
ГЛАВА 2

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ СОСРЕДОТОЧЕННОЙ ОБРАБОТКИ КОНКУРИРУЮЩИХ ПРОЦЕССОВ

2.1. Сосредоточенные конкурирующие процессы и режимы их взаимодействия

В главе 1 при изложении основных положений метода структурирования программных ресурсов введено понятие сосредоточенного и распределенного конкурирующего процесса. Отметим, что среди различных форм параллельной обработки одна из центральных ролей отводится такому виду параллелизма, как *сосредоточенная обработка* конкурирующих процессов в многопроцессорных системах и комплексах, *т.е. когда все блоки структурированного программного ресурса выполняются на одном и том же процессоре*. Особенно это касается тех сфер и приложений,

где требуется управлять многими процессами одновременно, например, в системах оперативного мониторинга, а также в системах, где осуществляется одновременная обработка информации сразу от многих датчиков.

Все это порождает сложные в математическом отношении задачи параллельного программирования, связанные, прежде всего, с получением соотношений для вычисления точных значений минимального общего времени при реализации заданных объемов вычислений для различных видов МС и сетевых комплексов. Определенный прогресс в решении таких задач может быть достигнут за счет применения методов дискретной оптимизации, в том числе и теории расписаний в многостадийных системах.

Один из механизмов существенного сокращения общего времени выполнения в МС множества конкурирующих процессов или заданных объемов вычислений базируется на методе структурирования программного ресурса на параллельно выполняемые блоки, который подробно изложен в главе 1.

Пусть p , $p \geq 2$ – число процессоров многопроцессорной системы, n , $n \geq 2$ – число конкурирующих процессов, s , $s \geq 2$ – число блоков структурированного программного ресурса. Причем из физических соображений на множестве блоков

установлен линейный порядок их выполнения $1, 2, \dots, s$. Предполагается, что все n процессов используют одну и ту же копию структурированного на блоки программного ресурса.

Предполагается также, что взаимодействие процессов, процессоров и блоков подчинено следующим условиям:

- 1) ни один из блоков программного ресурса не может обрабатываться одновременно более чем одним процессором;
- 2) ни один из процессоров не может обрабатывать одновременно более одного блока;
- 3) обработка каждого блока программного ресурса осуществляется без прерываний;
- 4) порядок предоставления процессам процессоров циклический, т.е. процесс с номером $i = lp + q$, $i = \overline{1, n}$, $l \geq 0$, $q = \overline{1, p}$, сосредоточен на процессоре с номером q .

Наряду с условиями 1–4 введем дополнительные условия, которые определяют режимы взаимодействия процессов, процессоров и блоков:

- 5) начало выполнения очередного блока определяется наличием процессора и готовностью блока к

выполнению (блок готов к выполнению, если он не выполняется ни на одном из процессоров);

6) для каждого из процессоров момент окончания обработки блока совпадает с моментом начала обработки следующего;

7) для каждого из блоков момент окончания его обработки процессором совпадает с моментом начала обработки следующим процессором.

Понятно, что условие 6 запрещает простои процессоров при обработке блоков структурированного программного ресурса внутри каждого процесса. Условие 7 означает, что каждый блок с момента начала его обработки первым процессором на остальных $(p-1)$ -м процессорах обрабатывается непрерывно.

В целом условия 1–7 близки к изучаемым в теории расписаний технологическим ограничениям выполнения операций в многостадийных системах.

Условия 1–5 определяют так называемый *асинхронный режим* взаимодействия процессоров, процессов и блоков, который предполагает отсутствие простоев процессоров при условии готовности блоков, а также “пролеживания” блоков при наличии процессоров.

Первый синхронный режим, определяемый условиями 1–4, 6, обеспечивает непрерывное выполнение каждого процесса с момента старта первого блока и до завершения последнего.

Второй синхронный режим, определяемый условиями 1–4, 7, обеспечивает непрерывное выполнение каждого блока с момента его старта всеми процессами.

Введем следующие определения.

Определение 2.1. Система n конкурирующих процессов называется *однородной*, если времена выполнения блоков программного ресурса в порядке 1, 2, ..., s не зависят от объемов и структуры обрабатываемых данных, т.е. одинаковые для каждого из процессов.

Определение 2.2. Если времена выполнения одних и тех же блоков программного ресурса зависят от объемов обрабатываемых данных или их структуры, т.е. разные для разных процессов, то система из n процессов называется *неоднородной*.

Пусть имеется система из n , $n \geq 2$, неоднородных конкурирующих процессов. Обозначим через $T^c = [t_{ij}]$, $i = \overline{1, n}$, $j = \overline{1, s}$, $n \times s$ –матрицу времен выполнения блоков программного ресурса Q_1, Q_2, \dots, Q_s каждым из

конкурирующих процессов, где t_{ij} – время выполнения i -м процессом j -го блока, $i = \overline{1, n}$, $j = \overline{1, s}$. Заметим, что для системы однородных процессов все строки в матрице T^c будут одинаковые.

Интерес представляют задачи параллельного программирования, связанные с получением математических соотношений для вычисления точных значений минимального общего времени при реализации заданных объемов вычислений в различных режимах взаимодействия процессов, процессоров и блоков программного ресурса.