**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра «Информационные системы»**

**Реферат № 6**

**по дисциплине «Системы реального времени»**

**Тема: Выполнимость набора задач**

Дата представления реферата на проверку: 29.05.2023

|  |  |  |
| --- | --- | --- |
| Студентгр. 8362 |  | Матвеев Н.Д.  nicetas.matthias@yandex.ru |
| Преподаватель |  | Сидельников В.В. |

Санкт-Петербург

2023

содержание

[1. Понятие выполнимости набора задач. 3](#__RefHeading___Toc134461661)

[2. Rate Monotonic (RM) анализ выполнимости 5](#__RefHeading___Toc134461662)

[3. Deadline Monotonic (DM) анализ выполнимости. 9](#__RefHeading___Toc134461663)

[Список использованных источников 10](#__RefHeading___Toc134461664)

# Понятие выполнимости набора задач.

Задача реального времени характеризуется следующими временными параметрами:

* Момент освобождения (releasetime, момент поступления − arrivaltime, момент запроса - requesttime) ri − момент, в который задача становится готова к исполнению;
* Время выполнения (время исполнения, время вычисления, computation time) Ci - время, необходимое процессору для выполнения задачи без прерываний;
* Абсолютный дедлайн (абсолютный критический срок, absolute deadline) di − момент, перед которым задача должна быть завершена во избежание некорректного поведения системы;
* Относительный дедлайн (relative deadline) Di− разность между абсолютным дедлайном и моментом запроса:

Также важной временной характеристикой задачи реального времени является время отклика (response time) Ri − разность между моментом завершения и моментом запроса: Ri= fi - ri , где fi (момент завершения, finishing time) – момент, в который задача заканчивает свое исполнение.

Переход от выполнения одной задачи к другой осуществляется в результате планирования и диспетчеризации. Работа по определению того, в какой момент необходимо прервать выполнение текущей активной задачи и какой задаче предоставить возможность выполняться, называется планированием. Набор правил и критериев, которые определяют порядок, в котором задачи выполняются на процессоре, называется алгоритмом (политикой) планирования.

С точки зрения теории планирования реального времени планирование называется выполнимым (feasible), если все задачи могут быть завершены в соответствии с набором указанных ограничений. Набор задач называют планируемым (schedulable), если существует хотя бы один алгоритм, который может производить выполнимое планирование .

Алгоритм называется оптимальным (optimal), если он минимизирует некоторую заданную функцию затрат (costfunction), определенную над набором задач. Примерами функций затрат могут быть среднее время отклика, общее время выполнения, взвешенная сумма времени выполнения и максимальное временное смещение. Классические алгоритмы планирования стараются свести к минимуму данные функции затрат. С помощью функций затрат зачастую определяется и производительность алгоритмов планирования.

Когда функция затрат не определена и единственной проблемой является достижение допустимого расписания, то алгоритм называется оптимальным, если он сможет найти выполнимое расписание, если такое существует.

В наши дни наблюдается возрастающий интерес к области планирования прикладных задач в СРВ на многоядерных платформах и практической реализации полученных методов и алгоритмов, позволяющих соблюдать гарантии реального времени. Под планированием (scheduling) понимается переход от выполнения одной задачи к другой. В научно-исследовательских работах, посвященных СРВ, большое внимание уделяется алгоритмам планирования реального времени – набору правил, согласно которому производится переключение задач (объектов планирования) и выделение им требуемых ресурсов. Данные алгоритмы должны предоставлять априорные доказательства того, что все задачи будут выполнять свои временные требования и ограничения, т.е. успевать завершать свое выполнение относительно критических сроков (дедлайнов, deadline), при условии, что все задачи удовлетворяют своим временным характеристикам и параметрам. Такие гарантии должны быть аналитически доказаны до начала фактического функционирования системы реального времени (в режиме офлайн). Это достигается путем проведения анализа планируемости. Данные алгоритмы базируются на модели задач реального времени, которая содержит важную информацию о временных характеристиках и параметрах задач в СРВ, модели дедлайнов задач и модели аппаратной платформы.

