

Audio Player 를 위한 경량 실시간 운영체제 설계 및 구현*

조문행⁰, 이정원, 강희성, 이철훈
충남대학교 컴퓨터공학과
root4567⁰, booster, hskang, cleec@cnu.ac.kr

The Design and Implementation of Light-Weight Real-Time Operating System for Audio Player

Moon-Haeng Cho⁰, Jung-Won Lee, Hui-Sung Kang and Cheol-Hoon Lee
Dept. of Computer Engineering, Chungnam National University

요 약

임베디드 시스템은 특정 임무를 수행하도록 설계된 전용 컴퓨팅 시스템으로 그 용도에 따라 다양한 하드웨어 구성요소를 가지며, 그 쓰임새에 따라 특정 하드웨어 중심으로 시스템을 구현할 수 있다. 이런 하드웨어 시스템의 자원을 효율적으로 관리하기 위해서는 그 시스템에서 요구하는 기능을 만족시키는 특정 운영체제가 필요하다. 본 논문에서는 적은 크기의 메모리에 실시간 운영체제와 파일시스템, 애플리케이션이 모두 탑재되어야 하는 Audio Player 시스템을 위한 경량 실시간 운영체제를 설계 및 구현하였다.

1. 서 론

오늘날의 임베디드 시스템은 다양한 형태를 띄고 있으며, 제 각각의 특성을 가지고 있어 하드웨어 디자인도 그 특정 목적에 따라 달라진다.

특정 기능을 수행하기 위해 하드웨어와 소프트웨어를 밀접하게 통합시킨 컴퓨팅 장치를 임베디드 시스템이라 하며, 외부 이벤트에 대해 일정한 시간에 응답해야 하는 시스템을 실시간 시스템이라 한다. 이 두 가지 특성 모두를 가지는 시스템을 실시간 임베디드 시스템이라 한다 [2].

실시간 임베디드 시스템의 한정된 자원을 효율적으로 관리하기 위해 필요한 소프트웨어가 실시간 운영체제(Real-Time Operating System)이며, 특정 하드웨어로 구성된 임베디드 시스템에 맞게 실시간 운영체제를 설계 및 구현할 필요성이 있다 [2].

본 논문의 구성은 2 장에서는 관련연구로서 실시간 운영체제가 갖는 일반적인 구성요소를 UbiFOS™을 중심으로 소개하고, 3 장에서는 Audio Player 를 위한 실시간 운영체제의 설계 및 구현 내용을 제시하며, 4 장에서는 테스트 환경 및 결과를 기술한다. 마지막으로,

5 장에서는 결론 및 향후 연구과제에 대해서 기술한다.

2. 관련 연구

연성 실시간 운영체제인 UbiFOS™은 멀티 태스킹 환경을 지원하고 256 단계 우선순위 기반의 선점형 스케줄러를 제공하며, 동일한 우선순위에 대해 FIFO 와 라운드로빈 스케줄링을 제공한다. 또한 태스크에 관련된 태스크 관리기능과 메모리를 동적으로 관리하기 위한 메커니즘을 가지고 있으며, 태스크간 동기화를 위해 세마포와 이벤트 플래그를 태스크간 통신을 위해서 메시지 큐, 메시지 포트, 메시지 메일박스, 태스크 포트를 가지고 있다 [1][4].

2.1 태스크 관리 기능

태스크 생성에 필요한 스택을 전역변수를 통해 할당 받거나, 동적 메모리 관리를 위한 힙(Heap) 영역에서 할당 받아 태스크를 생성할 수 있다. 또한 필요에 따라 태스크의 삭제가 가능하고, 동적으로 태스크의 우선순위를 바꿀 수 있다 [1][2].

본 연구는 정보통신부의 선도기반기술개발사업의 지원으로 수행되었습니다.

2.2 동적 메모리 관리

동적 메모리 관리를 위해 가변 크기의 메모리를 할당, 해제할 수 있는 힙 스토리지 매니저(Heap Storage Manager)와 고정 크기의 메모리를 할당 및 해제할 수 있는 메모리 풀(Memory Pool)을 제공한다 [3].

2.3 태스크간 동기화

태스크간 동기화를 위해 세마포(Semaphore)와 이벤트 플래그(Event Flag)를 제공한다. 세마포는 바이너리 세마포와 카운팅 세마포가 있다. 또한 바이너리 세마포의 특수한 경우로 우선순위 역전현상(Priority Inversion)을 해결하기 위한 우선순위 상속(Priority Inheritance), 삭제 안전장치(Deletion Safety), 재귀(Recursion)를 제공하는 상호 배제 세마포가 있다 [1][4].

