지능형 로봇 운영체제로서의 실시간 임베디드 리눅스 적용 방법

Application of Real-time embedded linux as an operating system for intelligence robots

최병욱, 박정호, 이수영
Byoung-Wook Choi, Jeong-Ho Park, Soo-Yeong Yi

Abstract - Currently many sensors and processing data in a robot based on USN environments need to real-time features. In this paper, we examine recent research trends on real-time operating systems, especially on real-time embedded Linux, RTAI and Xenomai, for intelligent robots. Xenomai is a real-time development framework and have special feature supporting RTAI, VxWorks, pSOS+ etc. through the "skin". This research gives a guide to researcher in using real-time embedded Linux in the sense of architecture, supporting real-time mechanisms, kinds of real-time device driver, performances.

Key Words : Real-Time Operating System, Xenomai, Real-Time Embedded Linux

1. 서론

최근 정보기기 및 제어기기와 같은 임베디드 시스템(embedded system)이 소형화, 경량화 되면서도 고성능의 다양한 기능을 제공하고 있다. 이러한 기기에는 주로 MPU(Micro Processing Unit), 메모리, DSP(Digital Signal Processor), 주변 장치 등이 하나의 칩으로 만들어진 SoC(System on Chip)가 사용되고 있다. SoC는 하나의 반도체 칩으로 만들어지기 때문에 소비 전력이 적고, 검증된 IP(Intellectual Property)를 사용하여 개발되므로 신뢰성이 높으며, 개발비용이 절감되는 이점이 있다. 그러나 SoC는 다양한 기능이 하나의 칩으로 구현되었기 때문에 개발에 많은 노력과 시간이 필요한 단점을 가지고 있다.[1] SoC의 단점을 보완할 수 있는 방법으로 임베디드 운영체제가 적용되고 있다. 임베디드 운영체제는 높은 신뢰성과 강력한 기능을 가지고 다양한 하드웨어를 지원함으로써 최근 SoC 기술과 더불어 주요한 기술로 부각되고 있다[1][2][3][4][7].

특히 지능형 로봇에서는 그동안 주행 알고리즘, 환경인식, 장애물 회피 등의 연구가 주로 이루어져 왔다. 이러한 복잡한 알고리즘과 데이터들을 처리하기 위해서 OS를 사용하여 시스템을 관리해왔다. 기존의 운영체제는 윈도우 또는 리눅스가 주류를 이루고 있으며, 이러한 운영체제는 실시간성을 보장하지 못함으로서 예상하지 못한 행위를 제어하지 못하는 경우가 발생한다. 지능형 로봇은 인간과 유기적으로 관계하기 때문에 이러한 예상치 못한 상황에 처하게 되면 인간에게 심각한 위협을 줄 수도 있다. 이러한 단점을 보완하기 위해서 현재는 실시간 운영체제를 적용하는 추세이다. 실시간 운영체제는 주로 군사용에 사용되어왔다. 상용 실시간 운영체제는 운영체제로의 기술 종속될 수 있으며, 초기 개발비용이 많이 들고 개발의 유연성이 떨어지는 단점이 있다.[5][6][7]

실시간 리눅스는 종전의 범용 리눅스 커널을 패치 하는 형태로 수정하여 Hard Real-Time 기능을 가진 RTOS(Real-Time Operating System)로 사용할 수 있게 해주는 리눅스 커널의 한 갈래이다. 실시간 리눅스에는 크게 New Mexico Institute of Technology 연구소의 Victor Yodailken 교수가 주도하는 RT-LINUX 프로젝트[14], Diapm 연구소의 Paolo Mantegazza가 주도하는 RTAI 프로젝트[11], Phillippe Gerum이 주도하는 XENOMAI 프로젝트[13]가 있다. 본 논문은 이러한 RTOS중에 XENOMAI에 대한 적용방법을 논의한다.

2. RTOS의 적용 사례

본 장에서는 현재 적용된 RTOS의 사례를 기술한다.

