

1.4. Математическое обоснование метода структурирования по использованию оперативной памяти

Предлагаемый метод организации взаимодействия конкурирующих процессов с программным ресурсом, основанный на его структурировании на параллельно выполняемые блоки, позволяет использовать один и тот же программный ресурс для множества конкурирующих процессов одновременно. При этом автоматически решается проблема предотвращения тупиковых ситуаций и получается существенный выигрыш как по использованию оперативной памяти, так и во времени реализации параллельных процессов. Выигрыш от использования оперативной памяти получается в результате того, что отпадает необходимость в копиях одной и той же программы. Для каждого процесса, использующего структурированную программу, нужно отвести только небольшой дополнительный участок памяти для хранения промежуточных данных и состояний.

Пусть L – объем оперативной памяти, который занимает структурированная программа, при этом она используется одновременно n , $n \geq 2$, процессами. Тогда объем оперативной памяти, который необходимо отвести для структурированной программы с учетом дополнительных участков оперативной

памяти l_i , для каждого из i -х процессов будет составлять величину $L + \sum_{i=1}^n l_i$. Если же программа не обладает свойствами структурированных программных ресурсов, то при одновременной реализации n , $n \geq 2$, процессов, использующих PR , необходимо из вычислительной среды запросить дополнительный объем памяти, который составит величину nL , что намного больше $L + \sum_{i=1}^n l_i$, т. е.

$$nL \gg L + \sum_{i=1}^n l_i, \text{ так как } L > l_i.$$

Заметим, что в тех случаях, когда $s < n$ величину $\sum_{i=1}^n l_i$ также можно уменьшить до величины $\sum_{i=1}^s l_i$ за счет динамического использования участков памяти l_i , $i = \overline{1, s}$.

Организация параллельных вычислений по методу структурирования позволяет также получать существенный выигрыш по времени реализации заданных объемов вычислений, так как:

во-первых, исключается этап повторной загрузки программных ресурсов с внешних носителей, требующий значительного времени;

во-вторых, в результате распараллеливания программного ресурса на блоки ввода–вывода и счета появляется возможность совмещенного выполнения этих блоков даже в мультипрограммных системах;

в-третьих, эффективно решается проблема создания параллельно–используемых (реентерабельных) программ большого объема.

Но, самый главный выигрыш состоит в том, что появляется возможность практически неограниченного распараллеливания алгоритмов и программ при их отображении на архитектуру МС.

Проблема обоснования метода структурирования программных ресурсов на параллельно выполняемые блоки по времени реализации заданных объемов вычислений носит комплексный характер, что порождает ряд сложных в математическом отношении проблем и задач параллельного программирования. Для их решения требуется прежде всего построение математических моделей, адекватно отражающих различные аспекты взаимодействия множества процессов, с учетом их физической специфики, а также архитектурных особенностей многопроцессорных систем, сетей и комплексов, вычислительных ресурсов, дополнительных системных затрат и т. д. Анализ показывает, что на пути

решения этой комплексной проблемы возникают математические задачи дискретно–комбинаторного характера. Поэтому в последующих главах авторами при построении и исследовании математических моделей и задач оптимальной организации параллельных процессов широко применяются методы и аппарат теории графов, сетевых графов и линейных диаграмм Ганта, теории расписаний, комбинаторной оптимизации, упорядочения, алгебры логики, теории множеств, алгебры матриц и др.