실시간 정보처리 시스템 구축 및 성능 검증

정 윤 석[†]·김 인 수[†]·김 태 완[††]·김 기 천[†††]·장 천 현[†††]

요 약

시스템 처리용량의 증가로 과거 메인프레임이 처리하던 작업을 PC가 처리할 수 있게 되었다. 특히 산업계에서는 기존의 메인프레임을 PC급으로 전환하면서 PC급 시스템에서 실시간성을 보장하는 실시간 정보처리와 실시간 정보감시를 위한 실시간 모니터링 기능이 요구되고 있다. 그러나 실시간성의 보장을 위해서는 실시간 기능의 추가 및 이에 대한 검증이 필요하다. 본 논문은 이러한 실시간성의 요구를 충족시키기 위해 실시간성을 제공할 수 있는 시스템의 구축과 이에 대한 성능 검증을 목적으로 한다. 또한 실시간 정보감시를 위해 웹기반의 실시간 모니터링 시스템을 구축하는 것을 목적으로 한다.

The Implementation and Performance Testing of Real Time Information Processing System

YoonSeok Jeong[†]·Insu Kim[†]·TaeWan Kim[††]·
Keecheon Kim[†††]·ChunHyon Chang[†††]

ABSTRACT

In recent days, pc can process many things that mainframe have processed. Especially, in the transition from mainframe to pc, industries have needed the capabilities of Real Time information processing and Real Time monitoring for information control. But to support Real Time properties, it's needed to add Real Time modules and verify them. In this dissertation, we implemented and verified the Real Time System to support Real Time property. In addition, we implemented Web-based Real Time Monitoring System for Real Time information monitoring.

키워드 : 실시간성(Real Time), 실시간 시스템(Real Time System), 실시간 운영체제(Real Time OS), 실시간 데이터베이스(Real Time Database), 웹 기반 모니터링(Web-based Monitoring)

1. 서 론

시스템 처리용량의 증가로 과거 메인프레임이 처리하던 작업을 PC가 처리 할 수 있게 되었다. 특히 산업계에서는 기존의 메인프레임을 PC급으로 전환하면서 PC급 시스템에서 실시간성을 보장하는 실시간 정보처리와 실시간 정보감시를 위한 실시간 모니터링 기능이 요구되고 있다. 그러나 실시간성의 보장을 위해서는 실시간 기능의 추가 및 이에 대한 검증이 필요하다.

본 논문은 이러한 실시간성의 요구를 충족시키기 위해 실시간성을 제공할 수 있는 리눅스 기반의 시스템의 구축과 이에 대한 성능 검증 및 리눅스 기반의 실시간 시스템이 제공할 수 있는 deadline 설정을 목적으로 한다. 또한 실시간 정보감시를 위해 웹기반의 실시간 모니터링 시스템을 구축하는 것으로 목적으로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 실시간과 실시간 시스템의 개념, 웹모니터링에 대해서 살펴보며 3장은 설계 및 구현 부분으로 실시간 시스템의 구성과 웹기반 모니터링을 위한 설계내용을 살펴본다. 4장에서는 실험 부분으로 실시간 데이터베이스의 실험을 통해 실시간성 검증내용을 다룬다. 마지막으로 5장 결론에서는 본 논문의 성과 및 향후 연구방향에 대해서 기술한다.

2. 관련 연구

2.1 실시간의 개념

일반적으로는 '특정 이벤트에 대해 즉시 응답하는 것'을 의미하며 여기에 시간 제약(timing constraints)이라는 개념을 추가하면 '특정 이벤트를 제한 시간 내에 처리하는 것'을 실시간(realtime)이라고 한다[1].

실시간은 시간 제약의 중요성에 따라 크게 하드 실시간(hard realtime)과 소프트 실시간(soft realtime)으로 분류한다.

† 준 회 원 : 건국대학교 대학원 컴퓨터공학과
†† 정 회 원 : 건국대학교 대학원 컴퓨터공학과
††† 종신회원 : 건국대학교 컴퓨터공학과 교수
논문접수 : 2001년 9월 27일, 심사완료 : 2001년 12월 28일

하드 실시간은 태스크가 정확하게 실행될 뿐 아니라, 정확한 시간에서 실행되어야 하며 시간 제약 위반 시 치명적 결과를 야기하는 경우를 말한다. 반면 소프트 실시간은 좀 더 완화된 경우로서, 가능한 빨리 실행되지만, 어떤 정해진 시간 내에 종료할 필요는 없는 경우이다.

