

論文96-33B-12-2

ISDN 사설교환기용 실시간 운영체제의 구현 및 성능평가

(The Implementation and Performance Evaluation of Real-Time Operating System for an ISDN PABX)

崔在洵*, 朴仁甲**

(Jae Weon Choe and In Kap Park)

요 약

본 논문에서는 높은 실시간 응답성을 가진 ISDN 사설교환기용 실시간 운영체제의 구현방법과 기술에 대해 연구하였다. 구현한 운영체제는 교환기 전용 실시간 운영체제로서 시스템의 부팅과 초기화, 다중 태스크의 실시간 병렬처리, 태스크간 통신과 동기화, 프로세서간 통신, 인터럽트 처리, 데이터 입출력과 실시간 응용을 위한 다양한 실시간 처리기능을 제공한다. 그리고 운영체제의 기능별 수행시간을 최단, 일상 및 최악의 경우로 나누어 측정하여 구현한 실시간 운영체제의 성능을 평가하였다.

Abstract

In this paper we researched the implementation methodology and techniques for the high real-time responsive operating system of an ISDN PABX system. The operating system we developed takes a high role as a dedicated real-time operating system for PABX Systems. It provides the functions such as system booting and initialization, real-time multitasking, intertask communication and synchronization, interprocessor communication, interrupt processing, and data I/O processing. And also it provides lots of real-time processing functions for supporting various real-time applications. Finally, we evaluated the performance by measuring the best, the average, and the worst-case execution time of all functions implemented in this real-time operating system.

1. 서론

실시간 시스템이란 외부의 요구에 대한 처리 및 응답이 제한된 시간내에 이루어져야 하는 특성을 갖는 시스템을 말한다^[1]. 다시 말해 실시간 시스템의 성공 여부는 결과의 타당성은 물론이고 시간 제약의 충족 여부에 의해 가름지워 진다. 이때 시간 제약의 허용치와 절대성은 시스템의 응용분야에 따라 달라질 수 있

다. 예를 들어 레이다 검색 시스템의 경우 응답시간이 1 msec 이내이어야 하는 반면 은행의 입출금 시스템인 경우에는 3 초 내외이면 충분하다. 그리고 항공제어 시스템은 응답시간이 엄격히 준수되어야 하는 하드 실시간 시스템인 반면 통신시스템은 어느 정도의 오차가 허용되는 소프트 실시간 시스템이다.

ISDN 교환시스템은 음성은 물론 데이터, 화상 등의 정보 교환을 매개하는 시스템으로 그 특성상 100 msec 내외에 응답되어야 하는 소프트 실시간 시스템이다^[2-3]. 교환시스템이 실시간 서비스를 제공하기 위해선 규정된 실시간 조건을 만족하면서 운영체제의 오버헤드는 최소화된 실시간 운영체제의 지원이 요구된다. 그래서 본 논문에서는 ISDN 교환기 응용프로그램의 다양한 요구에 신속한 응답을 하는 실시간 운영체

* 正會員, 三星電子 情報通信研究所
(Telecommunication R&D Center, Samsung Electronics)

** 正會員, 建國大學校 電子工學科
(Dept. of Elec. Eng., Kon Kuk Univ.)

接受日:1995年12月6日, 수정완료일:1996年12月10日

제의 구현에 대해 연구하였다.

본 논문은 ISDN 사설교환기용 실시간 운영체제의 구현과 성능평가에 관한 연구로서 우선 실시간 운영체제의 기능 요구사항을 분석하고, 이를 기초로 설계하고 구현한 후, 성능을 평가하였다. 구현한 운영체제는 ISDN 서비스의 실시간 지원을 위해 시스템의 부팅 및 초기화, 다중 TASK의 실시간 병렬처리, TASK간 통신 및 동기화, 프로세서간 통신, 각종 인터럽트의 처리, 데이터 입출력과 실시간 응용을 위한 다양한 타이머 기능들을 가지며 시스템의 확장성과 사용의 용이성을 고려하여 설계 및 구현하였다.

II. 실시간 운영체제의 기능

실시간 운영체제는 운영체제 본래의 고유기능을 수행하면서 외부의 요청이나 비동기적인 사건에 대해 정해지거나 미리 예견된 시간내에 응답할 수 있도록 구현된 운영체제를 말한다. 그러므로 ISDN 교환기용 실시간 운영체제는 교환기가 ISDN 서비스를 제공시 100 msec 내외에 응답 가능하도록 다양한 실시간 처리기능을 지원해야 한다. 실시간 운영체제의 구현에 앞서 우선 요구사항을 분석하였고, 실시간 처리를 위한 기능요건은 다음과 같다¹¹⁾.

1) 다중 TASK의 실시간 병렬처리 기능

ISDN 교환시스템내에는 호제어를 관장하는 호처리 TASK를 비롯하여 여러 TASK가 존재하므로 실시간 처리를 위해 우선순위가 높은 TASK가 우선순위가 낮은 TASK를 선점할 수 있는 기능이 요구된다. 또한 특정 TASK의 제어 독점을 방지하기 위해 실행중인 TASK에 대기조건이 발생시나 10 msec 주기로 재 스케줄링하고 우선순위가 동일하면 Round-robin 방식으로 스케줄링 되도록 하며 TASK 스위치(context switch)에 요구되는 시간은 최소화되도록 한다.

2) TASK간 통신 및 동기화 기능

TASK간에 정보교환을 위한 메시지 통신기능과 TASK간의 상호협력이나 상호배제를 위한 동기화 기능이 요구된다.

3) 프로세서간 통신기능

ISDN 교환기는 제어기능이 다수의 프로세서에 분산되어 있는 분산제어 시스템으로 공통메모리나 FIFO 큐를 이용한 프로세서간 통신기능이 요구된다.

4) 데이터 입출력 기능

입출력 장치를 통한 데이터의 입출력을 위해 입출력

제어기능이 요구된다.

5) 다양한 타이머 기능

ISDN 서비스의 효율적으로 지원하기 위해선 타임 단위별 다양한 타이머 기능과 주기성 TASK의 등록과 같은 기능이 제공되어야 한다.

