

# 비선점형 혼합 임계 실시간 시스템에서의 이월 작업에 대한 분석

윤지아<sup>○</sup> 이진규<sup>‡</sup>

성균관대학교 컴퓨터공학과

yoona4591@gmail.com, jinkyu.lee@skku.edu

## Analysis of Carry Over Job in Non-Preemptive Mixed-Criticality Real-Time System

Jia Yoon<sup>○</sup> Jinkyu Lee<sup>‡</sup>

Dept. of Computer Science and Engineering, Sungkyunkwan University

### 요 약

최근 내장형 실시간 시스템의 유행은 하나의 하드웨어에 여러 가지의 시스템을 혼합하는 것이다. 이런 시스템을 혼합 임계 시스템이라고 하는데 같은 작업이라도 시스템의 중요도에 따라 서로 다른 실행 시간을 가진다. 한마디로, 중요도가 바뀌는 순간의 일들은 실행 시간도 바뀌기 때문에 이 양을 분석하는 것은 스케줄링 분석에서 가장 중요한 핵심이다. 본 논문에선 아직 활발히 연구가 되지 않은 비선점형 혼합 임계 시스템에서 앞서 이야기한 이월 작업(Carry Over Job)을 분석한다. 또한 이를 바탕으로 혼합 임계 시스템에서 실행양을 비교하는 스케줄링 분석의 초석을 제시한다.

### 1. 서 론

실시간 시스템은 한정된 프로세서 자원에서 마감시간이라고 불리는 명시된 시간 안에 요청된 수행에 관한 응답을 보장해야 하는 시스템이다. 이와 관련하여 최근 내장형 실시간 시스템(Embedded Real-Time System) 설계의 유행은 더 저렴한 비용과 효과적인 전력 사용에 관한 제약 사항을 위해 여러 가지 구성요소들을 공유된 하드웨어로 혼합하는 것이다. 예를 들어 항공기에는 탑승객에게 현재의 비행 상태 등을 알려주는 시스템(In-Flight Information System)과 비행 자체를 제어하는 시스템(Flight Control System)이 혼합될 수 있다. 만약 시스템이 낮은 수준의 실시간성 보장을 해야 하는 상황이라면 기존의 실시간 시스템의 모형에서 사용하는 WCET(Worst Case Execution Time)의 보수적인 상계는 너무 비관적인 값이다. 반면 높은 수준의 보장을 해야 하는 상황에선 비행 제어 시스템이 실행하는 동안 다른 시스템은 굳이 마감시간을 만족할 필요가 없다. 결과적으로 각각의 작업의 WCET는 서로 다른 중요도에 따라 다르게 설정되어야 한다.

이런 필요에 따라 2007년 Vestal에 의해 혼합 임계 시스템(Mixed-Criticality System)이 제시되었다.[1] 시스템이 높은 중요도로 동작할 때는 매우 비관적인 WCET로 정확성을 보장해야 하고 더 낮은 중요도로 동작할 때는 꽤 딱 맞는 WCET으로 실행된다. 따라서 혼합 임계 시스템의 가장 큰 특징은 시스템의 중요도에 따라 해당하는 작업에 대해 각각 마감시간을 만족해야 한다는 것이다.

실시간 시스템에서 각각의 작업(Task)에 의해 발현되는 연속적인 일(Job)에 우선순위를 할당하는 알고리즘에는

대표적으로 RM(Rate Monotonic)과 EDF(Earliest Deadline First)를 들 수 있다.[2] 이런 알고리즘에 기반하여 프로세서로의 배치 여부를 결정할 때 선점형 (Preemptive)와 비선점형(Non-Preemptive)방식이 주로 제시된다. 비선점형은 선점정보보다 문맥 교환(Context Switching)의 부하가 적고 상호 배제(Mutual Exclusion)를 위한 간단한 체제를 가진다. 또한 커널(Kernel)의 인터럽트(Interrupt), DB의 트랜잭션 등 비선점형을 적용하면 더욱 구현이 쉬워지는 분야가 존재한다. 그러나 선점형에 비해 연구 진행이 더뎠다.

