

J·BOSN RTOS 개발하기⑥

매크로 커널-파일시스템 서버

J·BOSN RTOS에서 파일시스템을 사용하려면 파일시스템을 파일시스템 서버(FileSystem Server)에 반드시 연결시켜야 하며 애플리케이션에서는 이 파일시스템 서버의 도움으로 파일에 접근을 할 수가 있도록 되어 있다. 이러한 기능은 실시간 시스템을 설계할 때 시스템에 최적으로 맞는 구현할 수 있는 해결책을 제시할 수 있다. 사용자는 파일시스템을 구현할 때, 버퍼를 포함한 파일시스템과 관련된 모든 데이터 구조를 직접 관리하고, 접근할 수 있으므로, 제작하는 시스템의 크기, 성능에 따라서 내부변수를 직접 제어하여 최적의 시스템을 만들 수 있다.

글 : 정병오 대표 / 아이보슨시스템즈 www.jbosn.com

연재 차례

- | | |
|-------------------------|-------------------------------|
| 1. 나노커널(nano-kernel) | 5. 장치서버(Device Server) |
| 2. 시간서버(Time Server) | 6. 파일시스템서버(FileSystem Server) |
| 3. 시스템서버(System Server) | 7. 윈도우서버(Window Server) |
| 4. 동기화서버(Sync. Server) | 8. 네트워크서버(Network Server) |

J·BOSN RTOS의 구조는 그림 1에 나타나 있다. 구조적인 측면에서 보면 그림 1에서 보여지 듯 계층적이고 모듈화되어 있는 각 모듈들 중에 JBOSN RTOS의 DISK 관리 및 파일관리 관련된 기능을 제공하는 목적으로 제작된 것이 파일시스템 서버(FileSystem Server)이다. 운영체제의 측면에서 보면, 파일시스템 서버(FileSystem Server)는 DISK 장치와 파일관리와 관련된 기능을 제공해주는 운영체제의 필수적인 부분이다.

JBOSN RTOS의 구조는 나노커널을 기본으로하여 마이크로 커널(시스템서버, 시간서버, 동기화서버)을 사용하면 순수한 운영체제의 핵심기능을 제공해준다. 매크로커널은 외부확장기능을 담당한다. 이 곳은 디바이스 관리와 통신을 위한 디바이스 서버, 블록디바이스의 데이터를 관리해 주는 파일시스템 서버인 File System Server, 네트워크 통신을 담당하고 있는 Network Server, 그래픽과 Window 시스템의 제어하여 GUI를 제공하여 주는 Window Server 등 4가지로 이루어져 있다.

파일시스템 서버를 사용하여 부가적으로 타깃시스템에서 사용하려는 DISK 디바이스를 관리하고 파일을 관리하여 데이터를 애플리케이션에게 제공하여 주는 부분을 담당하려는 확장이다. J·BOSN RTOS에서 파일시스템을 사용하려면 파일시스템을 파일시스템 서버(FileSystem Server)에 반드시 연

결시켜야 하며 애플리케이션에서는 이 파일시스템 서버의 도움으로 파일에 접근을 할 수가 있도록 되어 있다. 이러한 기능은 실시간 시스템을 설계할 때 시스템에 최적으로 맞는 구현할 수 있는 해결책을 제시할 수 있다.

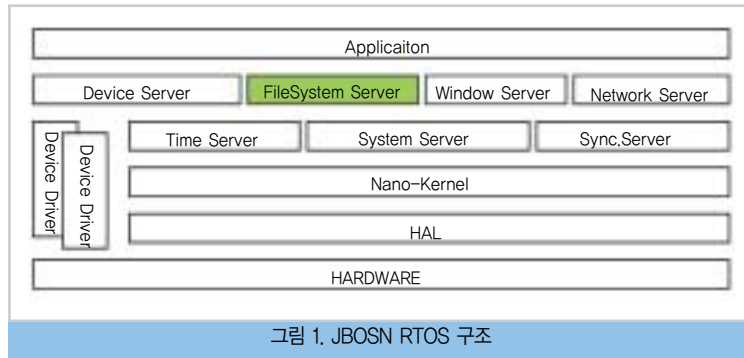


그림 1. J·BOSN RTOS 구조

파일시스템 서버(FileSystem Server)

이 모듈은 J·BOSN RTOS의 DISK 디바이스에 존재하는 파일을 관리하는 역할을 담당하는 서비스를 제공해 주며 서버의 형태를 갖고 있다.

