합성 이용율을 이용한 혼합 태스크 스케줄링

문석환* · 김인국**

Mixed Task Scheduling Using Synthetic Utilization

Seok-Hwan Moon* · In-Guk Kim**

요 약

시간 복잡도가 $O(1)$인 실시간 비주기 태스크 스케줄링 방법 중 하나인 합성 이용율 방법은 주기 태스크들을 고려하지 않고 단지 비주기 태스크들을 위한 스케줄링 방식이다. 하지만 실제로 비주기 태스크는 대부분의 경우에 주기 태스크와의 혼합된 형태로 스케줄링이 이루어지며, 주기 태스크의 스케줄링을 보장하면서 비주기 태스크의 스케줄링 가능성을 판단해야 한다. 본 논문에서는 주기태스크와 비주기 태스크가 혼합된 태스크 집합을 합성 이용율을 이용하여 스케줄링하기 위한 방법을 제시하였으며 기존의 비주기 서버를 이용하여 혼합된 형태의 태스크 집합을 스케줄링 하는 방법에 비해 최대 20%의 스케줄링 성능이 향상됨을 보였다.

ABSTRACT

$O(1)$ time synthetic utilization is not considered periodic tasks, except scheduling methods for aperiodic tasks where one of the aperiodic tasks is a scheduling method. But really aperiodic tasks scheduling method is composed of mixed task types. Aperiodic task scheduling method guarantee an analysis of the schedualibility of aperiodic task. The set of mixed tasks periodic and aperiodic tasks scheduling method uses synthetic utilization that is presented in this paper. The new method shows that schedulability increases 20% aperiodic server method.

키워드

실시간 스케줄링, 비주기 태스크, 합성 이용율

Key word

Real-Time Scheduling, Aperiodic Tasks, Synthetic Utilization

* 영동대학교 임베디드소프트웨어학과
(교신저자, shmoon@youngdong.ac.kr)
** 단국대학교 컴퓨터과학과

접수일자 : 2010. 05. 24
심사완료일자 : 2010. 08. 11

Ⅰ. 서 론

태스크의 수행결과의 정확성뿐만 아니라 처리시한의 제한인 종료시한(deadline)을 지키는 것을 요구하는 시스템을 실시간 시스템(real-time system)이라 한다. 이러한 실시간 시스템은 종료시한이 엄격하게 지켜져야 하는 경성 실시간(hard real-time) 시스템과 그렇지 않은 연성 실시간(soft real-time) 시스템으로 구분 할 수 있다. 또한 일정 시간마다 도착하여 반복적으로 실행되는 주기(periodic) 태스크와 도착 시간이 정해져 있지 않은 비주기(aperiodic) 태스크로 구분할 수 있다. 이제까지 제안된 스케줄링 알고리즘들 중에서 Rate Monotonic(RM)[1] 스케줄링 알고리즘은 고정 우선순위 기반 알고리즘들 중 최적이고, Earliest Deadline First(EDF)[1,2] 스케줄링 알고리즘은 동적 우선순위 알고리즘들 중 최적인데 이들은 모두 주기적 태스크 모델을 기본으로 하고 있다.

최근에 제시된 프로세서 이용율을 이용한 합성 이용율(synthetic utilization) 알고리즘은[3,4,5] 비주기 태스크로만 구성된 태스크 집합들을 스케줄링 하는 방법으로 제시되었으며, [6]에서는 개선된 합성 이용율을 통해 비주기 태스크들의 스케줄링 가능성을 증가시키는 방법이 연구되었다.

주기와 비주기 태스크가 혼합되어 스케줄링되는 경우에는 주기 태스크의 스케줄링 가능성을 우선 보장하면서 비주기 서버를 이용하여 비주기 태스크들을 스케줄링한다. 비주기 태스크의 수행은 비주기 서버의 대역폭에 의해 제한되기 때문에 주기 태스크의 스케줄링에 영향을 주지 않으면서 비주기 태스크들을 스케줄링 할 수 있다. 이러한 방법을 사용하는 알고리즘으로 가장 간단한 구현 복잡도를 가지는 대역폭 서버(total bandwidth server, TBS)[7] 알고리즘이 있다. 본 논문에서는 경성 실시간 시스템에서 EDF 스케줄링에 의해 비주기 태스크들만을 스케줄링 하는 합성 이용율 방법을 확장하여 주기와 비주기 태스크가 혼합된 경우의 태스크 스케줄링 방법을 제시하고, TBS 알고리즘과 비교 분석하였다.

