

ВЫБОР ОПТИМАЛЬНОГО АЛГОРИТМА ПЛАНИРОВАНИЯ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

О.Г. Самохвалова

Настоящая работа посвящена вопросам из области планирования процессов в системах реального времени. Задачей исследований являлось: создание методики выбора приемлемого алгоритма планирования на основе анализа сведений об алгоритмах, предполагаемой модели задач и структурных характеристиках будущей системы, формирование теоретической базы для дальнейшего создания программных инструментов проектирования планировщиков систем реального времени. Статья кратко излагает результаты исследований и разработок автора по данному направлению. Предлагаемая в статье методика адресована разработчикам программного обеспечения управляющих систем.

Управление процессом предоставления ресурсов системы задачам, нитям, процедурам обработки прерываний и т.д. является одной из основных функций любой операционной системы и осуществляется при помощи такого механизма, как планирование. Данный механизм обеспечивает системе возможность параллельного выполнения нескольких задач. В системах реального времени планирование должно также гарантировать предсказуемое поведение, безопасность, возможность длительной, безотказной работы, выполнение задач к поставленному сроку. От метода планирования во многом зависит успешная работа системы в целом [1, 2].

Создание программного обеспечения управляющих систем может производиться различными способами, в том числе и с использованием "готовых" операционных систем. Однако, как правило, универсальные ОС РВ поддерживают лишь ограниченное число алгоритмов планирования (2-3), которые являются очень простыми и имеют ограниченные возможности. При построении заказных ИУС, специализированных ОС и т.п. коллектив разработчиков часто вынужден создавать свои планировщики [3–5].

При создании планировщика основной проблемой, которую необходимо решить разработчикам, является нахождение оптимального алгоритма планирования для данной системы, в соответствии с которым вычислительные ресурсы предоставляются задачам. (С точки зрения автора, алгоритм считается оптимальным, если его параметры соответствуют требованиям разрабатываемой системы, а характеристики, связанные с его практической реализацией и эффективностью функционирования, являются наилучшими для данного случая.)

Часто решение этого вопроса сопряжено с различными трудностями. Это ограниченная материальная и информационная база, сжатые сроки разработки, невозможность проведения предварительной НИР. Наличие методик, на базе которых возможен указанный выбор, существенно облегчило бы труд разработчиков систем реального времени.

Задачами проведенных исследований являлось:

- создание методики выбора приемлемого алгоритма планирования на основе анализа сведений об а) алгоритмах; б) предполагаемой модели задач; в) структурных характеристиках будущей системы;
- формирование теоретической базы для создания программных инструментов проектирования планировщиков систем реального времени.

Для решения поставленных задач было проведено изучение различных вопросов в области планирования в системах реального времени – политик планирования, их модификации, влияния аппаратных средств на механизмы планирования и т.д. Сформирован ряд критериев для сравнительного анализа стратегий планирования. Некоторые из них, например, коэффициент эффективности планировщика, предложены для оценки алгоритмов впервые. Также была разработана, приведена к

полуформальному виду и оформлена в виде методики, кратко рассмотренной ниже, последовательность действий выбора оптимального алгоритма.

Согласно Дж. Фоксу [6], процесс разработки программного обеспечения систем, в том числе и планировщиков, может быть разбит на шесть этапов: определение требований, проектирование, написание команд, компоновка, тестирование и документирование.

Разработанная методика охватывает первые два этапа, а дальнейшее ее усовершенствование позволит включить в этот список и стадию написания команд. Рис. 1 и 2 показывают, как изменяется процесс разработки ПО (планировщиков) с использованием «автоматизированного разработчика планировщиков». Схема методики отражена на рис. 3.

Первым шагом при создании планировщика, согласно схеме, является формирование на основе технического задания или иного описания системы сведений о структурных характеристиках элементов системы, задачах и окружающей среде. Эти данные являются исходными для проводимого далее анализа. Методика подробно описывает, каким образом можно получить интересующие нас данные и привести к виду, пригодному для дальнейшего использования.

