

□신기술해설□

실시간 JAVA

박 남 섭[†] 이 정 배^{**} 이 승 호^{***}

◆ 목 차 ◆

- | | |
|----------------------|----------------------|
| 1. 서 론 | 3. 실시간 JAVA PERC의 구현 |
| 2. 실시간 JAVA PERC의 설계 | 4. 결 론 |

1. 서 론

자바 기술은 플랫폼의 종류에 관계없이 독립적으로 운영이 될 수 있으며, 인터넷상에서 웹 환경과 더불어 멀티미디어 서비스, 원격 교육, 인터넷 게임, 비디오-폰, 인터넷 TV, 웹 TV 등의 정보 가전, 개인 정보 단말기, 휴대용 PC, 무인 공장, 네트워크 라우터/스위치 등의 통신 장비 분야 등에 핵심적으로 사용되는 정보통신 산업의 기반 기술이다.

이러한 첨단분야에 사용되는 소프트웨어 개발은 기존의 프로그래밍 언어와 개발 환경 사이의 불일치를 야기하였다. 이런 문제점의 해결방안의 일환으로 웹을 기반으로 한 분산 프로그래밍 환경의 표준으로 Sun Microsystems는 최근 자바 프로그래밍 환경을 제안했다. JAVA의 발표 후 지금까지 Netscape과 Oracle 등 다수의 회사들이 JAVA의 기술에 참여하여 개발하고 있는 실정이다. 그리고, 개발 초기에 JAVA는 내장형(embedded) 시스템 소프트웨어로서 개발되었지만[1] 현재는 웹 프로그래밍에서 그 능력을 인정받고 있다[2].

또한 목표 도메인에서의 생산성의 증진에도 한 몫을 하고 있다.

많은 내장형 컴퓨터 시스템에서 제기하고 있는 JAVA의 하드웨어 및 소프트웨어적인 실시간성의 지원에 관해서 현재의 JAVA는 부적합하다는 판정이다. 내장형 시스템의 개발에 있어 다양한 방법들이 제기되고 있다. 최근의 노력은 주로 JAVA의 기본 API에 추가적으로 API를 개발, 그리고 실시간 JVM(Java Virtual Machine)의 개발 등에 초점을 두고 있는 실정이다.

JAVA는 객체지향적이고, 간단하며, 프로그램의 개발 주기가 타 언어에 비해 훨씬 빠르고, 응용 프로그램이 모빌코드로 만들어지므로 이식성(portability)이 매우 높다. 그리고 언어 자체가 강건하며 프로그램 환경에 독립적이다. 또한 보안이 언어 차원에서 지원이 된다. 최근 이러한 타 언어에 대비되는 다양한 강점을 가진 JAVA를 실시간 시스템에 도입하려는 노력이 여러 가지 형태로 이루어지고 있다.

실시간 시스템 개발자라면 위와 같은 JAVA가 갖는 강점을 충분히 고려할 것이다. 그렇지만, 내장형 실시간 시스템 도메인에서 고려해야 하는 중요한 고려사항을 살펴보면 다음과 같다.

먼저, 내장형 실시간 시스템 개발자들은 매우 많은 다른 프로세서를 다루어야만 하는 필요를

[†] 정 회 원 : 부산외국어대학교 컴퓨터공학과 석사과정

^{**} 종신회원 : 부산외국어대학교 부교수

^{***} 종신회원 : (주)한국이시시스템 대표이사

느낄 것이다. 예를 들어 Motorola 68000, Power PC, Intel 960 등의 호스트가 혼재되어 있는 연구실은 매우 빈번할 것이다. 이러한 상황에서 개발자들은 반드시 이식성을 중요하게 고려해야 한다. 그리고, 빈번히 일어 나는 일로 소프트웨어를 추가하는 것이 있다. 이때 반드시 시스템의 재부팅 및 다운을 하지 않고서도 인스턴이 가능해야만 한다. 즉, 동적 연결성(Dynamic Adaptability)을 고려해야 한다. 또한, 네트워크이나 노드의 정전 등의 문제 발생시 정보를 재분배하는 등의 결함에 대한 고장허용(Fault Tolerance)을 고려해야 한다.

