

CDMA 시스템용

실시간 운영체제의 설계 및 구현*

°류 석, 편현범, 이재호, 윤기현, 이재경*, 이철훈
충남대학교 컴퓨터공학과, *한국 전자 통신 연구원

Design and Implementation of A Real-Time Operating System for the CDMA Systems

Seok Ryu°, Hyun-Beam Pyun, Jae-Ho Lee, Ki-hyun Yoon, Jae-kyung Lee*, Cheol-Hoon Lee
Dept. of Computer Engineering, Chungnam National Univ., *ETRI

요 약 문

본 논문에서는 CDMA 단말기의 운영을 위한 Embedded 운영체제를 설계하고 구현 하였다. 이 운영체제는 일반적인 운영체제와는 달리 CDMA 단말기에 적합하게 구현 되었다. 실시간 운영체제는 우선순위를 기본으로 하는 선점형(Preemptive) 운영 체제인 것이 일반적인 추세이고 본 논문에서 언급한 실시간 운영체제 역시 우선순위를 가지는 선점형 운영체제이다. 본 논문에서는 CDMA 단말기용 실시간 운영체제의 개발 환경과 그리고 개발된 운영체제의 구조와 개발 내용에 대해 언급하였다.

1. 서론

일반적으로 Embedded 시스템은 시스템을 구동하기 위한 소프트웨어를 탑재하게 된다. 최근 들어 이러한 Embedded 시스템상의 소프트웨어들이 보다 많고 복잡한 일들을 수행하는 추세에 있으며, 이를 운영하기 위한 운영체제의 필요성이 더욱 커지게 되었다. 그런데 Embedded 시스템용으로 개발되어진 운영체제 들은 대부분 실시간 처리를 목적으로 하는 것이므로 실시간 운영체제를 Embedded 시스템의 운영체제로 탑재하는 것이 최근 경향이다.

본 논문에서는 CDMA 단말기를 위한 실시간 운영체제의 설계 및 구현내용을 담고 있다. 본 논문은 2장에서 실시간 운영체제가 가지는 특성을 기술하고, 3장에서는 실시간 운영체제의 구현 내용을 기술 하였다. 그리고 4장에서는 결론 및 향후 연구 과제를 기술 하였다.

2. 실시간 운영체제의 특징

본 논문에서 개발을 목적으로 한 실시간 시스템은 논리적인 정확성과 시간적인 정확성을 가져야 하는 시스템이다. 따라서 이러한 운영체제는 다음과 같은 특성을 가진다[6].

- Multi-Tasking
- Priority Base
- Preemptive

본 논문의 운영체제도 이를 개발함에 중점을 두었다.

3. 운영체제 설계 및 구현

본 연구는 ARM evaluation Board 인 PID7T를 사용하여 실험을 하고 있다. PID7T 보드는 ARM CPU를 탑재하고 있으며 따라서 운영체제 중 Hardware와 관련 있는 부분은 Assembly로 작성이 되어 있다[1].

본 절에서는 스케줄링 정책, 상태 전이도, Task의 구조, ITC(InterTask Communication)구조, Memory Management, I/O Interface을 중심으로 한 구현 내용에 대해 기술하였다.

3.1 스케줄링 정책

개발된 실시간 운영체제는 우선순위를 가지는 선점형 운영체제다. 즉, 이 운영체제는 항상 가장 우선순위가 높은 Task가 CPU를 사용하도록 CPU 선점하는 구조를 가지고 있고, 같은 우선순위의 Task들은 Round Robin 방식으로 스케줄링이 된다.

실시간 운영체제에서 가장 중요한 요소는 Determinism이 제공되어야 한다. 그런데 실제 스케줄링에서 사용하는 Ready List는 Task의 수에 따라 우선 순위를 결정하는 시간이 달라지는 특징이 있는데 이러한 시간을 줄이기 위해 우선순위 Table을 가지고 관리를 하여 고정된 시간에 가장 우선순위가 높은 Task를 찾을 수 있도록 한다[2].

