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

Кафедра "Программное обеспечение вычислительной техники и автоматизированных систем"

РАЗРАБОТКА ФУНКЦИЙ И МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКОГО УПРАВЛЕНИЯ ПАМЯТЬЮ ВНУТРИ ВЫДЕЛЕННОГО УЧАСТКА ПАМЯТИ

Методические указания к выполнению лабораторной работы №1
по дисциплине «Операционные системы»

Ростов-на-Дону, 2011 г.

Составители: к.т.н., доц. Долгов В.В.

Разработка функций и моделирование динамического управления памятью внутри выделенного участка памяти: методические указания – Ростов н/Д: Издательский центр ДГТУ, 2011. – 7 с.

В методической разработке рассматриваются вопросы динамического выделения участков оперативной памяти в операционных системах. Способы учета свободных и занятых участков и алгоритмы нахождения свободного участка при выделении памяти. Даны задания к лабораторным работам помогающим 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».

Рецензент: к.т.н., доц. Гранков М.В.

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

СПОСОБЫ УЧЕТА ДИНАМИЧЕСКОЙ ПАМЯТИ

В процессе перехода многозадачных операционных систем (ОС) от систем с фиксированными разделами к системам с переменными разделами перед ОС стала задача учета, управления и отслеживания изменений, происходящих при запуске и остановке процессов. Ещё больше эта проблема усугубилась, когда ОС стала предоставлять процессам возможность выделять память из «кучи», то есть в общем случае иметь раздел памяти, размер которого не был постоянен, а мог увеличиваться во время работы процесса.

Существует два способа, которыми ОС может учитывать использование памяти: списки блоков памяти и битовые карты памяти.

Списки блоков памяти – это, как правило, одно или двусвязные списки структур данных, каждая из которых содержит информацию об одном блоке (участке) памяти, определяя начальный адрес, размер участка, занят этот участок или нет и к какому процессу он относиться. Например, как это представлено на рис. 1.

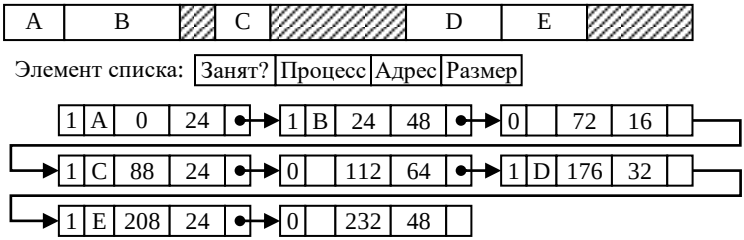

рис. 1. Пример схемы занятости оперативной памяти с соответствующей информацией в виде списка блоков

Как правило, список блоков бывает упорядочен по адресу начала блока, что упрощает объединение соседних блоков при освобождении занятого (в этом случае необходимо проанализировать только два соседних элемента списка по отношению к освобождаемому).

Битовая карта памяти. При использовании этого метода учета памяти, вся оперативная память системы условно делиться на участки равного размера (например, 64 байта), после чего

состояние занят/свободен для каждого участка может быть описано логической переменной, то есть одним битом. ОС выделяет соответствующий массив данных размера (<размер-памяти> / <размер-блока> + 7) / 8 байт, где каждый бит соответствует одному блоку. Так, для ситуации, изображенной на рис. 1., и, предполагая, что минимальный блок памяти, учитываемый битовой картой, составляет 8 байт, двоичный массив будет выглядеть как 11111111100111000000001111111000000.

Работа с подобными структурами данных требует быстрого доступа и манипулирования с одиночными битами, что в современных процессорах не представляет проблем. Однако, несмотря на отсутствие проблем со скоростью доступа и возможностью почти мгновенного определения занятости того или иного адреса памяти, такой способ учета обладает и своими недостатками, как, например, невозможность определить к какому процессу относиться адрес памяти на основе только лишь битовой карты. Для хранения необходимой информации в этом случае приходиться использовать дополнительные списки внутри ОС и специальные блоки данных, хранящиеся перед блоками выделенными процессам (рис. 2).

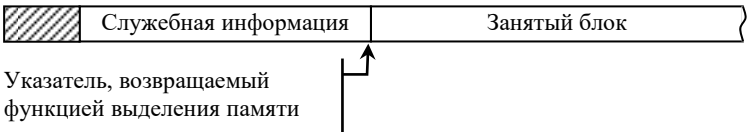

рис. 2. Хранение служебной информации перед занятыми участками (без соблюдения масштаба участков)

АЛГОРИТМЫ ВЫДЕЛЕНИЯ ПАМЯТИ

Когда к менеджеру памяти поступает запрос на выделение участка памяти определенного размера, он должен выяснить, существует ли участок, подходящий для выполнения запроса, и если таких участков несколько, то какой именно из них будет использован.

