Планирование систем с асинхронным стартом

А.И. Грюнталь

Аннотация. В статье рассматривается планирование систем, содержащих конечное количество задач с произвольным моментом старта. Изучаются условия, при которых система разрешима, то есть существует планирование, при котором каждая из образующих систему задач может быть завершена до заданного момента времени. Изучаются монотонные планирования, характеризуемые тем, что исполнению в текущий момент времени подлежит задача с наименьшим ресурсом времени. Основной результат состоит в том, что если система разрешима, то существует монотонное разрешающее планирование. Результаты статьи могут найти применение при проектировании вычислительных систем, функционирующих в режиме жесткого реального времени.

Ключевые слова: системы реального времени, программное обеспечение, многозадачность, планирование, разрешимые системы, монотонное планирование, асинхронный старт.

Введение

Вычислительные системы реального времени – это системы, для которых специфицированы ограничения на время выполнения решаемых задач [1].

Одна из задач планирования для систем реального времени состоит в следующем: распределить ресурсы процессора таким образом, чтобы обеспечить выполнение решаемых системой задач к требуемому для каждой задачи моменту времени.

Основной вопрос, возникающий в связи планированием систем реального времени, следующий: является ли система реального времени *разрешимой*, то есть, существует ли планирование, при котором все задачи системы исполняются своевременно. При положительном ответе на этот вопрос возникает проблема нахождения конструктивного алгоритма или класса алгоритмов, задающих разрешимые планирования.

В настоящей статье объектом изучения являются конечные множества задач общего вида, для которых не заданы ограничения на периодичность возникновения задач, требуемый ресурс и максимальную длительность исполнения. Ниже такие множества мы и будем называть *системами*.

Рассматривается специальный класс функций планирования – *монотонные* планирования. Планирование называется монотонным, если исполнению в текущий момент времени подлежит такое задание, у которого наименьший запас времени для своевременного исполнения.

Основной результат заключается в том, что если система разрешима, то существует монотонное разрешающее планирование. Этот результат является обобщением ряда результатов, полученных для периодических или псевдопериодических систем, в которых введены определенные ограничения на моменты возникновения задач, требуемый ресурс для их выполнения [2, 3].

Настоящая статья продолжает цикл статей [4, 5], в которых изучаются различные аспекты планирования произвольных конечных систем.

Результаты статьи могут быть использованы при создании приложений, функционирующих в среде разработанной в НИИСИ РАН операционной системы реального времени ос2000 [6].

Основные определения

Заданием назовем четверку чисел (t, L, T, I), такую что $t \geq 0$, $0 < L \leq T$, I – целое положительное число. Число t называется *моментом старта* задания, L – *ресурсной длительностью*, а T – *максимальной длительностью выполнения* задания. Целое число I назовем *индексом* задания. Момент старта задания, ресурсную длительность и максимальную длительность задания с индексом I будем обозначать соответственно через t_I, L_I и T_I.

Системой S называется конечное множество заданий, такое что индексы входящих в систему заданий попарно различны. *Подсистемой* системы S называется любое непустое подмножество заданий из S. Допускается случай пустой системы, то есть системы, не содержащей ни одного задания.

Среди всех моментов старта входящих в систему заданий есть минимальный. Минимальный момент старта называется моментом старта системы. Если S – система, то момент старта системы будем обозначать через start(S).

Начальной подсистемой системы S будем называть подсистему, состоящую из тех заданий, момент старта которых совпадает со start(S). Начальная подсистема обозначается через beg(S).

Через IndSet(S) будем обозначать множество индексов заданий, входящих в систему S, а через IndSet(S)+ - *расширенное множество индексов* системы S, получаемое добавлением к множеству IndSet(S) целого числа нуль. Полуинтервал [start(S), $+\infty$) будем называть *областью определения системы*.

Если даны две системы S_1 и S_2, и множества индексов этих систем IndSet(S_1) IndSet(S_2) не пересекаются, то определено объединение этих систем. Оно образовано всеми заданиями из S_1 и S_2. Объединение двух систем будем обозначать через $S_1 + S_2$.

Система S называется системой с *синхронным стартом* (*синхронной системой*), если моменты старта всех составляющих систему заданий совпадают. В противном случае система называется системой с *асинхронным стартом* (*асинхронной системой*).

Пусть X – конечное множество. Тогда через card(X) будем обозначать количество элементов множества X. Если S – система, то, соответственно, через card(S) обозначается количество заданий этой системы.

Будем пользоваться также следующими определениями. Пусть дана пара чисел a, b, и $a < b$. Полуинтервалом [a, b) будем называть множество чисел t таких, что $a \leq t < b$. Число a называется левой границей полуинтервала, а b – правой границей. Если c – полуинтервал, то левая граница обозначается через inf(c), а правая – через sup(c). Длину полуинтервала c будем обозначать через long(c). Если $b = +\infty$, то полуинтервал называется *бесконечным*. В противном случае полуинтервал называется *конечным*.

Два полуинтервала будем называть смежными, если левая граница одного полуинтервала совпадает с правой границей другого.

Ниже нам придется рассматривать множества (наборы) непересекающихся полуинтервалов. Мы будем различать множество полуинтервалов и множество чисел, из которых состоит объединение этих полуинтервалов.

Очевидно, что одно и то же множество чисел может быть представлено в виде объединения непересекающихся полуинтервалов поразному. Например, множество чисел, составляющих один полуинтервал [0, 2), можно также представить в виде объединения двух полуинтервалов [0, 1) и [1, 2). Однако представление множества в виде объединения несмежных полуинтервалов однозначно.

Обобщим определение левой и правой границы на множества, представимые в виде объединения конечного количества непересекающихся полуинтервалов. Пусть C такое множество, и $c_1,\ldots,c_n$ - набор непересекающихся полуинтервалов, объединение которых дает C. Тогда наименьшая из левых границ полуинтервалов $c_1,\ldots,c_n$ называется левой границей множества C, а наибольшая из правых границ – правой границей множества C. Левая и правая границы множества C, представимого в виде набора непересекающихся полуинтервалов, обозначаются соответственно через inf(C) и sup(C). Сумму длин полуинтервалов $c_1,\ldots,c_n$ будем обозначать через long(C). Если C – пустое множество, то будем считать, что long(C) = 0.

Заметим также, что если множество C представимо в виде объединения непересекающихся полуинтервалов, и [a, b) – произвольный полуинтервал, то пересечение [a, b) с C либо пусто, либо представимо в виде набора непересекающихся полуинтервалов.

Напомним определение прообраза значения функции. Пусть функция p определена на конечном или бесконечном полуинтервале $[\tau_1, \tau_2)$. Пусть x – одно из значений, которые принимает функция p. Тогда прообразом значения x для функции p называется такое множество чисел t из $[\tau_1, \tau_2)$, для которых p(t) = x. Прообраз значения x для функции p обозначается через coimg(p)(x).

В [4] было дано определение функции планирования. Напомним его. *Функцией планирования* системы S назовем такую функцию p, для которой выполнены следующие требования:

фп1) функция p определена на полуинтервале start(S) $\leq$ t < +∞;

фп2) множество значений функции p равно IndSet(S)+;

фп3) для каждого индекса I из IndSet(S) прообраз индекса для функции p представляет собой объединение конечного количества непересекающихся полуинтервалов;

фп4) для каждого индекса I из IndSet(S) сумма длин полуинтервалов, составляющих прообраз индекса I функции p, равна ресурсной длительности задания с этим индексом;

фп5) для каждого индекса I из IndSet(S) выполняется условие: если число t такое, что p(t) = I, то $t_I \leq$ t, где t_I - момент старта задания с индексом I.

Требования фп3) – фп5) заключаются в том, что для каждого из заданий системы S выполняется определенное условие, свое для каждого из требований. Для удобства изложения эти условия также будем обозначать через фп3) – фп5).

В соответствии с введенными выше обозначениями прообраз индекса I для функции p обозначается через coimg(p)(I). Свойство фп3) заключается в том, что для каждого I множество coimg(p)(I) однозначно представимо в виде объединения конечного количества несмежных полуинтервалов.

С использованием этих обозначений, требования фп4) и фп5) можно сформулировать следующим образом: для каждого задания (t, L, T, I) из S должны выполняться следующие условия:

- long(coimg(p)(I)) = L;
- inf(coimg(p)(I)) $\geq t_I$.

Число inf(coimg(p)(I)) будем называть *моментом (или временем) начала исполнения задания*, а число sup(coimg(p)(I)) - *моментом (или временем) завершения задания* с индексом I. Момент завершения задания с индексом I обозначается через fin_I. Таким образом,

fin_I = sup(coimg(p)(I)).

Пусть S – система, p – функция планирования этой системы и S' – подсистема S. Пусть I – один из индексов подсистемы S'. Тогда множество, представляющее собой объединение всех прообразов coimg(p)(I) по всем I из IndSet(S'), будем называть *прообразом системы* S' для функции планирования p. Прообраз определен и для системы S в целом.

Прообраз системы S' будем обозначать через coimg(S',p). Прообраз системы S состоит из тех чисел t из [start(S), +∞), для которых p(t) > 0.

Будем говорить, что функция планирования *разрешает систему* S, если для каждого задания (t, L, T, I) из S выполнено следующее *условие разрешимости*:

$fin_I \leq$ t + T.

Система называется *разрешимой*, если существует хотя бы одна функция планирования, разрешающая систему. Условие разрешимости также будем обозначать через фп6).

Пусть S – система и p – функция планирования этой системы. Тогда пару (S,p) будем называть *системой с (заданным) планированием*.

Замечание. Ниже мы будем конструировать функции планирования, исходя из одной или нескольких заданных функций планирования. При этом нам часто потребуется проверять выполнение условий фп3) – фп5), а также - выполнение условия разрешимости. В связи с этим сделаем очевидное замечание, последовательное применение которого позволит избежать включения в текст изложения повторяющихся, по существу, проверок.

Пусть S – система и p – функция планирования этой системы. Рассмотрим систему S' и некоторую функцию p', определенную на полуинтервале [start(S'), +∞). Пусть, далее, (t, L, T, I) – задание, принадлежащее как S, так и S'. Тогда, если coimg(p')(I) = coimg(p)(I), то для функций p' и для задания (t, L, T, I) выполнены условия фп3) – фп5).

Кроме того, если для задания (t, L, T, I) и функции p выполнено условие разрешимости, то оно будет выполнено и для этого задания и функции p'.

В [5] были даны определения *связной системы* и *полной функции* планирования. Приведем их.

Пусть дана система S. Тогда для каждого s, такого что start(S) < s, обозначим через long(S)(s) сумму ресурсных длительностей всех заданий системы S, момент старта которых меньше чем s. Сумму всех ресурсных длительностей системы S будем обозначать через ||S||. Таким образом, ||S|| = long(S)(+∞).

Отметим также, что для любой функции планирования p выполнено равенство ||S|| = long(coimg(S,p)).

Система S называется *связной*, если для любого задания (t, L, T, I) из S выполняется неравенство t < start(S) + long(S)(t). Система с синхронным стартом является связной по определению.

