Отчёт об испытаниях алгоритмов на случайных данных

Выполнил: Оленников Вадим

Почта: ovd2001@mail.ru

Stepik: stepik.org/users/278999948

Задание

Разработать методику и организовать сравнение различных алгоритмов (OPT, FIFO, LRU, CLOCK) на случайных данных при разных исходных параметрах (количество доступных кадров, количество страниц в адресном пространстве процесса, количество запросов к страницам). В процессе анализа можно ориентироваться как на число страничных ошибок, так и на другие характеристики. Результатом должен быть отчёт, включающий в себя:

- описание методики сравнения
- порядок проведения испытаний и формирования отчёта
- статистические оценки
- визуализацию полученных результатов
- заключение по итогам сравнения

Общая информация и терминология

Процессом называется единица исполнения в операционной системе. *Адресное пространство процесса* — это вся память, используемая для хранения кода и данных. Адресное пространство делится на последовательно пронумерованные страницы одинакового размера. В системах с виртуальной памятью считается, что в оперативную память (физическую память компьютера) загружается только часть страниц адресного пространства процесса, тогда как все остальные находятся на внешнем носителе. Оперативная память делится на кадры того же размера, что и страницы. Процесс попеременно обращается к разным своим страницам, причём работать он может только со страницами, загруженными в кадры оперативной памяти.

Будем считать, что количество кадров, доступных процессу, является фиксированным. Когда процессу требуется некоторая страница, операционная система должна убедиться, что она загружена в оперативную память. Ситуация, когда страницы в оперативной памяти в момент запроса нет, называется страничной ошибкой (***page fault***). В этот момент операционная система должна загрузить её в один из выделенных процессу кадров оперативной памяти. Если свободных кадров нет, то операционная система должна выгрузить одну из загруженных ранее страниц и загрузить на её место требуемую. Для определения выгружаемой страницы операционная система использует алгоритм замещения (***page replacement algorithm***).

На всех последующих шагах мы будем считать, что количество выделенных процессу кадров оперативной памяти и количество страниц в адресном пространстве процесса находятся в диапазоне от 1 до 10^3, причем процесс делает от 1 до 10^5 обращений (запросов) к своим страницам.

Описание алгоритмов можно прочитать в предыдущих заданиях.

Описание методики сравнения

Для сравнения эффективности алгоритмов были получены следующие зависимости:

- зависимость среднего времени работы алгоритма от количества запросов к страницам ***time(pageNumber)*** при фиксированных значениях количества кадров и количества страниц в адресном пространстве процесса
- зависимость количества страничных ошибок от количества кадров процесса ***pageFaults(frameNumber)*** при фиксированных значениях количества обращений и количества страниц в адресном пространстве процесса
- зависимость количества страничных ошибок от количества запросов к страницам ***pageFaults(pageNumber)*** при фиксированных значениях количества кадров и количества страниц в адресном пространстве процесса
- зависимость количества страничных ошибок от количества страниц в адресном пространстве ***pageFaults(maxPageValue)*** при фиксированных значениях количества кадров и количества запросов к страницам

Порядок проведения испытаний и формирования отчёта

Генерация данных происходит автоматически с помощью функции *generatePages()*. Для уменьшения случайной погрешности все измерения проводились несколько раз (количество замеров определяется константой *MaxCalcNumber*). При измерении времени и увеличения точности каждый алгоритм выполнялся несколько раз (количество замеров определяется константой *TimeRepeatNumber*).

Если при измерении зависимости количество страниц было фиксированным, то бралось наибольшее возможное количество страниц (константа *MaxPageNumber*) для более явного проявления изучаемых зависимостей. При фиксированных значениях количества кадров или количества страниц в адресном пространстве эти значения приравнивались к величинам, при которых изучаемые зависимости проявлялись в наибольшей степени (константы *StandartFrameNumber* и *StandartPageValue* соответственно).

Количество отсчетов для каждой зависимости определяется своей констаной (*PointsPage*, *PointsFrames*, *PointsMaxPageValue*).

Для каждой зависимости полученные данные автоматически записываются в папку *src* в соответствующий файл в формате, значения отделяются одним пробелом:

Значение_параметра OPT FIFO LRU CLOCK

Название алгоритма в записи выше означает полученный результат изучаемой величины для данного алгоритма при указанном значении параметра.