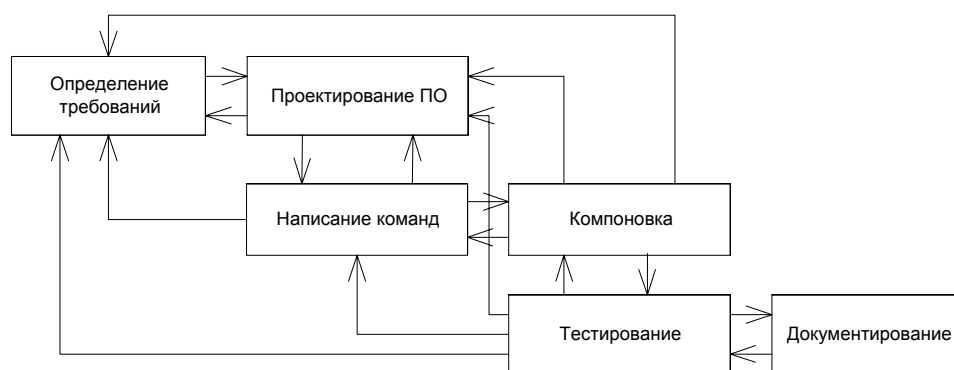


Рис. 1. Традиционный процесс разработки ПО (планировщиков)

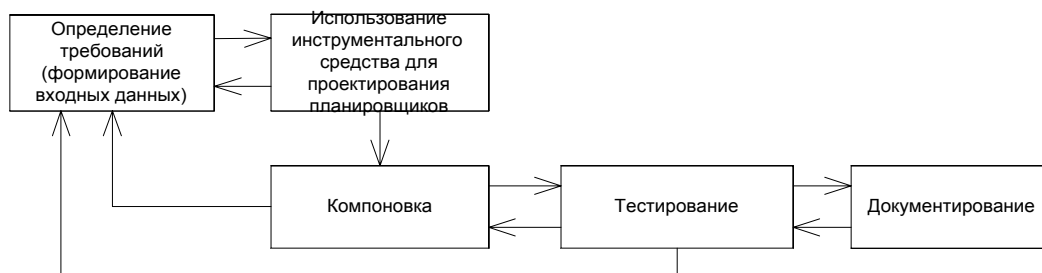


Рис. 2. Процесс разработки с использованием инструментального средства

Вторая часть методики охватывает этап проектирования, который включает в себя, как минимум, четыре стадии – проведение НИР, анализ и принятие решения, структуризацию и описание.

Использование методики позволяет опустить некоторые из названных пунктов или значительно формализовать и упростить их. Так, сведения об алгоритмах планирования в качестве входных параметров и их блок-схемы снимают необходимость в проведении предварительной исследовательской работы и описании. Сравнительные таблицы и методы выбора наилучшего решения сводят работу по

анализу, принятию решений и структуризации к набору указанных, формальных действий, снижая время на проектирование.

Часто в результате некорректных решений, возникновения тупиковых ситуаций и т.д. приходится возвращаться к более ранним стадиям разработки. Этот факт нашел свое отражение в методике в виде этапа коррекции исходных данных, где указаны пути внесения изменений во входные параметры, в том числе и без глобальных изменений в структуре и функциональности системы.

