

$\delta^{(r)} = 0$, а $\delta^{(l)}$ равным задержке сигнала в схеме, реализующей функцию Пирса вида (5).

Для остальных временных переключательных функций, схемно реализуемых одним логическим элементом и не рассмотренных в наших работах, определение временных координат существования сигнала можно получить применением суперпозиции уже разработанных алгоритмов.

ЛИТЕРАТУРА

1. Сиймон А., Изв. АН ЭССР, Физ. Матем., 17, 270 (1968).
2. Сиймон А., Изв. АН ЭССР, Физ. Матем., 17, 391 (1968).
3. Сиймон А., Изв. АН ЭССР, Физ. Матем., 18, 347 (1969).
4. Рабинович З. Л., Элементарные операции в вычислительных машинах, Киев, 1966.
5. Сиймон А., Изв. АН ЭССР, Физ. Матем., 19, 172 (1970).

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
2/IV 1971

EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. 20. KÕIDE
FÜSIKA * MATEMAATIKA. 1971. NR. 4

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ 20
ФИЗИКА * МАТЕМАТИКА. 1971, № 4

<https://doi.org/10.3176/phys.math.1971.4.16>

УДК 681.3.06

К. МЯРТИН

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ ТРЕХ АЛГОРИТМОВ РАСПРЕДЕЛЕНИЯ ПАМЯТИ ДЛЯ ПРОГРАММНЫХ МОДУЛЕЙ

К. MARTIN. PROGRAMM-MOODULITE JAKS KOOSTATUD KOLME MALUJAOTUSE ALGORITMI EFEKTIIVSUSE VÕRDLUS

К. MARTIN. COMPARISON OF EFFICIENCY OF THREE ALGORITHMS OF MEMORY ASSOCIATION FOR PROGRAMMING MODULES

В работе [1] приведены три алгоритма распределения оперативной памяти (ОП) для программных модулей (алгоритмы Ф1, Ф2 и Ф3) и дана краткая характеристика эффективности распределения памяти ими. Ниже приведем более точную количественную характеристику этих алгоритмов.

Отличительной чертой алгоритма Ф3 является подвижность в ОП удаляемых (т. е. использованных и оставленных в ОП) модулей. Это достигается тем, что во время загрузки модуля в ОП в конец модуля записывается информация о его адресных константах в виде списка, т. е. относительный к началу модуля адрес и признак длины. Используя

эту информацию, программой распределения ОП можно произвести перемещение модуля из одного места в другое. Ясно, что эффективность распределения памяти алгоритмом ФЗ зависит от количества адресных констант в модуле. Для выяснения того, на сколько удлиняются подвижные модули, в нескольких программах операционной системы DOS SIEMENS 4004/45 (в обслуживающих программах, макрокомандах ассемблера и вычислительных процедурах ФОРТРАНа) был определен объем памяти для адресных констант. В результате выяснилось, что удлинение в среднем не превышает 3%.

Время загрузки модуля можно выразить формулой

$$T = K_1 + K_2 \cdot l,$$

где K_1 — время нахождения модуля во внешней памяти;
 K_2 — время передачи одного байта информации в ОП;
 l — длина модуля в байтах.

Для характеристики эффективности распределения ОП измерялось время загрузки τ (при алгоритме ФЗ добавлялось еще время передвижения модулей) при пропуске конкретной программы (см. рис. 1). На рисунке приведены также длины модулей в байтах.

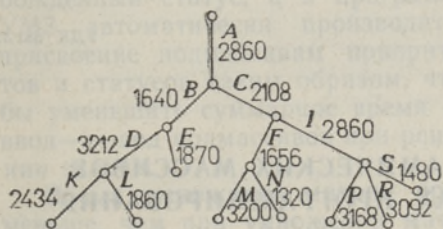


Рис. 1. Структура программы.

