

УДК 681.3.01

*А. Г. МУРАШКО*, канд. техн. наук, *С. К. КОЛУБАЙ*

## **ОБ ОДНОМ ПРИНЦИПЕ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА**

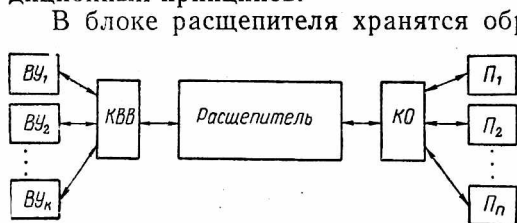
Поиск новых принципов структурной и программной организации вычислительных систем (ВС), являющихся частью человеко-машинных, связан с успехами микроэлектроники в создании миниатюрных средств переработки данных — так называемых микропроцессоров.

Организация вычислительного процесса, по нашему мнению, — один из центральных вопросов организации ВС на базе микропроцессоров. От решения его в значительной степени зависят принципы структурной и программной организации, положенные в основу построения ВС, а также эффективность функционирования ВС.

Представляется целесообразным избрать такую организацию вычислительного процесса, которая наиболее близка к организации вычислительного процесса человеком. Результаты экспериментирования с испытуемыми по определению их действий при решении задач с применением известных численных методов (задачи численного дифференцирования и интегрирования, решения систем линейных алгебраических уравнений, обыкновенных дифференциальных уравнений, дифференциальных уравнений в частных производных, интегральных уравнений и др.) позволяют сформулировать принцип организации вычислительного процесса в параллельных ВС на базе микропроцессоров, названный принципом расщепления программ.

Нами используются обозначения и терминология, принятые в работах В. Е. Котова и А. С. Нариньяни [1, 2].

Принцип расщепления программ удобно рассмотреть при помощи структуры гипотетической вычислительной системы (рис. 1), состоящей из расщепителя, канала обработки КО, процессоров  $\Pi_1, \dots, \Pi_n$  канала ввода — вывода КВВ и внешних устройств  $ВУ_1, \dots, ВУ_k$ . Предполагается, что мультипроцессорная вычислительная система МВС с расщеплением программ относится к классу мультипрограммных многопроцессорных систем. Все структурные единицы МВС с расщеплением программ, исключая блок расщепителя, построены с использованием традиционных принципов.



Вычислительная система с расщеплением программ.

В блоке расщепителя хранятся обрабатываемые программы  $A^1, \dots, A^l$ , исходные данные и получаемые результаты, т. е. все то, что в обычной МВС хранится в оперативной памяти (ОП). Однако в отличие от обычной МВС блок расщепителя не только хранит программы, а выполняет следующие функции.

Для всякого момента времени  $t$  расщепитель на основании анализа информационно-управляющих связей в программах  $A^i$  ( $i = \overline{1, l}$ ) осуществляет расщепление каждой из программ на два непересекающихся множества: множество  $^*A_t^i$  готовых и множество  $^{\neg}A_t^i$  неготовых к выполнению команд. При этом команда программы  $A^i$  называется готовой к выполнению в момент времени  $t$ , т. е.  $a \in ^*A_t^i$ , если в следующий момент времени  $t + 1$  она может быть включена в работу. Под включенной в работу понимается команда, переданная из расщепителя на процессор или на внешнее устройство.

Пусть  $^+A_t^i$  — множество команд программы  $A^i$ , включенных в момент времени  $t$ , тогда  $\forall t \forall a (a \in ^+A_t^i \Rightarrow a \in ^*A_{t-1}^i)$ .

Каждое из множеств  $^*A_t^i$  состоит из двух непересекающихся подмножеств: множества  $^*B_t^i$  команд ввода — вывода и множества  $^*O_t^i$  команд обработки, причем одно из них или одновременно оба могут быть пустыми.

