

Для получения корректного результата нужно также учитывать вероятности отказа сетевых и/или терминальных узлов, перепады вычислительной нагрузки при возникновении ЛП или предпосылок к ним и т.п. Эти параметры оцениваются в процессе реальной эксплуатации сети и вводятся в функционал (8) для поиска оптимального решения.

Заключение. В данной работе поставлена задача определения факторов влияния характеристик информационно-вычислительной сети, входящей в состав системы ОрВД, на общую эффективность функционирования системы. Для получения корректных решений необходимо вводить дополнительные данные о параметрах и состоянии сети непосредственно в процессе эксплуатации системы ОрВД.

Литература

1. Андрусак А.І. Мережа авіаційного електрозв'язку / А.І. Андрусак, В.С. Дем'янчук, Ю.М. Юр'єв. – К.: НАУ, 2001. – 448 с.
2. Солонин В. Технологическая авиасвязь: новые тенденции / В. Солонин // [Электронный ресурс]. Режим доступа : <http://www.cnews.ru/reviews/free/transport2006/articles/aviacommunication>
3. Якість та ефективність системи організації повітряного руху // [І.С. Биковцев, В.С. Дем'янчук, В.О. Клименкота інш.]. – К.: ДП ОНР, 2010. – 316 с.
4. Рунион Р. Справочник по непараметрической статистике/ Р. Рунион. – М.: Финансы и статистика, 1982. – 198 с.
5. Применение системы ключевых показателей эффективности для оценивания параметров компьютерных сетей [Э.А. Амирханов, С.В. Водопьянов, В.А. Заруцкий, Е.А. Зубарева]// Вісник Державного університету інформаційно-комунікаційних технологій. – 2012. – Т.10, №4. – С. 82 – 87.
6. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач / А.Н. Тихонов, В.Я. Арсенин. – М.: Наука, 1979. – 288 с.

УДК 004.451.4:004.451.7.031.43

Станко П.А., асп. (Национальный авиационный университет)

МЕТОДЫ ОПТИМИЗАЦИИ ДИСЦИПЛИНЫ ОБСЛУЖИВАНИЯ В ОПЕРАЦИОННЫХ СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ

Станко П.О. Методи оптимізації дисципліни обслуговування в операційних системах реального часу. Розглянута задача оптимізації дисципліни обслуговування в операційній системі реального часу, вживаної в системах управління виробництвом і в системах критичного застосування. Проаналізовані затримки реакції залежно від типу моделі ОС РЧ як системи масового обслуговування. Виведені вирази для середнього часу очікування звернень до ядра операційної системи для процесів з різними пріоритетами.

Ключові слова: ОПЕРАЦІЙНА СИСТЕМА РЕАЛЬНОГО ЧАСУ, ДИСЦИПЛІНА ОБСЛУГОВУВАННЯ, ЗАТРИМКА РЕАКЦІЇ, СИСТЕМА МАСОВОГО ОБСЛУГОВУВАННЯ

Станко П.А. Методы оптимизации дисциплины обслуживания в операционных системах реального времени. Рассмотрена задача оптимизации дисциплины обслуживания в операционной системе реального времени, применяемой в системах управления производством и в системах критичного применения. Проанализированы задержки реакции в зависимости от типа модели ОС РВ как системы массового обслуживания. Выведены выражения для среднего времени ожидания обращений к ядру операционной системы для процессов с разными приоритетами.

Ключевые слова: ОПЕРАЦИОННАЯ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ, ДИСЦИПЛИНА ОБСЛУЖИВАНИЯ, ЗАДЕРЖКА РЕАКЦИИ, СИСТЕМА МАСОВОГО ОБСЛУЖИВАНИЯ

Stanko P.A. The methods of optimisation of service disciplines in the real-time operational systems. The problem of optimisation of discipline of service in the real-time operating system is considered, applied in the control systems by production and in the systems of critical application. The delays of reaction depending on the type of model

of real-time operating system as queuing system are analysed. Expressions for mean time of expectation of addresses to the kernel of the operating system for processes with different priorities are shown out.

Keywords: REAL-TIME OPERATIONAL SYSTEM, SERVICE DISCIPLINE, REACTION LATENCY, QUEUING SYSTEM

Введение. Рассмотрим принципиальные отличия операционных систем реального времени (ОС РВ) от операционных систем (ОС) общего назначения.

