УДК 004.4

В.В. Никифоров, В.И. Шкиртиль

ОЦЕНКА ФАКТОРА БЛОКИРОВАНИЯ ЗАДАЧ В СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ НА МНОГОЯДЕРНЫХ ПРОЦЕССОРАХ

Никифоров В.В., Шкиртиль В.И. **Оценка фактора блокирования задач в системах реального времени на многоядерных процессорах.**

Аннотация. Представлен подход к оценке значений времени отклика задач в системах реального времени, реализуемых многоядерными процессорами. Показано, что в системах на многоядерных процессорах возможно возникновение составного блокирования не только при использовании протокола наследования приоритетов, но и в случае использования других известных протоколов доступа к разделяемым ресурсам. Приведен метод оценки значений фактора блокирования для систем с составным блокированием на многоядерных процессорах.

Ключевые слова: многозадачные программные приложения, разделяемые ресурсы, синхронизационные механизмы, многоядерные процессоры, фактор блокирования.

Nikiforov V.V., Shkirtil V.I. **Estimation of blocking factor for tasks in real-time systems with multi-core processors.**

Abstract. An approach is described for estimate of task response time in real-time systems that are implemented by multi-core processors. The fact is shown, that in systems on multi-core processors the composite blocking of tasks is possible not only with use of priority inheritance protocols, but also with use other known protocols of mutual data access. A method is given for estimate of blocking factor for systems with composite blocking of tasks on multi-core processors.

Keywords: multi-task software applications, mutual resources, synchronization mechanisms, multi-core processors, blocking factor.

1. Введение. Методы оценки времени отклика задач в системах реального времени (СРВ) на классических одноядерных процессорах развивались на протяжении нескольких десятилетий [1–6]. Известные обобщения этих методов на системы с многоядерными процессорами относятся в основном к программным комплексам, состоящим из независимых задач — задач, исполнение которых не сопровождается ожиданием действий, осуществляемых другими задачами [7–10]. В настоящей статье представлен подход к оценке выполнимости на многоядерных процессорах программных приложений СРВ с задачами, разделяющими общие информационные ресурсы [4].

Критерием своевременности реализации функций СРВ является выполнение ограничения D_i времени отклика R_i соответствующих задач (то есть, выполнение требования $R_i \le D_i$). Значение времени отклика R_i зависит от принятой дисциплины планирования и исполь-

зуемых протоколов доступа к разделяемым ресурсам. Условимся считать, что используется дисциплина планирования со статически назначаемыми значениями базовых приоритетов, и что базовый приоритет задачи τ_i определяется значением ее индекса (задача τ_i является более приоритетной по отношению к задаче τ_l, если $i < l$).

Составляющими времени отклика R_i задач τ_i для случая систем с многоядерными процессорами являются те же факторы, что и для случая систем с одиночным исполнительным ресурсом — фактор веса C_i, фактор приоритета I_i, фактор блокирования B_i [4]:

$$R_i = C_i + I_i + B_i . \quad (1)$$

При рассмотрении подхода к оценке R_i условимся использовать индекс j для высокоприоритетных задач (задач τ_j с базовым приоритетом выше, чем у τ_i) и индекс l для относительно низкоприоритетных задач.

В случае исполнения на многоядерных процессорах интервал существования очередного задания типа τ_i (интервал очередного исполнения задачи τ_i) состоит из участков двух типов:

а) участки, на которых задание типа τ_i либо владеет ресурсом процессора, либо ждет момента времени, когда низкоприоритетное задание освободит требуемый информационный ресурс),

б) участки, на которых задание типа τ_i ждет момента предоставления одного из ядер процессора.

При исполнении программного приложения на m-ядерном процессоре интервалы существования заданий типа τ_i при $i \le m$ не содержат участков типа *б*). Действительно, в соответствии с дисциплиной обслуживания по приоритетам в любой момент времени одно из ядер процессора готово исполнять либо коды τ_i, либо коды критического интервала задания, унаследовавшего приоритет τ_i на время исполнения блокирующего критического интервала.

Максимально возможная суммарная продолжительность участков типа *а*) для экземпляров задачи τ_i соответствует сумме $C_i + B_i$ значений фактора веса и фактора блокирования этой задачи. Максимально возможная суммарная продолжительность участков типа *б*) соответствует значению фактора приоритета I_i для задачи τ_i.

