АЛГОРИТМЫ СТРАНИЧНОГО ЗАМЕЩЕНИЯ 15

Курс лекций

«Системное программное обеспечение»
«System Software»
«Операционные системы»

для студентов специальностей АСОИ и ИИ

Павел Кочурко
доцент кафедры ИИТ, к.т.н.

БрГТУ

Стратегии управления виртуальной памятью: стратегия выборки

Стратегия выборки (fetch policy) - в какой момент следует переписать страницу из вторичной памяти в первичную:

- *по запросу* – страница подкачивается при обращении к отсутствующей странице
- *с упреждением* – наряду с той страницей, к которой было обращение, загружаются и соседние

  Поскольку размер буфера ввода-вывода диска в несколько раз больше размера страницы, за одно обращение к диску можно прочитать несколько страниц и сэкономить на возможном обращении к диску за ними в ближайшем будущем

Стратегии управления виртуальной памятью: стратегия размещения

Стратегия размещения (placement policy) - в какой участок первичной памяти поместить поступающую страницу.

- В системах со страничной организацией все просто - *в любой свободный страничный кадр*.
- В случае систем с сегментной организацией необходима стратегия, аналогичная стратегии с динамическим распределением.

Стратегии управления виртуальной памятью: стратегия замещения

Стратегия замещения (replacement policy) - какую страницу нужно вытолкнуть во внешнюю память, чтобы освободить место в оперативной памяти.

Разумная стратегия замещения, реализованная в соответствующем *алгоритме замещения страниц*, позволяет хранить в памяти самую необходимую информацию и тем самым снизить частоту страничных нарушений.

Замещение должно происходить с учетом выделенного каждому процессу количества кадров.

Должна ли замещаемая страница принадлежать процессу, который инициировал замещение, или она должна быть выбрана среди всех кадров основной памяти?

БрГТУ

Процесс замещения страниц

При замещении необходимо:

- найти некоторую занятую страницу основной памяти;
- переместить в случае надобности ее содержимое во внешнюю память (медленно);
- переписать в этот страничный кадр содержимое нужной виртуальной страницы из внешней памяти (медленно);
- должным образом модифицировать необходимый элемент соответствующей таблицы страниц;
- продолжить выполнение процесса, которому эта виртуальная страница понадобилась.

Чем реже это делается, тем выше производительность

.

Область замещения

Локальные алгоритмы распределяют фиксированное или динамически настраиваемое число страниц для каждого процесса

Глобальный алгоритм замещения в случае возникновения исключительной ситуации освобождает любую физическую страницы, независимо от того, какому процессу она принадлежала

Алгоритмы замещения страниц

- OPT: Оптимальный алгоритм

  Замещается страница, которая не будет использоваться в течение самого длительного периода времени.

  +: оптимальность

  -: нереализуемость

- LRU, Least Recently Used

  Выталкивание дольше всего не использовавшейся страницы (FIFO)

  Необходимо иметь связанный список всех страниц в памяти, в начале которого будут хранится недавно использованные страницы. Причем этот список должен обновляться при каждом обращении к памяти. Много времени нужно и на поиск страниц в таком списке.

  +: хорошая аппроксимация оптимального

  -: самая старая страница не обязательно неиспользуемая

БрГТУ

Алгоритмы замещения страниц, продолжение

- NFU, Not Frequently Used

  Выталкивание редко используемой страницы.

