

DVD Player 용 실시간 운영체제 설계 및 구현*

안희중⁰, 백대현, 성영락[†], 이철훈
충남대학교 컴퓨터공학과, [†]국민대학교 전자정보통신공학부
{hjahn⁰, dhbaek, chlee}@ce.cnu.ac.kr, [†]yeong@mail.kookmin.ac.kr

Design and Implementation Real-Time Operating Systems for DVD Players

Hee-Joong Ahn⁰, Dae-Hyun Baek, Yeong Rak Seong[†], and Cheol-Hoon Lee
Dept. of Computer Engineering, Chungnam National University
[†]School of Electrical Engineering, Kookmin Univ.

요 약

범용 컴퓨터가 아닌 홈 네트워크로 연결된 정보가전용 실시간 시스템에 내장되는 운영체제인 정보가전용 실시간 운영체제는 디지털 TV, 웹 TV, WebPad, PDA, Smart Phone 등에 사용된다. 그러나 정보가전용 실시간 운영체제는 선점형 스케줄링, 멀티 태스킹 및 태스크간 예측 가능한 동기화 방법 그리고 짧고 제한된 인터럽트 처리와 작은 실행 이미지 등과 같은 특징들을 갖는다. 본 논문에서는 이와 같은 특징들을 만족하는 정보가전용 실시간 운영체제를 DVD 플레이어에 적용하여 설계 및 구현한 내용을 기술한다.

1. 서 론

실시간 운영체제는 실행 결과의 논리적인 정확성과 미리 정의된 데드라인(Deadline)을 초과하지 않도록 시간적인 정확성을 보장하는 운영체제이다. 실시간 운영체제를 임베디드 시스템에 적용하여 정보가전 제품, 핸드폰 및 PDA 단말기, 공장 자동화 및 자동제어 그리고 첨단 특수 분야 등에서 적용되고 있다. 우선순위에 기반한 선점형 스케줄링, 멀티 태스킹 지원 및 태스크간 예측 가능한 동기화 방법 제공, 짧고 제한된 시간의 인터럽트 전달 지연과 처리, 저전력 관리 기법, 임베디드 시스템에 적합한 적은 이미지 크기, 다양한 정보가전에 적용할 수 있도록 조립형 커널 지원, 윈도우 기반의 GUI 지원 등과 같은 많은 필요조건을 만족하여야 한다.

본 논문은 이와 같은 조건을 만족하는 정보가전용 실시간 운영체제를 DVD Player 에 적용하여 설계하고 구현한 내용을 기술한다.

본 논문의 주요구성은 2 장에서 실시간 운영체제의 전체적인 구성을 설명하고, 3 장에서는 DVD Player 를 위한 실시간 운영체제의 설계 및 구현 내용을 설명하고, 4 장에서는 테스트 환경 및 결과를 설명하고, 마지막으로 5 장에서는 결론 및 향후 연구 과제에 대해서 기술하였다.

2. 관련 연구

2.1 실시간 운영체제

실시간 운영체제의 핵심은 태스크를 관리하고 태스크에 CPU 를 할당하는 태스크 스케줄링 기법이다. 따라서 모든 태스크는 중요도에 따라 우선순위를 할당 받으며, 스케줄러는 실행 준비된 태스크들 중에서 가장 높은 우선순위의 태스크를 실행 가능하도록 한다. 그리고 태스크간 동기화 및 통신을 위해 세마포, 메시지 큐, 메시지 메일박스 및 이벤트 플래그를 제공하고, 동적 메모리 관리 기법 및 타이머를 제공한다.

2.2 태스크 스케줄링 정책

모든 태스크는 생성시에 자신만의 스택 영역과 실행 함수, 0 부터 255 까지의 256 단계의 우선순위를 할당 받고 별도의 자료구조로 관리된다. 서로 다른 우선순위의 태스크에 대해서는 우선순위 기반의 선점형 스케줄링을 제공하고 동일한 우선순위의 태스크에 대해서는 라운드 로빈 방식으로 스케줄링 한다.

