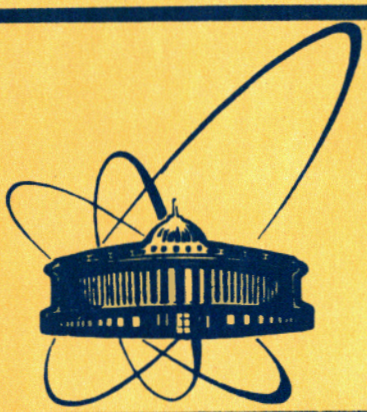

сообщения
объединенного
института
ядерных
исследований
дубна

10-84-463

Г.Балука

УСЛОВИЯ ПРИМЕНИМОСТИ ПРОГРАММНЫХ СИСТЕМ УПРАВЛЕНИЯ ПАМЯТЬЮ В СИСТЕМАХ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТА

1984

ВВЕДЕНИЕ

Современное состояние вычислительной техники, используемой при создании систем автоматизации эксперимента /САЭ/, и средств программирования[1,2] позволяет строить весьма громоздкое программное обеспечение /ПО/, необходимость которого вызвана постоянно растущими требованиями методики эксперимента. Практика такова[3,4], что при использовании ЭВМ типа СМ-3, Электроника-60 с обычным ОЗУ в 28К слов, САЭ после 2-3 этапов развития становится "большой"[5], т.е. не может быть размещена в ОЗУ полностью.

Этот процесс протекает еще быстрее, если возникает потребность использования языков высокого уровня. Наконец, когда облегчено использование ранее разработанных программ, например, путем простого включения их в библиотеку[6], возникает вопрос, включать ли в новую САЭ ранее накопленный багаж универсальных вспомогательных /тестовых, диагностических, отладочных и др./ программ? Обычно на этот вопрос отвечают положительно, что, по крайней мере на этапе создания и опытной эксплуатации системы, оправдано, однако резко увеличивает ее объем.

Наращивание объема ОЗУ сверх 28К для размещения всех включенных в САЭ программ связано с переходом к более высокому классу машин, что не всегда целесообразно, т.к. при этом стоимость САЭ возрастает, а оборудование /в частности память/ будет использоваться неэффективно[7]. В этом случае некоторое компромиссное решение может быть получено на пути сегментирования программного обеспечения САЭ и использования системы управления памятью /СУП/, при которой часть сегментов хранится в памяти ВЗУ и загружается в ОЗУ по мере необходимости.

Использование СУП в САЭ ухудшает одну из эксплуатационных характеристик САЭ - пропускную способность, однако способствует снижению стоимости САЭ за счет повышения технологичности разработки ПО и за счет возможной экономии оперативной памяти.

В данной работе рассмотрим вопрос о влиянии СУП на пропускную способность САЭ с целью определения границ применимости СУП и путей оптимизации САЭ.

1. РАСПРОСТРАНЕННЫЕ СПОСОБЫ ПРОГРАММНОГО УПРАВЛЕНИЯ ПАМЯТЬЮ

1.1. Свопинг - способ временного получения дополнительных ресурсов памяти для программ, отсутствующих в текущий момент

в ОЗУ, при котором СУП копирует определенный участок памяти на ВЗУ, загружает на это место программу и выполняет ее, а затем восстанавливает прежнее содержимое памяти, считывая образ ее с ВЗУ. Таким образом работают три фазы, которые условно можно обозначить схемой "запись1-чтение2-чтение 1". Существуют реализации этого алгоритма, которые используют для свопинга поле,занятое программами или данными, не подлежащими модификации во время работы. Тогда устраняется фаза сохранения, и алгоритм работает по схеме "чтение 2 - чтение 1".

Инициация свопинга обычно происходит без усилий со стороны пользователя, что является большим его достоинством.

К числу недостатков следует отнести существенное увеличение системных затрат /время/ и усложненное /позиционно-независимое/ кодирование, т.к. начало области свопинга может изменяться в процессе работы.

Ввиду отмеченных особенностей свопинг обычно используется для выполнения сравнительно редко требующихся внутрисистемных операций.

