

ОБ ИСПОЛЬЗОВАНИИ ВОЗМОЖНОСТЕЙ ОПЕРАЦИОННЫХ СИСТЕМ В ПАКЕТАХ ПРИКЛАДНЫХ ПРОГРАММ

И.Н. Парасюк, И.В. Сергиенко

В настоящее время вопросам создания пакетов прикладных программ (ППП) уделяется много внимания. Различными авторами в этой области прикладной кибернетики предлагаются ключевые понятия и определения, принципы создания ППП, изучаются составные части ППП и их функциональная взаимосвязь, способы машинной реализации ППП и использование возможностей современных операционных систем при их разработке.

Создание ППП на ЭВМ второго поколения сопровождалось преодолением ряда трудностей, связанных, в основном, с ограниченными ресурсами ЭВМ, их быстродействием и наличием математическим обеспечением и другими объективными факторами. В связи с этим в первых ППП приходилось зачастую создавать элементы служб современных операционных систем, таких как загрузчик и редактор связей, управление данными и задачами, управление памятью и др. Разработчиками создавались методы, позволяющие организовать вычислительный процесс в ППП с экономным использованием ресурсов ЭВМ.

Одним из основных принципов создания ППП является развитая модульность его программного обеспечения, позволяющая достигнуть гибкости пакета, т.е. реализации таких принципов

построения ППП как адаптируемость, модифицируемость, расширяемость, совершенствование и др. Кроме того, современные ППП должны быть просты в применении, позволять автоматизировать процессы выбора методов решения прикладных задач и построения соответствующих программ, допускать оперативное сопряжение с другими пакетами, программами, системами и т.п.

В настоящем докладе будем изучать ППП, которые функционируют под управлением операционной системы (ОС) в режиме пакетной обработки данных. Такой ППП в ОС рассматривается как проблемная программа, выполняющая в одном из ее разделов. ОС осуществляет общее управление вычислительной системой, операциями обмена данных, вычислительным процессом в целом, обеспечивая при этом оптимальное использование ресурсов системы.

Процесс решения задач с помощью такого ППП может быть представлен схемой рисунка I.

