ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

11 - 7941

В.М.Котов

ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ УПРАВЛЯЮЩЕЙ ЭВМ СПИРАЛЬНОГО ИЗМЕРИТЕЛЯ

1974

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И АВТОМАТИЗАЦИИ

11 - 7941

В.М.Котов

ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ УПРАВЛЯЮЩЕЙ ЭВМ СПИРАЛЬНОГО ИЗМЕРИТЕЛЯ

Направлено в ПТЭ

Котов В.М. 11 - 7941

Операционная система для управляющей ЭВМ
спирального измерителя

В работе описывается диспетчер операционной системы (ОС) сканирующего автомата "Спиральный измеритель" (СИ). Программа-диспетчер обеспечивает смешанную дисциплину обслуживания заявок от электронных устройств СИ как с относительными приоритетами для вклинивающихся программ обслуживания прерываний, так и абсолютными приоритетами для основных программ, работающих при разрешенном прерывании. Приводятся оценки эффективности выбранной дисциплины диспетчеризации.

Препринт Объединенного института ядерных исследований.
Дубна, 1974

ВВЕДЕНИЕ

Создание развитых мультиплексорных каналов, обеспечивающих работу электронных устройств спирального измерителя (СИ) в режиме разделения времени /I/, позволило сильно упростить сами устройства, ограничив их функции только предварительной обработкой и формированием сигналов, полученных в отсчетных системах перед передачей их в ЭВМ, а также хранением в статусно-командных регистрах выданных из ЭВМ команд управления в течение времени, необходимого для отработки этих команд исполнительными механизмами. Выполнение функциональных алгоритмов обеспечивается, в основном, программой управления, и поэтому можно предположить, что все дальнейшее усовершенствование системы, увеличение её производительности и повышение надёжности может быть достигнуто за счет развития программ управления. Выбор способа разделения программы управления на модули, а также структура самих модулей были определены, исходя из этих соображений.

Считая, что все изменения при дальнейшем развитии системы должны укладываться в рамки существующих аппаратурных средств, мы определили один из основных модулей операционной системы, называемый в дальнейшем монитором. Этот модуль включает в себя прог-

рамму-диспетчер и набор подпрограмм обслуживания прерываний; предполагается, что все дальнейшие изменения программ не должны затрагивать его структуру.

Диспетчер и организация программных приоритетов

Идеальная эффективность управляющей системы, зависящая от качества функциональных алгоритмов, может быть достигнута только при отсутствии потерь информации в процессе управления. Конечные характеристики управляющей ЭВМ по быстродействию и необходимость решения задач в реальном масштабе времени снижает эффективность управления на величину, которая в общем случае может рассматриваться как функционал, зависящий от потока заявок и параметров ЭВМ:

$$C = \sum_{i=1}^{n} di\, \lambda i\, Wi , \qquad (3.I)$$

где Wi – средняя длительность ожидания в очереди заявок i -го приоритета;

di – относительная важность заявок i – го типа;

λi – интенсивность заявок i – го потока;

n – число потоков заявок.

В первом приближении потери можно представить как сумму следующих составляющих:

– потери на задержку включения и начала работы данной программы; особенно это важно при обслуживании заявок по управлению внепрограммным обменом;

– потери из-за задержек в решении периодических задач по контролю и определению отдельных параметров системы; обычно это вызывает ухудшение точности вычисления их величин, а значит и потерю качества регулирования.

Неравноценность задач по допустимому времени задержки или допустимой вероятности пропуска решения данной задачи позволяет изменять суммарные задержки в обслуживании, уменьшая задержки в обслуживании более важных заявок за счет увеличения задержек для остальных заявок, и тем самым повышать общую эффективность управляющей системы при заданных характеристиках ЭВМ путем оптимального выбора дисциплины включения задач в реальном масштабе времени. Но, с другой стороны, выбор достаточно сложной дисциплины обслуживания вызывает увеличение потерь оперативной памяти ЭВМ, связанных с размещением диспетчера, и повышает временные затраты на настройку, а следовательно, и на реализацию данной управляющей системы в целом.

