

실시간 운영체제 연구 개발 동향

김 대 영, 도 운 미*, 김 주 홍, 조 혜 영, 성 종 우

한국정보통신대학교, University of Florida*

I. 서 론

고성능 마이크로프로세서와 유무선통신 기술의 발달, 그리고 소프트웨어기술의 발달에 따라 실시간 임베디드 시스템의 응용이 과거 산업용과 같이 제한된 분야에서 벗어나 정보가전, 휴대폰, PDA, 셋탑박스, 디지털 TV, 자동차 텔레메틱스, 항공기, 산업자동화 등 일상생활로 광범위하게 스며들고 있으며, 한층 더 복잡한 기능을 요구하고 있다. 일반적으로 이러한 실시간 임베디드 시스템은 제한된 메모리 용량이나 전력 소모 및 안전과 직결된 실시간성을 요구하기도 한다. 이러한 복잡 다양한 기능을 가진 실시간 임베디드 시스템의 요구에 따라 효율적인 자원관리와 프로 그래밍 인터페이스를 제공해주는 전용 운영체제는 필수적인 요소가 되었고, 현재 광범위하게 연구 및 사용이 되고 있다. 실시간 임베디드 운영체제의 연구 개발은 각 응용시스템에 적합한 다양한 종류의 특수 목적 운영체제를 개발하는 방향과 실시간 임베디드 리눅스와 같이 범용성을 가지며, 오픈소스를 활용하는 방향으로 크게 나누어진다.^[1]

본 논문에서는 실시간 운영체제를 광의의 개념으로 해석하여 임베디드 운영체제와 실시간 운영체제를 포함하는 것으로 정의한다. 협의의 임베디드 운영체제란 제한된 자원을 효율적으로 운용하고, 특히 메모리 사용을 최소화하기 위한 목적으로 개발된 운영체제로, 현재 인터넷 정보가전 제품이나 PDA 등에 많이 활용되고 있는 임베디드 리눅스가 이에 해당된다. 한편, 협의의 실시간

운영체제란 안전에 매우 민감한 항공기나 자동차, 그리고 핵발전소 등의 응용에 사용되는 운영체제로, 각 실시간 태스크의 실시간 타이밍 요구사항들을 만족시켜줄 수 있어야 한다.

본 논문은 실시간 임베디드 운영체제의 정의와 연구 동향, 특히 리눅스등 오픈소스와 특정 응용영역에서 연구 개발되고 있는 실시간 운영체제들을 소개한다. 우선 실시간과 임베디드 운영체제에 대한 정의를 2장에서 기술하며, 3장에서는 로알티가 없으며, 다양한 개발자 층을 확보하고 있는 임베디드 리눅스, 그리고 실시간 응용 시스템을 위한 실시간 리눅스의 연구동향을 살펴본다. 4장에서는 실시간 임베디드 시스템의 대표적인 응용 분야인 자동차, 항공기, 네트워크 라우터/스위치 장비, 센서 네트워크, 그리고 모바일 장비에서 사용되고 있는 운영체제들에 대해서 기술하며, 5장에서 간단히 결론을 맺는다.

II. 실시간 임베디드 운영체제

과거의 임베디드 시스템에는 운영체제를 탑재하지 않고 손수 작성된 기본 스케줄러를 이용하여 임베디드 소프트웨어를 구현하기도 하였지만, 임베디드 시스템의 응용 분야가 확장되고 많은 기능들을 요구함에 따라, 시스템의 복잡도가 증가하고 있기 때문에 더 이상 이러한 방법은 효과적이지 못하다. 임베디드 시스템의 다양한 요구사항을 만족시키고 빠른 시장 투입을 위해서는 운영체제를 사용한 개발이 필수적이다. 특히 임

베디드 시스템의 내외적인 환경 변화를 감안할 때, 더욱 강력한 기능을 가진 운영체제가 필요하다. 예를 들어, 현재 임베디드 CPU의 경우 1 GHz의 속도까지 사용 가능하며, 저가격의 대용량 메모리와 디스크온칩(Disk-On-Chip)의 개발로 메모리 제한 사항들이 완화되고 있다. 또한, 최소 메모리 사양의 마이크로 GUI들, 유무선 통신 및 인터넷 접속의 일반화, 다양한 네트워크 인터페이스/프로토콜(USB, PCI, TCP/IP, Firewire, Bluetooth 등)도 임베디드 시스템에 적용 가능하게 되었다.