Носителем системы S называется полуинтервал, левая граница которого совпадает с моментом старта системы start(S), а правая граница равна start(S)+||S||.

Носитель системы S обозначается через Supp(S). В соответствии с данными определениями long(Supp(S)) = ||S||.

Функция планирования p связной системы называется *полной*, если для всех t из Supp(S) выполняется неравенство p(t) ≠ 0.

Критические значения

Пусть дана система S. Точка t называется *критическим значением* (или *критической точкой*) системы S, если существует хотя бы одно задание (t_I, L_I, T_I, I), такое что t = t_I. Одному критическому значению может соответствовать несколько заданий, в том смысле, что одно критическое значение может совпадать с моментами старта нескольких заданий.

Момент старта системы S является, очевидно, критическим значением этой системы. Количество критических значений системы не превосходит количества заданий в системе.

Множество критических значений системы S обозначается через crit(S), а количество критических значений системы S обозначается соответственно через card(crit(S)).

Критические значения будем нумеровать в порядке возрастания, начиная с 1. Тем самым, критические значения имеют номера от 1 до card(crit(S)). Пусть τ_i и τ_{i+1} – два последовательных критических значения, $\tau_i < \tau_{i+1}$, 1 ≤ i < card(crit(S)) - 1. Тогда полуинтервал [τ_i, τ_{i+1}) называется *стационарным*. Также стационарным будем называть бесконечный полуинтервал [τ_i, τ_{i+1}), где i = card(crit(S)), а $\tau_{i+1} = +\infty$.

Самый левый стационарный полуинтервал, т.е. полуинтервал, левая граница которого равна τ_1, называется *начальным стационарным* полуинтервалом.

Ограничения систем

В этом разделе мы определим понятие ограничения системы на полуинтервал и ограничение системы на критическое значение.

Пусть S – произвольная система и τ_1, τ_2 такие точки, что start(S) ≤ $\tau_1 < \tau_2$. *Ограничением системы* S на полуинтервал [τ_1, τ_2) называется подсистема S[τ_1, τ_2), состоящая из тех заданий (t, L, T, I), для которых выполняется неравенство $\tau_1 \le t < \tau_2$.

Допускается случай, когда $\tau_2 = +\infty$. В этом случае S[τ_1, τ_2) состоит из тех и только тех заданий (t, L, T, I), для которых выполняется неравенство $\tau_1 \le t$.

Заметим, что ограничение системы на стационарный полуинтервал всегда синхронная система - общим моментом старта всех заданий этой системы будет левая граница стационарного полуинтервала.

Пусть τ – критическое значение системы S. *Ограничение системы S на критическое значение* τ – это подсистема системы S, состоящая из таких заданий (t,L,T,I), что t = τ. Ограничение системы на критическое значение обозначается через S(τ).

Теперь дадим определение *левого и правого ограничения системы*, а также *левого и правого ограничений функции планирования*. В отличие от ограничения системы левое и правое ограничения зависят не только от системы, но и от функции планирования.

Минимальный момент начала исполнения задания (среди всех моментов начала исполнения задания системы S) называется *моментом начала исполнения системы*. Соответственно, максимальный момент завершения исполнения задания называется *моментом завершения исполнения системы*. Минимальный момент начала исполнения системы обозначается через timebeg(S,p), момент завершения исполнения системы – timefin(S,p).

Заметим, что если S - связная система, а p – полная функция планирования, то

timebeg(S,p) = start(S), а timefin(S,p) = start(S) + long(Supp(S)).

Теперь определим левое и правое ограничения системы. Пусть (t, L, T, I) – произвольное задание системы S. Для каждого $\tau >$ start(S) введем определение исполненной части задания.

Обозначим через $\sigma(I,p,\tau)$ длину пересечения coimg(p)(I) с полуинтервалом [start(S), τ). Число $\sigma(I,p,\tau)$ называется *исполненной частью* задания с индексом I. Очевидно, исполненная часть задания зависит от функции планирования p.

Пусть τ_1 – момент начала исполнения этого задания, а τ_2 – момент завершения исполнения. Если $\tau \leq \tau_1$, то $\sigma(I,p,\tau) = 0$. Если $\tau_1 < \tau < \tau_2$, то $0 < \sigma(I,p,\tau) < L$. Если $\tau_2 \leq \tau$, то $L = \sigma(I,p,\tau)$.

Рассмотрим задание (t, L, T, I), такое что $t < \tau$. Если для задания с индексом I выполнено неравенство $0 < \sigma(I, p, \tau) < L$, то это задание называется *частично выполненным* (на момент времени τ). Если же $L = \sigma(I, p, \tau)$, то задание называется *выполненным*. Если же $\sigma(I, p, \tau) = 0$, то задание называется *не выполнявшимся*.

Если $\tau \leq t$, то задание (t, L, T, I) называется *не стартовавшим* к моменту времени τ.

Пусть (t, L, T, I) - частично выполненное задание. *Левым ограничением* задания назовем такое задание (t', L', T', I), для которого

t' = t,

L' = σ(I, p, τ),

T' = T.

Задание (t, L, T, I) по отношению к левому ограничению(t', L', T', I) называется *исходным*. Исходное задание и его левое ограничение имеют один и тот же индекс.

Пусть $\tau >$ start(S). *Левым ограничением системы* с планированием (S,p) назовем систему, состоящую из заданий, которые являются левыми ограничениями, и заданий, выполненных на момент времени τ. Левое ограничение системы с планированием (S,p) обозначается через lft(S,p)(τ).

Если start(S) $< \tau \leq$ timebeg(S,p), то левое ограничение системы с планированием (S,p) – пустая система.

Теперь введем понятие *правого ограничения системы*. Сначала определим понятие правого ограничения для частично выполненных заданий и для заданий, не выполнявшихся на момент времени τ.

Правым ограничением для частично выполненного задания (t, L, T, I) назовем такое задание (t', L', T', I), для которого

t' = τ,

L' = L - σ(I, p, τ),

T' = T – (τ – t).

Правое ограничение для не выполнявшегося задания по определению имеет вид (τ, L, T-(τ-t), I).

Задание (t, L, T, I) по отношению к правому ограничению(t', L', T', I) также называется *исходным*. Исходное задание и его правое ограничение имеют один и тот же индекс.

Правым ограничением системы с планированием (S,p) назовем систему, состоящую из заданий, которые являются правыми ограничениями на момент времени τ, и заданий, не стартовавших к моменту времени τ. Правое ограничение системы с планированием (S,p) обозначается через wrt(S,p)(τ).

Если timefin(S,p) $\leq \tau$, то правое ограничение – пустая система.

Заметим, что IndSet(S) = IndSet(lft(S,p)(τ)) + IndSet(wrt(S,p)(τ)).

Лемма (*о совпадении ограничений*). Пусть функции планирования p_1 и p_2 совпадают на полуинтервале [start(S), τ). Тогда система lft(S,p_1)(τ) совпадает с lft(S,p_2)(τ), а wrt(S,p_1)(τ) совпадает с wrt(S,p_2)(τ).

Доказательство леммы

В силу условия леммы для любого задания справедливо равенство

coimg(p')(I) $\cap$ [start(S), τ) = coimg(p)(I) $\cap$ [start(S), τ),

где через $\cap$ обозначено теоретико-множественное пересечение. Поэтому для каждого задания (t, L, T, I) величины σ(I, p_1, τ) и σ(I, p_2, τ) совпадают. Отсюда следует, что множества выполненных и частично выполненных заданий для систем (S,p_1) и (S,p_2) совпадают, и что левые ограничения (а значит и правые ограничения) частично выполненных заданий также совпадают. Поэтому $lft(S,p_1)(\tau)$ совпадает с $lft(S,p_2)(\tau)$.

Множества не стартовавших на момент времени τ заданий систем (S,p_1) и (S,p_2) совпадают – они не зависят от функция планирования. Поэтому множества не начавших выполняться заданий совпадают. Поэтому $wrt(S,p_1)(\tau)$ совпадает с $wrt(S,p_2)(\tau)$.

Лемма доказана.

Дадим теперь определение левого и правого ограничений функции планирования. Левое ограничение функции планирования определено в том случае, когда левое ограничение системы не пусто. Соответственно правое ограничение функции планирования определено, когда правое ограничение системы планирования не пусто.

Левым ограничением функции планирования p называется такая функция lft(p), которая на полуинтервале [start(S), τ) совпадает с p, а на полуинтервале [τ, $+\infty$) равна нулю. Левое ограничение функции планирования определено на полуинтервале [start(S), $+\infty$).

Рассмотрим частично выполненное задание (t, L, T, I). Так как p совпадает с lft(p)(τ) на [start(S), τ), и при $t \geq \tau$ выполнено $lft(p)(\tau) = 0 \neq I$, то $coimg(lft(p)(\tau))(I) = coimg(p)(I) \cap [start(S), \tau)$, поэтому

$$long(coimg(lft(p)(\tau))(I)) = \\ =long(coimg(p)(I) \cap [start(S), \tau)) = \sigma(I, p, \tau).$$

Правым ограничением функции планирования называется функция wrt(p), определенная на полуинтервале [τ, $+\infty$) и совпадающая на нем с p.

Пусть (t, L, T, I) – частично выполненное задание. Так как p совпадает с wrt(p)(τ) на [τ, $+\infty$), а на [start(S), τ) функция wrt(p)(τ) не определена, то

$$coimg(wrt(p)(\tau))(I) = coimg(p)(I) \cap [\tau, +\infty).$$

Поэтому $coimg(p)(I) = coimg(lft(p)(\tau))(I) + coimg(wrt(p)(\tau))(I)$, где знаком + обозначена операция объединение множеств.

Отсюда следует, что

$$long(coimg(wrt(p)(\tau))(I)) = \\ =long(coimg(p)(I) \cap [\tau, +\infty)) = L - \sigma(I, p, \tau).$$

Кроме того,

$$long(coimg(p)(I)) = long(coimg(lft(p)(\tau))(I)) + long(coimg(wrt(p)(\tau))(I)).$$

Лемма (*об ограничениях функции планирования*). Левое и правое ограничения функции планирования системы являются функциями планирования левого и правого ограничений этой системы. Левое и правое ограничения разрешающей функции планирования являются разрешающими функциями планирования.

Доказательство леммы

Пусть p – функция планирования системы S.

Требование фп1) для lft(p)(τ) и wrt(p)(τ) выполнено очевидно.

Проверим выполнение фп2). Задание с индексом I входит в lft(S,p)(τ) в том случае, когда существует принадлежащий [start(S), τ) полуинтервал c, на котором функция p принимает значение I. Но тогда значение I на полуинтервале c принимается и функцией lft(p)(τ), поскольку на полуинтервале [start(S), τ) функции p и lft(p)(τ) совпадают. Верно и обратное: если lft(p)(τ) принимает на некотором полуинтервале из [start(S), τ) значение I, то это значение принимает и функция p.

Поэтому IndSet(lft(S,p)(τ)) совпадает с множеством индексов заданий из функции lft(p)(τ).