파일시스템서버는 자료를 파일단위로 저장하거나 삭제하는 기능을 제공하여 준다. 파일을 저장하는 방법은 매우 다양하게 존재하여(FAT FS, ROM FS, RAM FS, ...) 임베디드 시스템에서 사용하는 것은 표준화되어 있지 않다.

제품을 제작할 때는 선택한 파일시스템만을 탑재하여 최적의 시스템개발이 필요하다. 파일시스템서버와 파일시스템을 포함하고 있다면, 위의 요건에 따라 사용자가 매번 원하는 파

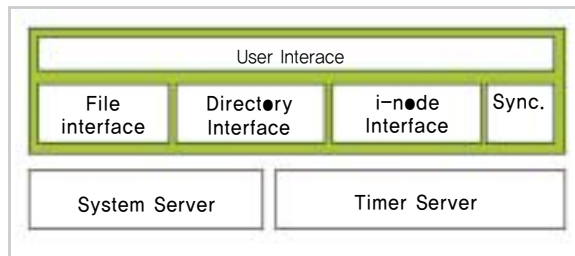


그림 2. 파일시스템서버 구성도

일시스템을 선택하거나 제작하여 파일시스템서버를 재 컴파일하거나 변경하여야 한다. 이것은 RTOS의 안정성을 해칠 수 있어서 시스템을 불안정하게 만들 수 있다. 또한 파일시스템서버와 파일시스템의 메모리나 코드가 공유되어서 문제가 발생할 때, 원인이나 해결방법을 찾기가 어렵다.

J·BOSN RTOS는 파일시스템서버가 특정 파일시스템을 포함하지 않고 파일을 다루는데 필요한 기본기능과 필요한 데이터구조와 모든 파일시스템을 수용할 수 있는 인터페이스만을 제공한다. 더 나아가, 파일시스템들을 사용할 때 속도향상과 기타 목적으로 사용하는 대용량의 데이터 버퍼들도 제공하지 않는다. 사용자는 파일시스템을 구현할 때, 버퍼를 포함한 파일시스템과 관련된 모든 데이터 구조를 직접 관리하고, 접근할 수 있으므로

로 제작하는 시스템의 크기, 성능에 따라서 내부변수를 직접 제어하여 최적의 시스템을 만들 수 있다.

파일시스템서버는 파일시스템의 인터페이스와 관리만을 담당하여, 매우 안정적인 시스템을 구축할 수 있고, 파일시스템에서 발생할 수 있는 에러나 논리적인 문제에서 벗어날 수 있다.

다음은 파일시스템서버(FileSystem Server)에서 제공하는 기능에 대하여 자세히 살펴보겠습니다.

파일시스템 계층구조

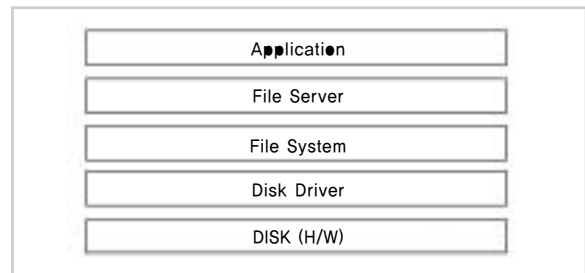


그림 3. 파일시스템서버의 계층적 구성도

파일시스템서버는 파일시스템과 연결되어 있어서 하드웨어에 해당하는 DISK에 직접 접근하지 않는다. 따라서 하드웨어의 불안정성으로부터 파일시스템서버는 영향을 받지 않는다. 하지만 파일시스템에서 발생하는 에러에 대해서는 에러를 검출할 수 있는 부분을 가지고 있지 않으므로, 파일시스템을 제작할 때, 에러처리를 확실히 하여야 한다. 파일시스템은 하드웨어를 직접 접근하지는 않지만 DISK 디바이스 드라이버를 통하여 하드웨어에 접근하게 한다. 하드웨어 에러에 대하여 직접적인 영향을 받게 되어서 파일시스템 에러 검출을 반드시 하여야 한다. 이렇게 함으로써 파일시스템서버는 에러문제에서 자유로울 수 있으며, 특정 DISK나 파일시스템에서 발생하고 에러에 대하여 직접적인 영향을 받지 않으므로, 여러 가지 파일시스템과 DISK를 사용할 때, 특정 부분에서 발생하는 에러 때문에 시스템이 멈추는 현상을 방지할 수 있다.

