

РАСПРЕДЕЛЕНИЕ ПАМЯТИ В НЕОГРАНИЧЕННОМ ЛИНЕЙНОМ АДРЕСНОМ ПРОСТРАНСТВЕ

Ивашенко В. П.

Кафедра интеллектуальных информационных технологий, Факультет информационных технологий и управления, Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: ivashenko@bsuir.by

В статье рассматривается подход к построению и реализации алгоритмов распределения неограниченно линейно адресуемой памяти для универсальных моделей решения задач

ВВЕДЕНИЕ

В технических системах, ориентированных на решение задач [1], особый интерес представляют универсальные модели [2], способные обеспечить интеграцию различных моделей решения задач [3]. В соответствии со схемой уровней управления [4] для систем управляемых знаниями, относящихся к системам ориентированным на решение задач, одной из ключевых задач управления устройствами является задача распределения памяти [5]. Ранее были рассмотрены программная модель и алгоритмы, обеспечивающие близкое к оптимальному решение этой задачи для ограниченного, конечного линейного адресного пространства (ЛАП) [6]. Однако, решение задач в рамках универсальных моделей требует эффективных механизмов работы с неограниченной памятью [7], при этом аналогичные задачи работы с объектами неограниченного размера актуальны для уровней управления данными и знаниями. Для реализации этих механизмов необходимо дать ответ на вопросы: «Какую платформу выбрать?», «Как кодировать участки памяти?», «Какие структуры следует использовать для доступа к участкам памяти, чтобы обеспечить их поиск и поддержать их изменение?», «Какова стратегия распределения памяти?».

1. ПРЕДСТАВЛЕНИЕ СВОБОДНЫХ УЧАСТКОВ НЕОГРАНИЧЕННОЙ ПАМЯТИ

Рассмотрим ответ на поставленные вопросы, исходя из известных свойств реализованных моделей распределения памяти для ограниченного ЛАП. Основными из этих свойств являются следующие:

1. стратегия «первый подходящий» является одной из близких к оптимальным стратегиям с точки зрения минимизации требований на размер памяти, включающей используемую для хранения данных и подверженную внешней фрагментации;

2. для стратегии «первый подходящий» временная сложность реализованных алгоритмов моделей распределения памяти оценивается как полилоглинейная;

3. для хранения структур, описывающих свободные участки памяти, используются сами

эти участки и дополнительный участок памяти ограниченного (неизменного в соответствии с размером адресуемой памяти) размера.

Предлагаемое представление свободных участков памяти ориентировано на сохранение этих свойств. С целью минимизации затрат памяти на дополнительные структуры данных, предлагаемое представление так же отталкивается от моделей наиболее экономичных в этом плане, а именно – от «систем близнецов» [5]. Недостатком «систем близнецов» является увеличение затрат (до двух раз) на размер памяти подверженной внутренней фрагментации.

Так как память имеет неограниченный размер, то и адреса ячеек памяти требуют неограниченного количества разрядов. С целью минимизации затрат на адаптацию используемых структур данных к возрастанию значащих разрядов адреса, для их хранения используются отдельные ячейки-регистры. Именно поэтому в качестве платформы реализации рассмотрена ранее предложенная модель обработки информации, которая как раз ориентирована на обеспечение решения подобных задач [7].

Структура свободных участков памяти, как это характерно для «систем близнецов» организуется в виде дерева, каждый полностью или частично свободный блок имеет размер равный суммарному размеру $2^k - 1$ ячеек памяти (рис. 2). Центральная ячейка любого полностью или частично свободного блока хранит служебную информацию, описывающего состояние блока и состояние ячеек блоков следующего уровня в дереве. В центральной ячейке для хранения этой информации используется до семи бит, соответствующих признакам: 1) занятости (отсутствия) младшего блока-потомка блока в дереве; 2) занятости (отсутствия) старшего блока-потомка блока в дереве; 3) блока как старшего блока-потомка у блока-родителя; 4) блока как текущего корневого блока в дереве; 5) младшего блока-потомка как блока со свободным блоком максимального размера; 6) блока, как не являющегося полностью свободным блоком, блоком-листом в дереве; 7) старшего блока-потомка как «блока-фаворита», поиск и выделение свободного блока в котором осуществляются в первую очередь,

при прочих равных. Установка значения признака «блока-фаворита», соответствует реализуемой стратегии распределения памяти. Для распределения памяти «первый подходящий» с выделением блоков в направлении только одной стороны, достаточно установить значение признака одинаковым для всех блоков в дереве. Для распределения памяти с выделением блоков с двух сторон (см. рис. 1) можно чередовать значение этого признака при подъёме от предыдущего корневого блока дерева к новому текущему корневному блоку дерева, при этом значение этого признака в новых блоках поддерева нового блока-потомка корневого блока дерева совпадает со значением этого признака в корневом блоке дерева.

