РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ИНФОРМАЦИИ
(Вычислительные системы)
1984 год Выпуск 105

УДК 681.324

ОРГАНИЗАЦИЯ МЕЖМАШИННЫХ ВЗАИМОДЕЙСТВИЙ В ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ МИКРОС

О.Г.Монахов, Э.А.Монахова

I. Введение

Вычислительная система (ВС) с программируемой структурой представляет собой совокупность элементарных машин (ЭМ), объединенных программно-управляемой сетью связи [I]. ВС с программируемой структурой МИКРОС формируется из ряда мини- и микромашин, программно совместимых с СМ-3 и СМ-4 и объединяемых друг с другом посредством системных устройств и линий связи [2,3]. Граф межмашинных связей ВС МИКРОС выбирается из числа связных графов, степень вершин которых не превосходит заданного значения $v \leq 6$. Внешние устройства ВС МИКРОС распределены между машинами системы, и каждое устройство может быть подключено к нескольким машинам.

Поступление заданий в ВС возможно как с любого внешнего устройства, так и из самих ЭМ. Каждое задание представляется параллельной программой и может быть реализовано на любом связном подмножестве ЭМ (виртуальной подсистеме), содержащем необходимое количество информационно-вычислительных ресурсов. В каждой виртуальной подсистеме реализуются определенные межмашинные взаимодействия, предписанные выполняемым на данной подсистеме заданием.

В работе рассматриваются реализованные в распределенной операционной системе ВС МИКРОС способ адресации ЭМ виртуальных подсистем, алгоритмы маршрутизации пакетов (алгоритмы определения трасс передачи данных между ЭМ подсистемы), способы организации и протоколы межмашинных обменов, способы управления коммуникационными ресурсами, что в совокупности составляет средства управления сетью связи ВС и соответствует сетевому уровню в модели архитектуры открытых систем [4].

2. Задание виртуальных подсистем

Виртуальной подсистемой называется абстрактный распределенный объект [5], который объединяет в функциональное целое физические и логические ресурсы, выделенные для исполнения одного задания и расположенные, возможно, в разных машинах системы. С точки зрения пользователя виртуальная подсистема представляет собой единый ресурс и является функциональным эквивалентом вычислительной системы с тем же числом машин и графом межмашинных связей, что и у подсистемы, и эмулируется средствами операционной системы на реальной ВС.

Виртуальная подсистема Q_s, выделенная для исполнения задания s, определяется как четверка $Q_s = \langle V^s, L^s, I^s, C^s \rangle$, где V^s – множество виртуальных ЭМ подсистемы Q_s: $L^s \subseteq V^s \times V^s$ – множество виртуальных линий связи между ЭМ подсистемы; I^s – множество функций идентификации ЭМ подсистемы, задающих, в частности, отображение $I_A: V^s \to A^s$, A^s – пространство адресов ЭМ подсистемы; C^s – множество коммутационных функций, задающих отображение пространства адресов ЭМ подсистемы в множество выходных полюсов ЭМ. Обозначим $X_i^+ = \{0,1,2,\ldots,v\}$ и $X_i^- = \{0,1,2,\ldots,v\}$ множества входных и выходных полюсов $ЭМ_i$.

Программирование структуры ВС осуществляется путем порождения на реальных ресурсах системы виртуальных ЭМ и настройки между ними виртуальных линий связи, образующих подсистему требуемой конфигурации. Каждая реальная ЭМ может входить в несколько виртуальных подсистем и исполнять одновременно в мультипрограммном режиме функции нескольких виртуальных ЭМ. Общее число как виртуальных подсистем, так и виртуальных ЭМ, порождаемых в системе, ограничено только физическими ресурсами системы. Это делает скрытым от пользователя реальную макроструктуру ВС и позволяет независимо от нее создавать виртуальные подсистемы, структуры которых наиболее соответствуют структурам решаемых задач.

Виртуальная линия связи $(i,j) \in L^s$ между двумя соседними виртуальными машинами $ЭМ_i^s$ и $ЭМ_j^s$ задается в каждой ЭМ входной и выходной очередями (структурированными семафорами [6]): $ВХ_p^i$, $ВЫХ_p^i$ в $ЭМ_i^s$ и $ВХ_q^j$, $ВЫХ_q^j$ в $ЭМ_j^s$, при условии, если линия (i, j) соединяет полюс p $ЭМ_i$ с полюсом q $ЭМ_j$. Будем обозначать эту линию также (i_p, j_q). Виртуальная линия (i_p, j_q) обеспечивает передачу данных из выходной очереди $ЭМ_i^s$ во входную очередь соседней $ЭМ_j^s$, если на пе-

редающем конце в очереди $ВЫХ_p^i$ появились данные, а на принимающем конце в очереди $ВХ_q^j$ есть место в памяти для их приема. Выделяемая для обмена сообщениями память виртуальной $ЭМ_i^s$ образует буферную память или буферный пул $ЭМ_i^s$. Буферные пулы различных виртуальных $ЭМ_i^s$, разделяющих данную физическую ЭМ, не пересекаются. Подробнее способ разделения одной физической линии связи несколькими виртуальными и алгоритм функционирования виртуальной линии связи рассмотрен в [6].

Для организации межмашинных обменов внутри виртуальной подсистемы при ее порождении задаются путевые процедуры (алгоритмы маршрутизации), определяющие трассы передачи данных между виртуальными ЭМ на основе коммутационных функций, которые вычисляются динамически при формировании виртуальных подсистем.

Над виртуальными подсистемами, как над абстрактными объектами, определены следующие операции: I) порождение подсистемы (по заданному количеству ресурсов и типу графа межмашинных связей: дерево, кольцо, линейка, решетка, произвольный связный подграф структуры системы); 2) инициация работы подсистемы; 3) уничтожение подсистемы; 4) реконфигурация подсистемы при отказах (динамическое изменение структуры подсистемы). Множество виртуальных подсистем с определенным над ним множеством операций образует абстрактный тип данных.

Виртуальные подсистемы позволяют: I) обеспечить функциональную целостность ресурсов системы, выделенных для исполнения задания (как пользователя, так и операционной системы); 2) организовать разделение ресурсов системы между заданиями в режиме мультипрограммирования; 3) скрыть от пользователя реальную структуру ВС и автоматически настраивать структуру виртуальных подсистем, адекватную решаемым задачам и мультипрограммной ситуации в системе.