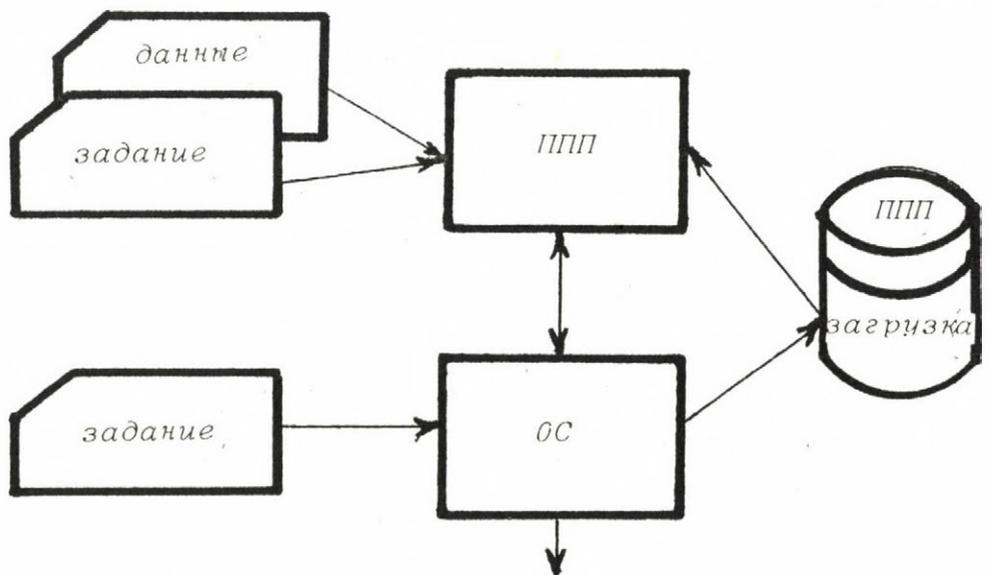


Рис. I

Задание на решение прикладных задач с помощью ППП состоит из двух составных частей: задания для ОС и задания для ППП. В задании для ОС на языке управления заданием [2] описываются ресурсы ЭВМ, необходимые для работы ППП, данные, с которыми ОС должна будет работать, а также сведения о размещении в памяти ЭВМ резидентной части ППП*), которую необходимо загрузить в свободный участок основной памяти и осуществить ее запуск. В задании ППП во входном его языке описывается режим работы ППП, задачи, которые необходимо решать, формат, тип и носитель данных, с которыми будет работать ППП. Задание ОС и задание ППП последовательно включаются во входной поток ОС и обрабатываются соответственно управляющей программой ОС и управляющей программой ППП. Когда управление передано управляющей программе ППП (ее резидентной части, в дальнейшем просто - р е з и д е н т у), последняя согласно заданию пакета с помощью средств ОС и под ее контролем осуществляется построение алгоритма (программы) решения прикладных задач и его выполнение. В некоторых случаях, например, когда некоторые задания ОС систематически используются, их целесообразно оформить в виде каталогизированных процедур. Это значительно упрощает процесс подготовки задания [3].

В современных ППП можно выделить следующие основные части (программы):

*) Под резидентной частью ППП понимается та часть управляющей программы ППП, которая в процессе работы ППП постоянно находится в основной памяти. Хранить весь ППП в основной памяти нецелесообразно, так как при этом не рационально используется основная (дорогостоящая) память. Выделение резидентных частей применяется в настоящее время во многих системах ППП и т.п.

1) управляющая программа пакета, которую часто называют монитором пакета или ведущей программой пакета;

2) программа управления данными пакета, которую также называют администратором пакета;

3) программа совершенствования пакета;

4) библиотека баз данных пакета, называемая иногда в упрощенных вариантах ППП библиотекой пакета, телом пакета;

Программа управления данными совместно с библиотекой баз данных составляют банк данных ППП.

Функциональная взаимосвязь этих программ ППП схематически показана на рис. 2.

На этом рисунке стрелочки " \Rightarrow " указывают информационные связи, а стрелочки " \rightarrow " - связи по управлению.

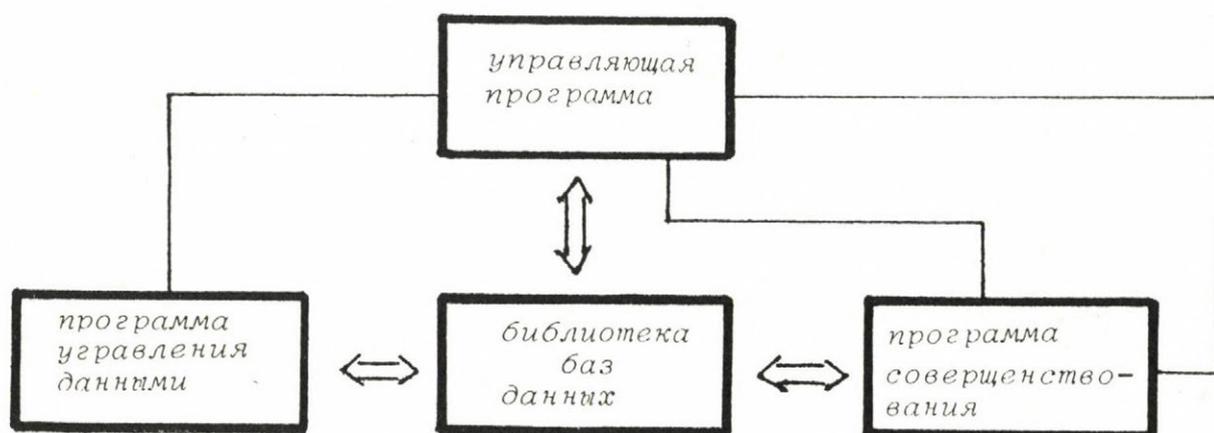


Рис. 2

Для работы ППП целесообразно предусмотреть следующие режимы.

I. Режим счета. Это основной режим работы ППП. В этом режиме осуществляется выбор необходимых математичес-

ких методов для решения заданной совокупности проблемных задач и организация процесса их решения. Режим счета применяется в ППП как автоматически, так и в сопряжении с другими программами.