6) 각종 인터럽트의 처리기능

시스템 내외부에서 발생하는 각종 인터럽트에 대한 신속한 처리가 요구된다. 즉, 인터럽트의 지연시간이나 응답시간이 최소화되도록 한다.

7) 각종 부가기능 및 예외상황 대처기능

현재 시각의 조화나 수정과 같은 유용한 부가기능과 시스템의 오동작시에 적절한 조치를 취하는 예외상황 대처기능이 요구된다.

8) 시스템의 확장성과 사용의 용이성

운영체제는 기능의 수정 및 추가가 용이하도록 만들고, 사용자 인터페이스는 운영체제로의 접근이 용이하도록 만든다.

III. 실시간 운영체제의 설계

운영체제는 그림 1과 같이 4 개의 계층으로 구분되며 각 계층을 모듈화시킴으로써 운영체제의 재구성 및 기능의 추가와 삭제가 용이하도록 하였다¹⁵⁻⁶⁾.

계층 1은 수행 대기중인 시스템 TASK들을 수행한 후, 우선순위가 가장 높은 TASK에게 제어를 할당하는 디스패치 관리자이다. 시스템 TASK는 인터럽트 서비스 루틴의 수행시간을 짧게하기 위한 시스템 레벨의 TASK로서 인터럽트 발생시 인터럽트 서비스 루틴내에서는 필수기능만을 수행하고 디스패치 루틴내에서 해당 작업을 하게 하는 기법을 말한다.

계층 2는 TASK간의 메시지 송수신을 지원하는 TASK간 통신처리, 지정한 시간후에 해당 TASK로 메시지를 넘겨주는 타임메시지 처리, 다른 프로세서와의 메시지 송수신을 지원하는 프로세서간 통신처리, TASK간의 동기화 및 상호배제를 지원하는 세마포 처리, 각 TASK의 상태와 우선순위를 변경하고 다중 TASK의 제어를 관장하는 TASK상태 처리, 10 msec 간격으로 클럭 인터럽트가 발생시마다 타임메시지, 주기성 타임메시지 및 실행TASK의 무한루프를 점검하여 처리하는 실시간클럭 처리, 시스템내에서 발생하는 모든 오류에 적절한 조치를 취하는 예외상황 처리, 입출력 장치를 통한 데이터의 입출력을 지원하는 데이터 입출력처리, 시스템의 오동작시 실행정지

나 리셋 요구시 초기화를 수행하는 리셋·실행중지 처리기, 그리고 현재 시각의 조희나 수정과 같은 각종 부가기능을 지원하는 부가기능 처리기로 구성되어 있다.

계층 3은 응용프로그램이 소프트웨어 인터럽트를 이용하여 시스템호출(system call)을 하면 그 타당성을 검토하여 올바르면 요구한 기능을 활성화시켜 주는 시스템호출 관리자이다.

계층 4는 음성 혹은 데이터 호처리, 시스템의 유지보수 등의 기능을 수행하는 교환기의 응용프로그램으로 이를 사용자 타스크라 한다.

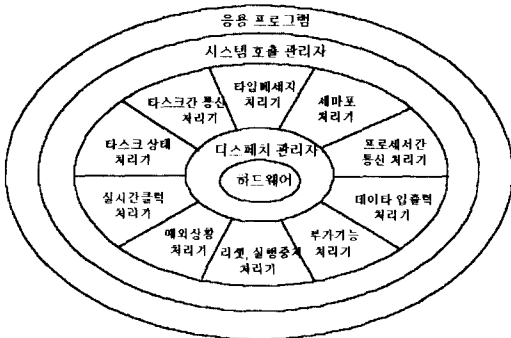


그림 1. ISDN 사설교환기 운영체제의 구조
Fig. 1. Structure of operating system for ISDN PABX.

1. 운영체제의 데이터 흐름

그림 2는 운영체제의 데이터 흐름을 나타낸 것으로 입출력 장치, 신호처리 모듈, 사용자 타스크간의 메시지 교환을 도시하였다.

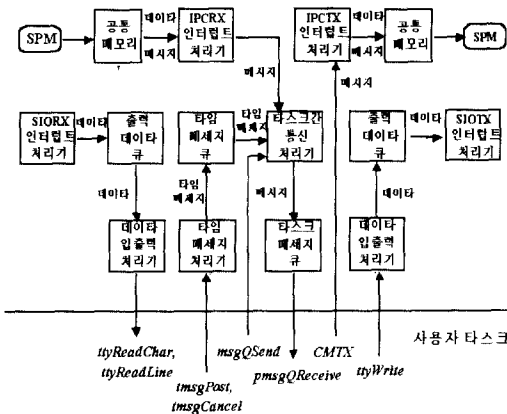


그림 2. ISDN 사설교환기 운영체제의 데이터 흐름
Fig. 2. Data flow of operating system for ISDN PABX.

2. 운영체제의 제어흐름

그림 3은 운영체제의 제어흐름을 나타낸 것으로, 사용자 타스크, 시스템호출 관리자, 디스케치 관리자, 인터럽트 처리기 상호간의 제어흐름을 도시하였다.

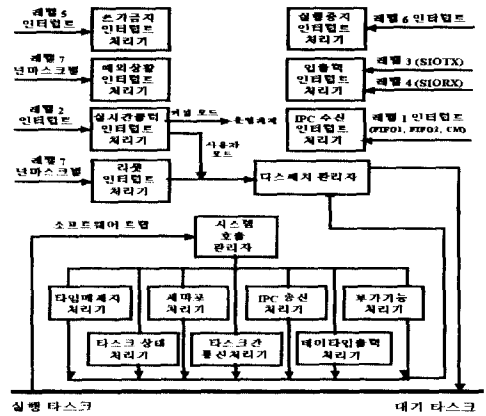


그림 3. ISDN 사설교환기 운영체제의 제어 흐름
Fig. 3. Control flow of operating system for ISDN PABX.

