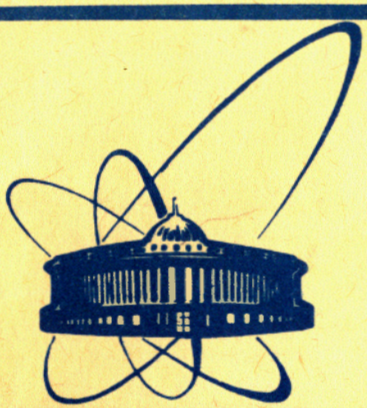

сообщения
объединенного
института
ядерных
исследований
дубна

Р10-84-789

Г.Балука

РЕАЛИЗАЦИЯ ДИНАМИЧЕСКОГО РАСПРЕДЕЛЕНИЯ ПАМЯТИ С КОЛЬЦЕВОЙ ОРГАНИЗАЦИЕЙ БУФЕРА ДЛЯ ПРОГРАММНЫХ СИСТЕМ НА ЯЗЫКЕ ПАСКАЛЬ

1984

ВВЕДЕНИЕ

Целесообразность разработки и применения механизма динамического управления составом программ в оперативной памяти, доступной процессору ЭВМ, связана в основном с двумя факторами. Первым является ограниченный объем оперативной памяти ЭВМ, вторым - ограниченные возможности адресации памяти для выборки требуемой информации. Поскольку возможность расширения адресации памяти решается в основном аппаратно, путем усложнения устройств управления памяти, что характерно для более развитых конфигураций ЭВМ, в данной работе этот вопрос не рассматривается. Проблемы, связанные с ограниченным объемом оперативной памяти, представлены в/1/. Там же указаны способы решения этих проблем. В связи с развитием технологии создания систем автоматизации эксперимента /САЭ/ появились новые требования и возможности относительно систем управления памятью.

Представленное в/1/ решение проблемы нехватки оперативной памяти заключается в расчленении исполняемой прикладной программы /или системы/ на резидентную управляющую часть и открытую библиотеку рабочих модулей /РМ/, загружаемых в память по мере надобности системой динамического распределения памяти /ДРП/. Система ДРП в случае нехватки в ОЗУ ЭВМ поля для загрузки требуемой для работы САЭ программы, по определенному алгоритму /зависящему от реализации системы ДРП/ освобождает участок поля, зачеркивая ненужную в данный момент программу /или программы/, и на это место помещает заказанный рабочий модуль.

Первая попытка реализации ДРП в нашей Лаборатории для ЭВМ с системой команд СМ-3, "Электроника-60" была выполнена для системы САНПО и описана в/1-3/. В этих работах решены главные вопросы управления размещением РМ на участке памяти системой ДРП, но при ее эксплуатации выявились некоторые недостатки, а именно:

1. Программный интерфейс для управления ДРП оказался сравнительно сложным.

2. Введение специального способа передачи параметров рабочим модулям и фиксация назначения некоторых параметров не дали ожидаемого эффекта.

3. Примененный алгоритм ведет к зачеркиванию на поле ДРП большего числа РМ, чем этого требуют текущие условия, что приводит к снижению его эффективности.

В данной разработке устранены перечисленные выше недостатки. Система реализована на языке ПАСКАЛЬ и ассемблере для машин, программно совместимых с СМ-3. Описываемая разработка в первую очередь предназначена для систем автоматизации экспериментов ядерной физики. Однако, учитывая успешное использование системы ДРП в других организациях /см., например,/4,5/и др./, можно сделать утверждение о целесообразности ее применения для построения систем реального времени в разных областях исследований и народном хозяйстве.

1. ОПИСАНИЕ АЛГОРИТМА ДРП

Система ДРП в описываемой разработке для размещения рабочих модулей использует специально выделенный непрерывный буферный участок оперативной памяти, который будем называть полем ДРП. Логически в данной реализации этот буфер /вернее, часть его/ рассматривается как кольцевой.

Система ДРП может принимать заказы на ряд операций, конечной целью которых является исполнение рабочего модуля, потребовавшегося в соответствии с алгоритмом САЭ. Исполнение РМ разбито на два этапа: 1/ подготовка РМ к работе, 2/ передача ему параметров и управления.