Аналогично, если задание с индексом I входит в wrt(S,p)(τ), то на [τ,$+\infty$) существует полуинтервал, на котором функции p и wrt(p)(τ) принимают значение I. Поэтому IndSet(wrt(S,p)(τ)) совпадает с множеством индексов заданий из функции wrt(S,p)(τ). Итак, требование фп2) для lft(p)(τ) и wrt(p)(τ) выполнено.

Пусть (t, L, T, I) – задание, выполненное на момент времени τ. Тогда $coimg(lft(p)(\tau)(I)) = coimg(p)(I)$. Значит, для этого задания и функции lft(p)(τ) выполнены условия фп3) – фп6). Аналогично проверяется выполнение условий фп3) – фп6) для не начавших выполняться заданий, не стартовавших заданий и функции wrt(p)(τ).

Пусть (t, L, T, I) входит в wrt(S,p)(τ) и $\tau \leq t$. Тогда прообразы функций wrt(p)(τ) и p для значения I совпадают. Значит, для (t, L, T, I) условия фп3) – фп6) выполнены.

Проверим выполнение условий фп3) – фп6) для левых и правых ограничений частично выполненных заданий.

Пусть (t, L, T, I) – частично выполненное задание, (t', L', T', I) и (t'', L'', T'', I) –его левое и правое ограничения соответственно.

Тогда coimg(lft(p)(τ))(I) является пересечением coimg(p)(τ))(I) с полуинтервалом [start(S), τ), а coimg(wrt(p)(τ))(I) – пересечением coimg(p)(τ))(I) с [τ, +∞). Поэтому coimg(lft(p)(τ))(I) и coimg(wrt(p)(τ))(I) представимы в виде объединения конечного количества полуинтервалов. Значит, условие фп3) для правых и левых ограничений выполнено.

Выполнение свойства фп4) для правого и левого ограничений частично выполненного задания непосредственно следует из того, что σ(I, p, τ) = long(coimg(lft(p)(τ))(I)), и L - σ(I, p, τ) = long(coimg(wrt(p)(τ))(I)). Таким образом, фп4) для частично выполненных заданий выполнено.

Проверим выполнение фп5).

Пусть (t, L, T, I) – частично выполненное задание и (t', L', T', I) – левое ограничение этого задания. Тогда t' = t. Так как для функции p требование фп5) выполнено, то t ≤ inf(coimg(p)(I)). Но для частично выполненного задания inf(coimg(p)(I)) = inf(coimg(lft(p)(τ))(I)). Поэтому для задания (t', L', T', I) требование фп5) выполнено.

Пусть (t'', L'', T'', I) – правое ограничение задания (t, L, T, I). Поскольку t'' = τ = start(wrt(S,p)(τ)), то условие фп5) для (t'', L'', T'', I) выполнено очевидно. Итак, требование фп5) выполнено.

Осталось проверить, условие: если p - разрешающая функция, то правое и левое ограничения этой функции также будут разрешающими.

Если (t',L',T',I) – левое ограничение задания (t,L,T,I), то sup(coimg(lft(p)(τ))) ≤ sup(coimg(p)) ≤ t + T = t' + T'. Поэтому условие разрешимости выполнено.

Пусть (t'', L'', T'', I) – правое ограничение задания (t, L, T, I). По условию sup(coimg(p)(I)) ≤ t + T. Поскольку sup(coimg(p)(I)) > τ, а на [τ, +∞) функции p и wrt(p)(τ) совпадают, то sup(coimg(wrt(p)(τ))(I)) ≤ t + T = t'' + T''. Таким образом, условие разрешимости выполнено и для правого ограничения.

Лемма доказана.

Монотонные функции планирования систем с синхронным стартом

Пусть S – система с синхронным стартом. Пусть t_0 – момент старта системы и, следовательно, момент старта каждого из заданий системы. Введем упорядочение на заданиях, составляющих систему S.

Пусть (t_I, L_I, T_I, I) и (t_J, L_J, T_J, J) – два задания ($t_I = t_J = t_0$). Тогда задание (t_I,L_I,T_I,I) предшествует заданию (t_J,L_J,T_J,J), если $T_I < T_J$. Если $T_I = T_J$, то задание с индексом I предшествует заданию с индексом J, если I < J.

В смысле заданного порядка среди всех заданий существует единственное наименьшее задание, и для каждого задания, кроме наибольшего, существует единственное последующее.

Введем понятие *монотонной нумерации*. Пусть система S включает N заданий. Обозначим через [1:N] множество первых N натуральных чисел.

Монотонной нумерацией будем называть такое отображение

ind: [1:N] → IndSet(S), для которого выполнены следующие утверждения:

мн1) ind(1) – индекс наименьшего в смысле введенного порядка задания;

мн2) если задание с индексом I имеет номер k, то есть I = ind(k), и k < N, то число ind(k+1) равно индексу задания, следующего за заданием с индексом I.

Очевидно, что монотонная нумерация существует и единственна.

Теперь определим *монотонную функцию планирования*. Для этого построим систему полуинтервалов $h_1,\ldots,h_N$.

Положим $h_1 = [t_0, t_0 + L_{ind\ (1)})$. Далее, для $m = 2,\ldots,N$ положим $h_m = [\sup(h_{m-1}), \sup(h_{m-1}) + L_{ind(m)})$.

Функция планирования p синхронной системы называется *монотонной*, если значение этой функции на полуинтервале h_k равно ind(k) и на бесконечном полуинтервале $[\sup(h_N), +\infty)$ значение функции p равно нулю.

Заметим, что монотонная функция планирования для систем с синхронным стартом определена однозначно.

Пусть I – индекс задания и ind(k) = I. Тогда число k называется *монотонным индексом* этого задания и обозначается через mon(I). Также положим, что mon(0)=0. Таким образом, функция mon определена на IndSet(S)+.

Участвующие в определении монотонной функции планирования полуинтервалы $h_1,\ldots,h_N$ образуют разбиение носителя системы S. Это разбиение будем называть разбиением, *ассоциированным с синхронной системой* S.

Сформулируем критерий монотонности функции планирования системы с синхронным стартом. Напомним, что функция f называется *монотонно неубывающей*, если из $\tau_1<\tau_2$ следует, что $f(\tau_1) \leq f(\tau_2)$.

Пусть S – система с синхронным стартом, p – функция планирования этой системы, и пусть card(S) = N. Обозначим через mon(p) композицию функций mon и p. Другими словами значение функции mon(p) в точке t равно mon(p(t)).

Лемма (*критерий монотонности функции планирования системы с синхронным стартом*). Функция планирования p системы S с синхронным стартом является монотонной тогда и только тогда, когда ограничение mon(p) на носитель системы S является монотонно неубывающей функцией с положительными значениями.

Доказательство леммы

Если p – монотонная функция планирования, то на полуинтервале h_m значение функции mon(p) равно m. Поскольку $\sup(h_m) = \inf(h_{m+1})$, то функция mon(p) монотонная на носителе системы S.

Пусть, наоборот, p – функция планирования системы S, и mon(p) монотонно неубывающая на носителе системы S функция. Покажем, что p – монотонная. Пусть I – индекс одного из заданий системы S.

Тогда $\sup(\mathrm{coimg}(p)(I)) - \inf(\mathrm{coimg}(p)(I)) = L_I$ – ресурсной длительности задания с индексом I. Предположим, что это неверно. Это означает, что $\sup(\mathrm{coimg}(p)(I)) - \inf(\mathrm{coimg}(p)(I)) > L_I$.

Обозначим через cl – такой полуинтервал, что inf(coimg(p)(I)) = inf(cl) и значение p на cl равно I. Аналогично, через cw обозначим такой полуинтервал, что sup(coimg(p)(I)) = sup(cw) и значение p на cw равно I. Тогда существует такой полуинтервал c, что $\sup(cl) \leq \inf(c)$ и $\sup(c) \leq \inf(cw)$, и $p(c) \neq I$. Но тогда mon(cl) = mon(cw) $\neq$ mon(c). А это противоречит монотонности функции mon(p).

Таким образом, coimg(p)(I) представляет собой полуинтервал длиной L_I, и эти полуинтервалы образуют разбиение носителя S. Обозначим этот полуинтервал через c(I). Полуинтервалы c(I) образуют разбиение носителя системы S (в силу того, что на носителе S функция mon(p) принимает положительные значения).

Занумеруем полуинтервалы c(I) в порядке возрастания. Определим нумерацию ind': [1:N] → IndSet(S), полагая, что inf(c(ind'(1)))=start(S) и inf(c(ind'(k+1))) = sup(c(ind'(k))), k = 1,…,N-1.

В силу условия монотонности, последовательность mon(p(c(ind'(1)))),…,mon(p(c(ind'(N))) должна быть монотонно возрастающей и состоять из первых N натуральных чисел. Но это возможно лишь в том случае, когда mon(p(c(ind'(k))) = k, то есть, когда ind'(k) = ind(k). Отсюда следует, что полуинтервал c(ind'(k)) совпадает с полуинтервалом h_k. Значит p – монотонная функция планирования.

Лемма доказана.

Монотонные функции планирования систем с асинхронным стартом

Пусть S – система с синхронным стартом и p – монотонная функция планирования. Пусть τ > start(S). *Остатком системы* S на момент времени τ назовем систему, состоящую из правых ограничений на момент времени τ заданий системы S.

Остаток системы S обозначается через rem(S)(τ). Поскольку функция планирования определена синхронной системой однозначно, то rem(S)(τ) полностью определена самой системой S.

Теперь дадим определение монотонной функции планирования для систем с асинхронным стартом. Пусть S - система и $\tau_1,\ldots,\tau_n$ - критические точки этой системы, занумерованные в порядке возрастания. Положим также $\tau_{n+1} = +\infty$.

Определим последовательность систем с синхронным стартом $S_1,\dots,S_n$. Положим $S_1 = S(\tau_1)$. Далее положим $S_{i+1} = rem(S_i)(\tau_{i+1}) + S(\tau_{i+1})$, где $i=1,\dots,n-1$. Здесь знаком + обозначено объединение систем. Пусть q_i – монотонная функция планирования, однозначно определенная системой S_i.

Определение. Функция планирования p системы S называется *монотонной*, если на каждом из полуинтервалов $[\tau_i, \tau_{i+1})$ она совпадает с q_i.

Пользуясь индукцией, легко показать, что монотонная функция планирования системы S определена однозначно.

Докажем, что монотонная функция планирования действительно является функцией планирования системы S. Монотонная функция определена на полуинтервале $[start(S),+\infty)$. Значит, фп1) выполнено.

Пусть (t, L, T, I) – произвольное задание из S. Заметим, что при $k=1,\dots,n$ выполняется соотношение

$coimg(p)(I)\cap[\tau_k, \tau_{k+1}) = coimg(q_i)(I)\cap[\tau_k, \tau_{k+1})$.