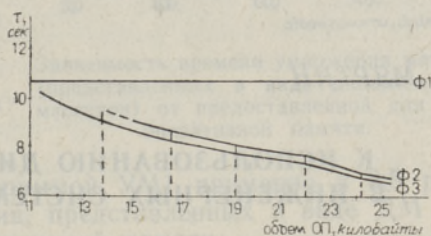


Рис. 2. Время загрузки модулей в зависимости от располагаемого объема оперативной памяти для алгоритмов Ф1, Ф2 и Ф3.

На рис. 2 представлена зависимость времени τ от предоставленного для решения данной задачи объема ОП. Приведенные для каждого алгоритма Ф1, Ф2 и Ф3 графики дают суммарное время, затраченное на загрузку модулей программы при решении с тремя разными входными данными (три разных последовательности передач управления из одного модуля в другой). Библиотека модулей находилась на дисковых устройствах внешней памяти со следующими параметрами:

среднее время позиционирования читающей головки — 0,0875 сек;
 скорость передачи информации в ОП — 156 килобайт в секунду.

Алгоритмы были запрограммированы на языке ассемблера SIEMENS 4004/45. Длины программ распределения памяти:

программа Ф1 — 896 байт;
 программа Ф2 — 652 байта;
 программа Ф3 — 968 байт.

Выводы. 1) Распределение памяти алгоритмом ФЗ является наиболее эффективным, так как он лучше остальных учитывает предоставленный для решения задачи объем памяти. Такой результат ожидался также теоретически [1], но сейчас, кроме того, показано, что длина программы ФЗ ненамного больше длины программы Ф1 — всего на 15%.

2) Минимальный необходимый для реализации алгоритма ФЗ объем памяти не более чем на 3% превышает таковой в случае алгоритма Ф1.

3) Из-за фрагментации для устойчивой работы алгоритма Ф2 требуется значительно больший минимальный объем ОП, нежели для Ф1 и Ф3 (на 20 + 30% больше).

ЛИТЕРАТУРА

1. Мартин К. О., Тамм Б. Г., Распределение памяти в инженерных системах программирования с проблемной ориентацией, Препринт-2 Ин-та кибернетики АН ЭССР, Таллин, 1971.

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
16/IV 1971

EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. 20. KÕIDE
FÜSIKA * МАТЕМАТИКА. 1971, NR. 4

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ 20
ФИЗИКА * МАТЕМАТИКА. 1971, № 4

УДК 681.3.06

К. МЯРТИН

К ИСПОЛЬЗОВАНИЮ ДИНАМИЧЕСКИХ МАССИВОВ В ИНЖЕНЕРНЫХ СИСТЕМАХ ПРОГРАММИРОВАНИЯ

K. MÄRTIN. DÜNAAMILISTE MASSIIVIDE KASUTAMISEST INSENERILISTES
PROGRAMMEERIMISSÜSTEEMIDES

K. MARTIN. ON USING DYNAMIC ARRAYS IN ENGINEERING PROGRAMMING SYSTEMS

Решение инженерных задач обычно связано с обработкой данных, объем которых может варьироваться в больших пределах. В работе [1] указано, что в инженерных системах программирования на ЭЦВМ третьего поколения целесообразно использовать динамические массивы (ДМ) для внутрисистемного представления данных. Использование ДМ является удобным для разработчика системы, так как освобождает его от трудностей распределения оперативной памяти (ОП), что необходимо производить при использовании-dimensional массивов. В то же время использование ДМ дает возможность решать задачи с таким объемом данных, решение которых в случае-dimensional массивов явилось бы невозможным (хотя время решения при использовании ДМ значительно возрастает).

В работе [1] рассмотрены средства представления данных в виде ДМ и распределение памяти для них. В данном сообщении мы охарактеризуем количественно работу с ДМ. Для этой цели рассмотрим скалярное произведение матриц $C = A \cdot B$, где матрицы A и B имеют размерность 10×10 . Элементами матриц являются целочисленные константы длиной в слово (4 байта). Четыре программы умножения матриц УМ1, УМ2, УМ3 и УМ4 написаны на языке ассемблера SIEMENS 4004/45. В программе УМ1 матрицы представляются в виде-dimensional массивов. В программах УМ2—УМ4 матрицы представлены в виде ДМ, и перед началом работы программ подмассивы находятся