Как отмечено выше, для задач τ_1, τ_2, . . . , τ_m значение фактора приоритета равно нулю.

2. Подход к оценке времени отклика. Ключевой принцип, определяющий подход к оценке времени отклика задач для систем на многоядерных процессорах состоит в следующем предположении: при $i > m$ для экземпляров задачи τ_i на участках типа *а*) имеющиеся ядра процессора свободны как от исполнения кодов высокоприоритетных задач τ_h, так и от исполнения тех критических интервалов низкоприоритетных задач τ_l, которые блокируют высокоприоритетные τ_h. На этих участках имеющиеся ядра процессора либо обслуживают (наряду с τ_i) низкоприоритетные задачи, либо простаивают.

Каждая из задач τ_h вносит свой вклад в общий объем вычислений, выполняемых в рамках интервала существования экземпляра задачи τ_i Величина этого вклада определяется выражением $(C_h + BI_h(\tau_i))N(i,h)$, где $BI_h(\tau_i)$ — опосредованный фактор блокирования τ_h (максимально возможная продолжительность блокирования высокоприоритетной τ_h со стороны тех задач, приоритет которых ниже приоритета задачи τ_i), $N(i,h)$ — максимально возможное в рамках временного интервала длинной R_i число активизаций задачи τ_h.

При $i > m$ максимально возможный объем вычислений, выполняемых на участках типа *б*) интервала существования $\tau_i^{(k)}$, определяется выражением $\sum_{i<h}(C_h + BI_h(\tau_i))\lceil R_i / T_h \rceil$, где $\lceil x \rceil$ — ближайшее сверху к x целое число. Отсюда следует, что в соответствии со сформулированным выше ключевым принципом (эти вычисления выполняются всеми m ядрами процессора), значение фактора приоритета I_i определяется выражением:

$$I_i = (1/m)\sum_{i<h}(C_h + BI_h(\tau_i))\lceil R_i / T_h \rceil. \qquad (2)$$

Частный случай, когда все значения B_i в выражении (1) и все значения $BI_h(\tau_i)$ в выражении (2) равны нулю, соответствует системам без критических интервалов по доступу к разделяемым ресурсам. Опуская слагаемые B_i и $BI_h(\tau_i)$, подстановкой (2) в (1), получаем метод оценки времени отклика для реализации на многоядерных процессорах программных приложений без разделения информационных ресурсов.

Для систем с разделением информационных ресурсов необходимо располагать методами оценки значений факторов B_i и $BI_h(\tau_i)$. Оценка значений B_i и оценка значений и $BI_h(\tau_i)$ выполняется различно при применении различных протоколов доступа к разделяемым ресурсам [6].

3. Протоколы доступа к разделяемым ресурсам. Протокол доступа определяет специфику исполнения операции запроса ресурса:

– требуемые условия для выдачи задаче разрешения занять запрашиваемый ресурс,

– побочные системные эффекты, возникающие при входе в критический интервал по доступу к предоставляемому ресурсу.

Протокол наследования приоритетов. При использовании простейшего протокола (ПП) условия предоставления запрашиваемого ресурса ограничиваются единственным (обязательным для всех протоколов) требованием: ресурс должен быть свободен, побочных эффектов (кроме обязательного — ресурс переводится в состоянии «занят») при входе в критический интервал не возникает. Использование ПП может привести к инверсии приоритетов, когда ожидание высокоприоритетным заданием ресурса, занятого низкоприоритетным заданием, откладывается из-за действий задач с промежуточными значениями приоритетов. Исключение возможности возникновения инверсии приоритетов обеспечивается использованием протокола наследования приоритетов ПНП. Условия предоставления запрашиваемого ресурса в случае ПНП те же, что и в случае ПП (ресурс должен быть свободен), но при запросе задачей τ_i ресурса, занятого задачей τ_l, реализуется побочный системный эффект: приоритет задачи τ_l, владеющей занятым ресурсом, временно, до ее выхода из критического интервала, повышается до приоритета τ_i (τ_l наследует приоритет τ_i).

