

МУЛЬТИПРОГРАММИРОВАНИЕ НА ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ

Н. С. ИСАЕВА

(МОСКВА)

ВОЗНИКНОВЕНИЕ МУЛЬТИПРОГРАММНЫХ СИСТЕМ

Быстродействие современных вычислительных машин непрерывно возрастает. Медленнореагирующие электромеханические вычислительные устройства были вытеснены машинами, построенными на электронных лампах. Последние в свою очередь все более вытесняются машинами на полупроводниковых элементах. С развитием микроэлектронной технологии новые микроминиатюрные элементы, обладающие высоким быстродействием и малыми размерами, приходят на смену современным полупроводниковым элементам. Быстродействие современных машин исчисляется микросекундами и наносекундами. Однако только за счет введения новых элементов и схем скорость работы машин может быть повышена в настоящее время лишь в десятки раз. Для увеличения же быстродействия машин в сотни раз требуется улучшение их логической организации, т. е. осуществление новых процедур обработки информации. Основной особенностью новой организации работы является мультипрограммирование, состоящее в одновременном выполнении на машине нескольких программ, использующих общую память, а также в параллельном выполнении нескольких команд. Мультипрограммный режим работы обеспечивается соответствующей логической структурой системы. Эта структура реализуется и аппаратными схемами (так называемой встроенной логикой), и программным путем (программируемой логикой).

Целесообразность использования мультипрограммирования связана прежде всего с тем, что скорость работы центральных устройств в машинах значительно превышает скорость работы внешних устройств. По этой причине значительная часть времени «последовательных» машин (т. е. машин, не использующих мультипрограммный режим работы) расходуется на ввод и вывод информации при бездействующем центральном вычислителе. С ростом быстродействия центральных вычислительных устройств процент общего машинного времени, расходуемого на работу внешних устройств, все более увеличивается, так как скорость последних возрастает весьма незначительно. Это особенно заметно при решении задач с небольшим объемом вычислений, но с большим количеством информации, поступающей на обработку (таковы, например, многие экономические задачи).

Простой, возникающие при последовательной работе и приводящие к большим потерям машинного времени из-за значительной разности в скоростях работы внешних устройств и центрального вычислительного устройства, ликвидируются мультипрограммными системами путем совмещения во времени работы внешних и центральных устройств. В частности, если для одной из решаемых задач вводится некоторый массив данных, без которых дальнейшее решение этой задачи невозможно, то одновремен-

но центральное устройство может производить вычисления по другой программе.

Мультипрограммирование позволяет существенно повысить эффективность использования оборудования в вычислительной системе благодаря временной работе нескольких устройств. В идеальном случае время использования каждого устройства мультипрограммной системы должно стремиться к 100%, т. е. к общему машинному времени.

Еще одним преимуществом мультипрограммной работы является возможность получения ответа в реальном масштабе времени, когда вычислительная машина должна немедленно реагировать на сигналы от устройств ввода-вывода и передавать управление соответствующей специальной программе.

ТРЕБОВАНИЯ, ПРЕДЪЯВЛЯЕМЫЕ К МУЛЬТИПРОГРАММНЫМ СИСТЕМАМ

Обеспечение эффективного мультипрограммного режима работы требует решения специфических проблем — размещения различных программ в памяти машины, предохранения этих программ от взаимных помех, обеспечения возможности перехода в любой момент от решения одной задачи к решению другой и т. д. В связи с этим к мультипрограммным системам предъявляется ряд требований.

Прежде всего, для программиста не должны устанавливаться никакие дополнительные ограничения, связанные с мультипрограммированием. Процесс программирования должен происходить без учета одновременного выполнения на машине нескольких программ, составленных, быть может, разными авторами и относящихся к независимым задачам.

В распоряжении программиста должна быть вся память машины. Это означает, что программирование должно осуществляться в математических адресах, независимых от действительных физических адресов, которые будут устанавливаться уже самой системой без вмешательства программиста.