Внешние устройства распределяются по программам. При  $B_t^i \neq \emptyset$  и свободном соответствующем ВУ КВВ запускает его в работу. Процессоры  $\Pi_1, \dots, \Pi_n$  с расщепителем связывает КО следующим образом. Пусть программа  $A^i$ , имеющая меньший номер  $i$ , наделена более высоким приоритетом выполнения. Каждый из свободных процессоров выставляет запрос на получение инструкции обработки на КО, который при наличии хотя бы

одного такого запроса от процессоров выставляет запрос на получение инструкции от расщепителя.

Пусть запрос КО на получение инструкции от расщепителя поступил в момент  $t_1$ . Если  $\exists i (*O_i^i \neq \emptyset)$ , то расщепитель формирует инструкцию обработки. При этом, если  $\exists i \exists j (*O_i^i \neq \emptyset \wedge *O_j^j \neq \emptyset \wedge i \neq j)$ , то для формирования инструкции обработки выбирается команда из множества  $*O_i^i$  такого, что  $\forall j (j < i \Rightarrow *O_j^j = \emptyset)$ . Обозначим  $|A|$  количество элементов множества  $A$ , тогда, если  $|*O_i^i| > 1$ , то выбор конкретной команды  $a \in *O_i^i$  для формирования инструкции обработки пусть производится произвольным образом. Инструкция обработки имеет следующий формат:

Код операции	Данное 1	Данное 2	Адрес результата
--------------	----------	----------	------------------

После формирования инструкции обработки расщепитель передает ее через КО на один из свободных процессоров, который переходит при этом в состояние «занято» и выполняет полученную инструкцию. Затем процессор выставляет на КО запрос на передачу результата в расщепитель в следующем формате:

Адрес результата	Результат
------------------	-----------

Если в этот момент времени КО не занят передачей инструкции обработки, то он передает результат в блок расщепителя, где по адресу результата производится его запись. Процессор, передав результат, переходит в состояние «свободно» и выставляет запрос на получение инструкции обработки.

С целью повысить производительность такой МВС блок расщепителя может параллельно с работой процессоров организовывать небольшую очередь инструкций обработки с учетом приоритетов программ. В этом случае при наличии запроса от КО на получение инструкций и хотя бы одной инструкции в очереди запрос сразу удовлетворяется, т. е. процессор, запросивший инструкцию, сразу же ее получает.

Возможна также и иная организация обмена инструкциями между расщепителем и процессорами, например передача инструкций обработки на КО и получение ответов с КО по нескольким линиям связи одновременно и т. п.

Остановимся подробнее на функции расщепления программ и процессе счета по программам.

Пусть в начальный момент времени  $t = 0$  осуществлен ввод в блок расщепителя программ  $A^1, A^2, \dots, A^l$ . Число  $l$  программ, введенных в блок расщепителя, ограничено объемом памяти. Для определенности установим принятый ранее порядок выполнения программ и заметим, что установка системы приоритетов

выполнения программ в рассматриваемой МВС не обязательна. Определена она лишь для удобства рассмотрения организационных решений задач на МВС.

Блок расщепителя, анализируя информационно-управляющую связь в каждой из программ, определяет множества  $*A_0^i$  ( $i = \overline{1, l}$ ). Для программ, не зависящих друг от друга, например программы разных задач, выполняется условие  $\forall i (*A_0^i \neq \emptyset)$ . В противном случае, если  $\exists i (*A_0^i = \emptyset)$ , программа  $A^i$  не может быть включена в работу. Это связано с тем, что для независимых программ, обрабатываемых в блоке расщепителя, выполняется условие  $\forall t (*A_t^i = \emptyset \wedge {}^pA_t^i = \emptyset \Rightarrow *A_{t+1}^i = \emptyset)$ , где  ${}^pA_t^i$  — множество выполняющих команд программы  $A^i$  в момент времени  $t$ .