IV. 실시간 운영체제의 구현

본 운영체제는 시스템의 부팅 및 초기화, 다중 타스크의 실시간 병렬처리, 타스크간 통신 및 동기화, 프로세서간 통신, 데이터의 입출력, 인터럽트의 처리, 시스템 자원의 관리와 실시간 처리를 위한 다양한 기능들이 모토롤라 68020 마이크로프로세서상에 구현된 교환기 전용 실시간 운영체제이다 [7-8].

본 운영체제는 그림 4와 같이 크게 4 개의 관리자로 구성되고 모듈구조로 되어 있어 운영체제의 재구성 및 기능의 추가와 삭제가 용이하다. 시스템호출 관리자는 사용자의 시스템호출이 올바른지를 검증한 후, 해당 처리기로 제어를 분배하는 기능을 관장한다. 서비스 관리자는 사용자의 요구에 따른 다양한 서비스를 실시간으로 제공하는 기능을 관장한다. 디스케치 관리자는 시스템 내외부에서 발생한 이벤트를 처리하고 우선순위가 가장 높은 타스크에게 제어를 할당하는 기능을 관장한다. 그리고 인터럽트 관리자는 시스템 내외부에서 발생한 인터럽트의 처리를 관장한다.

1. 시스템호출 관리자

사용자 응용프로그램이 운영체제로부터 서비스를 받기 위해 소프트웨어 인터럽트를 이용하여 시스템호출

을 하면 시스템호출 관리자는 시스템호출의 종류와 파라메타에 대한 검증을 한 후, 해당 서비스 처리기로 제어를 넘기는 기능을 수행한다. 운영체제가 지원하는 시스템호출은 표 1과 같다.

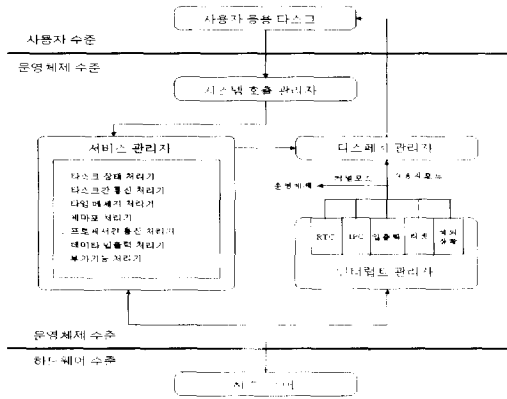


그림 4. ISDN 사설교환기 운영체제의 구성
Fig. 4. Organization of operating system for ISDN PABX.

표 1. ISDN 사설교환기 운영체제의 시스템호출
Table 1. System calls of operating system for ISDN PABX.

시스템호출	기능
<i>taskInit</i>	타스크를 초기화시킨다.
<i>taskActivate</i>	타스크를 구동한다.
<i>taskDeactivate</i>	타스크를 정지시킨다.
<i>taskSuspend</i>	타스크의 수행을 일시 정지시킨다.
<i>taskResume</i>	일시 정지된 타스크의 수행을 재개한다.
<i>taskPrioritySet</i>	타스크의 우선순위를 변경한다.
<i>taskPriorityGet</i>	타스크의 우선순위를 읽는다.
<i>msgQReceive</i>	자신의 메시지 큐로부터 메시지를 읽는다.
<i>msgQSend</i>	해당 타스크의 메시지 큐로 메시지를 전송한다.
<i>tmmsgPost10</i>	10 msec 단위의 타임메세지를 등록한다.
<i>tmmsgPost100</i>	100 msec 단위의 타임메세지를 등록한다.
<i>tmmsgPost1000</i>	1 초 단위의 타임메세지를 등록한다.
<i>tmmsgCancel10</i>	등록한 10 msec 단위의 타임메세지를 취소한다.
<i>tmmsgCancel100</i>	등록한 100 msec 단위의 타임메세지를 취소한다.
<i>tmmsgCancel1000</i>	등록한 1 초 단위의 타임메세지를 취소한다.
<i>semTake</i>	타스크간 동기화를 위해 세마포 사용을 요구한다.
<i>semGive</i>	사용한 세마포를 반환한다.
<i>CMTX</i>	CPM-SPM의 프로세서간 통신을 요구한다.
<i>FIFOIX1</i>	CPM-IPM의 프로세서간 통신을 요구한다.
<i>FIFOIX2</i>	CPM-CPM의 프로세서간 통신을 요구한다.
<i>ttyDevInit</i>	데이터 입출력 장치를 초기화한다.
<i>ttyDevCtrl</i>	데이터 입출력 장치의 통신모드를 변경한다.
<i>ttyReadChar</i>	데이터 입출력 장치를 통해 한 자를 읽는다.
<i>ttyReadLine</i>	데이터 입출력 장치를 통해 한 줄을 읽는다.
<i>ttyWrite</i>	데이터 입출력 장치를 통해 데이터를 출력한다.
<i>clockRead</i>	시계칩의 현재 시각을 읽는다.
<i>clockSet</i>	시계칩의 현재 시각을 수정한다.
<i>msgTrace</i>	시스템내 메시지 흐름의 추적을 요구한다.
<i>memoryProtect</i>	메모리에 쓰기금지 영역을 설정한다.
<i>pmmsgChange</i>	추기성 타임메세지의 제어 테이블을 수정한다.

2. 서비스 관리자
사용자의 요구에 따른 다양한 서비스를 실시간으로 제공하는 서비스의 관리자로서 타스크 상태 처리기, 타스크간 통신 처리기, 타임메세지 처리기, 세마포 처리기, 프로세서간 통신 처리기, 데이터 입출력 처리기, 부가기능 처리기 등으로 구성되어 있다.

