Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

«Сравнительный анализ методов замещения страниц»

Студент ИУ7-23М (Группа) _____ (Подпись, дата) Мицевич М. Д. (И. О. Фамилия)

Руководитель НИР _____ (Подпись, дата) Тассов К. Л. (И. О. Фамилия)

2024 г.

РЕФЕРАТ

Расчетно-пояснительная записка к научно-исследовательской работе содержит 16 страниц, 1 иллюстраций, 0 таблица, 8 источников.

Научно-исследовательская работа представляет собой изучение предметной области управления памятью и замещения страниц, описание основных методов, а также преимуществ и недостатков каждого из них. Рассмотрены различные подходы управления памятью и замещения страниц. Представлено описание методов исключения недавно использовавшейся страницы, first in first out и его модификации, замещения наименее востребованной, алгоритм рабочего набора и WSClock.

Ключевые слова: Операционная система, память, алгоритм замещения страниц.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

С течением времени были разработаны два основных подхода для преодоления перегрузки памяти [1]. Один из них, называемый свопингом, заключается в полном размещении процесса в памяти при его запуске на некоторое время, а затем в его сбросе на диск. Бездействующие процессы хранятся на диске, освобождая оперативную память. Второй подход, называемый виртуальной памятью, позволяет программам запускаться, даже если они находятся в оперативной памяти лишь частично. В этом случае страницы загружаются в оперативную память и сбрасываются на диск по мере необходимости во время выполнения программы.

Процесс сброса страниц на диск и возвращение их в оперативную память является трудоемким, так как требует обращения к ресурсам жесткого диска и упирается в его пропускную способность. Минимизация числа таких операций может повысить производительность системы [2].

Алгоритмы замещения страниц являются важной частью управления памятью в операционных системах. Они определяют, какие страницы памяти должны быть удалены из физической памяти при нехватке места и какие страницы должны быть загружены обратно при необходимости. Различные алгоритмы замещения страниц имеют разные стратегии и критерии для принятия решений о замещении страниц.

Целью данной работы является сравнительный анализ методов замещения страниц. Для достижения поставленной цели нужно выполнить следующие задачи:

— провести анализ предметной области управления памятью;

— сформулировать критерии сравнения методов;

— провести сравнительный анализ методов замещения страниц.

1 Анализ предметной области

1.1 Управление памятью

Виртуальная память представляет собой ключевую концепцию в управлении памятью современных компьютерных систем. Она позволяет программам использовать объем оперативной памяти, превышающий физически доступный, за счет автоматического перемещения данных между основной памятью и вторичным хранилищем . Это достигается благодаря использованию виртуальных адресов, которые транслируются в физические адреса с помощью аппаратных средств.

Каждая программа работает с собственным адресным пространством, которое разбивается на страницы, представляющие собой непрерывные диапазоны адресов. Эти страницы не обязательно должны все одновременно находиться в оперативной памяти для выполнения программы, что позволяет эффективно использовать доступную память.

Когда программа обращается к данным, которые уже находятся в физической памяти, аппаратное обеспечение обеспечивает необходимое отображение адресов. Когда программа пытается получить доступ к странице, которая присутствует в виртуальном адресном пространстве, но отсутствует в физической памяти возникает системное прерывания отсутствия страницы после чего управление передается операционной системе.

Операционная система реагирует на ошибку отсутствия страницы, выбирая страницу, которая редко используется, и сбрасывая её содержимое на диск, если оно уже не находится там. Затем система извлекает нужную страницу с диска и помещает её в освободившееся место в памяти. После этого в таблицы вносятся соответствующие изменения, и прерванная команда выполняется заново.

Таблица страниц содержит сведения о каждой странице, включая номер страничного блока, который является ключевым элементом страничного отображения. Также в информации содержится бит присутствия-отсутствия, если он равен 1, запись активна и может быть использована, иначе соответствующая виртуальная страница в данный момент отсутствует в памяти, и любое обращение к такой записи вызывает ошибку отсутствия страницы. Биты за-

щиты указывают на тип доступа, который разрешен для страницы. В самом простом случае это один бит, который равен 0 для чтения-записи и 1 для только чтения. В более сложных системах могут быть использованы три бита, каждый из которых разрешает чтение, запись или исполнение страницы. Биты модификации и ссылки служат для отслеживания использования страницы. Бит модификации автоматически устанавливается при записи в страницу и помогает операционной системе определить, нужно ли сохранять страницу на диск при ее выгрузке из памяти. Бит ссылки устанавливается при любом обращении к странице и помогает операционной системе определить, какую страницу следует выгрузить при возникновении ошибки отсутствия страницы.

