다중처리기 상의 실시간 스케줄링을 위한 동일 우선순위 처리 정책의 비교

박민규, 한상철, 김희헌, 조성제†, 조유근
서울대학교 전기.컴퓨터공학부
†단국대학교 정보컴퓨터학부
{mkpark, schan, hhkim}@ssrnet.snu.ac.kr, †sjcho@dku.edu, cho@cse.snu.ac.kr

Comparison of Tie-Breaking Policies for Real-Time Scheduling on Multiprocessor

Minkyu Park, Sangchul Han, Heeheon Kim, Seongje Cho†, Yookun Cho
School of Electrical Engineering and Computer Science, Seoul National University
†Division of Information and Computer Science, Dankook University

요 약

단일처리기에서는 우선순위가 동일한 작업들 사이의 수행 순서가 스케줄 가능성에 영향을 미치지 않는다. 그러나 다중처리기에서는 우선순위가 동일한 작업들의 수행 순서를 결정하는 정책에 따라 주어진 태스크 집합의 스케줄 가능성이 달라질 수 있다. 본 논문은 동일 우선순위 처리 정책 간의 관계를 연구하고, 모의실험을 통하여 스케줄 가능한 태스크 집합의 수와 스케줄 보장 이용률, 선점 회수의 측면에서 정책들의 성능을 비교하였다.

1. 서 론

단일처리기에서의 실시간 스케줄링 문제에 대해서는 많은 연구가 있었지만, 그러한 연구들의 결과를 다중처리기에 그대로 적용할 수 없는 경우가 많다. 그 중의 하나는 동일 우선순위 처리 정책(tie-breaking policy)이다. 단일처리기 상에서는 같은 우선순위를 갖는 작업의 수행 순서가 스케줄 가능성에 영향을 미치지 않는다. 그러나 다중처리기에서는 스케줄이 동일 우선순위 작업의 처리 방식에 따라 그 결과가 달라질 수 있다[1].

본 논문은 다중처리기 상의 스케줄링 알고리즘인 EDF(Earliest Deadline First)[2]와 EDZL(Earliest Deadline until Zero Laxity)[3]에서 동일 우선순위 처리 정책을 분석하고, 모의실험을 통해 정책의 성능을 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 EDF, EDZL 알고리즘과 동일 우선순위 처리 정책에 대한 관련 연구를 기술한다. 3장에서는 본 논문에서 고려한 동일 우선순위 처리 정책 간의 관계를 분석한다. 4장에서는 모의실험을 통하여 정책들의 성능을 비교 평가하고, 끝으로 5장에서 결론을 내린다.

2. 관련연구

현재 시간에 준비된 작업들에 대해 마감시한이 빠른 순서로 높은 우선순위를 부여하는 EDF는 단일처리기에서는 최적(optimal)이나 다중처리기에서는 최적이 아니며, 스케줄 보장 이용률이 상대적으로 낮다. 그럼에도 불구하고, EDF가 다중처리기에서 계속 연구되는 이유는 선점(preemption)회수와 처리기간 이주 회수가 작업의 개수에 제한되어 매우 효율적인 구현을 할 수 있기 때문이다[4,5].

EDZL은 마감시한과 여유시간을 모두 고려하여 우선순위를 부여한다. 여유시간(laxity)이 0인 작업이 없을 때에는 EDF에 따라 우선순위를 부여하며, 여유시간이 0인 작업이 발생하면(이 작업은 당장 수행하지 않으면 마감시한을 만족할 수 없다) 가장 높은 우선순위를 부여한다.

Goossens 등은 다중 처리기 상에서 EDF로 스케줄 할 때, 최적의 온라인 동일 우선순위 처리 정책이 존재하지 않음을 보였고, 다중처리기에서 동기적인(synchronous) 주기 태스크 집합을 EDF로 스케줄 할 때도 최적의 정적인(static) 동일 우선순위 처리 정책이 존재하지 않음을 보였다[1]. 동일 우선순위 처리에 대한 최적이 정책이 존재하지 않으므로, 여러 정책에 대한 분석과 실험적인 성능 평가가 필요하다.

3. 동일 우선순위 처리 정책

본 논문에서 고려한 동일 우선순위 처리 정책은 PR(Preemption Reduction)과 SRC(Shortest Remaining Computation), LRC(Longest Remaining Computation)이다. 본 장에서는 각 정책을 기술하고 정책 간의 관계를 분석한다.

