1.5. Метод структурирования программных ресурсов и дискретно–динамические системы

Среди большого числа понятий, которые возникли и исследуются в вычислительных науках, одним из наиболее общих является понятие дискретной динамической системы (ДС). Особое место среди дискретных динамических систем занимают системы с параллельно функционирующими и взаимодействующими компонентами или ДС параллельного типа. Примерами динамических систем могут служить ЭВМ, их элементы и устройства, различные программы и операционные системы, системы автоматического управления объектами и процессами, производственные системы дискретного характера, такие как промышленные конвейеры и т.д.

Изучение свойств динамических систем и развитие методов их конструирования производится с помощью различных математических моделей и методов в зависимости от класса систем, степени детализации их структуры и свойств, а также от характера исследуемых проблем. В ходе развития науки, техники и производства появляются новые виды ДС, их разнообразие и сложность постоянно растут.

Теория динамических систем, как математического аппарата проектирования систем преобразования информации, к настоящему времени содержит обширную библиографию. Основные понятия и определения, используемые в дальнейшем, взяты с некоторыми отличиями, связанными с рассмотрением процессов в непрерывном времени и выделении в множестве состояний ДС специального элемента, играющего, в зависимости от условий, роль состояния “ожидания” или состояния “завершения”. Это обусловлено необходимостью математического обоснования метода структурирования и связано с решением ряда оптимизационных задач по конвейерной реализации процессов, конкурирующих за использование структурированных программных ресурсов. Постановка этих задач и их решения даны в последующих главах.

Пусть некоторое непустое множество S с выделенным элементом $w \in S$ определяет пространство состояний системы, а T – множество моментов времени, совпадающее с множеством неотрицательных действительных чисел. Тогда под *процессом* будем понимать всюду определенное отображение $p : T \to S$. Будем говорить, что процесс p в момент времени $t \geq 0$ находится в состоянии *“ожидания”*, если $p(t) = w$, и существует момент времени $t' > t$ такой, что $p(t') \neq w$, и в состоянии *“завершения”*, если для любого момента времени $t' > t$ $p(t') = w$. Тем самым мы допускаем прерывание процессов и исключаем из рассмотрения процессы с бесконечным ожиданием.

Через $|p|$ будем обозначать *длительность* процесса *p*, т.е. $|p| = \sup\{t | p(t) \neq w\}$. Процесс *p*, для которого $|p| < \infty$, будем называть *конечным*. С учетом сказанного, через *Proc(S)* и *F(S)* будем обозначать соответственно множество всех процессов в пространстве *S* и множество конечных процессов.

Введем отношение частичного порядка “≤” на множестве *Proc(S)*, полагая $p \leq q$, если $|p| \leq |q|$ и $p(t) = q(t)$ для всех *t* таких, что $0 \leq t \leq |p|$. В этом случае будем говорить, что *p* – *начало* процесса *q*, а *q* – *продолжение* процесса *p*.

Процесс *p* *устойчив* в момент времени *t*, $0 < t < |p|$, если существует окрестность точки *t*, в которой функция *p* постоянна, в противном случае момент *t* будем называть моментом *переключения*. Если процесс *p* в момент времени *t*, $0 < t < |p|$, находится в устойчивом состоянии *s*, то через $d_p(t)$ обозначим *длительность* устойчивого состояния, т.е. если $d_p(t) = d$, то существуют такие ε_1, $\varepsilon_2 > 0$, что $\varepsilon_1 + \varepsilon_2 = d$, и для любого момента времени $t' \in (t - \varepsilon_1, t + \varepsilon_2)$ $p(t') = s$, а для любого $d' > d$ таких ε_1 и ε_2 не существует. Если момент времени t является моментом переключения процесса, то полагаем $d_p(t) = 0$.

В дальнейшем будем рассматривать только *дискретные* процессы, т.е. процессы, у которых всякое конечное начало содержит конечное число моментов переключения, и для которых моменты переключения являются состояниями ожидания w.