Изучив поведение программ, разработчики компьютерных систем пришли к выводу, что большинство программ часто обращаются к ограниченному набору страниц. Из-за этого только небольшая часть записей в таблице страниц активно используется, а остальная практически не задействуется. На основе этого наблюдения для повышения производительности системы было предложены добавить в аппаратуру специальное устройство, которое называется TLB, и отвечает за трансляцию виртуальных адресов в физические для самых используемых страниц.

При использовании программного управления TLB существует 2 типа ошибок: программные и аппаратные. Программная ошибка возникает, когда страница отсутствует в TLB, но есть в памяти, и ее можно исправить простым обновлением TLB без обращения к диску. Это занимает 10-20 машинных команд и несколько наносекунд. Аппаратная ошибка возникает, когда страница отсутствует в памяти и требуется обращение к диску, что занимает несколько миллисекунд. Она обрабатывается значительно медленнее программной ошибки.

При возникновении ошибки отсутствия страницы [3], операционная система должна определить, какую страницу из памяти выселить, чтобы освободить место для загружаемой страницы. Если страница, которую нужно выселить, была изменена с момента загрузки в память, то ее содержимое должно быть обновлено на диске. Если страница не подвергалась изменениям и дисковая копия актуальна, то перезапись не требуется. В этом случае новая страница просто замещает старую.

1.2 Оптимальный алгоритм

Оптимальной алгоритм предлагает вытеснять страницу, которая будет без ссылок в течение самого длительного времени. Этот алгоритм может быть реализован только во втором идентичном прогоне при условии истории использования страниц во время первого запуска [4]. Обычно у операционной системы этой истории нет, особенно в приложениях, получающих внешние данные. Адрес содержание и точное время ввода могут сильно изменить порядок и время обращения к страницам. Оптимальный алгоритм может быть использован для оценки других алгоритмов замещения страниц, которые могут быть применены и при первом прогоне.

1.3 Алгоритм исключения недавно использовавшейся страницы

Для того чтобы собирать статистику использования страниц виртуальной памяти, большинство компьютеров используют два бита состояния для каждой страницы. Бит R устанавливается при обращении к странице, а бит M устанавливается, когда страница изменяется.

Если аппаратура не поддерживает эти биты, то они могут быть созданы с помощью механизмов операционной системы. При запуске процесса все записи в его таблице страниц помечаются как отсутствующие в памяти. Когда происходит обращение к странице, возникает ошибка отсутствия страницы, и операционная система устанавливает бит R, изменяет запись в таблице страниц, устанавливая режим доступа только для чтения, и перезапускает команду. Если страница впоследствии изменяется, возникает другая ошибка страницы, позволяющая операционной системе установить бит M и изменить режим доступа к странице на чтение-запись.

Идея алгоритма исключения недавно использовавшейся страницы заключается в следующем: при запуске процесса оба этих бита для всех страниц устанавливаются в 0 операционной системой. При каждом прерывании от таймера бит R сбрасывается, чтобы отличить страницы, к которым не было обращений в последнее время, от тех, к которым были такие обращения.

При возникновении ошибки отсутствия страницы операционная система анализирует все страницы и на основе текущих значений битов R и M разделяет их на четыре категории:

1. К которым не было ни обращений, ни модификаций в последнее время.
2. К которым не было обращений в последнее время, но были модификации.
3. К которым были обращения в последнее время, но не было модификаций.
4. К которым были и обращения, и модификации в последнее время.

Для замещения выбирается произвольная страница из самого низкого непустого класса.

1.4 Алгоритм first in first out и его модификации

Операционная система ведет список всех страниц, находящихся в памяти в данный момент. Недавно поступившие страницы находятся в конце списка, а те, что поступили раньше всех, находятся в начале. Если возникает ошибка отсутствия страницы, удаляется страница из начала списка, и в конец добавляется новая страница.

Алгоритм второй шанс является простой модификацией алгоритма FIFO и решает проблему удаления часто востребуемой страницы. Для этого используется проверка бита R самой старой страницы. Если значение этого бита равно нулю, то это означает, что страница не только старая, но и невостребованная, поэтому она сразу же удаляется. Если бит R имеет значение 1, то он сбрасывается, а страница помещается в конец списка страниц, а время ее загрузки обновляется, как будто она только что поступила в память. Затем поиск продолжается.

Алгоритм часы является улучшением алгоритма второй шанс. Он основан на идее использования циклического списка страниц, представленного в виде часов, где стрелка указывает на самую старую страницу.

