

УДК 681.142.4

МИНИМАКС - ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ

Н.Н. Миренков

Описываются основные задачи, структура и состав программного обеспечения, разрабатываемого для однородной вычислительной системы (ОВС) [1] МИНИМАКС. Рассматриваются особенности работы этой системы в режиме коллективного доступа.

МИНИМАКС относится к вычислительным системам с программно-коммутируемыми связями между элементарными машинами (ЭМ). Система строится [2] на базе технических и программных средств АСВТ-М.

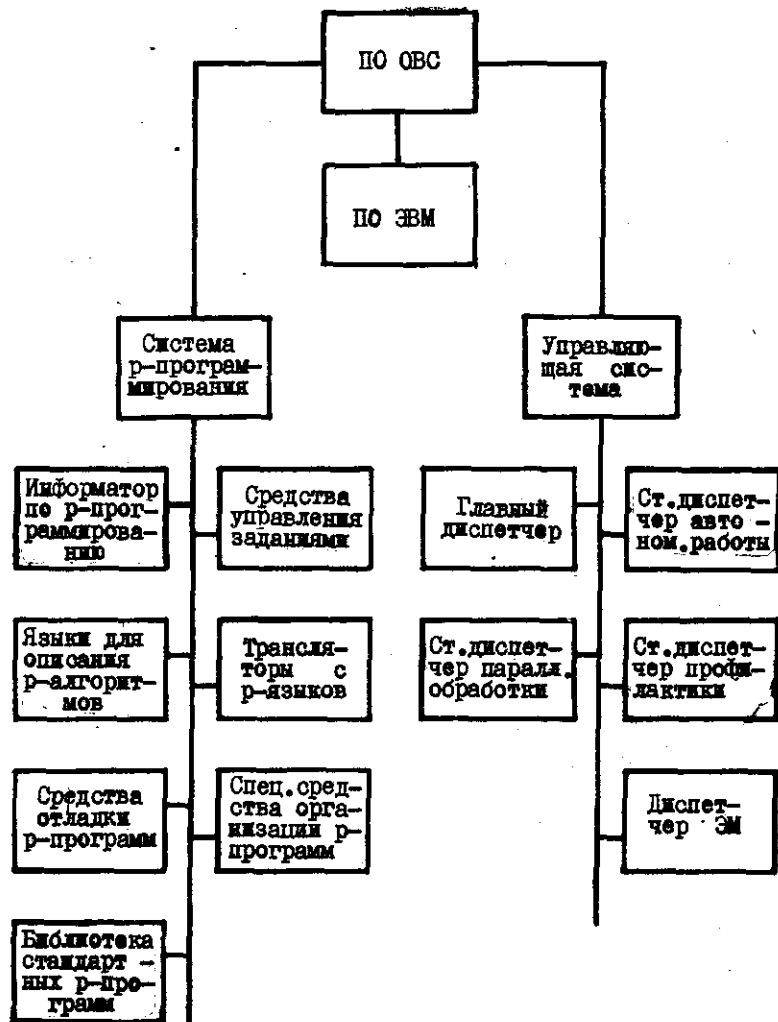
Программное обеспечение общего назначения для ОВС (см. рисунков), как и для любой современной электронной вычислительной машины (ЭВМ), состоит из двух частей:

- 1) управляющей системы, организующей счет задач;
- 2) системы программирования, включающей информатор по программированию, языки, трансляторы и т.п.

Программное обеспечение ОВС обладает следующими основными свойствами:

- 1) независимостью от числа ЭМ в системе;
- 2) функциональной идентичностью при реализации на любой ЭМ;
- 3) изменяемостью модулей и пополняемостью их состава;
- 4) возможностью использования ОВС в режиме коллективного доступа.

Программное обеспечение разрабатывается в рамках единых требований, накладываемых управляющей системой.



Состав программного обеспечения ОВС -МИНИМАКС

Управляющая система. Основными задачами управляющей системы являются обеспечение связи с внешним миром, рациональное использование ресурсов системы, продолжение счета при выходе машин из строя, синтез в ОВС различных по числу машин и топологии подсистем, спецификация функций подсистем.

Выделяются следующие основные режимы функционирования подсистем: автономной работы ЭМ, параллельной обработки, профилактики, диспетчера.