Нельзя требовать от программиста добавочной информации помимо той, которая содержится в самой программе. Однако для более рациональной организации работы системы программисту предоставляется возможность сообщать машине некоторые сведения, касающиеся, например, предполагаемого времени работы программы. Программа будет выполняться и без внесения этих дополнительных сведений, но время ее выполнения может увеличиться.

В мультипрограммной системе должна быть предусмотрена система защиты, исключающая взаимное влияние одной программы на другую из-за того, что все программы используют общую память. Участки памяти, используемые в данный момент одной программой, должны быть заблокированы от обращения в них из другой программы, за исключением случаев, когда в этих участках памяти содержится информация, общая для нескольких программ.

Мультипрограммная система должна устанавливать очередность выполнения различных программ, порядок перехода из одной программы в другую, последовательность использования внешних устройств, обнаруживать ошибки, исследовать их характер и принимать соответствующие меры для продолжения работы. Эти сложные задачи, стоящие перед встроенной и программируемой логикой машины, необходимо решать в минимальное время, чтобы выигрыш от совместного выполнения программ не был ликвидирован затратой времени на организацию мультипрограммного режима работы.

В мультипрограммной вычислительной машине должна быть система прерывания программ, позволяющая осуществлять переход от одной задачи к другой. Это означает, что в машине предусматривается возможность

сохранения программы и всех устройств, участвующих в ее реализации, в том состоянии, в котором они находятся в момент перехода к другой программе. При возврате в прерванную программу зафиксированное состояние системы должно восстанавливаться.

Распределение памяти машины и ее внешних устройств должно производиться в соответствии с действительными требованиями программ и не ограничиваться какими-либо жесткими требованиями, вызванными конструкцией вычислительной системы.

ОСОБЕННОСТИ СУЩЕСТВУЮЩИХ МУЛЬТИПРОГРАММНЫХ СИСТЕМ

Существующие в настоящее время мультипрограммные системы далеко не всегда удовлетворяют перечисленным выше требованиям. Мультипрограммирование осуществляется на самых разных уровнях, и, в зависимости от условий конкретной системы, возникают различные ограничения при реализации общих требований.

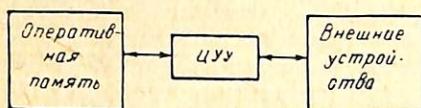


Рис. 1

Вопросы установления очередности выполнения программ, распределение памяти и внешних устройств для быстрого выполнения всего комплекса

программ в разных системах решаются по-разному. В данной консультации рассматриваются наиболее характерные особенности организации таких систем.

Локальный и нелокальный параллелизм. Локальный параллелизм состоит в совмещении во времени выполнения нескольких соседних команд одной программной последовательности (например, во время выполнения одной команды считывается следующая и записывается результат выполнения предыдущей). Это совмещение явилось первой формой параллелизма в системах. Ею обладала, в частности, машина IBMSSEC, в которой одновременно выполнялись три соседние команды, и ряд других машин.

Нелокальный параллелизм означает одновременное выполнение команд, относящихся к разным частям одной программы или даже к нескольким независимым программам.

Следует иметь в виду, что в более поздних типах машин, локальный параллелизм стал применяться одновременно с нелокальным. Так, в чехословацкой вычислительной системе EPOS* во время выполнения продолжительных операций умножения и деления разрешается выполнение некоторых других операций. В вычислительной системе STRETCH (США) одновременно выполняется до одиннадцати соседних команд.

Далее будет рассматриваться только нелокальный параллелизм. Под термином «мультипрограммирование» будет пониматься прежде всего организация нелокального параллелизма.

Организация управления в мультипрограммных системах. В мультипрограммных системах по новому ставится вопрос об управлении различными устройствами машины. Если в «последовательных» машинах центральное устройство управления руководило работой всех других устройств (устройствами ввода, печати и др.), то в мультипрограммных машинах это управление может осуществляться несколькими различными способами.