2. Р е ж и м к о н с т р у и р о в а н и я. В этом режиме осуществляется конструирование ППП программ, записанных на исходных языках высокого уровня типа АЛГОЛ, ФОРТРАН, РЛ/1 [2] и др. и предназначенных для решения определенной совокупности проблемных задач и организация их записи на заданные носители информации (ПК, ПФ, МД, АЦПУ и т.п.).

3. Р е ж и м с о в е р ш е н с т в о в а н и я. Этот режим предназначен для программной реализации идей совершенствования математического обеспечения ППП, основанного на накоплении и обработке "опыта" его функционирования [1].

4. Р е ж и м у п р а в л е н и я. Этот режим условно объединяет все операции по обновлению, расширению, отладке, выдаче копий математического обеспечения ППП и различного рода справок о нем (о его использовании, возможностях, хранении, инструкциях и т.п.).

Об особенностях и способах реализации этих режимов в ППП мы расскажем несколько ниже. Сейчас рассмотрим вкратце другой круг вопросов, связанных с созданием ППП, а именно, языки входных сообщений ППП и требования к ним.

Анализ перечисленных выше режимов работы ППП, требований пользователей и особенностей организации вычислительного процесса позволяют сделать вывод о целесообразности выделения в современных ППП двух типов языков, обладающих специфическими особенностями. Прежде всего выделим п р о б л е м -

н о - о р и е н т и р о в а н н ы й язык П П П, как правило, язык очень высокого уровня, предназначенный для описания решаемой совокупности проблемных задач. Этот язык часто называют языком пользователя. Он должен быть сравнительно прост по своей структуре, легок в освоении и применении. Он должен позволять пользователям, как правило, не имеющим специальной подготовки по программированию, формулировать решаемые ими задачи не вдаваясь в логику машинного алгоритма их решения.

Существуют два подхода к созданию проблемно-ориентированных языков ППП. Один из них состоит в расширении языков высокого уровня типа ФОРТРАН, АЛГОЛ, PL/1 и т.п. средствами, ориентированными на решение проблемных задач специальных классов. Процесс формулировки (описания) задач с помощью таких языков, обычно, заключается в том, что составляется проблемная программа, поочередно использующая необходимые средства соответствующего языка высокого уровня и его расширения.

Другой подход к созданию проблемно-ориентированных языков ППП, на котором по ряду причин мы остановим свое внимание, состоит в специальной разработке самостоятельного языка, называемого языком управления заданиями П П П. Языки высокого уровня типа АЛГОЛ, ФОРТРАН и т.п. используются в этом случае для написания программных модулей, реализующих определенные фрагменты применяемых алгоритмов. Эти программные модули в различном виде (исходные модули, объективные модули, модули загрузки) совместно с логикой алгоритмов решения задач скрыты от пользователя в ППП. Формулировка задач, подлежащих решению, осуществляется те-

перь лишь с помощью проблемно-ориентированных языков ППП очень высокого уровня. Она включает описание входных данных, результатов, укрупненной модели алгоритма решения задач, режима выполнения этого задания пакетом и другой необходимой управляющей информации. Обычно проблемно-ориентированные языки ППП имеют операторную структуру. Среди них можно выделить управляющие, вычислительные, информирующие, описательные, обслуживающие и др. операторы. Например, в проблемно-ориентированном языке пакета программ математической статистики (первая версия) наряду с операторами начала и конца задания включены операторы, описывающие формат и носители информации для выдачи результатов, ввода исходных данных, а также операторы, позволяющие формулировать задачи статистической обработки данных. Например, операторы оценки статистических характеристик, имитации данных, распределенных по заданному закону, проверки различных гипотез, фильтрации выборки, построения моделей и многие другие, позволяющие решать задачи из различных разделов математической статистики. Выделяют [4] три наиболее распространенных формата операторов: формат операторов языка управления заданиями операционной системы, формат макрокоманд и формат операторов обращения к процедурам. Каждый из форматов содержит поле метки оператора, поле имени оператора и полей параметров и операндов.

В процессе создания проблемно-ориентированного языка ППП следует учитывать принципы расширяемости, адаптируемости и модифицируемости. Это значит, что языковые средства пользователя должны быть открыты для расширения и модификации, а их структура должна позволять без особых затрат переориен-

тировать этот язык на решение задач из другой области. Отсюда вытекают два возможных подхода, которые можно применять при реализации входного языка ППП: 1) создать ядро языка, допускающее наращивание новыми элементами и адаптацию к новым областям применения; 2) создать мета-язык, позволяющий описывать языки, ориентированные на новые проблемы.

