**Конспект лекции ЛК5 Устройства. Файловые системы**

Что такое файловая система

Обычно вся информация записывается, хранится и обрабатывается на различных цифровых носителях в виде файлов. Далее, в зависимости от типа файла, кодируется в виде знакомых расширений – \*exe, \*doc, \*pdf и т.д., происходит их открытие и обработка в соответствующем программном обеспечении. Мало кто задумывается, каким образом происходит хранение и обработка цифрового массива в целом на соответствующем носителе.

Операционная система воспринимает физический диск хранения информации как набор кластеров размером 512 байт и больше. Драйверы файловой системы организуют кластеры в файлы и каталоги, которые также являются файлами, содержащими список других файлов в этом каталоге. Эти же драйверы отслеживают, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.

Запись файлов большого объема приводит к необходимости фрагментации, когда файлы не сохраняются как целые единицы, а делятся на фрагменты. Каждый фрагмент записывается в отдельные кластеры, состоящие из ячеек (размер ячейки составляет один байт). Информация о всех фрагментах, как части одного файла, хранится в файловой системе.

Файловая система связывает носитель информации (хранилище) с прикладным программным обеспечением, организуя доступ к конкретным файлам при помощи функционала взаимодействия программ API. Программа, при обращении к файлу, располагает данными только о его имени, размере и атрибутах. Всю остальную информацию, касающуюся типа носителя, на котором записан файл, и структуры хранения данных, она получает от драйвера файловой системы.

На физическом уровне драйверы ФС оптимизируют запись и считывание отдельных частей файлов для ускоренной обработки запросов, фрагментации и «склеивания» хранящейся в ячейках информации. Данный алгоритм получил распространение в большинстве популярных файловых систем на концептуальном уровне в виде иерархической структуры представления метаданных (B-trees).

Технология снижает количество самых длительных дисковых операций – позиционирования головок при чтении произвольных блоков. Это позволяет не только ускорить обработку запросов, но и продлить срок службы HDD. В случае с твердотельными накопителями, где принцип записи, хранения и считывания информации отличается от применяемого в жестких дисках, ситуация с выбором оптимальной файловой системы имеет свои нюансы.

**Основные функции файловых систем**

Файловая система отвечает за оптимальное логическое распределение информационных данных на конкретном физическом носителе. Драйвер ФС организует взаимодействие между хранилищем, операционной системой и прикладным программным обеспечением. Правильный выбор файловой системы для конкретных пользовательских задач влияет на скорость обработки данных, принципы распределения и другие функциональные возможности, необходимые для стабильной работы любых компьютерных систем. Иными словами, это совокупность условий и правил, определяющих способ организации файлов на носителях информации.

Основными функциями файловой системы являются:

* размещение и упорядочивание на носителе данных в виде файлов;
* определение максимально поддерживаемого объема данных на носителе информации;
* создание, чтение и удаление файлов;
* назначение и изменение атрибутов файлов (размер, время создания и изменения, владелец и создатель файла, доступен только для чтения, скрытый файл, временный файл, архивный, исполняемый, максимальная длина имени файла и т.п.);
* определение структуры файла;
* поиск файлов;
* организация каталогов для логической организации файлов;
* защита файлов при системном сбое;
* защита файлов от несанкционированного доступа и изменения их содержимого.

**Задачи файловой системы**

Функционал файловой системы нацелен на решение следующих задач:

* присвоение имен файлам;
* программный интерфейс работы с файлами для приложений;
* отображение логической модели файловой системы на физическую организацию хранилища данных;
* поддержка устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;
* содержание параметров файла, необходимых для правильного взаимодействия с другими объектами системы (ядро, приложения и пр.).

В многопользовательских системах реализуется задача защиты файлов от несанкционированного доступа, обеспечение совместной работы. При открытии файла одним из пользователей для других этот же файл временно будет доступен в режиме «только чтение».

Вся информация о файлах хранится в особых областях раздела (томах). Структура справочников зависит от типа файловой системы. Справочник файлов позволяет ассоциировать числовые идентификаторы уникальных файлов и дополнительную информацию о них с непосредственным содержимым файла, хранящимся в другой области раздела.

**Операционные системы и типы файловых систем**

Существует три основных вида операционных систем, используемых для управления любыми информационными устройствами: Windows компании Microsoft, macOS разработки Apple и операционные системы с открытым исходным кодом на базе Linux. Все они, для взаимодействия с физическими носителями, используют различные типы файловых систем, многие из которых дружат только со «своей» операционкой. В большинстве случаев они являются предустановленными, рядовые пользователи редко создают новые дисковые разделы и еще реже задумываются об их настройках.

В случае с Windows все выглядит достаточно просто: NTFS на всех дисковых разделах и FAT32 (или NTFS) на флешках. Если установлен NAS (сервер для хранения данных на файловом уровне), и в нем используется какая-то другая файловая система, то практически никто не обращает на это внимания. К нему просто подключаются по сети и качают файлы.