실시간 운영체제는 임베디드 운영체제보다 더욱 엄격한 요구조건을 만족시켜야 한다. 실시간 운영체제가 갖추어야 할 이러한 조건들로는, 첫째, 우선순위에 의한 선점(preemption)이 가능한 다중 태스크 스케줄링을 지원해야 한다. 둘째, 커널이 하위 우선순위를 가지는 태스크를 서비스하고 있을 때 상위 우선순위를 가지는 태스크가 수행준비가 되면, 커널 자체도 선점이 가능해야 한다. 셋째, 우선 순위 역전 현상(priority inversion)을 방지할 수 있는 우선순위 실링(priority ceiling) 프로토콜이나 우선순위 상속(priority inheritance) 프로토콜이 지원되어야 한다. 넷째, 운영체제 시스템 콜, 인터럽트 지연 시간 그리고 동기화 메카니즘 수행 시간 등이 예측 가능한 상한값을 가져야 한다.^[2]

III. 실시간 임베디드 리눅스

리눅스 커널은 1991년 핀란드의 헬싱키 대학에 다니던 Linus Torvalds에 의해 IBM PC용의 데스크탑 운영체제로 처음 만들어졌으며, GPL (GNU Public License) 라이선스의 오픈 소스 프로젝트로 계속 개발되고 있다. 일반적으로 Linus Torvalds의 커널과 GNU 유틸리티를 합쳐서 리눅스라고 부르며, 현재 안정성을 인정받아 24시간 무중단 시스템인 고성능 서버에서도 많이 사용되고 있다.

최근 수년간 임베디드 시스템에서 리눅스가 많이 사용되고 있는 이유는 다음과 같은 장점들 때문이다. 첫째, 오픈소스이기 때문에 제품당 로열티를 지불할 필요가 없다. 둘째, 그동안 많은 개발자들의 노력으로 현재 안정적이고 신뢰성이 있는 커널이 구현되어 있다. 셋째, 커널의 구성과 확장이 용이하여 커스텀화 하기가 쉽다. 넷째, 네트워킹, 그래픽스, 저장장치 등의 지원이 다양하고 강력하다. 그 외, 많은 수의 경험 있는 리눅스 개발자들에 의해 축적된 기술 및 다양한 개발 틀의 지원이 가능하며, 유일한 멀티벤더 운영체제로써 기존 실시간 임베디드 운영체제를 대체할 큰 대안으로 떠오르고 있다.

그러나, 임베디드 리눅스가 협의의 임베디드 시스템에서는 적합하지만, 실시간 시스템에 사용되기에는 몇 가지 문제점이 있다. 스케줄링과 시스템 콜 수행 시간이 예측가능하지 않고 시스템 부하에 따라서 달라지며, 타임 슬라이스의 정밀도가 일반적으로 10ms이기 때문에 고정밀도를 요구하는 실시간 시스템에는 부적합하다. 커널 또한 비선점적(non-preemptible)이며, 가상 메모리 사용과 평균적인 성능향상을 위해 입출력 요구를 재스케줄링 하기 때문에, 시스템의 동작을 정확하게 예측하기가 어렵다. 이러한 임베디드 리눅스의 단점을 보완하기 위한 실시간 리눅스가 활발히 연구 개발되고 있다.