Средства организации виртуальных подсистем представляют собой децентрализованные алгоритмы [7] распределенной операционной системы ВС МИКРОС и реализованы на языке ПАСКАЛЬ-С (язык ПАСКАЛЬ штатной операционной системы РАФОС, расширенный синхронизирующими примитивами ядра ОС МИКРОС, обеспечивающими порождение, уничтожение, взаимодействие (синхронизацию и коммуникацию) асинхронных вычислительных процессов).

Каждая виртуальная $ЭМ_i^s$, принадлежащая виртуальной подсистеме Q_s, в данной физической $ЭМ_i$ задается с помощью таблицы параметров, называемой окружением. Структура данных типа "ОКРУЖЕНИЕ"

(ENVIRE) определяется следующим образом:

```
ENVIRE = RECORD
           ADR:^WAY;
           E,L,G,F,W,G1,G2,
           FREBUF, MAXBUF, CTOK,
           G2COUNT,MEM,COL,
           PACLEN:INTEGER;
           LINK:ARRAY[VAL] OF CONNECT
           END;
```

где VAL=0..MAXDEG;

```
CONNECT=RECORD
          BX,BHX:INTEGER
        END;
```

ADR - указатель на адрес $ЭМ_i^s$ в подсистеме Q_s; E - номер машины в подсистеме; L - расстояние $ЭМ_i^s$ от корневой ЭМ подсистемы, G,F,W - указатели на путевую, трансляционную и весовую функции [8]; G1,G2 - число пакетов класса I и 2 [9]; FREBUF, MAXBUF - текущее и максимально возможное число свободных буферов в буферном пуле $ЭМ_i^s$; MEM - указатель на семафор $MEMORY_i^s$ [6], являющийся списком участков памяти, выделенной $ЭМ_i^s$; COL - имя виртуальной подсистемы, которой принадлежит $ЭМ_i^s$; PACLEN - длина пакета при передаче данных в подсистеме Q_s; BX[p] , BHX[p] - адреса входной и выходной очередей для p-го полюса; MAXDEG - максимальная степень графа межмашинных связей ВС.

При порождении виртуальной ЭМ формируется структура данных типа "ОКРУЖЕНИЕ", куда заносятся параметры порождаемой ЭМ, и структура типа:

```
AFBUF=RECORD
COL:INTEGER;
ADR:^WAY;
BUF:^ENVIRE;
LEN:INTEGER
END;
```

где ADR и BUF - указатели на адрес и элемент ENVIRE порождаемой ЭМ. Структура типа AFBUF заносится в структурированный семафор ADFULL , хранящий данные о всех виртуальных ЭМ, порожденных в данной физической ЭМ.

3. Адресация машин подсистемы

Адресное пространство виртуальной подсистемы Q_s определяется парой (A^s, M), где A^s – пространство адресов виртуальных ЭМ подсистемы, M – адресное пространство памяти виртуальной ЭМ. Адресация ЭМ подсистемы Q_s задает однозначное соответствие между элементарной машиной*) $ЭМ_i \in Q_s$, $1 \le i \le n$, $n=|V^s|$, и ее адресом A_i, который служит для определения трасс передачи данных между машинами подсистемы. Обозначим $A^{*m} = \{a \in A^* \mid |a| \le m\}$ множество слов над алфавитом $A=\{0,1,\ldots,v\}$, длина которых не более m, где v – степень графа межмашинных связей подсистемы. Множество A^{*m} представляет собой множество адресов ЭМ при MD(z,m) – адресации [7]. В этом случае адрес $ЭМ_i \in Q_s$ представлен упорядоченной последовательностью (словом) ярусных номеров (символов из алфавита A) длины $r \le m$: $A_i = A_i^1 A_i^2 \ldots A_i^r$, где $A_i^j \in A$, $m \cdot z = \lceil \log_2 |Q_s| \rceil$. При назначении адресов ЭМ соблюдается следующее правило: адрес каждой $ЭМ_i \in Q_s$ должен представлять собой последовательность номеров выходных полюсов машин, лежащих на кратчайшем пути от корневой $ЭМ_1$ до $ЭМ_i$. При этом в качестве корневой машины $ЭМ_1$ в подсистеме Q_s может быть выбрана любая ЭМ, адрес которой принимается равным нулю: $A_1 = 0$. Алфавит A в этом случае совпадает с множеством выходных полюсов ЭМ системы: $\{0,1,\ldots,v\}$. Машины, адреса которых имеют одинаковую часть (префикс) $a = A_i^1 A_i^2 \ldots A_i^{r-1}$ длины $r-1$, $2 \le r \le m+1$, должны образовывать связное подмножество ЭМ подсистемы, которое назовем адресной подсистемой яруса r и обозначим: $R_r(a)$.

В ВС МИКРОС адрес ЭМ представлен структурой данных следующего типа:

```
WAY=RECORD
    LEN:INTEGER;
    TAD:ARRAY[1..ADRLEN] OF CHAR
    END;
```

где LEN– длина адреса (число символов из алфавита A), ADRLEN – максимальная длина адреса в подсистеме, TAD – последовательность ярусных номеров.

---

*) Здесь и далее под элементарной машиной (ЭМ) понимается виртуальная машина, случаи употребления физических ЭМ оговариваются особо.

4. Децентрализованный динамический алгоритм назначения адресов

Алгоритм назначения адресов при MD(z,m) -адресации в вир - туальной подсистеме Q_s представляет собой децентрализованный алгоритм, реализуемый параллельными, асинхронными процессами,исполняемыми в каждой ЭМ подсистемы Q_s и взаимодействующими между собой посредством обмена сообщениями. Алгоритм назначения адресов исполняется при условии, что выделено множество реальных машин. входящих в подсистему Q_s, и целью данного алгоритма является задание адресов виртуальным машинам, входящим в эту подсистему.

Пусть состояние каждой $ЭМ_i$, $1 \le i \le n$, характеризуют две переменные: l_i - расстояние $ЭМ_i$ от корневой ЭМ подсистемы Q_s и A_i -адрес $ЭМ_i$ в этой подсистеме. Начальные значения переменной A_i следующие: $A_i = \emptyset$ при $2 \le i \le n$ и $A_1=0$ - у корневой ЭМ.Значения переменных l_i определены алгоритмом формирования подсистемы [7].