В режиме автономной работы осуществляется непосредственная связь пользователя с одной ЭМ (а через неё и со всей системой). При этом обеспечиваются возможности счета, трансляции, отладки, редактирования, сегментирования на одной машине, а также доступ к системной информации, формирование заданий системе, связь пользователей, работающих за разными терминалами.

В режиме параллельной обработки реализуются р-программы [1] и продолжение их счета при выходе машины из строя.

В режиме профилактики тестируются ЭМ программными и(или) аппаратными средствами.

В режиме диспетчера выполняются работы, связанные с управлением системой.

Для принятия решений в сложных мультипрограммных ситуациях, когда необходимо, с одной стороны, действовать немедленно, а с другой - попытаться лучше понять ситуацию, предлагается управляющая система иерархической структуры с блоками, названными "Диспетчер", "Старший диспетчер", "Главный диспетчер".

Главный диспетчер осуществляет разбиение системы на подсистемы, определяет режимы работы для образованных подсистем, управляет Старшими диспетчерами и операциями ввода/вывода для системных внешних устройств^{х)}. При реализации последовательности функциональных состояний ОВС [3] Главный диспетчер организует: динамическое (статическое) функционирование ОВС при решении заданного набора программ или стохастическое функционирование при обработке случайного потока программ

х) Каждая ЭМ ОВС может иметь свои внешние устройства, называемые индивидуальными, и, кроме того, иметь доступ к выделенным внешним устройствам, называемым системными.

[4]; распознает подсистемы заданной структуры [5]. Организация подсистем и определение режимов работы осуществляется им с использованием 1) директив оператора; 2) характеристик поступивших параллельных программ; 3) информации о выходе машин из строя.

Управление Старшими диспетчерами выполняется на основе правила поочередного обслуживания^{*}) и информационных таблиц. Правило определяет передачу управления одному из Старших диспетчеров с частотой, пропорциональной приоритету, присвоенному соответствующему режиму. Информационные таблицы содержат данные о завершении операций ввода/вывода на системных внешних устройствах, информацию служб времени, приказы на перевод ЭМ из одного режима в другой, информацию о состоянии системы и др. Управление операциями ввода/вывода на системных внешних устройствах Главный диспетчер осуществляет по запросам от Старших диспетчеров. Эти запросы содержат вид операции, тип внешнего устройства, объем и имена передаваемых массивов.

Старший диспетчер управляет работой подсистем. Выделяются три типа Старших диспетчеров [6]: для автономной работы, режимов профилактики и параллельной обработки. Основные функции этих диспетчеров — реализация очередности обслуживания ЭМ (подсистем) на основе заявленного приоритета или вида выполняемой работы; предоставление рабочих каналов для доступа к системным внешним устройствам и подсистеме, работающей в режиме диспетчера; прием и анализ запросов на ввод/вывод для системных внешних устройств от программ и директив пользователей; формирование заданий Главному диспетчеру; реализация обменов с ЭМ (подсистемой), затребовавшей ввод/вывод для системных внешних устройств; тестирование ЭМ.

Диспетчер управляет работой ЭМ: обеспечивает вход в систему и распознавание директив пользователя, взаимодействует со Старшим диспетчером, анализирует аварийные ситуации, управляет вводом/выводом для индивидуальных внешних устройств, участвует в реализации операторов обменных взаимодействий [7].

^{*}) В отдельных случаях может использоваться правило абсолютного приоритета [6].

Функции Диспетчера выполняются каждой ЭМ, Главного и Старших диспетчеров — подсистемой, работающей в диспетчерском режиме; при наличии в этой подсистеме нескольких ЭМ, Главный и Старшие диспетчеры реализуются в виде идентичных параллельных ветвей, каждая из которых обслуживает выделенную ей сферу влияния. Основные взаимодействия ветвей в этом случае связаны 1) с дублированием некоторой информации друг у друга, 2) с доступом к файлам по запросам из других сфер влияния, 3) с перераспределением сфер влияния в связи с изменением мультипрограммной ситуации.

Система программирования. Основная задача, возлагаемая на систему программирования, — обеспечение достаточно широких возможностей при написании и отладке параллельных программ. Для этого введены следующие элементы в программное обеспечение ОВС.

