умолчанию выводится 10 первых строк файла, но это количество можно изменить с помощью параметра.
- `tail`: Команда `tail` выводит конечные строки текстового файла. По умолчанию выводится 10 последних строк файла, но это количество также можно изменить с помощью параметра.

6. Приведите основные возможности команды `cp` в Linux.

Команда `cp` в Linux предназначена для копирования файлов и каталогов. Она имеет ряд опций, позволяющих выполнять различные операции с копированием. Вот основные возможности команды `cp`:

- Копирование файлов: С помощью `cp` можно скопировать один или несколько файлов из одного места в другое. Например: `cp file1.txt /путь/к/директории`.
- Копирование каталогов: Команда `cp` также позволяет копировать целые каталоги со всем их содержимым. Для этого используется опция `-r` (рекурсивное копирование).
- Подтверждение при копировании: При копировании файлов в уже существующий каталог с тем же именем, команда `cp` запрашивает подтверждение, если вы не используете опцию перезаписи.
- Переименование файлов: Можно использовать команду `cp` для переименования файлов, скопировав файл с новым именем. Например: `cp file1.txt newfile.txt`.
- Сохранение атрибутов файлов: Опции `cp` позволяют сохранить атрибуты файлов, такие как права доступа, временные метки и владельца файла.
- Рекурсивное копирование: С помощью опции `-r` или `-R` можно скопировать каталог и всё его содержимое, включая подкаталоги.
- Создание резервной копии: Команда `cp` может быть использована для создания резервных копий файлов и каталогов.
- Копирование множества файлов в один каталог: Можно одновременно скопировать несколько файлов в один каталог, указав последним аргументом имя каталога.

7. Приведите основные возможности команды `mv` в Linux.

Команда `mv` в Linux используется для перемещения файлов и каталогов. Она также может быть использована для переименования файлов и каталогов. Вот основные возможности команды `mv`:

- Перемещение файлов: Команда `mv` позволяет переместить один или несколько файлов из одной директории в другую. Например:
`mv file1.txt /путь/к/другой_директории.`
- Перемещение каталогов: Аналогично файлам, команда `mv` может перемещать целые каталоги со всем их содержимым.
- Переименование файлов: Путем использования `mv` можно также переименовывать файлы, перемещая файл с новым именем. Например: `mv oldname.txt newname.txt.`
- Перезапись файлов: Если файл с указанным именем уже существует в целевой директории, команда `mv` перезапишет его без предупреждения.
- Сохранение атрибутов файлов: При перемещении файлов команда `mv` сохраняет атрибуты файлов, такие как права доступа, временные метки и владельца файла.
- Перемещение и переименование: Команда `mv` может одновременно перемещать файлы и переименовывать их, предоставляя гибкость в управлении файлами и директориями.
- Безопасное перемещение множества файлов в один каталог: Можно одновременно переместить несколько файлов в один каталог, указав последним аргументом имя каталога.
- Перемещение с подтверждением: При перемещении файлов в уже существующий каталог с тем же именем, команда `mv` запрашивает подтверждение, если вы не используете опцию перезаписи.

8. Что такое права доступа? Как они могут быть изменены?

Права доступа определяют, какие действия могут выполнять пользователи и программы в отношении файлов и каталогов. В системах Unix-подобных операционных системах (таких как Linux), каждый файл и каталог имеет набор прав доступа, который определяет, кто может читать, записывать или выполнять файл, а также кто может изменять его.

Вот основные типы прав доступа:

- Чтение (Read): Пользователь или программа может просматривать содержимое файла или каталога.
- Запись (Write): Пользователь или программа может изменять файл или каталог, добавлять или удалять содержимое.
- Выполнение (Execute): Для каталогов это позволяет входить в каталог и выполнять команды внутри него. Для файлов это позволяет запускать исполняемые файлы.

Права доступа могут быть изменены с помощью команды `chmod` в терминале Linux. Вот основные способы изменения прав доступа:

- Использование символьной нотации: Например, `chmod u+r file.txt` добавляет право чтения для владельца файла.
- Использование числовой нотации: Назначение прав осуществляется через числовые значения, такие как 755, где каждая цифра представляет набор прав для владельца, группы и других пользователей соответственно.
- Изменение прав доступа по битам: Используя флаги `u`, `g`, `o` и комбинации `+` и `-` можно добавлять или удалять права доступа.

6 Выводы

В данной лабораторной работе мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов, а также приобрели практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

Список литературы

1. Руководство к лабораторной работе №4.