Алгоритм назначения адресов начинается с того, что корневая $ЭМ_1$ формирует сообщения $H_3(A, c)$, где $A=A_1 * p$, p - номер выходного полюса, * - знак операции конкатенации и c=1 - счетчик. Такие сообщения корневая ЭМ рассылает по всем выходным направлениям $p \in X^-$

По получении сообщения $H_3(A, c)$ с входного полюса $q \in X^+$, машина $ЭМ_i \in Q_s$, $2 \le i \le n$, выполняет следующие действия: если $l_i = c$ и $A_i = \emptyset$, то $A_i := A$, $c := c+1$, для каждого $p \in X^- \setminus \{q\}$ послать сообщение $H_3(A * p, c)$ по выходному полюсу p; иначе сообщение игнорируется.

Таким образом, в результате работы данного алгоритма сообщения H_3 распространяются из корневой ЭМ по подсистеме. При этом каждой элементарной машине подсистемы присваивается адрес, представляющий собой последовательность номеров выходных направлений машин, принадлежащих кратчайшему пути (длины l), пройденному сообщением H_3 из корневой ЭМ до данной машины.

Приведем некоторые оценки данного алгоритма. Максимальное количество битов, которое должно содержать сообщение H_3, определяется максимальной длиной адреса в подсистеме при MD(z,m)-адресации и диаметром подсистемы D и равно $D \cdot \lceil \log_2 v \rceil + \log_2 D$.

Минимальное количество сообщений H_3, необходимое для назначения адресов, определяется числом машин в подсистеме Q_s и равно n-1. Общее число сообщений, генерируемых данным алгоритмом, равно сумме минимально необходимого числа сообщений и удвоенного числа

хорд остова графа межмашинных связей подсистемы: $2|L| + 1-n$, где L – множество ребер графа межмашинных связей подсистемы Q_s.

Обозначим через τ_3 верхнюю оценку времени, необходимого для обработки сообщения H_3 в $ЭМ_i$, $1 \le i \le n$, и рассылки данного сообщения соседним ЭМ. Тогда T_3 – время выполнения алгоритма назначения адресов в подсистеме Q_s – пропорционально диаметру графа межмашинных связей данной подсистемы и не превосходит следующей величины: $T_3 \le (D+1)\tau_3$.

5. Коммутационные функции

Множество коммутационных функций включает путевые и трансляционные функции, которые задают в каждой ЭМ подсистемы Q_s отображение вида: $c: A^{*m} \to \{0,1,\dots,v\}$, т.е. сопоставляют каждому адресу из пространства адресов подсистемы Q_s некоторое множество выходных полюсов данной ЭМ.

Путевая функция $G_j(A_i)$ определена в каждой машине $ЭМ_j$, $1 \le j \le n$, подсистемы Q_s и ставит в однозначное соответствие каждому адресу $A_i \in A^{*m}$ номер выходного полюса $p \in \{0,1,\dots,v\}$ из множества выходных полюсов $ЭМ_j$. При этом значение $G_j(A_i)$, $i \ne j$, $1 \le i, j \le n$, равно номеру выходного полюса p, инцидентного ребру, принадлежащему кратчайшему пути от $ЭМ_j$ до адресной подсистемы $R_{r+1}(A_j^1 \dots A_j^{r-1} A_i^r)$, где $r = \min\{k | A_i^k \ne A_j^k, 1 \le k \le m\}$, к которой принадлежит $ЭМ_i$. При использовании табличного способа задания путевых функций при MD(z,m)-адресации необходимый объем памяти для их хранения в $ЭМ_i$ определяется следующим образом: $l.v.\lceil \log_2 v \rceil$, где $1 \le l \le m$ – расстояние $ЭМ_i$ от корневой ЭМ, а длина адреса $ЭМ_i$ в этом случае составит $l\lceil \log_2 v \rceil$ битов. Отметим, что в этом случае значение путевой функции $G_i(A_k^r)$ всегда определено при $r>1$ и равно A_k^r. Множество кратчайших путей, ведущих из каждой ЭМ подсистемы Q_s в одну выделенную машину $ЭМ_k \in Q_s$, и заданное множеством путевых функций $G_i(A_k)$, $1 \le i \le n$, образует покрывающее дерево D_k графа межмашинных связей подсистемы Q_s с корнем в $ЭМ_k$.

Трансляционная функция $F_i(A_k)$ определена в каждой $ЭМ_i$, $1 \le i \le n$, подсистемы Q_s и ставит в однозначное соответствие каждому адресу A_k некоторое подмножество из множества выходных полюсов $ЭМ_i$. Трансляционная функция определяется следующим образом: $F_i(A_k) = \{t | p = G_h(A_k) \ \&\ (i_t, h_p) \in L_s\}$. При этом функция $F_i(A_k)$ зада-

ет множество номеров выходных направлений машины $ЭМ_i$, ведущих только в те соседние с ней машины, кратчайший путь от которых в машину $ЭМ_k$ проходит через $ЭМ_i$. Объем памяти, необходимый для задания трансляционной функции в одной ЭМ, равен $m \cdot v^2$ битов.

Множество трансляционных функций $F_i(A_k)$, $1 \le i \le n$, $k \in \overline{1,n}$, определенных в каждой $ЭМ_i$ подсистемы Q_s, образует тоже самое покрывающее дерево D_k с корнем в $ЭМ_k$, что и множество путевых функций $G_i(A_k)$, $1 \le i \le n$, $k \in \{1,2,\ldots,n\}$. Но если путевые функции задают направление, ведущее к корню дерева D_k, то трансляционные функции - от корня дерева D_k.

Таким образом, путевые функции $G_i(A_k)$, $1 \le i \le n$, $k \in \{1,2,\ldots,n\}$, задают кратчайший (или близкий к нему) путь от любой машины $ЭМ_i$ в машину $ЭМ_k$, а трансляционные функции $F_i(A_k)$ - множество путей, принадлежащих покрывающему дереву D_k, из $ЭМ_k$ во все машины подсистемы. Это свойство данных функций используется для организации путевых процедур.