1. Языки для записи параллельных алгоритмов (ОВС-АЛГОЛ, ОВС-ФОРТРАН, ОВС-МНЕМОКОД [7,8]), полученные путем расширения соответствующих языков операторами системных взаимодействий.

2. Трансляторы с ОВС-языков, позволяющие при автоматизации параллельного программирования использовать трансляторы ЭВМ АСВТ-М-6000.

3. Информатор по параллельному программированию, обеспечивающий обучение пользователей представлению параллельных процессов в виде совокупности ветвей, взаимодействующих между собой с помощью системных операторов [7].

4. Средства отладки р-программ, анализирующие качество р-программ и выявляющие ошибки взаимодействий р-ветвей путем моделирования параллельного процесса на одной ЭМ [9].

5. Средства специальной организации р-программ: а) сегментирования р-ветви, дающего экономию оперативной памяти, необходимой для хранения всей р-программы [10]; б) подготовки высоконадежных вычислений, обеспечивающих продолжение счета при сбоях ЭМ и выходе их из строя.

6. Средства управления заданиями (языки директив), облегчающие пользователю реализацию различных видов работ (особенно связанных с новыми возможностями ОВС: запуск и снятие р-программ, создание и уничтожение р-файлов, задание графика работы подсистем, перевод ЭМ в режим профилактики и др.).

7. Библиотека стандартных параллельных программ (СПП), упрощающая процесс параллельного программирования.

Функционирование ОВС под управлением рассмотренной операционной системы обеспечивает возможности мультимедиа и пакетной обработки, режимов реального времени и параллельного счета.

Особенности мультимедиа. Существующие и проектируемые системы коллективного пользования представляют собой совокупность аппаратных и программных средств, включающих в себя мощный процессор (или несколько процессоров) с большой памятью, разветвленную сеть терминальных устройств и развитое программное хозяйство. Пользователи имеют эффект единичного использования машины, хотя на самом деле они конкурируют между собой за обладание ресурсами и делают их во времени. Режим разделения времени обеспечивает качественно новые возможности использования вычислительных средств, но требует значительных усилий на его организацию.

Некоторой альтернативой режиму разделения времени можно считать растущее потребление мини-машин. Они просты в использовании, имеют небольшую цену, но ограничены в областях применения. ОВС из мини-машин расширяют области применения этих машин, причем для достаточно большого круга задач оказываются более эффективными с точки зрения производительности, надежности, стоимости и т.д., чем одна или даже несколько ЭВМ третьего поколения [2, II]. При работе ОВС в режиме коллективного пользования реализуется не столько режим разделения времени, сколько режим разделения мини-машин, каждая из которых рассматривается как единица ресурса системы. Это значительно упрощает реализацию ряда модулей программных средств. Обладая основными возможностями режима разделения времени, построенного на базе мощных процессоров, ОВС из мини-машин имеют дополнительные свойства, определяемые возможностями программно-управляемой структуры.

Время реакции систем. Реактивность вычислительных систем мультимедиа характеризуется временем τ , через которое появляется ответ на директиву пользователя.

В системе МИНИМАКС число^{*)} одновременно работающих пользователей равно числу ЭМ, функционирующих в режиме автономной работы. Это число меняется в соответствии с заданиями пользователей и (или) по директиве оператора.

Находясь за пультом ЭМ, пользователь непосредственно обладает процессором мини-ЭВМ, её оперативной памятью и индивидуальными внешними устройствами; кроме того, он конкурирует с другими пользователями за системные внешние устройства и подсистемы, функционирующие в режиме параллельной обработки. По отношению к пользователю в ОВС МИНИМАКС выделяется четыре иерархических уровня памяти:

- 1) оперативная память ЭМ, с которой пользователь непосредственно связан;
- 2) внешняя память на индивидуальных внешних устройствах элементарной машины;
- 3) внешняя память на системных внешних устройствах;
- 4) оперативная и внешняя памяти других ЭМ.

От того, какие уровни ($i = 1, 2, 3, 4$) памяти приходится использовать системе для работы с файлами программиста, зависит время τ ответа на его директиву. Задача считается i -го класса, если i есть максимальный уровень памяти, используемой при выполнении задачи.

Среднее значение τ :

$$\bar{\tau} = \sum_{i=1}^4 \varepsilon_i \tau_i,$$

где ε_i - весовые коэффициенты, характеризующие долю задач (запросов) i -го класса из общего числа задач, решаемых на системе; τ_i - среднее время ответа для задач i -го класса.