2.3 태스크간 통신 및 동기화

□ 세마포

한 자원에 대한 상호배제 접근방법과 안전한 공유 자원 관리와 태스크간 동기화에 사용한다.

□ 메시지 큐

특정 태스크나 ISR 에서 다른 태스크로 여러 개의 메시지를 전달할 때 사용한다.

□ 메시지 메일박스

*이 논문은 산업자원부 중기거점과제 연구비 지원에 의한 것임

메시지 큐의 특수한 경우로 하나의 메시지를 빠르게 전달할 때 사용한다.

□ 이벤트 플래그

세마포의 확장된 개념으로 여러 개의 이벤트에 대한 동기화를 제공한다.

□ 타이머

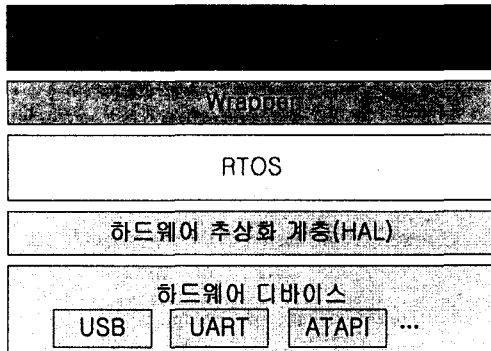
특정 시간 후에 특정 루틴을 수행하는 서비스

2.4 인터럽트 처리

내, 외부에서 인터럽트가 일어나면 현재 실행중인 태스크의 문맥을 저장하고 커널에서 필요한 인터럽트 처리 후 사용자가 정의한 인터럽트 처리 루틴을 실행한다. 그러나, 사용자가 정의한 인터럽트 처리 시간이 길어지면 다른 인터럽트가 지연되거나 심지어 무시 될 수 있다. 본 연구팀이 개발한 실시간 운영체제는 인터럽트를 LISR(Low-level ISR)과 HISR(High-level ISR)로 구분하고 LISR 에서는 인터럽트가 일어나면 해당 비트에 마크만 하고 실제 태스크 스케줄링에 앞서 태스크와 비슷한 구조인 HISR 을 먼저 수행 후 태스크 스케줄링 하여 인터럽트 처리의 지연을 방지한다.

3. DVD Player 용 실시간 운영체제 설계 및 구현

3.1 개발 구조



[그림 3-1] 개발 구조도

[그림 3-1]은 랩퍼(wrapper)를 사용한 개발 구조도를 보여주고 있다. 랩퍼는 응용프로그램의 개발을 실시간 운영체제와 독립적으로 개발할 수 있도록 하며, 실시간 운영체제의 이식성을 높여주는 역할을 한다. 랩퍼에서 정의한 API 함수와 본 연구팀에서 개발한 실시간 운영체제의 API 함수들 간의 사상(mapping)한다.

3.2 태스크 생성

```

SysCreateTask(
    *task /* TCB */, *name /* Name */,
    entry /* entry function */,
    argc, *argv, /* entry function argument */
    stack_size /* stack size */, priority /* priority */,
    time_slice /*time slice*/, preempt /*preemption*/,
```

```

auto_start, *reserved1 )
```

```

gTask_appl : 초기화 및 DVD player
gTask_osd : OSD 초기화 및 명령어 디스플레이
gTask_key : 입력 받은 키를 처리하는 태스크
gTask_play : 오디오와 비디오를 플레이
gTask_audio : 오디오를 관리하는 태스크
gTask_AVL : 오디오와 비디오를 동기화
```

```

SysCreateTask(&gTask_appl, "APPLMGR", ApplManagerMain,
0, 0, 1000000, 12, 1, TASK_PREEMPT, TASK_START, 0);
SysCreateTask(&gTask_osd, "OSDMGR", OsdManagerMain, 0,
0, 100000, 13, 1, TASK_PREEMPT, TASK_START, 0);
SysCreateTask(&gTask_key, "KEYRX", ReceiveKey, 0, 0,
50000, 13, 1, TASK_PREEMPT, TASK_START, 0);
SysCreateTask( &Task_play, "PLAY TASK", PlayTask,
0, 0, 4096, 13, 1, TASK_PREEMPT, TASK_START, 0);
SysCreateTask(&gTask_audio, "AUDIO TASK", Audio_Task,
0, NULL, 4096, 13, 1, TASK_PREEMPT, TASK_START, 0);
SysCreateTask(&Task_AVL, "TASK AVL", AVL_task,
0, NULL, 50000, 13, 1, TASK_PREEMPT, TASK_START, 0);
```

