

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра "Программное обеспечение вычислительной техники и
автоматизированных систем"

МОДЕЛИРОВАНИЕ СТРАНИЧНОЙ ВИРТУАЛЬНОЙ ПАМЯТИ И
АЛГОРИТМОВ СВОПИНГА

Методические указания к выполнению лабораторной работы №2
по дисциплине «Операционные системы»

Ростов-на-Дону, 2011 г.

Составитель: к.т.н., доц. Долгов В.В.

Моделирования страничной виртуальной памяти и алгоритмов свопинга: методические указания к выполнению лабораторной работы №2 – Ростов н/Д: Издательский центр ДГТУ, 2011. – 8 с.

В методической разработке рассматриваются принципы организации страничной виртуальной памяти, используемой в современных операционных системах, и алгоритмы свопинга (пэйджинга) при выгрузке страниц на внешние носители. Даны задания к лабораторной работе помогающие закрепить на практике полученные знания. Методические указания предназначены для студентов специальностей 230105 «Программное обеспечение вычислительной техники и автоматизированных систем» и 010503 «Математическое обеспечение и администрирование информационных систем».

Печатается по решению методической комиссии факультета «Информатика и вычислительная техника».

Рецензент: к.т.н., доц. Гранков М.В.

Научный редактор: д.т.н., проф. Нейдорф Р.А.

© Издательский центр ДГТУ, 2011

В современных персональных компьютерах для удобства организации виртуальной памяти вся оперативная (первичная) память разбивается на блоки фиксированного размера, называемые страницами. Размер страниц выбирается равным степени двойки (4096, 8192, 16384, ...). Так как объем виртуального адресного пространства больше чем объем оперативной памяти, часть страниц с данными размещаются операционной системой (ОС) в оперативной памяти, а часть – на внешнем носителе.

Для определения местоположения каждой виртуальной страницы используются специальные схема преобразования и таблицы, структура которых определяется используемым центральным процессором, например, как изображено на рис. 1.

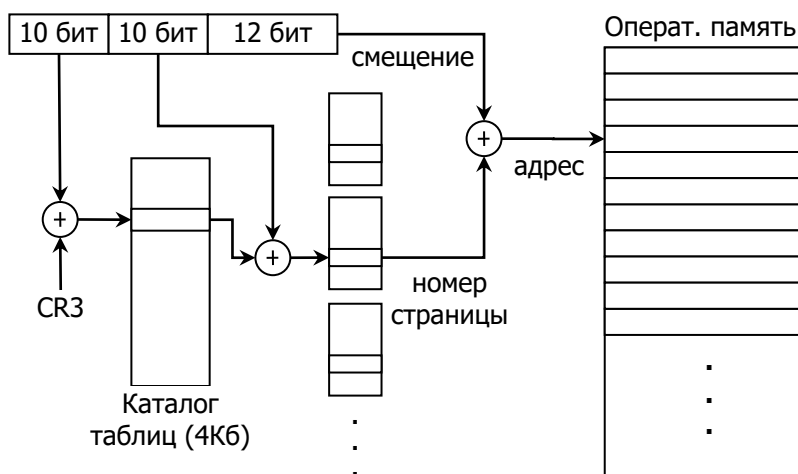


Рисунок 1. Двухзвенная схема страничного преобразования

В ходе преобразования, если процессор обнаруживает, что запрашиваемая страница находится на внешнем носителе, он инициирует аппаратное прерывание, в ходе которого ОС должна загрузить в оперативную память необходимую страницу. Чтобы освободить место под загружаемую страницу, ОС должна выбрать страницу для удаления из оперативной памяти. От эффективности данного алгоритма выбора напрямую зависит скорость работы ОС. Рассмотрим наиболее известные алгоритмы:

1. Не используемая в последнее время страница (NRU)

Данный алгоритм пытается удалить из памяти страницу, которая не использовалась в последнее время (как правило определяемое периодом аппаратного таймера) и трудоемкость удаления которой минимально. Для реализации подхода, ОС обычно используют специальные биты R (обращение) и M (модификация), устанавливаемые процессором на страницу в ходе соответствующих операций. Когда возникает необходимость в выгрузке страницы, ОС делит их на 4 группы (табл. 1).

Таблица 1. Группы страниц в алгоритме NRU

№ группы	Бит R	Бит M	Описание
0	0	0	Не было обращений и изменений.
1	0	1	Не было обращений, страница изменена. Данная ситуация является искусственной и создается самой ОС для отслеживания с какими страницами ведется активная работа.
2	1	0	Было обращение, страница изменена.
3	1	1	К странице обращались и она изменена.

Алгоритм выбирает для выгрузки случайную страницу из первой найденной непустой группы.

2. Первым прибыл – первым обслужен (FIFO)

Алгоритм FIFO использует для своей работы простую идею списка страниц, в котором первая страница является старейшей, т.е. попала в оперативную память раньше всех. При страничном прерывании на внешний носитель выгружается страница из начала списка, а загруженная вместо нее страница добавляется в конец списка. Т.о. из оперативной памяти всегда удаляется самая страница.

