실시간 운영체제 CNU_RTOS의 성능 분석

고영관°, 박세영*, 이철훈*
°*충남대학교 컴퓨터공학과
e-mail:{sourring, phy6693, clee}@cnu.ac.kr

Performance analysis of Real-Time operating system CNU_RTOS

Young-Kwan Ko°, Se-Young Park*, Cheol-Hoon Lee*
°*Dept. of Computer Science, Chung-Nam National University

● 요 약 ●

임베디드 시스템은 특정 임무를 수행하도록 설계된 전용 컴퓨팅 시스템으로 그 용도에 따라 다양한 하드웨어 구성요소를 가진다. 이러한 하드웨어 기술의 빠른 성장으로 임베디드 시스템의 발전은 가속화 되고 있고, 이에 따라 임베디드 시스템은 복잡한 작업을 빠르게 수행하기 위해 실시간 운영체제를 도입하게 된다. 임베디드 시스템이 실시간 운영체제를 사용함에 따라 실시간 운영체제의 관심이 증가하고 다양한 실시간 운영체제가 개발되었다. 따라서 시스템에 맞는 실시간 운영체제를 사용하기 위해 성능 및 특성을 분석해야 한다. 본 논문에서는 국산 실시간 운영체제인 CNU_RTOS의 성능을 분석함으로써 다양한 임베디드 시스템에 적용 될 수 있다는 것을 보여준다.

키워드: 임베디드 시스템(Embedded System), 실시간 운영체제(Real-Time Operating System), CNU_RTOS

I. 서론

최근 임베디드 시스템은 하드웨어의 발달로 여러 가지 작업을 동시에 할 수 있게 되었고 처리할 수 있는 작업량의 증가에 따라 작업의 복잡성 또한 증가하게 되었다. 이에 임베디드 시스템은 단순한 펌웨어 수준의 작업이 아닌 복잡한 작업을 처리하기 위해 운영체제의 사용이 필요하게 되었다. 더욱이 임베디드 시스템은 특정 임무를 수행하기 위한 시스템이기 때문에 멀티 태스킹을 중요시 하는 범용 운영체제보다 우선 순위 스케줄링 기반의 실시간 운영체제를(RTOS, Real Time Operating System) 많이 사용한다. 이에 따라 실시간 운영체제의 중요성도 높아지고 있고 최근엔 다양한 특성을 가지는 실시간 운영체제들이 개발이 되면서 시스템의 특성에 따라 임베디드 시스템에 사용 되어 지고 있다. 임베디드 시스템에 적절한 실시간 운영체제를 사용하기 위해서 기존의 실시간 운영체제의 특성 및 성능을 잘 파악해야 한다.

본 논문에서는 국산 실시간 운영체제인 CNU_RTOS의 성능 및 특성을 연구하기 위해 CNU_RTOS와 오픈 소스인 uC/OS II의 성능을 비교 분석하였다.

본 논문의 구성은 2장에서 관련연구로 실시간 운영체제인 CNU_RTOS에 대해 기술하고, 3장에서는 ARM 920T기반의 MBA2440 보드에 uC/OS II를 이용한 CNU_RTOS의 성능 분석과 실험 환경 및 결과를 기술하였으며, 마지막 4장에서 결론을 맺는다.

II. 관련 연구

1. 관련연구

1.1 실시간 운영체제(RTOS)

실시간 시스템이란 시스템의 수행 결과가 기능적으로 정확해야 할 뿐만 아니라, 결과가 도출되는 시간 역시 주어진 제약 조건을 만족시켜야 하는 시스템이다. 이러한 실시간 시스템에 사용되는 운영체제가 실시간 운영체제이다. 범용 운영체제는 CPU의 효율성과 각 프로세스의 공평성에 중점을 두고 스케줄링을 하지만, 실시간 운영체제는 시간적 제약사항에 더욱 중점을 두고 스케줄링 한다.

1.2 CNU_RTOS

연성 실시간 운영체제인 CNU_RTOS는 멀티 태스킹 환경을 지원하고 256단계 우선순위 기반의 선점형 스케줄러를 제공하며, 동일한 우선순위에 대해 First In-First Out(FIFO)와 Round Robin(RR) 스케줄링을 제공한다. 또한 태스크에 관련된 태스크 관리 기능과 메모리를 동적으로 관리하기 위한 메커니즘을 가지고 있으며, 태스크간 동기화를 위해 세마포어와 이벤트 플래그를 제공하며, 태스크간 통신을 위해서 메시지 큐, 메시지 포트, 메시지 메일박스, 태스크 포트를 가지고 있다.