2.2 실시간 시스템

실시간 시스템은 '시스템의 수행 결과가 기능적으로 정확하고, 결과 도출 시간이 주어진 제약 조건을 만족시키는 시스템'이다. 실시간성을 만족하는데 있어 중요한 요소는 인터럽트가 발생했을 때 짧은 시간 안에 인터럽트 핸들러를 호출하는 것과 우선순위가 높은 이벤트를 먼저 처리하도록 하는 것이다[1, 2]. 이를 위해 priority-based preemptive scheduling라는 스케줄링 방식을 이용한다.

2.3 실시간 운영체제

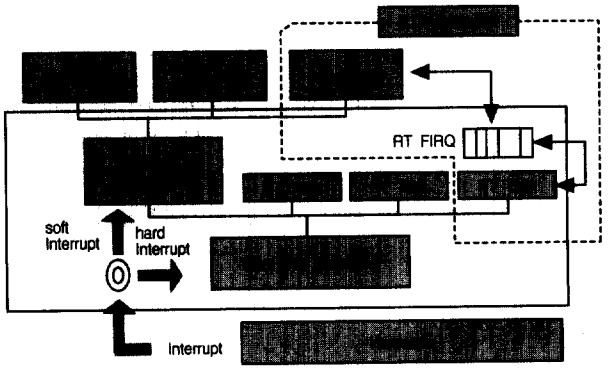
2.3.1 실시간 운영체제의 특징

다음은 실시간 운영체제가 가져야 할 몇 가지 특징이다[2].

첫째, 다중쓰레드를 지원 및 선점가능
둘째, 쓰레드간의 우선 순위를 보장
셋째, 쓰레드간의 동기화를 지원
넷째, 운영체제의 행동의 명확성

2.3.2 리눅스 기반의 실시간 리눅스

현재 대표적으로 사용되는 리눅스 기반의 공개 실시간 운영체제에는 RT-Linux[3], RTAI(Real-Time Application Interface)[4], KURT(The KU Real-Time Linux)[5], RED-Linux(Real-time and Embedded Linux)[6]등이 있다.



(그림 1) RT-Linux 구조

본 논문에서 사용한 RT-Linux[3]는 리눅스에 hard real time 기능을 부여하기 위하여 리눅스 커널을 RT-Linux 커널에서 돌아가는 하나의 프로세스로 만들어서 real time task와 기존의 리눅스 프로세스가 공존하는 형태로 만들어졌다. RT-Linux 커널이 실시간 프로세스를 생성, 스케줄링하며, 인터럽트 발생 시에 처리를 담당한다. 기존 리눅스 커널은 그대로 일반 프로세스를 관리하고, 자신의 인터럽트를 처리한다.

2.4 실시간 데이터베이스

실시간 데이터베이스는 데이터의 일관성 기준을 준수하면서 실시간 응답시간을 제공할 수 있는 시스템을 말한다[2]. 본 논문에서 선택한 RTDB인 Symphony-RTDB는 메모리 내장형 데이터베이스이다[7]. 실시간 어플리케이션을 지원하기 위하여 주 기억 장치(MAIN MEMORY)에 데이터베이스 스페이스를 구축하고 데이터베이스 Query에 대한 평균 응답시간을 시스템이 제공하는 수준까지 제시한다.

2.5 웹모니터링

웹모니터링은 기존의 터미널을 통한 모니터링과 달리 인터넷(TCP/IP)을 기반으로 웹상에서 실시간 정보를 모니터링한다. 관련분야는 디스플레이를 위한 자바애플릿 기술과, 데이터 취합을 위한 푸쉬기술이다.