Таким образом, при разработке программы-диспетчер для СИ необходимо было решить вопрос о выборе оптимальной дисциплины обслуживания заявок от различных электронных устройств, а также определить общее количество уровней приоритетов в обслуживании. Вместе с этим, должна была быть проведена оценка относительного выигрыша в эффективности применения данной приоритетной дисциплины по сравнению с бесприоритетным обслуживанием, т.е. обслуживанием заявок в порядке их поступления. Кроме того, необходимо было произвести анализ относительной сложности программной реализации выбранной дисциплины обслуживания в рамках принятых средств автоматизации программирования, с учётом состава команд и предоставляемой ассемблером SABR возможности в распределении оперативной памяти. /7,8/

Основным критерием необходимости введения диспетчеризации в обслуживании заявок является величина загрузки ЭВМ обслуживанием заявок внешних устройств, которая определяется отношением: /2/

$$\rho = \sum_{i=1}^{n} \rho_i = \sum_{i=1}^{n} \frac{\lambda_i}{\mu_i}, \qquad (3.2)$$

где: λ_i – интенсивность заявок i – го потока;

μ_i – интенсивность обслуживания i – го потока;

n – общее количество потоков.

Диспетчеризация оправдана только для случая ρ = 0.6 – I.0, потому что при меньших загрузках различие в характеристиках диспетчеризации резко уменьшается. Но для управляющих ЭВМ характерно наличие больших загрузок, т.к. именно этот параметр определяет полноту реализации ресурсов машины.

Стремление к полной автоматизации процесса управления сканирующей системой и повышение производительности определило необходимость работы её блоков в мультипрограммном режиме. Этот режим налагает жесткие требования на устойчивость выполнения функциональных алгоритмов и требует согласования фазовых соотношений в отработке исполнительными механизмами выданных им команд.

Устойчивость процесса управления в мультипрограммном режиме предполагает введение относительной важности отдельных заявок и выполнение требований на максимальное время ожидания их в очереди на обслуживание, что может быть выполнено только при введении приоритетных дисциплин диспетчеризации.

Потребность диспетчеризации и введение штрафов за задержку

в обслуживании определяется также тем, что максимальное использование ЭВМ и передача ей функций по накоплению данных, поступающих с отсчётных систем, обязывает операционную систему обеспечивать быстрое обслуживание заявок по управлению внепрограммным обменом информацией с внешними устройствами.

Применение алгоритмов диспетчеризации с относительными приоритетами всегда, даже в самом благоприятном случае, вносит задержку на обслуживание вновь поступившей заявки, пока не будет окончено начавшееся ранее обслуживание предыдущей заявки.

Наиболее эффективным способом уменьшения времени ожидания в очереди важных заявок является использование прерываний. Разработка и включение в состав электронной аппаратуры сканирующей системы СИ развитого мультиплексорного канала программного прерывания с относительными приоритетами, реализованными аппаратурно внутри мультиплексора /3/, позволяет значительно сократить не только задержки для коротких и важных заявок, передав их обслуживание вклинивающимся подпрограммам, но и уменьшить среднее время ожидания заявок в очереди. Кроме того, благодаря использованию прерываний стало возможным введение дисциплины с абсолютными приоритетами для заявок, программы обслуживания которых относятся к основным программам ОС и выполняются при разрешенном прерывании.

При разработке алгоритмов управления и настройке операционной системы СИ необходимо было выбрать оптимальное соотношение между основными программами ОС, выполняемыми при разрешенном прерывании, и вклинивающимися подпрограммами, вызываемыми по сигналам прерыва-

ния и выполняемыми при запрещенном прерывании. В общем случае эта задача может быть формализована следующим образом: если в ЭВМ поступает n пуассоновских потоков заявок, каждому из которых присвоен свой приоритет, то введением некоторого параметра S, значения которого могут быть произвольно выбраны между $S=\overline{0,n}$, суммарный поток может быть разделен на два класса. Первый класс с приоритетами $\kappa=\overline{1,S}$ соответствует вклинивающимся подпрограммам, обслуживающим сигналы прерывания по дисциплине с относительными приоритетами, а программы обслуживания заявок второго класса $\kappa=\overline{S+1,n}$ являются основными программами и выполняются при разрешенном прерывании по дисциплине с абсолютными приоритетами. При таком смешанном алгоритме диспетчеризации в качестве критерия эффективности удобно использовать линейный функционал: / 2 /

$$C\{S\}=\sum_{\kappa=1}^{n} d_\kappa \lambda_\kappa V_\kappa(S). \qquad (3.3)$$

Здесь $V_\kappa(S)$ – средняя продолжительность пребывания заявок к-го приоритета в памяти ЭВМ с учетом возможных прерываний их обслуживания.