На первом этапе система ДРП выполняет поиск РМ на поле ДРП и, если РМ не обнаружен, выполняет его загрузку. При этом на поле ДРП помещается служебная запись и тело РМ, служебная запись содержит название РМ, адрес входа в РМ, ссылки на служебные записи других РМ, как это показано на рис.1. После вычисления адреса передачи управления модулю этап завершается возвратом управления вызывающей программе.

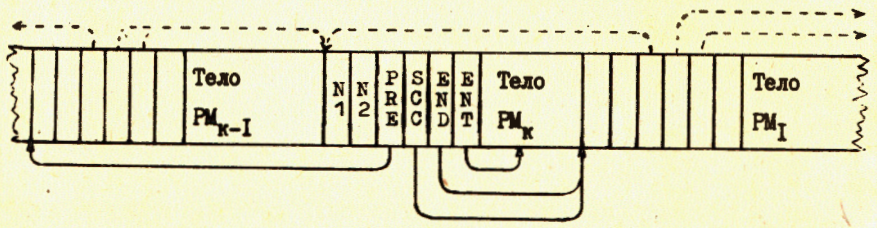

Рис.1. Структура участка памяти, выделенного рабочему модулю, и способ использования характеристик РМ.

На втором этапе в соответствии с правилами передачи параметров процедурам, принятыми в используемой реализации ПАСКАЛЯ/6/, формируется пакет фактических параметров, и управление передается РМ.

Принцип кольцевого буфера реализуется в программах загрузки РМ следующим образом. Загрузка модулей на поле ДРП начинается с младшего адреса. Место для загрузки очередного РМ выбирается непосредственно следом за последним загруженным. Если загруженный модуль не помещается на участке, оставшемся до границы поля ДРП, то этот участок временно оставляется свободным, а модуль будет помещен на начальный участок поля ДРП. При этом будут зачеркнуты РМ, ранее занимавшие эту часть поля ДРП, и сохранены РМ, размещенные дальше. Очевидно, при этом может возникнуть временно не используемая "щель" между концом последнего загруженного РМ и началом ранее загруженного модуля, сохраненного на поле ДРП.

Представленный выше способ использования поля ДРП является основным. Но во время работы САЭ часто приходится иметь дело с необходимостью фиксации в оперативной памяти некоторых РМ, что может быть связано с определенными требованиями алгоритма работы САЭ /управлением внешними устройствами, требованиями быстрой реакции на внешние запросы, обработки запросов прерывания и др./. В нынешней системе эти требования обеспечены путем предоставления возможности фиксировать избранные программы в момент их загрузки на поле ДРП. Для реализации операции фиксации /после возникновения запроса загрузки с фиксацией/ заказанный РМ помещается в начальную часть динамической области ДРП, которая при этом исключается из кольца ДРП, а занимавшие до сих пор этот участок памяти программы уничтожаются. Поскольку фиксация ведет к сокращению длины динамической области ДРП /кольца/, не рекомендуется ее использовать без достаточных оснований. Предусмотрена также возможность освобождения поля фиксированных программ /или его части/, такая операция сопровождается включением освободившейся области в кольцо ДРП.

Таким образом, поле ДРП состоит из двух смежных частей: области фиксации и кольца ДРП, граница между которыми меняется динамически, и длина области фиксации может быть нулевой. Примерная схема состояния поля ДРП с использованием фиксации РМ представлена на рис.2.

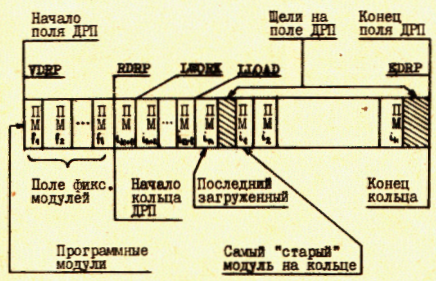

Рис.2. Структура поля ДРП.