1.1 이동 로봇

[7]에서는 RTAI를 이용한 실시간 임베디드 리눅스가 구축된 이동로봇을 구현한 바 있다. 이동 로봇의 구동부와 통신을 위해 RT-COM 디바이스 드라이버를 구축하여 구동부의 제어를 실시간으로 처리하였으며, 실험을 통해 실시간 시리얼 드라이버와 비실시간 시리얼 드라이버를 비교함으로써 구동부 속도의 감/가속과 이동거리를 데이터로 비교분석 하였다.

1.2 로봇 제어용 리눅스 기반 실시간 커널

[5]에서는 RTAI를 이용하여 실시간 임베디드 리눅스를 구축하고 초음파 어레이 센서를 이용하여 실시간 커널을 사용하는 경우와 비 실시간 커널을 사용하는 경우를 구현하였다.

저자 소개
* 박정호 : 서울산업대학교 전기공학과 석사과정
** 최병욱: 서울산업대학교 전기공학과 부교수
*** 이수영: 서울산업대학교 전기공학과 부교수

3. 실시간 운영체제

실시간 시스템은 기존의 시스템과 달리 시스템의 동작이 논리적으로 완벽해야 하고, 또한 동작이 정해진 시간안에 정확히 처리되어야 하는 시스템이다.

실시간 운영체제란 실시간 시스템에서 정해잔 시간에 주어진 작업을 수행하도록 지원하는 소프트웨어이다. 이것은 운영체제가 무조건 빨리 작업을 수행한다는 것을 의미하지는 않는다. 주어진 시간에 예측 가능한 시스템 함수를 이용하여 원하는 출력을 제공하는 시스템을 말한다. 이러한 실시간 운영체제는 시간을 제어하여 다중 태스크(multi task)구조의 프로그램을 지원하고, 실시간 시스템을 구현하기 위해 태스크들 간의 통신, 동기화 그리고 스케줄링(scheduling)등의 메커니즘(mechanism)을 제공한다. 실시간 운영체제의 특징은

- Multitasking
- Time management
- Semaphore management
- ITC(Inter Task Communication)
- Synchronization

그러나 실시간 운영체제는 많은 장점에도 불구하고 사용하기 어렵고 고가이며, 또한 대부분의 경우 소스가 공개되지 않을 뿐만 아니라 프로세서를 변경할 경우 커널과 개발 환경을 재구성하여야 하는 단점을 가지고 있다.

3.1 RTAI

RTAI는 1996년 이탈리아의 DIAPM의 Paolo Mantegazza에 의해서 RTLinux를 기본 개념으로 개발되어 현재는 RTLinux와 별개로 공동 커뮤니티의 프로젝트로서 개발이 진행되고 있다. RTAI는 사실 실시간 운영체제가 아니라 실시간 태스크를 위한 인터페이스이다. 즉 RTAI를 사용하기 위해서는 운영체제가 필요하다. 주로 RTAI와 인터페이스 하는 운영체제로 리눅스를 사용하고 있다.

