

Изучить работу в Linux, монтируемые файловые системы.

Цель работы:

Целью работы является изучение с монтируемыми файловыми системами ОС Linux и приобретение практических навыков применения команд для анализа файловой системы, управления файлами и процессами.

Краткие теоретические сведения

Устройства внешней памяти современных компьютеров очень разнообразны. К ним относятся различные типы дисков (магнитные, оптические, магнито-оптические), USB флэш-накопители, карты памяти и т.д. Наиболее широко применяются магнитные диски, поэтому в данной работе будет изучаться именно этот класс устройств.

Модель внешней памяти

Физическая модель диска

Информация на диске размещается вдоль концентрических окружностей, называемых *дорожками* (tracks). Дорожки с одинаковыми номерами на различных *поверхностях* диска образуют *цилиндр*, для записи-чтения дорожек на каждой поверхности диска выделяется отдельная магнитная *головка*. Дорожка содержит определенное количество *секторов* – это участок дорожки, хранящий минимальную порцию информации (обычно 512 байтов), которая может быть считана или записана за одно обращение к диску. Дорожки нумеруются в пределах поверхности, а сектора – в пределах дорожки.

Постоянство размера сектора при изменении номера дорожки обеспечивается изменением плотности записи, максимальная плотность записи используется на дорожках с минимальным радиусом. Для получения доступа к информации, хранящейся на диске, необходимо указать *физический адрес*, который включает номер цилиндра, номер головки и номер сектора (*CHS*).

Таким образом, физическая модель диска включает следующие понятия: дорожка, цилиндр, поверхность и сектор. Все диски отличаются друг от друга по набору этих параметров. Физическая модель диска используется его контроллером.

Синонимом термина «физический диск» является термин «физический том».

Логическая модель диска

С логической точки зрения все адресное пространство диска представляет собой набор последовательно пронумерованных секторов. Небольшая часть этих секторов выделяется для хранения служебной информации и называется системной областью диска (областью метаданных), остальные сектора предназначены для хранения файлов и каталогов и образуют область данных. С логической моделью диска работает операционная система.

Единица размещения дисковой памяти называется блоком (в ОС Windows – кластером), который состоит из одного или нескольких секторов. Например, в ОС Linux обычно размер блока по умолчанию равен 1Кбайт, то есть блок включает два сектора.

Все линейное дисковое пространство обычно делится на несколько частей – разделов (partitions). В один раздел объединяется группа смежных цилиндров, для каждого раздела следует хранить информацию о его начале и конце, т.е. номера первого и последнего из задействованных в нем цилиндров. Разделение всего дискового пространства на разделы полезно по нескольким причинам:

- уменьшаются «дальние перемещения» головок чтения/записи и увеличивается скорость выполнения операций чтения и записи;

- появляется возможность структурирования данных (например, можно отделить файлы пользователя от файлов ОС);
- на одном диске можно установить несколько операционных систем;
- на одном диске можно хранить информацию в разных файловых системах, или в одинаковых файловых системах, но с разным размером кластера.

Структура данных, хранящая информацию о логической организации диска вместе с небольшой программой, с помощью которой можно найти и загрузить в ОЗУ программу загрузки ОС, называется главной загрузочной записью (Master Boot Record, MBR), и располагается в самом первом секторе, т.е. в секторе с физическим адресом **0-0-1**.

В MBR расположена также таблица разбиения диска на разделы (*partition table*), которая содержит четыре записи по 16 байтов. Каждая запись может хранить информацию по одному разделу, который называется первичным. Таким образом число первичных разделов на диске ограничено, их не может быть более четырех. По мере увеличения объемов дисковой памяти стало ясно, что четырех разделов мало, и были предложены логические разделы. Для этого один из первичных разделов объявляется расширенными, в нем создаются логические разделы.