이렇게 특별히 내장형 실시간 시스템 개발자에게 고려되어야 하는 중요하고 특별한 특성이 있다. 이런 특성을 고려할 때 최선의 선택은 JAVA 언어가 가지는 강점을 그대로 도입하는 것이 된다. 현재 대다수 실시간 시스템의 개발자들은 기존 JAVA의 강점을 살리면서 여기에 실시간성을 보장하는 API, 실시간 JVM(Java Virtual Machine) 등의 개발에 주력하고 있다. 여기서는 PERC를 개발한 Newmonics사에서 제안하고 있는 일반적인 운영체제 하에서의 실시간성을 보장하는 방법 즉, 실시간 자바의 설계와 구현에 관한 연구를 소개하기로 한다.

본 논문 구성은 1장에서 서론을 기술하고, 2장에서는 PERC로 구현된 JAVA의 확장에 대한 설계에 대한 내용을 살펴 보고 3장에서는 PERC로 구현된 JAVA의 확장에 대한 구현 방법을 살펴 보고 4장에서 결론을 이야기한다.

2. 실시간 JAVA PERC의 설계

2.1 실시간 응용 프로그램 모델

대부분의 실시간 응용프로그램들은 (그림 1)에서 제공되는 이식성이 강한 모델로 설계된다. 실시간 task의 기능은 주기적(Cyclic) task와 계속적(Ongoing)

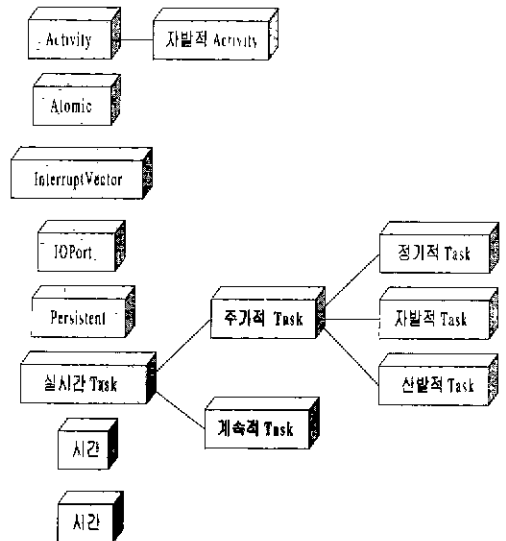
task로 나뉘어지며 주기적 task는 다시 정기적 task, 자발적 task, 산발적 task로 구성된다.

- 정기적(Periodic) Task

실시간 시스템의 많은 activity들은 일관된 수행 시간을 가진 정기적 task들을 가진다. 이러한 task들은 정해진 수행시간에 규칙적으로 호출된다. 예를 들면 펜 컴퓨터의 펜 포인터의 샘플링, 레이더 및 수중음파 신호 입력의 처리 등과 같은 activity들이 있다.

- 산발적(Sporadic) Task

전행적인 실시간 작업의 하나로 중요한 부분을 차지하는 것으로서 특별한 외부 이벤트에 반응하여 활성화되는 산발적인 activity들이 존재한다.



(그림 1) 실시간 패키지의 주요 클래스들

예를 들면 특별한 버튼에 대한 마우스 이벤트, 높은 온도나 압력의 입력에 반응하는 알람, 이전에 확인되지 않은 물체에 대한 레이더의 입력 등이 이에 해당된다.

- 자발적(Spontaneous) Task

대부분 실시간시스템에서 한 부분을 차지하는

task로서 외부환경에 대해서 확인된 이벤트나 조건들에 반응하여 예상치 못한 시기에 활성화되는 것은 산발적 task와 동일하다. 산발적 task와는 다르게 자발적 task가 수행시 빈번함에 대한 상한선이 없다. 그리고, 자발적 task는 수행시 어떤 리소스(자원)도 미리 할당이 되는 경우가 없다. 즉, 수행시마다 자발적 task는 시스템에 충분한 자원 여부를 확인한 후에 서비스에 대한 요청을 처리한다.