Значение параметра $S=S^*$, минимизирующее величину функционала (3.3), соответствует оптимальному распределению функций между программами, выполняемыми при разрешенном и запрещенном прерывании, т.е. основными и вклинивающимися программами. Таким образом, отдавая обслуживание наиболее важных и коротких заявок вклинивающимся программам, которые могут прерывать любую программу, выполняемую при разрешенном прерывании, можно практически исключить задержку в обслуживании наиболее важных заявок и обеспечить устойчивость выполнения функциональных алгоритмов при мультипрограммном режиме работы.

Заявки по управлению режимом прямого доступа в память относятся к числу наиболее важных и обслуживание их должно производиться по вклинивающимся подпрограммам. Все потоки этих заявок включены в первый класс с приоритетами $\kappa=\overline{1,S}$, и их количество определяет нижний предел D_H при изменении параметра S

Управляющая ЭВМ СИ имеет только один уровень прерывания. Это определяет относительные приоритеты обслуживания в мультиплексоре программного прерывания для вклинивающихся подпрограмм первого класса, и среднее время пребывания заявок к-го приоритета в памяти ЭВМ $V_\kappa(S)$ и выражении функционала (3.3) относится поэтому только к заявкам, обслуживаемым при помощи основных программ при разрешенном прерывании. При минимизации функционала (3.3) увеличение параметра S имеет верхний предел D_B, определяемый допустимым временем h_κ ожидания в очереди заявок высшего приоритета в классе $\kappa=\overline{1,S}$, потому что в первый класс могут попасть программы обслуживания из числа основных программ, имеющих большое время выполнения. Поэтому, хотя функционал (3.3) минимизируется и уменьшается время ожидания заявок к-го типа второго класса $\kappa=\overline{S+1,n}$, время ожидания заявок высшего приоритета из класса $\kappa=\overline{1,S}$ увеличивается. Таким образом, параметр $S=S^*$ должен выбираться в пределах $D_H \le S^* \le D_B$.

В операционной системе СИ нижний предел $D_H=6$ и определяется структурой электронных устройств. Причем, эти потоки заявок сведены в 2-е группы, первая из которых состоит из 2-х потоков заявок отсчетного канала и обслуживается в режиме пакетной обработки, а вторая группа обслуживается, как одиночные заявки с относительными приоритетами.

Оценка величины верхнего предела b_6 при аналитическом подходе затруднительна, так как полученные в работе /4/ для такого режима рекуррентные соотношения средней длительности ожидания заявок в очереди, позволяющие вычислить среднее время, дисперсию и коэффициент вариации длительности ожидания заявок в очереди очень громоздки. Поэтому выбор значения параметра производился в процессе настройки операционной системы при помощи экспериментальных оценок величины задержки для заявок высшего приоритета первого класса при увеличении параметра

В результате такой оценки в первый класс были добавлены 3 потока заявок с одиночным режимом обслуживания. При этом время обслуживания этих одиночных заявок было сделано равным времени обслуживания пакета заявок, имеющих высшие приоритеты в этом классе $\kappa = \overline{1,5}$. Это позволило свести к минимуму увеличение коэффициента вариации длительности ожидания в очереди высших заявок приоритетов, связанное с введением в состав первого класса этих дополнительных заявок.

Если взять теперь в целом оба класса, то между ними соблюдается дисциплина обслуживания с абсолютными приоритетами, так как каждая заявка первого класса может прервать обслуживание любой заявки второго класса, которая дообслуживается после окончания вклинивающихся подпрограмм заявок первого класса.

Следует заметить, что внепрограммный обмен имеет еще более высокий приоритет и может прерывать выполнение вклинивающихся подпрограмм.

Анализ и оценка характеристик входных потоков заявок второго класса, возникающих в процессе выполнения функциональных алго-

ритмов и обслуживаемых по основным программам при разрешенном прерывании, показал, что при выбранном значении параметра $S^* = 9$ в числе потоков второго класса с приоритетами $\kappa = \overline{S+1,n}$ имеется подкласс заявок, связанных с управлением работой функциональных блоков СИ в реальном масштабе времени.