PR은 가능한 한 선점이 발생하지 않도록 동일 우선순위를 처리하여 선점 오버헤드를 줄이는 정책이다. SRC는 남은 수행시간이 짧은 작업에게 처리기를 할당한다. 이렇게 함으로써 처리기의 유휴시간이 빨리 생겨 처리율(throughput)을 높일 수 있고, 비주기 작업의 수행에 유리하다. LRC는 남은 수행시간이 긴 작업을 선택한다. 마감시한이 같으므로 남은 수행시간이 긴 작업은 여유시간이 작다. 여유시간이 작은 것을 선택하므로 긴급한 작업을 먼저 수행하여 스케줄 가능성을 높인다. 이 후 본 논문에서는 동일 우선순위 처리 정책에 따라 EDF-PR,

EDF-SRC, EDF-LRC와 같은 방식으로 표기한다.

SRC와 LRC는 작업의 남은 수행시간을 참조하여 우선순위를 결정한다. 남은 수행시간은 작업의 수행에 따라 변동하는 값이므로 동일한 우선순위의 작업들 간의 수행순서가 시간이 흐름에 따라 바뀔 수 있다(우선순위가 다른 작업들 간의 수행순서는 변하지 않는다). 엄밀한 의미의 EDF에서는 우선순위가 동일한 작업들의 수행순서는 준비되는 시점에 결정되며 그 수행을 마칠 때까지 변하지 않는다. EDF-SRC와 EDF-LRC는 우선순위가 같은 작업들 간의 수행순서가 바뀔 수 있으므로 순수한 EDF와 차이가 있다.

정의 1. 알고리즘 A에 의해 스케줄 가능한 임의의 태스크 집합 τ가 알고리즘 B에 의해 항상 스케줄 가능하면, B는 A를 dominate한다고 하며, A는 B에 의해 dominate 된다고 한다.

알고리즘 간의 domination 관계는 스케줄 가능성 측면에서 우수성을 비교하는 좋은 기준이다. dominate 하는 알고리즘은 dominate 되는 알고리즘이 스케줄 할 수 있는 모든 태스크 집합을 스케줄 할 수 있으므로 dominate 하는 알고리즘이 더 우수하다고 단정할 수 있다.

정리 1. EDF-PR, EDF-SRC, EDF-LRC는 서로 dominate 하지 못한다.

증명) 3개의 처리기 상에서 태스크 집합 τ_1={(3,8), (5,6), (6,9), (2,3)}[1]은 EDF-SRC와 EDF-LRC에 의해 스케줄 가능하지만 EDF-PR에 의해 스케줄 가능하지 않다. 2개의 처리기 상에서 태스크 집합 τ_2={(2,4), (5,6), (1,2), (1,6)}은 EDF-PR과 EDF-LRC에 의해 스케줄 가능하지만 EDF-SRC에 의해 스케줄 가능하지 않다. 2개의 처리기 상에서 태스크 집합 τ_3={(1,4), (1,7), (3,7), (7,8)}은 EDF-PR과 EDF-SRC에 의해 스케줄 가능하지만 EDF-LRC에 의해 스케줄 가능하지 않다. ■

정리 2. EDZL-PR, EDZL-SRC, EDZL-LRC는 서로 dominate 하지 못한다.

증명) 2개의 처리기 상에서 태스크 집합 τ_4={(1,4), (3,6), (5,8), (5,8)}은 EDZL-SRC에 의해 스케줄 가능하지만 EDZL-PR에 의해 스케줄 가능하지 않다. 태스크 집합 τ_5={(2,4), (2,4), (2,5), (6,10)}은 EDZL-LRC에 의해 스케줄 가능하지만 EDZL-PR에 의해 스케줄 가능하지 않다. 2개의 처리기 상에서 태스크 집합 τ_6={(2,3), (1,5), (3,6), (6,10)}은 EDZL-PR, EDZL-LRC에 의해 스케줄 가능하지만 EDZL-SRC에 의해 스케줄 가능하지 않다. 2개의 처리기 상에서 태스크 집합 τ_7={(1,2), (1,4), (3,4), (3,6)}은 EDZL-PR, EDZL-SRC에 의해 스케줄 가능하지만 EDZL-LRC에 의해 스케줄 가능하지 않다. ■

1) 태스크는 (C, P)로 표시되고, 각각 수행시간과 주기를 의미한다. 마감시한은 주기와 같고, 모든 태스크의 시작시간은 같다고 가정한다.