2.3 태스크간 통신

독립적인 태스크들 사이에 정보를 주고 받기 위해서 메시지 메일박스(Message MailBox)와 메시지 큐(Message Queue), 메시지 포트(Message Port), 태스크 포트(Task Port), 시그널(Signal)을 제공한다 [4].

3. Audio Player 시스템을 위한 실시간 운영체제

3.1 Audio Player 시스템을 위한 실시간 운영체제의 필요성

타겟 임베디드 시스템은 Digital Audio Player 로 음악 파일만을 구동시키기 위해 특별히 제작된 시스템이다. 이 시스템은 Calm16 CPU Core 와 적은 크기의 Mask Rom, 부트로더를 위한 NOR Flash, 적은 전력소모를 위해 512KB 의 SRAM 메모리 그리고 저장 장치로 NAND Flash 를 갖고 있으며 펌웨어로 구동이 된다. 기존의 펌웨어가 해당 시스템에서만 구동된다는 한계를 가지고 있었기 때문에 ARM CPU 계열과 같이 다른 타겟 시스템을 기반으로 하기 위해서는 실시간 운영체제로 구동될 필요성이 제기 되었다. 실시간 운영체제를 인식하기 위한 가장 큰 이슈는 SRAM 의 용량을 확장할 수가 없다는 것으로 이는 SRAM 의 가격과 크기가 Audio Player 에서는 문제가 되기 때문이다.

제한된 크기의 SRAM 영역에 크기가 21KB 인 기존 UbiFOS™ 실시간 운영체제와 크기가 26KB 인 NAND Flash 를 위한 플래시 파일시스템, 그리고 MP3, OGG,

WMA 파일 포맷을 지원하기 위한 코덱이 446KB, 코덱을 제외한 키 입력과 USB 기능 지원 등의 Audio 애플리케이션의 크기가 20KB 로 총 크기가 SRAM 의 용량을 넘어서는 것이 문제인데 파일 시스템과 코덱, Audio 애플리케이션에서 차지하는 메모리 공간을 축소할 수 는 없다. 이런 SRAM 의 제약으로 인해 시스템에서 필요로 하는 기능만을 갖는 경량화된 실시간 운영체제의 필요성이 제기되었다.

3.2 경량 실시간 운영체제의 설계 및 구현

기존 UbiFOS™은 관련 연구에서와 같이 실시간 운영체제가 갖는 모든 기능을 제공하였으나 타겟 시스템을 위해 구현된 실시간 운영체제는 다음과 같은 특징을 갖는다.

구현한 경량 실시간 운영체제는 우선순위 32 단계만을 지원하는 선점형 스케줄러로 이를 위해 비트맵을 사용하였다. 또한, 동일한 우선순위를 지원하기 위하여 FIFO 와 라운드 로빈 스케줄링을 제공한다.

동적 메모리 관리 기능은 제공하지 않으며, 특정 SRAM 의 메모리 영역에 버퍼를 생성하여 NAND Flash 에 Read/Write 할 수 있도록 함으로써 Play 시에는 음악 파일에 대한 디코딩 정보를 전역 변수인 ReadMemory 주소로 읽어오고 레코딩 시에는 WriteMemory 주소에 인코딩된 정보를 저장한다.

태스크간 통신과 동기화는 전역 변수를 통해 이루어지며, 상호 배제를 위한 바이너리 세마포만을 제공한다.

[그림 1]은 구현된 실시간 운영체제의 32 단계 우선순위와 태스크간의 통신을 위한 변수와 바이너리 세마포어를 제공하는 세마포 제어 블록이다.

MK_U32_t t_Priority;	// 32 단계의 우선순위를 제공하기 위한 비트맵
MK_U32_t *ReadMem;	// 음악 파일에 대한 디코딩 정보를 저장하는 메모리 주소
MK_U32_t *WriteMem;	// 파일에 대한 인코딩 정보를 저장하는 메모리 주소
MK_U32_t *ITCom;	// 태스크간 통신을 위한 전역 변수 주소
MK_U32_t ITSync;	// 태스크간 동기화를 위한 Flag

```
typedef struct mk_semaphore_struct {
    MK_U32_ts_Magic;
    mk_pending_list_struct s_PendingList;
    MK_S8_t *s_pName;
} MK_SEMAPHORE;
// 상호배제를 위한 바이너리 세마포어
```