1. 임베디드 리눅스

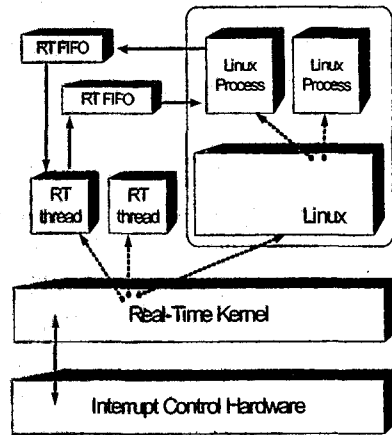
임베디드 리눅스는 PDA, 핸드헬드 장치, IMT-2000 웹폰, 웹패드, PVR(Personal Video Recorder), 셋톱박스, 인터넷 TV, 자동차의 MP3 플레이어, 네트워크 카메라나 블루투스 액세스 포인트와 같이 다양한 분야의 응용 시스템에 사용되고 있다. 임베디드 리눅스에 관련된 프로젝트로는, 임베디드 시스템에 사용되는 다양한 마이크로프로세서 대한 포팅 및 관련 정보들을 제공하는 오픈 프로젝트가 있는데, PowerPC, ARM, SH, VR 등 다양한 플랫폼을 취급한다. 최소한의 메모리 자원을 사용하여 임베디드 리눅스를 구현하기 위한 프로젝트로는 LOAF(Linux

On A Floppy disk), ThinLinux, ELKS (Embedded Linux Kernel Subset) 등이 있다. 그 외 GUI기반의 개발 환경을 포함하는 상용 임베디드 리눅스 제품으로는 Lineo의 Embedix, LynuxWorks의 BulecatLiux, Montavista의 Hardhat 등이 있다.

2. 실시간 리눅스

실시간 리눅스는 크게 두 가지로 방법으로 구현되고 있다. 첫째, RT-Linux와 RTAI(Real-Time Application Interface)의 경우 기존의 리눅스 커널 아래에 새로운 실시간 커널을 만들어서 실시간 태스크들을 스케줄하며, 실시간 태스크가 휴식하는 동안 기존 리눅스 커널을 수행시킨다.¹³⁾ RT-Linux의 경우, 인간형 로봇인 동경대의 Isamo와 후지쯔의 HOAP에 사용되고 있기도 하다. 둘째, KURT의 경우와 같이 기존 리눅스 커널의 일부 부분을 수정하는 방식이 있는데, RT-Linux와 같은 첫 번째 방법이 실시간성을 보다 잘 만족시켜주기 때문에 현재 많이 사용되고 있다. 상업용으로 사용되고 있는 실시간 리눅스로는 FSMLabs의 Open RT Linux, Lineo의 Embedix, LynuxWorks의 BlueCat RT, Montavista의 RT Extension, TimeSys의 Linux/RT 등이 있으며, 그 외 ART Linux, KURT, Linux-SRT, QLinux, RTAI, RED-Linux 등과 같은 연구 프로젝트가 수행되고 있다.

본 고에서는 실시간 리눅스 중 하나인 RT-Linux를 소개한다. RT-Linux는 오픈소스 프로젝트로, x86, PowerPC, Alpha칩을 지원하며, 일반 리눅스 소스코드의 패치로 제공되므로, 단순히 패치를 설치한 후 재 컴파일함으로써 RT-Linux를 쉽게 사용할 수 있다. 실시간 태스크는 RT-Linux상에서 실시간 스케줄링 되며, 비실시간 태스크를 수행하는 기존 리눅스 커널은 RT-Linux상에서 최하위 우선 순위를 가지는 태스크로 수행 된다. RT-Linux는 실시간 시스템 개발자를 위하여 RT API의 제공 및 고해상도의 타이머 서비스와 빠른 인터럽트 지연시간의 보장, 그리고 기존 리눅스 커널의 수정을 최소화함으로



〈그림 1〉 RT-Linux 구조와 태스크간 통신

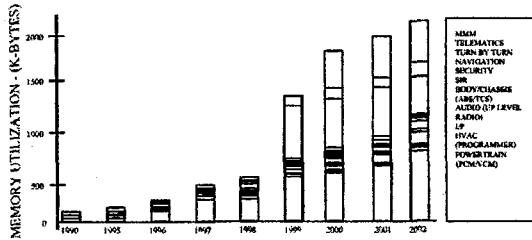
써 비실시간 리눅스 응용 프로그램의 수정을 필요로 하지 않는 것을 특징으로 한다. 실시간 태스크와 리눅스상에서 수행되는 비실시간 태스크는 단방향 실시간 FIFO를 사용하여 통신을 하는데, 〈그림 1〉에 RT-Linux의 구조와 태스크간 통신을 나타내었다.