Принцип работы алгоритма часы следующий:

1. В начале работы алгоритма все страницы помещаются в циклический список в виде часов, где каждая страница имеет бит R, который указывает на ее актуальность.

2. При возникновении ошибки отсутствия страницы проверяется страница, на которую указывает стрелка в циклическом списке.

3. Если бит R этой страницы равен 0, она удаляется из памяти, на ее место загружается новая страница, и стрелка сдвигается вперед на одну позицию.

4. Если бит R равен 1, он сбрасывается, и стрелка перемещается на следующую страницу в списке.

5. Этот процесс повторяется до тех пор, пока не будет найдена страница с битом R = 0.

1.5 Алгоритм замещения наименее востребованной страницы

Алгоритм замещения наименее востребованной страницы основан на идее, что страницы, которые долгое время не были востребованы, скорее всего останутся невостребованными, в то время как страницы, которые интенсивно использовались в последнее время, вероятно будут снова востребованы. Поэтому стратегия замещения страниц в этом алгоритме основана на выборе наименее востребованной страницы для удаления.

Для реализации алгоритма NRU каждая страница в памяти связывается с программным счетчиком, который имеет начальное значение 0. При каждом прерывании от таймера операционная система сканирует все страницы в памяти. Для каждой страницы к счетчику добавляется значение бита R, который равен 0 или 1. Таким образом, счетчики позволяют приблизительно отслеживать частоту обращений к каждой странице.

При возникновении ошибки отсутствия страницы для замещения выбирается та страница, у которой счетчик имеет наименьшее значение, то есть та страница, которая дольше всего не была востребована.

Основная проблема этого алгоритма заключается в том, что он никогда не сбрасывает счетчики и страницы, которые активно использовались в прошлом, и сейчас не востребованы все равное будут оставаться в памяти [5].

Для борьбы с этой проблемой существует алгоритм старения, который предлагает при каждом прерывании таймера не прибавлять 1 к счетчику, а делать сдвиг вправо и прибавлять 1 к левому биту счетчика.

1.6 Алгоритм рабочий набор

Текст описывает процесс использования замещения страниц в операционных системах. Процессы начинают работу без каких-либо страниц в памяти, что приводит к ошибкам отсутствия страниц при первом обращении к данным. Операционная система загружает страницы по мере необходимости. Постепенно процесс получает большинство необходимых ему страниц и начинает работу более стабильно. Рабочий набор страниц, используемых процессом в данный момент, важен для эффективной работы. Многие системы замещения страниц стремятся отслеживать рабочий набор каждого процесса и обеспечивать его присутствие в памяти, перед перезапуском процесса.

Для реализации модели рабочего набора необходимо, чтобы операционная система отслеживала, какие страницы именно входят в рабочий набор. Имея эту информацию, можно использовать следующий алгоритм замещения страниц: при возникновении ошибки отсутствия страницы следует выселить ту страницу, которая не принадлежит рабочему набору.

Рабочий набор представляет собой набор страниц, используемых в k последних обращениях к памяти. Для реализации алгоритма можно отслеживать страницы, использованные в k последних миллисекундах выполнения, вместо поиска страниц, используемых в k последних обращениях. Для получения этой информации можно добавить специальное поле в таблицу страниц и обновлять его на основе бита R по тику таймера.

Если "возраст"страницы превышает заранее выбранное значение на момент возникновения ошибки, она становится кандидатом на замену. В противном случае удаляется страница с самым большим "возрастом или случайная, если у всех страниц одинаковый параметр.

1.7 Алгоритм WSClock

Алгоритм WSClock (Working Set Clock) является модификацией алгоритма рабочего набора и базируется на структуре данных, аналогичной циклическому списку страничных блоков, используемой в алгоритме часы. Основные принципы работы данного алгоритма следующие:

1. Создается пустой циклический список страничных блоков.
2. При загрузке первой страницы она добавляется в список. По мере загрузки следующих страниц они также попадают в список, формируя замкнутое кольцо.
3. В каждой записи списка содержится поле времени последнего использования из базового алгоритма рабочего набора, а также биты R и M.
4. При возникновении ошибки отсутствия страницы сначала проверяется страница, на которую указывает "стрелка"в списке. Если бит R установлен в 1, это означает, что страница была использована в течение текущего такта и не является идеальным кандидатом на удаление.
5. Затем бит R устанавливается в 0, стрелка перемещается на следующую страницу в списке, и процесс повторяется уже для нее.
6. После того, как бит R у страницы, на которую указывает стрелка, равен 0 и ее возраст превышает заданное значение, а также страница не изменена, происходит замещение этой страницы.
7. Если страница была изменена, то планируется запись на диск.