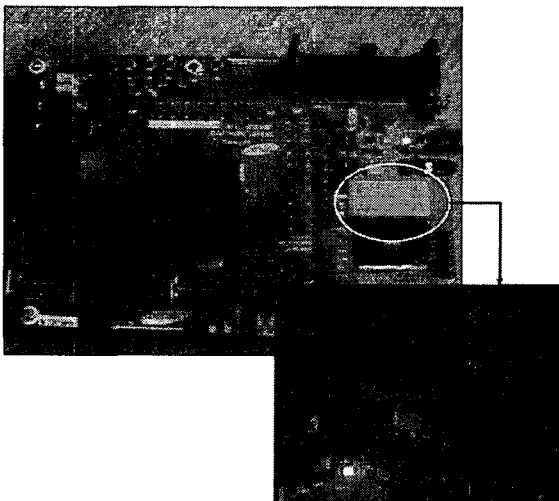
[그림 1] 경량 실시간 운영체제의 구조체와 변수

이와 같이 Audio Player 를 위해 필요한 기능만을 제공하여 512KB 의 제한된 메모리를 갖는 시스템에 적합한 경량 실시간 운영체제를 구현하였다.

4. 테스트 환경 및 결과

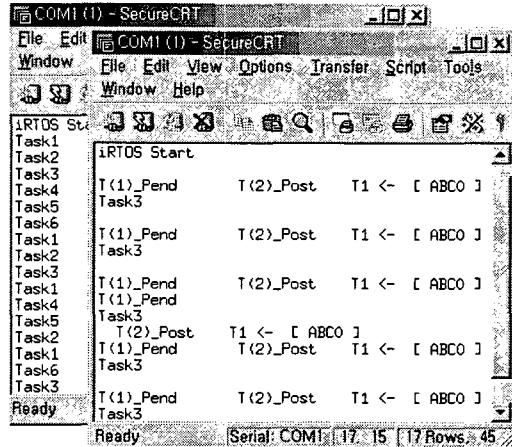
본 논문에서 구현한 경량 실시간 운영체제는 Digital Audio Player 인 삼성에서 제작한 Calm16 기반의 Orchestra Evaluation Board(S5L841F)에 탑재하였고, 컴파일러는 CalmShine16 Media 를 사용하였다.

[그림 2]는 실제 애플리케이션 프로그램을 경량 실시간 운영체제에 탑재하여 MP3 포맷의 음악파일을 실행시킨 결과이다.



[그림 2] 테스트 환경 및 결과

[그림 3]의 왼쪽 그림은 Audio 애플리케이션 구동을 위해 구현한 경량 실시간 운영체제에 대한 스케줄링 테스트로 6 개의 서로 다른 우선순위를 갖는 태스크를 생성하여 수행한 결과이며, 오른쪽 그림은 태스크간 통신을 테스트한 결과로 태스크 1 과 태스크 2 는 태스크 3 보다 높은 우선순위를 가지고 서로간에 전역변수를 사용하여 통신한다.



[그림 3] 테스트 환경 및 결과

이와 같이 구현된 실시간 운영체제는 실시간 운영체제의 기본 기능인 스케줄링 기능과 태스크간 통신 기능을 만족하였으며, 상용 Audio Player 애플리케이션이 필요로 하는 기능들만을 중심으로 구현된 경량 실시간 운영체제로 펌웨어를 대체할 수 있다.

5. 결론 및 향후 연구 과제

본 논문에서는 특정 애플리케이션을 위한 경량 실시간 운영체제를 설계 및 구현한 내용을 기술하였다.

구현한 경량 실시간 운영체제가 성능 보다는 실행 이미지 크기에 초점을 맞추어 설계 및 구현되었기 때문에 실제 시스템에서 요구하는 성능 문제 또한 고려해야 한다. 향후 연구과제는 타겟 임베디드 시스템을 위해 실행 이미지 크기가 작고 성능이 우수한 실시간 운영체제에 대해 연구하는 것이다.

참고문헌

- [1] Embedded System & RTOS, <http://www.aijisystem.com>.
- [2] David Stegner의 2명, " Embedded Application Design Using a Real-Time OS ", IEEE, 1999
- [3] 박희상, 안희중, 김용희, 이철훈, " Design and Implementation of Memory Management Facility for Real-Time Operating System, iRTOS™ ", 한국정보과학회, 2002
- [4] 박희상, 정명조, 조희남, 이철훈, "Design of Open-Architecture Real-Time OS Kernel", 한국정보과학회, 제 31권 제 1호, p.163-165, 2002.04.