Расширенные разделы сами по себе не используются, они могут лишь хранить информацию о логических разделах. Первый сектор расширенного раздела хранит таблицу разделов с четырьмя записями: одна (первая) используется для логического раздела, другая (вторая) для еще одного расширенного раздела, а две другие не используются. Таким образом, каждый расширенный раздел имеет свою таблицу разбиения, в которой используются только две записи, задающие один логический и один расширенный, то есть получается цепочка из таблиц разделов. В последнем расширенном разделе из четырех записей таблицы разбиения используется только первая. В семействе ОС Unix диски и их разделы представляются пользователю в виде файлов устройств, зарегистрированных в каталоге /dev. Имена этих файлов формируются по определенным правилам. Так, до недавнего времени наиболее распространенные IDE – диски с интерфейсом Parallel ATA, PATA именовались **hda, hdb** и т.д. Современные диски с интерфейсом Serial ATA (SATA) предстают перед системой как SCSI – диски и именовются **sda, sdb** и т.д.

Дисковые разделы идентифицируются порядковыми номерами. Цифры с 1 по 4 отведены под первичные разделы, а логические разделы нумеруются, начиная с цифры 5.

Пример. Если на SATA– диске имеются два первичных раздела, второй из которых определен как расширенный и поделен на три логических раздела, то соответствующие им устройства будут именоваться так:

/dev/sda1 – первичный раздел;

/dev/sda2 – первичный раздел, определенный в качестве расширенного;

/dev/sda5, /dev/sda6, /dev/sda7 – логические разделы.

Дисковые разделы могут создаваться не только для разделения, но и для объединения, в результате которого физические диски или их отдельные части предстают перед системой в качестве непрерывного дискового пространства, на котором может быть создана единая файловая система. Существует два основных способа объединения – организация дисковых массивов и применение менеджера логических томов.

Технология дисковых массивов (RAID) используется для объединения нескольких дисков в один логический объект для избыточности и повышения производительности. Существует более 10 различных спецификаций RAID. Например, RAID уровня 1 предназначен для увеличения надежности хранения данных и реализуется путем зеркалирования (дублирования) дисков.

Менеджер логических томов (LVM) – компонент Linux, реализованный с помощью подсистемы device mapper (dm), позволяющий использовать разные области одного жёсткого диска и области различных жёстких дисков как один логический том. LVM позволяет:

- иметь файловую систему, которая превышает размер наибольшего диска;
- добавлять диски или разделы в дисковую группу и расширять существующие файловые системы «на лету»;

- заменить два жестких диска размером 100 ГБ одним диском на 200 ГБ без необходимости выключения компьютера для переноса системы или ручного перемещения данных между дисками
- уменьшить размеры файловых систем и удалить диски из дисковой группы, когда их емкость больше не требуется
- создавать резервные копии на основе мгновенных копий файловой системы.

В LVM используются физические тома (physical volume – PV), группы томов (volume groups – VG) и логические тома (logical volume – LV). *Физические тома* являются физическими дисками или разделами физических дисков, *группы томов* объединяют физические тома. *Группа томов* может быть логически разделена на логические тома, которые для пользователей представляются физическими дисковыми разделами.

Файлы блочных логических томов хранятся в каталоге /dev/mapper.

Каждый раздел дисковой памяти, включая логические тома, имеет собственную файловую систему и характеризуется следующими основными параметрами:

- имя, используемое ядром Linux, например dm-1;
- тип (первичный, расширенный, логический);
- тип установленной файловой системы;
- точка монтирования – имя каталога, к которому подключается раздел;
- имя устройства – имя, под которым раздел представлен пользователю, например centos-tmp;
- номер устройства, состоящий из двух частей: старший номер (major) идентифицирует драйвер, ассоциированный с устройством, а младший номер (minor) указывает номер устройства. Например, диск sda может идентифицироваться парой номеров 8:0, а разделы sda1 и sda2 – номерами 8:1 и 8:2 соответственно;
- тип устройства, например Virtual CD-ROM.