IV. 응용 분야별 실시간 운영체제 동향

1. 자동차용 실시간 운영체제

전통적인 기계 장치였던 자동차에서도 실시간 임베디드 시스템과 관련 소프트웨어의 중요성이 증대하고 있다. 폭스바겐에서는 미래의 자동차 이노베이션 중 90% 이상이 전자장비와 소프트웨어에 의해 좌우될 것이라 하였으며, BMW는 80%의 자동차 전자장비가 소프트웨어를 탑재할 것이며, 미래 자동차의 기능과 성능은 소프트웨어에 의해 결정이 되어질 것이라고 하였다. 〈표 1〉에서는 이러한 자동차 소프트웨어의 탑재 증가 추세를 잘 보여주고 있다. 그러므로 자동차에서도 다양한 소프트웨어의 기반 플랫폼인 운영체제가 중요하게 여겨지고 있으며, 일부 표준화 작업도 이루어지고 있다. 운영체제를 설명하기 전에 특기할 만한 표준화 작업을 소개하면, 자동차의

〈표 1〉 자동차의 임베디드 소프트웨어
(인용 : Tom Fuhrman, General Motors)

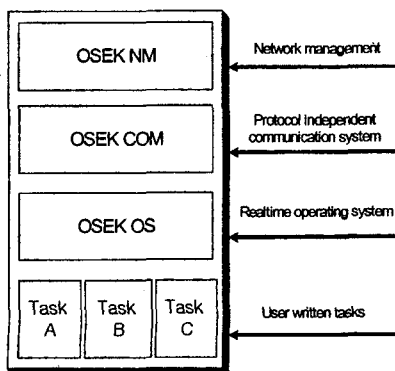


텔레메틱스/네비게이션/엔터테인먼트 기능 확대를 위한 인포테인먼트 버스 (IDB-C, MOST, Bluetooth, IntelliBus, USB, 1394 등 사용) 표준인인 AMIC (Automotive Multimedia Interface)가 주목할만하다.

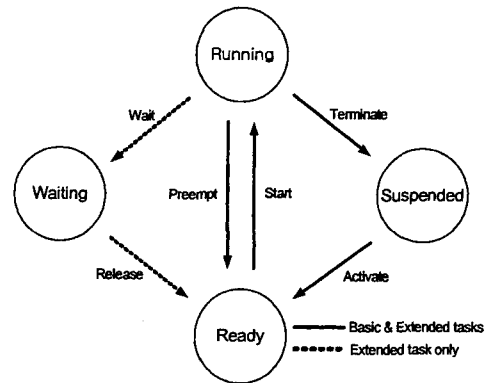
OSEK/VDX는 유럽에서 표준화된 대표적인 자동차용 실시간 운영체제이며, 자동차용 전자장비를 위한 개방 시스템과 인터페이스를 표준화하고 있다.^[4,5] 1993년 독일에서 표준화 작업이 시작되었으며, 오펠, BMW, 벤쯔, 르노, 폭스바겐, 지멘스, 보쉬 등 유럽의 대형 자동차 회사와 전자장비 회사들이 참여하고 있다. 〈그림 2〉는 기본적인 OSEK/VDX 시스템 구조를 보여 준다.

OSEK은 실시간 운영체제인 OSEK OS, 하위 미디어와 프로토콜에 독립적인 통신 시스템인 OSEK COM, 자동차 네트워크 관리에 사용되는 OSEK NM으로 구성된다.

OSEK OS는 태스크 스케줄링 및 관리, 자원



〈그림 2〉 OSEK/VDX 시스템 구조

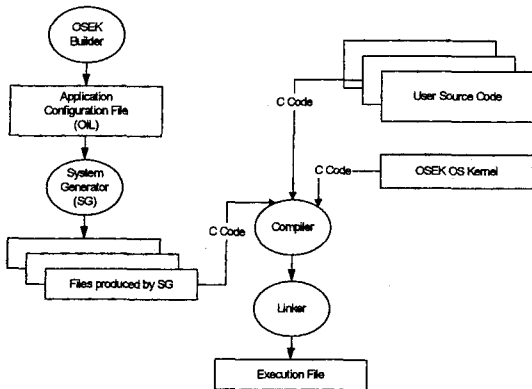


〈그림 3〉 OSEK 태스크 상태천이도

관리, 태스크 통신, 예러 처리, 혹 루틴, 타이머, 알람 그리고 표준 API (Application Programming Interface)를 포함한다. OSEK OS의 태스크는 기본 태스크와 확장 태스크로 구분되며, 확장 태스크의 경우 〈그림 3〉의 태스크 상태천이도에서 보듯이, waiting 상태가 추가된다.