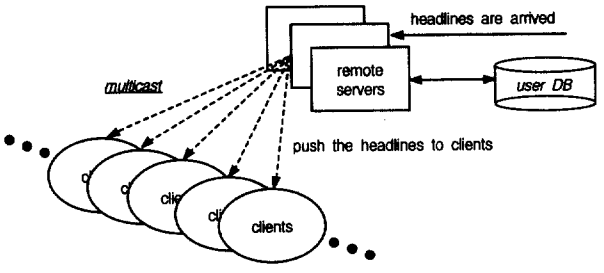
2.5.1 자바 애플릿

자바 애플릿[8]은 웹브라우저 내부에 있던 기존 특성을 활용함으로서, 최소한의 코드로 풍부한 특성을 표현할 수 있게 해준다. 애플릿은 그 크기가 적으며 다운로드 속도를 높이고 프로그래밍을 단순화하기 위해 브라우저의 특성을 이용한다.

2.5.2 푸쉬(Push)

푸쉬는 사용자 PC에 백그라운드에서 스스로 새로운 정보를 서버로부터 가져와 하드디스크에 저장하는 방법으로, 사용자가 원할 때 바로 하드디스크에 저장된 정보를 보여주므로 속도가 매우 빠르고, 사용자의 작업중을 방해하지 않는다는 특징이 있다.

푸쉬 클라이언트는 수동적으로 들어오는 Data를 받아들이며 반대로 서버는 자발적으로 들어온 컨텐츠들을 바로 클라이언트들에 푸쉬한다.

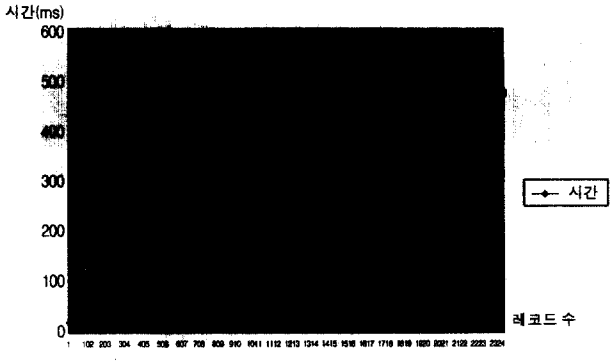


(그림 2) 푸시기술

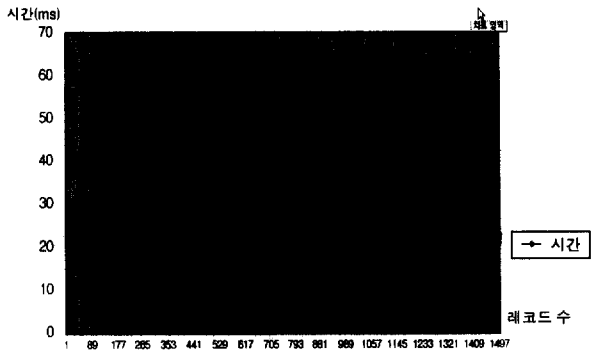
2.6 비실시간 시스템 연구사항

다음은 기존의 수행되었던 비실시간 시스템 상의 트랜잭션 처리 결과이다. 비실시간 시스템과 실시간 시스템간의 OS 및 데이터베이스의 튜닝 및 처리되는 자료의 특성 등에 따라

실험결과가 달라질 수 있기 때문에 두 시스템 간의 단순비교는 어렵지만 기본적으로 실시간 시스템의 결과를 비교 판단해 볼 수 있다.



(그림 3) Access 데이터베이스처리시간

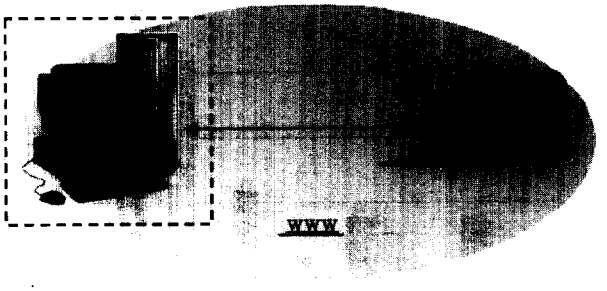


(그림 4) MS-SQL 데이터베이스 처리시간

3. 설계 및 구현

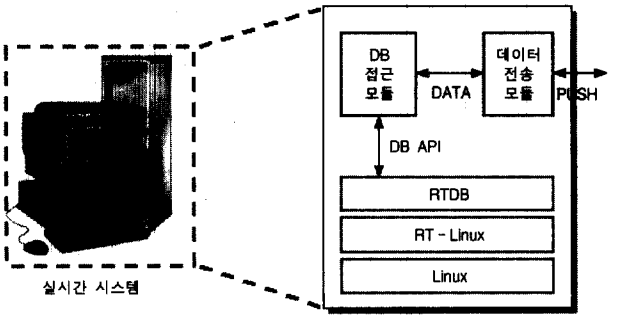
실시간 운영체제인 RT-Linux와 실시간 데이터베이스 Symphony-RTDB를 이용해서 실 시간성 검증 및 향후 실시간 제공과 관련된 응용을 위한 시스템을 구성하였다.