  Требуются программные счетчики, по одному на каждую страницу, которые сначала равны нулю. При каждом прерывании по времени (а не после каждой инструкции) операционная система сканирует все страницы в памяти и у каждой страницы с установленным флагом обращения увеличивает на единицу значение счетчика, а флаг обращения сбрасывает

  +: хорошая аппроксимация LRU

  -: алгоритм ничего не забывает

- Second Chance

  Модификация LRU, не выталкивающая самую старую, но используемую страницу

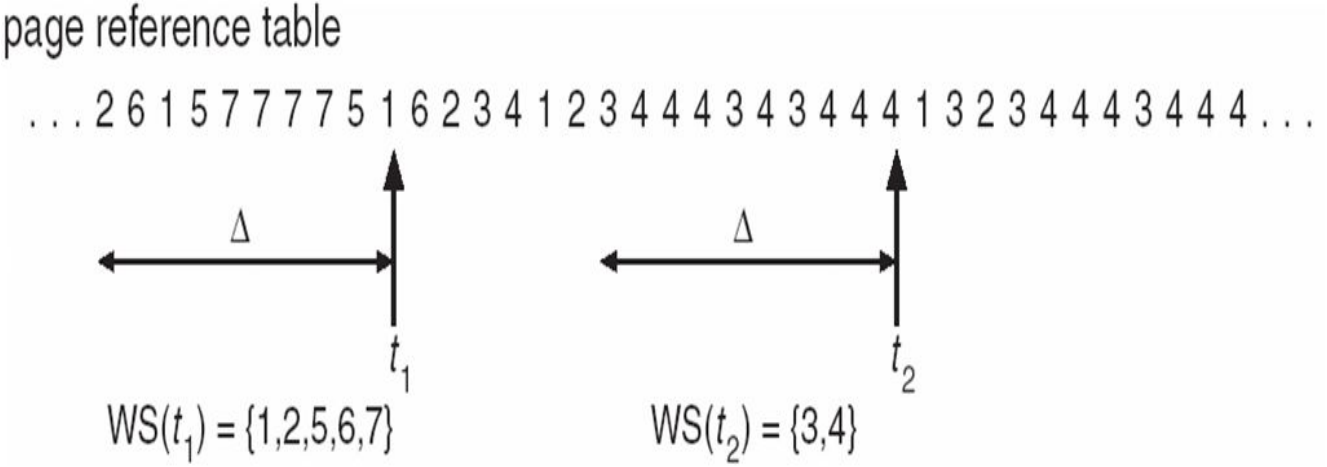
  С помощью анализа флага обращений (бита ссылки) для самой старой страницы. Если флаг установлен, то страница, в отличие от алгоритма FIFO, не выталкивается, а ее флаг сбрасывается, и страница переносится в конец очереди.

БрГТУ

Рабочее множество

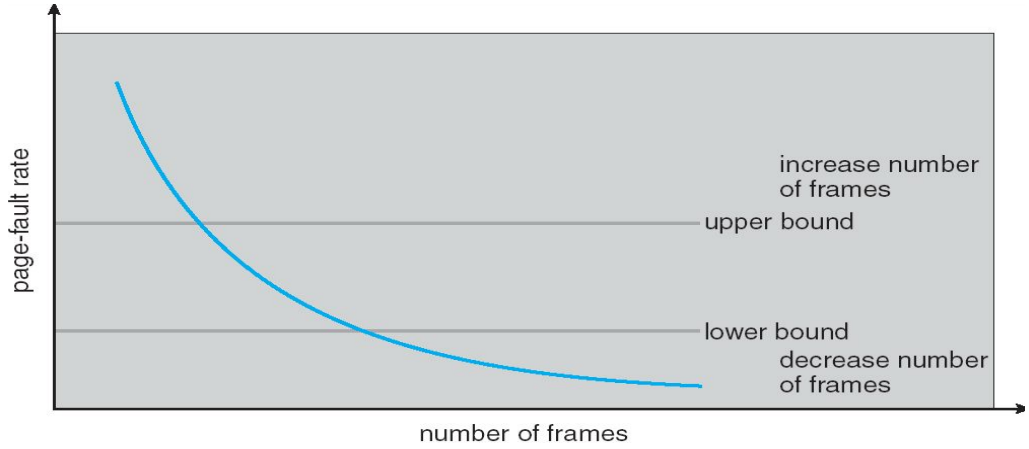
Резидентное множество процесса – набор физических кадров, выделенных процессу

Рабочее множество процесса - набор страниц (P_1, P_2, **...** P_n), активно использующихся вместе, который позволяет процессу в момент времени t в течение некоторого периода T производительно работать, избегая большого количества страничных нарушений.