* Локальный параллелизм, заключающийся в одновременном выполнении нескольких операций одной программы, в системе EPOS называется «внутренним мультипрограммированием».

Прежде всего, конечно, сохраняется случай управления работой всей машины (в том числе и внешних устройств) одним центральным устройством, но на основе разделения времени (рис. 1). Если, например, в оперативную память машины нужно записать некоторый массив информации, или наоборот, выдать массив информации из памяти, то выполнение основной (рабочей) программы в центральном вычислителе будет прерываться на время записи (или считывания) каждого слова информации, а также на время выполнения соответствующей стандартной программы приема (или выдачи) этого слова. Это, конечно, снижает производительность центрального вычислителя. Приостановки для записи (считывания) каждого слова в память снижают, конечно, быстродействие системы, но в некоторых машинах это допускается. Кроме того, при больших скоростях работы центральных устройств машины и, следовательно, малом времени обращения к оперативной памяти на запись или считывание слова уходит сравнительно мало времени.

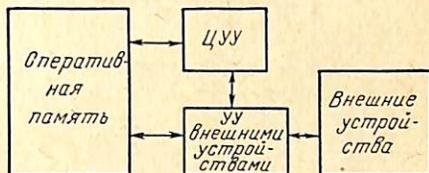


Рис. 2

Наряду с этим существует способ централизованного управления работой внешних устройств (в том числе и внешней памяти), при котором в машине имеется специальное, в некоторых случаях сравнительно несложное, устройство управления внешним оборудованием. Это устройство управления может адресоваться к оперативной памяти так же, как и Центральное устройство управления (ЦУУ), и независимо от последнего производить запись и считывание информации. Таким образом, работа центрального вычислителя прерывается только для включения того или иного устройства, а в дальнейшем осуществляется параллельно с работой данного устройства (рис. 2). Реализация этого способа в каждой конкретной системе может быть различной. Так, в системе STRETCH одновременно действующие устройства ввода-вывода связаны с памятью и с центральным процессором (ЦП) через Основное устройство обмена (ЦП включает в себя Центральное устройство управления, арифметические устройства и ряд логических устройств). Основное устройство обмена, получив от ЦП команду, способно далее самостоятельно производить не только запись в память и считывание из нее, но и контролировать и исправлять возникающие при этом ошибки, формировать слова, которые могут быть затем записаны в память.

В системе IBM360 (США) внешние устройства связаны с оперативной памятью и с центральным вычислителем через мультиплексорный и селекторный каналы. Эти каналы являются устройствами, которые способны записывать информацию в память и считывать ее без участия центрального вычислителя.

Такая организация обеспечивает высокую степень временного перекрытия в работе устройств ввода-вывода и центрального вычислителя и значительно повышает общее быстродействие системы.

Возможна и такая организация мультипрограммной системы, при которой каждое устройство системы имеет собственное устройство управления и может работать автономно.

Примером такой системы является машина ГАММА60 (Франция). Устройства (или группы устройств) в этой машине, способные работать автономно с момента, когда получена команда, определяющая, что делать, и команда, задающая адреса исходных данных (операндов) в памяти на сердечниках, называются элементами. Элементы могут быть центральными (арифметическое устройство, логическое устройство и т. д.) и перифе-

рийными (устройства ввода-вывода и внешняя память). Элементы способны работать одновременно. Их деятельность координируется центральным устройством (рис. 3). Каждый элемент содержит так называемую «местную программу», которая после получения команды от центрального устройства управляет автономной работой элемента. Информация в ГАММА60 не может передаваться непосредственно от элемента к элементу. Передача всегда осуществляется через память, и при необходимости



Рис. 3

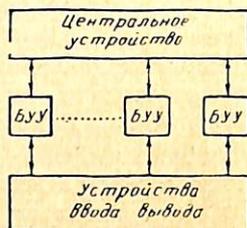


Рис. 4

частых переключений между элементами, время, затрачиваемое на управление работой машины, может быть значительным.