우선순위 선점 (priority preemption)에 관련된 특이한 점은 한 운영체제 내에서 높은 우선순위를 가지는 태스크가 수행 준비가 되더라도 하위 태스크가 끝나기를 기다려야 하는 비선점 스케줄링과, 하위 태스크를 몰아내고 바로 수행 가능한 선점 스케줄링을 동시에 지원한다는 점이다. 그리고, 우선순위 역전 현상을 해결하기 위하여 우선순위 실링 프로토콜을 지원한다.

OSEK COM은 아래로부터 데이터링크 계층, 네트워크 계층, 그리고 인터액션 계층으로 구성된다. 인터액션 계층은 응용 프로그램과 하위 네트워크 서비스 사이에 위치하여, 응용 프로그램에 하드웨어나 마이크로컨트롤러에 독립적인 1:1 또는 1:N 통신 서비스를 제공한다. 네트워크 계층은 목적지로 적절히 통신 패킷을 전달해 주는 기능과 패킷 순차처리 등을 수행한다. 하드웨어에 인접하는 데이터 링크계층에서는 통신 하드웨어와 디바이스 드라이버로 구성되며, 상위 계층에 네트워크 서비스를 제공한다. 태스크 간의 통신은 메시지 교환을 통해서 이루어지며, 물리적인 하드웨어 구성과는 투명하게 동일 프로세서 내에서 또는 다중 프로세서간에 메시지 교환이



〈그림 4〉 OSEK 응용 소프트웨어 생성 절차

이루어진다. 메시지는 최신 센서 데이터 정보를 전달하기 위한 비 큐(non-queued) 메시지와 이벤트 정보 등을 교환하기 위한 큐(queued) 메시지로 구분된다.

OSEK NM은 다양한 버스 프로토콜로 통합 구성된 자동차 네트워크를 관리 하는데, 이는 통신하드웨어 초기화, 네트워크 구성과 유지, 진단 기능 등을 포함한다.

자동차의 통합 전자 장비 시스템을 효율적으로 구성하기 위해 OIL(OSEK Implementation Language)을 사용하는데, 응용시스템의 이식성 향상과 시스템 구성 및 통합에 유리하다. OIL을 사용하여 시스템을 구성하는 각 마이크로프로세서의 태스크와 관련 옵션들을 지정하며, 〈그림 4〉에 보여진 바와 같이, OIL 프로그램은 OSEK Builder와 시스템 자동 생성기에 의해 시스템 구성 관련 C 프로그램으로 변환되며, 프로그래머들이 작성한 C 프로그램과 OSEK OS 커널을 함께 컴파일하여, 최종 실행 파일을 생성한다.

2. 항공기용 실시간 운영체제

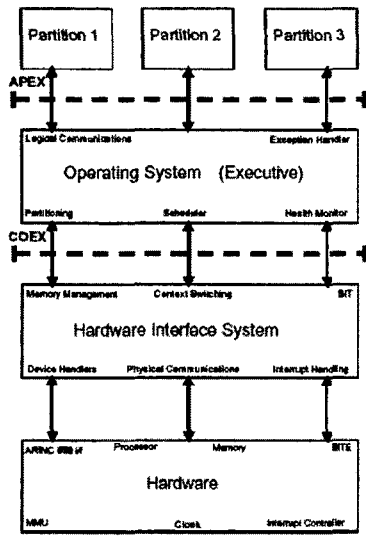
항공기 내에 탑재되는 전자장비와 이들을 제어하는 소프트웨어를 통틀어서 항공전자(avionics)라고 한다. 보잉의 일반적인 여객기의 경우 제작비 중 항공전자의 비중이 30% 이상을 차지하며, 다양한 항공전자 장비를 탑재한 조기경보기의 경우는 제작비의 70% 이상을 차지하는 것으로 봐서 알 수 있듯이 항공전자의 중요성이 날