Допустимое время ожидания этих заявок в очереди на обслуживание не может превышать 2–3 мсек, так как в противном случае возможно нарушение устойчивости выполнения алгоритмов или ухудшение качества регулирования. В то же время в состав основных программ входят программы, имеющие очень большое (0.6–I.5 сек) время выполнения. Поэтому соблюдение требований на допустимое время ожидания заявок высшего приоритета в этом классе $\kappa = \overline{S+1,n}$ потребовало введения диспетчеризации обслуживания заявок внутри этого класса по дисциплине с абсолютными приоритетами.

Анализ суммарных временных задержек заявок второго класса с приоритетами $\kappa = \overline{S+1,n}$ может быть проведен с использованием методов теории массового обслуживания. Оценка среднего времени ожидания при таком подходе для диспетчеризации с абсолютными приоритетами приведена в работе /2/ и заключается в следующем:

При произвольном времени обслуживания заявок и выполнении условия стационарности ($\rho < 1$) среднее время пребывания заявки к-го приоритета в ЭВМ, начиная с момента поступления заявки и до завершения её обслуживания, определяется выражением:

$$V_\kappa = \omega_\kappa + h_\kappa , \qquad (3.4)$$

где: ω_κ – значение среднего времени ожидания очереди заявки к-го приоритета от момента поступления её до начала обслуживания ;

$h_к$ характеризует среднюю длительность пребывания заявки к-го приоритета в памяти ЭВМ от начала и до завершения её обслуживания с учётом возможных прерываний.

Для случая, когда заявка после прерывания её выполнения заявкой с высшим приоритетом возвращается обратно в очередь и ожидает там продолжения обслуживания, можно получить среднюю длительность ожидания в очереди заявок к-го приоритета по формуле:

$$W_к = V_к - T_к ,$$

Здесь $T_к$ представляет собой чистое время, необходимое для выполнения заявки машиной. Развернутая формула этой величины /2/:

$$W_к = \frac{T_к R_{к-1}}{1-R_к} + \frac{\frac{1}{2}\sum_{i=1}^{к} \rho_i T_i (1+\upsilon_i^2)}{(1-R_{к-1})(1-R_к)}, \qquad (3.5)$$

T_i и $T_к$ – чистое время выполнения в ЭВМ заявки i-го и к-го приоритетов;

$R_{к-1} = \sum_{i=1}^{к-1} \rho_i$ – суммарная загрузка ЭВМ обслуживанием заявок к-I-го и высших приоритетов;

υ_i – коэффициент вариации времени обслуживания i-го приоритета;

позволяет сделать следующие выводы:

Наиболее эффективным способом сокращения времени ожидания обслуживания к-го приоритета при фиксированных значениях $T_к$ и $\rho_к$ является уменьшение коэффициента вариации времени обслуживания заявок к-го и высших приоритетов, что достигается постоянством длительности программ их обслуживания.

Из сопоставления графиков зависмости $W_к$ от уровня общей загрузки ЭВМ для пяти уровней приоритетов, полученных в работе /2/ и приведенных на рис. I для дисциплины с абсолютными приоритетами и на рис. 2 для относительных приоритетов, видно, что выигрыш в эффективности для дисциплины с абсолютными приоритетами обслуживания особенно значителен при высокой общей загрузке вычислительной машины ($\rho = 0.8$).

Одним из критериев эффективности выбранной дисциплины обслуживания может служить функционал (3.I). Причем, для получения оценок эффективности удобнее пользоваться не абсолютными значениями величин функционала C, а его относительными изменениями при использовании данной дисциплины диспетчеризации в сравнении с каким-нибудь эталонным алгоритмом диспетчеризации (например, с бесприоритетным обслуживанием). При этом эффективность алгоритма диспетчеризации определяется отношением:

$$B_{от} = \frac{C_о}{C_т} . \qquad (3.6)$$

В работе /5/ показано, что если задачи, выполняемые при мультипрограммной работе ЭВМ, отличаются по относительной важности или времени решения, то в принципе можно получить $B_{от} > 1$ при выборе более сложного алгоритма диспетчеризации.

