

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите приведенную лекцию, законспектируйте основные понятия, дайте ответы на контрольные вопросы.

Ответы на вопросы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com **до 15.05.2023.**

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: <https://vk.com/daykini>

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лекция

Тема: Эффективные методы управления памятью.

Методы управления памятью

При построении систем с иерархической памятью целью является получение максимальной производительности подсистемы памяти при ее минимальной стоимости. Эффективность той или иной системы кэш-памяти зависит от стратегии управления памятью. Стратегия управления памятью включает: метод отображения основной памяти в кэше; алгоритм взаимодействия между медленной основной и быстрой кэш-памятью; стратегии замещения информации в кэше.

Существует три основных способа размещения блоков строк основной памяти в кэше: кэш-память с прямым отображением `direct-mapped cache` полностью ассоциативная кэш-память `fully associative cache`. частично ассоциативная или множественно ассоциативная, `partial associative`, `set-associative cache` кэш-память

Память с прямым отображением.

В этом случае каждый блок основной памяти имеет только одно фиксированное место, на котором он может появиться в кэш-памяти. Все блоки основной памяти, имеющие одинаковые младшие разряды в своем адресе, попадают в один блок кэш-памяти. При таком подходе справедливо соотношение:

Адрес блока кэш-памяти = Адрес блока основной памяти mod Число блоков в кэш-памяти.

Этот тип памяти наиболее прост, но и наименее эффективен, так как данные из разных областей памяти могут конфликтовать из-за единственной строки кэша, где они только и могут быть размещены.

Полностью ассоциативная память

Может отображать содержимое любой области памяти в любую область кэша, но при этом крайне сложна в схемотехнике.

Частично-ассоциативный кэш

Является наиболее распространенным в данный момент среди процессорных архитектур. Характеризуется тем или иным количеством n каналов степенью ассоциативности, n -way и может отображать содержимое данной строки памяти на каждую из n своих строк. Этот вариант является разумным компромиссом между полностью ассоциативным и кэшем прямого отображения.

В современных процессорах, как правило, используется либо кэш-память с прямым отображением, либо двух- четырех- канальная множественно ассоциативная кэш-память. Например, в архитектурах K7 и K8 применяется 16-канальный частично-ассоциативный кэш L2.

Стратегия замещения информации в кэше определяет блок, подлежащий замещению при возникновении промаха. Простота при использовании кэша с прямым отображением заключается в том, что аппаратные решения здесь наиболее простые: легко реализуется сама аппаратура, легко происходит замещение данных. При замещении просто нечего выбирать — на попадание проверяется только один блок и только этот блок может быть замещен.

При полностью или частично ассоциативной организации кэш-памяти имеются несколько блоков, из которых надо выбрать кандидата в случае промаха. Как правило, для замещения блоков применяются две основные стратегии: случайная Random — блоки-кандидаты выбираются случайно равномерное распределение. В некоторых системах используют псевдослучайный алгоритм замещения

замещается тот блок, который не использовался дольше всех LRU — Least-Recently Used. В этом случае чтобы уменьшить вероятность удаления информации, которая скоро может потребоваться, все обращения к блокам фиксируются.

Достоинство случайного способа заключается в том, что его проще реализовать в аппаратуре. Когда количество блоков увеличивается, алгоритм LRU становится все более дорогим и часто только приближенным.

Алгоритмы обмена с кэш-памятью свопинга включают следующие разновидности:

- алгоритм сквозной записи Write Through или сквозного накопления Store Through
- алгоритм простого свопинга Simple Swapping или обратной записи Write Back
- алгоритм свопинга с флагами Flag Swapping или обратной записи в конфликтных ситуациях с флагами CUX
- алгоритм регистрового свопинга с флагами FRS.

Алгоритм сквозной записи

Самый простой алгоритм свопинга. Каждый раз при появлении запроса на запись по некоторому адресу обновляется содержимое области по этому адресу как в быстрой, так и в основной памяти, даже если копия содержимого по этому адресу находится в быстром буфере. Такое постоянное обновление содержимого основной памяти, как и буфера, при каждом запросе на запись позволяет постоянно поддерживать информацию, находящуюся в основной памяти, в обновленном состоянии.

Поэтому, когда возникает запрос на запись по адресу, относящемуся к области, содержимое которой не находится в данный момент в быстром буфере, новая информация записывается просто на место блока, которое предполагается переслать в основную память без необходимости пересылки этого слова в основную память, так как в основной памяти уже находится его достоверная копия.

Алгоритм простого свопинга

Обращения к основной памяти имеют место в тех случаях, когда в быстром буфере не обнаруживается нужное слово. Эта схема свопинга повышает производительность системы памяти, так как в ней обращения к основной памяти не происходят при каждом запросе на запись, что имеет место при использовании алгоритма сквозной записи. Однако в связи с тем, что содержимое основной памяти не поддерживается в постоянно обновленном состоянии, если необходимого слова в быстром буфере не обнаруживается, из буфера в основную память надо вернуть какое-либо устаревшее слово, чтобы освободить место для нового необходимого слова. Поэтому из буфера в основную память сначала пересылается какое-то слово, место которого занимает в буфере нужное слово. Таким образом, происходят две пересылки между быстрым буфером и основной памятью.

Алгоритм свопинга с флагами

Данный алгоритм является улучшением алгоритма простого свопинга. В алгоритме простого свопинга, когда в кэш-памяти не обнаруживается нужное слово, происходит два обращения к основной памяти — запись удаляемого значения из кэша и чтение нового значения в кэш. Если слово с того момента, как оно попало в буфер из основной памяти, не подвергалось изменениям, то есть по его адресу не производилась запись оно использовалось только для чтения, то нет необходимости пересылать его обратно в основную память, потому что в ней и так имеется достоверная его копия; это обстоятельство позволяет в ряде случаев обойтись без обращений к основной памяти. Если, однако, слово подвергалось изменениям с тех пор, когда его копия была в последний раз записана обратно в основную память, то приходится перемещать его в основную память. Отслеживать изменения слова можно, пометив слово блок дополнительным флаг-битом. Изменяя значение флаг-бита при изменении слова, можно сформировать информацию о состоянии слова. Пересылать в основную память необходимо лишь те слова, флаги которых оказываются в установленном состоянии.

Алгоритм регистрового свопинга с флагами

Повышение эффективности алгоритма свопинга с флагами возможно за счет уменьшения эффективного времени цикла, что можно получить при введении регистра регистров временного хранения между кэш-памятью и основной памятью. Теперь, если данные должны быть переданы из быстрого буфера в основную память, они сначала пересылаются в регистр регистры временного хранения; новое слово сразу же пересылается в буфер из основной памяти, а уже потом слово, временно хранившееся в регистре, записывается в основную память. Действия в ЦП начинают опять выполняться, как только для этого возникает возможность. Алгоритм обеспечивает совмещение операций записи в основную память с обычными операциями над буфером, что обеспечивает еще большее повышение производительности.

Эксклюзивным называется кэш, в котором данные, хранящиеся в кэш-памяти первого уровня, не обязательно должны быть продублированы в кэшах нижележащих уровней. Инклюзивный кэш — когда любая информация, хранящаяся в кэшах высших уровней, дублируется в кэш-памяти.