ОС общего назначения, особенно многопользовательские UNIX-подобные, ориентированы на оптимальное распределение ресурсов компьютера между пользователями и задачами (системы разделения времени). В ОС РВ подобная задача отходит на второй план. Главная задача ОС РВ – своевременная реакция на события, происходящие на объекте управления.

Другое отличие – применение ОС РВ всегда связано с конкретным объектом и с событиями, происходящими на нем. Система реального времени, как аппаратно-программный комплекс, включает в себя датчики, для сбора информации о параметрах и состоянии объекта, модули ввода-вывода, преобразующие показания датчиков в цифровой вид, пригодный для обработки этих показаний на компьютере, и, наконец, компьютеризованную систему управления. ОС РВ ориентирована на обработку внешних событий. Именно это приводит к коренным отличиям (по сравнению с ОС общего назначения) в структуре системы, в функциях ядра, в построении системы ввода-вывода.

Различают системы реального времени двух типов – системы жесткого реального времени и системы мягкого реального времени.

Системы жесткого реального времени не допускают задержек реакции ни при каких условиях. Примеры систем жесткого реального времени – бортовые системы управления, системы аварийной защиты, регистраторы аварийных событий.

Системы мягкого реального времени характеризуются тем, что задержка реакции не критична, хотя и может привести к дополнительному расходу ресурса и снижению производительности системы в целом. Пример – работа сети. Если система не успела обработать очередной принятый пакет, это приведет к таймауту на передающей стороне и повторной посылке. Данные при этом не теряются, но производительность сети снижается.

В дальнейшем под ОС РВ будем понимать такую систему, которая может быть использована для построения систем жесткого реального времени.

Ключевые показатели эффективности систем реального времени (СРВ). ОС РВ должны реагировать на различные типы внутренних и внешних событий (периодических и непериодических). Необходимо отметить, что принадлежность системы к классу СРВ никак не связана с ее быстродействием. Исходные требования к времени реакции системы и другим временным параметрам определяются или техническим заданием на систему, или просто логикой ее функционирования. Интуитивно понятно, что быстродействие СРВ должно быть тем больше, чем больше скорость протекания процессов на объекте контроля и управления.

Ключевым показателем эффективности ОС РВ, который приводят производители систем реального времени, является время реакции системы на прерывание (interrupt latency). В настоящее время нет стандартных методологий измерения этого параметра, хотя и разрабатываются методы сравнения операционных систем реального времени, которые включают в себя и разработку методологии тестирования.

Не менее важным параметром является размер системы исполнения, а именно суммарный размер минимально необходимого для работы приложения системного набора (ядро, системные модули, драйверы и т. д.).

С совершенствованием ОС РВ значение этого параметра уменьшается, тем не менее, он входит в число ключевых параметров, и производители систем реального времени стремятся к тому, чтобы размеры ядра и обслуживающих модулей системы были минимальны.

Возможность исполнения системы на основе постоянного запоминающего устройства (ПЗУ) (Read Only Memory – ROM) также относится к числу ключевых показателей. Такие

ОС РВ являются встроенными, компактными, имеют высокую надёжность и пониженное энергопотребление, поскольку не связаны с внешними накопителями.

Система приоритетов и алгоритмы диспетчеризации. Базовыми инструментами разработки сценария работы системы являются система приоритетов процессов (задач) и алгоритмы планирования (диспетчеризации) в ОС РВ.

В многозадачных ОС общего назначения используются, как правило, различные модификации алгоритма круговой диспетчеризации, основанные на понятии непрерывного кванта времени ("time slice"), предоставляемого процессу для работы. Планировщик по истечении каждого кванта времени просматривает очередь активных процессов и принимает решение, кому передать управление, основываясь на приоритетах процессов. Приоритеты могут быть фиксированными или меняться со временем – это зависит от алгоритмов планирования в данной ОС.

В чистом виде алгоритмы круговой диспетчеризации в ОС РВ неприменимы. Основное препятствие – неделимый квант времени, в течение которого процессор занят только одним процессом. Планировщики же ОС РВ имеют возможность сменить процесс до истечения "time slice", если в этом возникла необходимость. Один из возможных алгоритмов планирования при этом – "приоритетный с вытеснением".

