

Лабораторная работа №5. Файловые системы

Цель работы: исследование организации файловой системы, получение практических навыков использования виртуальных файловых систем ОС Linux.

Теоретические сведения

Структура блока данных директории

Информация о содержимом директории хранится в блоке данных, на который указывает соответствующий этой директории inode. Этот блок данных составлен из последовательности записей переменного размера, каждая из которых соответствует файлу в данной директории. Пример такой записи для файла abcdef:

```
0c000000 0e000601 61626364 65660000 00000000 .... abcd ef.. ....
```

Здесь:

- номер inode = 000000c0 = 12
- длина записи = 000e = 14
- длина имени = 06 = 6
- тип файла = 01 (01 - регулярный файл, 02 - директория, и т. д.)
- имя = 616263646566 ("abcdef" в соответствии с кодами символов в таблице ASCII).

Обратите внимание, что на диске числовые значения хранятся в формате little-endian, т. е. первым храниться байт наименее значащий байт числа.

Команды для работы с файловой системой

stat путь/к/файлу - позволяет получить, содержащуюся в файловой системе, информацию о файле.

mount - позволяет просмотреть текущий список точек монтирования файловых систем в операционной системе. Просмотреть список точек монтирования устройств с файловой системой ext4 можно с помощью команды:

```
mount | grep ext4
```

sudo debugfs - позволяет получать отладочную информацию для файловой системы ext4 на указанном устройстве при выполнении некоторой команды. Утилита принимает на вход устройство, на котором располагается отлаживаемая файловая система. По умолчанию утилита работает в интерактивном режиме. Чтобы выполнить одиночный запуск необходимо указать флаг -R "cmd <номер inode>", где номер inode указывает на inode, для которого выполняется команда cmd (угловые скобки обязательны), а вместо cmd указывается команда утилиты debugfs:

- stat - аналогичная программе stat. Позволяет получить информацию о файле с указанным номером inode из таблицы inode.

- **blocks** - позволяет получить номера блоков данные, на которые ссылается данный inode.
- **cat** - выводит содержимое блоков, на которые указывает inode. В некоторых случаях может быть полезно перенаправить вывод утилиты **debugfs** через неименованный канал в команду **hexdump -C**.
- **ls** - выводит содержимое директории с указанным inode в "сыром"-формате.

Пример использования утилиты **debugfs**:

```
$ ls -l tmp/
total 0
-rw-r--r-- 2 user user 0 Dec 5 18:54 a
$ stat tmp | grep Inode | awk '{print $4;}'
321956
$ sudo debugfs -R "ls -l <321956>" /dev/sdb
321956 40755 (2) 1000 1000 4096 7-Dec-2021 18:59 .
44040 40755 (2) 1000 1000 4096 8-Dec-2021 05:58 ..
321964 100644 (1) 1000 1000 0 7-Dec-2021 18:54 a
```

dd - позволяет считывать/записывать содержимое блоков диска в файл. Принимает на вход параметры вида ключ=значение, где ключами могут быть:

- **if** - устройство или файл с которого производится чтение блока данных.
- **of** - устройство или файл на которое производится запись данных.
- **bs** - размер блока данных в байтах.
- **count** - количество передаваемых блоков данных.
- **skip** - смещение в числе блоков данных во входном потоке блоков данных.
- **seek** - смещение в числе блоков данных в выходном потоке блоков данных.

Задания

1. В своем рабочем каталоге создать директорию, указав в качестве ее название собственные инициалы на латинице. В этой директории нужно создать файл с названием **abcdef**, в который следует записать номер inode ранее созданной директории, полученный с помощью команды: **stat . | grep Inode: | awk '{print \$4;}' > abcdef**, выполненной из созданной директории.
2. Перейти обратно в рабочий каталог. Создать здесь жесткую и символическую ссылку на ранее созданный файл **abcdef**, назвав их соответственно **hl** и **sl**.
3. Вывести содержимое файлов **abcdef**, **hl**, **sl**. Удалить файл **abcdef**.
4. Повторно создать файл **abcdef** в ранее созданной директории. Записать в этот файл свою фамилию на латинице. Вывести содержимое файлов **abcdef**, **hl**, **sl**.
5. Получить номер inode, соответствующего ранее созданной директории, а затем номер блока данных, на который указывает этот inode.
6. Записать блок данных в файл **/tmp/dump** используя команду **dd**.

7. Открыть файл /tmp/dump в шестнадцатеричном редакторе, например, в hexedit. Отредактировать его таким образом, чтобы название файла abcdef изменилось на fedcb.
8. Записать измененный блок данных из файла /tmp/dump обратно на диск с помощью команды dd.
Внимание! Смещение в выходном потоке указывается с помощью параметра seek. Ошибочная запись блока данных может повредить файловую систему.
9. Вывести содержимое ранее созданной директории с помощью команды ls -l.
10. В отчет по работе включить результаты выполнения команд.

Ссылки

1. A Minimum Complete Tutorial of Linux ext4 File System: <https://metebalci.com/blog/a-minimum-complete-tutorial-of-linux-ext4-file-system/>
2. Ext4 Disk Layout: https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout
3. Memory Addressing: <https://notes.shichao.io/utlk/ch2/>
4. Introduction to the Linux virtual filesystem (VFS) -- Part I: A high-level tour: <https://www.starlab.io/blog/introduction-to-the-linux-virtual-filesystem-vfs-part-i-a-high-level-tour>
5. The Linux Kernel. Overview of the Linux Virtual File System: <https://www.kernel.org/doc/html/latest/filesystems/vfs.html>