Протоколы доступа к ресурсам на основе пороговых приоритетов. Использование ПНП не исключает возможности возникновения составного блокирования высокоприоритетных задач и возможности возникновения ситуаций со взаимным блокированием задач (возникновения тупиков и клинчей) [11]. Для исключения возможности возникновения взаимного и составного блокирования применяются протоколы доступа к ресурсам, использующие *пороговые приоритеты* ресурсов — статически определяемые параметры каждого из разделяемых ресурсов: пороговый приоритет ресурса равен высшему уровню базового приоритета задачи, которая может занять ресурс. Известны две разновидности протоколов такого типа — протокол пороговых

приоритетов (ППП) и протокол превентивного наследования приоритетов (ППНП) [6].

ППП характеризуется тем, что вдобавок к механизму ПНП ставится дополнительное условие предоставления запрашиваемого ресурса: запрашиваемый ресурс предоставляется задаче только тогда, когда ее базовый приоритет выше, чем максимальный из пороговых приоритетов всех ресурсов, уже занятых в текущий момент времени другими задачами.

ППНП характеризуется усилением (относительно ПНП) побочного системного эффекта, возникающего при входе в критический интервал: при предоставлении задаче τ_i затребованного ресурса приоритет τ_i немедленно повышается до значения порогового приоритета захватываемого ресурса; такое повышенное значение приоритета сохраняется за задачей τ_i до освобождения ею предоставленного ресурса.

При исполнении приложений с разделением ресурсов на одноядерных процессорах как ППП, так и ППНП гарантируют невозможность возникновения взаимного блокирования задач при любых конфигурациях приложений. При исполнении на многоядерных процессорах ППП сохраняет это полезное свойство, но ППНП его теряет. Для одноядерных процессоров и ППП и ППНП гарантируют невозможность возникновения составного блокирования. При исполнении на многоядерных процессорах составное блокирование возможно как при использовании ППП, так и при использовании ППНП [11].

4. Вклад составного блокирования в оценку времени отклика задач. В рамках настоящей статьи рассматриваются программные приложения с задачами, отвечающими следующему ограничению.

Ограничение 1. Входящие в состав приложения задачи не содержат пересекающихся критических интервалов по доступу к ресурсам.

Включение опосредованного фактора блокирования $BI_h(\tau_i)$ в формулу (2) расчета фактора приоритета I_i обеспечивает учет возможного наследования приоритета задачи τ_h экземплярами тех задач, приоритет которых ниже приоритета τ_i. В случае возникновения такого наследования приоритета критический интервал низкоприоритетной задачи исполняется на приоритетном уровне τ_h, что дает основание рассматривать исполнение этого критического интервала как части

вычислений для τ_h в рамках участков типа *б*) исполнения очередного экземпляра задачи τ_i.

При блокировании экземпляра высокоприоритетной задачи τ_h со стороны τ_i исполнение блокирующего критического интервала относится к участкам типа *а*) интервала существования экземпляра задачи τ_i.

Оценим максимальную продолжительность интервала блокирования высокоприоритетной задачи τ_h при $h < i$ со стороны низкоприоритетного задания типа τ_l в условиях следующего ограничения.

Ограничение 2. Любая блокируемая задача содержит единственный критический интервал.

Как видно, ограничение 2 является усилением ограничения 1.

Пусть в кодах ряда низкоприоритетных задач имеются критические интервалы, способные блокировать τ_h при выполнении запроса ресурса g и $C(\tau_l, g)$ — длина критического интервала по доступу к ресурсу g в коде задачи τ_l. $C(\tau_l, g) = 0$ при отсутствии доступа к g в коде τ_l.

Состав критических интервалов, определяющих значение $BI_h(\tau_i)$, зависит от используемого протокола доступа к ресурсам. В случае использования ПНП при запросе ресурса g высокоприоритетным заданием типа τ_h это задание может блокироваться только критическими интервалами низкоприоритетных задач, выполняющими доступ именно к тому самому ресурсу g, который затребован со стороны высокоприоритетной τ_h. Следовательно, в случае использования ПНП максимально возможное значение опосредованного фактора блокирования $BI_h(\tau_i)$ составляет

$$BI_h(\tau_i) = \max\{C(\tau_l, g) \mid l > i\}, \qquad (3)$$

где g — ресурс, используемый при исполнении задачи τ_h.