При запуске скрипта SciLab CompareAlgoritms.sce автоматически генерируются и сохраняются в папку *src* графики полученных зависимостей, таблицами же данных являются сами файлы с данными о работе алгоритмов (*.txt)

Статистические оценки

Из свойств изучаемых алгоритмов очевидно, что алгоритм OPT будет давать наименьшее число ошибок для любой последовательности запросов к страницам, но будет самым долгим по времени. Алгоритмы FIFO, LRU, CLOCK очень трудно сравнить между собой в плане количества страничных ошибок, так как зависимости эти не являются очевидными, в плане времени же самым быстрым является алгоритм FIFO (за счёт своей простоты).

При этом сразу можно определить, что трудоемкость для алгоритмов FIFO, LRU, CLOCK от количества запросов имеет вид O(n). Для OPT в приведенной реализации трудоемкость имеет вид $O(n^2)$.

Количество страничных ошибок для каждого алгоритма будет уменьшаться с ростом количества кадров процесса, так как вероятность, что страница уже загружена в данном случае увеличивается.

Кроме того, очевидно, что количество страничных ошибок будет расти с увеличением количества страниц в адресном пространстве.

Визуализация полученных результатов

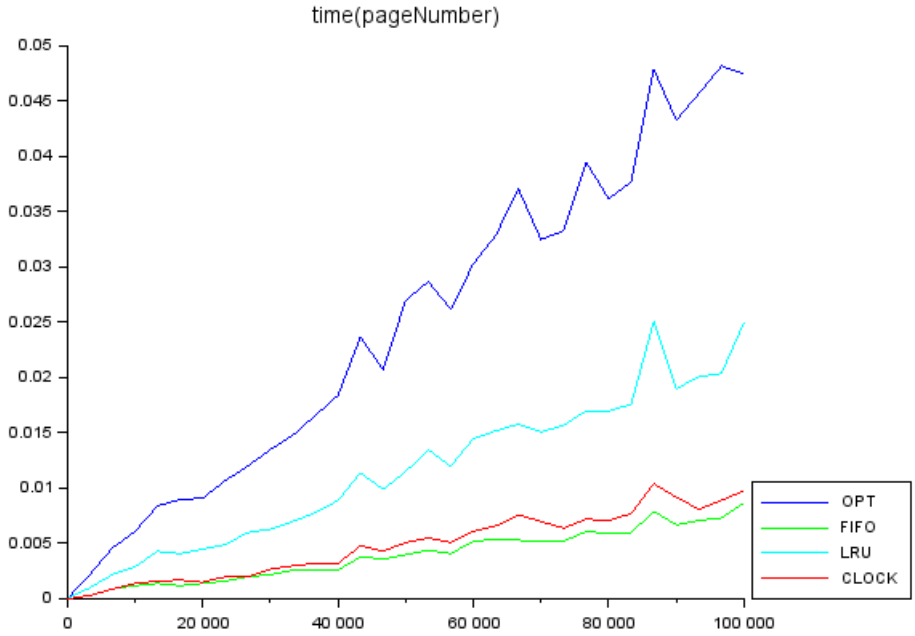

Рисунок 1. Зависимость времени работы программы от количества запросов к страницам; горизонтальная ось — количество страниц, вертикальная — время выполнения алгоритма (в сек.)

Из данного графика видно, что алгоритм OPT является самым медленным, а FIFO – самым быстрым. Скачки на графике могут объясняться влиянием сторонних процессов на производительность устройства, а также влиянием кэша на скорость обработки.

По данным из файла Time_PageNumber.txt видно, что при 99991 запросе к страницам время работы алгоритмов OPT, FIFO, LRU и CLOCK составляет 0.0475, 0.0087, 0.0250 и 0.0098 секунд соответственно. Таким образом, OPT медленнее FIFO в 5,5 раз; LRU медленнее FIFO в 2,8 раз, а скорость работы CLOCK больше FIFO на 12%.