로 부각되고 있다. 전자장비 뿐만 아니라, 항공전자에서 운용되는 소프트웨어도 그 중요성이 인식되고 있으며, 고신뢰성과 안전을 최우선으로 여기는 특수한 상황에 맞게끔 운영체제가 연구되고 있다. 특히 통합모듈라항공전자(IMA, Integrated Modular Avionics)라는 새로운 개념하에서 운영체제가 활발히 개발되고 있다.^[6,7,8]

기존의 항공기는 네트워크로 연결된 다수의 독립된 항공전자 컴퓨터로 구성되며, 각 컴퓨터들은 일반적으로 네비게이션, 디스플레이, 자동항법과 같은 한 개의 응용/서브시스템을 탑재한다. 이와 같은 경우, 각 서브시스템별로 전원 공급 장치, 마운팅 랙, 그리고 케이ابل링이 개별적으로 요구되므로, 항공기의 무게와 전력 사용은 서브시스템의 수에 따라 증가하는 단점이 있다. 또한 항공기 제작과 유지보수에 많은 비용이 들고, 조그만 수정이 가해질 시에도 재인증을 받기 위해서 천문학적인 비용이 들 수 있다.

한편, IMA는 다수의 응용/서브시스템이 CPU, 메모리, 통신, 디스플레이, 전력 자원들을 공유하며, 더 나아가 라이브러리, 툴, 그리고 데이터베이스와 같은 소프트웨어 자원들도 공유함으로써, 항공기의 무게 및 전력 사용을 감소시켜 비용을 대폭 절감할 수 있다. 하나 또는 소수의 컴퓨터에 다수의 항공 응용 서브시스템의 기능을 탑재하는 것은 고속의 CPU 개발과 메모리 가격의 하락으로 가능하게 되었다. 이처럼 IMA 개념을 사용하면 비용절감 뿐만 아니라, 같은 컴퓨터 상에서 안전을 최우선으로 하는(Safety Critical) 레벨—A급의 응용과 항공기의 안전과는 무관한 응용소프트웨어가 함께 수행될 수 있다. IMA의 핵심 기능인 강력분할(Strong Partitioning)을 포함하는 새로운 소프트웨어 플랫폼(운영체제 포함)이 ARINC-651, ARINC-653 등의 표준화를 통해서 확산되고 있으며, 이를 구현한 Green Hill의 INTEGRITY나 WindRiver의 VxWorks AE653 등이 차세대 항공기 개발에 사용되고 있다.

〈그림 5〉에 2단계의 계층적 스케줄링 구조를 가지는 ARINC 653 소프트웨어 플랫폼을 나타



〈그림 5〉 ARINC 653 IMA 소프트웨어 플랫폼

내었다. 각 응용 서브시스템은 상위 계층에 위치한 파티션에서 수행되며, 이러한 파티션들은 각자의 로컬 운영체제에 의해 해당 응용소프트웨어의 수행을 책임진다. 하위 계층에는 다수의 이러한 파티션들을 총괄하여 스케줄링하는 핵심 운영체제가 위치한다. ARINC 653 운영체제의 필수 기능인 강력분할은 공간분할(Spatial Partitioning)과 시간분할(Temporal Partitioning) 개념으로 나뉘어진다. 공간분할은 각 파티션에 할당된 메모리 영역과 같은 물리적인 자원들이 다른 파티션으로부터 영향을 받지 않는 것을 말하며, 시간분할은 각 파티션에 할당된 시간 자원 역시 다른 파티션으로부터 영향을 받지 않는 것을 말한다. 이와 같이 강력분할을 구현한 운영체제를 사용하면, 동일한 프로세서상에서 비행안전에 매우 중요한 비행제어 시스템과 상대적으로 덜 중요한 객실 온도 제어 시스템이 혼합되어 수행될 수 있는 장점이 있다.

IMA의 하위 계층 핵심 운영체제는 최고 수준의 신뢰성있는 소프트웨어이어야 하며, 커널 모드에서 하드웨어 자원을 액세스한다. 일반적으로, MMU(Memory Management Unit)를 사용하여 공간분할 방화벽을 구현하고, Cyclic 스케줄러를 사용하여 시간분할 방화벽을 구현한다.

