1.3. Основные положения метода структурирования

При создании многопроцессорных систем и вычислительных комплексов различной архитектуры, включая их системное и программное обеспечение, важное место занимают задачи управления множеством вычислительных процессов, использующих одни и те же программные ресурсы. От того, как решаются эти задачи, зависит как эффективность использования вычислительных мощностей МС, так и возможность решения в реальное время сложных задач из различных областей знаний.

Основная идея метода структурирования состоит в обеспечении специального способа структурирования программного ресурса на блоки и организации параллельного использования этих блоков множеством конкурирующих процессов. Достигается это с помощью специальных операционных и языковых средств параллельного программирования.

Пусть *PR* – программный ресурс, *n,* $n \geq 2$, число конкурирующих процессов. Требуется организовать вычислительный процесс таким образом, чтобы общее время выполнения *n* процессов, использующих *PR*, было минимальным.

Одной из стратегий решения данной задачи с p, $p \geq 2$, процессорами является предоставление каждому процессу отдельной копии *PR*. Но этот путь не всегда осуществим из–за ограниченного объема ресурсов вычислительной системы и тем более трудно достижим в случае больших программ, используемых в качестве программных ресурсов. Поэтому при решении данной задачи применяется стратегия последовательного обслуживания n процессов с использованием различных механизмов их синхронизации (семафоры, мониторы, аппарат событий и т. п.). В этом случае суммарное время выполнения процессов составит величину $T_{sum} = nT$, где T – время выполнения каждым из процессов программного ресурса. Дополнительные временные затраты на синхронизацию процессов здесь не учитываются.

Время T_{sum} можно существенно сократить, если обеспечить структурирование программного ресурса на блоки Q_1, Q_2, ..., Q_s с последующей конвейеризацией как блоков по процессам, так и процессов по процессорам многопроцессорной вычислительной системы. Для этого необходимо выполнить следующие основные этапы:

- структурирование (декомпозиция) программного ресурса на блоки Q_1, Q_2, ..., Q_s;
- организация одновременного (параллельного) взаимодействия процессов, процессоров и блоков структурированного программного ресурса;
- совмещение во времени выполнения различных процессов,
- запоминание после завершения использования очередного блока и восстановление перед началом выполнения следующего блока промежуточных состояний процессов;
- запуск процессов на выполнение и их завершение;
- выбор способов (режимов) взаимодействия процессов, процессоров и блоков (асинхронный режим, синхронные режимы и т.д.);
- наличие специальных языковых средств описания взаимодействия процессов, процессоров и блоков программного ресурса, а также разработка алгоритмов реализации такого взаимодействия;
- обеспечение операционной поддержки взаимодействия процессов, процессоров и блоков.

Структурирование программного ресурса на блоки осуществляется, как правило, либо исходя из физического

смысла задачи на этапах создания математической модели и алгоритмов её решения, либо путём анализа готовой, последовательной программы с целью её декомпозиции. Далее каждый блок оформляется с помощью специальных операторов языка представления параллельных программ. Число блоков, на которое осуществляется структурирование (декомпозиция) программного ресурса, зависит от количества процессов и процессоров, длительности выполнения программного ресурса, накладных расходов и других параметров. В последующих главах будут доказаны соответствующие критерии эффективности и оптимальности структурирования по числу блоков, процессов, процессоров и другим параметрам с учетом накладных расходов.

Один из возможных способов (механизмов) взаимодействия процессов, процессоров и блоков следующий. Блоки, процессы и процессоры вычислительного комплекса нумеруются в порядке 1, 2, …, s, 1, 2, …, n и 1, 2, …, p соответственно. Причем на множестве блоков установлен линейный порядок их выполнения. Предполагается, что все n процессов используют одну и ту же копию структурированного программного ресурса. В дальнейшем под процессом будем понимать выполнение всех

блоков программного ресурса в порядке 1, 2, …, s. При этом процесс называется *сосредоточенным*, если все блоки программного ресурса выполняются на одном и том же процессоре, и *распределённым*, если все блоки или часть из них выполняются на разных процессорах.

Операционная система или специально выделенный организующий процесс предоставляет блоки структурированного программного ресурса Q_1, Q_2, ..., Q_s каждому из процессов в порядке 1, 2, …, n. При этом в случае сосредоточенной обработки возможна монополизация процессора *i*–м процессом. Если блок Q_j, $j=\overline{1,s}$, освобождается очередным *i*–м процессом, то он предоставляется *(i+1)*–му процессу, а сам *i*–й процесс получает в своё распоряжение *(j+1)*–й блок либо переводится в состояние ожидания до освобождения *(j+1)*–го блока, $i=\overline{1,n-1}$, $j=\overline{1,s-1}$ и т. д. В случае распределённой обработки монополизация процессоров процессами не происходит, а блоки одного и того же процесса выполняются на разных процессорах.

Очевидно, что при наличии в сетевой многопроцессорной системе p процессоров возможно совмещённое во времени выполнение процессов.

Запоминание и восстановление промежуточных состояний процессов, запуск процессов на выполнение и их завершение, выбор режимов взаимодействия процессов, процессоров и блоков осуществляет специальная подсистема операционной системы или организующий процесс.

Следует отметить, что при организации вычислений по методу структурирования в качестве блоков структурированного программного ресурса могут служить наборы программ, циклические участки программ, потоки заданий на обработку запросов пользователей в мультипрограммных системах, отдельные микрооперации, выполнение которых подразделяется на несколько фаз и др.