Важным моментом в реализации языкового обеспечения ППП является создание так называемого **внутреннего языка**, общего для всей совокупности языков данного пакета, как, например, в системе УСОД, что позволяет более эффективно адаптировать пакеты на новые области применения.

Второй тип языков предназначен для описания заданий администратору данных и программе совершенствования. Характерной особенностью этих языков является то, что они, как правило, при сохранении структур данных пакета, инвариантны к областям его применения. Опыт показывает, что язык администратора можно построить на базе следующих основных операторов: **ВЫДАТЬ, КОПИРОВАТЬ, СКОНСТРУИРОВАТЬ, ИЗМЕНИТЬ, ЗАМЕНИТЬ, УДАЛИТЬ (ИСКЛЮЧИТЬ), УПЛОТНИТЬ, УПОРЯДОЧИНИТЬ, КОНСТРУИРОВАТЬ, РАСПЕЧАТАТЬ, СЛИТЬ, ЧЛЕНИТЬ, ВОССТАНОВИТЬ, ВКЛЮЧИТЬ (ПОПОЛНИТЬ)** и др.

В язык программы совершенствования пакета можно включить такие операторы как: **ОЦЕНИТЬ, ПОДГОТОВИТЬ, ПРЕДЛОЖИТЬ** и т.п.

Рассмотрим вкратце функциональную структуру пакета и назначение ее составных частей.

Управляющая программа представляет собой совокупность управляющих и обрабатывающих программ, которые предназначены для анализа входных сообщений, настройки пакета на определен-

ный режим работы и управления процессом выполнения задания. Управляющая программа обычно имеет динамическую структуру. Она состоит из программы трансляции текста, записанного во входном языке ППП и представляющего задание пользователя пакета, в текст, записанный на промежуточном языке ППП программы анализа этого текста, программы загрузки в рабочие области памяти соответствующих программ, таких как: программа автоматического построения обобщенной вычислительной схемы алгоритма решения задач; программа компоновки программ из исходных модулей или модулей загрузки; программа организации интерфейсов и др.

Рассмотрим основные элементы баз данных ППП.

Под программными модулями (ПМ) ППП будем понимать автономные, упорядоченные фрагменты алгоритмов решения прикладных задач, описанные на одном из языков высокого уровня, оформленные в виде процедур.

Каждый ПМ имеет свой паспорт, в котором содержатся все данные, необходимые для применения данного ПМ в различных ППП, если это целесообразно. Например, ПМ пакета программ математической статистики (ПМС), разработанного в Институте кибернетики АН УССР и функционируемого под управлением ЕС ЭВМ, написаны на языке ФОРТРАН, а их паспорта содержат следующие данные: имя ПМ, количество глобальных (внешних) переменных ПМ, параметры глобальных переменных (формат, размерность, относительные адреса в поле глобальных переменных) и др.

П М з а д а ч и - это собранная из необходимых ПМ программа для решения данной задачи. Содержать ПМ задачи це-

десообразно в тех случаях, когда вероятность того, что данная задача будет включена в задание, сравнительно велика. Определение таких задач - одна из функций программы совершенствования ППП.

Остановимся вкратце на вопросах построения системы программных модулей 5 представляющей структуру класса задач. Опыт показывает, что от того, какой выбран метод определения совокупности программных модулей, существенно зависят показатели, характеризующие качество функционирования пакета.

Общий случай, т.е. случай, когда класс решаемых задач, а следовательно, и алгоритмы их решения окончательно не определены (хотя бы на этапе проектирования) связан с многими трудностями при решении вопроса о создании оптимальной в некотором смысле системы программных модулей.

Для случая, когда класс задач, допустимых для решения пакетом, окончательно определен, причем этот класс конечный и такой, что каждой задаче можно поставить в соответствие некий вполне определенный алгоритм ее решения, авторами разработан формальный метод, названный ими "ψ-метод", позволяющий построить систему программных модулей, оптимальную в том смысле, что число входящих в нее ПМ минимально при следующем ограничении - суммарное количество функциональных операторов [5] при решении всевозможных комбинаций задач - минимальное.