3.1 실시간 시스템의 구성



(그림 5) 전체 구성도

전체시스템의 가장 특징적인 것은 서버와 클라이언트간의 인터넷을 통한 연결이다. 즉 서버와 클라이언트의 위치에 관계없이 인터넷이 연결되어 있는 곳이라며 어디서든지 실시간 시스템을 이용할 수 있다. 따라서 로컬 뿐 아니라 원격지에서도 데이터베이스에 저장되어 있는 정보나 현재 입력되는 정보 및 데이터 상황들을 실시간적으로 확인할 수 있게 된다.



(그림 6) 실시간 시스템 내부 구성도

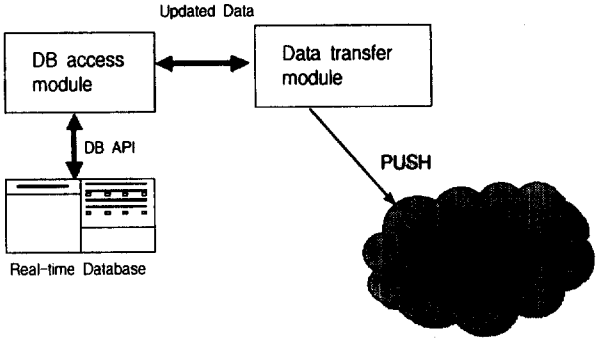
실시간 시스템은 시스템의 하부구조와 모듈로 크게 나눌 수 있다. 하부구조는 리눅스가 설치되어 일반적인 작업들을 처리한다. 그 위에 RT-Linux가 올려 실시간성을 확보한다. 그 위에 실시간 데이터베이스가 올라가서 작동을 하게 된다. 이로써 기본적인 실시간 시스템의 하부구조가 형성된다.

모듈은 크게 DB 접근 모듈과 데이터 전송 모듈로 나뉜다. DB 접근 모듈은 실시간 데이터베이스에 대한 인터페이스의 역할을 하며 데이터 전송 모듈은 DB 접근 모듈을 통해 얻게 된 데이터 WWW(인터넷)으로 전송하는 역할을 한다.

3.2 실시간 환경의 웹기반 모니터링 시스템

3.2.1 모니터링 서버

모니터링 서버에는 웹서버를 실행하며, 웹브라우저를 사용하여 접속하는 클라이언트들을 관리한다. 모니터링 서버는 모니터링할 자료를 수집하고 저장하는 데이터베이스, 갱신된 자료를 추출하여 모니터링 쓰레드로 전달하는 데이터베이스 액세스 쓰레드, 서버측에서 갱신된 자료를 클라이언트로 뿌려주는 모니터링 쓰레드로 구성된다.

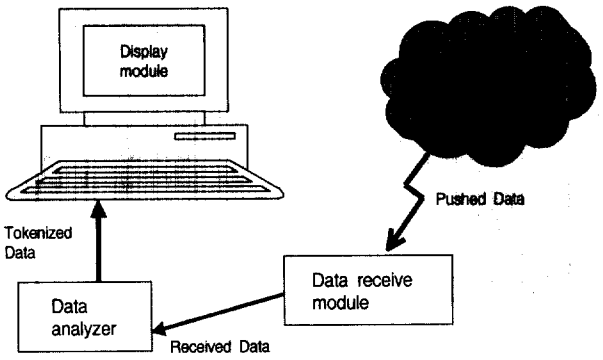


(그림 7) 모니터링 서버

3.2.2 모니터링 클라이언트

모니터링 클라이언트는 웹 상에서 동작하는 자바 애플릿으

로 구현되었다.