이 구조처럼 파일시스템이 파일시스템서버와 분리되어 존재한다. J·BOSN RTOS는 파일시스템서버와 별도로 많은 파일시스템라이브러리를 제공하려 노력하고 있다.

파일시스템 서버의 자료구조

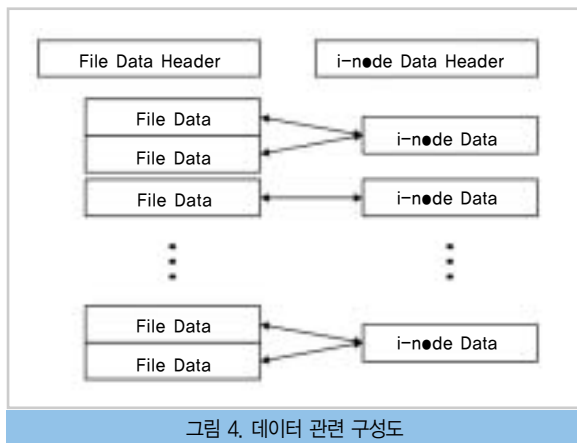


그림 4. 데이터 관련 구성도

파일시스템서버의 구조는 사용자를 위한 인터페이스와 파

일시스템을 관리할 때, 필요한 데이터의 종류로 나눌 수 있다.

-현재 사용되고 있는 파일과 관련된 정보를 관리하는 FILE 데이터 구조를 가지고 있다. 이 데이터는 사용자가 사용하고 있는 파일과 관련된 정보를 가지고 있다. 현재 파일과 관련된 보다 하드웨어적인 정보를 담고 있는 i-node 정보를 포함하여 현재파일을 읽거나 쓰는 위치 등 파일자체의 내용보다 사용자가 변경할 수 있는 모든 정보를 담고 있다. 사용자는 FILE 데이터를 언제든지 얻을 수 있으며, 사용자 인터페이스에 의해 내려진 명령들은 FILE 데이터를 참조하여 수행된다.

-i-node 데이터는 파일이 저장되어 있는 형태와 관련된 모든 정보를 가지고 있다. i-node 데이터는 파일의 종류, 크기, 저장된 장소, 관련된 날짜정보 등등 파일과 관련된 정보를 가지고 있다. 이 정보들은 파일시스템과 매우 관련이 깊어서, 파일 시스템을 제작할 때 i-node 데이터 정보를 읽는 부분을 추가하여 주어야 한다. i-node 데이터는 고정된 데이터이며, 파일을 지우거나 변경하지 않으면 항상 고정되어 있다.

-하나의 파일을 여러 개의 프로세스에서 동시에 사용할 경우 하나의 i-node 데이터를 여러 개의 File 데이터가 링크할 수 있다. 파일을 나타내는 i-node는 현재 자신을 사용하고 있는 File 데이터를 기억하고 있어야 하며, 사용자의 파일접근에 동시에 지원하여야 한다.

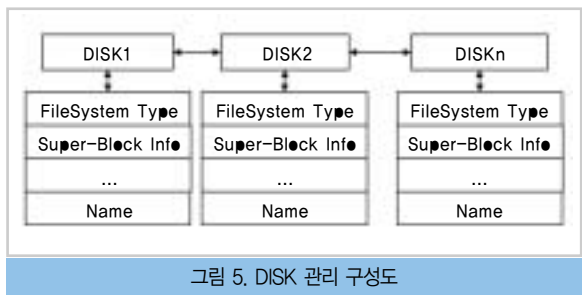


그림 5. DISK 관리 구성도

파일시스템 서버의 내부에서 사용되는 DISK들은 그림 5

에서처럼 등록된다. 디스크는 filesystem type과 이름 그리고 super-block의 정보를 담고 있다. 이 데이터 구조를 토대로 모든 디스크를 관리하고 있다.

