

Доклад по теме - Методы управления памятью в операционных системах

Дисциплина: Операционные системы

Неустроева Ирина Николаевна

Содержание

1	Актуальность	5
2	Цель работы	6
3	Основная часть	7
3.1	Управление памятью	7
3.2	Распределение памяти фиксированными разделами	8
3.3	Распределение памяти динамическими разделами	9
3.4	Перемещаемые разделы	10
3.5	Понятие виртуальной памяти	11
3.6	Страничное распределение	12
3.7	Сегментный способ распределения виртуальной памяти	12
3.8	Странично-сегментное распределение	13
4	Выводы	14
	Список литературы	15

Список иллюстраций

Список таблиц

1 Актуальность

Память является важнейшим ресурсом, требующим тщательного управления со стороны операционной системы. В ранних ОС управление памятью сводилось просто к загрузке программы и ее данных из некоторого внешнего накопителя (магнитной ленты, магнитного диска) в память. С появлением мультипрограммирования перед ОС были поставлены новые задачи, связанные с распределением имеющейся памяти между несколькими одновременно выполняющимися программами.

2 Цель работы

Изучить методы управления памятью в операционных системах.

3 Основная часть

3.1 Управление памятью

Управление памятью - это процесс управления и координации памяти компьютера для оптимизации общей производительности системы. Это наиболее важная функция операционной системы, которая управляет основной памятью. Это помогает процессам перемещаться вперед и назад между основной памятью и исполнительным диском. Для однозадачных ОС управление памятью не является серьезной проблемой, поскольку вся память, может быть отдана в распоряжение единственного процесса. Процедуры управления памятью решают следующие задачи:

1. выделение памяти для процесса при его запуске и освобождение этой памяти при завершении процесса.
2. обеспечение настройки программы на выделенные адреса памяти.
3. управление выделенными областями памяти по запросам программы

Совершенно иначе обстоят дела в многозадачных ОС. Суммарные требования к объему памяти всех одновременно работающих в системе программ, как правило, превышают имеющийся в наличии объем основной памяти. В этих условиях ОС не имеет другого выхода, кроме поочередного вытеснения процессов или их частей на диск, чтобы использовать освободившуюся память на нужды других процессов. Процедуры управления памятью решают следующие задачи:

1. предоставление процессам возможностей получения и освобождения дополнительных областей памяти в ходе работы.
2. эффективное использование ограниченного объема основной памяти для всех работающих процессов, в том числе с использованием дисков как расширения памяти.
3. изоляция памяти процессов, исключая неправильное обращение одного процесса к областям памяти, занимаемым другим процессом.
4. предоставление процессам возможности обмена данными через общие области памяти. Все методы управления памятью могут быть разделены на два класса: методы, которые используют перемещение процессов между оперативной памятью и диском, и методы, которые не делают этого.

3.2 Распределение памяти фиксированными разделами

Память разбивается на несколько разделов областей фиксированной величины. Такое разбиение может быть выполнено вручную человеком во время старта системы или ее установки. После этого границы разделов не изменяются. Разделы могут быть разного размера. Распределение процессов в разделы происходит следующим образом: Очередной новый процесс, поступивший на выполнение, помещается либо в общую очередь либо в очередь к некоторому разделу. Можно условно разделить программы на три типа: большая, средняя, маленькая. Соответственно, есть раздел большой, средний и маленький, то маленькие программы встают в очередь к маленькому разделу, средний к среднему, большие к большому. Но если количество задач в системе неодинаковое, то возникает длинная очередь, например, из маленьких программ. А большие разделы простаивают. Если мы поместим маленькую программу в большой раздел, но когда придет большая программа её поместить некуда. Эту ситуацию легко исправить если сделать единую очередь задач, они могут поступать в любой раздел, как показано на рисунке а. Планировщик задач будет пробегать по очереди

задач и распределять их по разделам.