본 논문은 2장에서 비주기 태스크만을 대상으로 하는 합성 이용율을 기술한 후 이것을 확장하여 주기태스크와 비주기 태스크가 혼합된 태스크 집합들을 스케줄링 하는 방법을 제시한다. 3장에서는 모의실험 결과를 보이고, 마지막으로 4장에서는 결론 및 향후 연구 과제에 대하여 기술한다.

Ⅱ. 합성 이용율

2.1 합성 이용율

Abdelzaher등에 의해 제시된 합성 이용율 방법[8]은 경성 비주기 태스크 집합에 대하여 스케줄링 가능한 합성 이용율의 상한을 제시하였다. 또한 이를 이용하여 비주기 태스크가 도착했을 때 합성 이용율을 상한과 비교하여 도착한 비주기 태스크를 수락할 것인지, 거절할 것인지를 결정하게 된다.

비주기 태스크 집합 T_a를 $\{T_1, T_2, T_3 \ldots T_i \ldots\}$라 할 때, T_1은 T_2보다 먼저 도착(arrival)된 태스크라 가정한다. 즉 T_i는 T_{i+1}보다 먼저 도착한 비주기 태스크이다. 비주기 태스크 T_i의 수행시간(execution time)을 $C_i(>0)$, 도착시간(arrival time)을 A_i, 상대적 종료시한(relative deadline)을 $D_i(>0)$ (여기서 절대적 종료시간(absolute deadline) d_i는 $A_i + D_i$로서 정의됨.)라 할 때 임의의 시점 t에서의 합성 이용율은 다음과 같이 정의된다.

$$U_{(t)} = \sum_{T_i \in S_{(t)}} \frac{C_i}{D_i} \quad (1)$$

식(1)에서 $S_{(t)}$는 임의의 시점 t를 기준으로 t이전에 도착한 태스크 중 아직 종료시한이 지나지 않은 태스크들의 집합을 나타내며 $S_{(t)} = \{T_i | A_i \le t < A_i + D_i\}$ 로서 표현된다.

[8]에서는 DM(deadline-monotonic) 스케줄링 방법을 이용하여 임의의 시점 t에서의 합성 이용율 $U_{(t)} = \sum_{T_i \in S_{(t)}} \frac{C_i}{D_i}$이 상한인 0.59를 넘지 않으면 비주기 태스크가 스케줄링 가능하다는 것을 증명하였다. 그리고 EDF 스케줄링 방법을 이용하게 되면 합성 이용율 $U_{(t)} = \sum_{T_i \in S_{(t)}} \frac{C_i}{D_i}$의 상한이 1임을 증명하였다.

또한 합성 이용율을 이용한 스케줄링 분석은 $O(1)$ 이므로 수락제어에 적합하다. 본 논문에서는 상한이 1인 EDF 스케줄링에 의한 합성 이용율을 이용한다. 비주기 태스크들의 합성 이용율을 구하기 위해서는 임의의 시점 t에 대한 현재 요청 집합 $S_{(t)}$를 먼저 구해야 한다.

그림 1에서 임의의 시점 t=7에 대한 현재 요청 집합 $S_{(7)} = \{T_1, T_2, T_4\}$이다. (그림 1)에서 T_3가 $S_{(7)}$에 속하지 못하는 이유는 시점 t=7 보다는 먼저 도착 했으나 종료시한이 t=7 이전인 t=5 에 종료되었으므로 $S_{(7)}$에 속하지 않고 합성 이용율을 계산할 때 실행 시간은 포함하지 않게 된다.

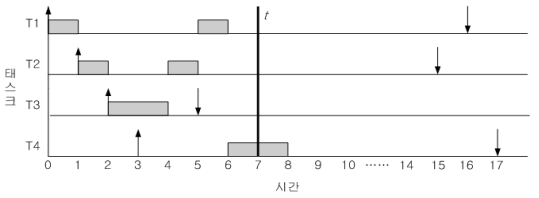

그림 1. t=7에서의 현재 요청 집합
Fig. 1 Current Invocation at t=7

합성 이용율에서는 특정시간 t에서의 이용율을 계산할 때 t이후에 실제로는 스케줄링이 가능하지만, 스케줄링이 불가능하다고 판단되는 문제가 발생하게 된다. [6]에서는 이러한 문제를 개선하여 스케줄링 가능성을 높이는 개선된 합성 이용율 방법을 제시하였다.

