

실시간 멀티태스크 처리 기법을 이용한 선박엔진 안전시스템 개발에 관한 연구

황동희* · 류길수**

A Study on the Development of Marine Engine Safety System Using Multi-tasking Method in Real Time

D. H. Hwang, K. S. Rhyu

Abstract

A purpose of this paper is the development of a marine engine safety system in real time.

In a general real time system using sequential programming method, it is very difficult to realize object operating systems in real time when their scales are large. Therefore multi-tasking environment which contains basic functions of multi-tasking should be constructed in order to easily realize the real time system. In this paper, the function of marine engine safety system is constructed with several tasks which are classified into specific domain characteristics. And multi-tasking is executed by means of time-slice and event-driven multi-tasking method.

The marine engine safety system developed in this paper has two merits.

- Since the switching time between tasks is short, the system can execute all the tasks with small time relatively.
- Since the tasks are classified into specific domain characteristics, modification can be easily executed.

* 한국해양대학교 제어계측공학과 석사과정

** 한국해양대학교 컴퓨터공학과 부교수

1. 서론

종래에 사용되어 왔던 마이크로프로세서를 이용한 실시간 제어 및 감시 시스템들은 대상 시스템의 특성에 맞는 기능들을 순차적으로 실행해 나가는 절차형 방식이 대부분이었다. 그러나 선박제어와 같이 대상시스템이 대형화되고 작업영역이 방대해지면 절차형의 실행방식만으로는 실시간 제어 시스템의 작성이 대단히 어렵게 된다. 이와 같은 시스템에서는 대상영역의 특성에 따라 여러개의 태스크로 분할되어 있는 작업내용들을 필요시에만 실행해 주는 기능이 미리 갖추어져 있는 것이 바람직하며 이러한 환경을 실시간 멀티태스크 환경이라고 한다.^[1] 따라서 실시간 멀티태스크 환경에서는 대상영역의 응용태스크를 특성에 따라 분류하여 작성해 놓는 것만으로 시스템의 구축이 완료되므로 기능의 추가 및 삭제가 용이하여 확장성이 뛰어나게 된다.

UNIX와 같은 운영체제에서는 타임슬라이스(Time-slice) 기법을 이용한 멀티태스크 환경이 구축되어 사용되고 있으며, PC에서도 MS-WINDOWS와 같은 멀티태스크 기능을 가진 운영체제의 개발이 활발하게 이루어지고 있다.^{[2]-[7]} 경제적인 측면을 고려하여 주로 저급 마이크로프로세서로 구성되어 있는 실시간 시스템^{[8]-[10]}의 경우는 이벤트드리븐(Event-driven) 방식을 이용한 실시간 멀티태스크 시스템이 제공되고 있어서 이러한 멀티태스크 시스템을 이용하여 실시간 시스템을 개발하기에는 개발비가 너무 많이 소요되며, 또한 많은 영역을 대상으로 하기 위해 일반화된 기능을 가지고 있어서 실행속도가 느리다는 점과 불필요한 메모리를 많이 차지한다는 단점을 가지고 있다. 따라서 메모리의 절약과 실행속도의 향상이라는 관점에서, 대상영역에 따라서는 이벤트드리븐 방식과 타임슬라이스 방식을 혼용한 멀티태스크환경의 구축이 요구되어 오고 있다.^[11]

이상과 같은 배경으로부터 본 논문에서는 멀티태스크 기능을 가진 선박엔진 안전시스템의 구현에 관하여 논하고자 한다. 선박엔진 안전시스템은 엔진을 감시, 보호하기 위해 기준값과 측정값을 비교하여 오버 스피드, 셋 다운 및 슬로우 다운 등을 수행하는 시스템^{[12], [13]}을 말하며, 인텔 8088과 같은 저급 마이크로프로세서를 이용하고 있어서 그 기능의 보다 효율적인 수행을 위해서는 멀티태스크 기능을 가진 효율적인 시스템의 개발이 요구된다. 즉 선박엔진 안전시스템에서 키값의 입력이나 센서값의 입력과 같은 기본적인 기능은 타임슬라이스 방식을 이용하여 구동시키고 파라미터의 변환, 원격제어시스템과의 통신 및 특정상황에 따른 동작등은 이벤트드리븐 방식을 이용하여 처리되게 함으로써 프로그램 작성을 용이하게 함과 동시에 태스크의 절환시간을 줄여 실시간 처리를 가능하게 하였다.