Подсистема управления памятью в этом случае выполняет следующие задачи:

- Сравнивает объем памяти, требуемый для вновь поступившего процесса, с размерами свободных разделов и выбирает подходящий раздел.
- Осуществляет загрузку программы в один из разделов и настройку адресов. Уже на этапе трансляции разработчик программы может задать раздел, в котором ее следует выполнять. Это позволяет сразу, получить машинный код, настроенный на конкретную область памяти.

При очевидном преимуществе - простоте реализации, данный метод имеет существенный недостаток - жесткость. Так как в каждом разделе может выполняться только один процесс, то уровень мультипрограммирования заранее ограничен числом разделов. Независимо от размера программы она будет занимать весь раздел. Так, например, в системе где 3 раздела памяти, невозможно выполнять одновременно более трех процессов, даже если им требуется совсем мало памяти. С другой стороны, разбиение памяти на разделы не позволяет выполнять процессы, программы которых не помещаются ни в один из разделов. Такой способ управления памятью применялся в ранних мультипрограммных ОС.

3.3 Распределение памяти динамическими разделами

В этом случае память машины не делится заранее на разделы. Сначала вся память свободна. Каждому вновь поступающему процессу выделяется вся необходимая ему память. После завершения процесса память освобождается, и на это место может быть загружен другой процесс. Таким образом, в произвольный момент времени оперативная память представляет собой случайную последовательность занятых и свободных участков (разделов) произвольного размера. Но тут возникает проблема, фрагментации- это наличие большого числа участ-

ков свободной памяти очень маленького размера (фрагментов) настолько маленького, что ни одна из вновь поступающих программ не может поместиться ни в одном из участков

К функциям данного метода относятся:

- Ведение таблиц свободных и занятых областей -Загрузка программы в выделенный ей раздел и корректировка таблиц свободных и занятых областей.

Данный метод обладает гораздо большей гибкостью, но ему присущ очень серьезный недостаток – фрагментация памяти. Распределение памяти динамическими разделами лежит в основе подсистем управления памятью многих мультипрограммных операционных системах 60-70-х годов.

3.4 Перемещаемые разделы

Одним из методов борьбы с фрагментацией является перемещение всех занятых участков в сторону старших или младших адресов, так, чтобы вся свободная память образовала единую свободную область. В дополнение к функциям, которые выполняет ОС при распределении памяти динамическими разделами в данном случае она должна еще время от времени копировать содержимое разделов из одного места памяти в другое, корректируя таблицы свободных и занятых областей. Эта процедура называется сжатием. Сжатие может выполняться либо при каждом завершении процесса, либо только тогда, когда для вновь создаваемого процесса нет свободного раздела достаточного размера. Хотя процедура сжатия и приводит к более эффективному использованию памяти, она может потребовать значительного времени, что часто перевешивает преимущества данного метода.

Способы, которые мы рассмотрели, они древние. Современным подходом к управлению памятью является виртуальная память.

3.5 Понятие виртуальной памяти

Уже достаточно давно пользователи столкнулись с проблемой размещения в памяти программ, размер которых превышал имеющуюся в наличии свободную память. Решением было разбиение программы на части, называемые оверлеями. 0-ой оверлей начинал выполняться первым. Когда он заканчивал свое выполнение, он вызывал другой оверлей. Все оверлеи хранились на диске и перемещались между памятью и диском средствами операционной системы. Однако разбиение программы на части и планирование их загрузки в оперативную память должен был осуществлять программист.