정리 1~2에서 보듯이 PR과 SRC, LRC는 서로 dominate 하지 못한다. 정책들이 서로 dominate 하지 못하므로 어느 한 정책이 다른 정책보다 우수하다고 단정할 수 없다. 다음 장에서는 정책들의 성능을 모의실험을 통하여 비교, 평가한다.

4. 성능 평가

우리는 모의실험을 통해 EDF와 EDZL에서 동일 우선순위 처리 정책으로 PR, SRC, LRC를 사용할 때 무작위 태스크 집합에 대한 스케줄 가능성, 스케줄 보장 이용률(schedulalbe utilization), 선점회수 등을 비교하였다.

태스크 집합은 [6]의 방법을 응용하여 두 개의 그룹으로 생성하였다. 그룹1에는 이용률이 1~7 사이인 10만 개의 태스크 집합이 속한다. 각 태스크의 주기는 [2,10]의 구간에서, 수행시간은 [1,9]의 구간에서 균일분포(uniform distribution)에 따라 무작위로 생성하였다. 그룹2에는 이용률이 1~7 사이인 1만 개의 태스크 집합이 속하며, 각 태스크의 주기는 [2,100]의 구간에서, 수행시간은 [1,20]의 구간에서 균일분포에 따라 무작위로 생성하였다.

그룹1에 속한 태스크 집합의 특징은 작업들의 마감시한이 일치하는 경우가 많고 태스크의 이용률이 높아 우선순위 경합이 많이 발생한다는 점이다. 그룹2에 속한 태스크 집합은 주기가 비교적 길어서 우선순위 경합이 적게 발생한다. 본 논문에서는 그룹1에 대한 실험 결과만 제시한다. 그룹2에 대한 실험 결과는 그룹1과 유사한 경향을 보이며 정책 간의 차이가 크지 않다.

표 1. 처리기가 4개일 때, 이용률에 따른 스케줄 가능한 태스크 집합의 수(그룹1)

정책 \ 이용률	[1, 2)	[2, 3)	[3, 4)	4
EDF-PR	16877	16558	9423	1
EDF-SRC	16877	16539	8195	0
EDF-LRC	16877	16602	12168	6

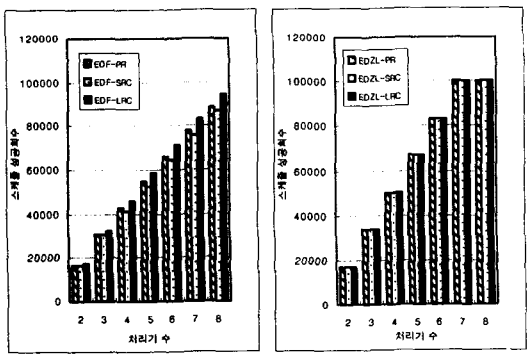

그림 1. 처리기의 수에 따른 스케줄 가능한 태스크 집합의 수(그룹1)

그림 1은 그룹1에 대하여 처리기의 수에 따른 스케줄 가능한 태스크 집합의 수를 보여준다. EDF-LRC는 가장 많은 태스크 집합을 스케줄 할 수 있고, EDF-SRC는 가장 적은 태스크 집합을 스케줄 할 수 있다. LRC는 여유시간이 작은 작업을 선택함으로써 긴급한 작업을 우선적으로 처리한다. 반면에, SRC는 여유시간이 큰 작업을 선택함으로써 여유시간이 작은 작업의 실패 확률을 높인다. 이러한 차이는 태스크 집합의 이용률에 따른 스케줄 가능한 태스크 집합의 수를 나타낸 표 1에서 더욱 잘 알 수 있다. 이용률이 낮은 태스크 집합에 대해서는 각 정책 간의 차이가 크지 않으나, 이용률이 높은 - 이용률이 [3,4) 의 구간인 - 태스크 집합에 대해서는 EDF-LRC의 스케줄 가능한 태스크 집합의 수가 상당히 높다.

그림 1에서 EDZL은 동일 우선순위 처리 정책에 의한 영향이 거의 없음을 보인다. 정책에 의하여 스케줄 가능성에 불리한 방향으로 작업이 선택되어도, 여유시간이 0이 되는 긴급한 작업은 가장 높은 우선순위를 부여받기 때문이다.