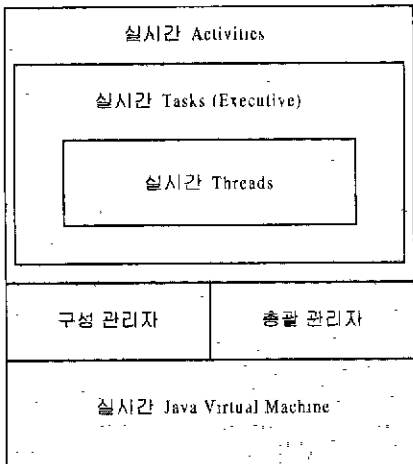
● 계속적(Ongoing) Task

실시간 activity의 마지막 task는 thread이다. 이러한 task들은 영원히 수행하지만 시간 경과의 비율에 따라 진행되어야만 한다. 예를 들면 주식시장에서 해당 주식이 최고의 시세일 때 거래의 기회를 알리는 경보 응용 프로그램이 이에 해당된다.

2.2 자비에서 PERC로의 실시간 확장

● 실시간 Activity

실시간 activity란 같은 목적으로 동작하는 태스크의 집합으로서 다수의 표준 메시지를 구현하는데 필요한 JAVA 객체를 의미한다. 이러한 메시지들로는 configure(), negotiate() 등이 있으며 (그림 2)와 같이 구성된다[12].



(그림 2) 실시간 Activity의 구성

● 구성관리자(Configuration Manager)

구성관리자는 JIT컴파일러를 통하여 자바 바이트 코드를 런타임시 네이티브 코드로 변환하는 기능을 수행하고 task의 실행시간을 계산하고, 동작에 필요한 메모리 요구사항을 파악한다. 구성관리자는 필요시 task사이의 블로킹을 분석 및 처리한다.

● 총괄관리자(Administrator)

총괄관리자는 각각의 실시간 activity에 필요한 리소스 요구량에 따라 실시간 task의 CPU시간과 메모리 사용량을 계산한다. 이것을 기초로 실행시간과 동적 메모리에 실시간 task와 협상을 진행한다.

● Atomic code Segments

JAVA에서의 synchronized와 유사한 기능을 갖는다. PERC에서 추가된 키워드로써 동일한 코드 세그먼트에 한순간 오직 한 스레드만이 접근이 가능하도록 지원한다. 즉, 코드블록이 마치 하나의 명령어처럼 동작한다.

● 실시간 Activity의 전개

(그림 3)에서는 실제로 개발부터 수행까지의 순서를 차례로 보여 주고 있다. 그림에서처럼 개발자는 표준 프로그래밍 에디터를 사용해서 소스 프로그램을 작성하고 컴파일해서 클래스파일들을 얻는다. 그렇게 얻어진 클래스 파일들을 선행로딩(Preloading) 및 로딩(loading)하여 메모리에 activity들이 활성화되도록 한다. 그후 구성관리자, 총괄관리자에서 activity의 우선순위(priority)를 결정한다. 그런 후 개별적인 task들의 수행이 이루어진다.

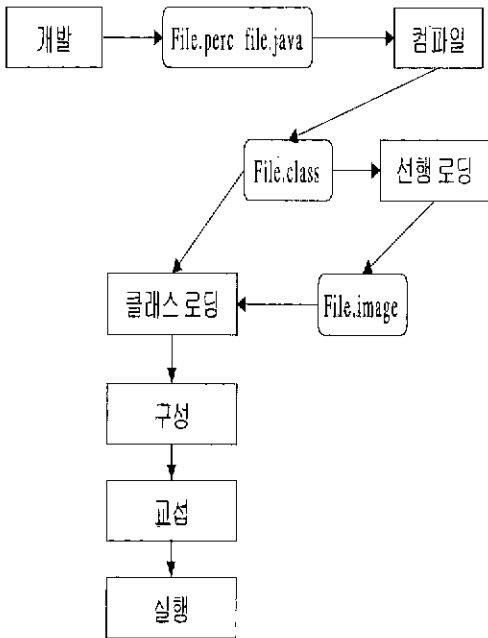
● 실시간 가비지 콜렉션

실시간 가비지 콜렉션은 응용프로그램이 메모리 할당을 요구시 실시간 제약성과 함께 task의 안정성을 보장하며 할당이 가능하도록 메모리를 확인하는 것이다.