Оценка относительного выигрыша в эффективности применения данной дисциплины диспетчеризации для конкретного случая использования абсолютных приоритетов в операционной системе СИ была произведена по величине возможного уменьшения быстродействия ЭВМ при введении приоритетной дисциплины по сравнению с бесприоритетным обслуживанием. Иными словами, выигрыш в эффективности в этом случае эквивалентен некоторому повышению производительности уп-

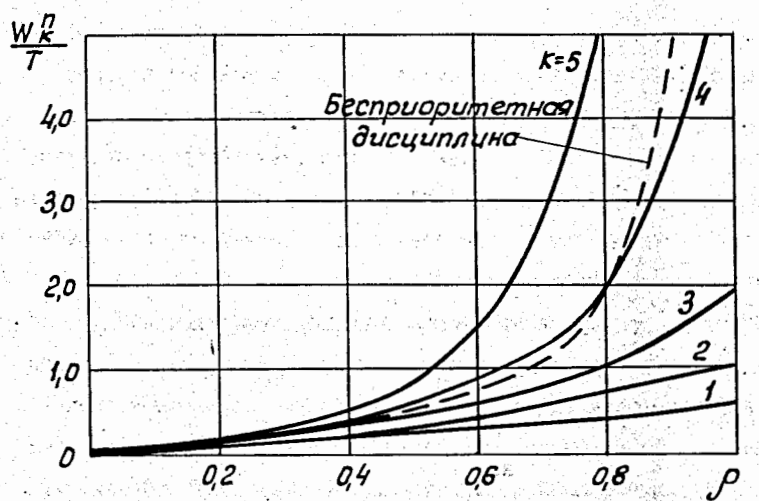

Рис. I Зависимость средней длительности ожидания заявок в очереди от загрузки ЭВМ для системы с пятью относительными приоритетами и постоянным временем ожидания

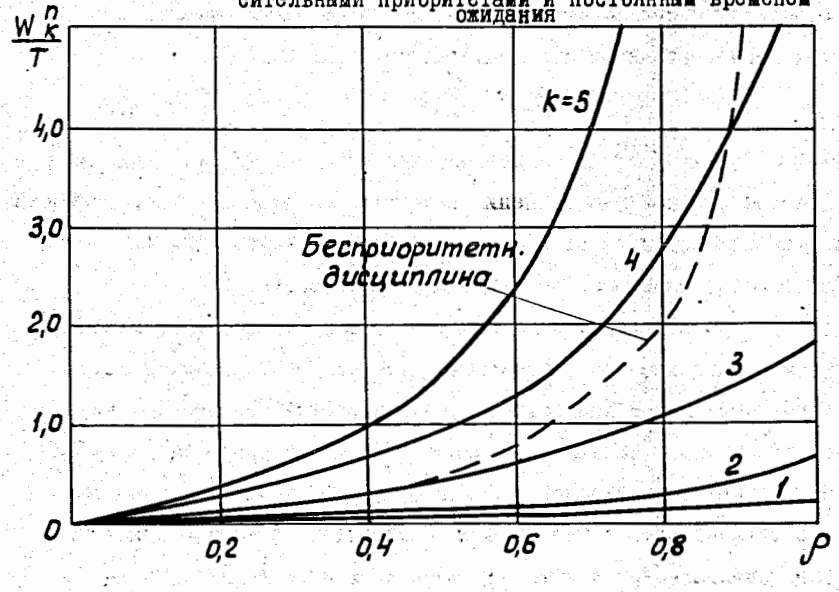

Рис. 2 Зависимость средней длительности ожидания заявок в очереди от загрузки ЭВМ при абсолютных приоритетах и постоянном времени обслуживания:
$n=5$, $\lambda_i=\lambda$, $T_i=T$

равляющей ЭВМ, критерием которого может служить отношение:

$$Э_{or} = \frac{T_{io}}{T_{ir}},$$

где T_{ir} T_{io} – время решения задач, соответствующее выбранному и эталонному алгоритмам диспетчеризации.

