Лекции по ОС

24.11.17

Физическая память

<…> То есть осуществить отображение пространства имен, на физическую память компьютера. Отображение происходит в 2 этапа: сначала системой программирования, затем отображение осуществляется ОС. Между этими этапами обращения к памяти принимает форму виртуального адреса.

Возможно 3 вида ситуации:

1. Объем виртуального адресного пространства меньше объема физической памяти.
2. Объем виртуального адресного пространства равен объему физической памяти.
3. Объем виртуальной памяти гораздо больше объема физической памяти.

Всю память необходимо распределить между программами. Бывает 2 типа распределения:

1. Простое, непрерывное распределение, при котором вся память разделена на 3 области:
   1. Область занимаемая ОС.
   2. Область в которой размещаются исполняемые задачи.
   3. Свободная область памяти.

Остальные модули ОС не относящийся к его ядру загружаются в ОЗУ по мере необходимости, и называются транзитными.

1. Распределения с перекрытием.

Если нужно создать программу, логическое адресное пространство которой должно быть больше чем свободная область памяти, или больше чем весь объем памяти, используется распределение с перекрытием. Этот метод предполагает, что вся программа разбивается на сегменты, и в ОЗУ располагаются только выполняющиеся сегменты. Память может выделяться программе одним сплошным участком (это метод неразрывного распределения памяти), или несколькими порциями, которые могут быть размещены в разных областях памяти (это разрывное распределение памяти).

Разбиение всего объема ОЗУ на нескольких разделах может осуществляться единовременно, или по мере необходимости, оператором системы. Первые мультипрограммные ОС строились по этой схеме. Увеличить число параллельно выполняющихся приложений можно за счета свопинга. То есть эта задача на время выгружается на магнитный диск, а ее место занимает или задача стоящая на очереди, готовая к выполнению, или задача находящийся в более привилегированном положении. Недостатком данного способа является фрагментация памяти. Специальный планировщик ведет список адресов свободной ОЗУ, при появлении новой задачи, он выделяет для нее раздел, объем которой равен необходимому или чуть больше, после чего список свободных областей модифицируется.

Способы выделения памяти под новый раздел:

1. Первый подходящий участок.
2. Самый подходящий участок.
3. Самый неподходящий участок. При этом методе наблюдается сильная фрагментация памяти, однако можно проводить ее уплотнение. Данный способ применялся достаточно длинное время.

Главные способы организации памяти:

1. Страничная.
2. Сегментная.
3. Сегментно-страничная.

Все эти способы являются разрывными, и вся программа разбивается на логические элементы – сегменты, и каждому сегменту выделяется физическая память. Вся информация о размещении сегментов сводится в таблицу сегментов, которая по-другому называется **дескрипторной таблицей**. Она состоит из 2 частей: номер сегмента, и смещения относительно начала сегмента. Если сегмент задачи находится в ОЗУ, то об этом делается пометка в дескрипторе. В поле адреса записывается адрес, а в поле длины сегмента указывается количество адресуемых ему ячеек. Там же в дискрипторе содержится данные о типе сегмента, правах доступа к нему, отметка об обращении к данному сегменту. В ОЗУ размещаяют не все сегменты, а только задействованные в данный момент. Перемещение сегментов из ОЗУ на ЗУ осуществляется по следующим алгоритмам замещения:

1. FIFO – первым пришел, первым вышел.
2. LRU – дольше других используемый.
3. LFU – реже других используемый.
4. Случайный выбор сегмента.

Процедура перемещения называется свопингом, преимущество сегментного метода в том, что можно размещать программу в памяти не целиком, а по мере необходимости, и второе преимущество в том, что программные модули могут быть разделяемые. Недостатки в том, что для доступа искомой ячейки приходится трать много времени, второй недостаток – фрагментация, третий – много памяти и процессорного времени теряется на размещении и обработку дескрипторных таблиц. Второй способ - страничный способ. ОЗУ разбивается на физические страницы, а программа на виртуальные, и их размер не совпадает. Часть виртуальных страниц, размещается в ОЗУ, часть в внешней, которой в данном случае служит расширение оперативной. Для каждой задачи, имеется таблица страниц, для трансляции адресных пространств. Для описания каждой страницы, менеджер памяти заводит соответствующий дескриптор. По номеру виртуальной страницы таблица дескриптора страниц находится соответствующий элемент. Если би-присутствие (?) равен 1, значит данная страница находится в ОЗУ, значит в дескрипторе находится физический адрес страницы, если он равен 0, то он расположен во внешней памяти. Плюсы метода — это минимальная фрагментация, а недостатки — это страничная трансляция виртуальной памяти требует существенных накладных расходов, а второй недостаток это то, что программа разделяется на страницы случайно, без учет логических взаимосвязей.

27.11.17

Организация файловой системы

Системы управления данными во внешней памяти появились при появлении магнитных лентах, но тот облик, который есть сейчас мы будем изучать с появлением магнитных дисков. На первых парах, каждая прикладная программа, самостоятельно управляла своими данными. После этого произошел переход к использованию централизованных систем управления файлами. Система управления файлами распределяет внешнюю память, отображает имена файлов, адреса внешней памяти и обеспечивает доступ к данным.

**Файловая система** – это часть ОС, назначение которой в организации эффективной работы с данными, хранящимися во внешней памяти, и обеспечении пользователя удобным интерфейсом, для доступа к ним.

Для того что бы избавить обычного пользователя от сложностей взаимодействия с аппаратурой, особенности работы с регистрами контроллера диска, была придумана абстрактная модель файловой системы.