1) 타스크 상태 처리기

사용자의 요청에 따라 타스크의 상태와 우선순위를 변경하고 우선순위에 기초하여 다중 타스크를 병렬처리하는 다중 타스크의 제어를 관장한다. 이를 위해 실행중인 타스크에 대기조건이 발생하여 수행을 계속할 수 없을 경우 그 원인에 따라 해당 대기큐에 등록하고, 대기 타스크의 특성과 실행상태를 타스크 제어블록 (Task Control Block: TCB)에 보존한 후, 우선순위가 가장 높은 실행 대기중인 타스크를 수행한다¹¹⁾. 타스크의 상태는 CPU를 할당받아 실행하는 실행상태, CPU 할당을 기다리는 실행대기 상태, 에세지를 기다리는 메세지대기 상태, 세마포 사용이 완료되기를 기다리는 세마포대기 상태, 입출력이 완료되기를 기다리는 입출력대기 상태, 수행이 정지되어 구동되길 기다리는 실행정지 상태로 구분된다. 각 타스크는 시스템호출, 자원할당 여부, 내외부 이벤트에 의해 상태천이를 일으키며, 타스크의 상태 천이도는 그림 5와 같다.

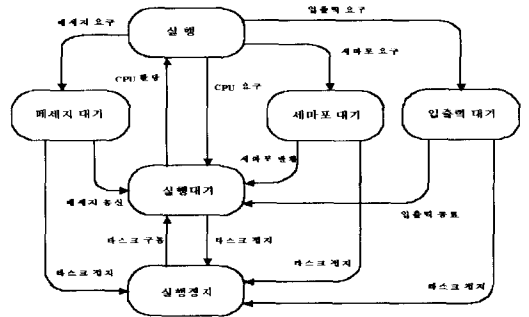


그림 5. 타스크의 상태 천이도
Fig. 5. Task state transition diagram.

2) 타스크간 통신 처리기

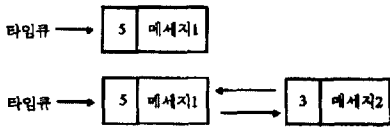
타스크간의 메세지 송수신을 위한 기능을 수행한다. 각 타스크는 자신의 메세지 큐를 가지고 있어 해당 타스크의 메세지 큐로 메세지를 송신하고, 그 타스크는 자신의 메세지 큐로부터 수신된 메세지를 읽음으로써 타스크간 통신이 이루어진다. 그리고 긴급메세지 처리 기능을 구현하여 긴급메세지를 전송하면 해당 타스크

가 즉시 동작하여 긴급메세지를 우선 처리하도록 하였다. 또한 특정 TASK의 메세지 큐를 보류 혹은 해제하는 기능을 구현하여 해당 TASK의 수행을 일시적으로 중지시키거나 재개할 수 있도록 하였다.

3) 타임메세지 처리기

실시간 서비스의 지원을 위해 사용자 TASK가 등록된 타임메세지를 지정한 시간 후에 되돌려 주거나 등록된 타임메세지를 취소하는 기능을 수행하며, 응용에 따라 10 msec, 100 msec, 1 sec 단위로 타임메세지의 등록 및 취소가 가능하다. 타임메세지의 등록과 처리는 Δt 알고리즘을 이용하였고, 이의 처리과정을 그림 6에 예시하였다. 타임메세지 큐에 타임메세지의 등록시 첫번째 메세지는 요구된 타임대로 등록하고, 두번째 메세지는 타임이 먼저 등록된 메세지의 타임보다 큰 경우에는 그 차이만큼만 등록한다(a). 만약 타임이 작을 경우에는 그 차이만큼을 첫번째 메세지의 타임으로 하고 두번째 메세지의 타임은 그대로 한 후, 메세지의 순서를 바꾸어 등록한다(b). 이와 같은 방법을 통해 큐에 연결된 모든 타임메세지는 첫번째 메세지의 타임을 감소시킴으로써 전체 메세지들의 타임을 감소시키는 효과를 볼 수 있으며, 이를 양방향 링크 큐로 구현함으로써 노드의 삽입과 제거가 용이하도록 하였다.

(a) 타임이 5인 메세지1을 먼저 등록하고, 다음에 타임이 8인 메세지2를 등록한 경우



(b) 타임이 5인 메세지1을 먼저 등록하고, 다음에 타임이 3인 메세지2를 등록한 경우

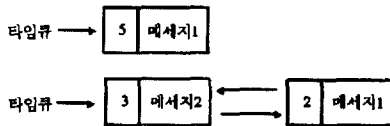


그림 6. 타임메세지의 등록 절차
Fig. 6. Registration procedure of timed messages.

4) 프로세서간 통신 처리기

공통메모리나 FIFO 큐를 이용하여 다른 프로세서상의 TASK와 메세지 통신을 위한 기능을 제공한다. 공통메모리와 FIFO 큐는 송신영역과 수신영역으로 구분되며 송신영역에 보낼 메세지를 쓴 후 상대 프로세서로 인터럽트를 발생시키면, 상대 프로세서가 수신영역

에서 메세지를 읽어 해당 TASK로 메세지를 전송한다. 공통메모리는 CPM-SPM간의 통신에 이용되고, FIFO1 큐는 CPM-IPM간의 통신에 이용되며, FIFO2 큐는 이중화를 위한 CPM-CPM간의 통신에 이용된다¹⁹⁾.

5) 세마포 처리기

다중 TASK간의 상호협력이나 상호배제를 위해 이진 세마포와 카운팅 세마포를 구현하였고, TASK간 동기화를 위한 이진 세마포의 구현 알고리즘은 그림 7과 같다. 세마포 변수(S)는 사용가능한 자원의 수를 나타내며 카운팅 세마포의 동시에 최대 S 개의 TASK가 접근 가능하며 이진 세마포의 경우 1 개의 TASK만이 접근 허용된다.

```
semTake : P(S) 오퍼레이션
S = S - 1;
if (S < 0)
    현재 실행중인 TASK의 상태를 세마포대기 상태로 바꾼다.
    실행 TASK를 세마포 대기큐에 등록한다.
    스케줄링 요구 플래그를 셋트한다.

semGive : V(S) 오퍼레이션
S = S + 1;
if (S <= 0)
    세마포 대기큐에서 하나의 TASK를 꺼낸다.
    그 TASK의 상태를 실행대기 상태로 바꾼다.
    스케줄링 요구 플래그를 셋트한다.
```

(여기서 S는 세마포 변수로 초기값은 1 이다.)