1.2. Оверлейный алгоритм загрузки увеличивает ресурсы памяти путем вычеркивания не используемого в данный момент сегмента и загрузки на его место нужного. Этот алгоритм эффективен,если программу можно расчленить на несколько функционально независимых сегментов примерно одинаковой длины. В такой структуре выделяется один постоянно присутствующий в ОЗУ организующий сегмент и несколько исполнительных, которые чередуются на оверлейном поле памяти. Может быть введено несколько оверлейных полей, каждое для закрепленной за ним группы сегментов.

К достоинствам такой СУП можно отнести следующие:

1. Практически любая современная система программирования предоставляет возможность строить оверлейные структуры /стандартные средства/.

2. Пользователь не управляет загрузкой сегментов.

Недостатками являются:

1. Состав программ на оверлейных полях полностью определяет пользователь. Ввиду этого небольшие изменения иногда приводят к существенному перераспределению программ между сегментами.

2. Длина оверлейного поля автоматически принимается равной длине наибольшего из сегментов, приписанных этому полю. В результате при загрузке на это поле короткого сегмента память используется неэффективно, т.к. остающийся свободным участок поля не может быть использован.

3. При наличии нескольких оверлейных полей соответственно уменьшается предельная длина сегмента.

4. Программы, используемые в циклах, приходится размещать в данном сегменте и на смежных полях либо мириться с многократным выполнением операции загрузки. В результате снова возникает проблема дефицита памяти, но на новом уровне.

Несмотря на обилие недостатков, построение оверлейной структуры следует считать универсальным и наиболее распространенным способом решения проблемы дефицита памяти для пользовательской программы. Наиболее эффективно этот алгоритм может быть использован при решении системных и вычислительных задач.

1.3. Динамическое распределение памяти /ДРП/ было введено М.Р.Шура-Бурой на ЭВМ типа М-20[8]. Этот алгоритм позволяет во время исполнения программы загружать в память ЭВМ требующиеся программные модули /ПМ/, если они в данный момент отсутствуют в ОЗУ, с использованием для этого свободного места либо высвобождением его путем вычеркивания из памяти не использующихся модулей.

Наиболее важным достоинством данного алгоритма является то, что при загрузке очередного модуля занимается лишь часть свободной памяти, равная его длине. Благодаря этому модули на поле размещаются "плотно", и их количество в ОЗУ в среднем больше, чем при использовании оверлейной организации.

Однако программные модули для того, чтобы их можно было настраивать на произвольное место загрузки, кодируются в перемещаемом формате /например,.REL в ОС РАФОС/, ввиду чего время загрузки несколько увеличивается.

Такого типа алгоритмы иногда используются на современных больших ЭВМ[9], а также на мини- и микроЭВМ[10].

1.4. Для сравнения перечисленных алгоритмов СУП основные их характеристики сведены в табл.1. Видно, что по совокупности свойств выбор должен быть выполнен между ДРП и оверлейной структурой, причем ДРП, предоставляя ряд технологических качеств, уступает по следующим показателям:

- для его применения необходимо выполнить дополнительные разработки;
- во время загрузки модуля будет расходоваться дополнительное время на настройку его по месту загрузки.

Рассмотрим на примере ДРП зависимость пропускной способности САЭ от некоторых параметров СУП и особенностей области приложения.

2. КРИТЕРИЙ ЭФФЕКТИВНОСТИ СУП

Поскольку рассматриваются большие САЭ[5], мы можем представить ПО САЭ в виде совокупности "к" программ /или модулей/.Во время работы САЭ эти модули в некоторой последовательности загружаются в ОЗУ /или разыскиваются в ОЗУ/ и выполняются. За время эксперимента i-й модуль может быть исполнен N_i раз.