2. 실시간 시스템에서 요구되는 멀티태스킹

멀티태스킹을 구현하는데 있어서는 타임슬라이스에 의한 방식과 이벤트드리븐에 의한 방식이 있는데 타임슬라이스에 의한 멀티태스킹은 태스크가 CPU를 점유하는 시간을 일정한 시간 간격의 단위(Time-slice)로 정하고 그 시간이 경과할 때 마다 실행 태스크를 바꾸어 가는 방식으로서 각 태스크에 대한 공평성을 최우선으로 하고 있다. 즉 타이머카운트 기능에 의해 시간을 측정하여 일정시간이 경과하면 다른 태스크로의 전환이 이루어지고 이전에 작업하던 태스크는 대기상태에 들어가게 된다. 이 방식에서는 외부에서 특별한 요구가 없더라도 우선순위와 스케줄링의 기법에 따라 무조건 태스크의 전환이 이루어 지게 되므로 때로는 불필요한 태스크의 전환으로 인한 전체 시스템의 속도저하로 실시간 처리가 어렵게 될 수 있다.

이벤트드리븐 방식의 멀티태스크는 모니터에서 항상 이벤트의 발생을 감시하고 있고 이벤트가 발생하게 되면 해당 태스크를 구동시키게 된다. 이벤트가 발생하면 해당 태스크로 메시지가 전송되고 메시지 수신대기 중이던 해당 태스크는 대기가 해제되어 구동이 시작된다. 구동된 태스크는 작업을 실행할 때 공유자원을 배타적으로 점유하여 사용함으로써 작업이 종료되기 전에는 다른 태스크로의 전환이 이루어 지지 않게 된다. 이 방식은 타임슬라이스 방식과는 달리 불필요한 태스크 전환이 없기 때문에 태스크의 전환시간은 줄일 수 있지만 외부 이벤트에 의해서만 태스크의 전환이 이루어 지므로 우선순위의 선정이나 스케줄링의 구현이 대단히 복잡하게 된다.

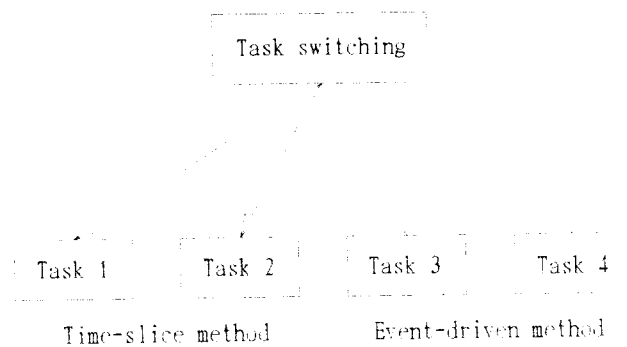


Fig. 2.1 Real time multi-tasking system using time-slice and event-driven method

따라서 Fig. 2.1에서와 같이 태스크 1이나 태스크 2와 같이 주기적인 실행이 필요

한 태스크는 타임슬라이스 방식에 의해 구동되도록 하고 태스크 3이나 태스크 4와 같이 특정한 이벤트가 발생할 때만 한번씩 실행해 주면 되는 태스크는 이벤트드리븐 방식에 의해 구동되도록 두 방식을 혼용함으로써 태스크의 불필요한 절환시간을 줄여 시스템의 실행속도를 향상시킬 수 있다.

이상에서 설명한 실시간 멀티태스크 환경을 구축하기 위한 기본요소로는 각 태스크 상태의 관리와 태스크에서 사용하는 자원에 대한 관리 및 태스크간의 메시지 송수신 기능이 있으며 이러한 기능들을 수행하기 위한 기본요소로는 태스크 테이블(Task Table), TCB(Task Control Block), RCB(Resource Control Block), 메일박스(Mail Box) 등의 영역이 필요하며, 특히 시스템의 속도향상을 위해 이러한 요소들이 차지하는 메모리의 크기를 최소화하는 것이 매우 중요하다.

3. 선박엔진 안전시스템의 구성 및 기능

선박엔진 안전시스템은 선박의 엔진을 감시, 보호하기 위한 안전장치로 전면에는 푸시버튼과 램프 및 디스플레이어가 있고 시스템의 내부에는 마더보드를 비롯한 전기전자적 카드 및 어댑터 카드가 내장되어 있다. 이 안전시스템은 Fig. 3.1과 같은 구조로 전체 선박엔진 시스템을 구성하는 요소 중의 하나이며 그림에서와 같이 원격제어 시스템과 주엔진으로부터 정보를 입력받아 상태에 따라 경보의 출력, 셋 다운 및 슬로우 다운 신호의 출력 등을 수행하게 된다.