각 태스크는 자신의 메시지 큐에 동기화 되어 명령을 받고 특정 작업을 수행한다. 각 태스크의 time slice 는 모두 1 로 하고, 우선순위는 gTask_appl 태스크를 12 로 가장 높은 우선순위로 부여하고 나머지의 태스크들은 13 으로 동일한 우선순위를 부여하였다. 다른 태스크가 수행 중에 gTask_appl 태스크가 실행 가능한 READY 상태가 되면 실행 중인 태스크는 수행을 중지하고 gTask_appl 태스크에 선점당하여 인터럽트를 제외하고 항상 gTask_appl 태스크가 가장 먼저 스케줄링 된다.

3.2 메시지 큐 생성

```

SysCreateMsgQ( *queue /* QCB */,
*name /* Name */, queue_size /* # of message */,
msg_type /* variable or fixed message */,
msg_size /* message size */, suspend_type,
*reserved1 )
```

```

gQ_key : 키 입력
gQ_osd : OSD
gqueue_play : play 관련 명령
gAudio_queue : 오디오
```

```

SysCreateMsgQ( &gQ_key, "KEYRX", 1, MSG_FIXED_SIZE,
sizeof( Key_t ), SYS_FIFO, 0);
SysCreateMsgQ( &gQ_osd, "OSDQ", 3, MSG_FIXED_SIZE,
sizeof( OsdMsg_t ), SYS_FIFO, 0);
SysCreateMsgQ(&queue_play, "PLAYQUEUE", 32,
MSG_FIXED_SIZE, 4, SYS_FIFO, 0);
SysCreateMsgQ(&gAudio_queue, "AUDIO QUEUE", 32,
MSG_FIXED_SIZE, 4, SYS_FIFO, 0);
```

각 태스크에 하나의 메시지 큐가 할당되며, 메시지 큐에 메시지가 도착하면 메시지를 분석하여 필요한 연산을 수행한다.

3.3 이벤트 플래그 생성

```
SysCreateEventGroup(*group /* ECB */,
*name /* Name */, *reserved1)
```

play_event : 플레이 이벤트

```
SysCreateEventGroup(&play_event, "PLAY_Event", 0);
```

네 개의 태스크에 의해 조건이 만족될 경우에 이벤트 플래그에 동기화 되어 gTask_paly 태스크가 실행된다.

3.4 타이머 생성

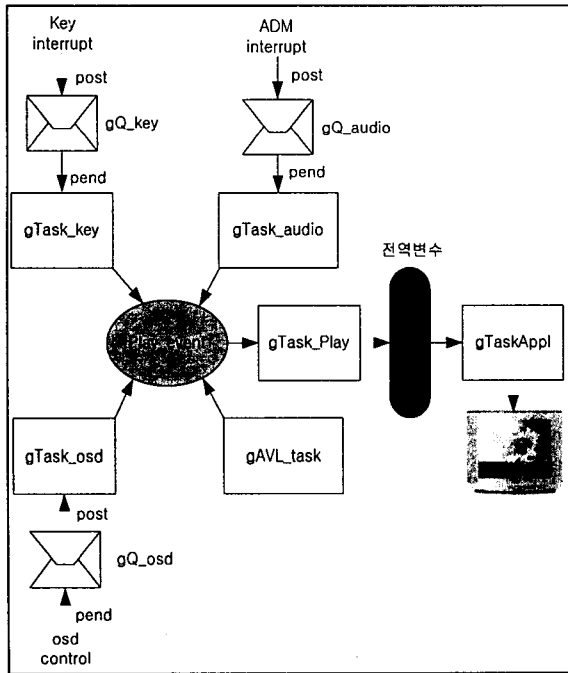
```
SysSetTimer(*timer /* Timer Control Block */,
*name /* Name */, func /* timer expire function */,
id /* function argument */, init_time /* initial time */,
reached_time /* repeat time */, enable, *reserved1)
```

gTimer_osd : OSD 타이머

```
SysSetTimer(&gTimer_osd, "OSD_TIME", OsdTimerExpired,
0, 1, 0, TIMER_DISABLE, 0);
```