이는 혼합 임계 시스템에서도 마찬가지였다. [1]은 주어진 작업 집합(Task Set)이 스케줄링할 수 있는지 분석할 때 널리 쓰이는 효과적인 분석 방법인 RTA(Response Time Analysis)를 선점형 혼합 임계 시스템에 맞게 적용하였고 [3]은 이를 다중 프로세서로 확장하였다. [4]는 EDF로 스케줄링할 때 어떤 구간에서의 각 작업이 스케줄링할 수 있는 최대 실행 시간인 DBF(Demand Bound Function)[5]의 합을 이용하여 각 중요도에 따른 스케줄링 분석을 제시하였다. 그러나 이런 연구는 모두 선점형 환경에 제한되었다.

혼합 임계 시스템에서 시스템 중요도가 동일하다면 기존의 스케줄링 분석과 같은 방식으로 접근할 수 있다. 그러나 시스템의 특성상 중요도 전이(Criticality Transition)가 발생하며 그 순간에 남은 실행 시간이 변하는 이월 작업(Carry over job)은 분석의 초점이 된다. 본 논문에서는 비선점형 혼합 임계 시스템에서 스케줄링할 때 이월 작업의 특성을 분석한다. 특히 우리는 중요도가 바뀌는 순간에도 마감시간을 지키기 위하여 중요도마다 가상 마감시간(Virtual Deadline)을 할당하는 EDF-VD 스케줄링 알고리즘에 초점을 맞춘다. 결과적으로 이런 특성을 이용하여 비선점형 혼합 임계 시스템의 스케줄링 분석의 초석을 제시한다.

<sup>‡</sup> 교신저자 (Corresponding author)

## 2. 시스템 모델

우리는 본 논문에서 마감시간이 주기보다 더 짧거나 같은 주기적이고 산발적인 작업 모델(Periodic and Sporadic Task Model)을 혼합 임계 시스템으로 확장하는 기존의 시스템 모델을 사용한다.[1][3][4] 본 논문에서는 이중 중요도로 제한하며 작업 집합  $\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$  에 속한 각각의  $\tau_i = (T_i, C_i^{LO}, C_i^{HI}, D_i, L_i)$  로 표현된다.  $T_i$  는 작업의 다음 일이 배포되기까지의 최소한의 간격, 즉 주기를 의미한다.  $C_i^{LO}$ ,  $C_i^{HI}$  는 각 중요도에 따른 WCET이다.  $D_i$  는 연관된 마감시간이며  $L_i$  은 작업의 중요도를 나타낸다. 각각의 작업은 같은 인자를 가진 비슷한 일을 무한히 연속적으로 생산한다. 우리는 대기 중인 일이 있다면 유휴 시간(Idle Time)을 삽입하지 않는 일 보존 스케줄링(Work Conserving Scheduling)을 가정한다. 각각의 작업은 중요도에 따른 WCET과 마감시간, 주기에 대하여 다음 부등식을 반드시 만족한다.

$$C_i^{LO} \leq C_i^{HI} \leq D_i \leq T_i \quad (1)$$

혼합 임계 시스템은 먼저 낮은 중요도에서 시작한다. 배포부터 마감시간 까지를 스케줄링 윈도우(scheduling window)라고 할 때 만약 어떤 일이 스케줄링 윈도우에서 끝났다는 신호 없이  $C_i^{LO}$  만큼 실행한다면 시스템 중요도는 HI로 전이된다. 이때  $L_i$  이 HI인 모든 일은  $C_i^{HI}$  로 실행되며  $L_i$  이 현재의 시스템 중요도 보다 낮다면 해당 일들을 무시한다. 한번 전이가 발생하면 시스템 중요도는 낮아지지 않는다. 위와 같은 조건으로 시스템의 중요도에 따라 모든 일이 마감시간을 지키는 것을 스케줄링 가능하다고 하며 그의 역도 성립한다.