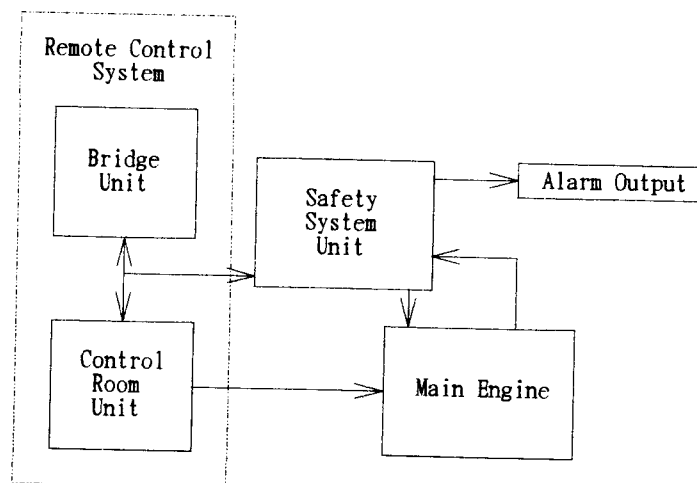


Fig. 3.1 Configuration of the marine engine control system

선박엔진 안전시스템의 하드웨어는 인텔 8088 마이크로프로세서를 CPU로 하고, RAM, ROM과 같은 메모리 이 외에도 전원 소멸시 파라미터 데이터를 기억해 두기 위해 EEPROM을 사용하고 있다. 그 외에 데이터의 병렬 입출력을 위한 8255, 인터럽트를 관리하는 8259, 시스템 타이머인 8253과 9513, 시리얼 통신을 위한 8251, 세븐 세그먼트 구동용 7218 등 소프트웨어로 구동이 가능한 주변소자들이 있다.

4. 선박엔진 안전시스템 구동 소프트웨어의 구축

선박엔진 안전시스템의 기능을 만족하며 실시간 멀티태스킹이 가능한 시스템의 구현을 위해서는 다음 4가지의 서브시스템이 요구된다.

- 센서측정 서브시스템
- 파라미터조절 서브시스템
- 통신 서브시스템
- 워치도그 타이머 서브시스템

이 4가지 서브시스템의 기능과 전체 시스템의 구성은 다음과 같다.

(1) 파라미터조절 서브시스템

파라미터조절 서브시스템에서는 전원 소멸시에도 파라미터 값을 안전하게 보존하기 위해 '0x80000 - 0x87fff' 어드레스의 EEPROM 영역을 이용하고 있는데 이 영역을 C 언어에서 이용하기 위해서는 절대어드레스를 직접 참조하는 방식이 필요하다. 단 직접참조시에는 원하는 데이터가 물리적으로 어느 어드레스에 저장되어 있는지를 사전에 약속해 둘 필요가 있으며, 특히 자주 사용 되는 값들은 이해하기 쉬운 변수명으로 정의하여 두고 참조할 수 있도록 값이 수정되었을 때마다 해당 변수영역으로 복사하여 둘 필요가 있다.

다음은 물리어드레스 '0x80000'을 직접 참조하는 기법을 나타내고 있는데 아래와 같이 far pointer로 정의하여 참조한다

```
char far *a;

a = (char far *) 0x80000000L;
                          
                segment offset
```

(2) 센서측정 서브시스템

이 시스템은 어댑터 카드를 통해 들어오는 센서값과 RPM 데이터들을 일괄적으로 측정하여 메모리 상에 저장해 두고 EEPROM에 저장되어 있는 파라미터 값과 비교하여 현재 시스템의 상황을 램프를 통해 표시해 주는 시스템으로 그 값들은 8255를 통해 입출력 된다.

이 때 어댑터카드를 통해 들어오는 값들은 셋 다운, 슬로우 다운, 비상정지에 대한 데이터들이고 RPM 값은 A/D 컨버터를 통해 '0'에서 '4096'사이의 데이터로 변환되어 입력된다.

(3) 통신 서브시스템

이 시스템은 다른 유니트와의 통신을 수행하는 시스템으로 8253과 8251의 소프트웨어 구동기법을 이용하여 구현하였다. 현재는 다른 유니트의 구동 소프트웨어가 구현되어 있지 않아서 컴퓨터와 안전시스템과의 통신만을 실행확인 한 상태이므로 본 논문의 프로그램에는 추가하지 않았다.

실제 운전시에 통신해야 할 주요내용은 주엔진시동, 비상후진, 속도제어시 다른 유니트에서 전달되는 비정상적인 상태를 나타내는 값들로서 데이터량이 그다지 많지 않기 때문에 비동기 통신을 이용하여 실험하였다.