Использование кольцевого буфера имеет свои положительные качества и недостатки. Положительной чертой, улучшающей пропускную способность САЭ, является тот факт, что в оперативной памяти ЭВМ в любой момент времени /за исключением момента запуска/ находится максимальное количество рабочих модулей, загруженных в память последними. Это дает возможность выполнять работу САЭ в условиях, когда

при определенных фазах работы САЭ в оперативной памяти /на поле ДРП/ динамически формируются кластеры программных модулей, необходимых для обеспечения нужного на этой фазе программного сервиса. Назовем подсистемой такой кластер модулей, компоненты которого в течение данной фазы работы САЭ вызываются многократно. Если такая подсистема помещается целиком на поле ДРП, то использование механизма ДРП практически не портит пропускной способности САЭ/1/, а применение кольцевого буфера минимизирует количество операции загрузки РМ.

Недостатком является наличие двух свободных пространств в поле ДРП, связанное с принципом кольцевого буфера. Первый свободный участок появляется, как правило, после последнего РМ, загруженного в кольцо, а второй - за модулем, помещенным на поле перед концом ДРП. Такая общая схема представлена на рис.2. Значение объема упомянутых выше свободных пространств на поле ДРП уменьшается при росте отношения длины кольца к длине РМ.

2. ИНТЕРФЕЙС МЕЖДУ ОРГАНИЗУЮЩЕЙ ПРОГРАММОЙ И РАБОЧИМ МОДУЛЕМ

Определенная сложность разработки интерфейса для САЭ, построенной с помощью языка ПАСКАЛЬ, обусловлена тем, что транслятор просматривает списки аргументов, передаваемых процедурам, и в случае обнаружения несовпадений типов аргументов сообщает об ошибке и прекращает трансляцию. Это прямо связано со свойствами языка ПАСКАЛЬ и являлось серьезным препятствием при создании универсальной процедуры передачи аргументов и управления в РМ.

Данная проблема в представляемой системе ДРП решена двумя взаимно не исключающими путями:

1. Спецификация специальной вспомогательной процедуры для вызова каждого из используемых в САЭ рабочего модуля в теле резидентной программы САЭ.
2. Построение универсальной процедуры передачи параметров и управления.

Первый способ требует заранее поместить в теле САЭ все потребующиеся в дальнейшем вспомогательные процедуры вызова РМ. Такие процедуры очень просты. Продемонстрируем это примером. Пусть модуль EX1 имеет следующий заголовок:

```
PROCEDURE  EX1 (VAR  J,K:  INTEGER;  VAR R: REAL);
```

Тогда для его запуска нужно создать процедуру вида:

```
PROCEDURE  STEX1 (VAR  J,K:  INTEGER;  VAR R:  REAL);
 BEGIN
  STPAR
 END;
```


При этом передача параметров модулю обеспечивается исполняющей системой ПАСКАЛЯ в соответствии с заголовком вспомогательной процедуры, а передача управления - процедурой STPAR, включенной в тело вспомогательной.

Схема САЭ, использующая этот способ запуска РМ, будет иметь вид:

```
PROGRAM SAE1;
 VAR
  NOM:NAME;
  ...
 PROCEDURE  STEX1  (VAR  J,K: INTEGER;  VAR R:REAL);
  BEGIN
   STPAR
  END;
 ...
 BEGIN                    (* начало исполнительной части САЭ *)
  ...
  NOM:= 'EX1'
  LM(NOM);                (* загрузка РМ *)
  STEX1 (J,K,R);          (* запуск РМ *)
  ...
 END.                     (* конец исполнительной части САЭ *)
```

Второй способ запуска модуля с передачей параметров /или без нее/ использует ту особенность, что компилятор не проверяет количество и типы аргументов внешней процедуры. Используя это, получим следующую конструкцию:

```
PROGRAM SAE2;
 VAR
  ...
 PROCEDURE NUCLEUS (PROCEDURE ST);
  VAR
  NOM:NAME;
  ...
  BEGIN                    (* начало исполнительной части САЭ *)
   ...
   NOM:='EX1';
   LM(NOM);                (* загрузка РМ *)
   ST(J,K,R);              (* запуск РМ /произвольные типы *)
     ...                   (* и количество аргументов/*)
  END;
PROCEDURE STPAR;EXTERNAL;
 BEGIN
  NUCLEUS (STPAR)
 END.
```

При этом вся резидентная исполнительная часть САЭ должна помещаться в указанном месте.

Поскольку РМ являются внешними (EXTERNAL) по отношению к резидентной части САЭ, проверка соответствия списков параметров в строках вызова РМ и его декларации ложится на автора САЭ при обоих способах вызова РМ.