## 3. 이월 작업과 가상 마감시간

본 장에서는 혼합 임계 시스템에 맞는 알고리즘을 소개하고 이를 시스템에 적용한다. 2장에서 언급한 것처럼 전이의 원인은 특정 일의 실행 시간이다. 만약 끝났다는 신호 없이 특정 일이  $C_i^{LO}$  만큼 실행한다면 전이가 발생하며 전이하는 시점 전에 배포되었으나 아직 끝나지 않은 일들은 전이 후에 최소한  $C_i^{HI} - C_i^{LO}$  만큼 추가로 더 실행해야 한다. [4]에서는 이런 일들을 이월 작업이라고 정의한다.

**정의 1. (이월 작업[4])** 이월 작업이란  $L_i = HI$  인 작업에 의한 일로써 시스템 중요도가 전이하는 순간에 활동하는 일(active job)을 의미한다.

WCET가  $C_i^{LO}$  혹은  $C_i^{HI}$  인지 모르는 이월 작업의 특징 때문에 기존의 연구는 가상 마감시간을 이용한다. 이에 대한 근거로 시스템 중요도가 LO일 때 task  $\tau_i$  의 어떤 일이  $D_i - (C_i^{HI} - C_i^{LO})$  를 넘어서까지 계속해서 실행했다고 가정하자. 만약 해당 일이 결국 이월 작업이 되었다면 남아있는 공간은  $C_i^{HI} - C_i^{LO}$  보다 큰 남아있는 실행 시간을 위해 충분하지 않으므로 마감시간을 넘어갈 수도 있다. 이런 점에서 우리는 다음을 만족하는  $D_i^{LO}$  와  $D_i^{HI}$  를 추가로 설정한다.

$$D_i^{LO} \leq D_i^{HI} \leq D_i \quad (2)$$

우리는 이를 이용하여 혼합 임계 시스템에서 EDF로 스케줄링할 때 중요도가 LO일 때는  $D_i^{LO}$  로, HI일 때는  $D_i^{HI}$  를 이용하여 스케줄링한다. 이런 방식을 EDF-VD[6]라고 한다.

## 4. 비선점 이월 작업의 분석

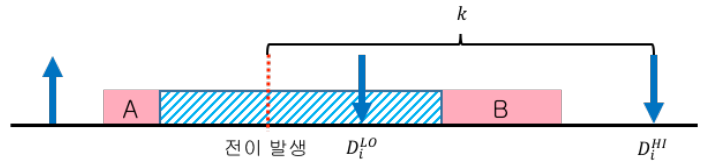


그림 1. k만큼 떨어진 지점에서 전이가 발생한 경우

본 장에선 비선점 혼합 임계 시스템에서의 이월 작업의 특성을 분석한다. 먼저 그림 1과 같이  $D_i^{HI}$  로부터 k 만큼 왼쪽으로 떨어진 지점에서 전이가 발생했다고 가정해보자. 전이를 기준으로 두 개 이상의 일의 실행 시간의 일부분을 배치한다면 이는 반드시 선점에 의한 것이어야만 한다. 정리 1은 이런 현상이 비선점형에서는 발생할 수 없음을 보인다.

**정리 1. 비선점형 단일 프로세서일 때 이월 작업은 반드시 하나 존재한다.**

**증명.** 우선 정의 1에 의해 이월 작업이 존재함으로 전이가 발생했으며, 따라서 비선점형 시스템에서 끝났다는 신호 없이  $C_i^{LO}$  만큼 실행하고 최종적으로  $C_i^{HI}$  까지 실행하는 일이 존재하는 것은 자명한 사실이다. 모순을 찾기 위해 그림 1과 같이 전이 전에 A만큼 실행( $A > 0$ )하고 전이 후에 나머지를 실행하는 이월 작업  $\tau_j$  가 있다고 가정하자.  $\tau_j$  에 의해 전이가 발생하지 않았으므로  $A \leq C_i^{LO}$  를 만족해야 하고 따라서  $B \geq C_i^{HI} - C_i^{LO}$  이다. 결과적으로  $A \neq 0$  이고  $B \neq 0$  이므로 선점에 의해서만 발생함을 알 수 있다. 따라서 비선점형 시스템의 이월 작업은 단 하나 존재한다. □

정리 1에 의해 우리는 단일 프로세서에서 단 하나의 이월 작업에만 초점을 맞추면 된다는 것을 알 수 있었다. 정리 2는 정리 1에 추론될 수 있는 필연적 결과로써 이월 작업의 실행 시간을 결정한다.