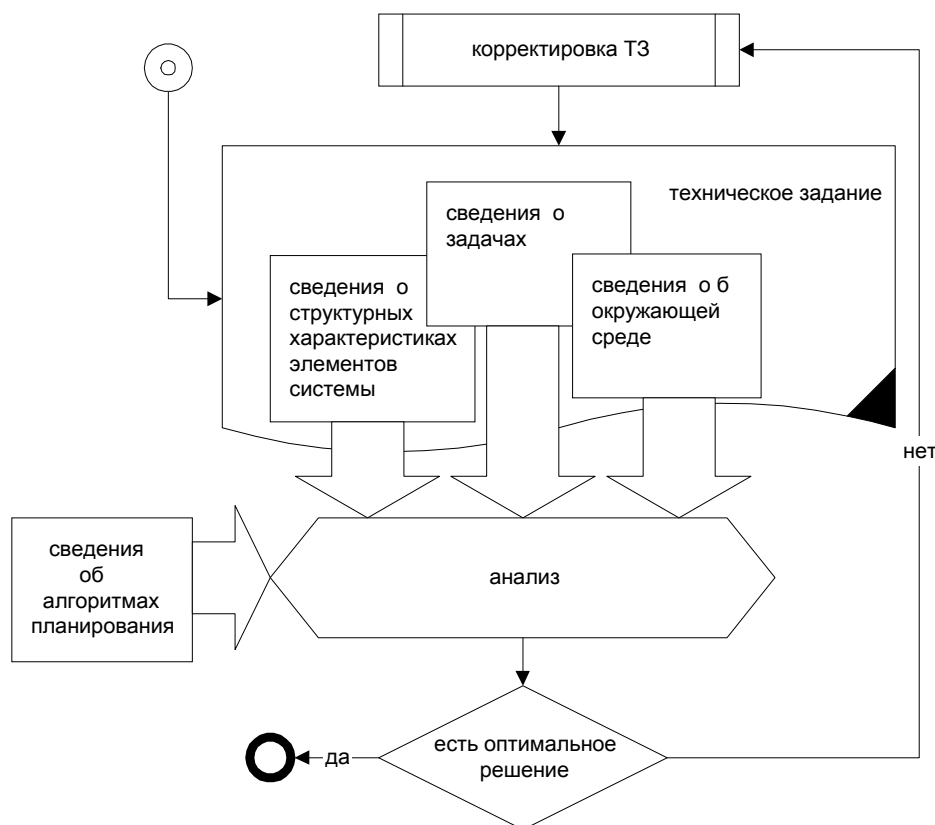


Рис. 3. Схема методики

Анализ данных и выбор алгоритма планирования разбит на 4 этапа. В первую очередь происходит проверка совместимости параметров системы. Она необходима, поскольку может быть выявлено, что при имеющихся структурных характеристиках системы невозможно поддерживать и эффективно обслуживать заданный набор задач. Для этого необходимо сравнить данные о системе с таблицей зависимостей, где указаны возможное быстродействие, прерываемость заданий и их количество в зависимости от типа процессора. После прохождения первой ступени мы можем перейти к следующей – проверке соответствия самих алгоритмов планирования входным параметрам системы. Стратегии, по каким-либо причинам не удовлетворяющие условиям разрабатываемой системы, отсеиваются на данном этапе.

После этого мы получаем некоторое множество алгоритмов, которое необходимо сравнить между собой по указанным характеристикам, что и производится на третьей стадии.

Характеристики, по которым производится сравнение, также сведены в таблицы. Анализируя их, мы отбираем алгоритм с наилучшими характеристиками по всем позициям. Если такого не существует, отсеиваем наихудшие по всем пунктам и переходим к четвертому этапу.

Здесь, применяя методы принятия оптимального решения – ранжирование критериев выбора по важности и присвоение относительных весовых коэффициентов, мы отбираем наилучший алгоритм среди дошедших до данной ступени. Сведения о предпочтительности того или иного фактора можно выявить из технического задания или иного описания системы.

В результате мы должны получить единственную стратегию планирования, которая будет максимально соответствовать заданным условиям. Поскольку блок-схемы политик планирования приведены в методике, после выбора алгоритма можно приступить непосредственно к написанию программного кода.

Разработанная методика предоставляет информацию по планированию разработчикам систем реального времени, значительно сокращает сроки на разработку планировщиков, проведение НИР, снижает стоимость системы за счет. Она также является методической базой для дальнейшего создания программных инструментов для проектирования планировщиков систем реального времени.

Методика адресована разработчикам программного обеспечения управляющих систем.

Литература

1. C.W. Mercer. An Introduction to Real –Time Operation Systems: Scheduling Theory // Technical Report, Carnegie Mellon University (USA), 1992.
2. E.A. Lee. Embedded software // To appear in Advances in Computers, vol.56, Academic Press, London, 2002.
3. S.A. Aldarmi, A. Burns. Time-Cognizant Value Function for Scheduling Real -Time System // Technical Report, University of York (England), 1998.
4. Bestavros. Scheduling. Boston University (USA), 1995.
5. M. Gonzales Harbour, L. Sha. An Application-Level Implementation of the Sporadic Server // Technical Report, Carnegie Mellon University (USA), 1991.
6. Фокс Дж. Программное обеспечение и его разработка. М.: Мир, 1985.