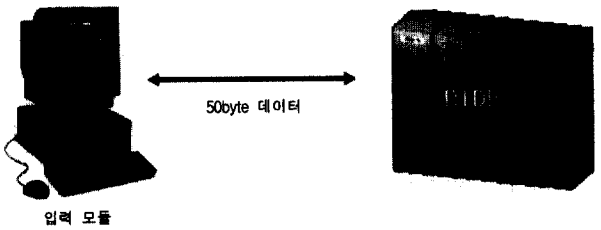
(그림 8) 모니터링 클라이언트

4. 실 험

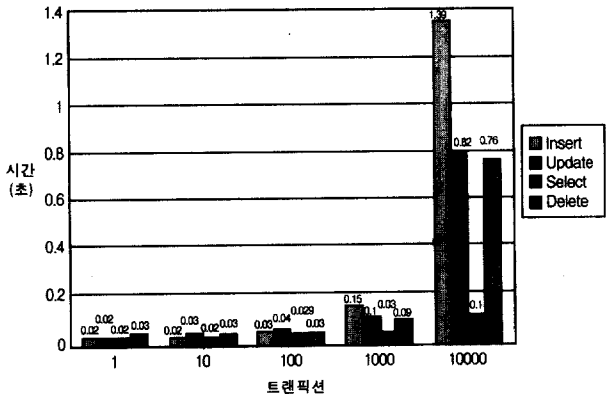
4.1 실시간성 실험

4.1.1 트랜잭션 증가에 따른 평균응답시간 체크

본 실험은 DB를 Memory에 적재하고 트랜잭션을 10건, 100건, 1000건, 10000건으로 증가시키며 평균응답시간을 체크하였다.



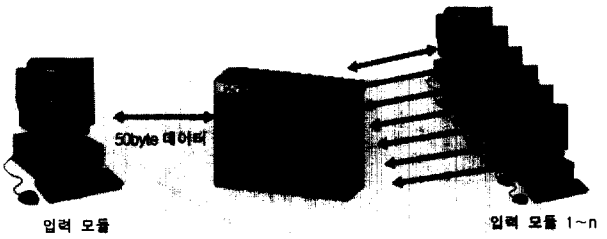
(그림 9) 테스트 1 구성도



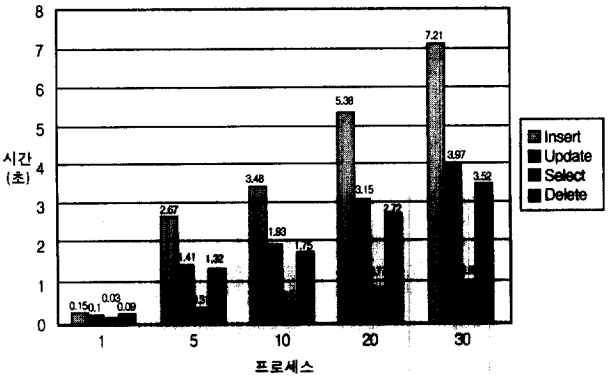
(그림 10) 테스트 1 결과표 : 왼쪽 막대부터 insert, update, select, delete

4.1.2 프로세스 증가에 따른 평균응답시간 체크

본 실험은 DB를 Memory에 적재 후 프로세스의 증가가 평균응답시간에 미치는 영향을 알아보기 위한 실험이다. 트랜잭션 수 10000건으로 고정하고 프로세스의 수 5, 10, 20, 30으로 증가시켰다. 입력 모듈 1-n은 DBMS에 접근하여 select 작업을 처리하는 모듈(프로세스)들이다.

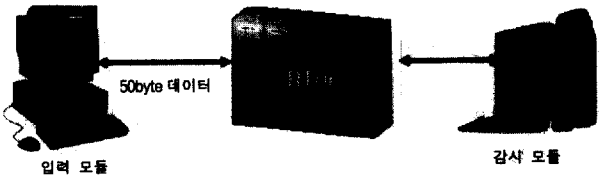


(그림 11) 테스트 2 구성도

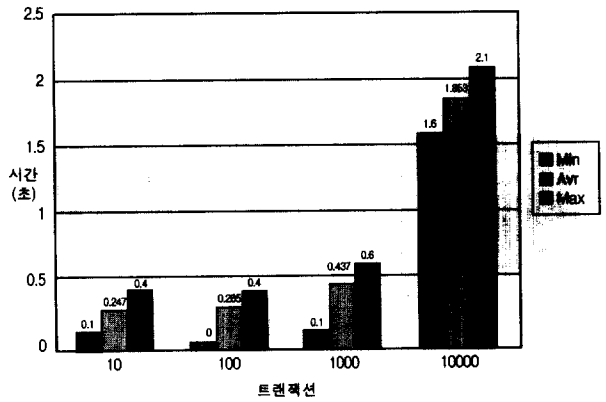


(그림 12) 테스트 2 결과표 : 왼쪽 막대부터 insert, update, select, delete

4.1.3 트랜잭션 증가에 따른 평균응답시간 체크(감시모듈 삽입)