(4) 위치도그 타이머 서브시스템

이 서브시스템은 전체 시스템의 정상동작을 확인하기 위한 것으로 8253과 8259의 소프트웨어 조작법을 이용하여 구현하였고 그 하드웨어의 구성은 Fig. 4.1과 같다.

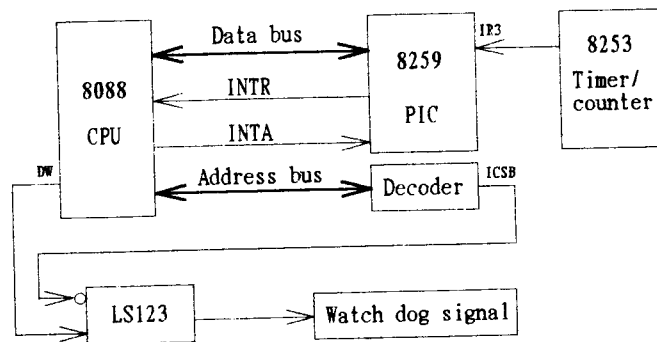


Fig. 4.1 Hardware block diagram of watch dog timer sub-system

안전시스템은 CPU 및 주변소자들의 동작이상유무를 판단하기 위해 'LS123'을 이용한 위치도그 타이머기능을 가지고 있는데 Fig. 4.1에서 'LS123'의 9번 입력단자는

CPU에서 나오는 'DATA WRITE' 신호에 접속되어 있고 10번 입력단자는 디코더에서 나오는 'ICSB' (어드레스 영역: 0x58 - 0x5F)에 접속되어 있으며 출력신호는 위치도그 신호로 사용되고 있다.

LS123에서 9번 입력은 반전되어 들어가므로 9번입력은 'L', 10번입력은 'H'가 되어야 출력이 'L'로 되어 안정한 상태가 된다. 따라서 8253에서 발생하는 주기적인 클럭신호를 8259의 인터럽트 컨트롤러에 접속하여 매 주기마다 인터럽트를 발생시키고 이 인터럽트 신호를 이용해 '0x58'어드레스를 호출하여 'ICSB' 신호를 'ENABLE' 상태로 만들어 주므로써 위치도그 타이머 기능을 수행할 수 있다.

(5) 시스템의 구성

전체 시스템은 이상에서 설명한 기능을 바탕으로 Fig. 4.2와 같이 크게 초기화부와 태스크 구동부, 그리고 인터럽트 처리부로 나누어 구성하였다.

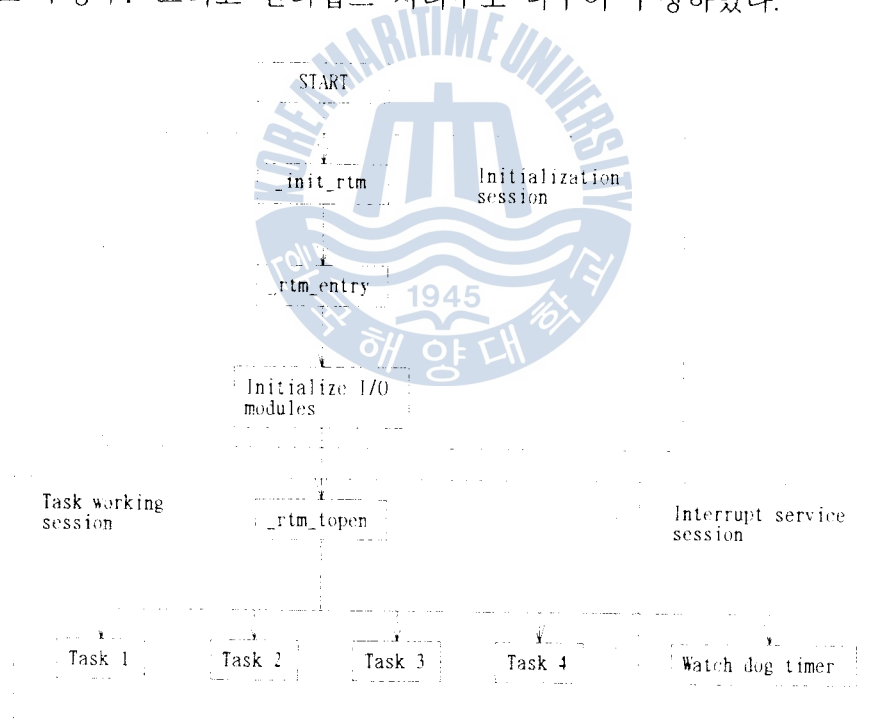


Fig. 4.1 Overall flowchart of a program to operating the safety system

초기화부에서는 시스템에서 필요한 작업영역 및 주변소자들을 동작특성에 맞게 초기화 시키는 작업을 실행한다.