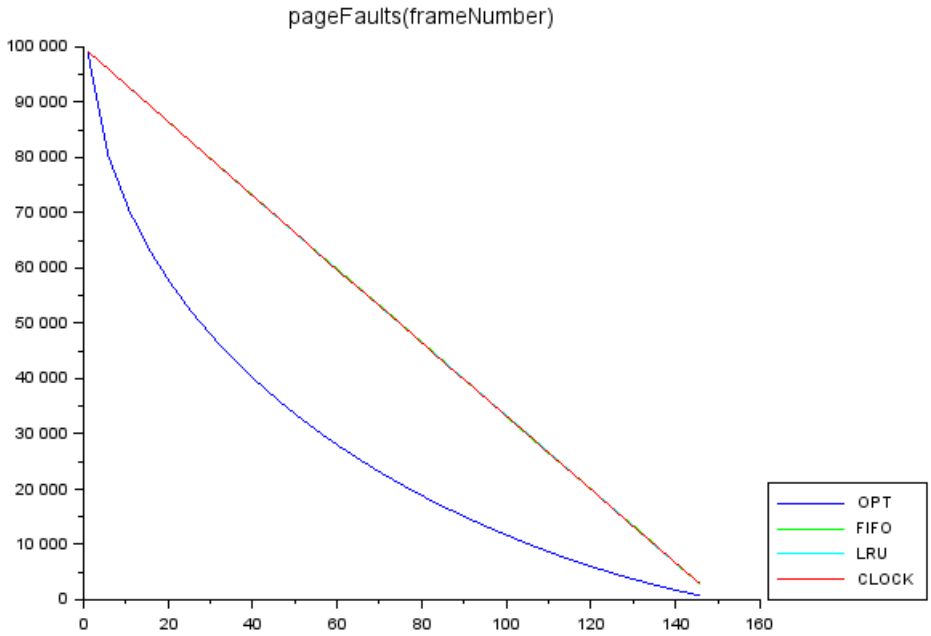

Рисунок 2. Зависимость количества страничных ошибок от количества кадров процесса

Из данного графика видно, что алгоритм OPT дает наименьшее количество ошибок, при этом значения для остальных алгоритмов настолько близки друг к другу, что сливаются в одну линию.

Все алгоритмы дают близкие результаты при малых и больщих количествах кадров (при малых — потому что вероятность нахождения новой страницы в кадрах стремится к 0, при больших — потому что вероятность стремится к 1). Для алгоритма FIFO аномалии Билэди не обнаружено.

По данным из файла Faults_Frames.txt видно, что наибольшая разница между кривыми достигается при количестве кадров, равном половине от количества страниц в адресном пространстве. При 100000 запросах к страницам, 150 страницах в адресном пространстве и 76 кадрах алгоритмы OPT, FIFO, LRU и CLOCK дают 20531, 49326, 49258 и 49270 страничных ошибок соответственно.

Таким образом результаты для алгоритмов, реализуемых на практике отличаются не более чем на 1%.

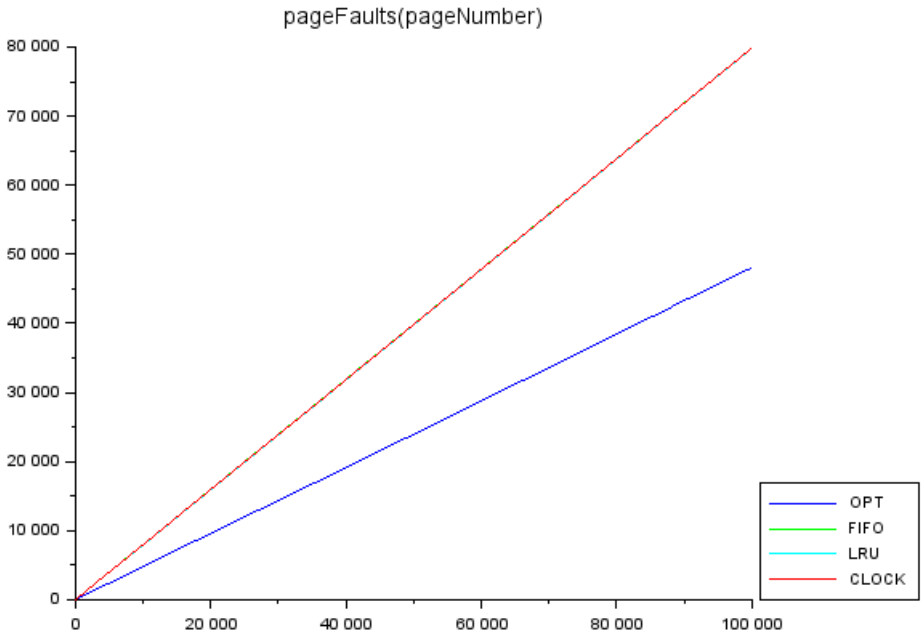

Рисунок 3. Зависимость количества страничных ошибок от количества запросов

Аналогично предыдущей зависимости, алгоритм OPT дает наименьшее количество ошибок, значения остальных алгоритмов близки друг к другу, сливаясь в одну линию.