(그림 13) 테스트 3 구성도



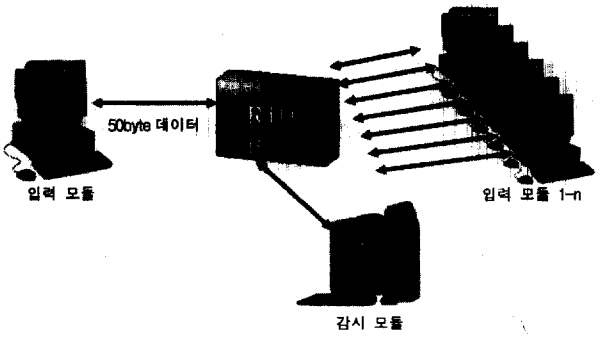
(그림 14) 테스트 3 결과표 : 왼쪽 막대부터 Min, Average, Max

본 실험은 물리적인 DB사용하여 데이터를 입출력하며 트랜잭션의 수를 10건, 100건, 1000건, 10000건으로 증가시키

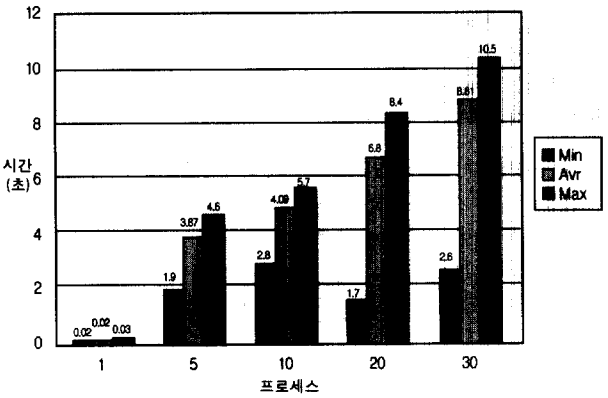
며, 물리적 DB접근 시 평균응답시간을 체크하였다. 이 때 사용한 감시모듈은 독립적인 프로세스로 동작하며 입력되는 데이터를 감시한다.

4.1.4 프로세스 수 증가에 따른 평균응답시간 체크(감시 모듈삽입)

본 실험은 물리적인 DB사용하여 데이터를 입출력하며 트랜잭션 수 10000건으로 고정하여 실험하였다. 프로세스의 수는 5, 10, 20, 30으로 증가시켰으며 입력 감시모듈을 통해 데이터의 입력을 감시하고 응답시간을 체크하였다.