Другой набор механизмов реального времени относится к средствам синхронизации процессов и передачи данных между ними. Для ОСРВ характерна развитость этих механизмов. В каждой ОС РВ есть, по крайней мере, один быстрый механизм передачи данных от процесса к процессу.

Для систем с жестким временным регламентом необходимы такие инструменты, как средства работы с таймерами, поэтому развитость средств работы с таймерами – необходимый атрибут ОС РВ.

Наконец, одним из важных показателей функциональности ОС РВ является степень масштабируемости. На базе этих ОС можно строить как компактные системы реального времени, так и большие системы серверного класса.

Здесь перечислены только базовые, обязательные показатели, которыми характеризуются ОС РВ. Для более детального рассмотрения возможностей ОС РВ представлены ориентировочные цифры, дающее представление о порядке времен реакции и подходящих операционных системах (см. табл. 1). Данная таблица сформирована на основании экспериментальных данных, по которым можно сделать выводы об уровне реакции на внешние события различных ОС РВ [].

Характеристики ОС		Табл. 1
Время реакции	Используемые ОС	
Менее 10 мкс	Только ОС РВ, но даже они могут не удовлетворять поставленным требованиям – необходим выбор между схемным и программным решениями	
10...100 мкс	Операционные системы реального времени (QNX и др.)	
100 мкс ... 1 мс	ОС РВ RTAI, RT LINUX, расширения реального времени для Windows	
1 мс	Можно применять ОС Linux и Windows, но не для систем критичного применения, где задержки реакции могут привести к тяжелым последствиям	

Рассмотрим более подробно ОС РВ QNX – первую ОС, построенную на принципах микроядра и обмена сообщениями [2]. Система реализована в виде совокупности независимых (но взаимодействующих через обмен сообщениями) процессов различного уровня (менеджеры и драйверы), каждый из которых реализует определенный вид сервиса.

При таком подходе удается добиться нескольких важнейших преимуществ:

– предсказуемость, означающую ее применимость к задачам жесткого реального времени; Ни одна версия UNIX не может достичь подобного качества, поскольку код ядра

слишком велик. Любой системный вызов из обработчика прерывания в UNIX может привести к непредсказуемой задержке (как и Windows);

– масштабируемость и эффективность, достигаемые оптимальным использованием ресурсов и означающие ее применимость для встроенных систем;

– расширяемость и надежность одновременно, поскольку написанный драйвер не нужно компилировать в ядро, рискуя вызвать нестабильность системы.

QNX является ОС РВ на основе микроядра размером около 10 Кб. В качестве основного средства взаимодействия между процессами система использует передачу сообщений. Благодаря этому в 32-битовой среде возможно взаимодействие процессов с 32- и 16-битовыми кодами, причем сообщения передаются между любыми процессами, независимо от того, находятся ли процессы на одном компьютере или на разных узлах сети. Сетевой протокол является полностью открытым, вплоть до структуры. Он содержит пакеты, которые применяются и для передачи сообщений. Сетевой администратор распознает эти пакеты и переправляет микроядру, которое, в свою очередь, переправляет их в шину локальных сообщений. QNX распознает не только пакеты сообщений QNX-процессов, но и протокола TCP/IP и др.

Для ОС РВ, работающих в составе систем критичного применения, весьма важной составляющей является дисциплина обслуживания (диспетчеризации). Основным направлением сравнительного анализа различных дисциплин обслуживания является расчет среднего времени ожидания требования в очереди или среднего времени пребывания в системе.

Предположим, что требования принадлежат одному из P различных приоритетных классов [3], обозначаемых индексом $p=1,2,3...P$. Каждому требованию, находящемуся в системе в момент времени t ставится в соответствие значение некоторой приоритетной функции $q_p(t)$. Чем больше значение этой функции, тем выше приоритет требования. Всякий раз, когда принимается решение для выбора требования на обслуживание, выбор делается в пользу требования с наибольшим значением приоритетной функции. В простейшем случае в качестве приоритетной функции выбирается просто значение p . В этом случае приоритет требования тем больше, чем больший номер класса принадлежности оно имеет.