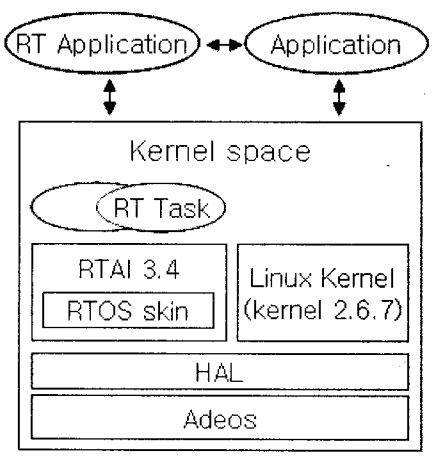


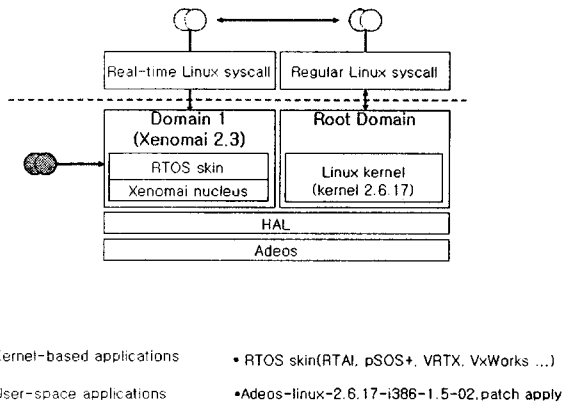
그림 1. RTAI 구조

그림 1은 RTAI의 Architecture이다. 리눅스 커널은 하나의 도메인(Domain)으로 취급된다. 지원 프로세서로는 i386, ppc, arm이 있지만, arm의 경우 pxa255 kernel-2.6.7 상태에서 머물러 있다. RTAI의 경우에는 패치가 i386에 한정되는 모습을 보이고 있다.

3.2 Xenomai

RTAI와 Xenomai는 2003년 RTAI/fusion 프로젝트명으로 통합된 기간이 있었다. 하지만 이 두 RTOS는 코어 레벨에서 많은 차이점을 갖고 있었다. 2005년에 RTAI로부터 독립하였다.

Xenomai는 여러 종류의 실시간 인터페이스(real-time interface)를 구축할 수 있는, 추상적인 RTOS 코어를 기반으로 하는 실시간 개발 프레임워크(Real-time development framework) 이다. "skins"라고 불리우는 RTOS들이 nucleus 상에서 구축된다. skins에는 RTAI, pSOS+, VRTX, VxWorks등이 있다. 그림 2는 Xenomai의 구조다.



<그림 2> XENOMAI 구조

XENOMAI native API 서비스

- Task management
  ·task scheduling, general management
- Timing services
  ·system timer management
- Synchronization
- Messaging and communication
  ·Intertask, Message queues, Memory heaps, Message pipes
- Device I/O Handling
- Registry support

Xenomai는 시리얼 통신을 위한 RT_COM, 이더넷 네트워킹을 위한 RT_NET, CAN(Controller Area Network)를 위한 RT_CAN등과 같은 실시간 디바이스 드라이버를 제공하며 RT_CAN의 가상 드라이버도 제공하고 있다.

4. Xenomai 설치

4.1 Xenmai-2.3 설치

Xenomai는 RTAI와 마찬가지로 실시간 시스템을 구현하기 위한 인터페이스이므로 운영체제가 필요하다. 본 논문에서는 Xenomai를 위한 운영체제로 리눅스를 사용하였다. 설치 환경은 타겟보드는 SBC(single board computer) P Mobile 1.6Ghz, Fedora Core4 kernel 2.6.11, gcc-4.0.0을 사용하였다. 리눅스 커널 2.6.17을 adeos-linux-2.6.17-i386-1.5-02.patch으로 패치를 하여 실시간 시스템을 구축하였다.

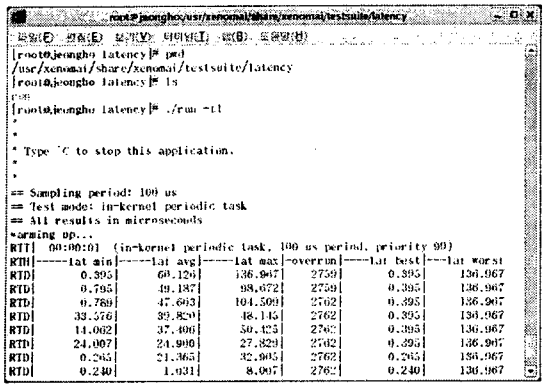


그림 3. Xenomai-2.3 latency test

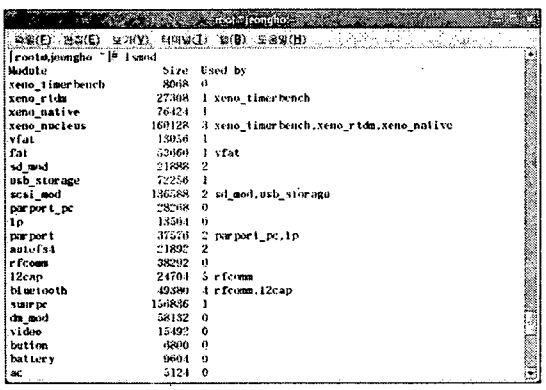


그림 4. 로드된 모듈

그림 3은 Xenomai-2.3에서 제공하는 latency test 프로그램을 커널 모드로 동작하는 화면이고, 그림 4는 이 테스트 프로그램이 동작할 때 lsmod란 명령어를 이용하여 로드된 모듈을 확인한 화면이다.