При использовании ППП состав блокирующих критических интервалов для τ_h может оказаться более широким, чем при использовании ПНП. В случае использования ППП экземпляр высокоприоритетной τ_h блокируется не только критическими интервалами по доступу к ресурсу g, но и критическими интервалами задач τ_l по доступу к любому ресурсу, пороговый приоритет которого не ниже, чем базовый

приоритет τ_h. Формально оценка значений опосредованного фактора блокирования при использовании ППП представляется выражением

$$BI_h(\tau_i) = \max\{C(\tau_l, g') \mid l > i, \pi(g') \le h\}, \quad (4)$$

где g' — любой из используемых низкоприоритетными задачами ресурсов, чей пороговый приоритет $\pi(g')$ не ниже, чем базовый приоритет задачи τ_h.

Максимально возможное значение фактора блокирования B_i определяется аналогично:

– отбираются те критические интервалы в кодах низкоприоритетных задач τ_l, которые способны блокировать τ_i,

– значение фактора блокирования B_i определяется как наибольшая из длин $C_l(g)$ отобранных критических интервалов.

При использовании ПНП значение фактора блокирования B_i представляется выражением

$$B_i = \max\{C(\tau_l, g) \mid l > i\}, \quad (5)$$

где g — ресурс, используемый при исполнении задачи τ_i.

При использовании ПНП для оценки времени отклика задач, отвечающих ограничению 2, значение (3) подставляется в (2), затем значения (2) и (5) подставляются в (1).

В случае использования ППП состав критических интервалов, блокирующих τ_i, может оказаться шире, чем при использовании ПНП. Причины этого различия аналогичны тем, которые отмечены для значений опосредованного фактора блокирования $BI_h(\tau_i)$. Значение фактора блокирования B_i при использовании ППП представляется выражением

$$B_i = \max\{C(\tau_l, g') \mid l > i, \pi(g') \le i\}, \quad (6)$$

где g' — любой из ресурсов, чей пороговый приоритет не ниже, чем базовый приоритет задачи τ_i. То есть, как это имеет место и для $BI_j(\tau_i)$, в общем случае значение B_i при использовании ППП не ниже, а в ряде случаев выше, чем при использовании ПНП.

При использовании ППП для оценки времени отклика задач, отвечающих ограничению 2, значение (4) подставляется в (2), затем значения (2) и (6) подставляются в (1).

Составное блокирование. Снимем ограничение 2: пусть код любой из задач может содержать несколько непересекающихся критиче-

ских интервалов. Тогда возможно возникновение составного блокирования — в ходе исполнения задачи она блокируется неоднократно при запросах доступа к различным ресурсам.

На рис. 1*а* средствами маршрутных сетей [12] изображено приложение с тремя задачами. Задача τ_1 содержит два непересекающиеся критических интервала по доступу к ресурсам g_1 и g_2. Вес каждого из этих интервалов равен 2. Момент первой активизации каждой задачи задается ее фазой $\varphi_1 = 4$, $\varphi_2 = 2$, $\varphi_3 = 0$.

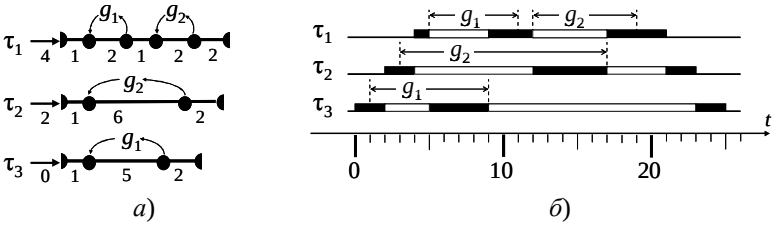

Рис. 1. Составное блокирование при использовании ПНП на одноядерном процессоре: а) конфигурация приложения из трех взаимосвязанных задач; б) порядок предоставления задачам используемых ими ресурсов.

На рис. 1*б* представлен начальный фрагмент диаграммы исполнения приложения $\{\tau_1, \tau_2, \tau_3\}$ на одноядерном процессоре с использованием протокола ПНП. В момент $t = 1$ задача τ_3 захватывает ресурс g_1, в момент $t = 3$ задача τ_2 захватывает ресурс g_2. В момент $t = 5$ задача τ_1 блокируется запросом занятого ресурса g_1; приоритет τ_1 наследуется задачей τ_3.