Поэтому coimg(p)(I) является объединением непересекающихся множеств $coimg(q_i)(I)\cap[\tau_k, \tau_{k+1})$. Каждое из этих множеств представимо в виде объединения непересекающихся полуинтервалов. Поэтому таким же является множество coimg(p)(I). Следовательно выполнено фп3).

Пусть $\sigma(k) = long(coimg(p)(I)\cap[\tau_k, \tau_{k+1})) = long(coimg(q_i)(I)\cap[\tau_k, \tau_{k+1}))$.

Ясно, что $\sigma(1) +\dots+ \sigma(k-1) = \sigma(I,p,\tau_k)$.

Заметим, что $IndSet(S_i)$ принадлежит IndSet(S). Поэтому множество значений монотонной функции принадлежит IndSet(S).

Пусть (t, L, N, I) – произвольное задание системы S. Тогда оно принадлежит одной из систем $S(\tau_m)$. Тогда $\sigma(1) =\dots= \sigma(m-1) = 0$.

Построим последовательность чисел L(m), L(m+1),…L(r), $r \le n$. Положим L(m) = L. Пусть $m \le k < n$, и $\sigma(k) < L(k)$. Тогда положим $L(k+1) = L(k) - \sigma(k)$. Если же $\sigma(k) = L(k)$, то r = k, и процедура построения последовательности заканчивается. Если k = n, то положим r = n. Таким образом, всегда $L(r) = \sigma(r)$. Из построения следует, что $L(k) = L - \sigma(I,p,\tau_k)$ и что $L = \sigma(m) +\dots+ \sigma(r)$.

Заметим, что согласно определению монотонной функции числа L(k), k=m,…,r, являются ресурсными длительностями заданий с индексом I системы S(k). При k > r задания с индексом I в S(k) не входят.

Поэтому пересечение coimg(p)(I) с полуинтервалами $[\tau_k, \tau_{k+1})$ не пусто только при k = m,…,r. Следовательно L = long(coimg(p)(I)). Значит, выполнены фп4) и фп2).

Требование фп5) выполнено очевидно: значение I не может приниматься ни одной из функций q_i при $i < m$. Поэтому $t = \tau_m \le inf(coimg(p)(I))$.

Итак, мы доказали, что монотонная функция планирования действительно является функцией планирования.

Так как ресурсная длительность входящего в $rem(S_{k-1})(\tau_k)$ задания с индексом I равна $L - \sigma(I,p,\tau_k)$, то это задание является правым ограничением на момент времени τ_k частично выполненного или не начавшего исполняться задания системы S с индексом I.

Поэтому система S_k совпадает с системой $beg(wrt(S,p)(\tau_k))$, а монотонная функция планирования на полуинтервале $[\tau_k, \tau_{k+1})$ совпадает с монотонной функцией планирования этой синхронной системы.

Это позволяет сформулировать критерий монотонности функции планирования. Пусть S – асинхронная система с критическими точками $\tau_1,\dots, \tau_n$ и p – функция планирования этой системы. Обозначим через p_k монотонную функцию планирования синхронной системы $beg(wrt(S,p)(\tau_k))$.

Лемма (*критерий монотонности функции планирования*). Функция планирования p будет монотонной в том и только в том случае, когда на каждом из полуинтервалов $[\tau_k, \tau_{k+1})$ функция p будет совпадать с p_k.

Доказательство леммы.

Пусть q – монотонная функция планирования системы S и q_k – монотонная функция планирования синхронной системы $beg(wrt(S,q)(\tau_k))$. Достаточно показать, что $p_k = q_k$ при всех k. Заметим, что $beg(wrt(S,p)(\tau_1)) = beg(wrt(S,q)(\tau_1)) = S(\tau_1)$. Так как монотонная функция планирования однозначно определена синхронной системой, при k = 1 утверждение верно. Пусть утверждение справедливо при некотором k. Докажем, что оно остается справедливым и при k+1.

По предположению функции p и q совпадают на полуинтервале $[\tau_1, \tau_{k+1})$. Тогда совпадают и системы $beg(wrt(S,p)(\tau_{k+1}))$ и $beg(wrt(S,q)(\tau_{k+1}))$. Это следует из леммы о совпадении ограничений. Отсюда следует, что функции p_{k+1} и q_{k+1} совпадают.

Лемма доказана.

Существование разрешающей монотонной функции планирования

Основным результатом настоящей статьи является следующая теорема.

Теорема (*о существовании разрешающей монотонной функции планирования*). Если система разрешима, то существует монотонная функция планирования, разрешающая систему.

В [4] теорема о существовании разрешающей монотонной функции планирования была доказана в частном случае, а именно, для систем с синхронным стартом. Справедливость утверждения теоремы в этом частном случае будет использована в настоящей статье для доказательства теоремы в общем виде для асинхронных систем.

Вначале теорема о разрешающей функции планирования будет доказана для связных систем, а затем, на основании результатов [5], – для систем общего вида.

Ограничения связных систем

Переходим к рассмотрению связных систем и полных функций планирования. Пусть S – связная система, p – функция планирования этой системы, и τ такое, что $start(S) < \tau < start(S) + \|S\|$. Так как p – полная функция планирования, то ее значения на полуинтервале $[start(S), start(S) + \|S\|)$ отличны от нуля. При $\tau \geq start(S) + \|S\|$ функция p равна нулю.

Рассмотрим системы $lft(S,p)(\tau)$ и $wrt(S,p)(\tau)$ и соответствующие функции планирования $lft(p)(\tau)$ и $wrt(p)(\tau)$. Очевидно, что носитель системы $lft(S,p)(\tau)$ представляет собой полуинтервал $[start(S), \tau)$, а носитель $wrt(S,p)(\tau)$ - полуинтервал $[\tau, start(S) + \|S\|$.

Заметим, что если S – связная, и p полная, то $lft(S,p)(\tau)$ может и не быть связной системой. Действительно, рассмотрим систему S, состоящую из трех заданий (0,1,3,1), (0,1,3,3) и (1,1,2,2). Определим функцию планирования p. Пусть на [0, 1) функция p равна 1, на [1,2) функция p равна 2, на [2,3) функция p равна 3. При $\tau \geq 3$ функция p принимает значение 0. Тогда $lft(S,p)(2)$ состоит из двух заданий (0,1,3,1) и (1,1,2,2) и не является связной.

Для правого ограничения это не так.

Пусть, как и раньше, S – связная система, p – полная функция планирования этой системы, $\tau > start(S)$, и τ принадлежит носителю S.

Лемма (*о связности и полноте*). Правое ограничение связной системы является связной системой. Правое ограничение полной функции планирования является полной функцией планирования.

Доказательство леммы

Пусть (t, L, T, I) – задание из $wrt(S,p)(\tau)$, и $t > \tau$.

Введем обозначения. Через S_1 обозначим множество заданий из S, выполненных на момент времени τ, через S_2 – множество частично выполненных на момент времени τ заданий, через S_3 - множество заданий, не начавшихся на момент времени τ. Наконец, через S_4 обозначим множество заданий, момент старта которых принадлежит полуинтервалу $[\tau, t)$.

Подсистемы S_1, S_2, S_3, S_4 взаимно не пересекаются. Обозначим через S' множество левых ограничений заданий из S_2, а через S'' – множество правых ограничений заданий из S_2. Поскольку в полуинтервал $[start(S), \tau)$ входят только прообразы систем S_1 и S', и на этом полуинтервале функция p отлична от нуля (в силу ее полноты), то справедливо равенство $\|S_1\| + \|S'\| = \tau - start(S)$.

Так как S – связная система, то $t < start(S) + long(S)(t)$.

Заметим, что $long(S)(t) = \|S_1\| + \|S'\| + \|S''\| + \|S_3\| + \|S_4\|$. Поэтому

$t < start(S) + \|S_1\| + \|S'\| + \|S''\| + \|S_3\| + \|S_4\|$. Отсюда следует, что

$t < \tau + \|S''\| + \|S_3\| + \|S_4\|$. Из определения правого ограничения системы видно, что

$\|S''\| + \|S_3\| + \|S_4\| = long(wrt(S,p)(\tau))(t)$.

Поэтому $t < \tau + long(wrt(S,p)(\tau))(t)$, что и требовалось доказать.

Полнота правого ограничения полной функции планирования непосредственно следует из определения.

Лемма доказана.

Теперь изучим, что происходит в случае, когда правое ограничение полной функции планирования заменяется на другую полную функцию планирования.

Пусть S – связная система, p – полная функция планирования и start(S) < τ < start(S) + ||S||. Пусть q – полная функция планирования системы wrt(S,p)(τ). Обозначим через p' функцию, совпадающую на полуинтервале [start(S), τ) с функцией p, а на [τ, +∞) – с функцией q.

Лемма (*о модификации правого ограничения функции планирования*). Функция p' будет полной функцией планирования системы S. Если p – разрешающая функция планирования, p' также будет разрешающей.

Доказательство леммы

Докажем, что p' – функция планирования. Требование фп1) выполнено очевидно. Пусть S_1, S_2, и S_3 - системы, состоящие из выполненных, частично выполненных и не начавшихся на момент времени τ заданий. Через S_4 обозначим множество заданий, не стартовавших к моменту времени τ.

Множество IndSet(S) представляет собой объединение множеств IndSet(S_1), IndSet(S_2), IndSet(S_3) и IndSet(S_4).

Множество значений функции p, а значит и функции p', на полуинтервале [start(S), τ) – это объединение IndSet(S_1) и IndSet(S_2). Обозначим это множество через Set_l. Множество положительных значений функции p на [τ, +∞) - это объединение IndSet(S_2), IndSet(S_3) и IndSet(S_4). Обозначим это множество через Set_w. Таким образом, IndSet(S) = Set_l + Set_w. Заметим, что Set_w также является множеством положительных значений функции q, а значит и p'. Поэтому IndSet(S) представляет собой множество положительных значений функции p'. Значит, фп2) выполнено.

Проверим выполнение фп3). При каждом I из IndSet(S) множество coimg(p')(I) представляет собой объединение непересекающихся множеств coimg(lft(p)(τ))(I) и coimg(q)(I). Каждое из них представимо в виде объединения непересекающихся полуинтервалов. Следовательно, таким же будет и множество coimg(p')(I). Выполнение требования фп3) доказано.

Проверим выполнение фп4).

Заметим, что long(coimg(p')(I)) = long(coimg(lft(p)(τ))(I)) + long(coimg(q)(I)).

Но long(coimg(q)(I)) = long(coimg(wrt(p)(τ))(I)), поскольку q и wrt(p)(τ) - функции планирования одной и той же системы wrt(S,p). Выше было показано, что

long(coimg(p)(I)) = long(coimg(lft(p)(τ))(I)) + long(coimg(wrt(p))(I)).

Поэтому long(coimg(p')(I)) = long(coimg(p)(I)). Требование фп4) выполнено.

Проверим выполнение фп5). Пусть (t, L, T, I) выполненное или частично выполненное на момент времени τ задание системы с планированием (S,p). Тогда inf(coimg(p')(I) = inf(coimg(p)(I). Тогда фп5) выполнено, поскольку по условию t < inf(coimg(p)(I)).