Время t_i, которое САЭ расходует на выполнение i-го модуля при j-м обращении к нему, можно представить формулой

Таблица 1

Сравнение алгоритмов программного управления памятью

Характеристика	Свопинг	Оверлейн. стр.	Дин. распр. памяти
Основная область применения.	Операционные системы общего назначения.	Счетные задачи, системы управления, операционные системы.	Системные задачи, системы управления.
Особенности способа использования программы или их компонентов.	Недостаток: не допускается свопинг областей программы, выполняющих операции обмена.	Недостаток: не допускается прямое взаимодействие сегментов, приписанных одному и тому же оверлейному полю, что ведет к возрастанию числа модулей и дополнительным ограничениям макс. длины сегмента.	Прямое взаимодействие модулей разрешено, однако пользователь в этом случае инициирует загрузку нужного модуля.
Характерные особенности способа построения программы.	Раздельная трансляция и раздельное редактирование связей.	Раздельная трансляция. Совместное редактирование связей всех компонентов программы.	I. Возможна раздельная трансляция и редактирование связей для каждого отдельно взятого модуля. 2. Организующая программа может быть транслирована и редактированы связи отдельно от пакета модулей.
Зависимость текста временного загружаемого сегмента от остальных компонентов программы.	Зависимость отсутствует.	Имеется, т.к. используется стандартный редактор связи при сборке программы.	Наличие обусловливается реализацией. Может отсутствовать такая зависимость.
Изменения программы при изменении характеристик временно загружаемых сегментов.	Изменения не требуются.	Обязательно повторить редактирование связей. Может потребоваться перераспределение памяти между полями и программ между сегментами.	Изменения не требуются.
Затраты при расширении набора модулей.	Не требует никаких изменений - программы пользователя.	Как и при модификации одного сегмента (см. предыдущий пункт).	Новые модули включаются в библиотеку, либо регистрируются в файловой системе.

Таблица 1 /продолжение/

Участие пользователя при распределении памяти между сегментами.	Не требуется, т.к. используется единственное поле.	Требуется во всем процессе создания программы.	Не требуется.
Выбор места загрузки сегмента.	Автоматически, но пользователь должен учитывать место размещения области свопинга выгружаемой области, иногда может определить начало области свопинга.	Фиксируется системой программирования при редактировании связей между сегментами.	Происходит динамически в пределах области, границы которой может определять пользователь.
Выбор длины временно загружаемого сегмента.	Ограничений нет. Доступно полное адресное пространство, отданное пользовательской программе. Доступна динамика.	Зависит от числа оверлейных полей.	Ограничений нет. Модуль может занять всю доступную СУП память.
Формат представления программы перед загрузкой.	Позиционно-независимое кодирование.	Копия образа памяти.	Перемещаемые программы, настраиваемые по месту загрузки.

$$t_i = t_{i\,раб} + \delta_{ij} \cdot t_{i\,дост} \qquad /1/$$

где: $t_{i\,раб}$ - время передачи управления модулю в сумме с временем его работы; $t_{i\,дост}$ - время поиска модуля на ВЗУ, загрузки его из ВЗУ в ОЗУ и настройки по месту загрузки; δ_{ij} - коэффициент, равный 0, если модуль находится в ОЗУ, или 1, если модуль нужно загрузить из ВЗУ.

Первый член этой формулы учитывает полезное время /время счета/, а второй - время потерь на работу СУП.

Определим эффективность работы СУП в конкретной САЭ как отношение полезного времени ко всему затраченному

$$\eta = \frac{\sum_{i=1}^{k} \sum_{j=1}^{N_i} t_{i\,раб}}{\sum_{i=1}^{k} \sum_{j=1}^{N_i} (t_{i\,раб} + \delta_{ij} \cdot t_{i\,дост})} . \qquad /2/$$

Поскольку величина

$$P_{i\,обн} = 1 - \lim_{N_i \to \infty} \sum_{j=1}^{N_i} \frac{\delta_{ij}}{N_i}$$

дает вероятность обнаружить i-ю программу в ОЗУ, формулу /2/ можно записать в виде

$$\eta = \frac{\sum_{i=1}^{k} t_{i\,раб}}{\sum_{i=1}^{k} [t_{i\,раб} + (1 - P_{i\,обн}) \cdot t_{i\,дост}]} \quad . \qquad /3/$$

Видно, что эффективность использования СУП увеличивается, если:

1/ увеличивать вероятность обнаружить программный модуль в ОЗУ при очередном обращении к нему;

2/ уменьшать время загрузки $t_{i\,дост}$;

3/ увеличивать соотношение $t_{i\,раб}/t_{i\,дост}$.