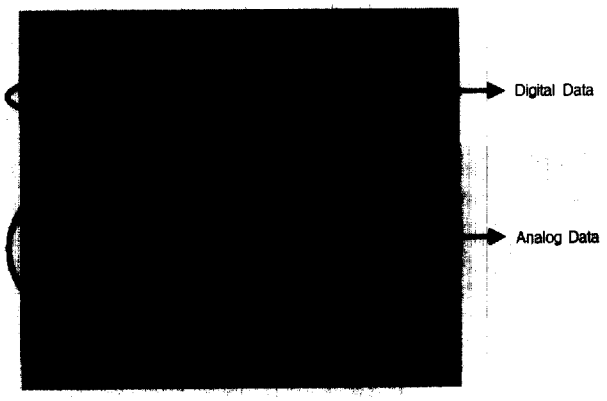


(그림 15) 테스트 4 구성도



(그림 16) 테스트 4 결과표 : 왼쪽 막대부터 Min, Average, Max

4.2 웹 모니터링 실험

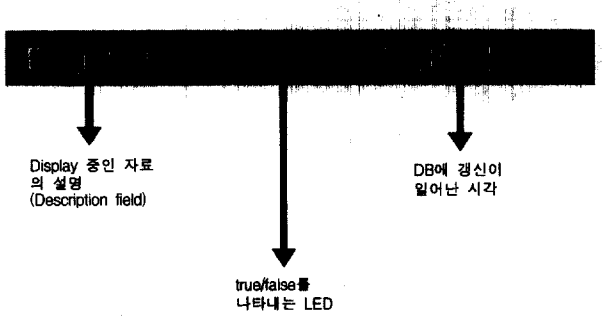


(그림 17) 모니터링 클라이언트 디스플레이

상기의 실험 3을 기초로 데이터를 입력하며 입력된 데이터를 모니터링 서버에 의해 푸쉬한 후 모니터링 클라이언트에 의해 디스플레이되는 결과를 웹 모니터링 실험을 실시하였다.

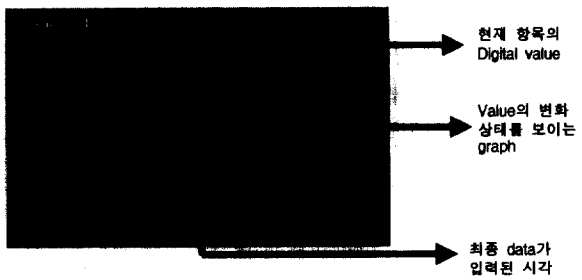
4.2.1 디지털 데이터

디지털 형태의 모니터링 데이터를 디스플레이하는 컴포넌트이며 한 화면에 최대 30개까지의 동시에 모니터링할 수 있다.



(그림 18) 디지털 데이터

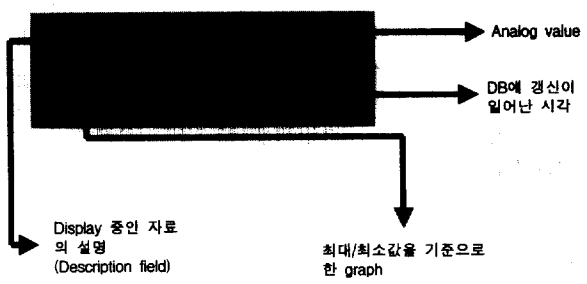
다음은 디지털 데이터의 디테일 모니터링 화면으로 특정 시간동안의 데이터의 변화를 모니터링 할 수 있다. 화면상의 그래프는 디지털 데이터의 특성을 보여준다.



(그림 19) 디지털 데이터 디테일

4.2.2 아날로그 데이터

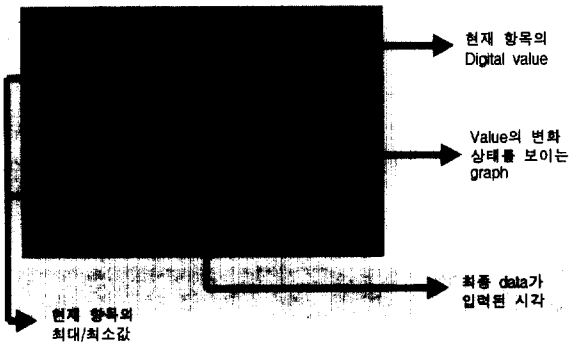
아날로그 형태의 데이터를 디스플레이하며 최대 60개까지 동시에 모니터링할 수 있다. bar 형식의 그래프를 사용하여 최대/최소 값의 범위내에서 그려진다.



(그림 20) 아날로그 데이터

특정 시간 동안의 데이터의 변화를 모니터링도 가능하다.

시간에 따른 변화를 나타내기 위해 꺾은선 그래프를 사용한다.



(그림 21) 아날로그 데이터 디테일

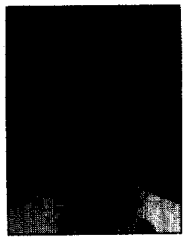
5. 결 론

본 논문은 실시간 운영체제와 실시간 데이터베이스를 이용해 실시간 시스템을 구축하였고 실시간성 검증을 위한 검증 실험을 하였다. 이를 통해 리눅스 기반의 실시간 시스템이 제공할 수 있는 트랜잭션에 대한 deadline의 기준을 마련하였다. 또한 실시간 정보감시를 위해 웹을 통해 모니터링할 수 있는 실시간 웹 모니터링 시스템을 구현하였다.