Можно выделить два типа алгоритмов планирования: политики планирования (scheduling policies, scheduler, планировщик) и тесты планируемости (schedulability tests). Политики планирования контролируют расписание задач. Они работают в системе одновременно с задачами на этапе выполнения и в режиме онлайн принимают решения по планированию, основываясь на характеристиках и ограничениях задач. Тесты планируемости до запуска системы производят проверку соблюдения гарантий выполнения задач к заданным дедлайнам

С точки зрения **теории планирования реального времени набор задач** называется **выполнимым (feasible)** на заданной процессорной платформе тогда и только тогда, когда существует некоторый алгоритм планирования А, который может планировать все возможные наборы работ, порождаемые набором задач Т, полностью выполняя все их временные требования и ограничения. В свою очередь, набор задач Т называется А-планируемым (schedulable) на заданной процессорной платформе тогда и только тогда, когда А соблюдает все временные требования и ограничения при планировании всех возможных наборов работ, которые могут быть порождены всеми задачами t\_i ∈ Т.

Алгоритм планирования А является оптимальным по отношению к процессорной платформе и модели задач, если он может планировать все наборы задач (∀ Т), которые соответствуют заданной модели задач и являются выполнимыми на данной платформе.

# Rate Monotonic (RM) анализ выполнимости.

Для СРВ, применяющихся в обработке периодических событий, в 1970 году Лиу и Лейленд предложили математический аппарат, позволяющий определить, является ли система диспетчируемой. Этот аппарат называется «Частотно монотонный анализ» (ЧМА) (Rate Monotonic Analyzing). Эффективность данного математического аппарата привела к тому, что ЧМА был принят в качестве стандарта такими организациями, как USA Department of Defense, Boeing, General Dynamics, Magnavox. Mitre, NASA, Panamax и др. Среди организаций, установивших ЧМА в качестве стандартного средства анализа и разработки систем жесткого реального времени можно также отметить IBM Federal Sector Division, US Navy и European Space Agency.

Подобная позиция ведущих производителей привела к тому, что разработчикам ОСРВ пришлось учитывать требования по применению ЧМА при разработке своих систем. Возможность применения ЧМА ограничена рядом условий, первым из которых является диспетчеризация потоков методом вытесняющей приоритетной многозадачности.

На сегодняшний день существует ряд инструментов математического анализа, позволяющих распределить приоритеты между несколькими потоками таким образом, чтобы они гарантировано выполняли свои критические сроки обслуживания.

Если же для данного набора потоков реализовать это невозможно, то результаты математического анализа покажут, какие именно потоки имеют критическое отношение срока обслуживания ко времени выполнения.

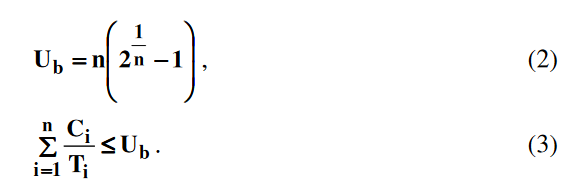
Упомянутый ранее аппарат ЧМА позволяет провести такое исследование для случая периодических критических времен обслуживания. Однако для его применения анализируемые потоки должны иметь уникальные значения приоритетов, определяемые периодом каждого потока. В связи с этим требованием разработчики ОСРВ закладывают в своих системах достаточно большое количество приоритетов. Для примера в QNX 6.x их 64, а в Windows СЕ и VxWorks — 256.

Существуют два распространенных способа назначения приоритетов:

* RMS (rate monotonic scheduling). Правило назначения приоритетов таково: чем меньше период задачи, тем выше у нее приоритет. Иными словами, чем чаще (отсюда га) задача переходит в состояние готовности, тем ее приоритет выше;
* DMS (deadline monotonic scheduling). B этом алгоритме приоритеты назначаются по немного другому правилу: чем меньше относительный крайний срок задачи, тем выше ее приоритет.

Алгоритм планирования, монотонный по частоте (ratemonotonic scheduling - RMS, ПМЧ) основан на использовании фиксированных приоритетов. Все задачи упорядочиваются по значению частоты порождения задачи (величине, обратной периоду задачи), задаче с наибольшей частотой назначается наибольший приоритет. В случае, когда в системе несколько задач имеют одинаковую частоту, их порядок определяется произвольно. Алгоритм ПМЧ предложен для систем, в которых относительный срок выполнения каждой задачи равен ее периоду. При выполнении всех условий алгоритм ПМЧ оптимален.