Пусть (t, L, T, I) - не начавшееся исполняться задание. Тогда inf(coimg(p')(I) = inf(coimg(q)(I). Но при этом t < inf(coimg(q)(I)), так как q - функция планирования системы wrt(S,p), а задание (t, L, T, I) принадлежит этой системе. Поэтому условие фп5) для не начавшегося задания выполнено. Аналогично проверяется фп5) для заданий, не стартоваших к моменту времени τ. Итак, требование фп5) для функции p' выполнено.

Функция p' полная, так как она, очевидно, отлична от нуля на носителе системы S.

Пусть функция p разрешима. Пусть (t, L, T, I) задание из S_1. Тогда sup(coimg(p'))(I) = sup(coimg(p))(I). следовательно, неравенство sup(coimg(p'))(I) $\le$ t + T выполняется.

Пусть (t, L, T, I) задание из S_2, и (t', L', T', I) – правое ограничение этого задания. Тогда t' = τ, T' = T – (τ – t) и sup(coimg(p'))(I) = sup(coimg(q))(I). Так как q – функция планирования, то sup(coimg(q))(I) $\le$ τ + T' = t + T. Значит, для частично выполненных заданий условие разрешимости выполнено.

Случай, когда (t, L, T, I) – не начавшееся исполняться задание, рассматривается аналогично.

Пусть (t, L, T, I) задание, не стартовавшее к моменту времени τ. Тогда это задание принадлежит wrt(S,p) и sup(coimg(p'))(I) = sup(coimg(q))(I). А для q неравенство разрешимости выполнено, поскольку q – разрешимая функция.

Лемма доказана.

Существование разрешающей монотонной функции планирования для связных систем

Следующей задачей является доказательство теоремы о существовании разрешающей монотонной функции планирования для связных систем. Будет доказано следующее утверждение: если связная система разрешима, то существует полная монотонная функция, разрешающая систему.

Пусть p – полная разрешающая функция планирования системы S, и $\tau_0,\ldots,\tau_n$ – критические точки системы S, занумерованные в порядке возрастания.

Для определения понятия *дефекта* введем ряд обозначений и определений.

Пусть D – набор непересекающихся полуинтервалов. Будем говорить, что набор D *ассоциирован* с функцией планирования p, если значение функции p на каждом полуинтервале d из D постоянно. Это значение будем называть индексом полуинтервала d и обозначать через p(d).

Напомним, что через beg(S) мы обозначаем *начальную подсистему* системы S, состоящую из заданий, момент старта которых равен τ_0. Множество индексов этой подсистемы обозначается через IndSet(beg(S)).

Прообразом начальной подсистемы называется подмножество полуинтервала $[\tau_0, +\infty)$, состоящее из таких τ, для которых выполнено условие: $p(\tau)$ принадлежит IndSet(beg(S)).

Пусть D – разбиение прообраза начальной подсистемы, ассоциированное с функцией планирования p. Разбиение называется точным в точке τ_1, если τ не является внутренней точкой ни одного из полуинтервалов D, другими словами, если для любого полуинтервала d из D либо $\tau \le \inf(d)$, либо $\sup(d) < \tau$. Итак, пусть D – разбиение, точное в τ_1.

Через L(D) обозначим *левую часть* разбиения D, через W(D) – *правую часть*. Набор L(D) включает в себя все полуинтервалы c из D, для которых выполнено условие $\sup(c) \le \tau_1$. Набор W(D) включает все полуинтервалы c из D, для которых выполнено условие $\tau_1 \le \inf(c)$.

Пусть c принадлежит L(D) и d принадлежит W(D).

Для такой пары полуинтервалов и функции планирования p определим число $p<c,d>$, которое будем называть *индикатором порядка* полуинтервалов c и d относительно функции планирования p.

Обозначим через mon монотонную нумерацию заданий из beg(S). Если $mon(p(c)) \le mon(p(d))$, то положим $p<c,d> = 0$. В противном случае, когда $mon(p(c)) > mon(p(d))$, положим $p<c,d> = 1$.

Распространим определение индикатора порядка на наборы полуинтервалов. Пусть набор полуинтервалов D_1 состоит из полуинтервалов, входящих в L(D), и набор полуинтервалов D_2 состоит из полуинтервалов, входящих в W(D). Тогда определим $p<D_1,D_2>$ как сумму чисел $p<d_1,d_2>$ по всем d_1 из D_1 и d_2 из D_2. Ясно, что индикатор неотрицателен при любых аргументах.

Пусть C – любое подмножество полуинтервалов из D. *Нижней гранью* набора полуинтервалов C относительно функции планирования p назовем наименьшее из чисел mon(p(c)), где c принадлежит C. Нижняя грань обозначается через inf(p)(C). Аналогично определим понятие верхней грани. *Верхней гранью* набора полуинтервалов C относительно функции планирования p назовем наибольшее из чисел mon(p(c)), где c принадлежит C. Верхняя грань обозначается через sup(p)(C).

Индикатор порядка обладает рядом свойств. Перечислим их.

Свойство *квазилинейности.*

Пусть, по-прежнему, D_1 и D_2 – подмножества L(D) и W(D) соответственно. Пусть d_1 принадлежит D_1 и d_2 принадлежит D_2. Тогда выполнены равенства

$p<D_1,D_2> = p<D_1\text{-}d_1,D_2> + p<d_1,D_2>$ и $p<D_1, D_2> = p<D_1,D_2\text{-}d_2> + p< D_1, d_2>$.

В этих формулах знаком «минус» обозначена теоретико-множественная разность.

Следующие свойства могут быть охарактеризованы как свойства *монотонности*.

Пусть полуинтервалы d_1 и c_1 принадлежат D_1, и полуинтервалы d_2 и c_2 принадлежат D_2. Пусть $p<d_1, d_2> = 1$. Если $mon(p(c_2)) \le mon(p(d_2))$, то $p<d_1, c_2> = 1$. Если $mon(p(c_1)) \ge mon(p(d_1))$, то $p<c_1, d_2> = 1$.

Пусть $p<d_1, d_2> = 0$. Если $mon(p(c_2)) \geq mon(p(d_2))$, то $p<d_1, c_2> = 0$. Если $mon(p(c_1)) \leq mon(p(d_1))$, то $p<c_1, d_2> = 0$.

Пусть d_1 принадлежит D_1 и $mon(p(d_1)) \leq inf(p)(D_2)$. Тогда $p<d_1, D_2> = 0$. Пусть, наоборот, d_2 принадлежит D_2 и $mon(p(d_2)) \geq sup(p)(D_1)$. Тогда $p<D_1,d_2>= 0$.

Определение. Определим *дефект* функции планирования $\Delta(D,p)$ как $p<L(D),W(D)>$.

Другими словами, дефект – это количество пар полуинтервалов, образующих нарушение порядка. Ясно, что $\Delta(D,p) \geq 0$.

Пусть p – полная разрешающая функция планирования системы S, и D – ассоциированное с p и точное в τ_1 разбиение прообраза начальной подсистемы beg(S).

Лемма (*об уменьшении дефекта*). Пусть $\Delta(D,p) > 0$. Тогда существует полная разрешающая функция планирования p' и ассоциированное с p' точное в τ_1 разбиение D' прообраза начальной подсистемы, такие что $\Delta(D',p') < \Delta(D,p)$.

Доказательство леммы об уменьшении дефекта

Пусть dl такой полуинтервал из L(D), что $mon(p(dl)) = sup(p)(L(D))$, а dw - такой полуинтервал из W(D), что $mon(p(dw)) = inf(p)(W(D))$. Если в W(D) находится сразу несколько полуинтервалов c, для которых $mon(p(c)) = inf(p)(W(D))$, то через dw обозначим самый правый из них.

В силу монотонности индикатора $p<dl,dw> = 1$.

Построим функцию планирования p'. Вне полуинтервалов dl и dw положим, что функция p' совпадает с p. Далее рассмотрим три случая.

Первый случай: длина dl меньше длины dw. Разобьем полуинтервал dw на два полуинтервала dw_1 и dw_2. Положим

dw_1 = [inf(dw), sup(dw) – long(dl)) и dw_2 = [sup(dw) – long(dl), sup(dw)).

Положим $p'(dl) = p'(dw_1) = p(dw)$, $p'(dw_2) = p(dl)$.

Второй случай: длина dl равна длине dw. В этом случае разбиение D оставим без изменений. Определим функцию p'. Положим p'(dl) = p(dw) и p'(dw) = p(dl).

Третий случай: длина dl больше длины dw. Разобьем полуинтервал dl на два полуинтервала dl_1 и dl_2. Положим

dl_1 = [inf(dl), inf(dl) + long(dw)) и dl_2 = [inf(dl) + long(dw), sup(dl)).

Положим $p'(dl_1) = p(dw)$, $p'(dl_2) = p'(dw) = p(dl)$.

Очевидно, что во всех рассмотренных случаях функция p' является функцией планирования. Проверим, что она разрешающая.

Заметим, что при $I \neq p(dl)$ и $I \neq p(dw)$ coimg(p)(I) = coimg(p')(I). Поэтому при этих I условие фп6) для p' выполнено.

В соответствии с построением функции p' выполнено неравенство

sup(coimg(p')(p(dw))) < sup(coimg(p)(p(dw)))

Так как условие фп6) выполнено для задания с индексом p(dw) и функция p разрешающая, то $sup(coimg(p)(p(dw))) \leq \tau_0 + T_{p(dw)}$. Но тогда и $sup(coimg(p')(p(dw))) < \tau_0 + T_{p(dw)}$, и, следовательно, фп6) выполнено для задания с индексом dw и функции планирования p'.

Проверим выполнение фп6) для задания с индексом p(dl) и функции p'. Заметим, прежде всего, что в силу выбора dw и выполнения фп6) для p выполнено:

$sup(coimg(p)(p(dw))) = sup(dw) \leq \tau_0 + T_{p(dw)}$.

Так как mon(p(dl))>mon(p(dw)), то $\tau_0 + T_{p(dw)} \leq \tau_0 + T_{p(dl)}$, и, следовательно,

$sup(dw) \leq \tau_0 + T_{p(dl)}$.

Если sup(coimg(p)(p(dl))) < sup(dw), то в соответствии с построением функции p' выполнено равенство sup(coimg(p')(p(dl))) = sup(dw). Поэтому $sup(coimg(p')(p(dl))) \leq \tau_0 + T_{p(dl)}$, и, значит, фп6) выполнено.

Если sup(coimg(p)(p(dl))) > sup(dw),

то sup(coimg(p')(p(dl))) = sup(coimg(p)(p(dl))) $\leq \tau_0 + T_{p(dl)}$, так как фп6) выполнено для p и dl. Поэтому условие фп6) выполнено во всех случаях, а функция p' – разрешающая.

Для окончания доказательства леммы осталось проверить, что в каждом из рассмотренных трех случаев выполнено неравенство $\Delta(D',p') < \Delta(D,p)$.