τ_1 определяется в основном временем работы программ пользователя над оперативной памятью ЭМ.

τ_2 связано с типами внешней памяти, используемой в индивидуальных внешних устройствах:

$$\tau_2 = f_1 \tau_1 \left(\sum_{j=1}^n t_j \alpha_j \right),$$

где t_j , α_j - время обращения и весовой коэффициент, характеризующие устройство j -го типа, n - число различных типов устройств.

*) Число пользователей может быть большим, если программное обеспечение одной ЭМ предусматривает режим разделения времени.

ройств памяти. Очевидно, что в случае доступа к ЭМ, работающей в режиме автономной работы, время ответа на директивы, связанные с задачами первого и второго классов, не зависит от числа пользователей, взаимодействующих в данный момент с системой.

τ_3 определяется τ_2 , средним временем θ_1 ожидания информации со стороны ЭМ, работающих в режиме диспетчера, и типами памяти, используемой в системных внешних устройствах:

$$\tau_3 = f_2(\tau_2, \theta_1, \sum_j^n t_j' \alpha_j'),$$

где t_j', α_j', n' - величины, аналогичные t_j, α_j, n .

θ_1 зависит от числа ЭМ и числа подсистем, функционирующих соответственно в режимах автономной работы и параллельной обработки. Эти режимы конкурируют между собой при обращении к Главному диспетчеру. Методами теории массового обслуживания, предполагая, что работа режима параллельной обработки является фоновой, можно получить следующие выражения для θ_1 :

$$\theta_1 = \sum_{k=1}^L \rho_k \sum_{s=0}^{k-1} \int_0^{\infty} g_s(t) dt,$$

$$\rho_k = \frac{L!}{(L-k)!} \left(\frac{\lambda}{\nu}\right)^k P_0, \quad P_0 = \left[\sum_{k=0}^L \frac{L!}{(L-k)!} \left(\frac{\lambda}{\nu}\right)^k \right]^{-1},$$

$$g_s(t) = e^{-\nu t} \frac{(\nu t)^s}{s!},$$

где L - число ЭМ, связанных с режимом автономной работы; λ - интенсивность простейшего потока заявок от этих ЭМ к Главному диспетчеру; ν - параметр показательного закона обслуживания заявок.

τ_4 связано с оперативным заданием режима параллельной обработки и определяется величиной типа τ_3 , а также средним временем θ_2 ожидания доступа к подсистеме требуемого ранга:

$$\tau_4 = f_3(\tau_3, \theta_2).$$

В предположении простейшего потока заявок с интенсивностью γ и показательного закона обслуживания с параметром μ :

$$\theta_2 = \frac{\gamma}{\mu(\mu - \gamma)}.$$

Функции f_i ($i = 1, 2, 3$) для многих конкретных типов задач могут считаться линейными, квадратичными или логарифмическими и уточняться на основе статистических данных.

При пиковых нагрузках время ожидания доступа к системным внешним устройствам уменьшается дублированием некоторых файлов на индивидуальных внешних устройствах.

Распределение сфер влияния между блоками диспетчеров. Надежность управления и быстрая реакция системы на директивы, связанные с любыми классами задач, обеспечивается подсистемой, работающей в режиме диспетчера. Диспетчерские программы представляют собой р-программы, ветви которых реализуются на выделенных для них машинах. Каждая ветвь имеет свою сферу влияния - подсистему, которой она управляет. Неравномерность нагрузок на ветви диспетчерских р-программ, связанная с постоянно меняющейся мультипрограммой ситуацией, устраняется перераспределением ЭМ по сферам влияния и перемещением блоков информации с одних системных внешних устройств на другие.

Нагрузку на ветвь ℓ характеризуют следующие показатели:

- 1) мгновенный коэффициент нагрузки $k_{\mu}^{\ell}(\Delta T) = \frac{1}{\Delta T} \sum_{h=1}^N t_h$, где N - число обращений к управляющей системе за время ΔT , t_h - время обслуживания h -го обращения;
- 2) средний коэффициент нагрузки $k_c^{\ell} = k_{\mu}^{\ell}(T)$, $T \gg \Delta T$;
- 3) обобщенный коэффициент нагрузки $k_0^{\ell} = \alpha k_c^{\ell} + (1 - \alpha) k_{\mu}^{\ell}$, $0 \leq \alpha \leq 1$.