* 1. **Граничная плотность загрузки.** Разработчиками алгоритма ПМЧ был также предложен метод анализа выполнимости многозадачных приложений. В качестве критерия используется так называемая граничная плотность загрузки (utilization bound). Если плотность загрузки процессора приложением не превышает рассчитываемой границы, то приложение выполнимо всегда. Предлагаемый метод анализа выполнимости не является точным, условие с граничной плотностью загрузки является достаточным, то есть если оно не выполняется, из этого не следует с неизбежностью, что приложение невыполнимо. Ниже приведены формула для расчета граничной плотности загрузки и основное условие критерия:



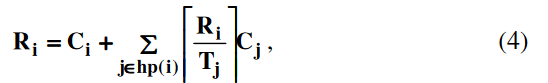
При больших n значение граничной плотности загрузки сходится к 69%. Отсюда следует интересный вывод, что любое приложение выполнимо, если плотность загрузки процессора приложением не превышает 69%. Однако не следует забывать, что это справедливо только при соблюдении всех изначальных условий метода. Применение этого критерия называют частотно-монотонным анализом (rate-monotonic analysis - RMA, ЧМ-анализ) выполнимости приложения реального времени. Более совершенные критерии, рассматриваемые ниже, часто также называют ЧМ-анализом.

Доказательство справедливости критерия оценки выполнимости базируется на понятии критического момента (critical instant), момента, в который одновременно порождаются все задачи. Показано, что время отклика задач, порожденных в критический момент, будет максимальным. В этом случае, если доказано, что задача, порожденная в критический момент, выполняется в установленные сроки, то она выполняется своевременно всегда. Все рассматриваемые далее методы анализа выполнимости для алгоритмов, использующих фиксированные приоритеты, также базируются на понятии критического момента.

* 1. **Базовый метод анализа выполнимости.** В предложен метод анализа выполнимости, точный при выполнении всех условий. Будем называть этот метод базовым, так как на его основе строятся практически все современные методы оценки выполнимости приложений реального времени.

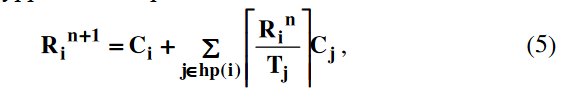
Суть базового метода анализа выполнимости заключается в определении максимального значения времени отклика задач (R) - общей продолжительности выполнения задач с учетом времени их вытеснения более приоритетными задачами. Выполнение задачи Ti всегда будет завершаться своевременно, если максимальное значение времени отклика не превышает относительного срока выполнения (Ri <Di).

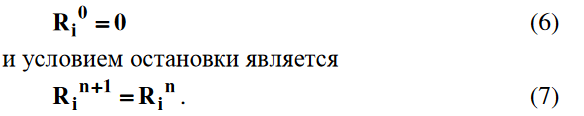
Итак, время отклика задачи т складывается из времени выполнения самой задачи и времени ее вытеснения (суммы длин временных интервалов, в течение которых задача ожидает выполнения более приоритетных задач):



где hр(i) - множество задач с приоритетами большими, чем приоритет задачи ti

Следует заметить, что время отклика фигурирует как в левой, так и в правой части уравнения, то есть формула рекурсивна. Найти RI можно, используя рекуррентное выражение:





Приложение реального времени признается выполнимым, если для всех задач максимальное значение времени отклика не превышает относительного срока выполнения.

# Deadline Monotonic (DM) анализ выполнимости.

# Монотонная по частоте модель разработана для систем, в которых для всех задач период совпадает с относительным сроком выполнения. Но на практике часто встречаются задачи, у которых относительный срок выполнения меньше периода. Для систем, содержащих такие задачи, алгоритм ПМЧ утрачивает свойство оптимальности, оптимальным для них является алгоритм планирования, монотонный по срокам (deadline-monotonic scheduling — DMS, ПМС). Согласно алгоритму ПМС все задачи упорядочиваются по значению относительного срока выполнения, задаче с наименьшим относительным сроком назначается наибольший приоритет. В случае, когда в системе несколько задач имеют одинаковый относительный срок выполнения, их порядок определяется произвольно.

# Алгоритм ПМС является обобщением алгоритма ПМЧ, и для него также применим рассмотренный ранее базовый метод анализа выполнимости.

# Список использованных источников

1. Ю.Б. Гриценко Учебное пособие «СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»: Томский межвузовский центр дистанционного образования
2. ЛЕКЦИИ ПО ДИСЦИПЛИНЕ «СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»
3. Горошко Егор, ОПЕРАЦИОННЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ
4. Никифоров В.В., Павлов В.А.. Операционные системы реального времени для встроенных программных комплексов. //Программные продукты и системы.- 1999.- №4.- С.24-30.
5. Kopetz H. Real-Time Systems. Design Principles for Distributed Embedded Applications. Dordrecht: Kluwer Academic Publishers. 1997.
6. Stankovic J.A., Spuri M., Natale M.D., Buttazzo G. Implication of Classical Scheduling Results For Real-Time Systems. Pisa, 1994.