Если обращение к одному и тому же РМ встречается в тексте САЭ неоднократно, и предполагается изменение спецификации параметров этого модуля, то рекомендуется использовать для обращения к нему первый из указанных способов. При этом транслятор обеспечит выявление всех строк обращений, в которых должна быть выполнена соответствующая коррекция. Использование универсальной процедуры передачи параметров требует меньше затрат и рекомендуется для остальных случаев.

Интерфейс между САЭ и РМ согласован со способом передачи аргументов и управления, принятым в использованной версии программного обеспечения ПАСКАЛЬ. Этот интерфейс в дальнейшем должен учитываться при построении РМ, выполненных на ассемблере.

3. РЕАЛИЗАЦИЯ

При реализации системы ДРП в максимальной степени использовались средства, предоставляемые операционной системой. Так как САЭ предназначена для обслуживания экспериментов в реальном масштабе времени, за основу разработки была выбрана система РАФОС, позволяющая достигнуть предельных скоростных характеристик продукта.

Чтобы получить возможность динамического размещения программ в произвольном месте на поле ДРП, требовалось выбрать соответствующий формат хранения программных модулей.

Существуют два подхода, обеспечивающих такую возможность. Первый - кодирование программ в позиционно-независимом формате. Но этот формат неудобен, поскольку доступен только при программировании на ассемблере и требует специального усложненного способа кодирования программ. Второй способ заключается в подготовке программ в специальном перемещаемом формате (REL),и для широкого круга программ /и языков/ обеспечивается с помощью стандартных средств ОС РАФОС. Его недостатком является потеря времени на настройку программ во время их загрузки. Потери эти пренебрежимы по сравнению с временем чтения тела программы с ВЗУ.

Данная система ДРП разработана для САЭ, модули которой написаны на языках ПАСКАЛЬ или МАКРО-11 /ассемблер/ и представляются в формате .REL.

Система ДРП включает некоторое поле памяти /поле ДРП/ и группу сервисных программ.

Структуру поля ДРП описывает статическая таблица из 5 слов и динамический связной список характеристик модулей. Названия и содержание ячеек статической таблицы следующие:

LLOAD — адрес в ОЗУ последнего загруженного модуля;
LWORK — адрес в ОЗУ модуля, подготовленного к работе;
DRPE — адрес конца поля ДРП;
VDRP — адрес начала поля ДРП;
RDRP — адрес начала кольцевой части ДРП /если нет фиксированных модулей, RDRP=VDRP/.

Все адреса - абсолютные. Переменные VDRP и DRPE определяют границы поля ДРП и обычно не меняются во время работы САЭ /хотя текущая реализация системы ДРП этого не запрещает/. RDRP и DRPE ограничивают часть поля ДРП, предназначенную для загрузки обычных /нефиксированных/ РМ. Содержание переменной RDRP меняется с изменением количества зафиксированных модулей.

Для управления работой модулей и содержанием поля ДРП в начале каждого участка памяти, занимаемого модулем, помещается запись из 6 слов с характеристиками модуля. Такая запись содержит следующую информацию:

1. N1, N2 - шестисимвольное название модуля /совпадающее с используемым во время загрузки с помощью процедуры LM/, представленное в коде RADIX-50.
2. PRE - адрес предыдущего загруженного модуля на ДРП.
3. SCC - адрес следующего модуля, который размещается в кольце за данным. Если кольцо содержит один модуль, то это слово указывает на него. В таком случае содержимое третьего и четвертого слов совпадает.
4. END - адрес первой свободной ячейки памяти, которая непосредственно следует за данным модулем.
5. ENT - адрес точки, в которую передается управление при запуске РМ /точка рабочего входа/.

Сервисные программы системы ДРП написаны на языках МАКРО-11 и ПАСКАЛЬ. Они построены в форме процедур, которые могут вызываться из программ, написанных на ПАСКАЛе или ассемблере, и выполняют следующие действия:

1. Инициация поля ДРП (INIT).
2. Загрузка РМ на поле ДРП (LM).
3. Загрузка РМ на поле ДРП и его фиксация (FIX).
4. Отмена фиксации РМ (UNFIX).
5. Передача параметров и управления /запуск/ РМ (ST).