Рассмотрим реальные значения параметров, влияющих на эффективность СУП.

3. ЗНАЧЕНИЯ ПАРАМЕТРОВ, ВЛИЯЮЩИХ НА ЭФФЕКТИВНОСТЬ СУП

3.1. Средняя длина программного модуля совместно с длиной рабочего поля /РП/ ДРП определяет вероятное количество модулей на ДРП. Длина РП в одной из наиболее развитых САЭ, эксплуатируемых в ЛНФ ОИЯИ[4], составляет 15226 слов. У других систем /например,[11,12]/ эта величина больше.

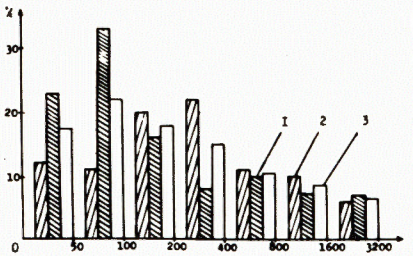

Рис.1. Распределение длин программных модулей: 1 - в системе [4]; 2 - библиотеки системы [13]; 3 - суммарное распределение.

Для оценки средней длины программного модуля воспользуемся данными о 100 модулях из системы МУР[4] и 100 модулях из нескольких систем, созданных в ИАиЭ[13]. На рис.1 гистограмма 1 соответствует данным о программах ЛНФ, 2 - данным ИАиЭ, 3 - суммарная гистограмма. Средняя взвешенная длина программного модуля для[4] составляет 341 слово, максимальная длина ℓ_{max} = 3044 слов.

3.2. Измеренная величина времени загрузки ПМ для различных дисководов и способов разметки дисков для ОС РАФОС приведена в табл.2 вместе с другими данными, позволяющими выполнить оценки $t_{i\,дост}$ при различных алгоритмах загрузки.

Эти данные показывают, что время загрузки модуля при использовании указанных дисководов может различаться в 10÷20 раз. Строки 4,6 соответствуют случаю, когда ПМ является файлом, а строки 5,7 соответствуют библиотечной организации[14] способа

Таблица 2

Времена выполнения дисковых операций

Операция	Тип диска					
	СМ 5401	СМ 5608 (гибкий диск)				
Разметка	—	Нум. блоков через 1			Нум. подряд	
ЭВМ	СМ-3	М60-30	М60-55	ДВК	М60-30	М60-55
Открытие файла	0,13-0,31	3,7-6	3,7-6	3,7-6	3,7-6	2,5-4
Чтение 1 блока	0,005	0,12	0,12	0,12	0,64	0,07
Поиск и чтение 1 блока	0,02	0,175	0,175	0,175	0,7	0,175
Загрузка короткого модуля	0,2-0,4 0,08 *	4,5 0,7 *	4,5 0,7 *	4,5 0,7 *	20 2,8 *	3,3 0,7 *
Загрузка длинного модуля	0,25-0,45 0,125 *	5,6 1,71 *	5,6 1,71 *	5,6 1,71 *	26 8,6 *	3,9 1,33 *

Данные, помеченные "*", относятся к загрузке модуля из библиотеки, остальные - из отдельного файла.

хранения модуля. Видно, что совершенствование способа доступа к модулям позволяет еще в 3÷7 раз сократить время доступа. Совместно эти два способа позволяют ценой применения более дорогого устройства /дисковода/ и более сложной организации ПО примерно в 100 раз сократить время доступа к программам, резидентным в дисковой памяти.

3.3. Вероятность обнаружения программного модуля в ОЗУ при очередном обращении к нему зависит от последовательности вызова модулей, т.е. от алгоритма эксперимента. Для каждой конкретной САЭ эта последовательность может определяться характеристиками потока информации и текущим состоянием САЭ. Оценка, полученная в предположении случайной последовательности вызовов модулей, даст нижнюю границу этой вероятности.

В табл.3 показаны условия и результаты двух расчетов вероятности $P_{обн}$ обнаружения ПМ в ОЗУ на модели со случайной последовательностью обращений к ПМ и алгоритмам СУП, описанным в работе[10].