Путевая и трансляционная функции $G_i(A_k)$ и $F_i(A_k)$ определяются в $ЭМ_i$ в зависимости от номера разряда рассогласования $r = \min\{j / A_i^j \neq A_k^j,\ j = \overline{1,m}\}$ и значения разряда рассогласования $A_k^r \in A$. Поэтому таблица коммутационных функций задается в виде структурированного семафора SGF, содержащего m элементов типа GFBUF:

```
GFBUF=RECORD
        COL:INTEGER;
        ADR:^WAY;
        BUF:^ARGF;
        LEN:INTEGER
      END;
```

где

```
GF=RECORD
   SF,LEN:INTEGER;
   G:VAL;
   F:ARRAY [VAL] OF BOOLEAN
  END;
ARGF=ARRAY[VAL] OF GF;
```

G - значение путевой функции, F - значение трансляционной. Определение значения коммутационных функций в зависимости от разряда рассогласования r и его значения A_k^r происходит следующим образом:
I) выбирается элемент типа GFBUF с номером r из семафора SGF ;

2) из массива ARGF выбирается элемент с номером A_k^r типа GF;
3) значения полей G и F выбранного элемента типа GF определяют искомые значения коммутационных функций.

6. Децентрализованный алгоритм вычисления коммутационных функций

Алгоритм вычисления коммутационных функций в виртуальной подсистеме Q_s представляет собой параллельный алгоритм, реализуемый посредством параллельных, асинхронных процессов, исполняемых в каждой ЭМ подсистемы Q_s. Целью этого алгоритма является формирование таблиц путевой и трансляционной функций при отсутствии априорных данных о графе межмашинных связей подсистемы. Алгоритм вычисления функций исполняется при условии, что выделены реальные машины, входящие в подсистему Q_s, и заданы их адреса в системе MD(z,m)-адресации. Таким образом, алгоритмы выделения элементарных машин подсистемы Q_s и алгоритмы задания MD(z,m)-адресации представляют собой первый этап в формировании подсистем, а алгоритм вычисления функций G,F - второй этап.

Введем функцию расстояний $J_i(A_k)$, которая определена в каждой $ЭМ_i$, $1 \le i \le n$, подсистемы Q_s и значение которой равно длине кратчайшего пути (расстоянию) от $ЭМ_i$ до адресной подсистемы $R_{r+1}(A_i^1 \ldots A_i^{r-1} A_k^r)$, к которой принадлежит $ЭМ_k$, где $r=\min\{j/A_i^j \neq A_k^j, 1 \le j \le m\}$. Аналогично путевой функции, функция расстояния $J_i(A_k)$ определяется в зависимости от двух параметров: номера разряда рассогласования r и значения разряда рассогласования $A_k^r \in A$. Объем памяти, необходимый для функции расстояния в одной ЭМ при табличном способе задания в случае MD(z,m)-адресации можно определить по формуле: $l \cdot v \cdot \lceil \log_2 D \rceil$, где $1 \le l \le m$ - расстояние данной ЭМ от корневой ЭМ подсистемы Q_s, D - диаметр графа межмашинных связей подсистемы Q_s.

Идея алгоритма вычисления коммутационных функций сводится к следующему. Каждая $ЭМ_i$ подсистемы Q_s посылает по всей подсистеме сообщение $H_s(A_i,c,u)$, где A_i - адрес $ЭМ_i$, c - счетчик транзитных ЭМ, пройденных сообщением, u = 1, если i = 1, и u = 0 при $i \neq 1$. Получив такое сообщение по линии связи (j_q,k_p) $ЭМ_k \in V^s$ подсистемы Q_s определяет по счетчику c величину расстояния до адресной подсистемы, к которой принадлежит $ЭМ_i$, и если оно меньше, чем имеющееся у нее в таблице функции расстояний $J_k(A_i)$, то $ЭМ_k$ заносит в таблицу путевой функции $G_k(A_i)$ номер полюса p, по которому при-

шло данное сообщение, а в таблицу функции расстояний $J_k(A_i)$ – значение счетчика c, т.е. величину расстояния до адресной подсистемы $ЭМ_i$. Далее, $ЭМ_k$ увеличивает значение счетчика c на единицу c:=c+1 и рассылает полученное сообщение по всем выходным полюсам, кроме полюса p. После этого $ЭМ_k$ формирует сообщение $H_9(A_i,y)$, если произошла коррекция таблиц функции расстояний и путевой функции в $ЭМ_k$, и рассылает это сообщение по всем выходным направлениям, кроме p ,с y= 0, а по направлению p – с y = 1.

Получив сообщение $H_9(A_i,y)$ по линии связи (j_q,k_p), $ЭМ_j \in V^s$ производит коррекцию трансляционной функции, а именно при y = 1 номер выходного направления q добавляется к множеству значений функции: $F_j(A_i):=F_j(A_i)\cup\{q\}$, так как кратчайший путь от адресной подсистемы $ЭМ_i$ до $ЭМ_k$ проходит через $ЭМ_j$. При y=0 номер полюса q удаляется из множества значений трансляционной функции: $F_j(A_i):=F_j(A_i)\setminus\{q\}$.

В результате работы данного алгоритма во всех ЭМ подсистемы формируются таблицы путевых и трансляционных функций.

Перед началом работы алгоритма вычисления коммутационных функций считаем, что адреса всех ЭМ подсистемы заданы, начальные значения функции расстояний во всех ЭМ принимаются равными ∞ (максимальному целому числу), начальные значения путевой функции $G_i(A_j)=0$, $1 \le i, j \le n$, трансляционной функции $F_i(A_i)=\overline{X}$, $F_i(A_j)=\emptyset$, при $i \ne j$, $1 \le i, j \le n$.

Действия по изменению значений функций G и J в ходе работы данного алгоритма выполняются с помощью процедуры КОРРЕКЦИЯ, входными параметрами которой являются: A_k – адрес $ЭМ_k$, выполняющей данную процедуру, G_k – таблица путевой функции $ЭМ_k$, J_k – таблица функции расстояний $ЭМ_k$, сообщение $H_8(A,c,u)$ и q – номер полюса, по которому оно получено; выходной параметр h = I, если произошла коррекция таблиц G_k и J_k, и h=0 – в противном случае. Процедура КОРРЕКЦИЯ $(A_k,G_k,J_k,H_8(A,c,u),q,h)$ состоит из следующих действий.

I. Присвоить h:=0, определить $r=\min\{t/A_k^t \ne A^t,\ t=\overline{1,m}\}$;

2. Если $J_k(A^r) > c$, то $J_k(A^r):=c$, $G_k(A^r):=q$, h:=1 .

