

J · BOSN RTOS 개발하기

나노 커널(Nano-Kernel)

J · BOSN 실시간 운영체제는 고성능의 멀티-태스킹(Multi-tasking)을 지원하고 매우 강력한 실시간 응용프로그램을 개발할 수 있는 바탕을 제공하여 준다. 고성능의 실시간 커널(나노커널)과 다른 서비스모듈(마이크로커널, 매크로커널)이 결합되어 전체시스템이 이루어진다. 모든 모듈은 각자의 독립성을 최대한 보장하고 있으며, 각 모듈은 실시간 시스템의 서비스를 확장시킨다. J · BOSN 운영체제에서 기본적으로 제공하는 모듈 등을 통해 실시간 운영 체제에 대해 자세해 소개하고자 한다.

자료제공 : 아이보슨시스템즈
www.jbosn.com

연재순서

- ① 나노커널(nano-kernel)
- ② 시간서버(Time Server)
- ③ 시스템서버(System Server)

- ④ 동기화서버(Sync. Server)
- ⑤ 장치서버(Device Server)
- ⑥ 파일시스템서버(FileSystem Server)
- ⑦ 윈도우서버(Window Server)
- ⑧ 네트워크서버(Networ Server)

JBOSN RTOS의 구조는 그림 1에 나타나 있다, 구조적인 측면에서 보면 그림 1에서 나타나듯이 계층적이고 모듈화되어서 각 모듈들을 통합하는 것이 나노커널의 목적이다. 운영체제의 측면에서 보면, 나노커널은 운영체제의 근간을 구성하고, 전체 JBOSN RTOS를 통합하기 위한 필수적이고, 근본적인 기능과 하드웨어(CPU+Memory)와의 연결을 담당하고 있다.

JBOSN RTOS의 설계원칙은 임베디드 시스템에 적용하기 쉽고, 다양한 시스템에 최적의 시템을 설계할 수 있도록, 작고 빠르면서 빠른 응답속도 및 레고처럼 조합이 가능한 설계이지만, 설계시 필요한 모든 기능을 지원하는 것이다.

커널은 멀티레이어 구조를 가지고 있다. 모든 기능은 모듈화되어 독립적으로 구동이 되며, 이식성이 뛰어나며, 특정한 기능들을 강화할 수 있는 확장성이 뛰어난 특성을 가지고 있다.

멀티태스킹이 가능한 구조이며, 하나의 태스크에 여러 개의 쓰레드가 동작하도록 설계가 되어 있다.

또한 이 독립된 쓰레드나 태스크 간의 유연한 통신을 제공하기 위하여, 유연하고 다양한 IPC(Inter-process communication) 기능들을 제공한다.