Первый случай: длина dl меньше длины dw. Разобьем полуинтервал dw на два полуинтервала dw_1 и dw_2. Положим

dw_1 = [inf(dw), sup(dw) – long(dl)) и dw_2 = [sup(dw) – long(dl), sup(dw)).

Положим $p'(dl) = p'(dw_1) = p(dw)$, $p'(dw_2) = p(dl)$.

Обозначим через DW' набор полуинтервалов, равный W(D) – dw + dw_1 + dw_2, и пусть D' = L(D) + DW'. Ясно, что L(D') = L(D), и $W(D') = DW'$. Кроме того, $inf(p)(W(D)) \leq inf(p')(DW')$.

Покажем, что $\Delta(D',p') < \Delta(D,p)$. В силу свойства квазилинейности

$\Delta(D',p') = p'<L(D), DW'> = p'<L(D) - dl, DW'> + p'<dl, DW'>$.

Рассмотрим слагаемое $p'<dl, DW'>$. По построению

$p'(dl) = p(dw) = inf(p)(DW) \leq inf(p')(DW')$. Поэтому $p'<dl, DW'> = 0$.

Далее, $p'<L(D) - dl, DW'> = p'<L(D) - dl, DW'- dw_1 - dw_2> + p'<L(D) - dl, dw_1> + p'<L(D) - dl, dw_2>$.

Заметим, что $DW'- dw_1 - dw_2 = DW – dw$. Так как на $L(D) – dl$ и $DW – dw$ функции p и p' совпадают, то

$p'<L(D) - dl, DW'- dw_1 - dw_2> = p<L(D) - dl, DW- dw>$.

Рассмотрим слагаемое $p'<L(D) - dl, dw_1>$.

Пусть c – произвольный полуинтервал из $L(D) – dl$. Тогда $mon(p'(c)) = mon(p(c))$. Так как $p'(dw_1) = p(dw)$, а на $L(D) – dl$ функции p' и p совпадают, то $p'<L(D) - dl, dw_1> = p<L(D) - dl, dw>$.

Так как $p'(dw_2) = p(dl)$, то $mon(p'(dw_2)) = mon(p(dl)) = sup(p)(L(D)) \geq sup(p)(L(D)- dl) = sup(p')(L(D)- dl)$.

Поэтому $p'<L(D) - dl, dw_2>=0$.

Отсюда получаем, что

$p'<L(D), DW'> = p<L(D) - dl, DW- dw> + p<L(D) - dl, dw> = p<L(D) - dl, DW>$.

Теперь заметим, что

$p<L(D), W(D)> = p<L(D) - dl, DW> + p<dl, DW - dw> + p<dl, dw>$.

Так как $p<dl, dw> =1$, то $\Delta(D,p) = p<L(D), W(D)> > p<L(D) - dl, DW> = \Delta(D',p')$, что и доказывает выполнение неравенства в первом случае.

Второй случай: длина dl равна длине dw. В этом случае разбиение D оставим без изменений. Определим функцию p'. Положим $p'(dl) = p(dw)$ и $p'(dw) = p(dl)$. В соответствии со свойством квазилинейности получаем

$p'<L(D), W(D)> = p'<L(D) - dl, W(D)- dw> + p'<L(D) - dl, dw> + p'<dl, W(D)>$.

Так как на $L(D) – dl$ и $W(D) – dw$ функции p и p' совпадают, то

$p'<L(D) - dl, W(D) - dw> = p<L(D) - dl, W(D) - dw>$.

Далее, $p'(dw) = p(dl) = sup(p)(L(D)) \geq sup(p)(L(D)- dl) = sup(p')(L(D)- dl)$. Поэтому $p'<L(D) - dl, dw> = 0$.

$p'<dl, W(D)> = p'<dl, W(D)-dw> + p'<dl,dw>$.

Так как $p'(dl) = p(dw)$, $p'(dw) = p(dl)$ и $mon(p(dw)) < mon(p(dl))$, то и $mon(p'(dl)) < mon(p'(dw))$, и $p'<dl,dw> = 0$.

Кроме того, $p'(dl) = p(dw) = inf(p)(W(D)) \leq inf(p)(W(D) - dl) = inf(p')(W(D)-dl)$. Поэтому $p'<dl, W(D)-dw> = 0$. Отсюда следует, что $p'<dl, W(D)> = 0$. Таким образом,

$\Delta(D,p') = p'<L(D), W(D)> = p<L(D) - dl, W(D) - dw>$.

Заметим теперь, что $\Delta(D,p) = p<L(D),W(D)> = p<L(D) - dl, W(D) - dw> +$

$p<L(D), W(D) - dw> + p<L(D) - dl, W(D)> + p<dl,dw>$.

Так как $p<dl,dw> = 1$, то

$\Delta(D,p) = p<L(D),W(D)> > p<L(D)-dl,W(D)-dw>=\Delta(D,p')$, то есть и во втором случае требуемое неравенство выполняется.

Третий случай: длина dl больше длины dw. Разобьем полуинтервал dl на два полуинтервала dl_1 и dl_2. Положим

$dl_1 = [inf(dl), inf(dl) + long(dw))$ и $dl_2 = [inf(dl) + long(dw), sup(dl))$.

Положим $p'(dl_1) = p(dw)$, $p'(dl_2) = p'(dw) = p(dl)$.

Обозначим через DL' набор полуинтервалов, равный $L(D) – dl + dl_1 + dl_2$, и пусть $D' = DL' + W(D)$. Ясно, что $L(D') = DL'$ и $W(D') = W(D)$. Кроме того, $sup(p)(L(D)) \geq sup (p')(DL')$.

Покажем, что $\Delta(D',p') < \Delta(D,p)$. В силу свойства квазилинейности

$\Delta(D',p') = p'<DL',W(D)> = p'<DL',W(D)-dw> + p'<DL',dw>$.

Рассмотрим слагаемое $p'< DL',dw>$. По построению

$p'(dw) = p(dl) = sup(p)(L(D)) \geq sup(p')(DL')$. Поэтому $p'<DL',dw> = 0$.

Далее, $p'<DL',W(D)-dw> =$

$p'<DL'–dl_1- dl_2, W(D)-dw> +$

$p'<dl_1, W(D)-dw> +$

$p'<dl_2,W(D)-dw>$.

Заметим, что $DL'–dl_1- dl_2 = L(D) – dl$. Так как на $L(D) – dl$ и $DW – dw$ функции p и p' совпадают, то

$p'<DL'–dl_1-dl_2,W(D)-dw > = p<L(D)-dl,DW-dw>$.

Рассмотрим слагаемое $p'<dl_1,W(D)-dw>$.

Так как $p'(dl_1) = p(dw)$, то $mon(p'(dl_1)) = mon(p(dw)) = inf(p)(W(D)) \leq inf(p)(W(D)-dw) = inf(p')(W(D)-dw)$.

Поэтому $p'<dl_1,W(D)-dw> = 0$.

Пусть c – произвольный полуинтервал из W(D)-dw. Тогда mon(p'(c)) = mon(p(c)). Так как $p'(dl_1) = p(dl)$, а на W(D)-dw функции p' и p совпадают, то $p'<dl_1,W(D)-dw> = p<dl,W(D)-dw>$.

Поэтому $p'<L(D) - dl, dw_2>=0$. Отсюда получаем, что $p'<DL',W(D)-dw> = p<L(D)-dl,W(D)-dw> + p<dl,W(D)-dw> = p<L(D),W(D)-dw>$.

Теперь заметим, что

$p<L(D),W(D)> = p<L(D),W(D)-dw> + p<L(D),DW-dw> + p<L(D), dw>$.

Так как $p<L(D),dw> = p<L(D)- dl,dw> + p<dl, dw>$ и $p<dl, dw> =1$, то

$\Delta(D,p) = p<L(D),W(D)> > p<L(D),W(D)-dw> = \Delta(D',p')$, что доказывает справедливость утверждения леммы в третьем случае и доказывает лемму в целом.

Лемма об уменьшении дефекта доказана.

Следствие *из леммы об уменьшении дефекта.*

Существует полная разрешающая функция планирования p' и ассоциированное с p' точное в τ_1 разбиение D' прообраза начальной подсистемы, такие что $\Delta(D',p') = 0$.

Предположим, что это не так. Тогда существует такая функция p' и такое разбиение D', что число $\Delta(D',p')$ минимально среди всех дефектов и положительно. Согласно лемме существуют такие D'' и p'', что $\Delta(D'',p'') < \Delta(D',p')$. Противоречие. *Следствие из леммы доказано*.

Замечание. Равенство нулю дефекта не зависит от разбиения, ассоциированного с функцией планирования, а определяется только самой функцией планирования.

Действительно, пусть снова p - полная разрешающая функция планирования системы S, и D – ассоциированное с p разбиение прообраза S, точное в точке τ_1.

Пусть set – любое подмножество индексов из IndSet(beg(S)). Обозначим через mon(set) множество, состоящее из чисел mon(I), таких что I принадлежит set. Через inf(mon(set)) обозначим наименьшее из этих чисел, а через sup(mon(set)) – наибольшее.

Обозначим через $set_1(p)$– множество значений ограничения функции p на полуинтервал $[\tau_0,\tau_1)$. Через $set_2(p)$ – обозначим множество значений ограничения функции p на $[\tau_1,\tau_n)$. Ясно, что $\Delta(D,p) = 0$ тогда и только тогда, когда

$\sup(mon(set_1(p))) \le \inf(mon(set_2(p)))$.

Свойства равенства нулю дефекта можно переформулировать совсем без участия разбиения. Пусть c_1 и c_2 два полуинтервала, на которых функция p принимает постоянные отличные от нуля значения, и c_1принадлежит $[\tau_0,\tau_1)$, а c_2 принадлежит $[\tau_1,\tau_n)$.

Дефект функции планирования равен нулю тогда и только тогда, когда для любых двух полуинтервалов c_1 и c_2 выполнено условие $mon(p(c_1))\le mon(p(c_2))$.

Отсюда следует, что равенство нулю дефекта функции планирования не зависит от ассоциированного с функцией разбиения, а зависит только от самой функции. Соответственно факт равенства нулю дефекта функции p будем обозначать через $\Delta(p)=0$.

Пусть p' – полная функция планирования, такая что $set_1(p') = set_1(p)$ и $set_2(p')$ $set_2(p)$. В этом случае $\Delta(p)=0$ тогда и только тогда, когда $\Delta(p')=0$.

Функции планирования с монотонным стартом

Пусть S – связная система, p – полная разрешающая функция планирования этой системы, и $\tau_0,\ldots,\tau_n$ - занумерованные в порядке возрастания критические точки системы S.

Пусть q – монотонная функция планирования системы beg(S).

Определение. Функция планирования p называется функцией планирования с *монотонным стартом*, если на полуинтервале $[\tau_0, \tau_1)$ функции p и q совпадают.

Лемма (*о существовании функции планирования с монотонным стартом*). Пусть S – связная разрешимая система. Тогда существует полная разрешающая функция планирования системы S с монотонным стартом.

