

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ДАННЫХ
НА ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ СУММА

И.Н.Кашун, Л.А.Седухина

Однородная вычислительная система СУММА предназначена для решения задач, представленных в виде системы асинхронных взаимодействующих процессов [1]. Класс таких задач охватывает довольно широкую область: сбор и обработка информации, управление в реальном времени и т.д. Для указанного класса задач характерны режимы матричной и конвейерной обработки, порознь применяющиеся во многих известных вычислительных системах [2]. В данной работе демонстрируется возможность организации матричной и конвейерной обработок в рамках операционной системы ОВС СУММА.

1. Главной особенностью матричного (векторного) способа обработки является то, что одна и та же процедура должна выполняться над большой совокупностью данных, распределенных между элементарными машинами (ЭМ) системы. В этом случае достаточно иметь только один управляющий процесс, который дешифрирует одиночный поток команд и вырабатывает управляющие воздействия для совокупности элементарных управляющих процессов, протекающих в указанных ЭМ системы. По окончании процедуры, вызванной получением управляющего воздействия, каждый управляемый процесс извещает управляющий. Вслед за этим инициируется новая матричная операция.

Схема реализации режима матричной обработки информации показана на рис.1. Структура управляющего процесса и процессов в каждой машине представлена ниже.

Процесс управления:

```
WRITE DATE  
START PROCESS  
WAIT S
```

МАТРИЧНЫЙ СПОСОБ ОБРАБОТКИ

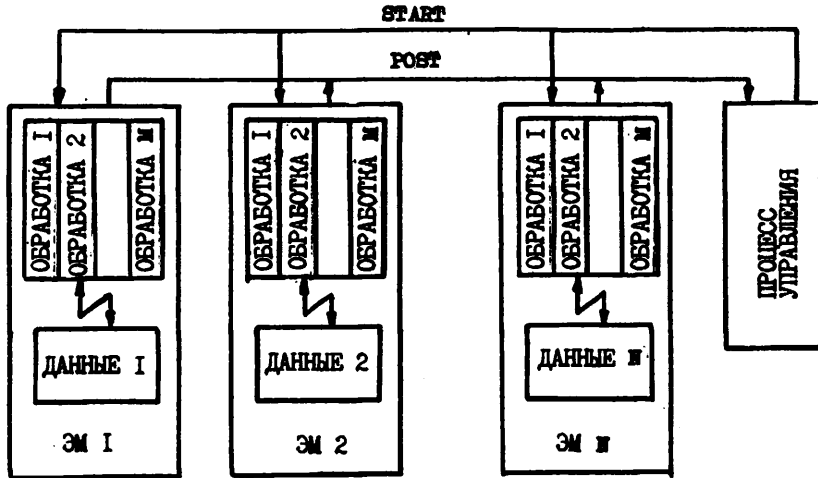


Рис. I

```
STOP ALL
READ DATE(I)
OUTPUT (DATE)
```

Процесс 1-й ЭМ:

```
PROCESS (I): PROCESSING DATE (I)
    POST S,I
    STOP
```

В управляющем процессе директивой **WRITE** записываются обрабатываемые данные и программа процесса **PROCESS** в каждую машину. После запуска процессов в ЭМ директивой **START** управляющий процесс, выполняя процедуру **WAIT**, становится в ожидание на семафоре **S**. При получении извещения от какого-нибудь процесса управляющий процесс останавливает процессы в ЭМ директивой **STOP ALL**, читает результат и выводит на внешнее устройство, выполняя процедуры **READ** и **OUTPUT**. Запущенные управляющим процессом управляемые процессы в ЭМ выполняют процедуру обработки **PROCESSING** и по

окончании ее извещают управляющий процесс директивной РОВТ по semaфору S и останавливаются, выполняя процедуру STOP .

Примером матричной обработки информации является информационно-поисковая задача обнаружения заданного числа в массиве чисел, распределенном между машинами. Поиск числа производится в M подмассивах одновременно. Элементарная машина, первая нашедшая искомое число, извещает об этом управляющий процесс. Управляющий процесс останавливает процедурные процессы во всех машинах и печатает на терминале искомое число и его адрес (при задании числа указывается номер терминала, на котором должен быть получен результат).

Как показал проведенный эксперимент, среднее время поиска на системе из двух машин по сравнению с временем поиска на одной машине в два раза меньше. При больших массивах эффект очевиден, так как накладные расходы по организации вычислений, вносимые операционной системой, занимают меньшую долю в общем времени обработки.

Вычислительная система допускает реконфигурацию при выходе из строя любой ЭМ, а также при подключении новой. Реконфигурация достигается за счет информационной избыточности (дублирования массивов). При любом изменении конфигурации информация об этом изменении выдается оператору на терминал.

2. При конвейерном (магистральном) режиме обработки одиночный поток операндов подвергается обработке на конвейере, элементы которого реализуют в общем случае различные процедуры*).

Задачи, решаемые конвейерным способом, разбиваются на несколько взаимодействующих процессов. Каждый процесс реализуется отдельной ЭМ системы, после чего обработанные данные передаются следующему процессу. Очевидно, что при конвейерном способе скорость прохождения данных через систему равна скорости обработки самого медленного процесса.