그림 7. 이진 세마포의 기본 알고리즘
Fig. 7. Basic algorithm of binary semaphore.

6) 데이터 입출력 처리기

설정된 통신모드에 따라 터미널 혹은 프린터 등의 입출력 장치를 통해 데이터의 입출력 서비스를 제공한다. 즉, 입출력 장치를 통해 데이터의 출력만을 위한 기본서비스, 아스키 문자열의 입출력을 위한 터미널서비스, TASK 메세지의 입출력을 위한 응용서비스를 제공한다. 통신모드의 설정은 ttyDevCtrl 시스템호출을 이용하여 선택 가능하며, 입출력시 편집기능, 흐름제어와 오류제어 기능을 제공한다.

7) 부가기능 처리기

시계칩의 현재 시각을 조회 및 수정하거나 메모리에 쓰기금지 영역을 설정하는 등의 유용한 부가기능을 제공한다.

3. 디스켓 관리자

사용자 TASK의 실행중 실시간클럭 인터럽트가 발생하거나 시스템호출을 하면 해당 처리기를 수행한 후,

제어는 디스패치 관리자로 넘겨진다. 디스패치 관리자는 우선 시스템 내외부에서 발생한 모든 이벤트에 대한 적절한 처리를 한 후, 우선순위에 기초한 타스크 스케줄링을 한다. 이때 수행중인 타스크가 non-preemptive 타스크이면 그 타스크로 제어를 넘기고 preemptive 타스크이면 재 스케줄링하여 우선순위가 가장 높은 타스크로 제어를 넘긴다. 시스템 내외부의 이벤트로는 타임메세지나 주기성 타임메세지의 시간 종료, 프로세서간 송수신 메세지나 데이터 입출력 메세지의 발생, 시스템 오류나 각종 인터럽트의 발생 등이 있다.

4. 인터럽트 관리자

시스템의 내외부에서 발생하는 인터럽트에 대한 처리하는 인터럽트의 관리자로서 실시간클럭, 프로세서간 통신, 입출력, 리셋, 예외상황, 실행중지, 쓰기금지 등의 인터럽트 처리를 위한 처리기로 구성되어 있다.

1) 실시간클럭 인터럽트 처리기

실시간클럭 인터럽트는 10 msec 주기로 발생하는 인터럽트로서 이의 발생시마다 타임메세지와 주기성 타임메세지의 시간 경과여부를 점검하여 시간이 경과하였으면 메세지를 해당 타스크로 전송한다. 이는 또한 우선순위에 기초한 타스크 스케줄링을 수행하는 기본 시간 주기이다.

2) 프로세서간 통신 인터럽트 처리기

다른 프로세서상의 타스크로부터 전달된 메세지를 수신하기 위한 인터럽트 처리루틴으로 이 인터럽트가 발생하면 공통메모리 혹은 FIFO 큐로부터 메세지를 읽어 해당 타스크로 전송한다.

3) 입출력 인터럽트 처리기

시리얼 포트를 통한 데이터의 입출력시 발생하는 인터럽트를 처리하는 루틴이다.

4) 리셋 인터럽트 처리기

시스템 리셋 혹은 power-on시 시스템을 초기화하는 인터럽트 처리루틴이다.

5) 예외상황 인터럽트 처리기

illegal instruction, divide by zero, endless loop, power fail, address error, bus error, 등의 예외상황이 발생하였을때 적절한 조치를 취하는 인터럽트 처리루틴이다.

6) 실행중지 인터럽트 처리기

시스템의 오동작시 실행을 중지시키거나 수행중인 프로그램의 상태를 알고자할 때 이용되는 인터럽트 처리루틴이다.

7) 쓰기금지 인터럽트 처리기

시스템의 정적 데이터와 프로그램을 보호할 목적으로 쓰기가 금지된 메모리 영역에 쓸 경우 수행되는 인터럽트 처리루틴이다.

V. 성능평가

본 장에서는 사용자의 시스템호출에 대한 응답시간, 시스템 내외부에서 발생한 인터럽트 및 이벤트의 처리 시간, 타스크 스케줄링과 타스크 스위치의 수행시간을 측정함으로써 실시간 운영체제의 성능을 평가를 하였다.

1. 성능평가 방법

운영체제의 성능평가를 위해 먼저 응용프로그램의 시스템호출로부터 요구된 기능을 수행하고 다시 응용프로그램으로 되돌아가기까지의 수행흐름을 분석하여 해당 기능을 위한 수행루틴들을 우선 파악하였다. 다음은 설정한 전제 조건하에 각 루틴별 수행시간을 실행의 최단경로, 통상의 실행경로, 실행의 최장경로로 나누어 모토롤라 68000 계열의 디버깅 툴인 XRAY68K¹¹⁾를 이용하여 최단, 일상 및 최장 수행시간을 측정하였다. 시스템호출 기능의 최단, 일상, 최장 수행시간은 해당 기능을 위한 모든 수행루틴의 최단, 일상, 최장 수행시간을 각각 합산한 값이 된다. 그런데 XRAY68K를 이용하여 측정한 수행시간의 기본단위는 클럭이며 본 ISDN 사설교환기의 중앙처리 모듈은 모토롤라 68020 마이크로프로세서를 사용하고 CPU 속도는 16 MHz이므로 한 클럭 사이클의 시간은 62.5×10^{-9} ($=1/(16 \times 10^6)$) 초이다. 그래서 XRAY68K를 이용하여 측정한 총 클럭수에 62.5×10^{-9} 를 곱하여 각 루틴 및 기능별 수행시간을 구하였다.

2. 성능평가의 전제조건

교환시스템내 운영체제의 성능평가를 위해선 실제 가입자의 통화나 통신량을 달리하여 최대용량에 도달하였을 때에 운영체제 각 기능의 수행시간을 측정 분석하여야 한다. 하지만 현실적으로 이러한 Worst Case의 상황 설정에는 여러 어려움이 따른다. 또한 수행루틴의 최단, 일상, 최장 실행경로가 동일한 경우에도 시스템 부하에 따라 수행시간이 달라진다. 예를 들어 실행 대기큐내 타스크의 갯수에 따라 스케줄링시 우선순위의 비교 횟수가 달라지므로 수행시간이 달라

질 수 있다. 그래서 가변적인 시스템 변수는 일반적인 상수값으로 대체하고, 발생 가능성이 희박한 상황은 무시한 후, 운영체제에 인위적인 부하(load)를 걸어 이의 성능을 평가하였다.