II. ВЫДЕЛЕНИЕ И ВЫСВОБОЖДЕНИЕ УЧАСТКОВ НЕОГРАНИЧЕННОЙ ПАМЯТИ

Задачи выделения свободного блока и высвобождения блока памяти по указанному адресу заданных размеров в неограниченном ЛАП сведены к следующим подзадачам: 1) вычисление размера блока-родителя и блока-потомка в дереве; 2) вычисление адреса ячейки блока-родителя; 3) вычисление адреса младшего или старшего блока-потомка; 4) вычисление адреса и размера корневого блока; 5) вычисление «блока-фаворита» у указанного блока; 6) вычисление адреса подходящего по размеру блока в поддереве; 7) сравнение размеров максимальных блоков в поддеревах младшего и старшего потомков; 8) создание нового корневого блока и блока-потомка. Каждая из этих задач решается соответствующим алгоритмом, описанном в виде процедуры на языке выбранной платформы реализации. Кроме структур, описывающих состояние участков памяти, используются дополнительные регистры (в количестве не менее 22-х), которые хранят следующую информацию: адрес выполняемой команды, адреса вызовов и возвратов из процедур, параметры процедур распределения памяти – размер распределяемого блока, адрес выделенного блока, размер блока в дереве участков памяти, адрес ячейки блока в дереве участков памяти, результаты процедур, их параметры и управляющие флаги, включающие значения признаков свободного блока, старшего блока-потомка родителя, «блока-фаворита», блока максимального размера, корневого блока и т.д.

Временная сложность реализованных алгоритмов выделения и высвобождения блоков па-

мяти может быть выражена как $O(\ln^3(2^n) * f(n))$, где 2^n – количество бит распределённой памяти (количество используемых значащих разрядов адресных регистров не превышает n), $f(n)$ – время доступа к ячейке памяти по адресу. Пространственная сложность реализованных алгоритмов – $O(\ln(2^n))$. Перерасход памяти из-за внешней фрагментации, выраженный размером фактически используемой памяти: $O(m * \ln(m))$, где m – максимальный размер памяти, занятой данными.

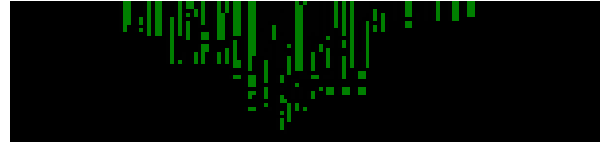


Рис. 1 – История изменений состояний памяти в неограниченном ЛАП в процессе её распределения

ЗАКЛЮЧЕНИЕ

Алгоритмы и демонстрационный пример (рис. 1) модели распределения неограниченной линейно адресуемой памяти реализованы средствами JavaScript (<https://bitbucket.org/version/openjsvmm/>).

СПИСОК ЛИТЕРАТУРЫ

1. Wolfram, S. Jeopardy, IBM, and Wolfram|Alpha. Mode of access: <http://blog.stephenwolfram.com/2011/01/jeopardy-ibm-and-wolframalpha/> Date of access: 17.09.2017.
2. Wolfram, S. A New Kind of Science. Champaign, IL: Wolfram Media, Inc. 2002. p. 1197.
3. Luger, G. Stubblefield, William (2004), Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5th ed.), The Benjamin/Cummings Publishing Company, Inc., p. 720.
4. Ивашенко, В. П. Операции управления массивами данных в линейно адресуемой памяти / В. П. Ивашенко, С. В. Синцов // Доклады БГУИР, №10, – Минск, 2016. – С. 86–93.
5. Кнут, Д. Искусство программирования. Том 3. Сортировка и поиск. – М.: Вильямс, 2014. – 832 с.
6. Ивашенко, В. П. Алгоритмы полилогарифмической временной и логарифмической пространственной сложности для системы динамического распределения линейно адресуемой памяти с однородным доступом к данным / В. П. Ивашенко // Карповские научные чтения : сб. науч. ст. – Минск, 2012. – Вып. 6, ч. 1. – С. 196–201.
7. Ивашенко, В. П. Модели обработки информации и программные средства для универсальных моделей решения задач / В. П. Ивашенко // Информационные технологии и системы 2017 (ИТС 2017) : материалы междунар. науч. конф. (Республика Беларусь, Минск, 25 октября 2017 года) / редкол. : Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2017. – С. 106–107.

...	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	...
	1	3	1	7	1	3	1	15	1	3	1	7	1	3	1		
		3		7		3		15		3		7		3			
				7				15				7					
								15									

Рис. 2 – Иерархия кодирования блоков соответствующих размеров в неограниченной памяти