Файловая система SYSTEM V

Файловая система SYSTEM V (s5) была одной из первых файловых систем, используемых в семействе ОС Unix. Система управляет всеми блоками раздела, разделяя его на две области – область метаданных и область данных. В области метаданных расположены три объекта – загрузочный блок, суперблок и индексные дескрипторы.

Загрузочный блок используется для загрузки ОС и занимает первый блок файловой системы в каждом разделе. Однако активным является только один загрузочный блок, находящийся в корневой файловой системе.

Суперблок располагается непосредственно за загрузочным блоком и содержит самую общую информацию о файловой системе (общий размер, размер области индексных дескрипторов, их число, список свободных блоков и индексных дескрипторов и т. д.). При монтировании файловой системы в оперативной памяти создается копия ее суперблока. Все последующие операции по созданию и удалению файлов влекут изменения копии суперблока в оперативной памяти, эта копия периодически записывается на магнитный диск. Часто причиной повреждения файловой системы является отключение электропитания или зависание ОС в тот момент, когда система производит копирование суперблока из оперативной памяти на магнитный диск.

Область индексных дескрипторов содержит дескрипторы файлов (i - узлы), в англоязычной литературе называемые inode. С каждым файлом связан один i - узел, но одному i - узлу могут соответствовать несколько файлов. В i - узле размером 64 байта хранится вся информация о файле, кроме его имени. Область индексных дескрипторов имеет фиксированный формат и располагается непосредственно за суперблоком. Общее число дескрипторов i, следовательно, максимальное число файлов раздела, задается в момент создания файловой системы.

Дескрипторы нумеруются натуральными числами. Первый дескриптор используется ОС для описания специального файла, содержащего информацию о сбойных блоках раздела, которые рассматриваются ОС, как принадлежащие специальному файлу, и поэтому считаются занятыми. Второй дескриптор содержит описание корневого каталога файловой системы.

В области данных расположены как обычные файлы, так и файлы каталогов (в том числе корневой каталог). Специальные файлы представлены в файловой системе только записями в соответствующих каталогах и индексными дескрипторами специального формата, т. е. места в области памяти не занимают.

Повышение быстродействия файловых операций осуществлялось дроблением файловых систем на относительно независимые части, получившие название групп цилиндров (файловая система ufs в UNIX BSD) или групп блоков (файловая система ext2 в Linux). В результате этого появилась возможность размещения логически связанной информации в физически смежных областях диска, что минимизировало перемещение дисковых головок, способствуя тем самым скорости доступа к данным.

Расчленение файловых систем также способствовало росту их надежности, т.к. давало возможность в разных ее частях дублировать критически важную информацию – суперблок, утрата которого (например, из-за физического повреждения дисковой поверхности) ранее влекла за собой невозможность доступа к данным вообще.

Файловые системы ОС Linux

Файловые системы ОС Linux делятся на два типа – локальные и распределённые (сетевые). Локальные файловые системы могут располагаться во внешней памяти (ext2, ext3, ext4 и др.) или в оперативной памяти (псевдо-файловые системы). К последней группе относятся:

- /proc – используется в качестве интерфейса к структурам данных в ядре; большинство расположенных в ней файлов доступны только для чтения, но в некоторые файлы можно записывать данные, что позволяет изменить переменные ядра;
- /tmpfs – позволяет не записывать на физические диски временные файлы, которые формируются в оперативной памяти, а затем удаляются; поддерживает работу с виртуальной памятью;
- /devfs – предназначена для управления устройствами;
- /sysfs – используется для получения информации о всех устройствах и драйверах.

Распределенные файловые системы предназначены для объединения на логическом уровне файловых систем отдельных компьютеров в единое целое. В Linux такой системой является nfs (Network File System).