- (1) 타스크간 또는 프로세서간 통신시 한번에 1 개의 메세지만을 송수신하는 것으로 하고 송수신 메세지의 평균길이는 30 Byte이므로 크기는 30 Byte로 한다.
- (2) 시스템내에는 타스크 제어용으로 실행대기 큐, 세마포대기 큐, 메세지 대기큐, 입출력 대기큐가 있고, 타스크간 통신용으로 각 타스크별로 256 Bytes의 메세지 큐를 가지며 프로세서간 통신용으로 1024 Bytes의 CM 큐와 각 512 Bytes의 FIFO1, FIFO2, SIO 큐를 가진다. 그리고 타임 메세지용으로 10 msec / 100 msec / 1 초 타임큐가 있으며 각각 최대 10 개의 타임메세지를 수용 가능하다.
- (3) 시스템 부하는 시스템내 모든 큐에 50 % 정도의 메세지나 TCB가 차있는 경우로 한다. 즉, 시스템내 응용타스크의 수는 8 이므로 타스크 제어큐에는 4 개의 TCB가 타스크간 통신큐에는 4 개의 메세지를 타임메세지 큐에는 5 개의 타임메세지가 있고, CM 큐에는 송수신 각각 8 개의 메세지가 FIFO1, FIFO2, SIO 큐에는 송수신 각각 4 개의 메세지가 있는 것으로 한다.
- (4) 시스템 타스크는 운영체제내에 처리 대기중인 이벤트의 수에 따라 최단, 일상, 최장 수행시간이 결정된다. 따라서 처리 대기중인 이벤트가 전혀 없는 경우는 최단이 되고, 모든 이벤트가 처리 대기중인 경우는 최장이 된다. 그리고 각 타임단위별로 처리해야 할 타임메세지가 1 개씩 있을 때를 일상인 경우로 한다.

3. 운영체제의 성능평가

운영체제의 성능평가를 위해 먼저 각 수행루틴의 수행시간을 측정후, 해당 기능을 수행하기 위한 수행루틴들의 수행시간을 합산하여 운영체제의 기능별 수행시간을 구하였다.

1) 시스템호출 관리자

사용자가 요구한 시스템호출을 분석하여 해당 서비스 처리기로 제어를 분배하는 루틴으로 수행시간은 표 2와 같다.

2) 디스페치 관리자

디스페치 관리자는 preemptive 타스크 스케줄링을 지원하지만 preemptive 커널인 경우 우선순위 무한역전(unbounded priority inversion)^[11] 등의 치명적인 문제가 발생 가능하므로 커널은 non-preemptive로 구현하였다. 이로 인해 인터럽트의 발생시 관련 타스크가 구동되기까지 걸리는 인터럽트 타스크 응답시간(interrupt task response time)은 증가하게 된다. 인터럽트 타스크 응답시간은 최장의 시스템호출 시간(=156.9 μ sec), 시스템 타스크 수행시간(=661.0 μ sec), 스케줄링 시간(=35.9 μ sec)을 더한 853.8 μ sec가 된다. 디스페치 관리자내에서 수행되는 시스템 타스크는 처리되어야 할 이벤트의 수에 따라 수행시간이 결정되고, 스케줄링 루틴은 스케줄링 여부, 실행대기 타스크의 유무, 실행중인 타스크의 특성(preemptive/non-preemptive), 타스크 스위칭 여부에 따라 수행시간이 결정되며 수행시간은 표 3과 같다.

표 2. 시스템호출 관리자의 수행시간
Table 2. Execution time of system call manager.

(단위: μ sec)

루틴 이름	수행 시간		
	최 단	일 상	최 장
시스템호출 관리자	12.1	12.1	12.1

표 3. 디스페치 관리자의 수행시간

Table 3. Execution time of dispatch manager.

(단위: μ sec)

루틴 이름	수행 시간		
	최 단	일 상	최 장
시스템 타스크	4.4	386.3	661.0
스케줄링 루틴	12.9	20.3	35.9

3) 서비스 관리자

본 운영체제가 제공하는 모든 서비스의 수행시간은 표 4와 같다.

4) 인터럽트 관리자

인터럽트 처리의 성능평가에 있어 인터럽트의 지연시간(interrupt latency time)과 인터럽트의 응답시간(interrupt response time)은 중요하다. 전자는 인터럽트의 발생시점부터 인터럽트 처리루틴(ISR)이 구동되기까지 걸리는 시간을 말하고, 후자는 복수의 인터럽트가 동시에 발생하였을 경우 개개의 ISR이 구동되기

까지 걸리는 시간을 말한다. 인터럽트의 지연시간은 프로세서의 지연시간(=0), ISR의 구동까지의 시간(=1.8 μsec), 운영체제의 인터럽트 금지시간(= 5.8 μsec)을 더한 7.6 μsec 이다. 인터럽트의 응답시간은 인터럽트의 지연시간에 최장 ISR의 수행시간(=122.9 μsec)과 ISR의 구동까지의 시간(=1.8 μsec)을 더한 132.3 μsec 가 된다. 시스템 내부에서 발생한 인터럽트를 처리하는 인터럽트 처리루틴의 수행시간은 표 5와 같다.