Выполнение алгоритма вычисления коммутационных функций подсистемы Q_s начинается с того, что корневая ЭМ подсистемы формирует сообщение $H_8(A,c,u)$, где $A=A_1$, c=1, u=1, и рассылает его по всем своим выходным полюсам.

Элементарная машина $ЭМ_k$ подсистемы Q_s, получив сообщение $H_s(A,c,u)$ с входного полюса p, выполняет следующие действия:

1. КОРРЕКЦИЯ $(A_k, G_k, j_k, H_s(A,c,u), q, h)$;

2. Если h=1, то y:=0, c:=c+1, разослать по всем выходным полюсам $q \in X_k^- \setminus \{p\}$ сообщение $H_s(A,c,u)$ и сообщение $H_9(A,y)$, y:=1, послать по выходному полюсу p сообщение $H_9(A,y)$;

3. Если u=1 и $J_k(A) = \infty$, то $A:=A_k$, c:=1, u:=0 и сообщение $H_s(A,c,u)$ разослать по всем выходным полюсам $ЭМ_k$.

Элементарная машина $ЭМ_k$ подсистемы Q_s, получив сообщение $H_9(A,y)$ с выходного полюса p, выполняет следующие действия:

1. Определить $r = \min\{t / A_k^t \neq A^t,\ t = \overline{1,m}\}$;

2. Если y=1, то $F_k(A^r) := F_k(A^r) \cup \{p\}$, иначе $F_k(A^r) := F_k(A^r) \setminus \{p\}$.

Алгоритм вычисления коммутационных функций заканчивает свою работу после того, как таблицы путевой и трансляционной функций в каждой ЭМ заполнены. Рассмотрим необходимое условие окончания работы данного алгоритма.

Обозначим через $p_r(A_i) = p_r(A_i^1 \ldots A_i^{r-1})$ число адресных подсистем уровня r, имеющих префикс $a = A_i^1 \ldots A_i^{r-1}$, $2 \le r \le m+1$, и множество значений, которые принимает r-й разряд префикса в этих подсистемах, обозначим через $B_i^r \subseteq \{0,1,\ldots,v\}$, $p_r(A_i) = |B_i^r|$. При MD(z,m)-адресации путевая функция $G_i(A_k)$ для каждого разряда рассогласования $r = \min\{j / A_i^j \neq A_k^j,\ 1 \le j \le m\}$ может принимать $p_r(A_i)$ значений. Следовательно, для того чтобы задать путевую функцию при данном разряде рассогласования r, необходимо определить $p_r(A_i)$ ее значений для каждого $b \in B^r$. Введем в $ЭМ_i$ переменную g(r,b), где $b \in B_i^r$, g(r,b) = 0, если значение $G_i(A_i^1 \ldots A_i^{r-1}b)$ – не определено, и g(r,b) = 1 – в противном случае.

Для определения условия заполнения таблицы путевой функции в $ЭМ_i$ введем следующее дополнение в алгоритм вычисления коммутационных функций. При получении в $ЭМ_i$ сообщения $H_s(A_k,c,u)$ в первый раз, т.е. если $g(r,A_k^r) = 0$, определяется разряд рассогласования r, выполняется присвоение $g(r,A_k^r) := 1$ и значение путевой функции $G_i(A_k^r)$ считается определенным при данном разряде рассогласования r и его значении $b = A_k^r$.

Аналогично для случая трансляционной функции введем в $ЭМ_i$ переменную f(r,b,p), где $p \in X_i^- \subseteq \{0,1,\ldots,v_i\}$, v_i – число линий связи $ЭМ_i$ с другими ЭМ данной подсистемы, f(r,b,p) = 1, если не

определено, принадлежит или нет выходной полюс p множеству $F_i(A_i^1 \ldots A_i^{r-1}b)$, и $f(r,b,p) = 0$ – в противном случае.

Для определения условия заполнения таблицы трансляционной функции в $ЭМ_i$ введем следующее дополнение в алгоритм вычисления коммутационных функций. При получении в $ЭМ_i$ сообщения $H_9(A_k,y)$ с полюса p определяется разряд рассогласования r и его значение $b = A_k^r$, и если $f(r,A_k^r,p) = 1$, то $f(r,A_k^r,p) := 0$. Значение трансляционной функции $F_i(A_k^r)$ считается определенным при данных r и $b=A_k^r$, если для каждого выходного полюса p, за исключением полюса $q = G_i(A_k^r)$, определено его отношение к множеству $F_i(A_k^r)$, т.е. если выполняется условие:

$$\bigvee_{p \in X_i \setminus \{q\}} f(r,b,p) = 0,$$

где $b \in B_i^r$, $1 \le r \le l_i$, $q = G_i(A_k^r)$, l_i – расстояние корневой ЭМ от $ЭМ_i$.

Таким образом, можно сформулировать

УТВЕРЖДЕНИЕ I. Необходимые условия заполнения таблиц коммутационных функций в $ЭМ_i$ для заданных разряда рассогласования $1 \le r \le l_i$ и его значения $b = B_i^r$ состоят в том, что:

а) $g(r,b) = 1$,

б) $\bigvee_{p \in X_i \setminus \{q\}} f(r,b,p) = 0$,

где $q = G_i(A_i^1 \ldots A_i^{r-1}b)$.

Отсюда необходимые условия окончания работы алгоритма вычисления коммутационных функций формулируются следующим образом.

УТВЕРЖДЕНИЕ 2. Необходимые условия окончания алгоритма вычисления коммутационных функций в $ЭМ_i$ состоят в том, что для каждого разряда рассогласования r, $1 \le r \le l_i$:

а) $\&_{b \in B_i} g(r,b) = 1$, и

б) $\bigvee_{p \in X_i \setminus \{q\}} f(r,b,p) = 0$ для $\forall b \in B_i$,

где $q = G_i(A_i^1 \ldots A_i^{r-1}b)$.

Приведем некоторые оценки данного алгоритма. Максимальное количество битов, которое должно содержать сообщение H_8, определяется максимальной длиной адреса в подсистеме и диаметром подсистемы и равно $D\lceil\log_2 v\rceil+\log_2 D+1$, а для сообщения H_9 эта величина равна $D\lceil\log_2 v\rceil+1$.