Доказательство леммы

Мы предполагаем, что p – полная разрешающая функция планирования системы S. В соответствии с леммой об уменьшении дефекта, можно считать, что $\Delta(p) = 0$.

Пусть q' – монотонная функция планирования системы $lft(S,p)(\tau_1)$. Рассмотрим функцию p', совпадающую с q' на $[\tau_0, \tau_1)$, и с функцией p на $[\tau_1,+\infty)$. Докажем, что функция p' искомая.

Надо показать, что p' – функция планирования, что она разрешающая, что она полная, и что на $[\tau_0, \tau_1)$ эта функция совпадает с монотонной функцией планирования системы beg(S).

Покажем, что p' – функция планирования.

Требование фп1) выполнено очевидно. Проверим фп2). Действительно,

$IndSet(lft(S,p)(\tau_1)) = IndSet(q') = IndSet(lft(S,p')(\tau_1))$,

$IndSet(wrt(S,p)(\tau_1)) = IndSet(wrt(S,p')(\tau_1))$,

$IndSet(S) = IndSet(lft(S,p)(\tau_1)) + IndSet(wrt(S,p)(\tau_1))$.

Поэтому $IndSet(S) = IndSet(lft(S,p')(\tau_1)) + IndSet(wrt(S,p')(\tau_1))$.

Правая часть этого равенства представляет собой множество положительных значений функции p'. Значит фп2) выполнено.

На $[\tau_1, +\infty)$ функция p' совпадает с функцией p. Поэтому для заданий, не начавших исполняться на момент времени τ_1, и заданий, не стартовавших к моменту времени τ_1, выполнение требований фп3) – фп6) для p' следует из выполнения этих требований для p.

Аналогично проверяется выполнение требований фп3) – фп5) для заданий, выполненных на момент времени τ. Пусть (t, L, T, I) – задание, выполненное к моменту времени τ_1. Тогда это задание принадлежит $lft(S,p)(\tau_1)$. Поскольку q – функция планирования системы $lft(S,p)(\tau_1)$, то для этого задания выполнены условия фп3) – фп5).

В соответствии с леммой об ограничениях функции планирования система $lft(S,p)(\tau_1)$ разрешима. Как показано в [4], функция q будет разрешающей. Поэтому для нее выполнено фп6).

Значит условия фп3) – фп6) выполнены и для p'.

Пусть (t, L, T, I) – частично выполненное задание. Тогда

$coimg(p')(I) = coimg(lft(p')(\tau))(I) + coimg(wrt(p')(\tau))(I)$ и

$long(coimg(p')(I)) = long(coimg(lft(p')(\tau))(I)) + long(coimg(wrt(p')(\tau))(I))$.

Но $coimg(lft(p')(\tau))(I) = coimg(lft(q')(\tau))(I)$,

и $coimg(wrt(p')(\tau))(I) = coimg(wrt(p)(\tau))(I)$.

Так как q' и wrt(p)(τ) – функции планирования, то каждое из множеств coimg(lft(q')(τ))(I) и coimg(wrt(p)(τ))(I) представимо в виде объединения конечного количества полуинтервалов. Следовательно, таким же является и множество coimg(p')(I). Поэтому свойство фп3) выполнено.

Также из приведенных выше равенств вытекает, что

$long(coimg(p')(I)) = long(coimg(lft(p)(\tau))(I)) + long(coimg(wrt(p)(\tau))(I))$.

Поэтому coimg(p')(I) = coimg(p)(I) = L. Значит фп4) выполнено.

Пусть (t', L', T', I) – правое ограничение частично выполненного задания (t,L,T,I). Тогда справедливость условия фп5) следует из того, что

$sup(coimg(p)(I)) = sup(coimg(p')(I))$ и $t+T= t'+T'$.

Итак, мы доказали, что p' - разрешающая функция планирования системы S.

Так как функция p полная, то p отлична от нуля на полуинтервале $[\tau_0,\tau_0+\|S\|)$. Поэтому p' отлична от нуля на $[\tau_1,\tau_0+\|S\|)$.

Заметим, что на $[\tau_0, \tau_1)$ функция q, а значит и p', отлична от нуля, поскольку $\tau_1 - \tau_0 = \|lft(S,p)\|$. Поэтому p' отлична от нуля на $[\tau_0,\tau_0+\|S\|)$. Значит p' – полная функция планирования.

Осталось доказать, что на $[\tau_0, \tau_1)$ функция q' совпадает с функцией q – монотонной функцией планирования системы beg(S).

Покажем, что Δ(p') = 0. Пусть D – разбиение, ассоциированное с p. Обозначим через $set_1(p)$ множество значений ограничения функции p на полуинтервале $[\tau_0,\tau_1)$. Через $set_2(p)$ обозначим множество значений ограничения функции p на $[\tau_1,\tau_n)$. Так как Δ(p) = 0, то $sup(mon(set_1(p))) \le inf(mon(set_2(p)))$.

Заметим, что $mon(set_1(p)) = mon(set_1(q)) = mon(set_1(p'))$ и $mon(set_2(p)) = mon(set_2(wrt(p)(\tau))) = mon(set_2(p'))$. Поэтому $sup(mon(set_1(p'))) \le inf(mon(set_2(p')))$. Значит, Δ(p') = 0.

Покажем теперь, что функция q' совпадает с q на $[\tau_0, \tau_1)$. Пусть $h_1,\dots,h_M$, где M=card(beg(S)) - разбиение, ассоциированное с синхронной системой beg(S), и пусть L такое, что $inf(h_L) < \tau_1 < sup(h_L)$.

Пусть k такое, что на $h_1,\dots,h_{k-1}$ значения q' и q совпадают, а на h_k различаются. Обозначим через c_k принадлежащий h_k полуинтервал, на котором эти значения различны. Если k = L, то считаем, что c_k принадлежит полуинтервалу $[inf(h_L), \tau_1)$.

Тогда $q'(c_k) > q(c_k)$. В силу того, что q' – неубывающая функция, ее значения на $[inf(c_k), \tau_1)$

больше, чем $q(c_k)$. Поэтому $long((coimg(p')(k))\cap[\tau_0,\tau_1)) < long(coimg(p)(k))$.

Так как $long(coimg(p'))(k))=long(coimg(p)(k))$, то на $[\tau_1,+\infty)$ найдется полуинтервал c, такой что $p'(c) = q(c_k) < q'(c_k)$. Значит, дефект функции p' отличен от нуля. Противоречие. Таким образом, функция p' на $[\tau_0,\tau_1)$ совпадает с монотонной функцией планирования системы beg(S).

Лемма о существовании функции планирования с монотонным стартом доказана.

Теперь приступим к доказательству леммы о существовании монотонной разрешающей функции планирования для связной системы.

Лемма (*о существовании монотонной разрешающей функции планирования для связной системы*). Пусть S – связная система, обладающая полной разрешающей функцией планирования. Тогда существует монотонная полная разрешающая функция планирования системы S.

Доказательство леммы

Доказательство будем вести с использованием индукции по количеству критических точек системы S. Если у системы одна критическая точка, то утверждение леммы следует из теоремы из [4].

Предположим теперь, что утверждение леммы справедливо для систем с n критическими точками. Докажем, что утверждение будет верным и для систем с n+1 критической точкой. Итак, пусть $\tau_0,\ldots,\tau_n$ занумерованные в порядке возрастания критические точки системы S.

Из леммы о существовании функции планирования с монотонным стартом следует, что для системы S существует полная разрешающая функция планирования p, совпадающая на $[\tau_0,\tau_1)$ с монотонной функцией планирования системы beg(S).

Из леммы об ограничениях функции планирования и о связности и полноте следует, что $wrt(p)(\tau_1)$ будет разрешающей полной функцией планирования связной системы $wrt(S,p)(\tau_1)$.

Система $wrt(S,p)(\tau_1)$ имеет n критических точек. По предположению индукции существует полная разрешающая монотонная функция планирования r системы $wrt(S,p)(\tau_1)$.

Определим функцию p'. Положим, что на $[\tau_0,\tau_1)$ функция p' совпадает с p, а на $[\tau_1,+\infty)$ функция p' совпадает с r. Покажем, что p' – искомая.

Из леммы о модификации правого ограничения функции планирования следует, что p' будет полной и разрешающей.

Надо доказать, что p' – монотонная функция планирования.

По построению на полуинтервале $[\tau_0,\tau_1)$ функция p' совпадает с монотонной функцией системы beg(S). Значит, на этом полуинтервале p' совпадает с монотонной функцией планирования системы S.

Докажем, что p' совпадает с монотонной функцией планирования на $[\tau_1,+\infty)$. Для этого достаточно показать, что при $k = 1,\ldots,n$ выполнено условие:

$beg(wrt(wrt(S,p)(\tau_1),r)(\tau_k)) = beg(wrt(S,p')(\tau_k))$.

Действительно, на $[\tau_k,\tau_{k+1})$ функции p и p' являются монотонными функциями планирования систем $beg(wrt(wrt(S,p)(\tau_1),r)(\tau_k))$ и $beg(wrt(S,p')(\tau_k))$. Соответственно, если эти системы совпадают, то совпадают и их монотонные функции планирования.

Для доказательства совпадения этих систем достаточно показать, что множества заданий, являющихся правыми ограничениями на момент времени τ_k для функций r и p', совпадают.

Пусть (t,L,T,I) такое задание из S, что $t = \tau_0$ и (t',L',T',I) – правое ограничение этого задания на момент времени τ_1,

Покажем, что для такого задания выполнено условие

$L - \sigma(I,p',\tau_k) = L' - \sigma(I,r,\tau_k)$.

Если $t = \tau_0$, то $L' = L - \sigma(I,p',\tau_1)$, и проверяемое условие получает вид

$\sigma(I,p',\tau_k) = \sigma(I,p',\tau_1) + \sigma(I,r,\tau_k)$.

По определению функции σ

$\sigma(I,p',\tau_k) = long(coimg(p')(I))$,

$\sigma(I,p',\tau_1) = long(coimg(p')(I)\cap[\tau_0,\tau_1))$ и

$\sigma(I,r,\tau_k) = long(coimg(wrt(r)(\tau_k))\cap[\tau_1,\tau_k)) = long(coimg(wrt(p')(\tau_k))\cap[\tau_1,\tau_k)) = long(coimg(p')(\tau_k)\cap[\tau_1,\tau_k))$.

Последнее равенство справедливо, поскольку на $[\tau_1,\tau_k)$ функция $wrt(r)(\tau_k)$ совпадает с p'.

Теперь проверяемое условие следует из равенства

$long(coimg(p')(I)) = long(coimg(p')(I)\cap[\tau_0,\tau_1)) + long(coimg(p')(I)\cap[\tau_1,\tau_k))$.

При $L - \sigma(I,p',\tau_k) = L' - \sigma(I,r,\tau_k) = 0$ задания с индексом I не входят в $beg(wrt(S,p')(\tau_k))$ и $beg(wrt(wrt(S,p)(\tau_1),r)(\tau_k))$, соответственно.