● 동적 메모리의 실시간 관리

이상적으로 실시간 프로그래머는 실시간 activity

가 요청시 요청된 크기만큼 동적메모리의 할당이 가능하다고 가정할 것이다. 그렇지만 일반적으로는 가능하지 않을 때도 많다. 앞에서 언급했듯이 구성관리자가 이것을 담당하게 된다.

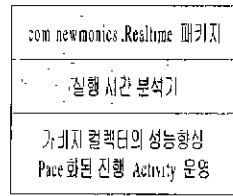


(그림 3) PERC 실시간 Activity의 전개

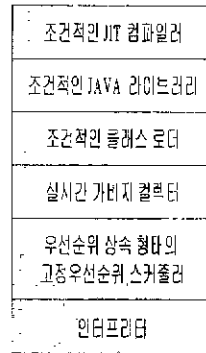
3. 실시간 JAVA PERC의 구현

PERC의 실시간 확장을 구현함에 있어 다양한 방법을 고려해 볼 수 있다. 가장 단순한 방법은 JVM을 다운로드 받아서 변형된 JVM을 개발하는 것이다. 그리고, 추가적인 API 수준에서의 지원도 고려해 볼 수 있다.

(그림 4)는 PERC에서 구현된 가상머신과 실시간 API확장의 내용을 보여 준다. 아래에서는 이렇게 구현된 PERC의 구현원리 및 방법에 대해서 중요하게 고려해야 할 부분만 간략하게 소개하기로 한다.



PERC의 실시간 API 확장



PERC의 가상 머신

(그림 4) PERC 실시간 확장 API와 PERC 가상 머신

● 자바 가상 머신

자바 가상 머신의 구현에 있어 도메인에 따라 다른 점들을 고려할 필요가 있다.

먼저, 데스크탑 컴퓨팅 환경에서의 가장 중요한 특징은 다른 프로세스들간의 간섭이 최소화되어야 한다. 그래서 이 경우 실시간 task의 수행계획이 가장 중요한 의미를 갖는다. 다음으로, 내장형 자바 시스템에서는 그러한 간섭현상이 제거된다. 반면, 이 경우에는 시스템 리소스들을 충분히 활용하여 결과물이 많이 생기도록 고안해야 한다. 그리고, 내장형 강 실시간 PERC 환경에서는 시간 제약적인 요소를 우선 고려해야 한다. 마지막으로 지연 한도를 긴밀하게 얻어 내기 위해서는 반드시 하드웨어적으로 가비지 컬렉션을 구현할 필요가 있다.

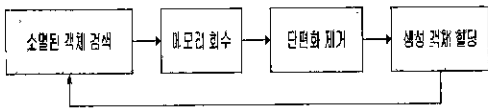
이렇게 도메인에 따라 자바가상머신을 구현시 주된 내용을 조금씩 다르게 고려할 수 있다.

● 스케줄링 기법

현재 실시간 시스템의 구현시 가장 인기있는 스케줄링의 기법 두 가지는 static cyclic schedule 기법과 Rate monotonic scheduling 기법이 있다. 전자는 수행시간이 정적으로 판단이 될 경우 주로 쓰여지게 되고 후자는 큐(Queue)를 이용해서 동적으로 우선순위를 결정하는 방법이다. 주로 이 두가지의 혼용형태로 스케줄러를 구현한다.

● 실시간 가비지 컬렉터

이상적인 가비지 콜렉터는 객체가 소멸할 때 바로 메모리 회수와 동시에 단편화 제거가 이루어지는 것이지만 실제로는 가비지 콜렉터가 먼저 소멸된 객체를 인식하는 작업이 진행된 후에 단편화 제거가 수행될 수 있다. 논리적으로는 할당 가능한 메모리가 존재함에도 불구하고 메모리가 회수되지 않거나 전체 할당 가능한 메모리의 합이 할당할 메모리 양보다 크지만 단편화가 제거되지 않았기 때문에 할당할 수 없는 경우가 발생하게 된다. 즉, 문제는 메모리 회수와 단편화 제거가 메모리 할당이 요구되는 객체 생성 이전에 완료될 수 있도록 하는가에 초점이 맞추어져 있다. 참고로 그림 5에서는 가비지 컬렉터의 동작 순서를 보여 준다[12].