그림 2는 처리기가 4개일 때, 태스크 수에 따른 평균 선점회수를 나타낸 것이다. 예상한대로 PR을 사용할 때 선점회수가 가장 낮다. 주목할 것은 LRC가 SRC 보다 선점회수가 많다는 점이다. 그 이유는 LRC를 사용하면 준비상태(ready)의 작업이 상대적으로 많아져서 선점이 발생할 가능성이 높아지기 때문인 것으로 보인다.

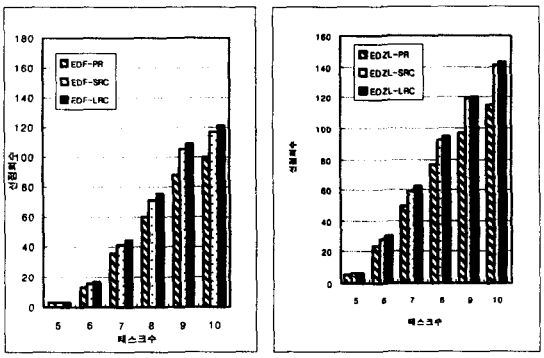

그림 2. 태스크의 수에 따른 평균 선점회수(그룹1)

표 2과 표 3는 처리기의 수에 따른 스케줄 보장 이용률을 보여준다. 스케줄 보장 이용률의 측면에서는 정책간의 차이가 거의 없으며, EDF에서는 LRC가 EDZL에서는 SRC가 다소 높다. EDZL-SRC는 우선순위가 같을 때 남은 수행시간이 작은 작업을 먼저 수행함으로써 그 작업이 수행을 빠르게 마칠 수 있게 한다. 한편, 남은 수행시간이 큰 작업은 여유시간이 빠르게 0이 되어 최고 우선순위를 받을 수 있게 한다.

5. 결론

본 논문은 다중처리기에서 EDF와 EDZL에 대해 동일 우선순위 처리 정책 PR, SRC, LRC의 관계를 보였고, 그 특성을 모의실험을 통해 제시하였다. EDF로 스케줄하는 경우 LRC 정책을 사용할 때 스케줄 가능한 태스크 집합의 수가 가장 많고 스케줄 보장 이용률이 가장 높았다. EDZL로 스케줄 할 경우에는 SRC 정책이 비교적 좋은 성능을 보였다. SRC는 여유시간이 0인 작업이 잘 발생하게 하여, 여유시간이 0인 작업에게 최고 우선순위를 부여하는 EDZL의 특성에 잘 부합한다. PR 정책은 EDF와 EDZL 모두에서 가장 적은 선점회수를 보였다.

표 2. 처리기수에 따른 스케줄 보장 이용률(그룹1)

처리기수	EDF-PR	EDF-SRC	EDF-LRC
2	1.427778	1.299603	1.488095
3	1.782143	1.779762	1.782143
4	2.154762	2.154762	2.168254
5	2.632937	2.632937	2.632937
6	3.060317	3.060317	3.060317
7	3.375794	3.375794	3.784127

표 3. 처리기수에 따른 스케줄 보장 이용률(그룹1)

처리기수	EDZL-PR	EDZL-SRC	EDZL-LRC
2	1.877778	1.877778	1.877778
3	2.893254	2.893254	2.893254
4	3.721429	3.787698	3.721429
5	4.605159	4.605159	4.605159
6	5.641270	5.641270	5.641270
7	6.377381	6.533333	6.377381

참고문헌

[1] Joël Goossens, Raymond Devillers, and Shelby Funk, Tie-Breaking for EDF on Multiprocessor Platforms, The 23rd IEEE International Real-Time Systems Symposium Work-In-Progress, December, 2002

[2] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time envronment. *Journal of the ACM*, 20(1):46.61, 1973.

[3] S. Cho, S. Lee, S. Ahn, K. Lin, "Efficient Real-Time Scheduling Algorithms for Multiprocessor Systems," IEICE Trans. Commun., Vol. E85-B, No.12, pp. 2859-2867, 2002

[4] A. Mok, Task management techniques for enforcing ED scheduling on a periodic task set, in: Proceedings of the 5th IEEE Workshop on Real-Time Software and Operating Systems, Washington, DC, May 1988, pp. 42-46.

[5] Anand Srinivasan and Sanjoy Baruah. Deadline-based Scheduling of Periodic Task Systems on Multiprocessors. Information Processing Letters 84 (2), pp. 93-98. 2002.

[6] Ripoll et al., Improvement in feasibility testing for real-time tasks., Real-Time Systems: The International Journal of Time-Critical Computing 11: 19-39, 1996