따라서 Table을 이용하여 가장 높은 우선순위의 Task를 선택하도록 하고 같은 우선순위의 Task들은 Round Robin 방식에 의해 스케줄링 되는 운영체제로 개발하였다[3].

* 본 연구는 99년부터 한국전자통신연구소의 위탁 과제로 수행되고 있다.

3.2 Task 의 구조와 상태전이도

```

Typedef struct os_tcb {
    void          *OSTCBStkPtr;
    UBYTE        OSTCBStat;
    UBYTE        OSTCBPrio;
    UWORD        OSTCBDly;
    UBYTE        OSTCBX;
    UBYTE        OSTCBY;
    UBYTE        OSTCBBitX;
    UBYTE        OSTCBBitY;
    Struct os_tcb *OSTCBNext;
    Struct os_tcb *OSTCBPrev;
    OS_EVENT     *OSTCBEvtPtr;
    UBYTE        Ouspend;
} OS TCB;
    
```

그림 1. Task 구성

Task 는 다음과 같은 구조를 가지고 있다[4].

- OSTCBStkPtr 은 Task 의 Stack Pointer 를 가리킨다.
- OSTCBStat 는 Task 의 상태를 나타낸다.
- OSTCBPrio 는 Task 의 Priority 를 나타낸다.
- OSTCBDly 는 Task 가 Delay 이 되었을때 몇 Clock Tick 동안 Delay 되어야 하는지에 대한 값을 가지고 있다.
- OSTCBX,Y,BitX,Y 는 이차원 배열로 되어있는 Task list 에서 Task 가 어디에 위치하고 있는지에 대한 위치 정보를 가지고 있다.
- OSTCBNext,OSTCBPre 는 현재 Task 가 Ready List 에 포함되어 있을 경우 다음 Ready Task 의 위치를 가리키는 Pointer 이다.
- OSTCBEvtPtr 는 Event Control Block 의 Pointer 에 대한 정보를 가지고 있다.
- OSsuspend 는 요청한 Memory 가 부족 시 Suspend 된 process id 의 Linked List 위한 자료구조이다.

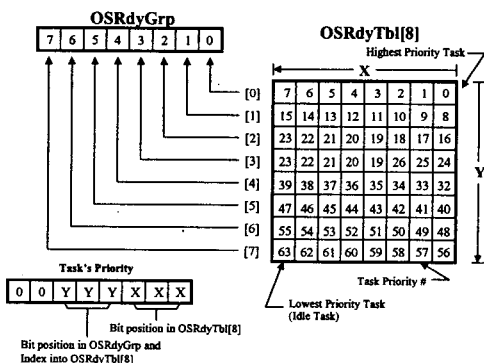


그림 2 Priority Table

그림 2 와 같이 Task 의 Priority 를 가지고 OSRdyTbl 의 위치를 찾아 내는데 OSRdyTbl 의 선택되어진 Bit 를 가지고 TCB Header 를 찾도록 되어 있다. TCB Header 는 그림 3 과 같이 되어 있는데 각 Header 들은 각 우선 순위 에 해당된다. 이런 Header 는 동일한 우선순위를 가지는 Task 들을 Doubled Linked List 형태로 관리를 하게 된다.

이러한 Task 들은 그 상태에 따라 그림 4 와 같은 Status 를 가지고 있다.