Схема реализации режима конвейерной обработки информации показана на рис.2. Структура процессов представлена ниже.

Процесс 1-й машины:

```
PROCESS (I): WAIT S1(I)
              PROCESSING DATE(I)
              WAIT S2(I)
```

*) В частности, в режиме конвейерной обработки функционирует система СУММА при автоматизации научного эксперимента [3].

КОНВЕЙЕРНЫЙ СПОСОБ ОБРАБОТКИ ИНФОРМАЦИИ

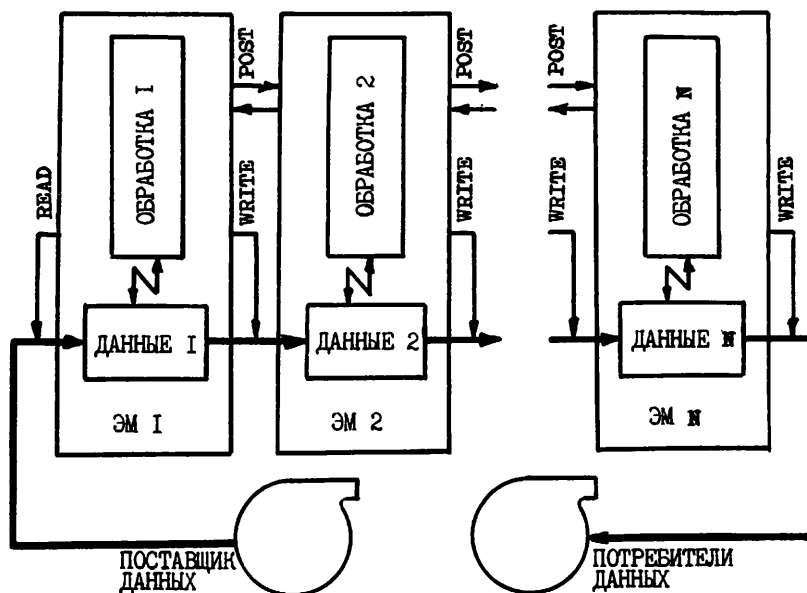


Рис. 2

```

WRITE DATE(I)
POST S2(I-1)
POST S1(I+1)
GO TO PROCESS(I)
    
```

Процессы отличаются только процедурой обработки данных `PROCESSING DATE(I)`. Процесс в i -й машине по директиве `WAIT S1(I)` ожидает на семафоре $S1$ извещение от $(i-1)$ -й машины о начале работы. Получив извещение, процесс обрабатывает данные процедурой `PROCESSING DATE(I)` и процесс становится в ожидание по директиве `WAIT S2(I)` извещения по семафору $S2$ от $(i+1)$ -й машины о том, что данные переданы $(i+2)$ -й машине и $(i+1)$ -я ЭМ готова к приему новых данных. После извещения данные записываются в $(i+1)$ -ю ЭМ по директиве `WRITE DATE(I)`. После передачи данных i -й процесс извещает $(i-1)$ -й процесс по семафору $S2$ директивой `POST S2(I-1)`.

о готовности i -го процесса к приему новых данных. Далее извещает $(i+1)$ -й процесс о возможности начать работу директивой `POST S1(I+1)` по семафору `S1`, после чего снова становится в ожидание, выполняя `WAIT S1(I)`.

Примером конвейерной обработки информации является исключение из текстовой информации нелегальных символов. При этом поставщиком данных служит оператор (может быть - магнитная лента и т.п.), производящий ввод строки текста с терминала. Каждая машина конвейера проверяет текстовую информацию на легальность определенного для данной машины символа. Обработанный текст распечатывается на терминале, т.е. потребителем данных также является оператор.

Рассмотренные в статье режимы работ ОВС СУММА ОКМД (одиночный поток команд-множественный поток данных) и МКОД (множественный поток команд-одиночный поток данных) не исчерпывают всех возможных вариантов параллельной обработки данных. Программно-аппаратные средства системы допускают также эффективную реализацию режима МКМД (множественный поток команд-множественный поток данных).

Л и т е р а т у р а

1. СЕДУХИН С.Г., ЖЕЛТОВ М.П., КАШУН И.Н. Ядро операционной системы СУММА. -В кн.: Вопросы теории и построения вычислительных систем. (Вычислительные системы, вып. 70.) Новосибирск, 1977, с.113-129.

2. ЕВРЕЙНОВ Э.В., ХОРОШЕВСКИЙ В.Г. Однородные вычислительные системы. -Новосибирск: Наука, 1978.- 316 с.

3. ЗАНИН В.И., ЛУКИН А.Ф., ЛУКИЧ В.Л., СПЕКТОР А.Е., ХОДОР - КОВСКИЙ В.А. Система автоматизированного эксперимента для исследования поведения биологических объектов в реальном масштабе времени. -УСМ, 1978, №6, с.115-118.

Поступила в ред.-изд.отд.
24 сентября 1979 года