Из файла Faults_PageNumber.txt получаем, что при 99991 запросе к страницам алгоритмы OPT, FIFO, LRU и CLOCK дают 48094, 79893, 79858 и 79880 страничных ошибок соответственно. Таким образом, OPT эффективнее остальных алгоритмов на 60%; расхождения между остальными алгоритмами составляют менее 1%.

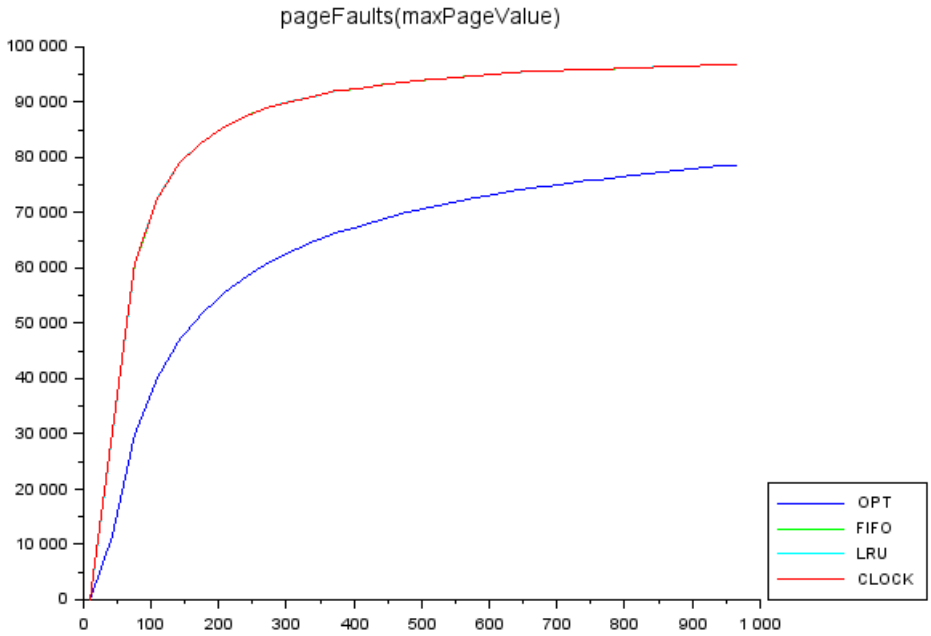

Рисунок 4. Зависимость времени работы программы от количества запросов к страницам; горизонтальная ось — количество страниц, вертикальная — время выполнения алгоритма (в сек.)

Аналогично предыдущим зависимостям, алгоритм OPT наиболее эффективен по количеству страничных ошибок, результаты работы остальных алгоритмов крайне близки друг к другу. Зависимость носит логарифмический характер.

По данным из файла Faults_MaxPageValue.txt видно, что при 100000 запросах к страницам, 30 кадрах и 967 страницах в адресном пространстве алгоритмы OPT, FIFO, LRU и CLOCK дают 78662, 96865, 96863 и 96864 страничных ошибок соответственно. Расхождения между алгоритмами, реализуемыми на практике, составляют менее 1%.

Заключение по итогам сравнения

Из полученных зависимостей можно сделать вывод, что алгоритм OPT дает наименьшее количество ошибок при изменении любых параметров, но имеет наименьшую скорость работы (медленнее FIFO более чем в 5,5 раз). Значения, полученные для остальных алгоритмов отличаются менее чем на 1%.

Таким образом при больших данных и случайном наборе запросов к страницам наиболее эффективными из рассматриваемых в работе и реализуемых на практике алгоритмов, учитывая, что отличие составляет не более 1%, являются **FIFO** (если важна производительность) и **CLOCK** (если важно и время работы, и количество страничных ошибок).