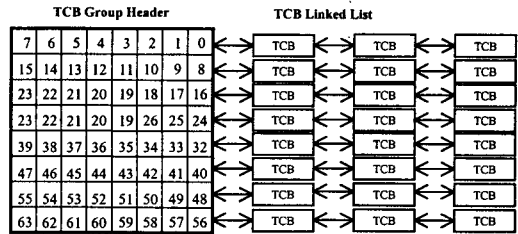


그림 3. TCB Group Header 와 TCB List

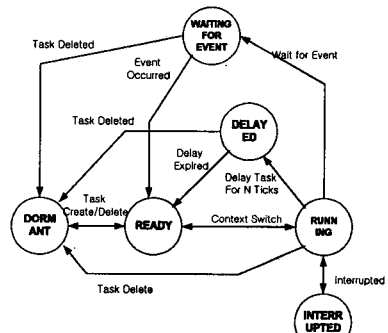


그림 4. 실시간 운영체제의 상태 전이도

3.3 ITC (InterTask Communication)

개발된 운영체제는 ITC 를 위하여 다음과 같은 설비를 가지고 있다.

- 세마포(Semaphores)
세마포는 동시적으로 진행되는 Task 들이 Shared Resource 를 안전하게 사용하는 기술이다. 어떤 리소스를 요청하는 Task 는 먼저 리소스에 해당하는 Semaphore 를 요청하게 되고 리소스를 획득하게 되면 Semaphore Count 를 1 감소 시킨다. 만약 Semaphore Count 가 0 이되면 더 이상 리소스를 획득 할 수 없기 때문에 요청한 Task 는 리소스가 Available 해질 때 까지 나 또는 Time-out 될 때 까지 Wait for Event 상태로 전이하게 된다.
- Mail Boxes
Mail Box 는 하나의 Task 에서 다른 Task 로 메시지 1 개를 전송하는 방식이다.

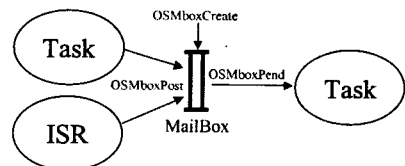


그림 5. Mail Box

송신을 하는 Task 나 ISR 은 메시지를 하나 Event Control Block 에 저장하게 되고 Event Control Block 은 그 메시지를 수신측에서 받아 갈 때 까지 저장하게 된다. 수신측 Task 는 역시 Event Control Block 에 대한 정보를 가지고 있어야 하고 이를 통해서 메시지를 하나 전송 받게 된

다. 만약 수신측이 메시지를 전송하기 전에 메시지를 기다리게 되면 Wait for event 상태로 Task Status가 변하게 된다. 모든 메시지는 Task로 전송이 가능하고 ISR은 메시지를 전송할 수는 있지만 받을 수는 없다. 만약 ISR이 메시지를 전송 받게 되면 ISR이 wait for event 상태가 되어 ISR수행시간이 Unbound 된다. 이것은 System에 심각한 영향을 미칠 수 있다[2].

● Message Queues

Message Queues는 하나의 Task나 ISR에서 다른 Task로 여러 개의 Message를 전송하는 방식으로 Mail Box가 모여있는 모습을 가지고 있다.

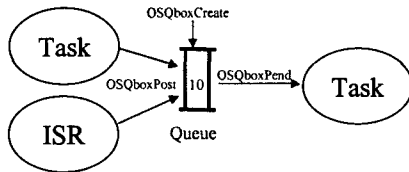


그림 6. Message Queue

3.4 Memory Management

메모리 구조는 메모리 Pool을 관리하는 Pool Control Block과 메모리 Pool을 관리하는 Memory Control Block으로 나누어 진다[5].

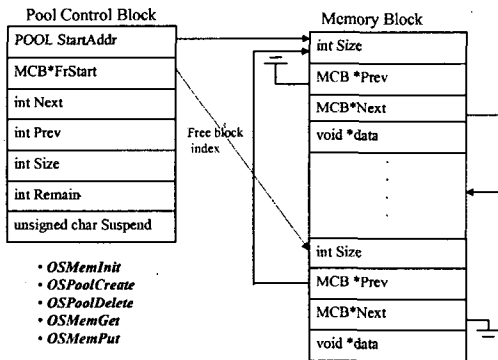


그림 7. Pool Control Block과 Memory Block

Pool Control Block은 다음과 같이 구성되어 있다.