특히 태스크 구동부에서는 센서측정기능과 파라미터 조절기능, 디스플레이기능

등을 4 가지의 태스크로 구분하여 작성한 후 각각의 태스크를 타임슬라이스 방식과 이벤트드리븐 방식에 따라 멀티태스킹 되도록 하였다. 이 때 태스크 1과 2는 각각 키값과 센서값을 주기적으로 입력받는 기능을 가지고 있으므로 타임슬라이스 방식을 이용하여 구동되도록 하였고 태스크 3과 4는 태스크 1이나 2의 요구가 있을 때만 구동되어 램프 디스플레이, 알람 및 릴레이 구동, 파라미터 저장 등의 기능을 수행하므로 이벤트드리븐 방식을 이용하여 구동되도록 하였다.

인터럽트 처리부에서는 별도의 인터럽트 서어비스 루틴을 작성한 후 매 인터럽트시 마다 구동되도록 하여 시스템의 정상동작 여부를 감시하는 위치도그 타이머 기능을 수행하도록 하였다.

본 논문에서는 실시간내에서 처리가 가능한 멀티태스크 기법을 이용한 선박엔진 안전시스템의 구현에 대해 논하였다.

5. 결 론

전체 시스템의 구현을 위해 미리 멀티태스크 환경을 구축해 놓고, 선박엔진 안전시스템의 주요기능들을 특징에 따라 4개의 태스크로 나누어 작성하였다. 이러한 시스템 구현법에는 다음과 같은 장점이 있다.

- 기존의 절차형 프로그램 작성법과 달리 대상영역의 기능에 따라 태스크만을 작성하면 되므로 대규모 시스템에 적합하다.
- 기능의 추가 및 삭제가 용이하여 새로운 환경변화에 대한 확장성이 뛰어나다.

또한 멀티태스크 환경 구축에 있어서는 태스크의 기능에 따라 타임슬라이스 방식과 이벤트드리븐 방식을 혼용하였으며 이로 인해 불필요한 태스크의 불필요한 절환시간을 줄임으로써 전체 시스템의 속도를 향상시킬 수 있었다.

금후의 연구과제로는 다음과 같은 점을 들 수 있다.

- 유동적인 구동시간의 할당이 요구된다.
- 스케줄링시 동적인 우선순위의 변경이 필요하다.
- 파라미터 데이터 구조의 범용화가 요구된다.

참 고 문 헌

- [1] 박재현, 이홍규. “엄격한 실시간 운영체제 커널의 현황 연구.” 정보과학회지, 1994. 11.
- [2] 김석일, 이충한, 지승현, 김대식. “병렬처리 컴퓨터의 특징과 처리기법.” 정보과학회지, Vol. 12, No. 5, 1994. 6.
- [3] 윤현주, 윤현수. “실시간 시스템의 하드웨어 아키텍처.” 정보과학회지, Vol. 11, No. 1, 1993. 2.
- [4] 김명주, 전주식. “병렬처리를 위한 함수형 수행방식을 갖춘 혼합형 프로그래밍 모델과 언어.” 정보과학회논문지, Vol. 20, No. 8, 1993. 8.
- [5] 구용완, 김영찬. “실시간 운영체제의 커널 설계 및 구현.” 정보과학회논문지, Vol. 17, No. 1, 1990. 1.
- [6] 金榮燦. 運營體制, 尚潮社, 1992.
- [7] David M. Auslander, Cheng H. Tham, *Real-time Software for Control*, Prentice hall, 1990.
- [8] “8086ボードコンピュータの設計&製作 series.” project 1. *Bootstrap No. 1-6*, CQ出版社, 1992.
- [9] John Uffenbeck, *THE 8086/8088 FAMILY*, Prentice-hall International inc., 1987.
- [10] 조순복, 김광희, *8086/8088 마이크로프로세서와 그 주변소자들*, 집문당, 1991.
- [11] 大原 茂之, 澤田 勉, Peter Petrov, *實踐リアルタイム プログラミング技法* オーム社, 1991.
- [12] NORCONTROL Automation AS, *Instruction Manual for Safety system unit*, Grafisk Design, 1993.
- [13] 유영호, 하주식, *自動制御應用*, 太和出版社, 1993.
- [14] 조순복, *8086 제어용 마이컴과 C 언어에 의한 제어프로그램의 ROM화 기법*, 기한재, 1993.