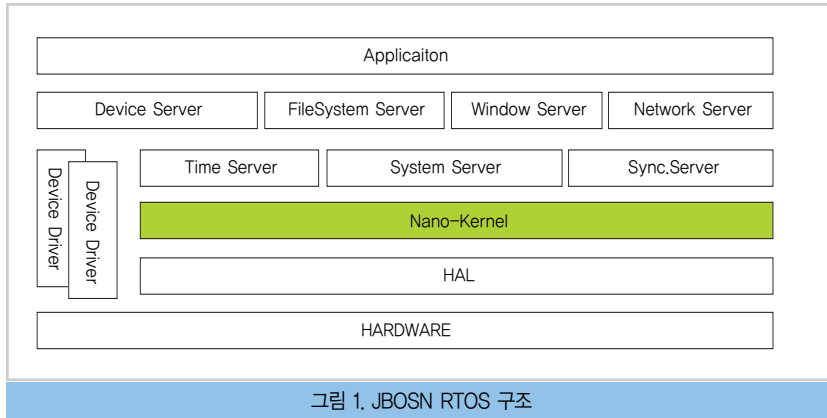


그림 1. JBOSN RTOS 구조

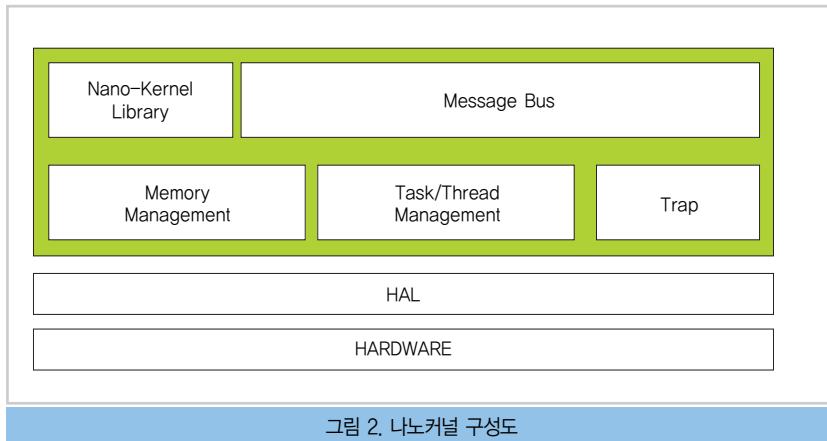


그림 2. 나노커널 구성도

스케줄링은 ‘precise timer and priority based real-time scheduler’ 방식을 채택하고 있으며, 동일한 priority를 가지고 있는 스레드는 라운드-로빈 방식의 스케줄링을 사용한다.

리얼타임 운영체제의 빠른 반응속도를 구현하기 위한 구조를 가지고 있어서 하드 리얼-타임 특성까지 지원이 가능하다. 이번호는 이러한 특성을 구현하기 위한 근본적인 개념을 담고 있는 나노커널의 설계와 구성 원리를 살펴보겠다.

나노커널

나노커널은 JBOSN RTOS의 모든 기능을 구현할 수 있는 근간을 제공하는 곳으로 가장 하단에서 서비스를 제공하여

주는 커널의 한 부분이다. 상위의 커널기능을 하는 모듈들이 유기적으로 결합하기 위한 방법을 제공하며, 하드웨어와 직접적으로 연결되는 부분이다. 커널의 다른 부분들은 하드웨어와 직접적으로 연결이 되어 있지 않으며, 반드시 나노커널을 통하여 하드웨어와 연결이 된다. 나노커널은 크게 두 부분으로 나뉘어 있는데, 스레드의 관리 기능, 모듈들 간의 통신을 지원하는 상위 커널 모듈들을 위한 기능을 담당하는 부분과 하드웨어와 연결하여 특정 하드웨어에서 커널을 동작하도록 대응하는 부분이다.

하드웨어 연결

하드웨어와 연결하는 부분은 CPU와 메모리를 고려하여야 한다.

주목할 부분은 나노커널은 하드웨어의 변경에 따라 재컴파일 되거나, 기능의 추가/삭제 및 변경은 전혀 이루어지지 않는다. 하드웨어에 따른 변경은 모두 ‘HAL layer’에서 제공하게 되며(porting guide 참조), 나노커널이 하드웨어에 접근하는 것은 모두 ‘HAL layer’를 통하여 이루어진다. 그림 2은 나노커널의 구성을 간략히 나타내고 있다.

“Trap” 부분은 하드웨어에서 발생하는 exception 기능(인터럽트, error reports 등)의 상황을 처리한다. 가장 대표적인 것이 하드웨어의 인터럽트가 발생할 때, 정하여진 순서의 기능을 실행하여 인터럽트에 대응할 수 있도록 조절하여 준다.