Рассмотрим модель типа $M/G/1$. Предположим, что требования из приоритетного класса p образуют поток со средней интенсивностью λ_p требований в секунду. Время обслуживания каждого требования из этого класса выбирается независимо в соответствии с распределением с плотностью вероятности $w_p(x)$ со средним значением

$$\bar{x}_p = \int_0^{\infty} x w_p(x) dx.$$

Введем следующие определения:

$$\lambda = \sum_{p=1}^P \lambda_p, \quad \bar{x} = \sum_{p=1}^P \frac{\lambda_p}{\lambda} \bar{x}_p, \quad \rho_p = \lambda_p \bar{x}_p, \quad \rho = \lambda \bar{x} = \sum_{p=1}^P \rho_p.$$

Здесь ρ – доля времени, в течение которого сервер занят ($\rho < 1$), а каждый из парциальных коэффициентов ρ_p – доля времени, в течение которого сервер занят обслуживанием заявок из приоритетного класса с номером p .

Если требование в процессе обслуживания может быть удалено из сервера и возвращено в очередь при поступлении требования с более высоким приоритетом, то говорят, что система работает с абсолютным приоритетом, если обслуживание любого требования, находящегося в сервере не может быть прервано, то говорят что ОС РВ работает с относительным приоритетом.

Обозначим через τ_p – среднее время ожидания в очереди требований из приоритетного класса p , а через $T_p = \tau_p + \bar{x}_p$ – среднее время пребывания в системе для требований этого класса.

Рассмотрим ОС РВ с относительным приоритетом некоторого маркированного требования из приоритетного класса p . Первая составляющая времени ожидания для маркированного требования связана с требованием, которое оно застает в сервере. Вторая составляющая времени ожидания для маркированного требования определяется тем, что перед маркированным требованием обслуживаются другие требования, которые маркированное требование застало в очереди. Обозначим число требований из класса i , которое застало в очереди маркированное требование (из класса p) и которые обслуживаются перед ним, через N_{ip} . Среднее значение N_{ip} есть среднее значение составляющей задержки $\sum_{i=1}^P \bar{x}_i \bar{N}_{ip}$. Третья составляющая задержки связана с требованиями, поступившими после того, как пришло маркированное требование, однако обслуженными раньше его. Число таких требований обозначим M_{ip} . Среднее значение этой составляющей задержки находится аналогично и составляет $\sum_{i=1}^P \bar{x}_i \bar{M}_{ip}$.

С учетом этих составляющих запишем выражение для среднего времени ожидания в очереди для маркированного требования:

$$\tau_p = \tau_0 + \sum_{i=1}^P \bar{x}_i (\bar{N}_{ip} + \bar{M}_{ip}), \quad p = 1, 2, \dots, P. \quad (1)$$

Очевидно, что независимо от дисциплины обслуживания число требований, N_{ip} и M_{ip} в системе не может быть произвольным, поэтому существует некоторый набор соотношений, связывающий между собой задержки для каждого из приоритетного класса, которые назовем условиями сохранения. Основой условий сохранения для задержек является тот факт, что незаконченная работа в любой ОС РВ в течение любого интервала времени занятости не зависит от порядка обслуживания, если система является консервативной (требования не исчезают внутри системы и сервер не простаивает при непустой очереди).

Для ОС РВ типа $M/G/1$ можно показать, что для любой дисциплины обслуживания должно выполняться следующее важное равенство:

$$\sum_{p=1}^P \rho_p \tau_p = \begin{cases} \frac{\rho \tau_0}{1 - \rho}, & \rho < 1, \\ \infty, & \rho \geq 1. \end{cases}$$

Это равенство означает, что взвешенная сумма времен ожидания не изменяется, независимо от выбора дисциплины обслуживания. Если удастся сократить задержку для одних требований, то она немедленно возрастет для других.

Для более общей системы с произвольным распределением времени поступления требований $G/G/1$ закон сохранения может быть записан в виде [4]:

$$\sum_{p=1}^P \rho_p \tau_p = \bar{U} - \tau_0.$$

Общий смысл этого соотношения заключается в том, что взвешенная сумма времен задержки остается постоянной.

Рассмотрим среднее время ожидания для ОС РВ с обслуживанием в порядке приоритета, задаваемого приоритетной функцией $q_p(t) = p$.

Так как маркированное требование будет находиться в очереди в среднем τ_p секунд, то число таких требований будет равно $\bar{M}_{ip} = \lambda_i \tau_p$.