- StartAddr는 Memory의 가장 처음 부분을 가르친다.
- FrStart는 현재 Free Memory Block의 처음을 가르친다. 여기서부터 Memory를 할당하게 된다.
- Prev, Next는 Pool control Block도 Table형식의 Doubled Linked List로 관리 되는데 이를 만들기 위한 자료구조이다.
- Remain은 이 Memory Block의 할당 가능한 Memory의 크기를 가르친다.
- Suspend는 초기값이 -1로써 메모리를 할당하지 못해서 Suspend되어진 Task중 가장 우선순위가 높은 Task를 가르치게 된다.

Memory control Block은 다음과 같이 구성되어 있다.

- Size는 이 Memory Block의 크기를 가르친다.
- Prev,Next는 Memory Block이 Used, Free Memory Block의 List를 Doubled Linked List로 만들기 위해서 필요한 구조이다.

- Data 실제 data가 있는 곳으로 실제 Task들은 메모리를 할당 받으면 이곳의 Pointer를 받게 된다.

3.5 I/O Device Interface

I/O Device Interface는 Task들과 Device들을 연결할 수 있도록 운영체제에서 제공하는 Interface이다. CDMA 단말기는 그 특성상 Character Device만이 존재하므로 일반적인 POSIX 표준이 Block Device는 고려하지 않고 설계하였다.[3]

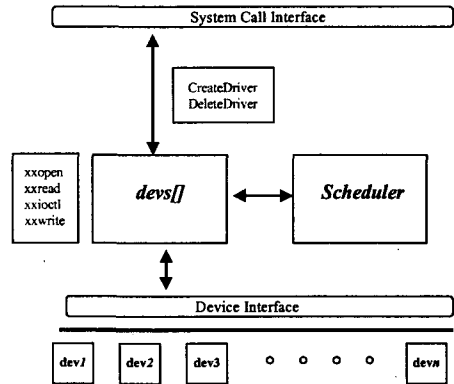


그림 8. Device Driver Interface

이러한 Device들은 그 생성할 때나 사용할 때 Suspend되는 경우가 있는데 운영체제는 이를 위해서 Task를 Suspend하고 Wakeup하는 기술을 사용한다.

4. 결론 및 향후 연구 과제

본 논문에서는 CDMA 단말기를 위한 실시간 운영체제를 설계하고 구현 하였다. 연구된 실시간 운영체제는 우선순위를 기반으로 하는 선점형 방식을 채택한 것이 그 특징이다.

본 연구에 이어 계속 연구되어야 할 부분은 CDMA 단말기에 가장 필요한 부분이 Timer기능과 Interrupt Sharing을 할 수 있는 ISR 기술에 대한 연구가 계속 이루어져야 하고 실제 CDMA 단말기에 적합한 성능을 가지고 있는 가에 대한 검증이 이루어져야 한다. 이러한 성능 평가는 외부 장치 또는 Microprocessor Manual을 이용하여 Context Switch 시간, 그리고 Interrupt Latency 및 운영체제의 설비에 대한 Delay의 조사가 이루어져야 하며, 또한 각각의 Task가 항상 정해진 시간에 수행을 하는지의 시간적 요소 및 결과에 대한 검증을 해야 할 것이다.

5. 참고문헌

- [1] Advanced RISC Machine Ltd(ARM), "ARM Software development Toolkit User Guide", 1996.
- [2] Jean, J. Labrosse, "uC/OS The Real-Time Kernel", R&D Publications, 1992.
- [3] D. Comer "Operating System Design VOL 1 : THE XINU APPROACH", 1988.
- [4] 류석,강용호,이철훈,"GPS 수신 Engine Board용 실시간 운영체제의 설계 및 구현", 한국정보과학회 학술 발표논문집, Vol. 26, No 1, pp.116~118, 1999.
- [5] Accelerated Technology, Inc "Nucleus PLUS Internal Manual", 1996.
- [6] Vrtx Homepage, "http://www.mentor.com/embedded/vrtxs"