# ARM 프로세서 기반의 Linux에 실시간 이식 커널(RTiK-Linux) 설계

이승율*, 이상길*, 이철훈†
충남대학교 컴퓨터 공학과
e-mail: {winrate92, sk0137, clee}@cnu.ac.kr

# Design of Real-Time implanted Kernel for Linux (RTiK-Linux) Ported to ARM Processor-based Linux

Seung-Yul Lee*, Sang-Gil Lee*, Cheol-Hoon Lee†
Department of Computer Engineering, Chungnam National University

요 약

실시간 시스템이란 요청된 작업의 논리적 정확성과 시간적 제약을 만족하게 하는 시스템으로 요청된 작업을 예측 가능한 시간 내에 완료하는 시스템을 말한다. 범용 운영체제인 Linux는 실시간성을 지원하지 않기 때문에 ARM 프로세서(Processor) 기반의 Linux 환경에서도 실시간성을 제공하기 위해 본 연구를 진행한다. 본 연구는 Exynos 5422 ARM 프로세서에서 제공하는 타이머를 활용하여 실시간성을 확보하여 Linux 환경에 대한 실시간성 지원 제약을 해결할 수 있다.

## 1. 서론

군용 기기들은 데드라인(Deadline) 안에 작업을 수행해야 하는 매우 엄격한 스케줄러를 통해 정확하고 오차 없는 동작이 요구된다. 엄격하지 않은 스케줄러에 의해 공평한 작업 처리가 진행될 경우 데드라인을 만족하지 못해 미사일 추진 제어 오작동과 같은 문제가 발생할 수 있다. 이처럼 데드라인 안에 작업을 수행시키기 위해 엄격한 스케줄링을 제공해주는 시스템을 실시간 운영체제라고 한다[1]. 실시간 운영체제는 대중적으로 많이 접할 수 있는 Windows나 Linux와는 다른 개발 환경을 제공하며, 이를 사용하기 위해서는 실시간 운영체제를 위한 인력이 별도로 필요하게 된다.

범용 운영체제인 Linux는 작업 간에 공평한 자원 사용을 위한 스케줄링을 제공하기 때문에 엄격한 스케줄러가 필요한 환경에서는 사용하기에 적합하지 않다. 하지만 Linux는 오픈소스로 개발되어 많은 개발자들이 사용하고 있어 접근성이 용이할 수 있다. 따라서 Linux에 실시간성을 제공한다면 실시간 운영체제를 위한 새로운 개발자가 아닌 기존 Linux를 사용할 줄 아는 개발자를 활용할 수 있는 이점을 얻을 수 있다. 이러한 이점 때문에 Linux에 실시간성을 제공하기 위한 프로젝트들이 진행되어 왔다[2].

Linux에 실시간성을 제공하기 위한 프로젝트로는 RTiK-Linux[3][4]가 있다. RTiK-Linux를 통하여 Linux에 실시간성을 제공할 수 있지만 Intel 프로세서 기반의 Linux에서만 동작하도록 개발되어 ARM 프로세서 기반의 Linux에서는 호환되지 않는 단점이 있다. ARM 프로세서는 임베디드 기기 및 스마트 기기에서 많이 사용되고 있는 프로세서로 스마트 기기에서 약 90% 이상의 시장 점유율을 보인다. RTiK-Linux의 사용영역 확장을 위해 임베디드 기기나 스마트 기기에도 실시간성을 제공할 수 있도록 기능 개선이 필요하다. 이를 위해서 RTiK-Linux에서 사용되는 Intel 프로세서의 Local APIC Timer를 대체하여 ARM 프로세서 환경에서도 동작할 수 있도록 하드웨어 종속적인 부분의 수정이 필요하다.

본 논문에서는 RTiK-Linux가 ARM 프로세서 기반의 Linux에서 동작할 수 있도록 기존의 타이머를 ARM 프로세서의 MCT(Multi-Core-Timer) 글로벌 타이머로 대체하여 실시간성을 제공할 수 있도록 연구를 진행한다.

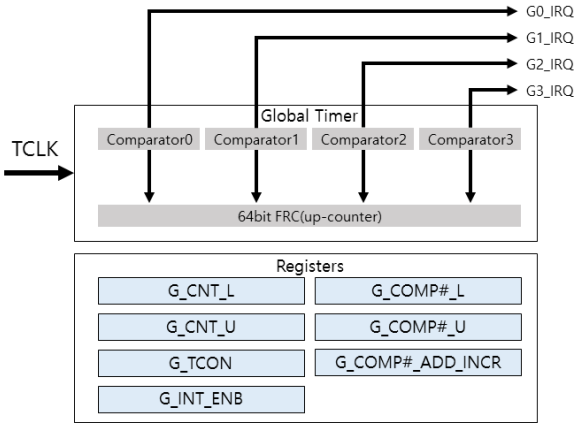
본 논문은 ARM 프로세서 기반의 Linux에 실시간성을 제공하기 위한 RTiK-Linux을 제안하는 것으로 구성은 다음과 같다. 제2장에서는 ARM 프로세서의 MCT 글로벌 타이머를 사용하기 위한 구조 분석과 MCT 글로벌 타이머 설정 방법을 소개하고 제3장에서는 연구 결론 및 향후 연구 방향에 대해 논한다.

## 2. 연구 내용

### 2.1 MCT 글로벌 타이머 구조

MCT 글로벌 타이머는 [그림 1]과 같이 TCLK과 FRC, Comparator들과 레지스터들로 구성되어 동작한다. TCLK은 외부에서 제공되는 24MHz 클럭(Clock)을 prescaler와 divider로 가공한 클럭이다. MCT 글로벌 타이머는 4개의 타이머 인터럽트가 존재하며 각각의 Comparator에 의해 발생한다. FRC는 TCLK이 발생할 때마다 1씩 증가하는 카운터 이다. 각 Comparator는 FRC값과 비교되어 값이 같

을 경우 인터럽트를 발생시켜 타이머 역할을 수행한다.