2.2 혼합 합성 이용율

기존의 합성 이용율 방법은 비주기 태스크들만을 고려하였기 때문에 주기태스크와 혼합된 태스크 집합에 적용하기 어려웠다. 합성 이용율을 이용하여 주기 태스크를 스케줄링하기 위해서는 주기 태스크를 고려한 구간에서 합성 이용율을 적용할 수 있어야 하고 주기 태스크를 비주기 태스크로 변환해야 한다.

기존의 합성 이용율은 임의시간 t의 시점에서 프로세서의 이용율을 구하는 방법으로 $t+1$과 같은 t이후에 도착되는 태스크들에 대한 고려를 하지 않는다. 이러한 문제를 해결하기 위해 특정구간 $[t, t')$에서의 합성 이용율을 제시하고 이것을 확장하여 주기 태스크에 적용하였다.

주기 태스크의 특성상, 앞으로 발생하는 인스턴스들의 도착을 미리 예측하여 이것을 이용율 계산에 적용할 수 있다면 주기 태스크들에 대해서도 합성 이용율을 이용한 스케줄링 기법을 제시할 수 있다. 즉, 합성 이용율 $U(t) = \sum_{T_i \in S(t)} \frac{C_i}{D_i}$를 주기 태스크에 적용하기 위해서는 임의의 구간에서 합성 이용율을 계산할 수 있어야 한다. 그 이유는 주기 태스크는 매주기마다 인스턴스들이 도착되어 실행되기 때문에 각 주기의 구간 마다 스케줄링 가능성을 보장해주기 위해서이다. 먼저 특정 구간에서 비주기 태스크의 합성 이용율을 구하기 위해 현재 요청집합 $S(t)$를 $S(t,t')$로 확장한다. $S(t,t') = \{T_i | A_i \le t', D_i > t\}$로 정의할 수 있으며 여기서 $t \le t'$이다.

그림 2는 특정구간에서의 현재 요청 집합에 대한 예제이다.

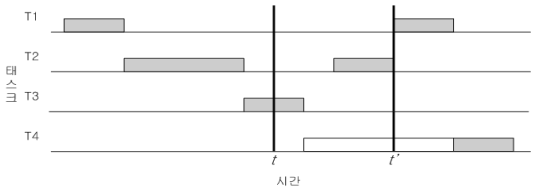

그림 2. $[t,t']$에서의 합성 이용율
Fig. 2 Synthetic Utilization at $[t,t']$

여기서 $S(t) = \{T_1, T_2, T_3\}$이지만 $S(t,t') = \{T_1, T_2, T_3, T_4\}$이다. 이러한 방법으로 특정구간의 현재 요청 집합을 구하여 합성 이용율을 계산할 수 있다. 또한 $S(t,\infty)$를 생각한다면, 시점 t에서 그 이후의 모든 태스크들에 대해서 합성 이용율을 정의할 수 있다.

주기 태스크를 비주기 태스크로 변환하면 합성 이용율을 적용할 수 있는데 주기 태스크 집합 $T_p = \{T_1, T_2, \cdots, T_n\}$에서 주기 태스크의 모든 인스턴스들을 비주기 태스크로 생각할 수 있다. 즉 주기 태스크의 임의의 인스턴스 $T_{i,j}$는 도착시각이 $A_k = A_i + (j-1)T_i$이고, 종료시한이 $d_k = T_i$인 비주기 태스크 T_k로 생각할 수 있다. 이 두 가지 방법을 이용하고, 비주기 태스크 집합 T_a가 EDF에 의해 스케줄링

될 때 합성 이용율 $U(t)=\sum_{T_i\in S(t)}\frac{C_i}{D_i}\le 1$을 이용하여 주기 태스크와 비주기 태스크가 혼합된 태스크 집합에 대해 합성 이용율을 적용할 수 있다.

비주기 태스크 집합을 T_a, 주기 태스크 집합을 T_p (단, $T=D$), 그리고 변환된 주기 태스크 집합을 T_p'이라할 때 T_a와 T_p'의 합집합을 T_s라 하자. 비주기 태스크가 도착할 때 마다 모든 $T_i\in T_s$에 대해 EDF에 의한 방식을 사용하므로 합성 이용율 $U_{Ts}(t,t')\le 1$가 만족되면 비주기 태스크 T_i는 스케줄링 가능하다. 하지만 이 방법은 모든 주기 태스크의 인스턴스에 대해 구간별로 모두 검사해야 하므로 실제로는 불가능하다. 또한 구간 [t, t']에서 현재 시간을 t라 할 때 현재 시간 t부터 t'까지 태스크 정보를 변환된 주기 태스크는 알 수 있지만 비주기 태스크는 전혀 알 수 없기 때문에 구간을 이용한 방법은 실제 불가능하다.