Первая процедура (INIT) не имеет параметров. Она задает начальные значения указателям, описывающим поле ДРП, и должна быть вызвана в начале работы системы ДРП до использования других процедур ДРП. Вызов INIT во время работы САЭ приведет к уничтожению всего содержимого поля ДРП /вместе с фиксированными РМ/, и может быть полезным при переходе от одного этапа работы САЭ к другому.

Процедура загрузки РМ для вызова из программы, написанной на языке ПАСКАЛЬ, декларируется следующим способом:

```
LM(VAR NOM:NAME);
```

Здесь тип NAME определен как ARRAY [1 ..6] OF CHAR, т.е. переменная NOM содержит 6-символьное название файла на диске (DKØ:), содержащего требуемый для загрузки РМ. Указанный файл должен иметь расширение названия .REL.

После вызова процедуры LM начинается поиск указанного РМ на поле ДРП и в случае его обнаружения повторная загрузка не выполняется, а только осуществляется обновление в системных ячейках значений параметров, необходимых для запуска РМ.

Если попытка найти модуль на поле ДРП оказывается неудачной, то процедура приступает к подготовке места для загрузки РМ,считывает тело РМ с ВЗУ и выполняет настройку РМ для работы его на выделенном участке поля ДРП. После завершения этой работы процедура записывает в системные ячейки значения параметров, необходимых для передачи управления РМ.

Процедура FIX загрузки модуля с фиксацией /синтаксис вызова, как для LM/ выполняет те же действия, что и процедура LM, с тем различием, что поиск модуля происходит не по всему полю ДРП, а только в той его части, где размещаются фиксированные модули. Если модуль не обнаружен, происходит загрузка РМ, сопровождающаяся расширением части поля ДРП, отведенной для фиксированных программ, т.е. загрузка с фиксацией. После такой загрузки значения параметров для запуска фиксированного РМ не передаются. Получение таких значений для запуска фиксированного модуля будет успешно выполняться с помощью процедуры LM.

Если РМ запускается несколько раз подряд без использования в промежутках между его работой операции загрузки других модулей или процедуры INIT, тогда можно не вызывать процедуру LM, а только выполнить передачу новых значений параметров и управления этому РМ. Об этом полезно помнить, особенно в случаях, когда модуль работает в условиях, критичных по времени.

Для отмены фиксации модуля служит процедура UNFIX:

```
UNFIX (VAR NOM:NAME);
```

с теми же параметрами, что и для процедуры LM. После вызова этой процедуры выполняется поиск указанного модуля на поле фиксированных модулей ДРП, и перемещение границы между динамической и "фиксированной" частями ДРП в точку непосредственно перед местом, занимаемым модулем, фиксация которого отменяется. Таким способом процедура UNFIX выполняет расфиксацию не только указанного модуля, но и всех, которые были фиксированы позже него.

После выполнения процедуры UNFIX расфиксированные модули остаются на поле ДРП и становятся доступны для использования, но могут быть зачеркнуты при любой операции загрузки, поскольку находятся на его динамической части /кольце/.

Последним действием для использования рабочего модуля, загруженного на поле ДРП, является передача параметров для исполнения и инициации исполнения, т.е. передача управления в загруженный модуль.

Описанные процедуры необходимы и достаточны для эффективного использования системы ДРП. Для удобства пользователи ДРП могут при построении конкретных САЭ на основе имеющихся создать дополнительные процедуры для выполнения комплексных задач при загрузке и запуске РМ на поле ДРП. Примером может служить процедура:

```
PROCEDURE LMANDSTEX1 (VAR NOM:NAME);
          VAR J,K:INTEGER;VAR R:REAL);
 BEGIN
  LM(NOM);
  STEX1(J,K,R)
 END;
```

которая объединяет представленные раньше процедуры LM и STEX1.

4. ОБЩИЕ ХАРАКТЕРИСТИКИ И РЕЗУЛЬТАТЫ

Процедуры поддержки представленной системы ДРП для экономии памяти и временных затрат при работе САЭ выполнены в основном на языке МАКРО-11. Вся система ДРП занимает в сумме только 644 слова в памяти ЭВМ, включая внутренний буфер длиной 256 слов для обмена с диском.

Одной из самых важных характеристик системы ДРП, определяющих ее применимость для конкретной САЭ, является время доступа к РМ, слагающееся из времени его загрузки и поиска на поле ДРП.