그 외, Publish/Subscribe 통신모델, 시스템 진단 기능 등 다양한 기능들을 제공해야 한다. ARINC 653을 지원하는 상용 실시간 운영체제로는 미국 차세대 전폭기인 JSF(Joint Strike Fighter)에 채택된 Green Hill의 INTEGRITY나 WindRiver의 VxWorks AE653 등이 있으며, 그 외 항공기용 실시간 운영체제로서는 Raytheon의 RT-SECURE과 Honeywell의 DEOS 그리고 오픈소스인 RTEMS가 있다.

3. 네트워크 라우터/스위치용 실시간 운영체제

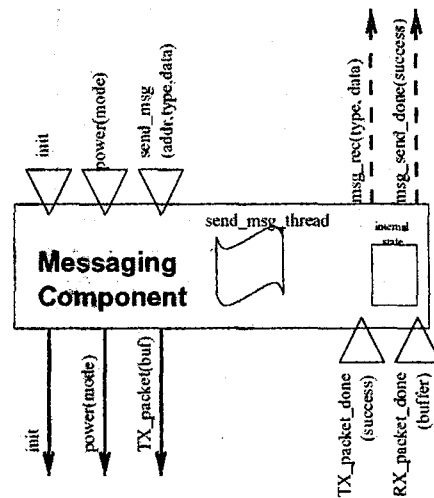
네트워크의 핵심 요소인 라우터나 스위치에 사용되는 운영체제의 경우, 패킷 전달을 위한 라우팅 프로토콜을 수행하는 컴퓨팅 환경에서, 보안 방화벽이나 네트워크 오버레이 등 새로운 서비스나 프로토콜을 동적으로 그리고 효율적으로 수행할 수 있도록 확장성을 고려한 연구가 많이 진행되고 있다. 현재 활발히 개발되고 있는 네트워크 장비를 위한 전용 운영체제로는 프린스턴대의 Scout, MIT의 Click, 워싱턴대의 Router Plugins, 그리고 퍼듀대의 CROSS(Core Router Operating System Support) 등이 있다.^[9] CROSS 라우터 운영체제의 경우, 액티브 네트워크의 액티브 패킷에 의해 동적으로 프로그램을 수행시킬 수 있는 지능형 라우터를 목적으로 개발되었다. 따라서, CROSS는 기존의 네트워크 대역폭 자원 관리 뿐만 아니라, CPU, 메모리, 하드디스크 저장장치와 같은 새로운 자원을 관리하기 위한 스케줄러를 포함한다. 또한 라우터 내에서의 다양한 기능을 동적으로 수행시키기 위해서 패킷 전달 기능을 총괄하며, 다음과 같은 세가지의 패킷 전달 방식을 지원한다. 첫째, 패킷이 최소한의 지연시간을 가지며, 목적지 포트로 바로 라우팅되는 cut-through 방식, 둘째, 액티브 패킷의 도착시 관련 프로그램을 커널 모드 또는 사용자 모드에서 수행을 시작시키는 메카니즘, 셋째, 라우터를 통과하는 각 네트워크 플로우에 라우터 기능 프로그램을 등록하여, 해당 패킷들이 프로그램에 의해 처리가 되도록 하는 메카니즘을 지원한다.

4. 센서 네트워크용 실시간 운영체제

실시간 임베디드 시스템의 새로운 영역으로 센서 네트워크가 최근 활발히 연구되고 있다. 센서 네트워크는 프로세싱, 센싱, 구동, 통신, 저장 장치 등을 가지는 초소형 장치들이 Ad-Hoc 네트워크를 형성하여, 서로 긴밀한 협력 하에서 목표를 수행하는 시스템이다. 미 버클리대의 TinyOS 그룹에서 수행한 센서네트워크의 예를 들어 센서네트워크를 간단히 설명한다. 본 예제 시스템에서는 미지의 정찰 지역에 수십 개의 센서 노드들을 무인 항공기로 떨어뜨린 후, 이들 센서 노드들이 자동적으로 Ad-Hoc 네트워크를 형성하게 한다. Ad-Hoc 네트워크를 형성한 센서 노드들은 근처를 지나가는 자동차의 수, 방향, 속도를 서로 협력해서 계산한 후, 추후 무인항공기가 다시 근처를 지날 때 연결되어서 수집한 정보를 무인항공기로 업로드 한다. 센서 네트워크의 다른 응용으로는 재해 관리, 생태계 관찰, 인체 탐험/진단, 그리고 군사적 목적 등이 있다.^[10]

