

## ARM Cortex-A15 프로세서를 위한 RTOS 포팅 구현

심철<sup>0</sup>, 최민<sup>\*</sup>

<sup>0</sup>충북대학교 정보통신공학부

e-mail: eisen@cbnu.ac.kr<sup>0</sup>, cuteconference@gmail.com<sup>\*</sup>

## Implementation of Porting RTOS to ARM Cortex-A15

Cheol Sim<sup>0</sup>, Min Choi<sup>\*</sup>

<sup>0</sup>Dept. of Information and Communication Engineering, Chungbuk University

### ● 요약 ●

본 논문에서는 가상화 기술을 지원하는 ARMv7 Cortex-A15 프로세서가 탑재된 임베디드 보드를 위한 실시간 운영체제 (RTOS) 구현을 위한 방법을 제안한다. ARM Cortex-A15 프로세서가 지원하는 Generic Interrupt Controller와 Generic Timer에 대해 알아보고, 가상화 시스템에서 인터럽트 오버헤드를 줄이는 GICv2와 가상 타이머인 Generic Timer를 사용하여 실시간 운영체제인 FreeRTOS를 임베디드 보드에 이식하였다.

**키워드:** OS이식(OS porting), 실시간 운영체제(Real-Time OS), 임베디드 시스템(Embedded System)

### I. Introduction

실시간 운영체제(RTOS)는 항공우주, 로봇, 군용 장비 등 엄격하고 작업 시간을 보장받아야 하는 분야에서 널리 쓰이고 있다. 즉, 주어진 작업을 정해진 시간 안에 수행하고 예측 가능한 일정 응답 시간을 요구되는 응용 프로그램을 위해 사용된다.

다수의 OS를 하나의 시스템에서 가동할 수 있는 가상화 (Virtualization) 기술을 지원하는 ARM 프로세서의 발전함에 따라 하나의 하드웨어 자원으로 Linux등의 OS와 RTOS를 동시에 사용할 수 있게 되었다. 이는 다른 OS를 사용하는 장비와의 통신비용 등의 하드웨어비용을 절감할 수 있게 되었고, 가상화를 지원하는 ARM 프로세서를 탑재한 임베디드 보드에 RTOS를 사용할 수 있도록 운영체제 포팅이 필요하게 되었다.

본 논문에서는 오픈소스 RTOS인 “FreeRTOS”를 가상화 기술이 지원되는 ARM Cortex-A15 프로세서가 탑재된 임베디드 보드에서 사용하기 위한 운영체제 포팅을 구현하였다.

### II. Background and Design

FreeRTOS 운영체제는 사용자에 의해 생성된 Task들을 시스템 시간이 변화할 때 문맥 전환 등의 스케줄을 진행한다. 시스템의 시간을 조정하기 위해 ARM Cortex-A15 프로세서에서 제공하는 Generic Timer와 인터럽트를 제어하는 GIC를 사용하여 구현하였다.

ARM Cortex-A15는 인터럽트가 발생했을 때 하이퍼바이저의 개입 없이 Guest OS가 인터럽트를 처리하여 오버헤드를 줄이는 인터럽트 가상화 기술인 GICv2을 지원한다. Fig. 1은 GICv2에 대한 개념도이다.

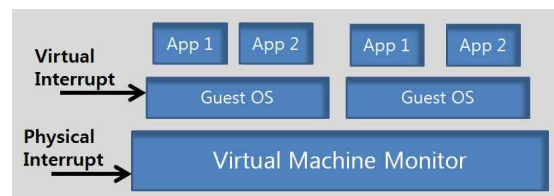


Fig. 1. Conceptual diagram of GICv2

Generic Timer는 각 가상 머신(Guest OS)의 시간 측정을 위한 가상 타이머 인터럽트를 발생시킨다.

GIC(Generic Interrupt Controller)는 가상 인터럽트들을 처리한다. Generic Timer 또한 각 가상 머신에서 발생한 인터럽트이므로 GIC를 사용해서 제어한다. 여기서 Generic Timer를 제어하기 위해 GIC는 가상 머신에게 발생한 인터럽트의 분배를 담당하는 GICD와 가상 머신이 발생한 IRQ에 대한 정보를 확인하는 인터페이스 GICC를 사용한다.