ОС делят память на блоки фиксированного размера. Файл, который обычно представляет собой не структурированную последовательность однобайтовых записей хоронится в виде последовательности блоков, не обязательно смежных. Каждый блок хранит целое число записей. Например, в ОС семейства UNIX адреса блоков данных файлов хранятся в индексном узле – отдельном блоке внешней памяти. Такой способ организации называется индексацией. Индекс файла состоит из списка элементов, каждый из которых содержит номер блока в файле, и сведения о месте нахождения данного блока. Считывание очередного файла осуществляется с текущей позиции, которой характеризуется смещение от начала файла. Зная размер блока легко вычислить номер блока, содержащего текущую позицию. Адрес нужного блока диска можно затем извлечь из индекса файла. Базовой операцией, выполняемой по отношению к файлу, является чтение блока с диска, и перенос его в буфер находящийся в основной памяти. Файловая система позволяет при помощи каталогов и директорий связать уникальное имя файла с блоками вторичной памяти содержащими данные файлов. Вся файловая система компьютера представляет собой большой индексированный файл, и помимо самих файлов и структур данных с помощью которых организованно управление файлами файловая система включает также программные средства, реализующие различные операции над файлами.

Основные функции файловой системы:

1. Идентификация файлов – это связывание имени файла с выделенным ему пространством внешней памяти.
2. Распределение внешней памяти между файлами – для работы с конкретным файлом обычному пользователю не требуется иметь информацию о физическом местонахождении этого файла, что бы загрузить этот документ в MS Word.
3. Обеспечение надежности и отказоустойчивости. Т.к. информация, хранящийся на ПК может быть дороже самого ПК.
4. Обеспечение защиты файлов от несанкционированного доступа.
5. Обеспечение совместного доступа к файлу, для того что бы пользователю не приходилось прилагать специальных усилий, по обеспечению синхронизации доступа.
6. Обеспечение высокой производительности.

**Файл** – это поименованный набор связанной информацией, записанной во вторичную память.

**Файловая система** – это механизм для онлайнового хранения и доступа как к данным, так и к программам для всех пользователей ОС.

Общие сведения о файлах: файлы это абстрактные объекты, задачей которых является хранение информации, и скрытие от пользователя деталей работы с устройствами.

Основные типы данных:

1. Обычные файлы
2. Директории

Иногда к файлам приписывают другие объекты ОС, это спец символьные файлы, спец блочные файлы, и обычные файлы.

**Обычные файлы** – представляют собой набор блоков, на устройстве внешней памяти, на которых поддерживается файловая система, и могут содержать как текстовою, так и другую произвольную информацию.

Атрибуты файлов

Содержат: адресную информацию (устройство, начальный адрес, размер) информацию об управлении доступа, информацию об использовании.

Организация файлов и доступ к ним

В большинстве ОС размер одной записи равен одному байту. Различные приложения оперируют данными. Записи объединяются в блоки для вывода, и разблокируются для ввода.

18.11.17



Размер логических блоков файла совпадает или является кратным размеру физического блока диска и может быть задан равным размеру страницы виртуальной памяти, которая поддерживается аппаратурой компьютера совместно с операционной системой. В структуре системы управления файлами мы выделяем базисную подсистему, которая отвечает за выделение дискового пространства конкретным файлам и выделяем логическую подсистему, которая использует структуру дерева директорий для предоставления модулю базисной подсистемы необходимой ей информацией, исходя из имени файла. Стандартный запрос на открытие поступает от прикладной программы к логической подсистеме, которая используя структуру директорий проверяет права доступа и вызывает базовую подсистему для получения доступа к блокам файла. После этого файл считается открытым. Он содержится в таблице открытых файлов, и используется в запросах прикладной программы на чтение-запись из этого файла. Запись в таблице открытых файлов указывает через систему выделения блоков диска на блоки данного файла. Если к моменту открытия файл уже используется другим процессом, т.е. содержится в таблице открытых файлов, то после проверки правд доступа файлу может быть организован совместный доступ, при этом новому процессу, так же возвращается дескриптор, т.е. ссылка на файл в виде таблицы открытых файлов.

**Современные архитектуры файловых систем**

Современные ОС предоставляют пользователю возможность работать сразу с несколькими файловыми системами. Файловая система становится частью общей многоуровневой системой. На верхнем уровне находится диспетчер файловых систем. Он связывает запросы прикладной программы конкретной файловой системой. Каждая файловая система (драйвер файловой системы) на этапе инициалицазии регистрируется у диспетчера, сообщая ему точки входа для последующих обращений к данной файловой системе (далее ФС), в рамках одной ОС может быть реализовано исходя из концепции виртуальной ФС. Виртуальная ФС представляет собой независимый от реализации уровень и опирается на реальные ФС виртуальных индексных узлов, которые обобщают индексные узлы конкретных систем. Реализация ФС связанна с такими вопросами как поддержка понятия логического блока диска связывания имени файла и блоков его данных, проблемами разделения и проблемами управления дискового пространства. Наиболее распространённые способы выделения дискового пространства — это непрерывное выделение, организация связанного диска и система с индексными узлами. ФС часто реализуется в виде слоеной модульной структуры. Нижние слои имею дело с оборудованием, а верхние символическими именами и логическими свойствами файлов. Директории могут быть организованы различными способами и могут хранить атрибуты файла и адреса блоков файла, а иногда для этого предназначается специальная структура (индексные узлы). Проблемы надежности и производительности ФС важнейшие аспекты ее дизайна.

ы