3. Алгоритм «вторая попытка»

Данный алгоритм является усовершенствованным вариантом

алгоритма FIFO у которого велик шанс выгрузить из памяти хоть и старую, но до сих пор активно используемую страницу. Чтобы избежать такой ситуации, алгоритм «вторая попытка» анализирует бит R (бит обращения) старейшей страницы. Если он равен 0, то страница тут же вытесняется из памяти и замещается новой. В противном случае бит сбрасывается, но страница не выгружается, а помещается в конец списка. Поиск идет до тех пор, пока ОС не найдет самую старую страницу к которой не было обращений. Даже если происходили обращения ко всем страницам, «вторая попытка» не заиклится, а вырождается в обычный алгоритм FIFO.

4. Не используемая дольше всего страница (LRU)

Основной идеей алгоритма является хорошо подтверждаемое практикой предположение, что страницы, к которым происходило частое обращение в недавнее время, имеют также высокую вероятность обращения в ближайшем будущем и наоборот. Следовательно, при страничном прерывании из оперативной памяти необходимо выгружать ту страницу, к которой дольше всего не было обращений.

Простейшим способом данный алгоритм реализуется при наличии у страниц аппаратно обновляемого поля, хранящего временную метку последнего обращения к странице. При таком подходе, страница, к которой дольше всего не было обращений, будет иметь минимальное значение такой метки.

5. Редко используемая страница (NFU)

Из-за отсутствия соответствующей аппаратной поддержки, реализация алгоритма LRU в чистом виде практически невозможно. Вместо этого в операционных системах часто используется приближенная программная реализация, называемая NFU (редко используемая страница). Для такой реализации необходим программный счетчик размером n бит, связанный с каждой страницей памяти и изначально равный нулю.

Во время каждого прерывания по таймеру, ОС производит побитовый сдвиг вправо каждого счетчика на один бит, а значением самого старшего $(n-1)$ -го бита (крайнего слева) становится значение бита R (бита обращения) страницы.

Понятно, что страница, к которой не обращались в течение 4-х тиков таймера, будет иметь в старших разрядах 4 нулевых бита, а ее счетчик будет иметь меньшее значение, чем у страницы, к которой не обращались в течение трех тиков.

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ

Написать программу, реализующую, согласно варианту (табл. 2), один из алгоритмов выгрузки страниц. В программе должны присутствовать два глобальных массива, один из которых олицетворяет оперативную память, а другой, – внешний носитель. Размер каждого из массивов должен быть не менее чем 256 байт. Использование других глобальных переменных в программе запрещено (то есть вся информация о местоположении страницы памяти и других ее характеристиках должна находиться в массиве «оперативной памяти»). Размер страницы для всех вариантов равен 32 байтам. Программа должна реализовывать сквозную адресацию ячеек «виртуальной памяти». Обязательными для реализации являются следующие функции:

- Чтение ячейки памяти. В результате операции чтения на экран должно быть выдано значение, хранящееся в ячейке памяти. Считать, что в начале работы программы все ячейки памяти равны нулю. В случае если страница с запрошенной ячейкой, расположена на «внешнем носителе», программа должна, используя соответствующий алгоритм, произвести обмен страниц между «оперативной памятью» и «внешним носителем», сообщив об этом пользователю.
- Запись в ячейку памяти. В результате операции, значение ячейки должно быть изменено на введенное пользователем. В случае если страница с запрошенной ячейкой, расположена на «внешнем носителе», программа должна, используя соответствующий алгоритм, произвести обмен страниц между «оперативной памятью» и «внешним носителем», сообщив об этом пользователю.
- Отображение карты распределения страниц виртуальной памяти между «оперативной памятью» и «внешним

носителем» (где и в какой позиции расположена каждая из страниц).

Таблица 2. Варианты заданий к лабораторной работе

№ вар-та	Алгоритм выгрузки страниц
1	Не использующаяся в последнее время страница (NRU)
2	Первым прибыл – первым обслужен (FIFO)
3	Алгоритм «вторая попытка»
4	Не использующаяся дольше всего страница (LRU)
5	Редко использующаяся страница (NFU)

КОНТРОЛЬНЫЕ ВОПРОСЫ К ЛАБОРАТОРНОЙ РАБОТЕ

1. Зачем нужны алгоритмы выгрузки страниц?
2. Что такое аномалия Билэди?
3. Почему для реализации страничной виртуальной памяти необходимы таблицы страниц?
4. Почему нельзя реализовать «Оптимальный алгоритм» выгрузки страниц?
5. Что дает учет бита R в алгоритме «Вторая попытка»?
6. В чем существенная проблема алгоритма «Наименее используемая страница»?

РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

1. Таненбаум Э. Современные операционные системы. 2-е изд. – СПб.: Питер, 2002. – 1040 с.: ил.
2. Дейтел Г. Введение в операционные системы: В 2-х томах. Пер. с англ. – М: Мир, 1987. – 359с.

Редактор А.А. Литвинова

ЛР № 04779 от 18.05.01.	В набор	В печать
Объем 0,5 усл.п.л., уч.-изд.л.	Офсет.	Формат 60x84/16.
Бумага тип №3.	Заказ №	Тираж 140. Цена

Издательский центр ДГТУ

Адрес университета и полиграфического предприятия:

344010, г. Ростов-на-Дону, пл. Гагарина, 1.