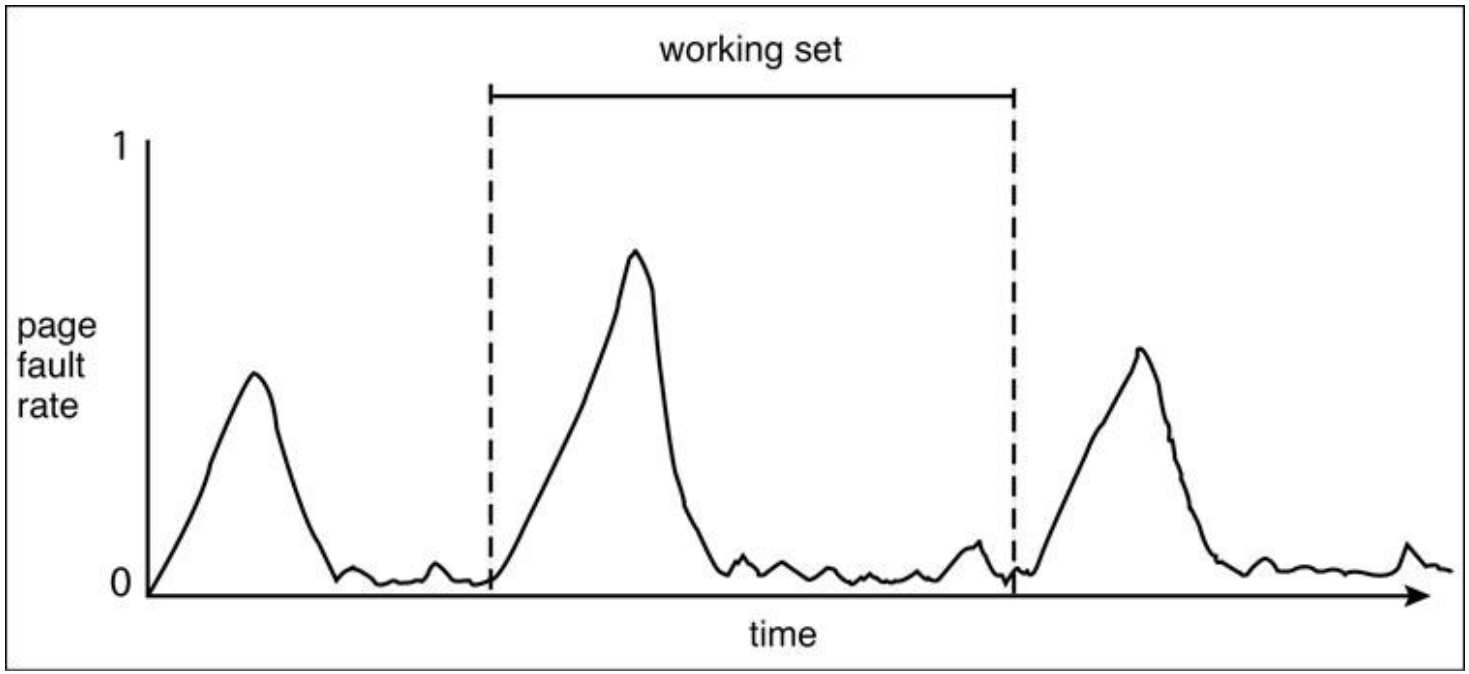
Частота страничных нарушений

- Если количество PF слишком низкое – процесс теряет кадры
- Если слишком высокое – получает дополнительные



Трэшинг (пробуксовка) – ситуация, при которой ЦПУ тратит больше времени на подкачку страниц, чем на выполнение команд

Рабочее множество vs. Страничные нарушения



ВОПРОСЫ?

http://iit.bstu.by/ss