Интересная система управления устройствами ввода-вывода была предложена при разработке сверхбыстродействующей вычислительной машины для Control Data Corporation (США). Все операции ввода-вывода осуществляются, минуя быстродействующий центральный вычислитель, специальными буферными управляющими устройствами (рис. 4). Их количество может достигать до десяти. БУУ представляют собой небольшие вычислительные машины, с собственной памятью объемом по 4000 двенадцатиразрядных слов и с собственным набором команд. При необходимости связаться с центральным вычислителем буферное управляющее устройство прерывает его. Однако каждое БУУ может осуществлять обмен информацией между своей собственной небольшой памятью и общей памятью машины, не прерывая центральный вычислитель, т. е. одновременно с выполнением основных вычислений, связанных непосредственно с решением задач. Основная задача организации мультипрограммирования состоит в сглаживании различия между скоростями ввода-вывода и основных вычислений.

Нужно отметить, что стоимость всех буферных управляющих устройств не превышает 5% всей стоимости системы. В то же время использование БУУ в структуре системы существенно уменьшает число обращений центрального вычислителя к главной памяти, т. к. он вообще не осуществляет ввод и вывод информации. Таким образом, рассмотренная организация мультипрограммной системы может быть признана весьма целесообразной.

Итак, можно отметить четыре характерных формы организации управления в мультипрограммных системах:

- а) одно центральное устройство управления руководит работой всех остальных устройств машины, как центральных, так и периферийных;
- б) общее руководство работой системы осуществляется центральным устройством управления, но есть вспомогательное централизованное управление устройствами ввода-вывода и внешней памятью;
- в) все самостоятельные устройства системы, как центральные, так и периферийные, имеют собственное местное управление. Центральное устройство управления координирует деятельность всех остальных устройств;
- г) производством вычислений в машине руководит центральное устройство управления, операции ввода-вывода осуществляются под руководством специальных буферных управляющих устройств.

Разделение времени. При разработке мультипрограммных систем возникает ряд проблем, связанных с распределением времени для выполнения каждой независимой программы. Имеется в виду не только определение момента, когда та или иная программа будет включаться для выполнения или прерываться, но и установление очередности при удовлетворении запросов на внешние устройства машины от этих программ.

Чем определяется момент, когда выполнение одной программы прерывается и управление передается другой программе? Этот вопрос решается одним из трех следующих способов.

1. Выполнение текущей программы прерывается, когда эта программа не может или не должна выполняться. Это может произойти, в частности, когда в оперативную память машины не введена информация, необходимая для продолжения решения задачи.

Программа прерывается также при обнаружении какой-либо ошибки в вычислениях или в работе схем. В «последовательных» машинах в этом случае происходит останов машины. В мультипрограммных системах останавливается только программа, при выполнении которой появилась ошибка. Машина в этом случае может перейти к выполнению другой программы.

При одновременном выполнении нескольких программ может оказаться необходимым ограничить время непрерывного выполнения каждой из них, чтобы одна программа не слишком задерживала выполнение других программ.

Примером такого переключения между программами является система ATLAS (Англия). Казалось бы, что для увеличения перекрытия операций ввода-вывода и центральных вычислений следовало бы заранее так подбирать задачи, чтобы, пока центральный вычислитель производит вычисления по одной программе, по другой программе осуществлялась операция выдачи результатов и по третьей — ввод информации. Таким подбором можно было бы понизить скорость запроса или выдачи информации центральным вычислителем до тех скоростей, с которыми работают внешние устройства. Однако, даже если не касаться трудностей, связанных с указанным подбором программ, реализация такого решения неэффективна, так как связана с частыми переключениями машины с одной программы на другую, на что уходит сравнительно много времени (такое переключение в системе ATLAS занимает около 2,5 мсек). Это время тратится в основном на работу управляющей программы и аппаратных схем по запоминанию и восстановлению содержимого всех рабочих регистров машины, общих для различных основных программ.