Основной файловой системой ОС Linux на начальном этапе была система ext2. Она позволяла хранить длинные имена файлов (до 255 символов) и обеспечить высокую производительность. В первом блоке файловой системы ext2 располагается загрузчик, а все остальное пространство делится на блоки равного размера (обычно 1 Кбайт). Блоки объединяются в группы, каждая из которых имеет суперблок, описатель группы, два битовых массива (для блоков и для i-узлов) и блоки для хранения данных.

Суперблок хранит информацию о количестве i-узлов и блоков данных в группе, о размере группы и т.д. Описатель группы содержит информацию о расположении битовых массивов, количестве свободных блоков и i-узлов в группе, а также о количестве каталогов в группе. Битовые массивы предназначены для учета свободных блоков и i-узлов, для хранения каждого массива выделяется один блок. При размере блока 1Кбайт размер группы равен 8192 блока и количество i-узлов равно 8192.

Далее находятся i-узлы, структура которых показана в таблице структуры. Размер каждого узла составляет 128 байт.

Таблица структуры

Режим	Счетчик связей	UID	GID	Раз-мер	Метки времени	Адреса блоков	Однократный косвенный блок	Двукратный косвенный блок	Трехкратный косвенный блок
-------	----------------	-----	-----	---------	---------------	---------------	----------------------------	---------------------------	----------------------------

Здесь обозначено:

- режим – тип файла, биты защиты;

- счетчик связей – число записей каталогов, указывающих на этот i-узел;
- UID – идентификатор владельца файла;
- GID – идентификатор группы владельца;
- размер – размер файла в байтах;
- метки времени – времена последнего доступа и последнего изменения файла, а также время последнего изменения i-узла;
- адреса блоков – адреса первых 12 блоков файла, размер адреса – 4 байта;
- однократный косвенный блок – адрес одинарного косвенного блока, который содержит адреса 256 дополнительных блоков файла;
- двукратный косвенный блок – содержит адреса 256 одинарных косвенных блоков, каждый из которых содержит адреса 256 блоков данных;
- трехкратный косвенный блок – содержит адреса 256 двукратных косвенных блоков.

Часть информации из i-узла можно посмотреть командой

stat имя_файла

Достоинством файловой системы ext2 является ее быстродействие, а недостатком – отсутствие средств журналирования, что ухудшает надежность хранения данных. Этот недостаток исправлен в современных файловых системах ext3 и ext4. Кроме того, в ext4 адресация блоков увеличена до 6 байтов, а размер i-узлов – до 256 байтов, что позволило увеличить максимальный размер одного файла до 16 Тбайт.

Файловая система ext2 до сих пор активно используется на твердотельных накопителях, например на SSD-дисках, где ячейки памяти необратимо изнашиваются после определённого количества операций записи. Отсутствие журналирования дает возможность значительно уменьшить количество перезаписей одного и того же сектора диска и продлить срок службы устройства.

Команды анализа файловой системы

Команда df

Назначение: показывает список файловых систем на смонтированных устройствах, их размер, занятое и свободное пространство, а также точки монтирования.

Формат: df [-опции][имя_файла]

Примеры:

а) df -T – вывод списка файловых систем с указанием их типа;

б) df ~ – вывод информации по файловой системе, в которой находится домашний каталог.

Комментарий: для того, чтобы вместо информации об использовании блоков выводить информацию об использовании индексных дескрипторов, надо применить опцию -i.

Команда du

Назначение: показывает использование дисковой памяти файлами.

Формат: du [-опции][имя_файла]

Примеры:

а) du – вывод размера всех подкаталогов текущего каталога;

б) du -ah – вывод размера (в байтах) всех файлов и подкаталогов текущего каталога;

Комментарий: размер файлов по умолчанию выводится в блоках, для вывода в байтах надо применить опцию -h.

Команда lsblk

Назначение: показывает список блочных устройств.