Если для процесса p и для моментов времени $t_2 > t_1$, $p(t_1) = s_1$, $p(t_2) = s_2$, и в интервале $[t_1, t_2]$ существует хотя бы один момент переключения, то будем писать $s_1 \xrightarrow{p} s_2$.

Если процесс p в момент времени t, $0 \le t < |p|$, находится в состоянии ожидания w, то это состояние можно продлить на произвольное конечное время $\Delta t > 0$. Тем самым получим процесс p' такой, что

$$p'(t) = \begin{cases} p(t'), & 0 \le t' < t, \\ w, & t \le t' \le t + \Delta t, \\ p(t' - \Delta t), & t + \Delta t < t'. \end{cases}$$

В этом случае будем говорить, что процесс p' является результатом *“конечной задержки”* процесса p в момент ожидания t, а процесс p является результатом *“сжатия”* процесса p' в момент ожидания t на величину Δt и обозначать $p \xrightarrow{t, \Delta t} p'$.

Абстрактной динамической системой будем называть пару $D=(F,S)$, если выполняется условие замкнутости множества допустимых процессов $F: p\in F$ и $q\le p$ $\Rightarrow$ $q\in F$. Абстрактная динамическая система *дискретна*, если каждый ее допустимый процесс дискретен. В дальнейшем будем рассматривать дискретные системы.

ДС, с одной стороны, являются моделями реальных систем, существующих в непрерывном времени и изменяющих свои состояния непрерывно, с другой – являются точными математическими объектами, предназначенными для изучения закономерностей внутри математических теорий. Примерами динамических систем могут служить машины Тьюринга, сети Петри, асинхронные логические сети, программы, записанные на некотором алгоритмическом языке и др.

В рамках данного выше понятия ДС сформулируем ниже основные понятия, связанные с методом структурирования программных ресурсов, прежде всего, понятие конкурирующих процессов и их конвейерной реализации.

Пусть $D=(F,S)$ – динамическая система, $P\subset F$ – конечное непустое множество допустимых процессов системы, $S_0\subset S$ – выделенное множество состояний (множество ресурсов системы), причем состояние ожидания $w \in S_0$. Состояние $s \in S_0$ назовем *общим ресурсом* множества процессов *P*, если существует момент времени $t \in T$ такой, что для любого процесса $p \in P$ $p(t) = s$ и это состояние является устойчивым $(d_p(t) > 0)$.

Множество процессов *P* будем называть множеством *конкурирующих процессов*, если они обладают хотя бы одним общим ресурсом. В этом случае о процессах из *P* будем говорить как о процессах, конкурирующих за использование общих ресурсов. Заметим, что через общие ресурсы происходит один из основных способов взаимодействия процессов в реальных вычислительных системах. Для решения проблемы распределения таких ресурсов, прежде всего *“проблемы взаимного исключения”*, предложены различные механизмы и стратегии (семафоры, обобщенные семафоры, мониторы и др.) Суть этих решений состоит в том, чтобы в каждый момент времени не более одного процесса находилось в так называемом *“критическом интервале”*. К таким ресурсам относятся, в частности, память ЭВМ, каналы ввода–вывода, функциональные устройства, программы и др.

Следует отметить, что не любое состояние дискретной системы является общим ресурсом для некоторого множества процессов.

В нашем случае “критическими интервалами” для каждого процесса p из множества процессов, конкурирующих за использование общего ресурса s, будут являться интервалы времени $(t-\varepsilon'_p, t+\varepsilon''_p)$, в которых каждый из процессов p находится в устойчивом состоянии s.