Время загрузки РМ, согласно[1], зависит от ряда факторов /типа устройства, используемого для хранения библиотеки РМ, способа организации хранения, длины и позиции РМ в хранилище/ и обычно составляет 125-450 мс.

Другой важной временной характеристикой системы ДРП является время доступа к модулю, присутствующему на поле ДРП, т.е. время поиска.

Время поиска зависит от позиции, занимаемой РМ на поле ДРП. Программа выполняет поиск РМ на поле фиксированных модулей, просматривая его с начала. Поиск продолжается по динамическому участку /кольцу/, причем здесь порядок поиска меняется, т.е. программа ведет просмотр кольца, начиная с последнего из загруженных модулей.

Время поиска описывается формулами:

$$T'_{\text{поиска}} = 480+к*40 \ [\text{мкс}],$$

если модуль занимает k-тую позицию на фиксированном участке поля ДРП, и

$$T_{\text{поиска}} = 470+м*40+n*34[\text{мкс}],$$

если РМ находится на динамическом участке. Здесь: м - число фиксированных РМ, n-1 - число модулей, загруженных после искомого.

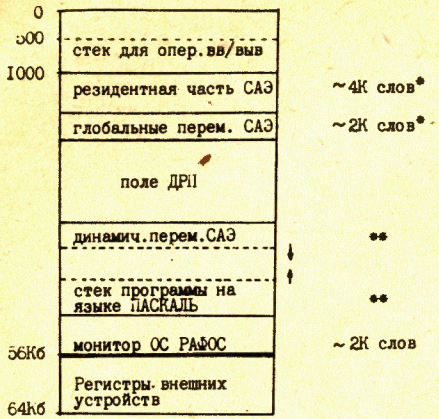

Рис.3. Примерное распределение памяти ЭВМ в САЭ.

Разработанная система автономна, может быть включена в любую программу или систему, как обычная группа процедур, и не использует никаких подпрограмм из библиотеки программного обеспечения языка ПАСКАЛЬ. Разработка выполнена таким образом, чтобы любой РМ мог быть оформлен в виде обычной процедуры на языке ПАСКАЛЬ, что в значительной мере облегчает создание и отладку САЭ.

Применение кольцевой организации рабочего поля ДРП гарантирует помещение на нем максимально возможного количества и наиболее "свежего" состава РМ. Такой подход снижает среднее количество операции загрузки модулей по отношению к количеству обращений к ним. Увеличению эффективности работы системы ДРП сопутствует тот факт, что в состав прикладной САЭ обычно входит набор подсистем, модули которых, будучи помещены на поле ДРП, могут быть многократно использованы без повторной загрузки.

На основании данных о нескольких системах /см.,например,/1//
ориентировочное распределение памяти между компонентами САЭ и ОС можно представить цифрами, приведенными на рис.3. Для известных автору систем, созданных с использованием ДРП, длина поля ДРП в пределах 10 К слов оказалась достаточной.

Заштрихованный на рис.3 участок представляет область памяти, используемой исполняющей системой ПАСКАЛЯ /стек и область динамических переменных для программ/. Автор САЭ должен планировать свою систему так, чтобы после размещения всех резидентных элементов системы и поля ДРП /длина которого регулируется/ оставался достаточный ресурс памяти для исполняющей системы ПАСКАЛЯ.

Отличительными особенностями данной работы являются следующие:

1. Реализация динамического распределения памяти для программ, написанных на языке ПАСКАЛЬ в ОС РАФОС, что позволяет, в случае необходимости, существенно увеличить объем разрабатываемой прикладной системы при одновременном упрощении процесса ее сборки/1/.
2. Применение кольцевой организации поля динамической памяти, позволяющей уменьшить количество операции загрузки модулей по сравнению с алгоритмами, описанными в/1,3,7,8/.
3. Реализация программного интерфейса между организующей частью прикладной программы /или системы/ и модулями, обеспечивающего модулям следующие свойства:

- трансляция и редактирование связей /т.е. получение загрузочного формата/ для каждого модуля могут быть выполнены отдельно от остальных;
- модуль может иметь структуру процедуры с произвольными типами и количеством параметров;
- модуль может быть записан на языках ПАСКАЛЬ, МАКРО-11, ФОРТРАН-4.