Пусть $L - \sigma(I,p',\tau_k) = L' - \sigma(I,r,\tau_k) > 0$

При $t = \tau_0$, $t' = \tau_1$, $L' = L - \sigma(I,p',\tau_1)$ и $T' = T - (\tau_1-\tau_0)$. Поэтому правые ограничения заданий (t,L,T,I) и (t',L',T',I) на момент времени τ_k имеют вид

$(\tau_k, L' - \sigma(I,r,\tau_k), T- (\tau_k - \tau_0), I)$ и $(\tau_k, L - \sigma(I,p',\tau_k), T- ((\tau_k - \tau_1)-(\tau_1 - \tau_0), I)$,

и, следовательно, совпадают.

Пусть $t > \tau_0$, тогда задание (t,L,T,I) из S входит и в $wrt(S,p)(\tau_1)$.

Покажем, что $\sigma(I,p',\tau_k) = \sigma(I,r,\tau_k)$. Это следует из равенств

$\sigma(I,p',\tau_k) = long(coimg(p')(I)\cap[\tau_0,\tau_k)) = long(coimg(p')(I)\cap[\tau_1,\tau_k)) = \sigma(I,r,\tau_k)$.

Отсюда следует, что $L - \sigma(I,p',\tau_k) = L - \sigma(I,r,\tau_k)$.

Если $L - \sigma(I,p',\tau_k) = L - \sigma(I,r,\tau_k) = 0$, то задания с индексом I не входят ни в $beg(wrt(wrt(S,p)(\tau_1),r)(\tau_k))$, ни в $beg(wrt(S,p')(\tau_k))$.

Если $L - \sigma(I,p',\tau_k) = L - \sigma(I,r,\tau_k) > 0$, то правые ограничения этого задания на момент времени τ_k в системах с планированием (S,p') и $(wrt(S,p)(\tau_1),r)$ имеют вид

$(\tau_k, L - \sigma(I,p',\tau_k), T-(t-\tau_k), I)$ и $(\tau_k, L - \sigma(I,r,\tau_k), T-(t-\tau_k), I)$ и поэтому совпадают.

Таким образом, совпадение систем $beg(wrt(wrt(S,p)(\tau_1),r)(\tau_k))$ и $beg(wrt(S,p')(\tau_k))$ доказано.

Лемма о существовании монотонной разрешающей функции планирования для связной системы доказана.

Существование разрешающей монотонной функции планирования для произвольных систем

Для доказательства теоремы о существовании разрешающей монотонной функции планирования необходимо воспользоваться определениями и результатами, изложенными в [5]. Напомним их. Прежде всего сформулируем определение *полной подсистемы*.

Пусть S – система и S_1 – подсистема системы S. Подсистема S_1 называется *изолированной слева*, если для любой подсистемы S' принадлежащей $S[start(S),start(S_1))$, носители систем S' и S_1 не пересекаются. Подсистема S_1 также называется изолированной слева, если $start(S_1) = start(S)$.

Подсистема S_1 называется полной, если выполнены следующие три условия:

п1) подсистема S_1 связная;

п2) подсистема S_1 изолирована слева;

п3) подсистема S_1 содержит все задания из S, момент старта которых содержится в носителе S_1.

Как показано в [5], условие п3) может быть переформулировано в следующем виде: подсистема S_1 содержит все задания (t,L,T,I) из S, для которых выполнено неравенство $t < start(S) + long(S)(t)$.

Из определения непосредственно следует, что любая полная подсистема является связной.

В [5] показано, что любая система представима и единственным образом в виде объединения связных подсистем с непересекающимися носителями. Такое представление называется *разложением* системы на полные подсистемы.

Замечание. Далее мы будем использовать нумерацию подсистем, составляющих разложение системы. Всегда будем считать, что нумерация хронологическая. Это означает, что если $S_1,\ldots,S_N$ –разложение, то $start(S_i) < start(S_j)$ при $i<j$.

Далее в [5] рассматривались полные функции планирования. Функция планирования S называется *полной*, если она отлична от нуля на носителе любой полной подсистемы, входящей в разложение системы S.

Это определение полной функции совпадает с ранее введенным в настоящей статье определением в том случае, когда сама система S является полной. Поэтому для полной подсистемы справедливы все доказанные выше утверждения, относящиеся к связным системам.

Пусть $S_1,\ldots, S_N$ – разложение системы S. *Локальной функцией* планирования p_i называется полная функция планирования системы S_i.

Расширением локальной функции планирования называется функция p_i', совпадающая с p_i на полуинтервале $[start(S_i),+\infty)$ и равная нулю на $[start(S), start(S_i))$. Все расширения локальных функций планирования представляют собой функции с одной и той же областью определения [start(S),+∞). Поэтому корректно определена их сумма.

Сужением полной функции планирования p называется функция, определенная на [start(Si),+∞) и совпадающая на этом полуинтервале с p. Для того, чтобы отличать функцию планирования p от функций планирования полных подсистем, функцию p будем называть *глобальной*.

В [5] доказаны следующие утверждения о свойствах локальных функций планирования.

1) Сужение функции планирования является локальной функцией планирования.

2) Глобальная функция планирования является суммой своих сужений.

3) Сумма всех локальных функций планирования является глобальной функцией планирования.

4) Если глобальная функция планирования является разрешающей функцией системы S, то каждое ее сужение p_i будет разрешающей функцией планирования подсистемы S_i.

5) Если $p_1,\ldots, p_N$ – разрешающие функции планирования систем $S_1,\ldots,S_N$ соответственно, то сумма этих функций $p_1+\ldots+ p_N$ будет разрешающей функцией планирования системы S.

Теперь приступим непосредственно к доказательству теоремы о существовании разрешающей монотонной функции планирования.

Доказательство теоремы

Пусть p – полная разрешающая функция планирования системы S и $S_1,\ldots,S_N$ - ее разложение на полные подсистемы. Через p_i' обозначим сужение функции p на подсистему S_i. В силу (1) и (4) все функции p_i' будут разрешающими локальными функциями планирования. Из леммы о существовании монотонной разрешающей функции планирования для связной системы следует, что существует полная разрешающая монотонная функция планирования системы S_i. Обозначим ее через q_i.

Пусть $q = q_1+\ldots+q_N$. В силу (5) функция q будет полной разрешающей функцией планирования системы S. Для завершения доказательства теоремы осталось показать, что функция q будет монотонной. Докажем это.

Заметим, прежде всего, что каждая критическая точка системы S_i при произвольном i является критической точкой системы S и наоборот.

Так же, как в доказательстве леммы о существовании монотонной разрешающей функции планирования для связной системы, достаточно показать, что для любой критической точки τ_{cr} система $beg(wrt(S,q)(\tau_{cr}))$ совпадает с системой $beg(wrt(S_i,q_i)(\tau_{cr}))$.

Для этого достаточно показать, что для любого задания (t,L,T,I), такого, что $t<\tau_{cr}$, выполнено условие $\sigma(I,q,\tau_{cr}) = \sigma(I,q_i,\tau_{cr})$.

Рассмотрим два случая. Пусть сначала задание (t,L,T,I) принадлежит S_i. Так как в этом случае значение функции q на $[start(S),start(S_i))$ равно нулю, то

$$coimg(q)(I)\cap[start(S),\tau_{cr})=coimg(q)(I)\cap[start(S_i),\tau_{cr}).$$

Поэтому $long(coimg(q)(I)\cap[start(S),\tau_{cr})) = long(coimg(q)(I)\cap[start(S_i),\tau_{cr}))$ и, следовательно, $\sigma(I,q,\tau_{cr}) = \sigma(I,q_i,\tau_{cr})$.

Пусть теперь задание (t,L,T,I) принадлежит S_k, где $k<i$. Обозначим $sup(Supp(S_k))$ через t_k. Так как S_k – полная подсистема, q_k – полная функция планирования подсистемы S_k, и на носителе S_k функция q совпадает с q_k, то $\sigma(I,q, t_k) = \sigma(I,q_k, t_k) = L$. Тогда равенство $\sigma(I,q,\tau_{cr}) = \sigma(I,q_i,\tau_{cr})$ следует из того, что $\sigma(I,q, \tau)$ – возрастающая по τ функция с максимальным значением L.

Таким образом, совпадение систем $beg(wrt(S,q)(\tau_{cr}))$ и $beg(wrt(S_i,q_i)(\tau_{cr}))$ доказано.

Теорема о существовании разрешающей монотонной функции планирования доказана.

Заключение

Статья посвящена планированию систем реального времени. Задача планирования состоит в том, чтобы распределить ресурсы процессора так, чтобы каждое задание было выполнено своевременно. Системы, для которых такое распределение ресурсов возможно, называются *разрешимыми*.

Рассматриваются произвольные системы, состоящие из конечного количества заданий. При этом ограничения на моменты возникновения заданий, длительность задания, частоту возникновения заданий и т.д. не накладываются.

Рассматривается класс монотонных функций планирования. Доказывается, что если система разрешима, то существует монотонная разрешающая систему функция планирования. Таким образом, вопрос о разрешимости системы сведен к вопросу о разрешимости системы в классе монотонных функций планирования.

В статье доказательство разрешимости сведено к случаю связных систем и полных функций планирования. Такая редукция осуществлена с помощью техники, развитой в [5].

Доказательство существенно опирается на результаты, полученные в статье [4], в которой доказательство существования разрешающей функции планирования приведено для синхронных систем, то есть систем, все задания которых имеют общий момент старта.

Также в настоящей работе развивается техника ограничений систем и функций планирования, которая может быть полезной при исследовании других вопросов, связанных с планированием систем.

Литература

1. Никифоров В.В., Павлов В.А., Операционные системы реального времени для встроенных программных комплексов.//Программные продукты и системы.- 1999. - №4. – С.24-30
2. Никифоров В.В., Выполнимость приложений реального времени на многоядерных процессорах, СПб, Труды СПИИРАН. 2009, вып. 8, С. 255 - 284.
3. Данилов М.В., Методы планирования выполнения задач в системах реального времени.//Программные продукты и системы. - 2001. - №4. –
4. Грюнталь А.И., Планирование заданий с синхронным стартом.//Программные продукты и системы. - Тверь, Научно-исследовательский институт «Центрпрограммсистем», 2010, № 4,С. 19 - 23
5. Грюнталь А.И., Локализация ресурсов вычислительных систем реального времени.//Информационные технологии и вычислительные системы. - Москва, Институт системного анализа РАН, 2011, № 2, С. 23 - 40
6. Безруков В.Л. [и др.] Введение в ос2000 // Вопросы кибернетики. Информационная безопасность. Операционные системы реального времени. Базы данных [под ред. В.Б. Бетелина]. М.: НСК РАН, 1999. С. 76-106.

Грюнталь Андрей Игоревич. Заведующий отделом математического обеспечения НИИСИ РАН. Окончил МГУ им. М.В.Ломоносова в 1974 году. Автор свыше 30 печатных работ. Область научных интересов: технология разработки приложений, функционирующих в реальном масштабе времени, разработка и применение средств защиты информации в системах реального времени. E-mail: grntl@niisi.msk.ru