Систему ПМ задач в данном случае можно характеризовать модульным графом [5], т.е. информационным ярусно-параллельным графом, вершинами которого являются программные модули, а информационные дуги указывают на глобальные переменные, вос-

принимаемые и вырабатываемые программными модулями в качестве исходных данных и результатов соответственно. Порядок выполнения программных модулей в этой системе определяется следующими правилами: ПМ одного яруса выполняются в произвольном порядке, ПМ i -го яруса можно выполнить только после выполнения смежных с ним ПМ $(i-1), (i-2), \dots, 0$ ярусов.

Вычислительная схема алгоритма решения произвольной задачи, т.е. последовательность (цепочка) имен ПМ, выполнение которых приводит к решению данной задачи, на этом графе определяется объединением всевозможных путей, ведущих из заданной начальной вершины (ПМ формата (типа) данных) в вершину, определяющую данную задачу. Система ПМ в пакете ПМС как и в системе УСОД представляется вычислительными схемами алгоритмов отдельных задач, полем глобальных переменных, таблицей паспортов ПМ. Вычислительные схемы алгоритмов хранятся в таблицах, называемые таблицами вычислительных схем алгоритмов, в которых содержатся следующие данные: имя задачи, количество ПМ, составляющих вычислительную схему и сама вычислительная схема.

Таблица вычислительных схем и таблица паспортов ПМ в пакете ПМС реализованы в виде невыполнимых многосекционных подпрограмм языка Ассемблер. Каждая такая секция описывает паспорт ПМ или вычислительную схему алгоритма.

Рассмотрим теперь вопросы функционирования управляющей программы пакета на примере монитора пакета ПМС. Процесс функционирования монитора начинается с загрузки и инициирования его резидента, осуществляющего дальнейшее управление пакетом. Он выполняет функции, инициирования программ и функ-

ции внутреннего управления. Интерфейс между программами монитора и его резидентом осуществляется через область связи путем передачи соответствующих кодов возврата и состояния. Область связи пакета ПМС, являющаяся аналогом поля фиксаторов системы УСОД, загружается резидентом в ОЗУ и используется для двухсторонней связи между программами монитора. В эту область помещается информация, которая используется для внутреннего управления (режим работы памяти, формат данных) адреса некоторых программ и т.п.). Структура области связи в пакете ПМС определяется специально введенной невыполнимой макрокомандой языка Ассемблер, макрорасширение которой представляет собой фиктивную секцию программы, что позволяет при написании программ на языке Ассемблер обращаться к ячейкам области связи по именам.

После загрузки резидентом монитора в оперативную память области связи, загружается в рабочую область программа анализа входных сообщений, которая тут же получает управление. Эта программа один за другим вводит и обрабатывает операторы задания (оператор задания, оператор режима работы пакета, оператор шагов задания). Результаты обработки этих операторов загружаются в область связи. В процессе обработки осуществляется синтаксический и семантический контроль предложений задания. В случае, если ошибка обнаружена хотя бы в одном из предложений шага задания, этот шаг задания аннулируется, а на АЦПУ выдается диагностическое сообщение об ошибках.

В том случае, если ошибок в предложениях шага задания не обнаружено, осуществляется перевод его предложений на внутренний язык пакета и из этой информации формируется рабо-

чий файл, который записывает на МЛ и содержит следующую информацию: список имен задач шага задания, имя формата данных и их размерность.

Процесс анализа и обработки предложений задания сопровождается диагностическими сообщениями на АЦПУ. После обработки последнего шага задания формируется код возврата резиденту монитора.

Следующей за программой анализа входных сообщений резидент монитора загружает программу планирования вычислительного процесса и организации его выполнения. Эта программа выполняет следующие функции: подготавливает к работе систему ПМ и информационные таблицы, необходимые для организации пошагового решения задач, конструирование вычислительной схемы обобщенного алгоритма для решения задач данного шага задания, распределение и контроль оперативной памяти, необходимой для исходных данных и глобальных переменных, подготовка списка фактических параметров для ПМ, загрузка в оперативную память ПМ и организация их выполнения.