이 연구를 기초로 향후 실시간 운영체제의 실시간성에 대한 보다 다양한 검증실험이 요구되며 또한 웹 모니터링에 대해서 실시간 모니터링에 영향을 줄 수 있는 변수를 추출하고 이들의 변화에 따른 웹 모니터링의 영향 정도를 측정하는 검증 실험도 요구된다.

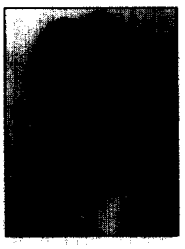
참 고 문 헌

[1] Nimal Nissanke, "Realtime Systems," Prentice Hall, 1997.
[2] Sang H. Son, "Advanced In Real-Time Systems," Prentice Hall, 1995.
[3] RTlinux 홈페이지 http : //www.rtlinux.org.
[4] RTAI 홈페이지 http : //www.aero.polimi.it/projects/rtai/.
[5] KURT 홈페이지 http : //www.ittc.ukans.edu/kurt/.
[6] Linux Kernel compile guide http : //kldp.org.
[7] 리얼시스텍, Symphony RTDB manual, 리얼시스텍, 2001.
[8] Laura Lemay and Charles L. Perkins, "teach yourself Java in 21 days," sams.net, 1996.



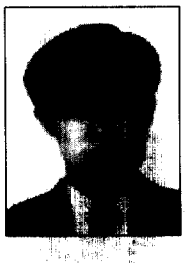
정 윤 석

e-mail : ysjeong@cse.konkuk.ac.kr
2000년 건국대학교 컴퓨터공학과(석사)
2000년~현재 건국대학교 컴퓨터공학과 박사과정
관심분야 : 실시간 시스템, 객체지향 프로그래밍, 시스템 모니터링, XML



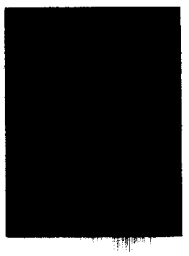
김 인 수

e-mail : darkguy@kkucc.konkuk.ac.kr
2000년 건국대학교 컴퓨터공학과(학사)
2000년~현재 건국대학교 컴퓨터공학과 석사과정
관심분야 : 이동인터넷, 이동단말기, 미들웨어



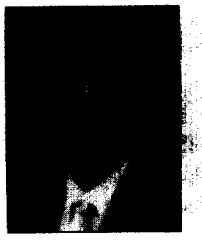
김 태 완

e-mail : twkim@konkuk.ac.kr
1994년 건국대학교 전자계산학과(학사)
1996년 건국대학교 전자계산학과(석사)
1996년~2001년 현대중공업 기전연구소 연구원
1996년~현재 건국대학교 컴퓨터공학과 박사과정 재학중
관심분야 : 프로그래밍 언어, 가상현실, 실시간 프로그래밍, 자동화 소프트웨어, 산업기기 감시 진단 제어 시스템



김 기 천

e-mail : kckim@kkucc.konkuk.ac.kr
1988년 서울대학교 계산통계학과(학사)
1992년 미국 노스웨스턴대 컴퓨터공학과 (박사)
1992년~1996년 한국통신기술(주) 연구소 선임연구원
1996년~1998년 신세기통신(주) 기술연구소 책임연구원
1998년~현재 건국대학교 컴퓨터공학과 조교수
1999년~현재 정보처리학회 편집위원
1993년~현재 정보통신응용연구회 연구위원
1999년~현재 개방형 컴퓨터시스템연구회(OSIA) 기술위원(표준전문가)
2000년~현재 개방형 컴퓨터시스템연구회(OSIA)지 편집위원
2000년~현재 개방형 컴퓨터시스템연구회(OSIA) Mobile IP WG 의장
2000년~현재 한국 인터넷정보센타(KRNIC) Name Committee 부위원장
관심분야 : 이동 인터넷, Mobile IP, IMT-2000 core network, 차세대 인터넷(IPv6와 이동성 차원), 메인 메모리 데이터베이스, 리눅스



장 천 현

e-mail : chchang@konkuk.ac.kr
1977년 서울대학교 계산통계(학사)
1979년 KAIST 전산학(석사)
1985년 KAIST 전산학(박사)
현재 건국대학교 컴퓨터 공학과 정교수
관심분야 : 프로그래밍 언어, 컴파일러, 실시간 처리