Чтобы избежать частых переключений, в основной памяти машины выделяются две специальные области: область ввода и область вывода. Они заполняются информацией и освобождаются от нее медленнодействующими периферийными устройствами. Центральный вычислитель требует и выдает информацию с различными скоростями, и периферийные устройства не могут передавать информацию со скоростью, требуемой им. Применение областей ввода и вывода сглаживает различие в скоростях, с которыми центральный вычислитель выдает или требует информацию. Наиболее эффективным в данном случае будет сокращение до минимума числа переключений между основными программами. Одновременно будут происходить вычисления по одной задаче и ввод-вывод информации для других задач.

Если переключение происходит, необходимо решить важный вопрос, какая программа будет выполняться следующей. В машине ATLAS программы обычно выполняются в порядке их поступления. Очередность задается программно (т. е. приоритет каждой программы определяется за-

ранее). Однако этот порядок (иерархия приоритетов) может быть изменен в зависимости от занятости частей системы. Программа, планирующая работу машины, может, например, временно остановить какую-либо программу, чтобы дать возможность другим программам максимально загрузить периферийные устройства, если они не работают.

В системе STRETCH установление очередности основных программ (т. е. назначение приоритетов) производит управляющая программа, которая на основе имеющихся сведений о размере требуемой памяти, предполагаемом времени выполнения программы и некоторой другой дополнительной информации определяет наиболее подходящий момент для включения той или иной программы в фазу выполнения.

Необходимо отметить, что в мультипрограммных системах обычно различаются не только приоритеты программ, но и приоритеты тех или иных устройств. При этом в некоторых системах рассматриваются только приоритеты программ, в других — только приоритеты устройств, и, наконец, есть системы, где учитываются те и другие приоритеты.

Так, в машине GAMMA60 своеобразная организация управления, при которой каждый элемент действует автономно, определяет и соответствующий порядок выполнения различных программ. Приоритеты для программ не задаются, они назначены только для элементов, причем наиболее медленнодействующие из них имеют наивысший приоритет. Центральное устройство управления удовлетворяет запросы, посылаемые элементами, в соответствии с их приоритетами, что обеспечивает наиболее эффективную загрузку всех элементов машины.

2. Другой способ переключения с одной основной программы на другую осуществляется в машинах Honeywell 800 и Honeywell 1800 (США). Коммутатор программ подключает в цикле для выполнения последовательно по одной команде от каждой программы. Одновременно может, таким образом, выполняться до 7 программ. Если какая-либо программа задерживается из-за того, что нужное ей периферийное устройство занято, остальные программы начинают выполняться быстрее, пока это устройство не освободится и не будет восстановлен прежний порядок.

3. Интересным способом переключения между основными программами является способ, основанный на оценке моментального приоритета программ. Он реализован в вычислительной машине EPOS. Переключение программ осуществляется специальным организующим устройством. Условия, на основании которых организующее устройство принимает решение о переключении программ, вытекают из состояния отдельных устройств периферийного оборудования и из состояния самих программ. Для каждой выполняемой программы во время исполнения каждой команды вычисляется значение ее моментального приоритета. Перед выполнением каждой команды организующее устройство определяет, разрешить ли данной программе продолжать работу или включить другую программу, имеющую в данное время максимальный моментальный приоритет. При оценке моментального приоритета учитывается как информация от оператора, дающего программе основной субъективный приоритет, так и собственно состояние программы в процессе ее выполнения. Кроме того, принимается во внимание общее состояние различных устройств системы. Устройство может быть:

а) в активном состоянии (требует команду), т. е. готово принять команду, выполнение которой предотвратит последующую задержку в его работе,

б) заблокировано, т. е. занято другой операцией и не в состоянии еще приступить к выполнению требуемой команды,

в) нейтрально, когда устройство не активно и не заблокировано.