(그림 5) 가비지 컬렉터의 동작 순서

4. 결 론

본 논문에서는 실시간 자바의 연구동향으로서 Newmonics사의 PERC에 대해서 살펴 보았다. 내장 시스템을 위한 자바 가상 머신을 생성하고, 실시간 제약조건들을 API 수준에서 포함하여 확장하고 있는 점들을 강조하였다. 이와 같이 실시간

자바에 대한 연구는 자바의 가장 큰 장점인 이식성과 단순성을 유지하면서 시간적 제약조건을 만족시키기 위한 기능을 확장하는 방향으로 나아가고 있는 실정이다.

현재까지는 실시간 자바의 응용 및 성공 사례가 많이 부족한 편이다. 그렇지만 상당히 많이 연구되는 있고 또한 많이 실시간 시스템에서 그 가능성 및 발전성을 인정 받고 있는 실정이다.

참고문헌

- [1] The Java Language Environment: A white Paper. Sun Microsystems, Inc. 1995.
- [2] Ritchey, T., Java! 1995, New Riders Publishing: Indianapolis, Indiana.
- [3] Wilner, D., Personal conversation at ACM Programming Languages Design and Implementation Conference, July, 1995.
- [4] Nilsen, K., Real-Time No Longer a Small Specialized Niche, Fifth Workshop on Hot Topics in Operating Systems (HotOS-V), 1995. Orcas Island, Washington: IEEE Computer Society Press, p.2-6.
- [5] Hoare, C.A.R., Communicating Sequential Processes. Communications of the ACM, 1978. 8(8): p.666-677
- [6] Sha, L, R. Rajkumar, and J.P. Lechoczky, Priority Inheritance Protocols: An Approach to RealTime Synchronization, IEEE Transactions on Computers, 1990, 39(9): p.1175-1185.
- [7] Amold. K. and J. Gosling, Native Methods, in The JAVA Programming Language, L Friendly, Editor. 1996, Addison-Wesley Publishing Company, Inc.: Reading, Massachusetts. p.333

[8] Liu, C.L. and J.W. Layland, Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of the ACM, 1973, 20(1): p44-61

[9] Nilsen, K. Progress in Hardware-Assisted Real-Time Garbage Collection. in Lecture Notes in Computer Science #986. 1995. Kinross, Scotland: SpringerVerlag.

[10] Nilsen, K. Reliable Real-Time Garbage Collection of C++. Computing Systems, 1994, 7(4): p.467-504

[11] Nilsen, K. Rygg, B. Worst-Case Execution Time Analysis on Modern Processors. in AGM SIGPLAN Workshop on Languages, Compilers, and Tools for Real-Time Systems, 1995. San Diego, California, 30(11): ACM SIGPLAN Notices: p. 20-30

[12] 경희대학교 김형일, 이승룡, 배수강. 실시간 자바(Real-Time Java), 1998. 4. 정보과학회지 제16권 제4호.



박 남 섭

1997년 부산외국어대학교 컴퓨터 공학과 (학사)
 1997년-현재 부산외국어대학교 컴퓨터공학과 석사과정
 관심분야 : 분산객체기술(CORBA), 프로그래밍 언어



이 정 배

1981년 경북대학교 전자공학과 전산전공 (학사)
 1983년 경북대학교 전자공학과 전산전공 (석사)
 1995년 한양대학교 전자공학과 (박사)

1982년-1991년 한국전자통신연구원 선임연구원
 1996년-1997년 U.C.Irvine 객원교수
 1991년-현재 부산외국어대학교 부교수
 관심분야 : 컴퓨터 네트워크, 실시간 자바, 실시간 프로토콜, 인터넷 응용



이 승 호

1977년 덕수상업고등학교
 1977년-1984년 한국전자기술연구소
 1984년-1990년 (주)한국에이아이소 프트 차장
 1990년-1992년 (주)CNI 개발담당이사

1992년-현재 (주)한국이지시스템 대표이사
 관심분야 : 인터넷/인트라넷 기술, 자바, CTI(Computer Telephony Integration), NT 서버 기술