Для каждого блока информации, находящейся на системном внешнем устройстве, вводятся (по аналогии с [14]) коэффициенты активности его использования:

- 1) средний коэффициент активности

$$r_c^j(T) = c_j / T, \quad j = 1, 2, \dots, n,$$

где c_j - общее количество обращений к блоку за время T из j -й сферы влияния, n - число таких сфер;

- 2) мгновенный коэффициент активности

$$r_{\mu}^j = \beta \frac{1}{t_{h_0}^j - t_{h_0-1}^j} + (1 - \beta) \frac{1}{(t_{h_0}^j - t_{h_0-2}^j)},$$

где $t_{h_0}^j, t_{h_0-1}^j, t_{h_0-2}^j$ - моменты времени трех последних обращений к блоку из сферы j ; $0 \leq \beta \leq 1$;

3) коэффициенты использования блока

$$r_j = \gamma_1 r_0^j + (1 - \gamma_1) r_M^j, \quad 0 \leq \gamma_1 \leq 1.$$

Процедуры по уменьшению взаимодействий ветвей диспетчерских программ и перераспределению сфер влияния основываются на следующем. Блок информации пересылается из внешней памяти ветви j_1 во внешнюю память ветви j_2 , если выполняется система неравенств:

$$R > r_{j_2} > \xi_1 r_{j_1}, \quad \frac{r_{j_2}}{\sum_{k=1}^n r_k} \geq \xi_2; \quad k_0^{j_2} < k_1,$$

где R, ξ_1, ξ_2, k_1 — заданные константы.

Блок информации дублируется во внешней памяти ветви j , если $r_j \geq R$. ЭМ из сферы влияния j^* с $k_0^{j^*} = \max k_0^j$ переводится в сферу влияния j^{**} с $k_0^{j^{**}} = \min k_0^j$, если выполняется условие $\xi_3 k_0^{j^*} > k_0^{j^{**}}$ ($\xi_3 < 1$). При этом для перевода выбирается ЭМ, имеющая наибольшую активность при обращении к блокам информации, находящейся на системных внешних устройствах, связанных с j^{**} -ой сферой влияния.

Константы, используемые в приведенных формулах, задаются на основе используемых типов вычислительных работ или определяются моделированием и далее уточняются в процессе эксплуатации.

Защита памяти. Защита различных уровней внешней памяти автоматически обеспечивается тем, что прямое программирование операций ввода/вывода запрещено. Пользователь запрашивает выполнение этих операций специальным обращением к Диспетчеру элементарной машины, который выполняет заказ сам, если тот связан с индивидуальными внешними устройствами, или передает Старшему (Главному) диспетчеру, если тот связан с системными внешними устройствами. При этом проверяется возможность выполнения заказанных операций до того, как начать их выполнение.

Общие и групповые массивы данных (архивы) обычно организуются на системных внешних устройствах, личные — на индивидуальных внешних устройствах; первые, как правило, используют носители с произвольным доступом типа диск, вторые могут использовать и магнитную ленту. Защита общих и групповых массивов может обеспечиваться известными режимами доступа, основан-

ными на паролях, списках пользователей, справочных зонах, и дублированием наиболее важных записей на индивидуальных внешних устройствах. Защита личных массивов обеспечивается размещением их на накопителях со сменными носителями и обращением к ним только через Диспетчер ЭМ.

Защита на уровне оперативной памяти нужна для предохранения Диспетчера от программ пользователя; её можно выполнить аппаратно, разбив память на две части (защищенную и незащищенную), или программно, например, путем анализа во время трансляции. Защита программ пользователей друг от друга обеспечивается разбиением системы на подсистемы. Каждая задача может изменять состояние регистров настройки ЭМ только в своей подсистеме. Изменение регистров других ЭМ и разбиение системы на подсистемы (управление структурой) выполняется Главным диспетчером.

