

УДК 681.324

Е. Рудина (3 курс, каф. ИБКС), М.В. Хлудова, к.т.н., доц.

СЕМЕЙСТВО ПРОТОКОЛОВ СИНХРОНИЗАЦИИ ПРОЦЕССОВ РЕАЛЬНОГО ВРЕМЕНИ

При планировании заданий в режиме реального времени мало выбрать подходящую стратегию диспетчеризации, кроме этого приходится решать ряд проблем, возникающих в связи с использованием этой стратегии в конкретных условиях. При применении наиболее естественных для планирования заданий реального времени стратегий с прерываниями обслуживания по приоритетности в условиях, когда логически независимые процессы в системе связаны физически (через использование общих информационных или физических ресурсов) возникают ситуации, когда работа планировщика приводит к иным, нежели ожидалось, результатам. К таким ситуациям относятся инверсия приоритетов процессов, тупиковые ситуации и отчасти цепочки блокировок.

Здесь под приоритетом процесса понимаем некоторый обобщенный признак процесса, отличающий его по важности от остальных. Если вновь поступивший в систему процесс более высокоприоритетный, чем выполняющийся, то согласно стратегии он должен прервать выполняющийся, иначе – ждать, пока его приоритет не станет самым высоким из приоритетов находящихся в системе процессов. Таким образом, предсказуемы рамки выполнения, что и требуется. Однако может возникнуть непредвиденный конфликт стратегии планирования с примитивами синхронизации доступа к разделяемым информационным или физическим ресурсам.

Инверсия приоритетов – это такое явление, когда высокоприоритетный процесс вынужден ждать завершения низкоприоритетных операций с разделяемым ресурсом. Время ожидания становится непредсказуемым, если в системе находится ряд процессов среднего приоритета. Тупиковые ситуации возникают, когда запросы на ресурсы среди выполняющихся процессов замыкаются в кольцо, т.е. каждому из процессов требуется ресурс, занятый другим. В этой ситуации ни один из процессов не может быть завершен. Цепочки блокировок сродни инверсии приоритетов, но в этой ситуации акцент делается на частые прерывания высокоприоритетного процесса при запрашивании очередного ресурса.

Очевидно, описанные проблемы серьезно влияют на выполнение заданий в режиме реального времени. В подобных случаях приходится применять более сложные алгоритмы планирования заданий, нежели стратегии с прерыванием обслуживания по какому бы то ни было признаку (задаваемый пользователем приоритет, директивный срок выполнения и т.п.). Один из подходов заключается в том, чтобы сделать критические области процессов (области использования разделяемых ресурсов) непрерываемыми. Очевидно, что в этом случае результат работы может быть удовлетворительным, только если критические области небольшие по времени.

Другим подходом к синхронизации заданий в реальном времени является применение так называемых протоколов наследования приоритетов. Основная идея всех протоколов этого семейства заключается в использовании текущим процессом во время его нахождения в критической области наивысшего приоритета из приоритетов всех процессов, заблокированных текущим. Это позволяет сделать текущий процесс непрерываемым для процессов среднего приоритета на время его нахождения в критической области. Механизм наследования приоритета лег в основу базового протокола наследования приоритета. Легко показать, что он избавляет от инверсии приоритетов, но не спасает от формирования тупиков и цепочек блокировок. Второй из рассматриваемых протоколов семейства – так называемый протокол

наивысшего приоритета («потолка» приоритета). Помимо простого наследования приоритета предусмотрено запрещение вхождения процесса в его критическую область, если это таит в себе потенциальную опасность возникновения тупика. Это делается с помощью некоторого «потолочного» значения приоритета, которое относится к разделяемым ресурсам и говорит о том, каков наивысший приоритет процесса, могущего использовать данный ресурс. Данный алгоритм несколько избыточен: довольно часто его использование приводит к возникновению ненужных блокировок.

Практическое исследование представляемых алгоритмов синхронизации осуществлялось с помощью модели ситуации в реальной системе планирования заданий. Модель представляет собой набор числовых значений-параметров выполнения заданий в системе: приоритет выполнения, временные рамки, число и номера используемых разделяемых однородных ресурсов, временные рамки их использования. Такая абстрактная модель является достаточной для получения качественных характеристик алгоритмов. Полученная ситуация обрабатывается специально разработанной программой-анализатором одного из трех механизмов синхронизации: простого планирования на основе стратегии с прерыванием обслуживания, базового протокола наследования приоритета и протокола наивысшего приоритета. Исследования, проведенные с помощью данной программы, дали следующие результаты:

1. Для обыкновенной стратегии с прерыванием обслуживания зависимость времени прерывания процесса остальными от приоритета процесса следующая: в связи с инверсией приоритетов, как локальной, так и на длительные промежутки времени, время прерывания процессов среднего и высокого приоритетов в типичном случае стремится сравняться. В худшем случае время прерывания для средне- и даже низкоприоритетных процессов меньше, чем для высокоприоритетных.

2. Применение базового протокола наследования приоритета разносит по величине значения времени прерывания для высоко- и низкоприоритетных заданий. Тем не менее в области высокоприоритетных заданий, в особенности требующих большого количества ресурсов, ситуация все еще трудно предсказуема из-за возможного формирования цепочек блокировок. Причем это относится даже к процессам самого высокого приоритета, а часто бывает необходимо, чтобы такие процессы были «аварийно» срочными. Тем не менее, характеристики этого алгоритма лучше характеристик предыдущего.

3. Для алгоритма наивысшего приоритета: полностью исключается формирование тупиков и цепочек блокировок, а также ситуации с инверсией приоритетов в связи с чем для наиболее высокоприоритетных процессов время прерывания минимизируется. Но в то же время из-за алгоритмической «перестраховки» образуются ненужные блокировки среднеприоритетных процессов, в связи с чем качественные характеристики их выполнения (относительное к длительности выполнения время прерывания) стремятся сравняться с характеристиками низкоприоритетных, что является отрицательной стороной алгоритма. Однако протокол на его основе хорош как для предотвращения тупиков-зацикливаний, так и для захвата системы экстренным заданием.

При разработке планировщиков заданий систем реального времени необходимо учитывать реальные условия работы этих систем для выбора подходящего алгоритма синхронизации, так как возможна и обратная ошибка: выбор избыточно сложного или неподходящего по своим характеристикам подхода.