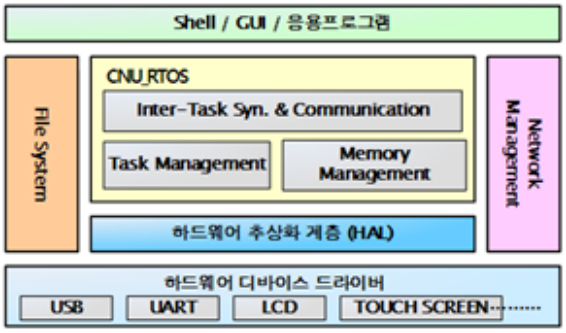

그림 1. CNU_RTOS 구성도
Fig. 1. CNU_RTOS Architecture

III. 본론

1. 본론

1.1 CNU_RTOS 및 uC/OS II의 성능 비교 분석

CNU_RTOS의 성능을 분석을 위해 상용 실시간 운영체제와 성능을 비교 분석 하고 싶었으나, 대부분의 상용 실시간 운영체제들의 소스가 공개되어 있지 않기 때문에 오픈 소스인 uC/OS II와의 비교 분석을 하였다. CNU_RTOS는 상용화된 실시간 운영체제로 교육용을 목적으로 둔 uC/OS II와 크기 및 성능 면에서 차이가 있다.

본 논문에서는 CNU_RTOS와 uC/OS II에서 제공하는 서비스들을 비교하고, 각 API들의 수행 속도를 측정 하고 그 값을 비교함으로써 성능 측정하는 것에 중점을 두어 설계 하였다. [표 1]은 CNU_RTOS와 uC/OS II에서 제공하는 서비스들을 비교한 것이다.

표 1. CNU_RTOS 및 uC/OS II의 서비스 비교
Table 1. CNU_RTOS and uC/OS II comparison of the services

		CNU_RTOS	uC/OS II
Task	최대 우선순위	255	63
	라운드 로빈	○	X
Event		○	○
HISR		○	X
Memory	Heap	○	X
	Pool	○	○
Mailbox		○	○
MsgPort		○	X
MsgQ	Fixed	○	○
	Variable	○	X
Semaphore		○	○
Signal		○	X
TaskPort		○	X

1.2 실험 환경 및 결과

실험환경은 ARM920T 기반의 MBA2440 보드에 CNU_RTOS와 uC/OS II를 포팅하였고, 디버거는 OpenICE와 AIJI시스템의 A1000 디버거를 사용하였다. 컴파일러는 ARM ADS1.2를 사용하였고, 정확한 속도 측정을 사용하기 위해 Tektronix TDS2002를 이용하여 동일한 실험 환경을 제공하였다.

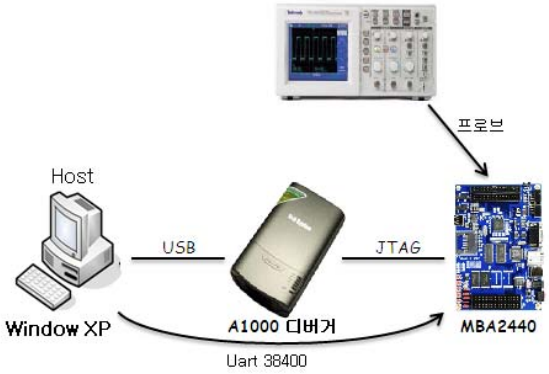

그림 2. 테스트 환경
Fig. 2. Test Environment

실험은 두 실시간 운영체제에서 제공하는 API들을 주기적으로 호출을 하게 되고, API의 시작과 끝에서 오실로스코프에 신호를 보내는 방식을 이용하였다. 아래 [그림 3]는 CNU_RTOS에서 스케줄러를 호출하기 위해 제공하는 MK_Schedule()함수의 시간 측정 결과로 us 단위로 출력 하였으며 uC/OS II도 같은 방식으로 결과를 얻었다. [표 2]는 CNU_RTOS와 uC/OS II에서 측정한 성능을 비교한 것이다.