Число сообщений H_8, генерируемых данным алгоритмом при вычислении путевых функций $G_i(A_k)$, $1 \le i \le n$, для каждого $k \in \overline{1,n}$, может быть оценено, как и для алгоритма назначения адресов, величиной, равной $2|L|-n+1$, а общее число сообщений H_8 для всех $1 \le k \le n$ соответственно величиной $n(2|L|-n+1)$, где L – множество ребер графа межмашинных связей подсистемы. Максимальное число сообщений H_9, при условии, что каждое сообщение H_8, вызывает коррекцию таблиц G и J, равно следующей величине: $v \cdot n \cdot (2|L|-n+1)$.

Обозначим через τ_8 верхнюю оценку времени, необходимого для обработки сообщения H_8 в $ЭМ_i$, $1 \le i \le n$, и рассылки его соседним ЭМ, а через τ_9 – верхнюю оценку времени, необходимого для пересылки и обработки сообщения H_9. Время выполнения алгоритма вычисления коммутационных функций T_8 можно приближенно оценить как сумму трех величин: T_1 – время распространения сообщения H_8 по всей подсистеме из корневой ЭМ (эта величина пропорциональна диаметру подсистемы); T_2 – время распространения сообщения H_8 по всей подсистеме из ЭМ, наиболее удаленной от корневой ЭМ (эта величина также пропорциональна диаметру подсистемы); T_3 – время, необходимое для пересылки и обработки сообщения H_9 (оценка этой величины – τ_9). Следовательно, верхнюю оценку времени выполнения данного алгоритма можно оценить по формуле: $T_8 = T_1 + T_2 + T_3 \le (D+1)\tau_8 + (D+1)\tau_8 + \tau_9 = 2\tau_8(D+1) + \tau_9$.

В вычислительной системе МИКРОС реализованы децентрализованные динамические алгоритмы формирования подсистем, задания в них MD(z,m)-адресации и формирования таблиц коммутационных функций.

7. Алгоритмы маршрутизации и протоколы межмашинных обменов в подсистеме

Передача информации в ВС МИКРОС происходит методом коммутации пакетов без предварительного установления соединения. Передающая ЭМ разбивает сообщение, помещенное в выходную очередь виртуальной ЭМ, на пакеты (блоки данных) одинаковой длины, заданной для подсистемы, и помещает их в соответствующую выходную очередь физической ЭМ для передачи по физической линии связи в соседнюю

ЭМ. Передаваемый пакет содержит служебную информацию, в которой указаны, в частности, имя (цвет) виртуальной подсистемы, адрес принимающей виртуальной ЭМ, тип обмена (алгоритм маршрутизации). Механизм виртуальной линии связи обеспечивает доставку пакета во входную очередь соседней виртуальной ЭМ, принадлежащей той же подсистеме, что и передающая ЭМ. Постановка пакета во входную очередь инициирует работу менеджера связи виртуальной ЭМ, принявшей пакет. Менеджер связи, реализованный в виде процесса MANCOM, на основе анализа служебной информации пакета (типа обмена), активизирует процесс интерпретации соответствующей путевой процедуры, реализующей заданный тип обмена. Исполняя путевую процедуру, процесс интерпретации определяет номер выходного направления для дальнейшей передачи данного пакета и ставит его в соответствующую выходную очередь. Передача пакетов по сети связи ВС в принимающую ЭМ происходит независимо друг от друга. Сборка сообщений происходит в принимающей ЭМ, где предварительно выделяется необходимая для всего сообщения память (при первом поступлении в ЭМ любого пакета данного сообщения с данным номером передающей ЭМ). Собранное сообщение помещается во входную очередь $ВХ_0$ принимающей виртуальной ЭМ. Более подробно процесс разбиения и сборки сообщений и работа уровня информационного канала изложены в [10].

Алгоритмы маршрутизации (путевые процедуры), реализованные в ВС МИКРОС, являются децентрализованными алгоритмами, основанными на использовании коммутационных функций. Это означает, что для реализации путевой процедуры или протокола обмена в каждую ЭМ виртуальной подсистемы закладывается один и тот же алгоритм.

При реализации дифференцированного обмена ("одна ЭМ – одной ЭМ"), передающая ЭМ формирует сообщение вида: $H\{A_k\}T$, где H = ДР – заголовок, содержащий признак, задающий тип обмена, $\{A_k\}$ – адрес принимающей $ЭМ_k$, T – текст сообщения. Любая $ЭМ_i$ данной подсисте – мы, получив такое сообщение (пакет) с признаком H = ДР (дифференцированный обмен), исполняет процедуру ROUTNG , которая состоит в следующем. Сравнивается содержащийся в пакете адрес A_k прини – мающей ЭМ с адресом A_i. Определяется $r = \min\{j \mid A_i^j \neq A_k^j,\ j = \overline{1,m}\}$, номер разряда рассогласования адресов и величина A_k^r значения раз-ряда рассогласования адресов. Если $A_k^r = p = 0$, то пакет предназна – чен для $ЭМ_i$. Если $A_k^r \neq 0$, то определяется значение путевой функции $p = G_i(A_k)$, задающее номер выходного полюса, по которому необходи-

мо отправить полученный пакет. Процедура ROUTNG по номеру p ставит пакет в выходной семафор $ВЫХ^i_p$, если $p \neq 0$, или во входной семафор $ВХ^i_0$, если $p=0$. Имеются две модификации описанной процедуры: процедура ROUTNG1, не заносящая пакет в выходной семафор, и процедура ROUTNG2, реализующая дифференциальный обмен с предотвращением возникновения тупиковых состояний при передаче пакетов. Указанные процедуры имеют следующее описание на языке ПАСКАЛЬ:

```
PROCEDURE ROUTNG1 (VAR ADR: WAY; VAR
AF:ENVIRE; VARP:VAL);

PROCEDURE ROUTNG2 (VAR AF:ENVIRE; VAR
Q:VAL);
```

где ADR – адрес принимающей ЭМ, AF – "окружение" виртуальной ЭМ, исполняющей путевую процедуру, P – номер выходного полюса, по которому необходимо передать пакет, Q – номер входного полюса, с которого поступил пакет.

При реализации группового обмена ("одна ЭМ – некоторому подмножеству ЭМ подсистемы") передающая $ЭМ_i$ формирует сообщение вида: $Н\{A_{i1},\ldots,A_{it}\}Т$, где заголовок Н=ГР & Л (групповой обмен типа "линейка"), A_{ir} $(1 \leq r \leq t)$ – множество адресов принимающих ЭМ. В системе МИКРОС групповой обмен реализован путевой процедурой типа "линейка" [8] с раздельным поиском путей к каждой из принимающих ЭМ.