Для этих расчетов конкретные данные о совокупности ПМ и ресурсах памяти взяты из работы[4]. Вероятность обнаружения ПМ в ОЗУ вычислялась по формуле $P_{обн} = f(m) + [1 - f(m)] * P_{ДРП}$, где

Таблица 3

Вариант расчета	Без фикс.	С фикс.
Суммарная длина поля памяти L	I0000	I0000
Длина поля для фиксированных программ L_ϕ	0	8400
Длина поля ДРП, $L_{ДРП}$	I0000	I600
Суммарное количество прогр. модулей k	60	60
Количество фиксируемых модулей k_ϕ	0	44
Количество не фиксируемых модулей	60	I6
Суммарная длина всех модулей $L_{тот}$	22800	22800
Средняя длина модуля (для ДРП) l	330	900
Максимальная длина модуля l_{max}	I600	I600
Отношение $L/L_{тот}$	0,438	0,438
Отношение k_ϕ/k	0	0,733
Вероятность $P_{обн}$	0,39I	0,390

$f(m) = m/k$ - вероятность обращения к ПМ, фиксированному в ОЗУ; m - число ПМ, фиксированных в ОЗУ; $P_{ДРП}$ - вероятность обнаружить ПМ на поле ДРП.

Видно, что фиксация максимально возможного числа ПМ в ОЗУ приводит к увеличению $P_{обн}$ примерно в 3 раза.

Этот эффект может быть существенно усилен при детальном учете особенностей алгоритма конкретной САЭ, например, путем использования измерений или оценок вероятности обращения к данному ПМ при решении вопроса о фиксации его в ОЗУ. Тогда максимум вероятности $P_{обн}$ будет соответствовать максимальному значению $f(m)$ из системы

$$\begin{cases} f(m) = \sum_{i=1}^{m} P_i ; \\ \sum_{i=1}^{m} \ell_i \le L_{ДРП} - \ell_{max}, \end{cases} \qquad /4/$$

где $P_i > 1/k$ - вероятность обращения к i-му ПМ в данной САЭ; ℓ_i - длина этого ПМ.

Формально на основании этих расчетов можно было бы ожидать вывода о том, что тактика распределения памяти между СУП и фиксированной конфигурацией модулей может быть принята следующая:

1. Выделить СУП поле памяти, длина которого равна длине наибольшего ПМ ℓ_{max}.

2. Всю остальную память использовать для фиксации с целью получения наибольшего значения $f(m)$ и тем самым - $P_{обн}$.

Однако столь детальный учет особенностей конкретной САЭ требует дополнительных затрат на этапе проектирования САЭ и не всегда возможен ввиду того, что:

1. На этапе проектирования САЭ не всегда имеются достаточные данные о характеристиках регистрируемого потока информации.

2. В процессе эксперимента могут быть выдвинуты требования к изменению методики; в результате проектные оценки перестанут отражать реальную ситуацию.

Возможным способом решения этих трудностей могло бы быть включение в СУП алгоритма, накапливающего данные о P_i для отдельных ПМ, решающего систему /4/ и с некоторой периодичностью выполняющего изменение состава ПМ, фиксированных в ОЗУ.

Однако автору представляется более перспективным путь улучшения характеристик СУП посредством введения ассоциативных свойств в алгоритм динамической компоновки состава модулей на РП СУП, например, построение кольцевой организации РП.

Выполненное рассмотрение дает конкретную информацию о характеристиках СУП и способах их оптимизации, если вопрос о применении СУП уже решен положительно.

Рассмотрим подход к решению вопроса о допустимости применения СУП в конкретной САЭ.

4. КРИТЕРИЙ ПРИМЕНИМОСТИ СУП В САЭ

Представим САЭ в виде совокупности вложенных процессов, манипулирующих данными /рис.2/. Процесс p^i /уровень i/ может включать n^i операций op^i_1, op^i_2, ..., $op^i_{n^i}$, исполняемых программами или оборудованием. Операциями манипулирования данными будем считать операции ввода данных, структурного преобразования, чтения и записи на носители, изменения условий эксперимента, управления

состоянием системы и т.п. Одну из таких операций op_1^i процесса p^i выполняет процесс p^{i-1}.