Если стрелка проходит полный круг и хотя бы одна запись на диск запланирована, поиск может продолжаться до тех пор, пока не будет найдена неизмененная страница. В противном случае все страницы считаются частью рабочего набора, и замещается любая страница, которая не была изменена. Если такой страницы нет, то замещается текущая страница.

1.8 Сравнение методов замещения страниц

Оптимальный алгоритм удаляет страницу с самым отдаленным предстоящим обращением. На практике реализовать такой алгоритм невозможно, но его можно использовать в качестве оценочного критерия.

Алгоритм исключения недавно использовавшейся страницы проводит разбиение всех страниц, основываясь на состоянии битов M и R, на 4 класса и проводит замещение произвольной страницы наименьшего непустого класса.

Алгоритм FIFO работает по принципу очереди и удаляет самую старую страницу. Алгоритм второй шанс борется с недостатками FIFO и перед удалением страницы проверяет не используется ли она в данный момент. Алгоритм часы является разновидностью алгоритма второй шанс, но требует меньше времени на выполнение.

Алгоритм замещения наименее востребованной страницы стремится удалять страницы, которые не были востребованы долгое время. У этого алгоритма есть недостаток, связанный с тем, что страница, которая активно использовалась в прошлом, не обязательно будет востребована сейчас. Для борьбы с этим недостатком был разработан алгоритм старения.

Алгоритм рабочего набора отслеживает набор страниц, используемых за определенный промежуток времени и замещает страницу, которая не относится к рабочему набору. Алгоритм WSClock является оптимизацией алгоритма рабочего набора.

На практике чаще всего используются алгоритм старения и WSClock. Оба обеспечивают неплохую производительность страничной организации памяти и могут быть эффективно реализованы, но не лишены недостатков на определенном наборе задач.

1.9 Постановка задачи

На вход метода поступает информация о текущей таблице страниц, в которой содержится информация о битах модификации и доступа каждой страницы, текущий момент времени и виртуальные адреса страниц. Idef0 диаграмма метода изображена на рисунке 1.1.

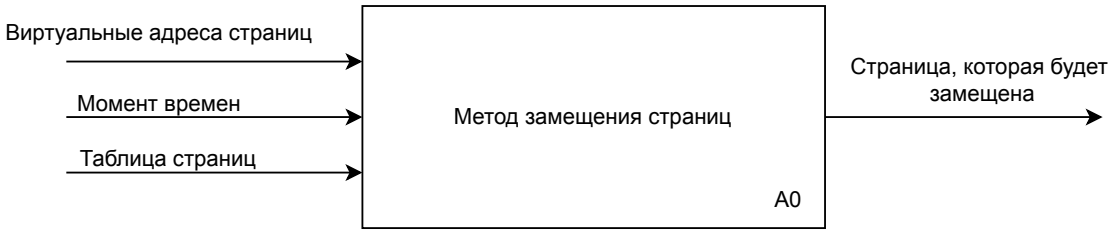

Рисунок 1.1 – Idef0 диаграмма

ЗАКЛЮЧЕНИЕ

Алгоритмы замещения страниц играют важную роль в оптимизации работы операционных систем и управлении памятью. Изучение и понимание различных алгоритмов позволяет эффективно управлять доступом к данным в памяти, минимизировать количество обращений к диску и повысить производительность системы в целом

В данной научно-исследовательской работе был:

— проведен анализ предметной области управления памятью;

— сформулированы критерии сравнения методов;

— проведен сравнительный анализ методов замещения страниц.

В рамках работы были выполнены все поставленные задачи. Цель работы была достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Управление процессами и памятью в операционных системах / С. В. Востокин [и др.] // Самара: Изд-во Самар. ун-та. — 2023. — Т. 1.
2. *Гусев К.*, *Леонтьев А.*, *Головин С.* Разработка алгоритмов учета влияния страничных сбоев на временные характеристики обработки заявок в вычислительных комплексах // International Journal of Open Information Technologies. — 2023. — Т. 11, № 11. — С. 10—18.
3. A page replacement algorithm based on a fuzzy approach to improve cache memory performance / D. Akbari Bengar [и др.] // Soft Computing. — 2020. — Т. 24, № 2. — С. 955—963.
4. *Tingare B. A.*, *Kolhe V. L.* Analysis of Various Page Replacement Algorithms in Operating System. — 2015.
5. *Таненбаум Э.* Современные операционные системы. — 2015.