Надежность вычислений. Программно управляемая структура ОВС позволяет организовывать высоконадежные вычисления в течение длительного времени при минимальном вмешательстве со стороны человека. Продолжение счета при сбоях и выходе элементарных машин из строя организуется двумя основными способами: перестройкой р-программ на меньшее число машин или заменой неисправной машины. В любом случае процесс решения задачи разбивается на шаги по времени, после каждого из которых реализуется система тестов, принимающая решение о достоверности счета на предыдущем шаге. При положительных результатах тестирования необходимая информация запоминается в индивидуальных внешних устройствах и дублируется в соседних ЭМ, при отрицательных — выбирается подсистема для продолжения счета, который повторяет вычисления неудавшегося шага.

Средства параллельного программирования [7,8], включающие в себя методику крупноблочного распараллеливания [1,13,14], позволяют создавать универсальные параллельные программы, настраиваемые на число ветвей как на параметр. Это облегчает автоматическую перестройку непосредственно самих р-программ с одного числа ЭМ на другое. Для продолжения прерванных вычислений нужно перераспределить массивы, над которыми р-программа работает. Это перераспределение требует заранее заготовленных ре-

резервов оперативной памяти и времени на проведение пересылок информации. Для построения универсальных р-программ средства программирования должны обеспечивать динамическое распределение памяти. Такую организацию продолжения вычислений при выходе из строя и сбоях ЭМ особо выгодно применять для тех классов задач, которые не требуют перераспределения массивов^{*)}, а также при отсутствии резервных ЭМ.

Организация продолжения вычислений, связанная с заменой вышедшей из строя ЭМ на исправную, не требует перестройки р-программ, заранее заготовленных резервов ОП и значительно уменьшает время на перезапуск вычислений. Этот способ базируется на оперативной перестройке функционального состояния системы и выявлении такой машины для замены, которая сохраняет структуру подсистемы. Обычно такая машина берется из подсистем, реализующих программы меньшего приоритета. При этом Главным диспетчером решается следующая задача. Задано множество машин $M = \bigcup_{i=1}^L M_i$ и его подмножество $M' = \bigcup_{k=1}^{L-1} M_{i_k}$. Требуется из M выбрать подмножество $M^2 = \bigcup_{k=1}^{L-1} M_{i_k}$ ($M^2 \cap M' = M'$), позволяющее организовать подсистему нужной структуры (например, связанной или линейной). В основе алгоритмов выбора подмножества M^2 могут использоваться методы [5].

Для подготовки задачи к счету в условиях изменения числа машин в системе пользователь должен обратиться к специальному блоку программного обеспечения. Этот блок подготавливает р-программы к реализации с пошаговым тестированием ЭМ или к счету с дублированием вычислений на двух подсистемах. Во втором случае тестирование ЭМ выполняется только при несовпадении результатов вычислений в подсистемах. Каждый пользователь может предложить для проверки ЭМ свои тесты, а для фиксирования состояния ЭМ - места оперативной памяти и файлы, которые не нужно запоминать или сравнивать.

Г е н е р и р о в а н и е с и с т е м ы. Задача приспособления, адаптации системы к конкретному составу оборудования и к конкретным нуждам пользователей, может решаться с помощью следующих возможностей:

*) Задачи, связанные, например, с работой в реальном времени, где вычисления определяются вновь поступившими данными.

1. Выбора числа ЭМ в системе. Однородность и переменность структуры ОВС обеспечивают произвольное наращивание числа машин в системе, что позволяет получать требуемую общую производительность.

2. Выбора размера подсистемы, работающей в режиме диспетчера. Этот размер влияет на время реакции системы, позволяет получать нужную скорость обменных операций с системными внешними устройствами и требуемую надежность управляющей системы.

3. Назначения режима управления. Статическое управление оставляет неизменным и разбиение системы на подсистемы и типы работ для каждой подсистемы; динамическое - устанавливает функциональное состояние системы на основе заранее заданного графика; стохастическое - то же самое, но на основе мультипрограммной ситуации в каждый момент времени, а смешанное - на основе перечисленных выше режимов для разных подсистем.

4. Разбиения имеющихся внешних устройств на индивидуальные и системные. Рациональное разбиение может снизить время реакции системы, обеспечить более высокую децентрализацию функционирования ЭМ как при автономной работе, так и в режиме параллельной обработки.

5. Генерирования Диспетчера ЭМ. Это генерирование позволяет настроить Диспетчер на аппаратуру ЭМ (в частности, на индивидуальные внешние устройства) и при необходимости включить функции мультипрограммного супервизора, позволяющего в режиме автономной работы рассматривать ЭМ как совокупность более мелких ресурсов системы.