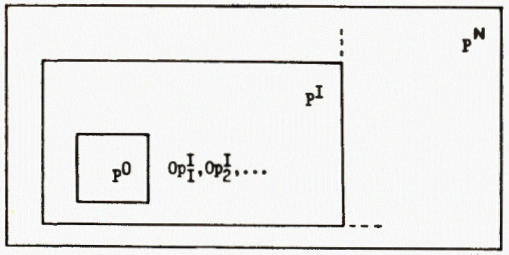

Рис.2. Схема САЭ.
Op_k^i - операции
p^i - процессы;
манипулирования данными.

Влияние процесса p^i на пропускную способность САЭ может быть представлено коэффициентом

$$\eta_i = \frac{\sum_{j=1}^{n^i} t^i_{j\,раб}}{\sum_{j=1}^{n^i} (t^i_{j\,раб} + t^i_{j\,дост})}. \qquad /5/$$

Рассмотрим процессы, начиная с $p^\emptyset$, и найдем некоторый граничный p^g такой, для которого суммарное время доступа к программам мало по сравнению с временем работы программ, т.е.

$$\sum_{j=1}^{n^g} t^g_{j\,раб} \approx r^g \sum_{j=1}^{n^g} t^g_{j\,дост}, \qquad /6/$$

где $r^g >> 1$.

Процесс, для которого выполняется условие /6/, назовем согласованным процессом, имея в виду, что его характеристики согласуются с использованием СУП без существенной потери пропускной способности САЭ.

Процессы, включенные в граничный, т.е. процессы, для которых не выполняется условие /6/, будем называть "несогласованными" процессами.

Рассмотрим влияние использования СУП в согласованных и несогласованных процессах на пропускную способность САЭ.

4.1. Согласованные процессы. Условие /6/ выполняется весьма часто. Во многих случаях наличие внутри процесса одной или нескольких медленных /длительных/ операций может привести к тому, что данный процесс и все охватывающие его становятся согласованными. В качестве примеров таких операций можно указать аппаратно выполняемые операции гистограммирования, занимающие десятки минут[15], накопления в буферных ЗУ и записи на МЛ последовательности описаний событий /десятки секунд/[11], управления электромеханическими устройствами перемещения или переориентации образцов /секунды, минуты/[3,4] и др. Представление о времена доступа к программам через СУП может быть получено по данным из табл.2, а о длинах ПМ - из рис.1, где приведено распределение программных модулей, полученное по 100 модулям системы МУР[4] /средняя длина модуля 341 слово/. На основании данных из табл.2 и рис.1 можно сделать вывод, что для часто используемых ВЗУ и для перечисленных медленных операций коэффициент r^g имеет значения 10÷100 и более.

Процессы предварительной обработки данных обычно выполняют сжатие этих данных и повторяются циклически, что увеличивает роль присутствующих в них медленных операций для охватывающих процессов. Наличие в охватывающем процессе значительного числа быстрых операций может ослабить такое влияние, и, чтобы этого не произошло, целесообразно, если возможно, сгруппировать быстрые операции в одном загрузочном модуле либо вынести их из состава часто повторяющихся циклов. Используя /6/, выражение /5/ можем записать в виде

$$\eta^g = \frac{r^g}{1 + r^g}, \qquad /7/$$

где r^g - среднее отношение времени работы программ ко времени их загрузки.

При достаточно больших r^g получим $\eta^g \approx 1$, т.е. использование СУП в согласованных процессах мало влияет на пропускную способность САЭ.

4.2. Несогласованные процессы. Для процессов $p^\emptyset, \ldots, p^{g-1}$, включаемых в граничный p^g, в случае $\Sigma t_{дост} > \Sigma t_{раб}$ использование СУП для включения этих операций может привести к существенному снижению пропускной способности САЭ.

Условием применимости СУП является достаточная емкость ОЗУ /либо поля СУП/ для одновременного размещения всех программ, исполняемых в процессах $p^\emptyset, \ldots, p^{g-1}$.

Это условие часто может быть выполнено. Так, в системах[3,4,11,12] необходимый ресурс памяти для размещения программ несогласованных процессов составлял <4К слов, причем при разработке этих программ не возникало требования минимизировать их длину.