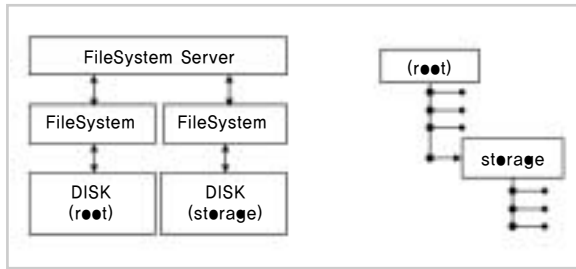


그림 6. 파일시스템 서버의 논리적 구성도

여러 개가 파일시스템을 동시에 사용하거나, 여러 개의 DISK를 동시에 사용할 때, 사용자에게 seamless filesystem을 제공하기 위하여 마운트(mount) 작업을 수행하게 된다. JBOSN RTOS는 root 파일시스템을 지정한 후, 다른 DISK를 mount시키면, 반드시 root directory 밑에 있는 디렉토리처럼 보이게 된다. 사용자는 루트 디렉토리(root directory)에 마운트된 DISK에 접근할 때, root DISK에 존재하는 디렉토리를 접근하는 방법과 동일하게 접근할 수 있다.

FAT 파일 시스템 예

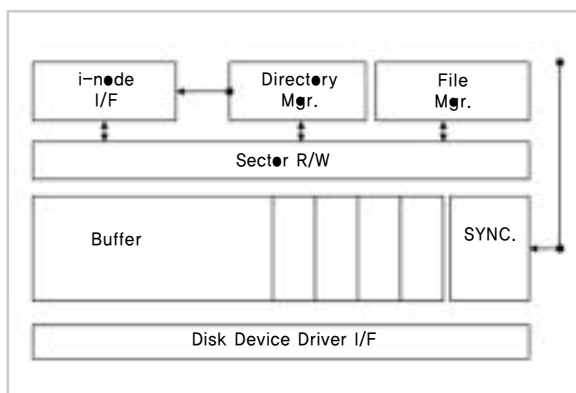


그림7. FAT 파일시스템 구성도

-i-node 인터페이스는 파일시스템의 파일 또는 디렉토리의 정보를 읽어 파일시스템 서버에서 필요한 i-node 데이터를 구성하는데 필요한 부분이다. 이곳엔 FAT table만을 참조하여 필요한 정보를 i-node table에 넣게 된다. I-node는 디렉토리나 파일의 생성과 소멸에 필요하므로 directory mgr 부분에서 호출되기도 한다.

-Directory Mgr 부분은 파일시스템의 생성/소멸 등을 관리하는 부분이다. 디렉토리에 들어 있는 파일 또는 디렉토리의 내용도 파일처럼 되어 있지만, 정규화된 directory entry 구조로 되어 있어서 별도의 file access routine을 가지고 있다.

-File Mgr 부분은 읽기/쓰기/변경/지우기 등의 행위를 할 때 사용된다.

-Sector R/W 부분은 섹터 단위로 DISK에 있는 데이터를 읽고 쓰는 부분이다.

-버퍼 부분은 DISK는 속도가 매우 느리게 동작하기 때문에, 미리 읽어서 데이터를 캐싱(caching)하여 놓는 곳이다. 이 곳에서 사용되는 버퍼의 크기는 사용자가 시스템을 작성할 때, 시스템에 따라 크기를 조절할 수 있다.

- Sync. 부분은 캐싱되어 있는 버퍼의 내용중 현재 DISK와 내용을 일치하지 않고 있는 데이터가 있을 경우, 캐싱 데이터와 DISK의 데이터의 내용을 일치시키는 기능을 하는 부분이다. 이 부분은 파일시스템서버에서 1초마다 한 번씩 호출되고 있다.

➔ 다음호에는 매크로커널 중 "Window Server"에 대하여 알아보겠습니다.