6. Генерирования Главного и Старшего диспетчеров. Позволяет настроить диспетчеры на аппаратуру системы (в частности, на системные внешние устройства и число ЭМ), выбрать алгоритмы планирования в зависимости от режима управления.

7. Выбора комплекта трансляторов, состава библиотеки и обслуживающих программ. Подбор средств программирования для классов задач, которые предполагается решать на системе, позволяет создавать высокоэффективные программы за короткие сроки.

Рассмотренные возможности позволяют получать системы из мини-машин, обладающие следующими свойствами: способностью к быстрой мобилизации ресурсов для удовлетворения пиковых нагрузок; сочетаемостью ресурсоемких вычислений с оперативным обслуживанием большого числа внешних воздействий; наличием больших,

но быстродоступных информационных массивов с высоконадежной защитой; разнообразием внешних устройств и языковых средств.

Л и т е р а т у р а

1. ЕВРЕЙНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск, "Наука", 1966.
2. ВИНУКОВ В.Г., ДМИТРИЕВ Ю.К., ЕВРЕЙНОВ Э.В., КОСТЕЛЯНСКИЙ В.М., ЛЕХОВА Г.М., МИРЕНКОВ Н.Н., РЕЗАНОВ В.В., ХОРОШЕВСКИЙ В.Г. Однородная вычислительная система из мини-машин. - В кн.: Вычислительные системы. Вып. 51. Новосибирск, стр. 127-146.
3. МИРЕНКОВ Н.Н. Диспетчирование в однородных вычислительных системах. - В кн.: Материалы 3-й Всесоюзной конференции по однородным вычислительным системам и средам. Таганрог, 1972, стр. 51-54.
4. КРЫЛОВ Э.Г., МИРЕНКОВ Н.Н. Алгоритмы планирования функциональных состояний однородной вычислительной системы. Отчет ИМ СО АН СССР, 1973.
5. МИРЕНКОВ Н.Н., ФИШЕРМАН С.Б. Алгоритмы распознавания подсистем заданных структур в однородных вычислительных системах. Отчет ИМ СО АН СССР, 1974.
6. КЕРБЕЛЬ В.Г., КОРНЕЕВ В.Д., МИРЕНКОВ Н.Н., ШЕРБАКОВ Е.В. Управляющая программа системы МИНИМАКС. - Настоящий сборник, стр. 129-142.
7. КЕРБЕЛЬ В.Г., КОЛОСОВА Ю.И., КОРНЕЕВ В.Д., МИРЕНКОВ Н.Н., ШЕРБАКОВ Е.В. Средства программирования системы МИНИМАКС. - Настоящий сборник, стр. 143-152.
8. ГРИШАЕВА Н.К., КЕРБЕЛЬ В.Г., КОЛОСОВА Ю.И., КОНСТАНТИНОВ В.И., КОРНЕЕВ В.Д., ЛЕВАГИНА Т.А., МИРЕНКОВ Н.Н., ФИШЕРМАН С.Б. Язык параллельных алгоритмов. - В кн.: Вычислительные системы. Вып. 57. Новосибирск, 1973, стр. 33-54.
9. КОЛОСОВА Ю.И. Комплекс средств производства параллельных программ. - В кн.: Вычислительные системы. Вып. 57. Новосибирск, 1973, стр. 98-114.
10. МИРЕНКОВ Н.Н., ГАМИДОВ В.М. Сегментирование параллельных программ. Отчет ИМ СО АН СССР, 1974.
11. RILEY W.B. Minicomputer network - challenge to main computers? - "Electronics", 1971, N 29, p. 56-62.
12. БЕРТЭН Ж., РИТУ М., РУЖИЕ Ж. Работа ЭВМ с разделением времени. М., "Наука", 1970.
13. КОСАРЕВ Ю.Г. Распараллеливание по циклам. - В кн.: Вычислительные системы. Вып. 24. Новосибирск, 1967, стр. 3-15.
14. МИРЕНКОВ Н.Н. Параллельные алгоритмы для решения задач на однородных вычислительных системах. - В кн.: Вычислительные системы. Вып. 57. Новосибирск, 1973, стр. 3-32.

Поступила в ред.-изд. отд.
9 июля 1973 года