4.3. Граничный процесс. Положение граничного процесса в структуре программного обеспечения конкретной САЭ определяет суммарный объем программ, которые необходимо разместить в ОЗУ одновременно. Влиять на эту характеристику, не меняя самих программ, можно путем изменения $t_{дост}$.

Выше было показано, что, варьируя способы хранения загрузочных модулей и типы носителей, можно на два порядка изменить значение r^g в соотношении /6/. Оптимизация механизма СУП может дать дополнительное увеличение отношения r.

ВЫВОДЫ

При построении программного обеспечения больших[5] САЭ в условиях дефицита оперативной памяти использование программных СУП позволяет решить проблему дефицита памяти без перехода к ЭВМ более высокого класса. Однако использование СУП без учета ее свойств и характеристик САЭ может привести к снижению пропускной способности САЭ.

В данной работе предложено в качестве критерия применимости СУП использовать влияние ее на пропускную способность САЭ. Для рассмотрения вопроса о применимости СУП введено понятие граничного процесса в иерархической структуре программного обеспечения САЭ как процесса, начиная с которого время доступа к программам, включенным в граничный и в охватывающие процессы, оказывает малое влияние на пропускную способность САЭ. Сформулировано условие применимости СУП. Необходимым условием применимости СУП следует считать возможность разместить на рабочем поле памяти СУП все программы, включенные в процессы, охватываемые граничным.

В заключение автор, пользуясь случаем, благодарит И.М.Саламатина за руководство и помощь в работе, а также Е.П.Козлову и Г.Я.Яновского за предоставленные материалы[4,13].

ЛИТЕРАТУРА

1. Наумов Б.Н. Приборы и системы управления, 1977, №10, с.5-7.
2. Силинов Е.М., Семик В.П. Приборы и системы управления, 1977, №10, с.15-17.
3. Вагов В.А. и др. ОИЯИ, Р10-80-826, Дубна, 1980.
4. Вагов В.А. и др. ОИЯИ, Р14-83-898, Дубна, 1983.
5. Балука Г. ОИЯИ, Р10-83-761, Дубна, 1983.
6. Балука Г. и др. ОИЯИ, Р10-12960, Дубна, 1980.
7. Дади Л. и др. Программирование, 1978, №2, с.26-32.
8. Шура-Бура М.Р. Интерпретирующая система для М-20. ВЦ АН СССР, М., 1965.
9. Мазный Г.Л. Программирование на БЭСМ-6. "Наука", М., 1978.
10. Балука Г., Островной А.И. ОИЯИ, Р10-13004, Дубна, 1980.
11. Бакалов Т. и др. ОИЯИ, 10-82-522, Дубна, 1982.
12. Гриднев Г.Ф., Саламатина Т.С. ОИЯИ, 10-83-598, Дубна, 1983.
13. Саламатин И.М., Штарк М.В., Яновский Г.Я. Автометрия, 1981, №4, с.60-69.
14. Балука Г., Саламатин И.М., Хрыкин А.С. ОИЯИ, 10-12545, Дубна, 1979.
15. Балука Г. и др. ОИЯИ, 13-82-367, Дубна, 1982.

Рукопись поступила в издательский отдел
2 августа 1984 года.

Балука Г. 10-84-463
Условия применимости программных систем управления памятью в системах автоматизации эксперимента

При построении программного обеспечения больших систем автоматизации экспериментов использование программных систем управления памятью /СУП/ часто позволяет решить проблему дефицита оперативной памяти без замены ЭВМ на машину более высокого класса. В работе рассмотрен вопрос о применимости СУП в системах автоматизации экспериментов /САЭ/. Предложено в качестве критерия применимости СУП использовать влияние ее на пропускную способность САЭ. Сформулировано условие применимости СУП.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1984

Перевод Т.А.Филимонычевой

Baluka G. 10-84-463
The Conditions of Applicability of Software Memory Management Systems in Experiment Automation Systems

The application of software memory management systems (MMS) in the large systems of experimental control and data acquisition is discussed. It is proposed to use the influence of MMS on the transmission capability of such systems as a criterion of MMS applicability. The condition of MMS applicability is formulated.

The investigation has been performed at the Laboratory of Neutron Physics, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1984