표 4. 서비스 관리자의 수행시간
Table 4. Execution time of service manager.
(단위: μsec)

시스템호출 기능	수행 시간		
	최단	일상	최장
taskInit	68.6	68.6	110.2
taskActivate	18.9	18.9	18.9
taskDeactivate	66.7	66.7	92.7
taskSuspend	96.5	96.5	113.9
taskResume	4.6	4.6	4.6
taskPrioritySet	12.4	12.4	13.9
taskPriorityGet	9.6	9.6	9.6
msgQReceive	17.1	78.8	81.6
msgQSend	84.3	101.3	120.1
tmsgPost	54.1	78.3	78.3
tmsgCancel	20.4	21.3	21.3
semTake	12.1	12.1	18.6
semGive	12.2	12.2	35.1
CMTX	37.1	37.1	37.1
FIFOTX1	42.6	47.8	47.8
FIFOTX2	42.6	47.8	47.8
ttyReadChar	22.5	26.1	29.7
ttyReadLine	36.9	54.8	54.8
ttyWrite	37.4	55.3	55.3

5) 시스템호출 기능

응용프로그램에서 요구한 기능을 수행하고 다시 응용프로그램으로 되돌아갈 때까지 시스템호출 기능의 총 수행시간은 앞서 측정한 수행시간표를 기초로 아래의 방법으로 구하여 표 6에 나타내었다. 시스템호출 기능의 일상(최단, 최장) 총 수행시간은 시스템호출 관리자의 일상 수행시간, 서비스 관리자의 일상(최단, 최장) 수행시간, 디스패치 관리자내 시스템 타스크의 최단 수행시간, 디스패치 관리자내 스케줄링 루틴의 일상 수행시간의 총합이 된다.

표 5. 인터럽트 관리자의 수행시간
Table 5. Execution time of interrupt manager.
(단위: μsec)

루틴 이름		수행 시간		
		최단	일상	최장
실시간클럭 처리루틴	커널모드	21.2	21.2	101.1
	사용자모드	41.3	41.3	122.9
FIFO RX1			108.7	
FIFO RX2			108.7	
CM RX			19.3	
SIO TX			10.3	
SIO RX			23.9	
쓰기금지			10.4	
리셋, 예외상황, 실행중지		16.1	16.1	16.5

표 6. 시스템호출 기능의 수행시간
Table 6. Execution time of system call functions.
(단위: μsec)

시스템호출 기능	수행 시간		
	최단	일상	최장
taskInit	105.4	105.4	147.0
taskActivate	55.8	55.8	55.8
taskDeactivate	103.5	103.5	129.5
taskSuspend	133.3	133.3	150.7
taskResume	41.4	41.4	41.4
taskPrioritySet	49.3	49.3	50.7
taskPriorityGet	46.4	46.4	46.4
msgQReceive	53.9	115.6	118.4
msgQSend	121.1	138.1	156.9
tmsgPost	90.9	115.1	115.1
tmsgCancel	57.3	58.1	58.1
semTake	48.9	48.9	55.4
semGive	49.0	49.0	71.9
CMTX	73.9	73.9	73.9
FIFOTX1	79.4	84.6	84.6
FIFOTX2	79.4	84.6	84.6
ttyReadChar	59.3	62.9	66.5
ttyReadLine	73.8	91.6	91.6
ttyWrite	74.3	92.1	92.1

4. 고찰

앞서 운영체제의 성능평가를 위해 각 수행루틴별 수행시간을 측정하고 이를 기초로 시스템호출의 수행시간, 인터럽트의 처리시 발생하는 인터럽트 지연시간과

응답시간, non-preemptive 커널구조에 기인한 인터럽트 타스크의 응답시간을 구하였다. 이는 교환시스템내 응용프로그램 관점에서의 성능이고, 교환시스템 사용자 관점에서의 운영체제 성능은 세마포대기, 메시지대기, 입출력대기로 인한 타스크의 최대 대기시간에 영향을

받는다.

대기조건의 발생으로 인해 대기상태로 빠진 타스크의 실행이 재개되기까지는 최소 2 번의 타스크 스위치와 대기조건의 해제를 위한 시스템호출이 요구된다. 타스크의 최대 대기시간은 스케줄링 방식, 시스템내 타스

표 7. 상용 실시간 운영체제와의 비교

Table 7. Comparison with commercial real-time operating systems.

항목		상용 실시간 운영체제								본논문
대분류	소분류	VRTX32	FlexOS	GALAXY	µSOS+ 68K	CROS	iRMX	RMS68K	IXOS	
1. 일반사항	개발회사	Ready System	Digital Research	Hughes Comm.	Software Component Group	ETRI	인텔	모토롤라	삼성전자	
	프로세서	680X0, 80X86	680X0, 80X86	68000	680X0, 8X86	680X0	80X86	680X0	680X0	
	개발상품	항공제어	POS	통신위성	통신, 화상, 의료기기	TDX-10	통신기기	통신기기	ISDN 교환기	
	커널크기	8KB	150KB	48KB	5.5KB	288KB	34KB	8.5KB	13KB	
	프로그래밍 언어	C언어, ASM	C언어	C언어, ASM	C언어, ASM	C언어, ASM	C언어, ASM	C언어	C언어, ASM	
2. 스케줄링 방식	Priority	0	0	0	0	0	0	0	0	
	Round-Robin	0	0	0	0	0	0	0	0	
	Preemption	0	0	0	0	0	0	0	0	
3. 타스크관리	타스크 갯수	256	무제한	13(256)	6335	256	8192	무제한	256	
	우선순위 레벨	256	256	256	255	256	256	256	256	
	실시간 다중처리	0	0	0	0	0	0	0	0	
	타스크 생성	0	0	0	0	0	0	0	0	
4. 타스크간 통신 및 동기화	타스크간 통신	메일박스, 메시지 큐	파이프	메시지 큐	메시지 큐	메일박스	메일박스	메시지 큐	메시지큐	
	타스크간 동기화	이벤트의 시그널	이벤트와 시그널, 세마포	메시지 큐	메시지 큐	세마포	세마포	시그널	세마포	
5. 프로세서간 통신	SIO 통신						시그널 이용	버스에 메시지 전달	0	
	FIFO 통신			0		0				
	CM 통신					0				
6. 메모리 관리	동적 할당	0	0		0	0	0	0		
	정적 할당			0						
7. 타이머 관리	주기성 타스크					0		0	0	
	타임 메세지	0		0	0	0	0	0	0	
	Sleep	0	0		0	0	0	0		
8. 입출력 관리	문자 I/O		문자, 블록, 스트림	문자, 스트림		문자, 블록, 스트림	문자		문자, 스트림	
	블록 I/O									
	스트림 I/O									
9. 기타 관리	예외처리			0		0	0	0	0	
	초기화 변경	0			0		0		0	