Достоинство такого критерия заключается в том, что можно сравнивать эффективность в сопоставлении величин производительности ЭВМ и, кроме, того, полученные результаты сравнения позволяют также оценить затраты на реализацию выбранного типа алгоритма диспетчеризации. В случае принятых допущений о пуассоновском распределении входных потоков в работе /5/ получено правило оптимального назначения абсолютных приоритетов при экспоненциальном распределении времени обслуживания заявок в ЭВМ, согласно которому приоритеты надо присваивать в порядке убывания отношений:

$$\frac{\alpha_k}{T_k} \geq \frac{\alpha_{k+1}}{T_{k+1}} \qquad (3.7)$$

где α_k – важность к-ой заявки;

T_k – время выполнения обслуживающей программы.

При этом достигается минимум линейного функционала (3.I).

Следовательно, высший приоритет должен присваиваться или заявкам с большой относительной важностью α_k или имеющим малое время обслуживания T_k. Обслуживание очень коротких заявок выполняется, как было сказано выше, при помощи вклинивающихся подпрограмм, работающих без вызова диспетчера, а все программы обслуживания заявок при разрешенном прерывании имеют относительно стабильное время выполнения для высших приоритетов, что сокращает среднее время ожидания W_i, т.к. уменьшается коэффициент вариации υ_i.

Поэтому остается один параметр в выражении B_{or}, соответствующий относительной важности данной заявки.

Таким образом, не только для выбора дисциплины диспетчеризации но и для определения её эффективности необходимо оценить характеристики входных потоков заявок. Если интенсивность входных потоков заявок в спиральном измерителе и среднее время обслуживания заявок, т.е. количественные характеристики были определены при помощи специальных часов реального времени достаточно точно (правда, эта методика оценки характеристики потока уже потребовала введения приоритетной дисциплины обслуживания заявок), то оценка относительной важности заявок, как и всякая качественная оценка информации, чрезвычайно затруднительна. Определение относительной важности отдельных заявок обслуживания для СИ было проведено путем экспериментальных оценок частоты и характера сбоев процесса управления при имитации увеличения задержки обслуживания данной заявки. При этом относительная важность заявок, зависящая от характера ошибок, связанных с потерей устойчивости из-за задержки в обслуживании, определялась по следующей шкале штрафов:

– фатальная ошибка, которая влечет за собой полную потерю управляемости и создает аварийную ситуацию. Это, в основном, относится к одновременному управлению движущимися частями при задержке реакции ЭВМ на срабатывание датчиков положения. Например, задержки, приводящие к потере управления механизмом смены проекций, который перемещает платформы с лентопротяжными механизмами общим весом около 120 кг. Такие ошибки могут возникать при нарушении процесса управления движением перископа или режима поиска кадров.

– Ошибки из-за задержки обслуживания заявок от устройств, ведущих внепрограммный обмен информацией, если длительность программ управления этим режимом превышает T_{max} (например, передача данных сканирования для записи на магнитную ленту). В этом случае возможны потери информации, обнаружить которые можно только при обработке уже на большой машине.

– Ошибки, связанные с задержкой в обслуживании заявок по учету реального времени. Это приводит к неверным вычислениям параметров и, следовательно, ухудшает качество регулирования. Например, ошибочное вычисление величины скорости перемещения XY стола в процессе управления его перемещением в следящем режиме резко снижает качество центрирования.

– Штраф за задержку в обслуживании заявок, если задержка приводит только к увеличению времени выполнения алгоритмов управления. Заявки этого типа отличаются большим разнообразием и диспетчеризация их обслуживания существенно влияет на эффективность использования ЭВМ.

Полученные оценки в эффективности для алгоритма диспетчеризации с абсолютными приоритетами по формуле (3.10) показывает, что при величинах загрузки ρ = 0,7-0,9 и значений относительной важности заявок $\beta = \frac{\alpha_1}{\alpha_2}$ = 20 ÷ 50 выигрыш составляет примерно 20-40%, а количество приоритетов достаточно иметь в пределах 2-3.

Следующая оценка, которая касается относительной сложности программной реализации самого диспетчера, была проведена для операционной системы СИ, исходя из увеличения допустимого времени ожидания за счет выполнения собственно программы-диспетчер.

Основным критерием здесь является сравнение времени исполнения той функциональной части диспетчера, работа которой происходит при запрещенном канале прерывания, с допустимым временем задержки следующего включения прерывания. Особенность выполнения функциональных алгоритмов СИ, связанная с необходимостью выполнения коротких вклинивающихся подпрограмм, требует быстрой реакции ЭВМ (это первый класс потоков заявок с приоритетами $\kappa = \overline{1,5}$).