4.2 RTAI와 Xenomai 기능적 차이점

Xenomai와 RTAI를 비교해서 RTAI는 immediate IRQ dispatching 과 syscall veroting에서 Xenomai와 비교하였을 때 이점이 있다. 타겟을 P133MHz으로 해서 CVN에서 RTAI-3.4 그리고 SVN에서 Xenomai 2.2를 비교했을 때 RTAI가 worst-case latencies가 10% 더 낮음을 보였다. 이 차이점에 대해 Xenomai에서는 direct syscalls에 의한 것으로 보고있다. 그 이유로 gdb가 항상 LXRT 프로세스와 일치하지 않기 때문이다.

RTAI는 Adeos(Adaptive Domain Environment for Operating Systems)와의 상호 협력의 부재로 인하여 Adeos/I-pipe에 의해 만들어진 것을 따라가는 형상이 되고있다.[12]

5. 결론

센서의 종류가 많아짐에 따라 방대한 데이터양을 갖는 지능형 로봇의 운영체제에서 실시간성을 보장하기 위한 가장 좋은 방법은 실시간 임베디드 리눅스를 구축하는 것이다. 본 논문에서는 Xenomai와 RTAI의 특징을 기술하고 비교함으로써 최근 실시간 임베디드 리눅스의 연구 동향을 분석 하였다. Xenomai는 "skins"라는 개념을 통해서 pSOS+, Vxworks, RTAI등 상용 RTOS 및 RTAI를 에뮬레이트 할 수 있는 개발환경을 제공한다. 이러한 특징은 향후 실시간 임베디드 리눅스를 지능형 로봇에 적용함에 있어서 "skin"을 통해서 간단히 컴파일만으로 적용해 볼 수 있는 유용한 개발 환경으로 기대 할 수 있겠다.

참 고 문 헌

[1] 조덕연, 최병욱,"임베디드 리눅스를 이용한 산업용 인버터의 웹기반 원격관리,"제어 및 자동화 시스템 공학회, 제9권 4호, 340-346,2003.

[2] 최병욱, 신은철, 이수영, "임베디드 리눅스를 이용한 웹 기반 빌딩자동화 시스템," 제어 및 자동화 시스템 공학회, 제 10권 4호, 335-341, 2004

[3] 최병욱, " Embedded operating system and it's applications in development ofr networked robot," 한국로봇공학회, 제1회 워크샵, 2004

[4] 최병욱, "실시간 임베디드 리눅스", 인간기능 생활지원 지능로봇 기술개발 사업단 교재,

[5] 신주호, "리얼 타임 리눅스 기반 개방형 로봇 제어기", 학위 논문, 2002

[6] 노현창, "로봇 제어용 리눅스 기반 실시간 커널의 설계 및 구현", 학위논문, 2000

[7] 신은철, 최병욱, "실시간 임베디드 리눅스를 이용한 로봇 플랫폼 구현", 제어자동화시스템공학 논문지 제 12권, 2006

[8] Ismael Ripoll,"RTLinux versus RTAI", www.linuxdevices.com, 2002

[9] J. Kiszka, "The Real-Time Driver Model and First Applications", IEEE 2004

[10] Anders Nilsson, " Tailoring native compilation of Java for real-time systems", Doctoral Dissertation, 2006

[11] www.rtai.org

[12] www.mail-archive.com/xenomai-help

[13] www.xenomai.org

[14] www.rtlinux.com