Формат: lsblk [-опции][имя_устройства]

Примеры:

а) `lsblk` – вывод списка активных устройств;

б) `lsblk -f` – вывод списка активных устройств с указанием типа файловой системы;

в) `lsblk -a` – вывод списка всех устройств;

г) `lsblk -p` – вывод списка активных устройств с указанием путей доступа

д) `lsblk -o KNAME,FSTYPE,SIZE,MODEL,MOUNTPOINT`

Комментарий: опция `-o` дает возможность самостоятельного формирования списка вывода, например, при задании параметров `KNAME, FSTYPE, SIZE, MODEL, MOUNTPOINT` будут выводиться имя устройства, тип его файловой системы, размер, модель и точка монтирования.

Команда `cat /proc/partitions`

Назначение: вывод файла, который содержит общую информацию о разделах файловой системы.

Формат: `cat /proc/partitions`

Комментарий: по каждому разделу выводится имя, размер и номер устройства, на котором расположен раздел.

Команда `cat /etc/fstab`

Назначение: вывод файла, который содержит общую информацию об используемых файловых системах.

Формат: `cat /etc/fstab`

Комментарий: выводится список смонтированных файловых систем.

Команда `ls /dev/mapper`

Назначение: отображение каталога, в котором содержатся файлы блочных логических томов.

Формат: `ls /dev/mapper`

Комментарий: логические тома используются при установке в Linux системы LVM, предназначенной для создания абстрактной логической внешней памяти вместо физического управления дисками.

2.5. Команды управления процессами

Для выполнения любой программы ОС создает процесс, имеющий уникальный идентификатор (`process ID, PID`). Процессы могут запускаться как с терминала пользователя, так и из другого (родительского) процесса. Пользователь Linux всегда может получать информацию о работающих процессах и управлять их выполнением.

Команда `ps`

Назначение: вывод информации о процессах.

Формат: `ps [опции]`

Примеры:

а) `ps` – вывод краткой информации о процессах, запущенных пользователем;

б) `ps -ix` – вывод подробной информации о процессах, запущенных пользователем;

в) `ps -e` – вывод краткой информации о всех процессах, запущенных в системе;

а) `ps -ef` – вывод подробной информации о всех процессах, запущенных в системе;

д) `ps -a` – вывод краткой информации о всех процессах, запущенных пользователями;

Комментарий: вывод команды содержит информацию, представленную в таблице 2.

Таблица 2

Название столбцов	Значение
USER	Идентификатор (имя) пользователя, запустившего процесс
PID	Идентификатор процесса
%CPU	Интенсивность использования процессора
%MEM	Интенсивность использования памяти
STAT	Состояние процесса
STIME	Время старта процесса
TTY	Терминал, с которого был запущен процесс
TIME	Использованное время процессора
COMAND	Выполняемая команда

Состояние процесса обозначается следующим образом:

R – процесс в режиме выполнения; *D* – процесс находится в ожидании дисковой операции; *I* – процесс в режиме ожидания более 20 секунд; *S* – процесс в режиме ожидания менее 20 секунд; *T* – процесс остановлен; *Z* – мертвый (зомби) процесс.

Команда kill

Назначение: формирование сигнала для уничтожения процесса.

Формат: `kill [опции] PID`

Примеры:

а) `kill -l` – вывод списка всех доступных сигналов;

б) `kill -9 521` – принудительно завершить процесс с идентификатором 521.

Комментарий: команда используется в случае некорректной работы или зависания процесса для формирования сигнала уничтожения процесса с указанным идентификатором PID. Наиболее часто используются сигналы `SIGTERM` (15), который генерируется по умолчанию, и `SIGKILL` (9), который используется в случае неудачи с сигналом `SIGTERM`.