Вопрос сохранения текущей ситуации на машине при переключении с одной программы на другую решается здесь следующим образом: каждый регистр центрального вычислителя, в котором сохраняются некоторые результаты после выполнения команды, оборудован специальной запасной памятью, в которой запоминается содержание этого регистра при переключении программ. В EPOS одновременно могут выполняться до пяти независимых программ. И соответственно каждая запасная память имеет 5 секций, предназначенных постоянно этим пяти программам. Это ведет, разумеется, к усложнению оборудования, однако, скорость запоминания и восстановления текущей ситуации может быть значительной, так как не требуется переписывать содержимое всех регистров в память и затем из памяти в регистры.

Разделение места. Эта задача включает в себя распределение памяти различных видов, а также распределение периферийных устройств. В соответствии с требованием, предъявляемым к мультипрограммным системам, распределение места в машине не должно быть неизменным и обусловленным только конструктивными особенностями машины. В мультипрограммных машинах можно выделить два основных метода распределения места.

1. Относительно постоянное распределение, когда память и внешние устройства распределяются между программами при их вводе. Так в машине KDF9 (Англия) большинство внешних устройств в каждый момент времени относится к одной определенной программе и недоступны для других программ. В системе EPOS устройства ввода-вывода, оперативная, внешняя и вспомогательная память делятся между программами в соответствии с действительными потребностями этих программ. При дальнейшей работе это распределение сохраняется. Оно может быть изменено оператором при вводе новой программы.

2. Переменное (динамическое) распределение, когда память и периферийные устройства отводятся программе только тогда, когда данная программа включается в фазу выполнения. Никакого присваивания действительных адресов памяти, определения адресов различных устройств, необходимых программе, не производится заранее. Такая организация обеспечивает наибольшую гибкость и экономность в распределении места. Распределение места по этому принципу осуществляется в машине STRETCH. В машине ATLAS распределение места также является переменным. Программа-супервайзер* отводит место для рабочей программы в памяти на сердечниках перед включением ее в работу. Ввод данных и вывод результатов происходит через области ввода и вывода основной памяти, поделенные между периферийными устройствами в зависимости от скорости поступления или выдачи информации ими, но не между программами. Если же возникает необходимость закрепить некоторое периферийное устройство за одной из основных программ, то это может быть сделано оператором.

Разумеется, принцип динамического распределения места имеет очевидные преимущества, так как позволяет более рационально использовать память машины и максимально загружать ее устройства. Однако, относительно постоянное распределение места может быть оправдано в тех случаях, когда структура самой машины предполагает, что переключения между основными программами происходят регулярно или достаточно часто, и постоянно сопутствующее им перераспределение места становится весьма сложным.

* Как правило, работу всей вычислительной машины координирует управляющая программа, называемая в некоторых машинах программой-диспетчером, в системах ATLAS и STRETCH — программой-супервайзером, в KDF9 — директором.

При создании мультипрограммной системы большую роль играет метод использования памяти. В системе ATLAS память на магнитных сердечниках и на барабанах образует память одного уровня, и адрес в команде относится к этой объединенной памяти. Память разбита на страницы по 512 слов в каждой. Фактическое положение каждой страницы в памяти не фиксировано. Если требуемая страница не находится в данный момент в памяти на сердечниках, то она вызывается с барабана на пустую страницу. После этого делается запись какой-либо страницы на барабан, чтобы организовать новую пустую страницу, причем выбором этой страницы занимается некоторая самообучающаяся программа. Эта программа предсказывает, какая страница из памяти на сердечниках не понадобится в течение самого длительного по сравнению с другими страницами отрезка времени. Такая организация памяти при наличии непрямои адресации позволяет супервайзеру осуществлять динамическое перераспределение областей памяти, менять их физические адреса и производить автоматическую передачу информации с барабана и на барабан.