하지만 주기 태스크와 비주기 태스와의 관계를 이용하면 이러한 문제를 해결할 수 있다. 주기 태스크 집합 T_p가 EDF에 의해 스케줄링 될 때 프로세서 이용율은 $U_{Tp}=\sum_{i=1}^{N}\frac{C_i}{T_i}$이다. 또한 변환된 주기 태스크 T_p'의 합성 이용율은 $U_{Tp}'(t,t')=\sum_{T_i\in S(t,t')}\frac{C_i}{D_i}$ 로 표현된다. 이 때 $U_{Tp}'(t,t')\le U_{Tp}$의 관계를 가진다. 즉, EDF 알고리즘에 의해 스케줄되는 원래의 주기 태스크의 프로세서 이용율은 비주기 태스크로서 변환되어 합성 이용율을 적용한 프로세서 이용율 보다 항상 크거나 같다.

다음 표 1은 $U_{Tp}'(t,t')\le U_{Tp}$의 예이다. T_1, T_4는 비주기 태스크로 (A_i, C_i, D_i)가 각각 (1,3,11), (4,2,10) 이고 T_2, T_3은 주기 태스크로 (A_i, C_i, P_i)가 각각 (3,2,7), (5,1,9) 이다.

여기서 P_i는 태스크 T_i의 주기다. 먼저 $U_{Tp(2,4)}'$를 구하면 변환된 주기 태스크 T_2, T_3에 대해 현재 요청 집합은 $S_{(2,4)}=\{T_2\}$이므로 $U_{Tp(2,4)}'=\frac{C_2}{T_2}=\frac{2}{7}=0.28$ 이다.

표 1. 혼합 태스크 집합
Table. 1 Mixed Task Set

	A_i	C_i	$D_i(P_i)$	비고
T_1	1	3	11	비주기
T_2	3	2	7	주기
T_3	5	1	9	주기
T_4	4	2	10	비주기

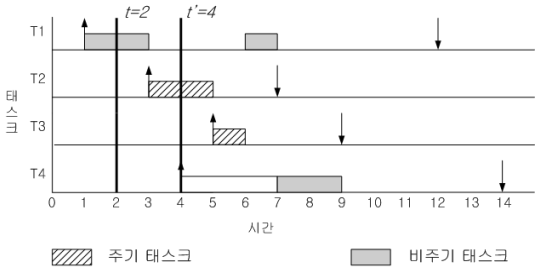

그림 3. 구간 [2,4]에서 주기 태스크의 합성 이용율
Fig. 3 Periodic Tasks Synthetic Utilization at [2,4]

또한 $U_{Tp}=\sum_{i=1}^{N}\frac{C_i}{T_i}$이므로 $U_{Tp}=\frac{2}{7}+\frac{1}{9}=0.39$이다. 그러므로 $U_{Tp}'<U_{Tp}$ 이다. 만일 주기 태스크 T_3이 존재하지 않는다면 $U_{Tp(2,4)}'=U_{Tp}=\frac{2}{7}$이다. 그러므로 $U_{Tp}'=U_{Tp}$이다. 또는 현재 요청 집합 구간이 $S_{(2,6)}$으로 변경되더라도 $U_{Tp}'=U_{Tp}$이다. 즉, U_{Tp}'는 최대 U_{Tp}값을 가질 수 있다. U_{TS}가 EDF로 스케줄링 될 때 $U_{TS}=U_{Ta}+U_{Tp}'$로 표현할 수 있고 U_{Tp}'는 최대 U_{Tp}를 가질 수 있으므로 $U_{TS}=U_{Ta}+U_{Tp}\le 1$이고 $U_{Ta}\le 1-U_{Tp}$의 값을 가진다. 즉, 합성 이용율을 이용한 비주기 태스크가 $1-U_{Tp}$의 범위에서 프로세서 이용율을 가진다면 스케줄링 가능하다. 정리하면 주기 태스크와 비주기 태스크의 혼합 집합 $T_s=T_a+T_p$는 모든 비주기 태스크 $T_i\in Q$에 대해 $U_{TS}\le 1$을 만족하면 스케줄링 가능하다.