Первый и самый простой в реализации алгоритм заключается в том, что менеджер ищет в списке первый подходящий участок, то есть участок, размер которого больше чем запрашиваемый размер. В случае учета памяти с помощью списка блоков это

приводит к прямому просмотру списка в поисках нужного участка. Как только подходящий участок найден, он разбивается на две части: первая отдается процессу выдавшему запрос, вторая – остается неиспользуемой (свободной). Для ситуации с рис. 1 запрос на выделение участка размером в 24 байта процессом «F» приведет к изменению списка блоков как представлено на рис. 3.

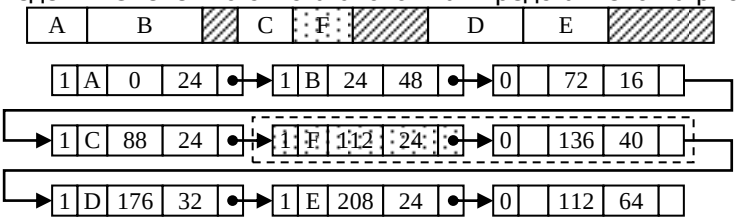

рис. 3. Изменение состояния памяти и списка блоков при выделении 24 байт по алгоритму «первый подходящий» (пунктиром обозначен свободный блок из которого происходило выделение участка)

Для случая учета памяти с использованием битовой карты отличие алгоритма заключается в необходимости рассчитывать размер свободных участков по формуле <количество-нулевых-битов-последовательности>*<размер-блока> и необходимости изменения битов, соответствующих занятой области, вместо оперирования с динамическим списком.

Два других алгоритма поиска участка называются соответственно «наиболее подходящий» и «наименее подходящий». Алгоритм выбора наиболее подходящего участка исходит из предположения, что если использовать под выделение памяти участок наиболее близко подходящий под требуемые размеры, то остаток неиспользуемой памяти от такой операции будет, очевидно, минимальным. Таким образом этот алгоритм пытается сохранить участки большого размера на потом, предполагая что они могут понадобиться в дальнейшем. Алгоритм наименее подходящего наоборот использует самый большой свободный блок из имеющихся, предполагая, что большой остаток от операции можно будет использовать и в дальнейшем. Для реализации этих алгоритмов с использованием списка блоков последний можно сортировать не по начальному адресу блока, а по размеру участка, используя сортировку по возрастанию для алгоритма «наиболее подходящий» и по убыванию для

«наименее подходящего». В этом случае для операции выделения не придется просматривать весь список целиком, однако платой за это станет «неудобство» оперирования со списком при освобождении занятого блока.

Ещё один алгоритм использует идею двоичного разбиения участков для более быстрого поиска свободного места. Этот алгоритм поддерживает несколько списков свободных блоков (список занятых процессами блоков храниться отдельно). Каждый список в такой схеме хранит свободные блоки одного и того же размера причем каждый последующий список содержит блоки в два раза большие блоков предыдущего списка. Такое хранение информации позволяет очень быстро искать необходимый свободный блок в условии, что он есть. Если же список с блоками необходимого размера пуст, система вынуждена рекурсивно запрашивать свободный участок в следующем списке, делить его на две равные части, одну из которых выделять процессу, а другую добавлять в список.

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ

Написать программу, моделирующую динамическое распределение памяти в операционной системе. В качестве модели оперативной памяти программа должна использовать байтовый массив размера не менее 256 байт. Использование других глобальных переменных в программе запрещено (то есть вся информация о свободных/занятых участках должна храниться внутри массива). В программе в обязательном порядке должны присутствовать следующие функции:

а) Выделить участок заданного размера. В случае успеха вывести начальный адрес выделенного участка. Если участка подходящего для выделения не найдено, необходимо вывести диагностическое сообщение о нехватке памяти.

б) Освободить ранее выделенный участок. В качестве параметра функция должна принимать начальный адрес освобождаемого участка. Ранее выделенный участок может быть освобожден только целиком (освобождение части участка не допускается).

в) Получить информацию о свободных/занятых участках в «оперативной памяти» (количество участков каждого типа,

начальные адреса, размеры, общее количество занятой и свободной памяти).

Варианты заданий комбинируются из возможных способов хранения информации о свободных занятых блоках и различных алгоритмов, применяемых при выделении участка. Соответствие варианта задания и указанных параметров представлено в таблице 1.

Таблица 1. Варианты заданий к лабораторной работе

Вариант задания	Алгоритм выделения	Способ хранения информации
1	Первый подходящий	Битовая карта
2	Наиболее подходящий	Битовая карта
3	Наименее подходящий	Битовая карта
4	Двоичного разбиения	Битовая карта
5	Первый подходящий	Список блоков
6	Наиболее подходящий	Список блоков
7	Наименее подходящий	Список блоков
8	Двоичного разбиения	Список блоков

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 75. Цена

Издательский центр ДГТУ
Адрес университета и полиграфического предприятия:
344010, г. Ростов-на-Дону, пл. Гагарина, 1.