В момент $t = 12$ задача τ_1 вновь блокируется в связи с запросом ресурса g_2, занятого задачей τ_2. Таким образом, рис. 1 демонстрирует возможность составного блокирования при исполнении взаимосвязанных задач на одноядерных процессорах с использованием ПНП. При исполнении взаимосвязанных задач на одноядерных процессорах с использованием ППП и ППНП составное блокирование исключается, но в случае исполнения на многоядерных процессорах ППП и ППНП теряют это свойство.

На рис. 2 изображена конфигурация приложения из трех задач и начальный фрагмент диаграммы исполнения этого приложения на двуядерном процессоре с использованием протокола ППП.

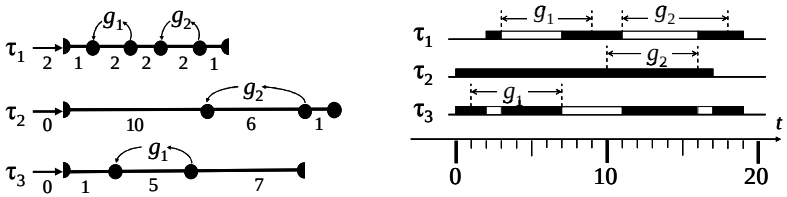

Рис.2. Составное блокирование при использовании ППП на двуядерном процессоре.

В момент $t = 1$ задача τ_3 захватывает ресурс g_1, в момент $t = 3$ задача τ_1 блокируется запросом занятого ресурса g_1 и продолжает вычисления после освобождения g_1 в момент $t = 7$. Все это время другое ядро процессора исполняет коды задачи τ_2, которая в момент $t = 10$ захватывает ресурс g_2. Задача вторично блокируется при запросе доступа к ресурсу g_2 в момент $t = 11$. Таким образом, рис. 2 демонстрирует возможность составного блокирования при исполнении взаимосвязанных задач на двуядерных процессорах с использованием ППП.

С учетом возможности составного блокирования задачи τ_i, в коде которой имеется последовательность критических интервалов по доступу к ресурсам $g_{i,1}$, $g_{i,2}$, ... $g_{i,k}$, ... для оценки значений фактора блокирования B_i при использовании ПНП следует вычислять сумму:

$$B_i = \sum_{1 \le k} \max\{C(\tau_l, g_{i,k}) \mid l > i\}, \qquad (7)$$

где $g_{i,k}$ — ресурс, используемый на $k^{ом}$ критическом интервале задачи τ_i, $C(\tau_l, g_{i,k})$ — максимальная длина критического интервала по доступу к ресурсу $g_{i,k}$ в коде низкоприоритетной задачи τ_l. При использовании ППП вместо суммы (7) следует вычислять сумму

$$B_i = \sum_{1 \le x} \max\{C(\tau_l, g') \mid l > i, \pi(g') \le i\}, \qquad (8)$$

где g' — любой из ресурсов, чей пороговый приоритет не ниже, чем базовый приоритет задачи τ_i. Фактически это означает, что для получения значения B_i выражение под знаком суммы следует умножить на число критических интервалов в теле задачи τ_i.

При использовании ПНП оценка значений опосредованного фактора блокирования $BI_j(\tau_i)$ задач с несколькими критическими интервалами представляется выражением

$$BI_h(\tau_i) = \sum_{1 \le x} \max\{C(\tau_l, g_{i,x}) \mid l > i\}, \quad (9)$$

где $g_{i,x}$ — ресурс, используемый на $x^{ом}$ критическом интервале задачи τ_h. При использовании ППП оценка значений опосредованного фактора блокирования $BI_j(\tau_i)$ задач с несколькими критическими интервалами представляется выражением

$$BI_h(\tau_i) = \sum_{1 \le x} \max\{C(\tau_l, g') \mid l > i, \pi(g') \le i\}, \quad (10)$$

где g' — любой из ресурсов, чей пороговый приоритет не ниже, чем базовый приоритет задачи τ_h.

Формулы (7–10) отражают вклад составного блокирования в оценку времени отклика задач для систем, в которых не выполняется ограничение 2. Для получения значений времени отклика R_i в системах, допускающих составное блокирование, формулы (7–10) следует подставлять в формулы (1) и (2) в том же порядке, как для систем без составного блокирования подставляются формулы (3–6).