После определения множеств  $*A_0^i$  начинается процесс счета по программам, который состоит в передаче готовых к выполнению команд на процессоры или внешние устройства МВС. Другими словами, начать счет — значит определить, какие из команд множеств  $*A_0^i$  могут быть включены, т. е. необходимо определить множества  ${}^+A_0^i$ . Выбор множеств  ${}^+A_0^i$  из множеств  $*A_0^i$  осуществляется с учетом приоритетов выполнения программ и начального состояния системы, однако может быть и произвольным.

Под состоянием системы понимается прежде всего число свободных процессоров и внешних устройств. В момент времени  $t = 0$  ни одна из программ не выполняется, поэтому состояние МВС определено начальной конфигурацией МВС и не зависит от программ, которые предстоит ей выполнить.

Для определенности полагаем, что для всякого момента времени  $t$  множества  ${}^+A_t^i$  выбираются, исходя из условий максимальной загрузки оборудования и приоритетности выполнения программ. Таким образом определяется множество  ${}^+A_0$  команд включенных в работу в момент времени  $t = 0$ , т. е.  ${}^+A_0 = \bigcup_i {}^+A_0^i$  и они же определяют множество  ${}^pA_0$  команд, выполняющих в момент времени  $t = 0$ , т. е.  ${}^pA_0 = {}^+A_0$ .

Функционирование описываемой МВС характеризуется наличием так называемого равновесного состояния, при котором либо нет готовых команд для свободных процессоров и внешних устройств, либо нет свободных процессоров и внешних устройств для готовых команд. Это состояние не изменится до тех пор, пока хотя бы одна из работающих команд  $a \in {}^pA_0$  не закончит выполнение.

Предположим, что в момент времени  $t = 1$  закончилось выполнение некоторое множество  ${}^-A_1$  команд. Команды, принадлежащие множеству  ${}^-A_t$ , назовем выключающимися в момент времени  $t$ . Множество  ${}^-A_1$  может не совпадать с множеством  ${}^pA_0$ , так как совсем не обязательно, чтобы все команды, включенные в момент

времени  $t = 0$ , имели одинаковое время выполнения, следовательно,  $\bar{A}_1 \subseteq {}^p A_0$ .

После того как команды множества  $\bar{A}_1$  передадут результаты в блок расщепителя, освободятся соответствующие процессоры и внешние устройства. При этом для общности рассуждений мы не учитываем, что на выполнение команды ввода — вывода, как правило, затрачивается значительно больше времени, чем на выполнение команд обработки.

По множеству  $\bar{A}_1$  блок расщепителя на основании результатов анализа информационно-управляющих связей в программах определит множества  $\bar{A}_1^i (i = \bar{1}, \bar{l})$ , так как  $\bar{A}_1 = \bigcup_i \bar{A}_1^i$ . Это в свою очередь даст возможность оценить множества  ${}^* A_1^i$  как  ${}^* A_1^i = ({}^* A_0^i \setminus {}^+ A_0^i) \cup (\bar{A}_0^i \setminus \bar{A}_1^i)$ .

После определения множества  ${}^* A_1^i$  с учетом приоритетов программ и условия максимальной загрузки оборудования из элементов множеств  ${}^* A_1^i$  будет сформировано множество  ${}^+ A_1$  команд, включающихся в момент времени  $t = 1$ . Таким образом, множество  ${}^p A_1$  команд, выполняющихся в момент времени  $t = 1$ , станет равным  ${}^p A_1 = {}^p A_0 \setminus \bar{A}_1 \cup {}^+ A_1$ . В этот момент в функционировании МВС снова наступит равновесное состояние, которое будет продолжаться до следующего момента времени  $t = 2$ .