**정리 2. 비선점형 단일 프로세서일 때 이월 작업은 전이 전에 반드시  $C_i^{LO}$  만큼 실행한다.**

**증명.** 정리 1에 의해 비선점형 단일 프로세서 환경에선 이월 작업에 의해 전이가 발생하며 그 역도 성립한다. 따라서 이월 작업은 끝났다는 신호 없이  $C_i^{LO}$  만큼 실행한다. □

$L_i = HI$  인 어떤 작업  $\tau_i$  가 어떤 구간 I에서 스케줄링할 수 있는 최대 실행을 한다고 가정해보자. 이때 최대 실행 시간에  $\tau_i$  의 이월 작업 포함 여부를 결정하는 것은 중요한 분석이 된다. 정리 3에서는 이월 작업이 발생할 수 있는 전이의 범위에 대해 한정한다.

**정리 3. 비선점형 단일 프로세서일 때 전이가 발생할 시점으로부터  $D_i^{HI}$  까지의 크기가 k라면  $D_i^{HI} - D_i^{LO} \leq k < D_i^{HI}$  인 경우에만 이월 작업이 존재한다.**

**증명.** 하나의 일에 대해 k의 가능한 위치는 i)  $D_i^{HI} < k < T_i$ , ii)  $k < D_i^{HI} - D_i^{LO}$  혹은 iii)  $D_i^{HI} - D_i^{LO} \leq k < D_i^{HI}$  로 나누어 볼 수 있다.

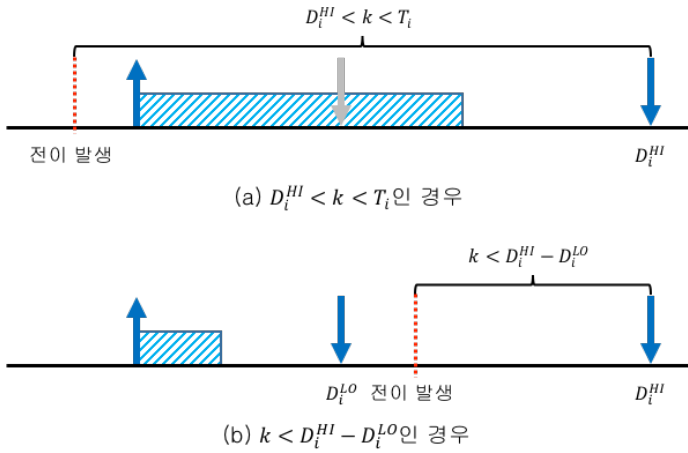


그림 2. 이월 작업이 발생할 수 없는 경우

- i)  $D_i^H < k < T_i$  인 경우(그림 2(a)) 일이 스케줄링 되기도 전에 전이가 발생했으므로 해당 일에 의해 스위치가 발생하지 않았다. 따라서 해당 일은 이월 작업이 아니다.
- ii)  $k < D_i^H - D_i^L$  인 경우(그림 2(b)) 마감시간을 넘지 않는 한에서 일의 실행을 최대한 뒤로 밀었을 때  $D_i^L$  전에  $C_i^L$ 를 모두 완료하고 끝난 상태에서 전이가 발생한다. 따라서 해당 일은 전이를 발생시키지 않았으며 마찬가지로 이월 작업이 될 수 없다.
- iii)  $D_i^H - D_i^L \leq k < D_i^H$  인 경우(그림 3) 이 경우 선점형 단일 프로세서라면 두 가지 종류의 이월 작업이 발생할 수 있다. 먼저 정의 1에 의해 직접 전이를 발생시키지 않았지만 전후로 활동하는 일은 이월 작업이 될 수 있다.(그림 3(a)) 이런 일은 전이 전에 최소한  $C_i^L - (k - (D_i^H - D_i^L))$  만큼은 끝냈지만 전이를 발생시키지 않았으므로 나머지 실행 시간 전에 선점에 의한 밀림이 있어야 한다. 따라서 이런 일은 비선점형에서는 고려하지 않는다. 두 번째는 직접 전이를 발생시키는 활동하는 일이다.(그림 3(b)) 이 경우 정리 1에서 증명한 것과 같이 비선점형에서도 발생할 수 있다.