Рассмотренные методы оценки фактора блокирования применимы к системам, в которых выполняется ограничение 1. В таких системах у входа в каждый критический интервал задача может ждать освобождения одного занятого ресурса. Для систем, в которых не выполняется ограничение 1, наряду с составным блокированием возможно цепное блокирование, при котором у входа в критический интервал задача может ждать освобождения двух и более занятых ресурсов. Оценка вклада цепного блокирования в значение времени отклика задач является предметом дальнейших исследований.

5. Заключение. Для СРВ на базе классических одноядерных процессоров разработаны методы оценки значений времени отклика как для систем независимых задач, так и для систем с задачами, разделяющими общие информационные ресурсы. Известные обобщения этих методов на системы с многоядерными процессорами относятся к программным комплексам, состоящим из независимых задач, исполнение которых не сопровождается ожиданием действий, осуществляемых другими задачами. В настоящей статье представлен подход к оценке выполнимости на многоядерных процессорах таких программных комплексов реального времени, которые состоят из взаимосвязанных

задач, разделяющих общие информационные ресурсы. Показано, что в системах на многоядерных процессорах возможно возникновение составного блокирования не только при использовании протокола наследования приоритетов, но и в случае использования других известных протоколов доступа к разделяемым ресурсам. Для программных приложений реального времени, реализуемых на многоядерных процессорах, представлен метод, позволяющий выполнять оценку значений времени отклика задач в условиях составного блокирования.

Литература

1. *Liu C.L., and Layland J.W.* Scheduling Algorithms for Multiprogramming in Hard Real-Time Environment // Journal of the ACM. 1973. V. 20, no 1. P. 46–61.
2. *A.D.Ferrari*. Real-Time Scheduling Algorithms. // Dr.Dobb's Journal. 1994. no.12. P. 60–66
3. *Никифоров В.В., Павлов В.А.* Операционные системы реального времени для встроенных приложений. // Программные продукты и системы. 1999. №4. С. 24–30.
4. *Никифоров В.В., Данилов М.В.* Статическая обработка спецификаций программных систем реального времени. // Программные продукты и системы. 2000. №4. С. 13–19.
5. *S.K.Dhall, C.L.Liu*. On a Real-Time Scheduling Problem. // Operating Research. 1978. V.26, no.1. P. 127–140.
6. *Грюнталь А.И.* Планирование систем с асинхронным стартом. // Информационные технологии и вычислительные системы. 2012. №1. С. 32–51.
7. *Никифоров В.В., Павлов В.А.* Структурные модели для анализа многозадачных программных систем. // Информационно-измерительные и управляющие системы. 2011. №9. С. 19–29.
8. *S.K.Baruah*. Fairness in Periodic Real-Time Scheduling Algorithms. In Proceedings of 16 IEEE Real-Time Systems Symposium. 1995. P.200–209.
9. *T.Baker*. Multiprocessors EDF and Deadline Monotonic Schedulability Analysis. In Proceedings of 24 IEEE Real-Time Systems Symposium. 2003. P. 120–129.
10. *Никифоров В. В.* Выполнимость приложений реального времени на многоядерных процессорах. // Труды СПИИРАН. 2009. Вып. 8. С. 255–284.
11. *Никифоров В.В., Шкиртиль В.И.* Составное блокирование взаимосвязанных задач в системах на многоядерных процессорах. // Известия ВУЗов. Приборостроение. 2012. №1. С. 25–31.
12. *Никифоров В.В., Шкиртиль В.И.* Маршрутные сети — графический формализм представления структуры программных приложений реального времени // Труды СПИИРАН. 2010. Вып. 14. С. 7–28.

Никифоров Виктор Викентьевич — д.тех.н., проф.; ведущий научный сотрудник лаборатории технологий и систем программирования СПИИРАН. Область научных интересов: программирование систем реального времени, встроенных систем, операционные системы. Число научных публикаций — 110. nik@iias.spb.su; СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)328-0887.

Nikiforov Victor Vikentievitc — Dr.Sci. (Tech.), Prof.; Senior researcher, Laboratory for Software Technology and Systems, SPIIRAS. Research interests: methods for real-time software development, embedded systems, operating systems. The number of publications —110. nik@iias.spb.su; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-0887.