ЛИТЕРАТУРА

1. Балука Г. ОИЯИ, Р10-84-463, Дубна, 1984.
2. Балука Г. и др. ОИЯИ, Р10-12960, Дубна, 1980.
3. Балука Г., Островной А.И. ОИЯИ, Р10-13004, Дубна, 1980.
4. Саламатин И.М. и др. Автометрия, 1981, №4, с.60-69.
5. Гриднев Г.Ф., Саламатина Т.С. ОИЯИ, 10-83-598, Дубна, 1983.
6. Валикова Л.И. и др. Операционная система СМ ЭВМ РАФОС. "Финансы и статистика", М., 1984.
7. Шура-Бура М.Р. Интерпретирующая система для М-20. Изд-во ВЦ АН СССР, М., 1965.
8. Дади Л. и др. Программа динамического распределения памяти для мини-ЭВМ. "Программирование", 1978, №2, с.26-32.

Рукопись поступила в издательский отдел
19 декабря 1984 года.

Внимание организаций и лиц, заинтересованных в получении публикаций Объединенного института ядерных исследований

Принимается подписка на препринты и сообщения Объединенного института ядерных исследований.

Установлена следующая стоимость подписки на 12 месяцев на издания ОИЯИ, включая пересылку, по отдельным тематическим категориям:

ИНДЕКС	ТЕМАТИКА	Цена подписки на год
1.	Экспериментальная физика высоких энергий	10 р. 80 коп.
2.	Теоретическая физика высоких энергий	17 р. 80 коп.
3.	Экспериментальная нейтронная физика	4 р. 80 коп.
4.	Теоретическая физика низких энергий	8 р. 80 коп.
5.	Математика	4 р. 80 коп.
6.	Ядерная спектроскопия и радиохимия	4 р. 80 коп.
7.	Физика тяжелых ионов	2 р. 85 коп.
8.	Криогеника	2 р. 85 коп.
9.	Ускорители	7 р. 80 коп.
10.	Автоматизация обработки экспериментальных данных	7 р. 80 коп.
11.	Вычислительная математика и техника	6 р. 80 коп.
12.	Химия	1 р. 70 коп.
13.	Техника физического эксперимента	8 р. 80 коп.
14.	Исследования твердых тел и жидкостей ядерными методами	1 р. 70 коп.
15.	Экспериментальная физика ядерных реакций при низких энергиях	1 р. 50 коп.
16.	Дозиметрия и физика защиты	1 р. 90 коп.
17.	Теория конденсированного состояния	6 р. 80 коп.
18.	Использование результатов и методов фундаментальных физических исследований в смежных областях науки и техники	2 р. 35 коп.
19.	Биофизика	1 р. 20 коп.

Подписка может быть оформлена с любого месяца текущего года.

По всем вопросам оформления подписки следует обращаться в издательский отдел ОИЯИ по адресу: 101000 Москва, Главпочтамт, п/я 79.

Балука Г.
Р10-84-789

Реализация динамического распределения памяти
с кольцевой организацией буфера
для программных систем на языке ПАСКАЛЬ

Описаны программа динамического распределения памяти и перемещающий загрузчик программных модулей, предназначенные для использования в системах автоматизации экспериментов в условиях дефицита оперативной памяти ЭВМ. Применена кольцевая организация буфера модулей. Программные модули могут быть написаны на языках ПАСКАЛЬ, МАКРО-11. Реализация выполнена для ЭВМ типа СМ-3, "Электроника-60" и ОС типа РАФОС.

Работа выполнена в Лаборатории нейтронной физики ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1984

Перевод О.С.Виноградовой

Baluka G.
P10-84-789

Realization of Dynamic Memory Scheduling
with Ring Buffer Organization for Programs
Written in PASCAL Programming Language

A program for dynamic memory scheduling and a relocatable program module loader are described. The described method can be applied in the systems with main memory shortage problems. High efficiency of the approach is guaranteed because of utilizing ring buffer organization for program modules. The realization has been performed for program modules written in PASCAL programming language or MACRO-11 (assembler) on SM-3, Elektronika-60 type minicomputers and RT-11 operation system.

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

Communication of the Joint Institute for Nuclear Research. Dubna 1984