그림 3에서처럼 EDF에 의한 주기 태스크의 이용율이 0.39이고, 임의의 시점 t에서 비주기 태스크의 이용율이 1-0.39를 넘지 않으면 주기 태스크 스케줄링을 보장하면서 비주기 태스크들도 스케줄링 가능하다. 이 방법은 앞

에서 제시한 구간을 이용하는 방법처럼 모든 구간을 점검하지 않고 주기 태스크의 이용율을 이용해서 비주기 태스크와 주기 태스크가 혼합된 태스크 집합에 적용할 수 있다. 또한 합성 이용율을 [6]에서 제시한 개선된 합성 이용율 방법을 사용하면 비주기 태스크들의 스케줄링 가능성이 증가하게 된다.

Ⅲ. 모의실험

3.1 모의실험 방법

본 장에서는 주기 태스크와 비주기 태스크가 혼합된 태스크 집합을 TBS 알고리즘과 본 논문에서 제시한 방법을 이용하여 스케줄링 가능성 여부를 비교한다. 모의실험에서 주기 태스크는 EDF 방법을 이용하여 주기 태스크의 이용율이 낮은 0.3과, 이용율이 높은 0.7인 경우로 나누어 실험하였다. 또한 비주기 태스크는 1000개를 랜덤하게 생성하여 주기 태스크와 함께 스케줄링 하였다.

입력 워크로드는 비주기 태스크로서 모든 도착된 태스크들의 수행시간의 합과 태스크들이 도착한 시간구간의 비율로 표현할 수 있는데 실험 범위는 50%~150%까지 10%씩 증가되는 시점에서 태스크들의 스케줄링 가능성을 측정하였다.

3.2 실험 결과

그림 4, 그림 5,는 주기 태스크의 이용율이 0.3, 07인 경우 워크로드에 따른 비주기 태스크들의 스케줄링 가능성 여부를 보여주고 있다. 주기 태스크의 이용율이 높은 경우에, 그리고 워크로드가 큰 경우에는 TBS와 본 논문에서 제시한 방법의 스케줄링 가능성이 큰 차이를 보이지 않지만 주기 태스크의 이용율이 낮은 경우에는 워크로드가 커지더라도 본 논문에서 제시한 방법이 TBS에 의한 알고리즘에 비해 약 20%정도 스케줄링 가능성이 향상된 것으로 나타난다. 그 이유는 TBS 알고리즘의 경우 수행시간이 길고 우선순위가 낮은(종료시한이 긴) 태스크가 먼저 실행되고 수행시간이 짧고 우선순위가 높은(종료시한이 짧은) 태스크가 나중에 도착하여 실행되면 높은 우선순위의 태스크들이 거절당하는 경우가 발생하기 때문이다.