Рассмотрим более подробно функциональную схему этой программы.

Выделяются рабочие буферы для набора (списка) задач k -го шага задания, таблицы вычислительных схем алгоритмов данного класса задач, вычислительной схемы обобщенного алгоритма решения задач k -го шага задания и заявка на требуемый объем оперативной памяти поступает к резиденту монитора. После выделения требуемой памяти, загружается в рабочие области список задач k -го шага задания: таблица вычислительных схем алгоритмов и по аналогии с системой УСОД конструиру-

ется (синтезируется) вычислительная схема обобщенного алгоритма совместного решения всей совокупности задач k -го шага задания. Далее, на поле таблицы вычислительных схем алгоритмов загружается таблица паспортов ПМ и определяются адреса паспортов ПМ, входящих в вычислительную схему обобщенного алгоритма. Эти адреса последовательно заносятся в подготовительный список параметров (аналог поля вычислительных схем в системе УСОД).

В процессе анализа паспортов ПМ эта программа осуществляет контроль описания исходных данных, определяет необходимый объем оперативной памяти, который и запрашивается у резидента. Если установлено, что требуемая память не может быть выделена или имеется несоответствие в описании данных, то рассматриваемый шаг задания аннулируется. В случае, если никаких аварийных ситуаций не возникло, осуществляется распределение памяти для входных и выходных параметров ПМ в поле глобальных переменных. Для простых глобальных переменных в пакете ПМС память не выделяется (ее значение находится в области связи, а адрес соответствующей ячейки поля глобальных переменных включается в список параметров). Если глобальная переменная - массив и если ей еще не выделено место в поле глобальных переменных, то определяется объем требуемой памяти и посылается соответствующий запрос резиденту. В случае выделения требуемого объема памяти, адрес области помещается в список параметров и в соответствующую ячейку поля глобальных переменных.

Когда список всех параметров ПМ подготовлен, ПМ загружается в оперативную память, пересылается этому ПМ адрес

списка параметров и передается ему управление. ПМ загружается на одно и то же поле в ОЗУ. При возникновении аварийных ситуаций управление предлагается программе обработки аварийных ситуаций. Нормальная (без аварийных остановок) работа ПМ завершается возвратом управления системе для подготовки и загрузки очередного ПМ.

Выполнение шага задания влечет аннулирование выделенной для него оперативной памяти и передачу управления на анализ следующего шага задания.

Выше, на примере пакета ПМС был рассмотрен процесс планирования и организации вычислительного процесса в обычном аспекте. Однако, как показывает опыт разработки и применения ряда ППП, что целесообразно к блокам ППП подключить программные средства сбора и обработки статистических данных о функционировании пакета. Обработка, накопленного таким образом "опыта" позволит совершенствовать организацию ППП. Например, уточнить класс решаемых задач, оптимально (в смысле минимизации времени загрузки ПМ в рабочие области) разместить ПМ, их паспорта и необходимые таблицы на внешних носителях информации, оценить уровень агрегации ПМ и задач, и если потребуется, предложить списки ПМ и задач, подлежащие укрупнению, членению, оценить параметры, характеризующие эффективность пакета и др. Для этой цели в определенных точках организующей программы необходимо подключить программы фиксирующие временные и частотные характеристики работы ее составных частей, как это сделано в системе УСОД.

В этой системе накапливаются такие данные как частота использования класса задач, различных форматов (типов) дан-

ных, объем обрабатываемой статистики, частота решения различных задач данного класса, частота использования программных модулей в процессе решения, длина сконструированной программы. Более того, в системе УСОД накапливается статистика об использовании программных модулей и параметры сконструированных программ для случая, если бы процесс конструирования рабочих программ не сопровождался их оптимизацией.

Очевидно, что наличие программ сбора статистики несколько замедляет процесс решения задач, поэтому целесообразно, чтобы была возможность отключать работу этих программ в экстренных ситуациях.

Если пакету передано задание для совершенствования, то подключаются программы, обрабатывающие накопленный "опыт" работы и вырабатывающие проект изменения пакета, который наряду с различными оценками некоторых параметров немедленно сообщается "хозяину" пакета. В случае, если принято решение совершенствовать пакет, то соответствующий проект передается администратору данных пакета, который и осуществляет перестройку пакета.