Шкиртиль Вячеслав Иванович — к.тех.н., доц.; заведующий лабораторией технологий и систем программирования СПИИРАН. Область научных интересов: Программное обеспечение систем реального времени. Число научных публикаций — 70. jvatlas@mail.rcom.ru; СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)328-0887.

Shkirtil Viacheslav Ivanovitch — PhD (Tech.), associate professor; Dr.Sci., Head of the Laboratory for Software Technology and Systems, SPIIRAS. Research interests: software development for real-time systems. The number of publications — 70. jvatlas@mail.rcom.ru; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-0887.

Рекомендовано лабораторией автоматизации научных исследований СПИИРАН.
Статья поступила в редакцию 09.03.2013

РЕФЕРАТ

Никифоров В.В., Шкиртиль В.И. **Оценка фактора блокирования задач в системах реального времени на многоядерных процессорах.**

Специфическими проблемами построения систем реального времени (СРВ) являются проблемы обеспечения гарантированной выполнимости — своевременности выполнения функций, возлагаемых на СРВ. Методы проверки гарантий своевременности опираются на оценку времени отклика задач, составляющих программное приложение СРВ. Методы решения таких проблем, ориентированные на СРВ, строящиеся на базе классических одноядерных процессоров, развиваются на протяжении нескольких десятилетий. Известные обобщения этих методов на системы с многоядерными процессорами, относятся к программным комплексам, состоящим из независимых задач — задач, исполнение которых не сопровождается ожиданием действий, осуществляемых другими задачами.

В отличие от независимых задач, взаимосвязанные задачи могут попадать в состояния ожидания, что приводит к увеличению значения времени отклика. Подобное увеличение времени отклика называется фактором блокирования. Значение фактора блокирования для конкретной задачи зависит не только от конфигурации межзадачных интерфейсов, но и от особенностей используемого протокола доступа задач к разделяемым информационным ресурсам.

Наряду с фактором блокирования дополнительный вклад в значение времени отклика задачи вносится фактором приоритета, отражающим приостановку исполнения задачи в связи с тем, что все ядра процессора заняты более приоритетными вычислениями. При этом к разряду более приоритетных вычислений относится исполнение критических участков, наследующих приоритеты блокируемых высокоприоритетных задач.

Известно, что при использовании протокола наследования приоритетов одноядерными процессорами возможно составное блокирование задач, когда ход исполнения высокоприоритетной задачи блокируется неоднократно. Анализ показывает, в СРВ на многоядерных процессорах составное блокирование возможно и при использовании протокола пороговых приоритетов. В статье представлены методы оценки времени отклика задач с учетом значений фактора блокирования и фактора приоритета для программных приложений СРВ, исполняемых многоядерными процессорами с использованием различных протоколов доступа к разделяемым информационным ресурсам.

SUMMARY

Nikiforov V.V., Shkirtil V.I. **Estimation of blocking factor for tasks in real-time systems with multi-core processors.**

The requirements of well-timed executions are indispensable characteristic of real-time systems. Verification methods for improving correctness of multi-task real-time software from this point of view are based on estimation of task response time. The methods of such estimation were primary developed for the systems with classical single-core processor. The known generalizations of these methods to the systems with multi-core processors are restricted by real-time software applications with independent tasks — such tasks that don't wait some actions from other tasks. For tightly cooperating tasks such waiting states entail increasing response time of waiting task. This increasing is named blocking factor. In the case of multi-core processors estimation of blocking factor shall be done in a specific manner. The blocking factor value is depended not only on the structure of intertask relations, but also on the protocol of access to mutual resources.

There is another factor that increases the task responce time: the priority factor represents delay of the task when all processor cores are busy by execution of higher priority codes. The set of higher priority codes includes the codes of critical intervals that inherit priority of blocked tasks.

Priority inheritance protocol permits compound blocking of task, when high priority task execution is blocked several times. Our analysis shows that the priority ceiling protocol also permits compound blocking of task. The paper describes methods for estimation of task respounce time on multicore processors with taking into account the priority factor and the blocking factor in the cases of using the priority inheritance protocol or priority ceiling protocol.