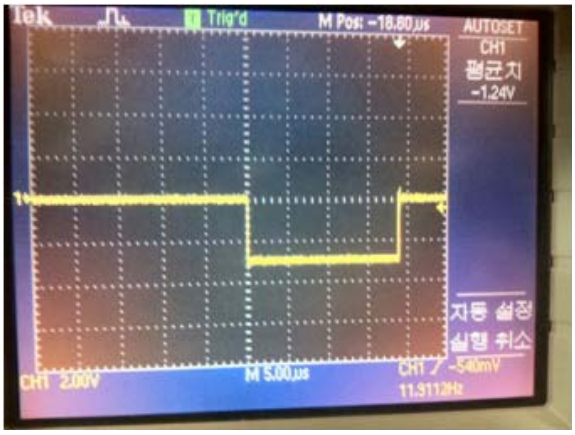

그림 3. CNU_RTOS API 실험 결과
Fig. 3. CNU_RTOS API Results

표 2. CNU_RTOS 및 uC/OS II의 수행 속도 비교
Table 2. CNU_RTOS and uC/OS II execution speed comparison

CNU_ RTOS	Time (uSec)	uC/OS II	Time (uSec)
MK_IRQHandler()	30	OSTimeTick()	8
MK_InterruptEnable()	4.5	Enter_Critical()	2.5
MK_InterruptDisable()	4.5	Exit_Critical()	2.5
MK_InterruptRestore()	4.5	-	
MK_ContextSwitch()	5	OSCTXSW()	11
MK_Schedule()	19	OSSchedule()	21
MK_CreateTask()	42	OSCreateTask()	60
MK_DeleteTask()	60	OSDeleteTask()	40
MK_GetMemory()	27	-	
MK_FreeMemory()	34	-	
MK_GetBuf()	23	OSMemGet()	9
MK_FreeBuf()	22	OSMemPut()	10
MK_SemaphorePend()	11	OSSemPend()	11
MK_SemaphorePost()	11	OSSemPost()	10
MK_MsgQueuePend()	78	OSQPend()	14
MK_MsgQueuePost()	81	OSQPost()	13
MK_MsgMboxPend()	63	OSMBoxPend()	10
MK_MsgMboxPost()	62	OSMboxPost()	3
MK_MsgPortPend()	19	-	
MK_MsgPortpost()	18	-	

위의 결과 값에서 CNU_RTOS와 uC/OS II간에 API 수행 속도의 차이가 나는 것을 확인 할 수 있다. 이는 교육용으로 만들어진 uC/OS II와 달리 CNU_RTOS는 상용화를 목적으로 만들어진 실시간 운영체제이기 때문에 제공되는 서비스의 차이가 나고 API들의 수행 속도들의 차이가 나게 된다.

Ⅳ. 결론

본 논문에서는 CNU_RTOS와 uC/OS II 상에서의 태스크, 스케줄링, 메모리 할당/삭제 및 ITC(Intertask Communication)등의 API에 대한 수행 시간을 측정 및 비교 하였다.

결과적으로 CNU_RTOS와 uC/OS II의 제공하는 서비스 및 수행 속도의 차이를 볼 수 있다. 다른 상용 실시간 운영체제와의 서비스 제공을 비교하였을 때 거의 동일한 서비스를 제공하는 CNU_RTOS의 성능을 분석하여 다양한 임베디드 시스템에 적용 될 것을 기대할 수 있다. 향후 본 연구의 성과를 바탕으로 CNU_RTOS의 서비스의 안정화를 위해 시스템 폴트 상황에 대해 연구하고 CNU_RTOS에 고장허용(Fault-Tolerance)을 지원해 주기 위한 개발이 필요하다.

참고문헌

[1] S3C2440A 32-bit RISC Microprocessor user's Manual Revision 0.12

[2] Jean, J.Labrosse, "uC/OS The Real-Time Kernel", R&D Publications, 1995

[3] Jean, J. Labrosse, "MicroC/OS-II The Real-Time Kernel", second edition, Lawrence, Kansas : R&D Publications, 1999

[4] Q. Li, "Real-Time Concepts for Embedded Systems", McGraw-Hill, 1997

[5] MicroC/OS-II, http://www.micrium.com