Ниже описана процедура LINE, реализующая групповой обмен в $ЭМ_k$ виртуальной подсистемы Q_s. Любая $ЭМ_k$ данной подсистемы, получив пакет с признаком Н = ГР&Л, сравнивает свой адрес с адресами из множества $\{A_{ir}\}$. Если он совпадает с одним из них, то $ЭМ_k$ принимает этот пакет (следует засылка пакета во входной семафор $ВХ_0$ данной ЭМ) и удаляет свой адрес из множества $\{A_{ir}\}$. Если $\{A_{ir}\} \neq \emptyset$ или в случае, если адрес $ЭМ_k$ не совпадает ни с одним из адресов множества $\{A_{ir}\}$, определяется значение путевой функции $p = G_k(A_{i1})$ по первому адресу из списка адресов принимающих ЭМ, и пакет ставится в выходной семафор $ВЫХ^*_p$. Если $\{A_{ir}\} = \emptyset$, т.е. достигнуты все принимающие ЭМ, то групповой обмен считается реализованным.

При реализации трансляционного обмена ("одна ЭМ – всем ЭМ подсистемы"), передающая ЭМ формирует сообщение вида: $Н\{A_i\}Т$, где Н = ТР (трансляционный обмен). Для организации трансляционного обмена использован распределенный реверсивный алгоритм, который реализован в виде процедуры TRANSL и сводится к следующим

действиям. Любая $ЭМ_k$ данной подсистемы, получив сообщение с признаком H = TP, определяет значение трансляционной функции: $\{p\} = F_k(A_i)$, которое задает множество номеров выходных полюсов, по которым необходимо отправить полученное сообщение. В выходные очереди по всем направлениям $p \in F_k(A_i)$ заносятся копии полученного сообщения.

Как следует из свойств адресации и коммутационных функций, приведенные протоколы межмашинных обменов в подсистемах обеспечивают передачу сообщений между ЭМ по кратчайшим путям (или близким к ним) и гарантируют отсутствие лишних копий сообщений.

Приведем описание процедур, реализующих групповой и трансляционный обмены:

```
PROCEDURE LINE (VAR MB:MESBUF; VAR AF:
ENVIRE; VAR P:VAL);
PROCEDURE TRANSL (VAR MB:MESBUF; VAR AF:
ENVIRE; VAR Q:VAL);
```

где AF - "окружение" ЭМ, исполняющей путевую процедуру, P - номер выходного полюса, по которому необходимо передать пакет, Q - номер входного полюса, с которого поступил пакет, MB - поступивший пакет (сообщение).

Пакеты (сообщения) в ВС МИКРОС описываются структурой данных следующего типа:

```
MESBUF=RECORD
      COL:INTEGER;
      ADR:^MESHEAD;
      BUF:^MES; LEN:INTEGER
      END;
MESHEAD=RECORD
      SIGN:INTEGER;
      HEAD:ARRAY[1..3] OF INTEGER;
      LEN,NADR:INTEGER;
      ADRS:WAY
      END;
MES=ARRAY[1..LEN] OF INTEGER;
```

где COL - имя виртуальной подсистемы, MES - информационная часть пакета, LEN - длина информационной части пакета, MESHEAD - заголовок (служебная часть) пакета, ADRS - список адресов принимающих

ЭМ, LEN - общая длина списка адресов (в байтах) +2; NADR - число принимающих ЭМ; HEAD - поле, используемое при сборке сообщений из пакетов; SIGN - поле признаков. При реализации дифференцированного и трансляционного обменов заголовок пакета имеет структуру данных следующего типа:

```
MESHEAD=RECORD
        SIGN:INTEGER;
        HEAD:ARRAY[1..3] OF INTEGER;
        ADR:WAY
        END;
```

где ADR - адрес принимающей ЭМ (в случае дифференцированного об - мена) или передающей ЭМ (в случае трансляционного обмена).

Общий объем блоков распределенной децентрализованной операционной системы ВС МИКРОС, реализующих протоколы межмашинных обменов, составляет 4 К слов. Язык программирования - ПАСКАЛЬ-С.

8. Процедура предотвращения тупиковых состояний при межмашинных обменах

Использование ВС с программируемой структурой является одним из способов повышения эффективности решения задач. Однако эффективность системы в случае возникновения дедлока (тупика) может быть снижена до нуля, если не используются средства предотвращения тупиковых ситуаций [4,9]. При этом под тупиком или дедлоком понимается такое состояние наполнения буферных пулов некоторого множества циклически связанных ЭМ, в результате которого ни один пакет сообщений из этих ЭМ не может быть передан или принят. Таким образом, в состоянии дедлока невозможно дальнейшее функционирование системы (подсистемы).

Проблема организации беступикового режима обмена информацией решена в ВС МИКРОС на основе реализации простого децентрализованного алгоритма предотвращения тупиковых состояний, использующего MD(z,m)-адресацию ЭМ и описанные выше протоколы обмена. Алгоритм основан на предложенной в [9] системе классификации передаваемых пакетов. Каждый пакет, находящийся в буферной памяти $ЭМ_i$ подсистемы и адресованный $ЭМ_j$ подсистемы, $j \neq i$, в зависимости от адресов $ЭМ_i$ и $ЭМ_j$ получает принадлежность к одному из трех возможных классов $P_i = P_i^1 \cup P_i^2 \cup P_i^3$ передаваемых пакетов. Класс $1(P_i^1)$ составляют пакеты, находящиеся в адресной подсистеме своей принимающей ЭМ

и направленные путевой процедурой вниз от корневой ЭМ по дереву путей, порождаемому MD(z,m)-адресацией. Класс $2(P_i^2)$ составляют пакеты, направленные путевой процедурой (или перенаправленные процедурой предотвращения дедлоков) вверх к корневой ЭМ по дереву путей, порождаемому MD(z,m)-адресацией. К классу $3(P_i^3)$ относятся пакеты, не вошедшие в классы I и 2. При обычных условиях (не перегруженной сети связи) маршруты передаваемых пакетов определяются путевой процедурой, а их прием в буферные пулы ЭМ определяется наличием свободных буферов. При критических условиях, ведущих к перегрузкам сети и возникновению тупиков, прием пакетов ограничивается таким образом, чтобы в буферной памяти в любой момент времени находилось по крайней мере по одному пакету классов I и 2 или по резервному пустому буферу для таких пакетов, а для принятых пакетов класса 3 устанавливаются принудительно фиксированные маршруты сначала вверх до корневой ЭМ, а затем вниз по корневому покрывающему дереву D_1 подсистемы до принимающей ЭМ. Алгоритм исключает возникновение дедлоков путем перенаправления при критических условиях пакета класса 3 в корневую ЭМ подсистемы. Указанное перенаправление в $ЭМ_i$ заключается в засылке пакета в очередь выходного направления с номером $G_i(A_1)$ независимо от адреса его приемника. После достижения корневой ЭМ дальнейший путь пакета проходит вниз по покрывающему дереву D_1 и определяется адресом его приемника. Тем самым ликвидируется возможность образования циклов из ЭМ с заполненными буферными пулами.