빠른 응답을 위하여 인터럽트가 발생하면 다른 인터럽트가 발생하지 못하게 되는 시간을 최대한 단축할 수 있는 설계 구

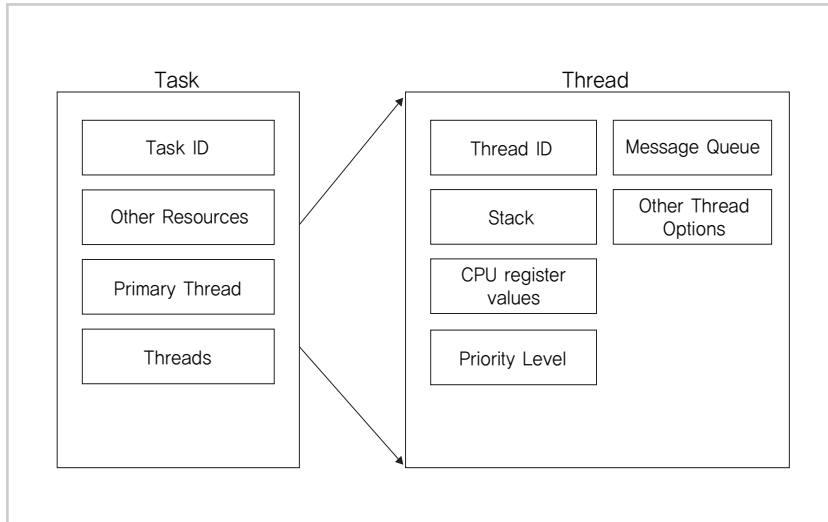


그림 3. 태스크와 쓰레드의 구성

조를 가지고 있으며, 인터럽트에 priority 레벨을 부여하여 중요한 인터럽트는 빠른 응답을 받을 수 있도록 구현이 되어 있다.

또 다른 것은 시스템 콜을 처리하는 기능이다. 나노커널을 이용하는 모든 모듈들에서 나노커널의 기능을 이용하기 위하여 함수형태로 호출을 하여야 한다. 이러한 접근도 exception 기능을 이용하여 나노커널을 접근할 수 있도록 설계되어 있다. JBOSN RTOS는 빠른 응답속도를 보장하기 위하여 system call을 두 가지(system call, system function)로 나누어서 정의하고 설계하였다. 첫째로 “System call”은 나노커널의 핵심적인 기능이며 실행하는데 상당한 시간을 요구하는 기능을 요청할 때 사용된다. “System call”이 호출되면 나노커널로 진입하기 위한 약간의 작업을 한 후, 곧바로 나노커널의 함수가 실행이 되는데, 실행되는 중간에도 외부 인터럽트를 받을 수 있다. 이 특성은 빠른 응답을 보장하기 위한 설계이다.

하지만 나노커널에 진입하거나 빠져나올 때 약간의 시간이 걸리는 단점이 있다. 둘째로, “System Function”은 일반 함수 호출과 동일하다고 생각하면 된다. 단점은 나노커널에서 이 함수를 수행하는 동안 외부 인터럽트는 받을 수 없는 상태가 유지된다. 그러나 이 함수는 매우 작은 수행시간을 요구하

여, 빠른 응답속도에 방해를 주지 않으며, 나노커널에 진입하거나 빠져나올 때 일반 함수 호출과 거의 동일한 성능을 보장한다.

“Memory Management” 부분은 하드웨어의 메모리의 관리를 담당한다. 이 부분도 빠른 응답과 나노커널의 부하(load)를 적게 하기 위하여 메모리관리에 필요한 기본 기능만을 제공하며, 전체적인 메모리관리는 “System Server”에서 담당한다. 메모리 사용자는 “System Server”에게 요청을 하면, “System

Server”는 나노커널의 이 부분을 이용하여 메모리의 관리 및 제어를 하게 된다.

결론적으로 메모리 관리에 대한 나노커널의 부하를 매우 적게 하여 시스템이 최상의 성능을 유지하게 한다. 여기까지가 하드웨어를 제어하는 부분이다.

상위 커널모듈 지원

상위 모듈들을 위한 기능으로 “Task/Thread management” 부분은 JBOSN RTOS의 스케줄링을 담당하고 있는 부분이다. 등록되어 있는 쓰레드는 이곳에서 CPU 자원을 획득할 수 있는 제어를 받는 부분으로 실시간 운영체제의 핵심 사항이다.

나노커널의 크기를 최소화하고 빠른 응답을 위하여 쓰레드 관리의 모든 기능을 담당하지는 않으며, 핵심사항인 ‘scheduling/dispatch’만을 담당하고 있다. 나머지 기능은 “System Server”에서 담당을 하고 있는 구조로 되어 있다. 시간 및 기타 관련된 기능은 “Time Server” 및 해당하는 기타 “Server”에서 담당한다. 그림 3은 JBOSN RTOS의 태스크와 쓰레드의 구성을 나타내고 있다.