결과적으로 오직  $D_i^H - D_i^L \leq k < D_i^H$  인 경우에만 이월 작업이 존재한다. □

## 5. 결론 및 향후 연구

혼합 임계 시스템이 기존의 시스템과 다른 가장 큰 특징은 전이가 발생한다는 점이다. 이에 따라 기존 연구는 중요도에 따라 나누어서 스케줄링 가능성을 분석하였다. 또한 전이에 의해 이월 작업이 발생한다는 점에 초점을 맞추었다. 따라서 이월 작업의 전이 전후 실행 시간을 한정 짓는 것은 최종적인 분석을 위해 반드시 거쳐야 할 단계 이다.

우리는 정리 3에 의해 비선점형 단일 프로세서에서 이월 작업이 어떤 순간에 발생하는지를 알 수 있다. 또한 정리 1과 2의 증명으로 만약 이월 작업이 발생했다면 얼마나 실행해 왔는지 분석했다. 이런 특성을 이용하면 이월 작업의 존재 여부에 따라 전이 이전의 실행 시간을 제외하는 방식으로 범위를 한정 지을 수 있을 것이다. 또한 정리 3의 iii)에서

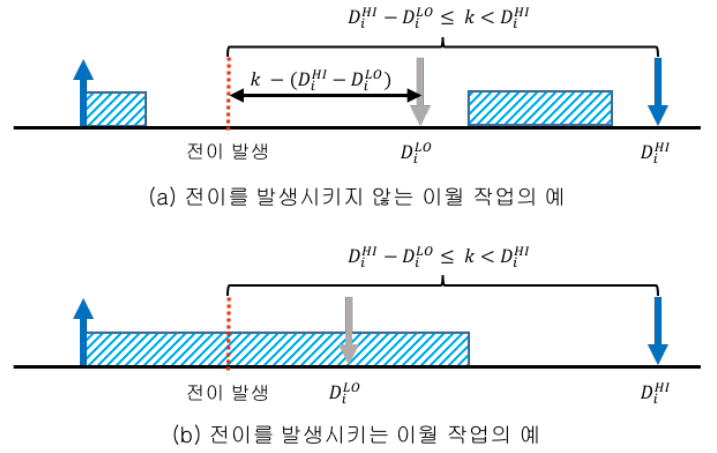


그림 3.  $D_i^H - D_i^L \leq k < D_i^H$  인 경우

비선점형이기에 제외했던 이월 작업은 다중 프로세서 에서 적용할 수 있다. 이런 정리를 바탕으로 한 비선점형 혼합 임계 시스템의 스케줄링 분석은 향후 연구로 남겨둔다.

## Acknowledgements

이 논문은 2014년도 정부(미래창조과학부)와 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2014R1A1A1035827, NRF-2016R1D1A1B03930580). 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학 지원사업의 연구결과로 수행되었음(R22151610050001002). 본 연구는 또한 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [No. R0190-15-2071, (2세부) 클라우드 맵 기반의 자율이동 서비스 다양성 지원을 위한 개방형 Pn형 플랫폼 기술 개발].

## 6. 참고 문헌

- [1] S. Vestal, "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance." In Proc. of RTSS, pages 239-243, 2007.
- [2] C.L. Liu and J.W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment." JACM, 20(1):46-61, 1973.
- [3] Pathan R. "Schedulability analysis of mixed-criticality systems on multiprocessors." In Proc. of ECRTS, pages 309-320, 2012.
- [4] W. Y and Pontus Ekberg, "Bounding and shaping the demand of mixed-criticality sporadic tasks," in Proc. of ECRTS, pages 135-144, 2012.
- [5] S. Baruah, A. Mok, and L. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor", in Proc. of RTSS, pages 182-190, 1990.
- [6] S. Baruah, K., V. Bonifaci, G. D'Angelo, A. Marchetti-Spaccamela, S. Ster, Van Der, and L. Stougie, "Mixed-criticality scheduling of sporadic task systems," in ESA, pages 555-566, 2011.