С точки зрения организации памяти известный интерес представляют машины с магазинной организацией памяти (B5000, KDF9). Так, в машине B5000 (США) имеется специальная таблица ссылок, с помощью которой может осуществляться обращение к главной памяти. В 12-разрядном программном слове указывается код операции или адрес строки в таблице ссылок. В каждой строке таблицы ссылок (48 разрядов) может находиться либо сам операнд, либо его адрес в большой памяти, либо описатель массива информации или сегмента программы (его начальный адрес, длина). В таблице ссылок можно указать, где находится массив информации, причем массив может быть любого размера (в пределах оперативной памяти). Таким образом, несмотря на короткое программное слово, имеется возможность использовать достаточно большую память, так как ее фактический адрес находится в 48-разрядной строке таблицы ссылок. Операнд, выбранный или непосредственно из таблицы ссылок, или из памяти, помещается в верхнюю ячейку магазина. Динамика памяти осуществляется здесь достаточно просто. Управляющая программа может вызвать любой массив на любое место в памяти, причем в таблице ссылок указывается, находится ли массив в оперативной памяти, или во внешней памяти. Защиты памяти почти не требуется, так как границы массива, используемого задачей, известны.

Два способа реализации мультипрограммирования — программный и схемный. Мультипрограммная система требует сложной организации управления: необходимо устанавливать соответствие математических и физических адресов, распределять память и периферийные устройства, осуществлять ввод и вывод информации, перераспределять информацию в памяти, осуществлять защиту программ от взаимных помех, исследовать различные причины прерываний и предпринимать соответствующие им действия и т. д. Какая часть этой работы должна осуществляться управляющей программой, какая выполняться аппаратной, «встроенной» в машину логикой? В настоящее время нет единого мнения о целесообразном разделении функций между этими средствами. Тем не менее, выбор того или иного способа отражается и на стоимости системы, и на ее быстродействии, и на логической гибкости. Аппаратное осуществление тех или иных функций требует увеличения оборудования и вызывает увеличение стоимости машины. При использовании программируемой логики реализуется большая гибкость и универсальность системы, возможна при желании полная замена одной управляющей программы на другую. Однако время, необходимое для работы управляющей программы, легко может оказаться недопустимо большим.

Мультипрограммирование можно осуществить программным путем на машинах с весьма небольшим количеством аппаратных возможностей. Однако в большинстве крупных мультипрограммных систем используется сочетание аппаратной и программной логики. Как правило, имеется аппаратная система прерывания программ, фиксирующая сигналы прерывания от устройств ввода-вывода, от некоторых ошибок в основных программах, а также сигналы прерывания, вызванные сбоями в машине.

Такого рода система имеется, например, в машине STRETCH. Использование же сигналов прерывания для мультипрограммирования осуществляется программным путем. Система защиты одной программы от другой в STRETCH также выполнена аппаратно. В каждый данный момент запрещаются обращения внутрь некоторой области, границы которой указаны. Изменение этих границ в процессе выполнения программ возлагается на управляющую программу. Кроме того, аппаратным способом в STRETCH организована служба времени, позволяющая автоматически прерывать работу программы, если она полностью использовала свой лимит времени. Программное наблюдение за временем выполнения каждой программы было бы, вероятно, более сложным. Программа-супервайзер, используя имеющиеся в ее распоряжении аппаратные средства, управляет распределением места и времени для каждой основной программы.

В машине ATLAS мультипрограммирование также обеспечивается сочетанием программы-супервайзера и «встроенной» логики. Супервайзер руководит осуществлением операций ввода и вывода, организует обмен информацией между оперативной памятью и внешними накопителями, исследует причины прерываний и определяет характер дальнейшей работы.

Как уже было указано, при распределении времени большое значение имеет приоритет программ и устройств. Этот приоритет может также задаваться программно и схемно. В большинстве машин приоритет программ задается программным путем или определяется оператором при вводе (например, в KDF9), приоритет же устройств чаще всего задается аппаратно (например, в GAMMA60). Иногда оба эти ряда очередности действуют одновременно.