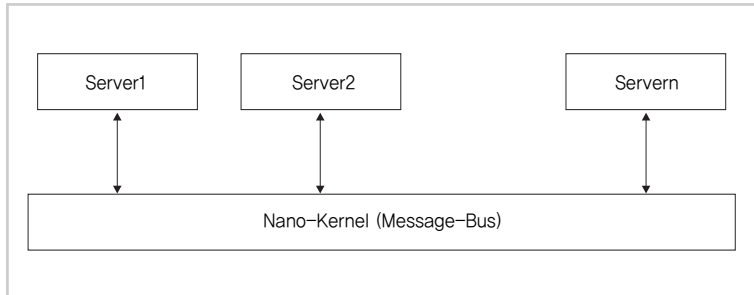


그림 4. 메시지 버스의 개략도

태스크는 여러 개의 쓰레드로 이루어지고 있고, 태스크와 쓰레드의 생성은 제한이 없으며, priority 레벨은 총 256개까지 가능하다. 쓰레드의 구성은 메시지큐가 존재하여 “Window Server”에서 사용을 하고 있다. 특이한 것은 쓰레드나 태스크에 힙이 존재하지 않는다.

“Nano-Kernel library”는 상위 커널모듈들이 나노커널에 있는 자료구조를 참조할 때 사용하는 라이브러리이다. 응용 소프트웨어에서는 사용할 수 없도록 되어 있으며, 커널모듈에서만 제한적으로 사용이 가능하다. 이 라이브러리는 이전에 설명한 “Trap” 부분을 사용하지 않고 직접적으로 나노커널에 접근이 가능하며, 매우 제한적으로 나노커널의 자료구조에 접근할 권한을 제공한다.

“Message Bus”는 상위모듈들이 통신을 할 수 있는 기능을 제공한다. 이 기능은 “Synchronization Server”에서 제공하는 IPC 기능과는 다르며, 독립되어 있는 상위 모듈의 통합을 위한 도구이다. 메시지 버스를 이용하여 모듈들은 하나의 유기적인 기능을 보장받는다. 구성의 원리는 매우 단순하다. 그림 4에서 보듯이 메시지 버스에서 보면, 상위 모듈들은 하나의 통신을 요청하고 전달받는 노드로 생각할 수 있다. 예를 들면, 데스크톱 PC를 생각해보자. PC의 기능을 확장하기 위해서는 PCI 버스에 특정기능의 각종카드(그래픽카드, 통신카드 등)를 장착하여 조립을 하듯이, PCI 버스에 대응되는 메시지 버스에 각종카드에 대응되는 기능 모듈(“Server”)들을 장착하는 것과 동일하다.


메시지 버스를 완벽하게 구현하기 위하여 노드인 모듈에서

포트와 채널이라는 두개의 핸들을 제공하여야 한다. 포트는 각 노드에서 데이터의 송/수신을 위한 자료구조체의 핸들이고, 채널은 노드에서 데이터를 전송할 때, 수신할 노드의 핸들을 나타낸다.

예를 들면, 미국 LA항에서 한국 부산항으로 화물을 전달하려면, LA항과 부산항이 존재하여야 한다. 보내는 화물에는 출발지와

목적지가 기록된 수화물표가 부착이 된다.

비교를 하면, LA항과 부산항은 데이터를 송/수신할 수 있는 포트에 대응되고, 수화물표의 도착지표시는 채널에 대응되는 것과 동일하다.

다음호에는 마이크로커널 중 “Timer Server”에 대하여 알아보겠다. 

**언제, 어디서나
빠르고 손쉽게
Embedded World 를 만나자!**

Embedded World 의 특징은 제작비용, 및 유통비용, 판매가격이 상대적으로 인쇄책자보다 저렴하며 수정 및 재가공이 가능합니다. 또한 멀티미디어 기능이 있어 표현이 다양하고 광고효과가 뛰어난 장점을 지니고 있습니다.

인터넷에서 읽는 Embedded World,
e-Magazine 서비스

☎ 02-2026-5700
www.Embeddedworld.co.kr