각 명령어가 OSD 타이머에 정의한 시간 내에 실행하도록 한다.

3.5 태스크와 메시지 큐의 구조



[그림 3-2] 태스크와 메시지 큐의 흐름도

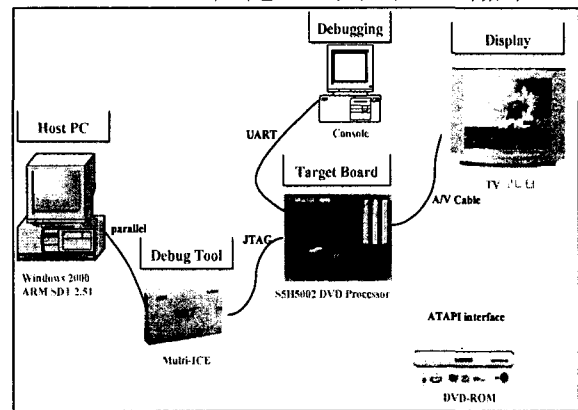
[그림 3-2]는 태스크와 메시지 큐를 사용한 전체적인 개발 흐름을 보여주고 있다. gTask_key 태스크는 key 인터럽트로부터 입력 받은 키를 저장하는 gQ_key 큐에서 메시지를 기다리고, gTask_audio 태스크는 ADM 인터럽트로부터 입력된 명령을 gQ_audio 큐에 저장된 메시지를 기다리고, 그 밖에 gTask_osd

태스크와 gAVL_task 태스크 등이 있다.

4 개의 태스크에서 Play_event 이벤트에 동기화 되어 gTask_Play 를 실행 가능하게 세마포를 사용한 전역변수를 이용 gTaskAppl 태스크가 실행하여 화면에 디스플레이 된다. 사용자 입장에서는 디스플레이 되는 화면이 가장 중요한 초점이 되므로, 화면이 깜빡이지 않도록 gTaskAppl 태스크에 가장 높은 우선순위를 부여하였다.

4. 테스트 환경 및 결과

본 논문에서 기술하고 있는 실시간 운영체제는 32-bit CPU 를 대상으로 설계되었으며, 컴파일러는 ARM SDT 2.51 을 사용하였다. 컴파일한 실시간 운영체제 코어의 실행 이미지는 약 23K 정도 이며, 이를 삼성에서 제작한 ARM940T 기반의 S5H5002 Evaluation Board 에 다운로드하여 테스트 하였다.



[그림 3-3] 테스트 환경

5. 결론 및 향후 연구과제

본 논문은 DVD player 용 실시간 운영체제를 설계하고 구현하였다. 이를 기반으로 정보가전용 실시간 운영체제의 개발의 기초가 된다.

향후 연구과제는 핸드폰, PDA 단말기를 위해 실시간 운영체제 상에 Java Virtual Machine 을 구현하고 실시간 운영체제에 저전력 기법을 도입하여 실시간 운영체제의 응용 폭을 넓히는 부분은 계속 연구되어야 할 것이다.

6. 참고 문헌

- [1] Embedded System, RTOS, <http://www.inestech.com>
- [2] (주)삼성전자, " S5H5002 DVD Processor USER' S MANUAL" , 2003
- [3] Jean J. Labrosse, uC/OS The Real-Time Kernel, R&D Publications, 1995.
- [4] WindRiver, VxWorks Programmer's Guide, 1997.
- [5] Accelerated Technology, Nucleus PLUS Internals Manual, 1996.
- [6] IEEE Std 1003.1b, Portable Operating System, 1993.