Наиболее жесткие требования на допустимое время ожидания налагаются со стороны заявок по управлению работой режима внепрограммного обмена информацией. Типичным примером, позволяющим получить оценку максимально допустимого времени ожидания заявки на включение прерывания, является работа счетчиков положения измерительного стола в режиме "Увеличение памяти" / 6 /.

Вибрация всей измерительной части, связанная с вращением перископа, вызывают появление сигналов реверса на выходе датчиков линейных перемещений измерительного стола, находящегося в состоянии покоя. В связи с тем, что цена отсчета координат положения этого стола составляет 2 мкм, даже небольшая величина вибрации перископа вызывает довольно высокую плотность поступления заявок на обслуживание сигналов реверса для счетчика положения измерительного стола. Экспериментально была получена величина периода повторения сигналов реверса T_p = 2 мсек. Учитывая то, что этот эффект может наблюдаться для счетчиков положения по обеим координатам X и У и считая, что входные потоки регулярны, т.к. перископ вращается с постоянной угловой скоростью, а загрузка ЭВМ $\rho = 0.8$, получим величину допустимого времени ожидания T_{max} = 400 мксек.

Это время определяет величину максимально допустимого полного цикла по включению прерывания вместе с длительностью выполнения вклинивающейся подпрограммы обслуживания прерывания (класс заявок с приоритетами $\kappa = \overline{1,5}$). При обращении к диспетчеру все нужные для этого операции, включая выход на прерывание, также должны укладываться в это допустимое время ожидания. Для выполнения этих требований диспетчер был разделен на два блока. В первый блок – супервайзер (время работы его не превышает 140 мксек) включена часть диспетчера, выполняемая при запрещенном прерывании, в функции которой входит запоминание приоритета и указателя стартового адреса обслуживающей программы данной заявки, для последующего выполнения её уже при разрешенном прерывании. Приоритет программы и указатель, определяющий её стартовый адрес, являются исходными данными для супервайзера и должны быть определены внутри подпрограммы, обрабатывающей сигнал прерывания от данной заявки.

Одним из основных условий, определяющих необходимость вызова супервайзера, а значит и диспетчера в целом, является время выполнения обслуживания заявки. Если полное обслуживание требует длительных вычислений и не может быть выполнено внутри самой подпрограммы обработки сигнала прерывания, время выполнения которой не должно превышать указанного выше максимально допустимого времени ожидания на включение следующего прерывания, то программа обработки сигнала прерывания и практически исключается, а её функции ограничиваются вызовом супервайзера с выдачей ему исходных данных для последующего выполнения обслуживания заявки, но уже при разрешенном прерывании. В этом случае суммарные временные затраты на выполнение всех перечисленных операций также не должны превышать

$T_{max.}$, а заявка по вызову супервайзера представляет собой одну из 3-х дополнительных заявок, включенных в первый класс с приоритетами $\kappa=1,6$, обслуживание которых производится при помощи вклинивающихся подпрограмм.

Второй блок диспетчера – исполнитель – представляет собой группу функциональных программ диспетчера, выполнение которых может производиться уже при разрешенном прерывании. В задачу программы-исполнитель входит обеспечение выполнения заданной приоритетной дисциплины обслуживания, т.е. дисциплины с абсолютными приоритетами, которая допускает прерывание низшего приоритета при поступлении заявки с более высоким приоритетом, а обслуживание равноприоритетных заявок производится в порядке их поступления. Структура обеих составляющих диспетчера в значительной степени зависит от количества уровней приоритетов в обслуживании. С одной стороны, уменьшение числа уровней приоритетов упрощает программу-диспетчер и убыстряет выполнение супервайзера, но, с другой стороны, выигрыш в величине задержки заявок высокого приоритета при данном быстродействии управляющей ЭВМ может быть получен только за счет увеличения задержек в обслуживании заявок более низкого приоритета. Поэтому увеличение числа приоритетов позволяет осуществлять более гибкое перераспределение ресурсов ЭВМ между заявками и получить меньшее время ожидания для заявок с высоким приоритетом.

