

마이크로 커널을 기반으로 하는 Nucleus와 μ ITRON 특성 및 성능 비교 연구

A study on Properties and Comparative Performance of Nucleus and μ ITRON based on microkernel

박상준, 박정형

(PARK SANG JOON, PARK JEUNG HYUNG)

삼성전자 DM 총괄 DM 연구소

Abstract - Generally microkernel has properties of Portability, reusability and scalability. In particular microkernel technique has been applied to development of real time kernel on embedded system because life cycle of micro processor is shortened. In this paper we study Properties of micro kernel and Comparative Performance of Nucleus and μ ITRON based on microkernel.

Key Words : 마이크로 커널, Nucleus, μ ITRON, RTOS

1. 서론

임베디드 시스템의 특성상 초기에는 실시간 처리 요소를 갖는 다중처리(multi-tasking)가 가능한 상용 실시간 운영체제(Real-Time OS)가 주로 사용되었다. pSOS+, VxWorks, VRTX등 이러한 상용 운영체제는 실시간 처리기능을 제공하는데 목적을 두고 신뢰성에 중점을 두며, 가변성이 아닌 특수한 목적에 최적화된 시스템 운영체제이다. 그러나 최근 임베디드 시스템에 네트워크 접속, 멀티미디어 처리 등의 기능이 요구됨에 따라 점차 범용 운영체제의 기능에 가까워지고 있다. 하지만 범용 운영체제와 달리 다양한 하드웨어 환경에 적용되어야 하는 요구사항에 따라 높은 이식성과 재사용성이 요구되며, 이러한 이유로 시스템의 기본기능과 응용기능을 분리하여 모듈성과 재사용성, 이식성을 향상시킨 마이크로 커널 개념을 도입하게 된다. 또한 프로세서 및 메모리 등의 한계성으로 인하여 소형이면서 고속 처리가 요구되는 임베디드 시스템에 필요한 기능만으로 최적화된 성능을 갖게 하는 것이 중요한 문제로 대두된다. [1][2]

본 논문에서는 마이크로커널 개념을 살펴본 후 Nucleus와 μ ITRON를 비교하여 마이크로 커널의 특성을 살펴본다.

2. 마이크로 커널의 특성

마이크로 커널은 스케줄러, 인터럽트 관리와 같은 가장

중요한 핵심 기능만 커널에 포함하고 나머지 기타 기능은 응용 프로세스 또는 서버 형태로 구성되거나 모듈 형식으로 제공하며 이것을 제어하는 기능만 갖는다. 기본적으로 프로세스 관리, 프로세스간 동기 및 통신, 시스템 콜 및 기본 입출력 제어 기능 등을 모듈로 구성된다.

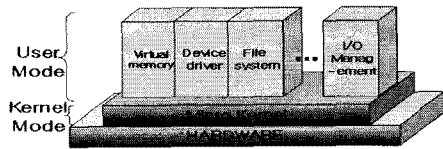


그림 1. 마이크로 커널의 기본 구조

이런 마이크로 커널은 새로운 기능을 추가할 때 커널의 모듈 또는 컴포넌트만 추가하므로 확장성(Extensibility)이 범용 커널에 비해 크다. 또한 마이크로 커널은 프로세서 종속(하드웨어 의존적인 코드) 코드가 마이크로 커널 내에 있기 때문에 새로운 처리기에 이식할 때는 이러한 하드웨어 의존적인 부분만 추가하면 되기 때문에 이식성(Portability)이 높다. 그러나 마이크로 커널은 사용자 모드에서 수행되는 다양한 서버로부터 커널 시스템 호출에 따른 모드 전환 및 프로세스 교환(switching)을 유발하는 경우로 인해 성능이 감소한다. 결국 마이크로 커널은 확장성, 이식성을 높이면서

퍼포먼스 감소를 최소화 하는 것이 중요한 이슈가 된다.

3. Nucleus와 μ ITRON 특성 및 성능 비교

이전부터 마이크로 커널을 표준화하려는 노력이 꾸준히 진행되어왔다. 이런 노력의 하나로 일본의 내장형 실시간 시스템용 마이크로 운영체제 규격 개발 프로젝트인 TRON(The Real-time Operating system Nucleus)이 있다. μ ITRON는 1984년 일본 정부의 지원 아래 사카무라 켄 교수를 중심으로 추진해온 실시간 운영체제 프로젝트인 TRON 중에서 내장형 실시간 운영체제(Embedded real-time OS)의 마이크로 커널 사양(Specification)을 말한다. 반면에 Nucleus는 미국 ATI사에서 개발된 마이크로 커널이다. 이 논문에서는 μ ITRON 4.0 사양의 커널과 Nucleus의 구조와 성능을 측정한다.