크의 갯수, 특성, 우선순위와 응용프로그램의 속련도에도 영향을 받는다. 세마포를 예로 들어 세마포대기 상태에 있는 TASK의 최대 대기시간을 말하면 우선 세마포를 가지고 있는 TASK가 제어를 받을 때까지의 TASK 스위치 시간, 제어를 거쳤던 각 TASK의 수행 시간, 세마포 반환을 위한 시스템호출의 수행 시간, 최종적으로 세마포를 기다리는 TASK크로의 스위치 시간의 총합이되며 그 외에도 수행이 재개되기까지 시스템 내에 발생한 인터럽트와 이벤트의 처리시간이 부가된다.

현재 수많은 실시간 운영체제가 통용되고 있으며 각 제조사마다 성능평가 자료를 공지하고 있지만 각 사마다 프로세서, 클럭주기, 메모리 속도가 서로 다른 하드웨어 상에서 성능을 평가하여 이들 수치간의 직접적인 비교로 의미있는 판단을 내리는데는 어려움이 있다. 예를 들면 TASK 스위치 시간은 일반적으로 TASK간의 문맥교환과 스케줄링 시간을 함께 말하지만 공지된 평가자료에는 스케줄링 시간을 제외시키는 경우가 많다. 그러므로 구현한 운영체제와 다른 상용 실시간 운영체제의 비교를 기능적 측면에서 표 7에 정리하였다.

VI. 결 론

본 논문에서는 교환시스템 응용프로그램의 다양한 요구에 대해 제한된 시간내에 신속한 실시간 응답을 하는 실시간 운영체제의 구현 및 성능평가에 대해 논하였다.

구현한 운영체제는 TASK의 우선순위에 기초하여 다중 TASK를 실시간으로 병렬처리하고, 시스템 내외부에서 발생한 각종 인터럽트와 이벤트에 대해서도 신속한 실시간 처리를 하며, 시스템의 부팅과 초기화, TASK간 통신과 동기화, 프로세서간 통신, 데이터 입출력과 실시간 응용을 위한 다양한 실시간 처리기능을 제공한다. 각 기능들은 계층구조와 모듈구조로 설계하고 구현하여 기능의 추가와 수정이 용이하도록 하였다. 운영체제의 실행속도는 전체 시스템의 성능에 미치는 파급효과가 크므로 빠른 처리를 위하여 각 기능은 어셈블리 언어로 구현하였고, 인터페이스부는 사용자의 이용이 쉽도록 C 언어로 구현하였다.

그리고 구현한 운영체제의 각 기능별 수행시간을 최단, 일상 및 최장의 경우로 나누어 측정된 결과 사용자의 시스템호출에 대한 응답시간은 41.4 ~ 156.9 μsec ,

시스템 내외부에서 발생한 인터럽트에 대한 처리시간은 10.3 ~ 122.9 μsec , 인터럽트의 지연시간은 7.6 μsec , 인터럽트의 응답시간은 132.3 μsec , 인터럽트 TASK 응답시간은 853.8 μsec , 시스템 내외부에서 발생가능한 모든 이벤트에 대한 처리시간은 4.4 ~ 661.0 μsec , TASK 스케줄링과 TASK 스위치에 요구되는 시간은 12.9 ~ 35.9 μsec 이었다. 그리고 구현한 운영체제와 다른 상용 실시간 운영체제와의 기능적 측면에서 비교 분석하였다.

참 고 문 헌

- [1] 최 재원, 박 인갑, "디지털 토폰 교환기의 실시간 운영체제 구현", 대한전자공학회 추계종합학술대회 논문집, 제 16권, 제 2호, pp. 140-144, 1993
- [2] 삼성전자, *Information Exchange System (INFOREX) 시스템 사용자 매뉴얼*, 통신연구소 교환연구실, July 1993
- [3] 최 재원, 박 인갑, "ISDN 공중망 접속을 위한 사설교환기의 ISDN BRI 트렁크 카드의 구현", 대한전자공학회 논문지, 제 33권 A편, 제 9호, 1996
- [4] B. Alan and W. Andy, *Real-Time Systems and their Programming Languages*, Addison-Wesley, Reading, Massachusetts, 1990.
- [5] M. J. Bach, *The Design of the UNIX Operating System*, Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
- [6] 배 기원 외, "사설교환기의 실시간 커널 구현 및 성능 분석", 한국정보과학회 가을학술발표 논문집, 제 18권, 제 2호, pp. 445-448, 1991
- [7] Motorola, *M68020 8 /16 /32 Bit Microprocessors User's Manual*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [8] Applied Microsystems Corporation, *MCC68K C Compiler User's Manual and ASM68K Documentation Set for 680x0 Family Microprocessors*, 1988.
- [9] 최 재원, 박 인갑, "ISDN 사설교환기의 결함내성 구현 및 성능평가", 대한전자공학회 논문지, 제 33권 A편, 제 11호, 1996
- [10] Applied Microsystems Corporation, *VALIDATE / XEI Source-Level Debugger*

User's manual for 68000 Family Micro-processors, 1989.

[11] Linda Thompson, "*Designing for Worst*

Case: The Impact of RTOS Performance on Real-World Embedded Design", Microtec Research Inc., 1993.

저 자 소 개



崔 在 洵(正會員)

1965年 4月 27日生. 1988年 2月 고려대학교 전산학과(학사). 1990年 8月 미국 미시간주립대학교 전산학과(석사). 1995年 8月 건국대학교 전자공학과(박사). 1990年 10月 ~ 현재 삼성전자 정보통신연구소 선임연구원.

주관심분야는 운영 체제, 네트워크, 정보통신망, 이동통신

朴 仁 甲(正會員) 第 31 卷 A編 第 10 號 參照

현재 건국대학교 전자공학과 교수