Для изучения команды `kill` рекомендуется написать простой сценарий, реализующий бесконечный цикл:

```
while true
do
true
done
```

Порядок выполнения, ход работы

1. Осуществите вход в систему, используя соответствующее имя пользователя.
2. Создайте в домашнем каталоге нижеперечисленные объекты файловой системы, где `kk` – номер по списку в журнале Вашей группы, и задайте им указанные права доступа:

```
drwxr--r-- ... australia_kk
```

```
drwx--x--x ... play_kk
```

```
-r-xr--r-- ... my_os_kk
```

```
-rw-rw-r-- ... feathers_kk
```

3. Прodelайте приведенные ниже упражнения, записывая в отчет используемые при этом команды:

3.1. Просмотрите содержимое файла /etc/passwd с использованием команды страничного просмотра. Сколько пользователей имеют учетные записи на устройстве? Приведите в отчете структуру учетной записи.

3.2. Скопируйте файл ./feathers_kk в файл ./file.old

3.3. Переместите файл ./file.old в каталог ./play_kk

3.4. Скопируйте каталог ./play_kk в каталог ./australia_kk .

3.5. Лишите владельца файла ./feathers_kk права на чтение.

3.6. Попытайтесь просмотреть файл ./feathers_kk командой **cat**.

3.7. Попытайтесь скопировать файл ./feathers_kk.

3.8. Дайте владельцу файла ./feathers_kk право на чтение и выполните п. 3.5; 3.6

3.9. Лишите владельца каталога ./play_kk права на выполнение.

3.10. Перейдите в каталог ./play_kk. Посмотрите что произошло.

3.11. Дайте владельцу каталога ./play_kk право на выполнение и выполните п. 3.10.

4. Создайте в файле loop сценарий, реализующий бесконечный цикл (см. п. Команда kill) и запустите его в фоновом режиме:

5. С помощью команды **ps** посмотрите список Ваших активных процессов и общий список процессов, запущенных в системе. Занесите списки в отчет и проведите их анализ.

6. Прервите выполнение сценария из п. 4.

7. Посмотрите с помощью команды **stat** и занесите в отчет информацию из индексного дескриптора файла ~/.bash_history.

8. С помощью команд **lsblk** и **df** определите основные характеристики разделов внешней памяти устройства (имя и номер устройства, имя и тип раздела, размер, тип файловой системы, коэффициент использования памяти).

Результаты представьте в виде следующей таблицы:

№ п/п	Имя устройства	Имя раздела	Тип раздела	Размер раздела (Гб)	Тип ФС	Номер устройства	Кэф-т использования
	centos-tmp	dm-2	lvm	9,8	ext4	253:2	2%

9. С помощью команд **df** и **du** определите типы файловых систем, используемых на устройстве, а также в каком из имеющихся разделов расположен ваш домашний каталог и размер домашнего каталога. Поясните назначение каждой из файловых систем.

10. Посмотрите и занесите в отчет содержимое файлов /proc/partitions и /etc/fstab, сопоставьте их с результатами, полученными в п. 8 и п. 9.

Отчет по практической работе должен содержать:

основные определения, рассуждения по выполнению заданий, необходимые изображения (фото экрана с выполняемыми действиями) по ходу работы, вывод по работе, ответы на контрольные вопросы.

Контрольные вопросы:

1. Дайте характеристику физической модели диска.
2. Дайте характеристику логической модели диска.
3. Дайте характеристику каждой файловой системе, существующей в компьютере, на котором Вы выполняли лабораторную работу.

4. Приведите общую структуру файловой системы ext2 и дайте характеристику каждому элементу этой структуры.
5. Опишите действия системы при обращении к некоторому файлу с запросом на чтение.
6. Как определить объем свободной памяти на жестком диске и объем вашего домашнего каталога?
7. Каким образом вы можете получить информацию о процессах в системе?
8. Как удалить «зависший» процесс?
9. Что такое MBR?
10. Что такое первичный раздел? Какое максимальное число первичных разделов может быть на диске? Каким образом нумеруются первичные разделы?
11. Что такое расширенный и логический разделы, как они нумеруются?
12. Будет ли информация о расширенном разделе в распечатке после выполнения команды **df** ? Если нет, то почему?
13. Поясните принцип работы LVM.