(그림 1) MCT 글로벌 타이머 구조

### 2.2 MCT 글로벌 타이머 설정 방법

MCT 글로벌 타이머는 레지스터를 이용하여 제어한다. 이를 위해 레지스터의 주소 값을 참고하여 레지스터에 접근하고 값을 설정하여 개발자에게 필요한 타이머 인터럽트를 제공한다. MCT에서 제공되는 타이머를 제어하기 위한 레지스터들은 0x101C_0000 주소를 기준으로 레지스터 맵 공간에 집합되어있어 오프셋 값을 통해 해당 레지스터의 값에 접근할 수 있다. 본 연구에서 사용할 MCT 글로벌 타이머 제어 레지스터들의 오프셋 값은 다음 [표 1]과 같이 정의되어 있다.

(표 1) MCT 글로벌 타이머 제어 레지스터 오프셋

| Register | Offset | Description |
|---|---|---|
| G_CNT_L | 0x0100 | FRC 하위 bit 값 |
| G_CNT_U | 0x0104 | FRC 상위 bit 값 |
| G_COMP1_L | 0x0200 | Comparator1 하위 bit 값 |
| G_COMP1_U | 0x0204 | Comparator1 상위 bit 값 |
| G_COMP1_ADD_INCR | 0x0208 | Comparator1 자동 증가 값 |
| G_TCON | 0x0240 | 글로벌 타이머 제어 |
| G_INT_ENB | 0x0248 | 인터럽트 활성화 |

각 레지스터는 32bit로 구성되어 있으며 레지스터를 통해 FRC와 Comparator 값을 변경하거나 읽어 올 수 있다. 또한, 64bit 범위의 데이터를 표현하기 위해 상위 bit와 하위 bit로 나누어 2개의 레지스터로 표현할 수 있다. FRC의 경우 G_CNT_U, G_CNT_L 레지스터를 통해 제어하고, 4개의 Comparator는 G_COMP#_U, G_COMP#_L 레지스터로 제어된다. 타이머의 동작을 위한 속성 값은 G_TCON 레지스터로 제어한다. G_TCON 레지스터에는 FRC의 활성화를 위한 Timer Enable bit와 각 타이머별 Comparator와 FRC의 비교처리를 활성화하는 Comp# Enable bit, Comparator 값을 자동 증가시키는 Auto-Increment# bit 등이 존재한다. 자동으로 증가할 값은 G_COMP#_ADD_INCR 레지스터에서 설정된다. G_INT_ENB 레지스터는 각 타이머에서 인터럽트 발생 조건이 만족될 때, 실제 인터럽트의 발생 여부를 설정하는 값이다.

## 3. 성능측정결과

본 논문에서는 MCT 글로벌 타이머가 0.1ms 마다 인터럽트를 발생시키도록 레지스터를 설정하였다. 설정 후 발생되는 인터럽트의 핸들러를 통해 태스크를 관리하여 실시간성을 제공할 수 있다. 수정된 RTiK-Linux의 성능 평가를 위해 실시간 태스크의 동작주기를 측정하였다.

(표 2) 오차율 측정 결과

| | 기존 RTiK-Linux 0.1ms 주기 | 수정된 RTiK-Linux 0.1ms 주기 |
|---|---|---|
| 평균(ms) | 0.0988152 | 0.1000002 |
| 표준편차(ms) | 0.0020712 | 0.0013846 |

실시간 태스크의 동작주기를 측정해본 결과 [표 2]와 같이 평균과 표준편차 값을 구할 수 있었다. [표 2]를 보면 짧은 주기로 설정된 태스크의 동작주기가 적은 오차범위로 동작되는 것을 볼 수 있으며, 기존 RTiK-Linux와 성능이 유사한 것을 확인할 수 있다.

## 4. 결론 및 향후 연구 방향

기존 RTiK-Linux는 Intel 프로세서 환경만을 고려하여 ARM 프로세서 환경에서는 동작할 수 없게 개발되어 있다. 본 논문은 RTiK-Linux가 ARM 프로세서 환경에서 동작할 수 있도록 Intel 프로세서의 Local APIC Timer를 ARM 프로세서의 MCT 글로벌 타이머로 대체하여 실시간성을 제공할 수 있게 수정하였다. 이를 통해 RTiK-Linux가 ARM 프로세서 기반의 Linux에 실시간성을 제공할 수 있게 되었다. 향후 연구 방향으로는 사용자 영역의 일반프로그램을 실시간 태스크로 등록함으로써 RTiK-Linux를 쉽게 사용할 수 있는 방안이 있다. 이를 위해서는 사용자 영역에서 RTiK-Linux를 제어할 수 있는 인터페이스와 실시간 태스크로 등록된 일반프로그램의 관리를 위한 구조를 개발하는 연구가 필요하다.

## 참고문헌

[1] Jean J. Labrosse, "uC/OS-III The Real-Time Kernel", Micrium Press, 2010

[2] National Instruments, http://www.ni.com/, 2016

[3] Joo-Man Kim, Chang-In Song, Cheol-Hoon Lee, "RTiK-Linux: The Design of Real-Time implemented Kernel for Linux", Journal of the korea contents association, Vol 11, No.9, 2011

[4] Sang-Gil Lee, Seung-Yul Lee, Cheol-Hoon Lee, "Middleware to Support Real-Time in the Linux User-Space", Journal of the korea contents association, Vol 16, No5, 2016

[5] Samsung, "Exynos 5422_POP User's Manual", 2014