В момент времени  $t = 2$  закончится выполнение некоторых команд из множества  ${}^p A_1$ , которые и образуют множество  $\bar{A}_2$  команд, выключившихся в данный момент времени. По множеству  $\bar{A}_2$  блок расщепителя определит множества  $\bar{A}_2^i$ , как описано для множеств  $\bar{A}_1^i$ . Далее будут сформированы множества  ${}^* A_2^i = {}^* A_1^i \setminus {}^+ A_1^i \cup \bar{A}_1^i \setminus \bar{A}_2^i$  и выбраны из них подмножества  ${}^+ A_2^i$ , которые составят множество  ${}^+ A_2$  команд, включившихся в работу в момент времени  $t = 2$  и т. д.

Таким образом, процесс счета по каждой из программ, находящейся в блоке расщепителя, может быть описан следующими рекуррентными соотношениями:

$$\begin{aligned} & \bar{A}_t^i \subseteq {}^p A_{t-1}^i; \quad \bar{A}_t^i = F^i(\bar{A}_{t-1}^i, -A_t^i); \quad {}^* A_t^i = \\ & = {}^* A_{t-1}^i \setminus {}^+ A_{t-1}^i \cup \bar{A}_{t-1}^i \setminus \bar{A}_t^i; \quad {}^+ A_t^i \subseteq {}^* A_t^i; \quad {}^p A_t^i = {}^p A_{t-1}^i \setminus -A_t^i \cup {}^+ A_t^i; \end{aligned}$$

с учетом того, что в начальный момент времени  $t = 0$  действуют следующие соотношения:

$$-A_0^i = \emptyset; \quad \bar{A}_0^i = F_0^i(A^i); \quad {}^* A_0^i = A^i \setminus \bar{A}_0^i; \quad {}^p A_0^i = {}^+ A_0^i.$$

Из рекуррентных соотношений следует, что процесс счета для программы  $A^i$  будет конечным, если для некоторого момента времени  $t$  выполняется следующее условие:  ${}^* A_t^i = \emptyset \ \& \ {}^p A_t^i = \emptyset$ . Этот момент времени называем моментом окончания работы программы  $A^i$ .

Таким образом, в процессе функционирования МВС на блоке расщепителя возлагается решение следующих основных задач:

1. Хранение программ  $A^i (i = \overline{1, l})$ , данных и результатов.
2. Определение для каждого момента времени  $t$  множеств  $A_t^i$ , готовых к выполнению команд.
3. Формирование множеств  $A_t^i$  с учетом запросов каналов обработки и ввода — вывода.
4. Наблюдение за состоянием множеств  $A_t^i$  и  $P A_t^i$  для определения момента окончания работы программ.
5. Осуществление повторного включения команд программ в работу при выходе из строя процессоров или внешних устройств.

Эффективность МВС тем выше, чем быстрее блок расщепителя справляется со своими задачами. При проектировании блока расщепителя следует потребовать, чтобы время решения задач 2 и 3 было значительно меньше времени выполнения одной команды. В противном случае трудно ожидать высокой загрузки параллельно работающих процессоров и внешних устройств.

Среди перечисленных задач, которые должен решать блок расщепителя, наиболее сложной и принципиально новой автор считает задачу определения множеств готовых к выполнению команд. Эта задача возведена в принцип расщепления программ, под которым понимается такой принцип функционирования вычислительной системы и организации вычислительного процесса, в результате которого ВС путем анализа информационных и управляющих связей в программах, производится расщепление программ на отдельные множества готовых, готовых к выполнению команд.

При практической реализации принципа расщепления программ традиционные методы составления и выполнения программ на современных вычислительных системах, а также последовательно-алгоритмический принцип функционирования ВС неприемлемы.

Выходом из этого положения, на наш взгляд, является программно-аппаратный подход к решению перечисленных задач с использованием новых принципов структурной и программной организации вычислительных систем.

Список литературы: 1. Котлов В. Е. Теория параллельного программирования. — «Кибернетика». 1974, № 1, с. 1—16, № 2, с. 1—18. 2. Нариньяни А. С. Теория параллельного программирования. — «Кибернетика», 1974, № 3, с. 1—15, № 5, с. 1—14.