Из формулы (1) получаем:

$$\tau_p = \tau_0 + \sum_{i=p}^P \bar{x}_i \lambda_i \tau_i + \sum_{i=p+1}^P \bar{x}_i \lambda_i \tau_p, \quad \tau_p = \frac{\tau_0 + \sum_{i=p}^P \rho_i \tau_i}{1 - \sum_{i=p+1}^P \rho_i}. \quad (2)$$

Уравнения решаются рекуррентно, начиная с $\tau_1, \tau_{s_2}, \dots$

$$\tau_p = \frac{\tau_0}{(1 - \sigma_p)(1 - \sigma_p)}; \quad \sigma_p = \sum_{i=p}^P \rho_i. \quad (3)$$

По полученным выражениям (1 – 3) можно рассчитывать характеристики качества обслуживания для всех приоритетных классов и оптимизировать дисциплину обслуживания ОС РВ.

Заключение. В данной работе представлены результаты анализа возможных дисциплин обслуживания операционных систем реального времени, применяемых в системах управления производственными и технологическими процессами. Используя полученные соотношения для характеристик качества обслуживания, можно выбирать дисциплину обслуживания ОС РВ в зависимости от состояния управляемого объекта и вида решаемой задачи.

Литература

1. Кертен Р. Введение в QNX/Neutrino 2: Руководство по программированию приложений реального времени в QNX Real-timePlatform / Р. Кертен. – СПб.: Петрополис, 2001. – 514 с.
2. Зыль С.Н. Операционная система реального времени QNX: от теории к практике / С.Н. Зыль. – [2-е изд.]. – СПб.: БХВ-Петербург, 2004. – 192 с.
3. Очереди и точечные процессы [П. Франкен, Д. Кёниг, У. Арндт, Ф. Шмидт]. – К.: Наукова думка, 1984. – 284 с.
4. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания / Б.В. Гнеденко, И.Н. Коваленко. – М.: Наука, 1987. – 336 с.

УДК 004.7.052:004.414.2

Лукашенко В.В., к.т.н. (Национальный авиационный университет)

ОПТИМИЗАЦИЯ ПАРАМЕТРОВ КОММУТАЦИОННЫХ УЗЛОВ В КОМПЬЮТЕРНЫХ СЕТЯХ

Лукашенко В.В. Оптимізація параметрів комутаційних вузлів в комп'ютерних мережах. Розглянута задача оптимального вибору параметрів маршрутизатора по критеріях мінімального об'єму службового трафіку і достовірності таблиць маршрутизації. Виведені вирази для функціонала якості роботи маршрутизатора при суперечливих критеріях ефективності. Розраховані ключові показники ефективності, по яких можна вибрати оптимальні параметри маршрутизаторів, що конфігуруються.

Ключові слова: КОМП'ЮТЕРНА МЕРЕЖА, КОММУТАЦІЙНИЙ ВУЗОЛ, МАРШРУТИЗАТОР, ЦІЛЬОВА ФУНКЦІЯ, БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ

Лукашенко В.В. Оптимизация параметров коммутационных узлов в компьютерных сетях. Рассмотрена задача оптимального выбора параметров маршрутизатора по критериям минимального объема служебного трафика и достоверности таблиц маршрутизации. Выведены выражения для функционала качества работы маршрутизатора при противоречивых критериях эффективности. Рассчитаны ключевые показатели эффективности, по которым можно выбирать оптимальные конфигурируемые параметры маршрутизаторов.

Ключевые слова: КОМПЬЮТЕРНАЯ СЕТЬ, КОММУТАЦИОННЫЙ УЗЕЛ, МАРШРУТИЗАТОР, ЦЕЛЕВАЯ ФУНКЦИЯ, МНОГОКРИТЕРИАЛЬНАЯ ОПТИМИЗАЦИЯ

Lukashenko V.V. Optimization of commutation node parameters in computer networks. The task of optimum choice of parameters of router on the criteria of minimum volume of official traffic and authenticity of routing directories is considered. Expressions for functional of quality of work of router at the contradictory criteria of efficiency are shown out. The key indexes of efficiency are expected, on which it is possible to choose the optimum configured parameters of routers.

Keywords: COMPUTER NETWORK, COMMUTATION NODE, ROUTER, CRITERION FUNCTION, MULTI CRITERIA OPTIMIZATION