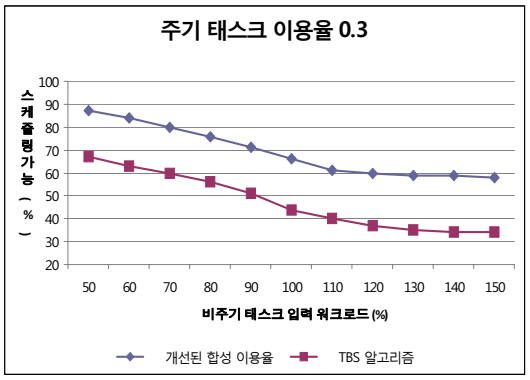

그림 4. $T_p = 0.3$ 인 경우 혼합 태스크 스케줄링
Fig. 4 Mixed Task Scheduling in $T_p = 0.3$

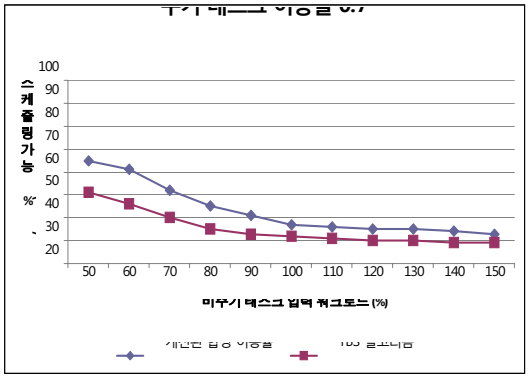

그림 5. $T_p = 0.7$ 인 경우 혼합 태스크 스케줄링
Fig. 5 Mixed Task Scheduling in $T_p = 0.7$

Ⅴ. 결 론

비주기 태스크들을 스케줄링하는 경우 대부분의 알고리즘이 주기 태스크를 먼저 스케줄링 한 후 나머지 대역폭을 비주기 태스크에 할당하여 스케줄링하는 방법을 사용한다. 본 논문에서는 EDF 스케줄링 방식을 이용하여 비주기 태스크만을 스케줄링하는 합성 이용율 방법을 주기 태스크에 적용할 수 있도록 변경하였고, 이를 이용하여 주기 태스크와 비주기 태스크가 혼합된 태스크 집합들을 스케줄링하는 방법을 제시하였다. 합성 이용율은 프로세서 이용율을 이용하기 때문에 스케줄링 가능성 여부를 빠르게 판단할 수 있다. 본 논문에서 제시

한 주기 태스크와 비주기 태스크의 혼합 집합에 대한 스케줄링 가능성 여부는 주기 태스크의 이용율에 따라서 차이를 보이게 되는데 주기 태스크의 이용율이 높아지면 비주기 태스크가 사용할 수 있는 대역폭이 줄어들기 때문에 상대적으로 스케줄링 가능성이 줄어들게 된다. 주기 태스크의 이용율이 낮은 경우 TBS 알고리즘과 본 논문에서 제시한 방법을 비교하면 약 20%정도 성능 차이를 보이며, 또한 워크로드에 따라 스케줄링 가능성이 TBS 알고리즘보다 개선된 것을 볼 수 있었다.

본 논문에서는 단일 프로세서 환경에서 주기 태스크와 비주기 태스크의 혼합된 태스크 집합에서의 이용율 측정을 통한 스케줄링 가능성 여부를 판단하였지만 다중 프로세서 환경과 분산 실시간 시스템 환경에서 혼합 태스크 집합의 스케줄링 가능성 판단을 위한 연구가 필요하다.

참고문헌

[1] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in hard real time environment", Journal of the ACM, vol. 20, pp. 46-61, Jan. 1973.

[2] H. Chetto and M. Chetto, "Some results of the earliest deadline first scheduling algorithm", IEEE Transactions on Software Engineering, vol. 15, no. 10, pp. 1261-1268, Oct. 1989.

[3] T. F. Abdelzaher and C. Lu, "Schedulability analysis and utilization bounds for highly scalable real-time services", in Proceedings of IEEE Real-Time Technology and Applications Symposium, May. 2001.

[4] T. F. Abdelzaher and B. Anderson, J. Jonsson, V. Sharma, and M. Nguyen. "The aperiodic multiprocessor utilization bound for liquid tasks." in Real-time and Embedded Technology and Applications Symposium, San Jose, California, Sep. 2002.

[5] T. F. Abdelzaher and V. Sharma, "A synthetic utilization bound for aperiodic tasks with resource requirements." in 15th Euromicro Conference on Real-Time Systems, porto, Portugal, July. 2003.

[6] 문석환, 김인국, "실시간 비주기 태스크 스케줄링을 위한 개선된 합성 이용율에 관한 연구," 디지털콘텐츠학회 논문지, 제9권 제3호, pp.441-448, 2008.

[7] M. Spuri and G. C. Buttazzo, "Scheduling Aperiodic Tasks in Dynamic Priority Systems," Journal of Real-Time Systems, vol. 10, no. 2, pp. 179-210, 1996.

[8] T. F. Abdelzaher, V. Sharma, and C. Lu, "A Utilization bound for aperiodic tasks and priority driven scheduling", IEEE Transactions on Computers, vol. 53, no. 3, pp. 334-350, Mar. 2004.

저자소개

문석환(Seok-Hwan Moon)

2001년 단국대학교 전자계산학과 석사
2009년 단국대학교 전자계산학과 박사
2006년 영동대학교 임베디드소프트웨어학과 교수
※관심분야:운영체제, 실시간시스템, 임베디드시스템

김인국 (In-Guk Kim)

1982년 단국대학교 학사
1985년 미국 에모리대학교 석사
1995년 아주대학교 박사
1986년~현재 : 단국대학교 컴퓨터학부 컴퓨터과학전공 교수
2001년 ~ 2003년 : 미국 뉴멕시코공과대학 방문교수
※관심분야:운영체제, 실시간시스템, 임베디드시스템