Передача пакета по линии межмашинной связи в ВС МИКРОС, состоит из трех этапов [IO]: передачи заголовка, передачи адресного пакета, передачи информационного пакета. Предварительная передача заголовка в $ЭМ_i$ имеет целью выделение свободной памяти в $ЭМ_i$ для последующего приема адресного пакета и информационного пакета. Если свободный массив памяти требуемой длины есть, то происходит передача адресного пакета (содержащего адрес приемника). При этом производится обращение к процедуре DEADLOCK, производящей анализ возможности возникновения дедлока в подсистеме и предотвращающей эту ситуацию. Обозначим текущее число свободных буферов в буферном пуле $ЭМ_i$ (FREBUF) через δ_i.

Ниже описана процедура DEADLOCK .

По адресу A_i $ЭМ_i$, исполняющей путевую процедуру, и адресу A_j (содержащемуся в адресном пакете) принимающей $ЭМ_j$ определяется номер p выходного направления, по которому необходимо отправить по-

лученный пакет. Если $p = 0$, то пакет предназначен для $ЭМ_i$ и происходит прием пакета. Если $p \neq 0$, то определяется номер класса, к которому принадлежит пакет.

Пусть пакет принадлежит классу I. Если в буферном пуле один пустой буфер, нет пакетов класса 2 и есть хотя бы один пакет класса I, т.е. $\delta_i = 1$, $\{P_i^1\} \neq \emptyset$, $\{P_i^2\} = \emptyset$, то пакет не принимается. В остальных случаях пакет принимается.

Пусть пакет принадлежит к классу 2. Если в буферном пуле один пустой буфер, нет пакетов класса I и есть хотя бы один пакет класса 2, т.е. $\delta_i = 1$, $\{P_i^1\} = \emptyset$, $\{P_i^2\} \neq \emptyset$, то пакет не принимается. В остальных случаях пакет принимается.

Пусть пакет принадлежит к классу 3. Пакет не принимается в буферный пул $ЭМ_i$ при выполнении тех же условий, при которых не принимается пакет класса 2. Если $\delta_i > 2$ или $\delta_i = 2$ и $\{P_i^1\} \cup \{P_i^2\} \neq \emptyset$, то пакет принимается без перенаправления. В остальных случаях пакет принимается в буферный пул $ЭМ_i$ с перенаправлением, т.е. данный пакет ставится в выходную очередь $p = G_i(A_1)$, где A_1 – адрес корневой ЭМ.

При реализации рассмотренной процедуры DEADLOCK в ВС гарантируются отсутствие дедлоков при передаче сообщений и достижение каждым пакетом своего приемника за конечное время. Максимальное увеличение времени передачи пакета в случае перенаправления равно $2DT$, где D – диаметр подсистемы, T – время передачи пакета между соседними ЭМ, включая время задержки в очередях.

Выделение памяти для буферного пула под сообщения производится пользователем путем задания длины пакета (PACLEN) и максимального числа свободных буферов в подсистеме (MAXBUF). Использование процедуры предотвращения дедлоков происходит при задании пользователем признаков РП (разрешение перенаправления) и ДР (дифференцированный обмен) в поле SIGN сообщения. При задании признаков ДР и ЗП (запрет перенаправления) работает программа дифференцированного обмена без программы DEADLOCK.

Литература

I. КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Вычислительные системы с программируемой структурой. –Электронное моделирование, 1979, №1, с. 42–52.

2. ХОРОШЕВСКИЙ В.Г. Вычислительная система МИКРОС. – Новосибирск, 1983. – 52 с. (Препринт/ИМ СО АН СССР: № 38).

3. ДИМИТРИЕВ Ю.К., КОРНЕЕВ В.В., ХОРОШЕВСКИЙ В.Г. Вычислительная система с программируемой структурой МИКРОС. -В кн.: Вычислительные системы с программируемой структурой (Вычислительные системы, вып. 94). Новосибирск, 1982, с. 3-15.

4. ЯКУБАЙТИС Э.А. Архитектура вычислительных сетей. -М.: Статистика, 1980. - 279 с.

5. JONES A.K. The object model: A conceptual tool for structuring software.- In: Operating Systems: Advanced course. Lecture Notes in Computer Science,v.60, Berlin, 1978, p.8-18.

6. КОРНЕЕВ В.В., МОНАХОВ О.Г., ТАРКОВ М.С. Ядро операционной системы ЭМ вычислительной системы с программируемой структурой.-В кн.: Однородные вычислительные системы (Вычислительные системы, вып. 90). Новосибирск, 1981, с. 22-42.

7. КОРНЕЕВ В.В., МОНАХОВ О.Г. О децентрализованном распределении заданий в вычислительных системах с программируемой структурой. -Электронное моделирование, 1981, № 6, с. 15-22.

8. КОРНЕЕВ В.В., МОНАХОВ О.Г. Организация межмашинных взаимодействий в вычислительных системах с программируемой структурой. -Электронное моделирование, №5, 1980, с. 16-22.

9. МОНАХОВА Э.А. Алгоритм предотвращения дедлоков при передаче сообщений в вычислительных системах с программируемой структурой и сетях. - Настоящий сборник, с. 37-44.

10. ЗАДОРОЖНЫЙ А.Ф., КОРНЕЕВ В.В., ТАРКОВ М.С. Об организации коммуникаций между процессами в вычислительной системе МИКРОС.-Настоящий сборник, с. 70-84.

Поступила в ред.-изд.отд.
2 октября 1984 года