Идея сопряжения ППП с другими программами в своей основе сходна с использованием ППП в режиме счета, что и позволило объединить их в один режим работы - режим счета. Особенностью организации сопряжения программ является то, что потребителям, вырабатывающим данные для обработки, формирующим соответствующие задачи и потребляющие результаты решения является некоторая активная программа (программа-пользователь). Ею может быть программа, пакет или система. В предположении, что программа пользователь (" - программа) (так бу-

дем называть программно-организованного пользователя пакета) обучена формулировать задание ППП и его запуск, ППП должен быть способен понять это задание, реализовать его и, осуществив передачу результатов на заданные поля, возвратить управление «-программе (в случае успешной реализации задания).

В работе 6 уже рассмотрены особенности организации сопряжения автоматизированной системы обработки данных и системы моделирования и приведена принципиальная схема сопряжения двух реально действующих систем - системы УСОД, предназначенной для обработки статистических данных и системы СЛЭНГ, предназначенной для моделирования систем с дискретными событиями.

Здесь мы укажем лишь на те принципиальные трудности, которые возникают в процессе разработки и реализации сопряжения ППП с «-программами.

1. Организация памяти. Так как на память «-программы никаких ограничений не накладывается (обычно это довольно сложные и громоздкие системы), то следует разработать некий универсальный механизм распределения и защиты памяти, который позволял бы поочередно для ППП и «-программы использовать выделенную основную память.

2. Организация взаимно-оперативной информационной связи. Общий случай обмена данными между ППП и «-программой на одном из шагов их совместной работы заключается в оперативной передаче данных «-программой (результатов работы «-программы) ППП и наоборот - передача результатов работы ППП «-программе для дальнейшей их обработки.

3. Планирование оперативного управления вычислительным

процессом в «-программе и ППП с учетом возможных аварийных ситуаций.

Другой аспект применения возможностей современных ППП заключается в автоматизации процесса конструирования программ из ПМ, написанных на исходных языках высокого уровня. Результатом работы ППП в данном случае является исходная программа, записанная на заданном носителе данных. Для работы ППП в этом направлении используется тот же, описанный выше, механизм анализа входных сообщений и построения вычислительной схемы алгоритма. Принципиально новым здесь будет механизм функциональной связи ПМ.

Л И Т Е Р А Т У Р А

1. И.Н. Парасюк, И.В. Сергиенко, Н.И. Тукалевская. Универсально-специализированная автоматизированная система обработки данных на ЭВМ (система УСОД), ж.УСиМ, №2, К., 1974.
2. К. Джермейн. Программирование на IBM/360, Изд-во "МИР", М., 1973.
3. Дж. Донован. Системное программирование, Изд-во "МИР", М., 1975.
4. А.С. Стукало. Вопросы разработки языков управления заданиями в пакетах прикладных программ, В сб. "Вопросы оптимизации и организации вычислений". Изд. РДЭНТП, Киев, 1975.
5. И.Н. Парасюк, И.В. Сергиенко. Некоторые вопросы разработки и исследования одного класса универсально-специализированных систем обработки данных, ж. "Кибернетика", №6, К., 1973.
6. И.Н. Парасюк, М.А. Сахнюк. Программный комплекс СЛЕНГ-УСОД, ж. "УСиМ", №2, К., 1974.

Összefoglaló

I.N. Paraszuk, I.V. Szegrienko

Operációs rendszerek tulajdonságainak alkalmazása programcsomagok készítésénél

A cikk az alkalmazott programcsomagok felépítésével, tervezésével foglalkozik. Az általános probléma felvetések és megoldások illusztrációjaként a szerzők konkrét példákat mutatnak be. A programcsomagok belső szerkezetén kívül tárgyalják a programcsomagok illesztésénél, ill. programcsomag is felhasználói program illesztésénél felmerülő problémákat is.

S u m m a r y

I.N. Paraszuk, I.V. Szegrienko

Application of operating system's properties in constructing program packages

This work deals with the structure and planning of the application program packages. Some concrete comles are given that illustrate the general problems and their solutions. In addition to the describing of the internal structure of the program packages the authors presents short discussion on the problems of connecting program packages with each other or with user programmes