3.1 기능 확장성 및 이식성 분석

일반적으로 모노리틱 운영체제에서 새로운 기능을 추가하기 위해서는 커널을 바꾸어야 하는 단점을 가지게 된다. 이에 비해 마이크로 커널 기반 운영 체제에서는 런타임에 커널의 기능을 새로 추가 하고 삭제가 가능하다. Nucleus와 μ ITRON는 멀티 서버(Multi Server)방법을 이용하여 기능을 추가한다. 이때 메시지가 전달되는 패스의 각단계에서 유저/커널 모드 변환, 주소 영역 변환이 많아지고 그에 따르는 오버헤드로 인해 전체적인 운영체제의 성능이 저하된다. 이런 성능저하의 개선은 다음 시간 제약성 비교 분석에서 논의한다.

Nucleus는 5개의 TASK 상태(executing, ready, suspended, terminated, finished)를 기반으로 동적으로 TASK를 생성·삭제가 가능하고 세마포어(semaphore), 이벤트(event), 큐(queue), 메일박스(mail box)외에 신호(signal)와 파이프(pipe)라는 독자적인 동기·통신 기능이 포함되며 입출력 드라이버(I/O Driver) API를 지원하여 기능을 확장할 수 있다. 반면에 μ ITRON는 보다 많은 7개의 TASK 상태(Running, Ready, Waiting, Suspended, Waiting_suspended, Dormant, non_existent)를 기반으로 하여 170개의 API를 지원하고 시스템 관련 API(System State Management)와 예외

Task Management	16	26
Task Exception Handling	0	8
Synchronization and Communication	48	77
Memory Management	14	22
Time Management	9	6
System State Management	0	15
Interrupt Management	11	11
System Configuration Management	0	5
I/O Driver	9	0
Total	107	170

표 1. Nucleus와 μ ITRON의 API비교

상황(exception) API, 다양한 Timer handler 지원으로 시스템 확장 기능을 지원한다. 표1은 두 커널의 기능 API를 보여준다.

이식성을 높이기 위해서는 커널의 프로세서 의존적인 부분을 줄이고 중요한 기능을 제외하고 모두 컴포넌트로 구성해야 한다. Nucleus는 프로세서 의존적인 파일이 프로세서 초기화 및 인터럽트 처리 파일, 스케줄러 파일, 타이머 처리 파일 등 총 8000라인 이하의 세 개 파일만 존재하여 프로세서 변경시 세 개의 파일만 수정할 수 있게 되어있다. 반면에 μ ITRON는 프로세서 의존적인 파일에 대한 규정이 없다. 표 2는 ARM7환경에서 컴포넌트를 제외하고 측정된 커널의 크기를 보여준다. 기본적인 커널의 기능을 남긴 채 대부분의 기능을 컴포넌트로 만들어서 커널 크기를 줄이고 필요한 컴포넌트만을 추가해 나갈 수 있도록 구

kernel size	18	11
		(단위 : Kbyte)

표 2. Nucleus와 μ ITRON의 커널 크기 비교

성함을 알 수 있다. 인터럽트 핸들러 부분에서 Nucleus와 μ ITRON는 문맥저장과 레지스터 처리부분을 제외한 인터럽트 처리부분을 유저레벨 서버로 올림으로 이식성을 높였다.

3.2 시간 제약성 비교 분석

리얼타임 운영체제에서는 외부의 이벤트에 대해 얼마만큼의 지연시간 안에 서비스를 해줄 수 있는가(interrupt response time) 그리고 낮은 우선 순위의 작업 중에 우선 순위가 높은 작업을 위해서 얼마나 빠른 시간안에 프로세서를 할당해 줄 수 있는나(preemptibility) 등이 중요한 문제가 된다.[5] 마이크로 커널에서는 커널 내에 몇 개의 필수적인 기능 모듈만을 가지고 있으므로 커널 내부의 작업 실행시간이 모노리틱 운영체제에 비해서 상당히 줄고 또한 이러한 실행시간을 예측하는 것이 쉬워진다.

Nucleus는 우선순위는 0-255 level로 지원하며 TASK 생성시 선점성을 부여할지를 결정할 수 있다. 반면에 μ ITRON는 TASK 생성시 기본적으로 선점성을 갖고 있으며 우선 순위는 0-16 level 이상으로 정의하고 있다. 두 커널 모두 동일 우선순위에서는 FCFS(First Come First Served) 방식에 의해 스케줄링(scheduling)을 실시한다. 또한 복수의 TASK를 공평하게 실행하려는 TSS(Time

Sharing System) 스케줄링 방식인 라운드 로빈(round robin) 방식을 지원한다. 이는 동일 우선순위에서 타스 크에 일정시간을 보장하고 프로세스를 할당하여 할당된 시간이 지나면 그 타스크는 잠시 보류한 뒤 다른 타스 크에게 기회를 부여하는 운영방식이다.