Система EPOS является примером осуществления мультипрограммирования исключительно за счет «встроенной» логики. Приоритеты программ здесь также определяются схемным путем. В организующем устройстве машины реализуется ряд алгоритмов, определяющих поведение всей системы. Целесообразность такого решения, может быть, покажется спорной, однако, последовательное применение «аппаратного» подхода для решения всех задач мультипрограммирования представляет несомненный интерес.

НЕКОТОРЫЕ РЕЗУЛЬТАТЫ ИСПОЛЬЗОВАНИЯ МУЛЬТИПРОГРАММНЫХ СИСТЕМ

Мультипрограммные системы начали эксплуатироваться сравнительно недавно. Однако уже получены некоторые результаты, свидетельствующие о значительном росте пропускной способности оборудования при параллельном решении нескольких задач. Так, Компания страхования жизни (США), имеющая две системы Honeywell 800, проводила при решении задач систематические записи «последовательного» и «мультипрограммного» времени. Было установлено, что за период с марта по июнь 1962 года пропускная способность машин (отношение времени последовательного выполнения программ к мультипрограммному времени выполнения тех же программ) увеличилась с 1 до 1,13. В ближайшем будущем ожидается рост пропускной способности до 1,5—2,5.

Известны также результаты испытаний, проведенных на одной из машин STRETCH фирмы IBM. Сравнивались времена прохождения различных комплектов программ через машину при однопрограммном и мультипрограммных способах выполнения. Однопрограммный способ в данном случае означал последовательное выполнение одной программы за другой, но с возможностью одновременного ввода других программ, установки и снятия магнитных лент и параллельного выполнения некоторых других операций. Мультипрограммный способ означал выполнение некоторого комплекта программ одновременно; все виды параллелизма, разрешенные для однопрограммного способа, использовались и при мультипрограммном способе.

В этих испытаниях брались три основных комплекта задач: 1) задачи, интенсивно использующие центральные устройства; 2) задачи, интенсивно использующие магнитные ленты, умеренно использующие центральные устройства и вообще не использующие диски; 3) ряд задач, состоящий из задач первого и второго типа.

Было установлено, что отношение времени однопрограммного выполнения к мультипрограммному составило для трех комплектов задач соответственно: 1,3; 1,5 и 2.

Как видно, наибольший выигрыш получается при одновременном выполнении задач различных типов. Это и понятно, так как повышение скорости решения задач при мультипрограммировании достигается прежде всего за счет совмещения оперативных вычислений и работы внешних устройств. По мере уменьшения количества запросов от программ на использование внешних устройств мультипрограммный режим работы по своей производительности все более приближается к однопрограммному.

Несмотря на ряд неудобств, связанных с предварительным подбором комплекта программ, общий выигрыш, получаемый в этом случае от мультипрограммирования, позволяет считать такой подбор целесообразным, хотя и не обязательным.

При современном состоянии вычислительных машин, для которого характерна существенная разница в скоростях центрального вычислителя и внешних устройств, мультипрограммный режим работы является одним из основных средств повышения производительности машин.

ЛИТЕРАТУРА

1. Д. Е. Киллнер. Характеристики вычислительных машин второго десятилетия. Киберн. сб., 1962, № 5.
2. Курс программирования для Гамма 60. Пер. с фр. М., Изд-во иностр. лит., 1962.
3. Е. Ф. Кодд, Э. С. Лоури, Э. Мак-Донаф, С. А. Скэлзи. Мультипрограммирование для машины СТРЕТЧ. Киберн. сб., 1961, № 2.
4. Т. Килбурн, Д. Ховарт, Р. Пэйп. Программа-супервайзер для машины АТЛАС. Киберн. сб., 1963, № 6.

Поступила в редакцию
12 VI 1965