Нетрудно заметить, что при решении проблемы взаимного исключения с помощью синхронизирующих примитивов общее время прохождения “критических интервалов” конкурирующими процессами составляет величину $\sum_{p \in P} d_p(t)$ (без учета дополнительных затрат на синхронизацию). Однако, для определенного класса общих ресурсов (программных, общей памяти др.) это время можно существенно уменьшить, если воспользоваться методом структурирования программных ресурсов с целью конвейерной реализации множества конкурирующих процессов P. При конвейерной реализации процессов, конкурирующих за использование общего ресурса s, ресурс s, согласно метода структурирования, представляется в виде последовательности состояний s_1, s_2, …, s_k, $k \geq 1$. В этом случае для каждого процесса $p \in P$ критический интервал $(t-\varepsilon'_p, t+\varepsilon''_p)$ разбивается на последовательность из k

интервалов, в каждом из которых процессы проходят состояния s_1, s_2, …, s_k соответственно. Причем суммарная длительность устойчивых состояний s_1, s_2, …, s_k совпадает (с точностью до временных расходов на структурирование) с длительностью $d_p(t)$ устойчивого состояния *s*. Решая последовательно для каждого интервала проблему взаимного исключения получим конвейерную реализацию множества конкурирующих процессов *P*.

Пусть $P \subset F$ – множество процессов дискретной системы (F, S), конкурирующих за использование общего ресурса s в момент времени *t*. Тогда множество процессов $P' \subset F'$ дискретной системы (F', S') будем называть *конвейерной реализацией* множества конкурирующих процессов *P*, если:

1) существует такое множество состояний $\{s_1, s_2, \ldots, s_k\} \subset S'$, $k \geq 1$, $s_i \neq w'$, $i = \overline{1,k}$, что для каждого $p' \in P'$ $s_1 \xrightarrow{p'} s_2 \xrightarrow{p'} \ldots \xrightarrow{p'} s_k$;
2) существует такая реализующая пара (ψ, φ) частичных отображений $\psi : F' \to F$, $\varphi : S' \to S$, что $\psi^{-1}(P) = P'$, $\varphi^{-1}(S) = \{s_1, s_2, \ldots, s_k\}$;

3) для каждого процесса $p' \in P'$ сумма длительностей устойчивых состояний s_i, $i = \overline{1,k}$, совпадает с длительностью устойчивого состояния s процесса $\psi(p')$, т.е. $\sum_{i=1}^{k} d_{p'}(t_i) = d_{\psi(p')}(t)$, и для любого момента времени число процессов $p' \in P'$, для которых $p'(t) = s_i$, $i = \overline{1,k}$, не превосходит 1.

Возможность конвейерной реализации множества конкурирующих процессов зависит лишь от практической возможности представления общего ресурса в виде последовательности составных частей. В частности, конвейерную реализацию множества конкурирующих процессов P можно получить при $k = 1$, используя операцию конечной задержки каждого из процессов $p \in P$ на соответствующее время в моменты времени, предшествующие переходу в состояние s.

Применительно к программным ресурсам конвейерная реализация множества конкурирующих процессов сводится к структурированию программного ресурса на параллельно–выполняемые блоки с использованием специальных языковых средств.

В данной главе изложен метод структурирования программных ресурсов на параллельно-выполняемые блоки, основные понятия которого формализованы с помощью математического аппарата дискретных динамических систем, проведено обоснование эффективности метода структурирования по использованию оперативной памяти, предложен механизм совмещения и взаимодействия конкурирующих процессов, указаны основные реализации механизма управления параллельными процессами.

ВОПРОСЫ ПО ГЛАВЕ:

1. Виды параллелизма. Формула производительности супер компьютера и ее основные параметры (связь с параллельными вычислениями).
2. Виды вычислительных ресурсов (процессоры, каналы обмена, память разных уровней и т.д.).
3. Понятие параллельного процесса и программного ресурса.
4. Концепция структурирования и основные положения метода структурирования программных ресурсов.
5. Операционная, языковая и программная поддержка метода структурирования.
6. Математическое обоснование метода структурирования по использованию оперативной памяти.
7. Особенности разработки параллельных программ.
8. Понятие дискретной динамической системы в параллельном программировании.
9. Параллельные конкурирующие процессы и проблемы синхронизации (взаимное исключение, критические интервалы, семафоры, обобщенные семафоры и д.р.).
10. Конвейерная реализация множества конкурирующих процессов.