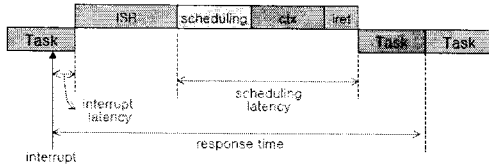


그림 2. 인터럽트 시간 지연

Nucleus는 인터럽트 지연 시간을 줄이기 위해 인터럽 트가 발생했을 때 IRC(Interrupt Request Controller) 조작 및 문맥(context) 저장등의 기본적인 처리를 하는 부분(LISR)과 실제로 개발자 레벨의 인터럽트 관련 처리 를 하는 부분(HISR)으로 구분한다. 이것은 가장 기본적인 처리부분을 어셈블러로 만들어 지연 시간을 줄이고 실제로 처리 부분을 타스크처럼 실행시키는 개념을 도입 한 것이다. 이 인터럽트 서비스 루틴은 타스크보다 높은 우선순위를 갖고 하나의 타스크 처럼 처리된다. μ ITRON 도 마찬가지로 기본적인 처리만 하는 interrupt handler 와 개발자 레벨에서의 처리를 하는 interrupt service handler로 구분하여 적은 시간안에 인터럽트 핸들러로 기본처리 한 후 유저레벨로 서비스 처리를 넘긴다.

그림 3은 Nucleus plus와 μ ITRON4.0 를 ARM7T프로세서 에서 측정한 인터럽트 지연시간과 문맥교환 시간 (context switching time)을 나타낸다. Clock speed는 75MHz이며 Cache는 enable 된 상태에서 측정 하였다.

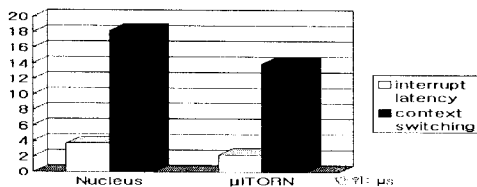


그림3. 성능 측정 결과

Nucleus는 ATI사에서 제공한 Timing sheet를 이용하였 고 μ ITRON는 커널 소스가 아니라 사양이기 때문에 μ ITRON4.0 사양에 맞게 마이크로 커널의 스케줄러, 인터 럽트 관리, 동기 통신 관리 부분을 구현하여 위의 두 가 지 기준으로 측정한 것이다. 그림3 에서 보는바와 같이 Nucleus와 μ ITRON이 적은 인터럽트 지연시간과 문맥 교

환 시간을 보여주고 있다. 프로세서에 차이가 있지만 대 체적으로 VxWorks 와 같은 리얼타임 커널의 인터럽트 지 연시간과 문맥 교환시간이 각각 2 ~ 6 μ s , 10 ~ 170 μ s임 을 감안하면 상대적으로 성능저하를 개선한 커널이라고 말할 수 있다. 또한 문맥교환 시간과 인터럽트 지연시간 은 일정하게 측정된 수치로 이것은 예측가능성을 보여준 다.

즉 인터럽트 지연시간치는 인터럽트가 발생하고 기본적인 처리를 하는 커널 레벨의 처리의 시간 값으로 인터럽트 핸들러와 관계없이 일정하여 실행 처리에 대한 예측이 가 능하다. 그러나 μ ITRON은 단지 사양만 존재하여 구현하는 개발자에 따라 다소 차이가 날 수 있음을 명시한다.

4. 결론

본 논문에서는 마이크로 커널의 특성에 대해 알아보고 그 성능을 Nucleus와 μ ITRON를 통해 확인해 보았다. 마 이크로 커널은 여러 장점에도 불구하고 다양한 서버로부터 커널 시스템 호출에 따른 모드 전환 및 프로세스 switching을 유발하는 경우로 인해 성능이 감소 한다. 결국 성능을 향상하기 위해서는 인터럽트 지연시간과 문 맥 교환시간을 줄이고 일정하게 하는 방법 밖에는 없다. 그런 관점에서 Nucleus와 구현된 μ ITRON는 작은 인터럽 트 지연시간과 문맥 교환시간을 실현함으로써 성능 저하 를 개선했다고 볼 수 있다. 또한 커널의 기본 크기를 줄 이고 90% 이상을 컴포넌트로 구성하여 이식성과 기능 확 장성을 보장하고 있다.

[참고 문헌]

- [1] D. Pan, "A tutorial on mpeg/audio compression," IEEE Trans. on Multimedia, Vol.2, No.2, pp.64-74, 1995.
- [2] Yu-Chi Chen, Chien-Wu Tsai, and Ja-Ling Wu, " Fast time-frequency transform algorithm and their applications to real-time software implementation of Ac-3 audio codec, " IEEE Transactions on Consumer Electronics, Vol. 44, No. 2, March 16, 1998.
- [3] J. Ready, "Real-time Standards," COMPUTER DESIGN, pp.21-23, Nov. 1990.
- [4] H. Nakamura and S. Kosai, "OS Functions for Testing Software," JC-CNSS'94 Proceedings, pp.13-18, Jul. 1994.
- [5] Soon-seop Yang, Young-woong Ko, Chuck Yoo, " Study of Microkernel supporting Multimedia"