АЛГОРИТМЫ ЗАМЕЩЕНИЯ СТРАНИЦ

При возникновении ошибки отсутствия страницы операционная система должна выбрать выселяемую (удаляемую из памяти) страницу, чтобы освободить место для загружаемой страницы. Если предназначенная для удаления страница за время своего нахождения в памяти претерпела изменения, она должна быть переписана на диске, чтобы привести дисковую копию в актуальное состояние. Но если страница не изме- нялась (например, она содержала текст программы), дисковая копия не утратила своей актуальности и перезапись не требуется. Тогда считываемая страница просто пишется поверх выселяемой.

Оптимальный алгоритм удаляет страницу с самым отдаленным предстоящим обра- щением. К сожалению, у нас нет способа определения, какая это будет страница, по- этому на практике этот алгоритм использоваться не может. Но он полезен в качестве оценочного критерия при рассмотрении других алгоритмов.

Алгоритм исключения недавно использованной страницы (NRU) делит страницы на четыре класса в зависимости от состояния битов *R* и *M*. Затем выбирает произвольную страницу из класса с самым низким номером. Этот алгоритм нетрудно реализовать, но он слишком примитивен. Есть более подходящие алгоритмы.

Алгоритм FIFO предполагает отслеживание порядка, в котором страницы были за- гружены в память, путем сохранения сведений об этих страницах в связанном списке. Это упрощает удаление самой старой страницы, но она-то как раз и может все еще использоваться, поэтому FIFO — неподходящий выбор.

Алгоритм «второй шанс» является модификацией алгоритма FIFO и перед удалением страницы проверяет, не используется ли она в данный момент. Если страница все еще используется, она остается в памяти. Эта модификация существенно повышает произ- водительность. Алгоритм «часы» является простой разновидностью алгоритма «вто- рой шанс». Он имеет такой же показатель производительности, но требует несколько меньшего времени на свое выполнение.

Алгоритм LRU превосходен во всех отношениях, но не может быть реализован без специального оборудования. Если такое оборудование недоступно, то он не может быть использован. Алгоритм NFU является грубой попыткой приблизиться к алгоритму LRU. Его нельзя признать удачным. А вот алгоритм старения — куда более удачное приближение к алгоритму LRU, которое к тому же может быть эффективно реализо- вано и считается хорошим выбором.

В двух последних алгоритмах используется рабочий набор. Алгоритм рабочего на- бора обеспечивает приемлемую производительность, но его реализация обходится слишком дорого. Алгоритм WSClock является вариантом, который не только обе- спечивает неплохую производительность, но и может быть эффективно реализован.

В итоге наиболее приемлемыми алгоритмами являются алгоритм старения и алгоритм WSClock. Они основаны на LRU и рабочем наборе соответственно. Оба обеспечивают неплохую производительность страничной организации памяти и могут быть эффек- тивно реализованы. Существует также ряд других хороших алгоритмов, но эти два, наверное, имеют наибольшее практическое значение.

|  |  |
| --- | --- |
| **Алгоритм** | **Особенности** |
| Оптимальный | Не может быть реализован, но полезен в каче- стве оценочного критерия |
| NRU (Not Recently Used) — алгоритм исключе- ния недавно использовавшейся страницы | Является довольно грубым приближением к алгоритму LRU |
| FIFO (First In, First Out) — алгоритм «первой пришла, первой и ушла» | Может выгрузить важные страницы |
| Алгоритм «второй шанс» | Является существенным усовершенствовани- ем алгоритма FIFO |
| Алгоритм «часы» | Вполне реализуемый алгоритм |
| LRU (Least Recently Used) — алгоритм заме- щения наименее востребованной страницы | Очень хороший, но труднореализуемый во всех тонкостях алгоритм |
| NFU (Not Frequently Used) — алгоритм неча- стого востребования | Является довольно грубым приближением к алгоритму LRU |
| Алгоритм старения | Вполне эффективный алгоритм, являющийся неплохим приближением к алгоритму LRU |
| Алгоритм рабочего набора | Весьма затратный для реализации алгоритм |
| WSClock | Вполне эффективный алгоритм |