На мобильных гаджетах с ОС Android чаще всего установлена ФС версии ext4 во внутренней памяти и FAT32 на карточках microSD. Владельцы продукции Apple зачастую вообще не имеют представления, какая файловая система используется на их устройствах – HFS+, HFSX, APFS, WTFS или другая. Для них существуют лишь красивые значки папок и файлов в графическом интерфейсе.

Более богатый выбор у линуксоидов. Но здесь настройка и использование определенного типа файловой системы требует хотя бы минимальных навыков программирования. Тем более, мало кто задумывается, можно ли использовать в определенной ОС «неродную» файловую систему. И зачем вообще это нужно.

Рассмотрим более подробно виды файловых систем в зависимости от их предпочтительного использования с определенной операционной системой.

**Уровни планирования**

**Планирование на верхнем уровне (high-level scheduling)** —определение заданий, которые могут быть допущены в систему активного участия в соревновании за системные ресурсы.

Планирование на верхнем уровне

• Используется в крупных мэйнфреймах, предназначенных для пакетной обработки данных

• Определяет степень многозадачности

**Степень многозадачности (degree of multiprogramming)** — общее количество процессов в системе в конкретный момент времени.

**Планирование на промежуточном уровне (intermediate-level scheduling)** — на этом уровне планирования определяется, какие процессы будут допущены к планировщику на нижнем уровне

**Планирование на промежуточном уровне**

• В большинстве современных систем планировщик верхнего уровня отсутствует, и запуск заданий осуществляется планировщиком промежуточного уровня

• В роли планировщика промежуточного уровня выступает менеджер памяти

**Планирование на нижнем уровне (low-level scheduling)** —определение процесса, который должен получить управление процессором. Осуществляется планировщиком процессов.

**Планирование с приостановкой процессов**

**Активный процесс (active process)** — это процесс, находящийся в оперативной памяти и имеющий возможность вести борьбу за процессорное время. Он может находиться в одном из активных состояний (выполняется, готов, блокирован).

**Приостановленный процесс (suspended process)** — это процесс, находящийся на диске, на некоторое время выбывший из борьбы за процессорное время, но не прекративший своего существования.

**Приостановка процессов (suspend)** — операция, осуществляемая менеджером памяти. Он выгружает процессы на диск, если система перегружена либо их в памяти слишком много и все они в ней не помещаются.

**Активация процессов (activate)** — операция, осуществляемая менеджером памяти. Он периодически просматривает процессы, находящиеся на диске, чтобы решить, какой из них переместить в память .

**Планирование с приоритетным вытеснением**

**Планирование без приоритетного вытеснения (nonpreemptive scheduling)** — политика планирования, которая не позволяет системе отбирать процессор у процесса до тех пор, пока сам процесс не отдаст его добровольно либо не закончит работу. Используется, например, в системах пакетной обработки данных.

**Планирование с приоритетным вытеснением (preemptive scheduling)** — политика планирования, позволяющая отбирать процессор у процесса, и основывающаяся на приоритетах процессов. Используется, например, в системах реального времени и интерактивных системах разделения времени.

**Приоритет (priority)** — мера важности процесса (потока), используемая для определения порядка и продолжительности его выполнения.

**Статические приоритеты (static priorities)** — не изменяющиеся во времени приоритеты. Они присваиваются процессам до начала их выполнения и остаются постоянными.

**Динамические приоритеты (dynamic priorities)** — изменяющиеся во время выполнения процессов приоритеты. Их значение может меняться в зависимости от изменения ситуации.

**Сигнал в операционных системах семейства Unix**

— асинхронное уведомление процесса о каком-либо событии, один из основных способов взаимодействия между процессами. Когда сигнал послан процессу, операционная система прерывает выполнение процесса, при этом, если процесс установил собственный обработчик сигнала, операционная система запускает этот обработчик, передав ему информацию о сигнале, если процесс не установил обработчика, то выполняется обработчик по умолчанию.

Названия сигналов «SIG…» являются числовыми константами (макроопределениями Си) со значениями, определяемыми в заголовочном файле signal.h. Числовые значения сигналов могут меняться от системы к системе, хотя основная их часть имеет в разных системах одни и те же значения. Утилита kill позволяет задавать сигнал как числом, так и символьным обозначением.

Спецификация сигналов включена в стандарты POSIX.

**Прерывание** — одна из базовых концепций вычислительной техники, которая заключается в том, что при наступлении какого-либо события происходит передача управления специальной процедуре, называемой обработчиком прерываний (ISR, англ. Interrupt Service Routine). В отличие от условных и безусловных переходов, прерывание может быть вызвано в любом месте программы, в том числе если выполнение программы приостановлено, и обусловлено обычно внешними по отношению к программе событиями. После выполнения необходимых действий, обработчик прерываний как правило возвращает управление прерванной программе.