Развитие методов организации вычислительного процесса в этом направлении привело к появлению метода - виртуальная память. Так, например, пользователю может быть предоставлена виртуальная оперативная память, размер которой превосходит всю имеющуюся в системе реальную оперативную память. Пользователь пишет программы так, как будто в его распоряжении имеется однородная оперативная память большого объема, но в действительности все данные, используемые программой, хранятся на одном или нескольких запоминающих устройствах, обычно на дисках. Если мы обращаемся к процессу, которого нет в ОП, мы просто подгружаем ее по частям. Таким образом, виртуальная память - это совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер которых превосходит имеющуюся оперативную память; для этого виртуальная память решает следующие задачи:

- размещает данные в запоминающих устройствах разного типа, например, часть программы в оперативной памяти, а часть на диске;
- перемещает по мере необходимости данные между запоминающими устройствами разного типа, например, подгружает нужную часть программы с диска в оперативную память;

Все эти действия выполняются автоматически, без участия программиста. Наиболее распространенными реализациями виртуальной памяти является

страничное, сегментное и странично-сегментное распределение памяти.

3.6 Страничное распределение

При таком способе все фрагменты программы, на которые она разбивается, получаются одинаковыми и называются страницами. Память разбивается на страничные кадры, Размер страницы будет занимать целиком кадр. А Кадр для удобства будет равен степени 2. Чтобы запустить процесс нам нужно будет несколько кадров. Например, процесс А занимает 2 мегабайта, каждый кадр по одному. Кроме всего прочего, нам нужно поддерживать таблицу страниц, в которой находятся номера кадров, занятые тем или иным процессом и такие таблицы должны существовать для каждого процесса. Память нарезана на странички одинакового размера и может эффективно размещать процессы в памяти при этом внешняя фрагментация отсутствует, а внутренняя будет не существенна. Важно чтобы ОС поддерживала таблицы страниц.

3.7 Сегментный способ распределения виртуальной памяти

Виртуальное пространство реализовано таким образом, что есть таблица сегментов, она может находится в регистрах или в ОП и в этой таблице есть длина сегмента и есть базовый адрес. И когда у нас приходит логический адрес программы, который состоит из смещения и номера сегмента, то операционная система, используя номер сегмента как индекс таблицы, берет базовый адрес к нему прибавляет смещение и проверяет, что получившийся адрес не выходит за размер сегмента. Так получается физический адрес. Получается, что мы берем адресное пространство нашего компьютера и располагаем программы таким образом, что каждое расположение создает запись в этой таблице, при

этом внутренняя сегментация отсутствует, потому что сегменты занимают ровно столько, сколько нужно программе, а внешняя фрагментация снижается, так нам не нужно перемещать сегменты, а достаточно указать их характеристики в этой таблице и процесс будет занимать ровно столько места, сколько мы ему отведем.

3.8 Странично-сегментное распределение

Данный метод представляет собой комбинацию страничного и сегментного распределения памяти и, вследствие этого, сочетает в себе достоинства обоих подходов. Виртуальное пространство процесса делится на сегменты, а каждый сегмент в свою очередь делится на виртуальные страницы, которые нумеруются в пределах сегмента. Оперативная память делится на физические страницы. Загрузка процесса выполняется операционной системой постранично, при этом часть страниц размещается в оперативной памяти, а часть на диске. Для каждого сегмента создается своя таблица страниц, в которой указываются адреса таблиц страниц для всех сегментов данного процесса. Адрес таблицы сегментов загружается в специальный регистр процессора, когда активизируется соответствующий процесс.

4 Выводы

Мы рассмотрели основные методы распределения памяти в операционных системах и выяснили, что методы, которые не используют внешнюю память, имеют фиксированный объем памяти для процессов пользователя, а методы, которые используют виртуальную память, способны на выполнение многих процессов, которые превышают объем однородной оперативной памяти.

Список литературы

1. Олифер В. Г. Сетевые операционные системы: учебник для вузов/ В.Г. Олифер, Н.А. Олифер. - СПб.: Питер, 2009. - 539 с.: ил
2. Попов И.И. Операционные системы, среды и оболочки/ И.И. Попов. - М.: Издательство “ФОРУМ: ИНФА-М”, 2010. - 400 с. - (Серия “Профессиональное образование”)
3. Глава 3. Управление памятью в операционных системах (studylib.ru)