FreeRTOS 포팅 구현을 위해서 운영체제의 스케줄이 시작하기 전에 Generic Timer와 GIC를 초기화하여 타이머 인터럽트를 발생시킨다. Generic Timer의 인터럽트 서비스 루틴에서는 시스템의 시간을 조정하는 vTaskIncrementTick() 함수와 대기 큐에 존재하는 Task와 현재 Task간의 문맥 전환을 수행하는 vTaskSwitchContext() 함수를 수행하도록 설계한다.

### III. Implementation

본 논문에서는 2장에서 설계한 내용으로 ARMv7 Cortex-A15 프로세서가 탑재된 NVidia Jetson TK-1 개발 보드에 FreeRTOS 포팅을 구현하였다. Fig. 2는 Jetson TK-1 개발 보드의 사진이다.

FreeRTOS에서 수행될 Task 3개를 생성하고, 타이머 인터럽트에 의해 Task 간의 문맥 전환이 정상적으로 이루어지는지 확인하였다. Fig. 3은 UART0을 통하여 3개의 Task 간의 정상적인 문맥 전환을 보여주는 화면이다.

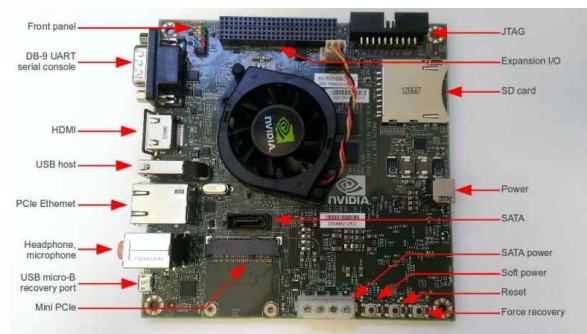


Fig. 2. NVidia Jetson TK-1

```
Tegra124 (Jetson TK1) # bootm 0xa0000000
## Booting kernel from Legacy Image at a0000000 ...
Image Name:   FreeRTOS
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:   8432072 Bytes = 8 MiB
Load Address: a0000000
Entry Point: a0000000
Verifying Checksum ... OK
Loading Kernel Image ... OK

Starting kernel ...

arch_timer_init: osc freq is 120000000Hz [0x00B71B00]
arch_timer_init: setting CNTFID0 to 0x00B71B00
Start FreeRTOS!
Create Tasks!
Generic Timer Setup!
GICD & GICC Initialization!
Enable Generic Timer(PPI4, IRQ 27)
=== FreeRTOS Schedule Start ===
Again TK1 --- Task 3
hello TK1 ----- Task 1
Goodbye TK1 ----- Task 2
Goodbye TK1 ----- Task 2
Goodbye TK1 ----- Task 2
Goodbye TK1 ----- Task 2
Again TK1 --- Task 3
Goodbye TK1 ----- Task 2
hello TK1 ----- Task 1
Goodbye TK1 ----- Task 2
Again TK1 --- Task 3
Goodbye TK1 ----- Task 2
Goodbye TK1 ----- Task 2
Goodbye TK1 ----- Task 2
hello TK1 ----- Task 1
Again TK1 --- Task 3
Goodbye TK1 ----- Task 2
```

Fig. 3. Context Switching between tasks

### IV. Conclusions

본 논문에서는 가상화 기술을 지원하는 ARM Cortex-A15 개발 보드에서 인터럽트 가상화 기술(GICv2)과 가상 타이머(Generic Timer)를 사용하여 실시간 운영체제인 FreeRTOS가 동작할 수 있도록 구현하였다. 향후 여러 OS를 동작시키기 위해 하이퍼바이저에서 동작하는 다양한 운영체제 포팅이 필요하다.

### References

- [1] FreeRTOS Project, <http://www.freertos.org>
- [2] Dongha Shin, Jiyeon Kim, "Implementation of Hypervisor for Virtualizing uC/OS-II Real Time Kernel" Journal of the Korea Society of Computer and Information, Vol. 12, No. 5, pp.103-112, 2007.
- [3] P. Varanasi, G. Heiser, "Hardware-supported virtualization on ARM" APSys '11 Proceeding of the Second Asia-Pacific Workshop on Systems, pp. 11:1-11:5, 2011.
- [4] Inkyu Han, "K-Hypervisor : Design and implementation of hypervisor based ARM Cortex-A15" Proceedings of the Korea Information Science Society, pp.1559-1561, 2014.