Программа-диспетчер, разработанная для операционной системы СИ с учётом анализа конкретных требований на выполнение функционального алгоритма управления, имеет 4 уровня программного приоритета в обслуживании заявок, распределение которых по уровням производилась следующим образом.

– Первый, высший уровень, отдан программе-исполнитель самого диспетчера, в функции которой входит: во-первых, опрос всех уровней и передача управления программе, имеющей в данный момент высший уровень обслуживания. Во-вторых, если в соответствии с принятой дисциплиной обслуживания с абсолютными приоритетами, супервайзер получил заявку на обслуживание, приоритет которой выше, чем приоритет выполняемой в данный момент программы, то программа-исполнитель производит смену последовательности их выполнения, инициируя начало работы программы обслуживания вновь пришедшей заявки, а незаконченная программа обслуживания заявки с более низким приоритетом отсылается обратно в очередь. Возврат к продолжению её обслуживания осуществляется при помощи этой же программы-исполнитель после завершения обслуживания заявки более высокого приоритета.

– Второй уровень программного приоритета не используется при выполнении функциональных алгоритмов, а был введен для работы со специальными часами реального времени, имеющими период следования 0.512 мсек, которые применяются для настройки самой программы-исполнитель и оценки временных характеристик обслуживания заявок третьего и четвертого приоритетов.

– Третий уровень приоритета присвоен программам обслуживания заявок, связанных с управлением работой функциональных блоков СИ в реальном масштабе времени, если время выполнения этих программ превышает T_{max}.

Четвертый уровень включает в себя программы обслуживания заявок, требующих длительных вычислений, а также все программы функциональных алгоритмов, в состав которых включена хотя бы одна

псевдоинструкция символического ассемблера SABR или использовании какой-нибудь программы из библиотеки стандартных программ FORTRAN 8К[7,8].

Последнее требование связано не только с тем, что выполнение псевдоинструкций требует значительного времени, но и с тем, что для этих инструкций необходимо использовать собственную операционную систему Run-Time Linkage Routines ассемблера SABR[8] которая имеет довольно много рабочих ячеек, расположенных в нулевой странице каждого куба памяти. Поэтому, если программа, использующая псевдоинструкции SABR , прерывается заявкой с более высоким приоритетом, в обслуживании которой также применяются псевдоинструкции, а значит и операционная система SABR , то при возвращении прерываемой заявки обратно в очередь потребовалось бы производить запоминание и последующие восстановления большого числа рабочих ячеек и практически свело бы к нулю эффективность диспетчеризации.

ЗАКЛЮЧЕНИЕ

Программа-диспетчер является одной из основных программ монитора операционной системы спирального измерителя и в значительной степени определяет эффективность использования ресурсов управляющей ЭВМ.

Многоуровневая дисциплина обслуживания заявок, реализуемая как программным способом при помощи диспетчера, так и аппаратурно в мультиплексорных каналах [9], обеспечивает устойчивость мультипрограммного режима работы сканирующей системы СИ.

В заключение автор выражает признательность Иванченко И.М. за полезные обсуждения и благодарит своих коллег по работе Буланову Г.Н., Филимонову Т.А., Кутуева Р.Х., А.А.Казакова.

ЛИТЕРАТУРА

1. Котов В.М. и др. Препринт ОИЯИ, 10-7939, Дубна, 1974г.
2. Липаев В.В., Колин К.К., Серебровский Л.А. Математическое обеспечение управляющих ЭВМ. "Советское радио". Москва, 1972 .
3. Котов В.М., Эсенски Й. Сообщение ОИЯИ, 11-7944, Дубна, 1974 .
4. Духовный И.М. "Известия АН СССР", Техническая кибернетика, 1, 1970.
5. Бронштейн О.И., Рыков В.В. "Известия АН СССР", Техническая кибернетика, 6, 1965.
6. Котов В.М., Понятовский М. Сообщение ОИЯИ, 11-7942, Дубна, 1974 .
7. Introduction to Programming. DEC. 1970.
8. Programming Languages. DEC. 1970.
9. Котов В.М., Понятовский М. Сообщение ОИЯИ, 11-7943, Дубна, 1974г.

Рукопись поступила в издательский отдел
13 мая 1974 года.