센서 네트워크의 일반적인 특징으로는 첫째, 센서 노드는 전원, CPU, 네트워크 대역폭, 입출력 면에서 매우 한정된 자원을 가진다는 점이다. 둘째, 센서 네트워크 시스템은 다수의 초소형 노드들이 서로 조화롭게 협동하여 일을 처리하며, 시스템 구조 자체가 동적으로 변화한다. 시스템 구조의 동적 변화 원인은 센서 노드의 고장물이나 유실률이 높으며, 응용 목적상 네트워크 연결 변화가 일반적이기 때문이다. 센서 네트워크의 초소형 센서 노드들을 위한 운영체제는 특성상 concurrency와 모듈성을 지원하는 프레임워크를 제공해야 하며, 블록킹/폴링 방식을 회피하며, 데이터 흐름과 이벤트 처리, 그리고 에너지를 관리하는 방법을 포함하여야 한다.

제한된 자원을 가지는 센서 네트워크 노드를 위해 개발되고 있는 TinyOS는 쓰레드(Thread)와 이벤트를 스케줄링하는 스케줄러와 컴포넌트들이 연결된 시스템 그래프, 두 가지 개념으로 설명된다. <그림 6>의 메시징 컴포넌트의 예에서 알 수 있듯이 TinyOS는 네 개의 컴포넌트로 구성된다. 하위 컴포넌트에게 년블럭킹 요구를 할



<그림 6> TinyOS 개념 구조 예제—메시징 컴포넌트
(인용 : TinyOS-버클리)

때 사용되는 커맨드(commands), 하드웨어 이벤트를 처리하는 이벤트 처리기(event handler), 각 컴포넌트를 위한 메모리 스토리지를 제공하는 프레임(frame), 마지막으로 저전력 지원 스케줄러에 의해 스케줄되는 쓰레드이다. <그림 6>에서, 커맨드의 예로는 역삼각형 마크를 가진 `init`이나 `TX_packet`이 해당되며, 이벤트 처리기는 `TX_packet_done`과 같은 이벤트를 처리하는 삼각형 마크로 표시된다. `Send_msg_thread`가 쓰레드에 해당되며, 이들 모든 컴포넌트들이 필요한 메모리 스토리지를 공급해주는 것이 프레임임이다. TinyOS는 시스템 전체가 하나의 상태머신(State Machine)으로 표현되는 상태머신 프로그래밍 모델을 사용하며, 네 개의 컴포넌트 중 커맨드와 이벤트 처리기의 작동에 의해 시스템의 상태전이가 이루어진다.

5. 모바일 장비용 실시간 운영체제

이동 통신의 발전으로 많이 사용되는 대표적인 모바일 장비로는 퀄컴의 MSM칩을 장착한 CDMA 휴대폰과 모토로라 드래곤볼과 같은 전용 마이크로 프로세서를 장착한 PDA가 있다. 퀄컴의 MSM은 전용 운영체제인 REX(Real-time EXecutive)를 사용하는데, REX는 휴대폰과 같은 소형

임베디드 시스템을 위해 만들어진 5K바이트 이하의 롬영역을 차지하는 간단한 운영체제로써, 우선순위에 의한 태스크 선점을 허용하며, 멀티태스킹을 지원한다. PDA용의 운영체제로는 PalmOS와 마이크로소프트의 Windows CE, PocketPC 2002가 가장 많이 사용되고 있다. 물론 최근에는 임베디드 리눅스를 탑재한 PDA도 많이 소개되고 있다.

V. 결 론

서버나 PC와 같은 일반적인 컴퓨터와는 달리, 실시간 임베디드 시스템은 사용되는 응용에 따라 그 종류가 다양하며, 수행되는 환경도 다르다. 따라서 UNIX 계열과 마이크로소프트의 Windows가 대부분인 일반적인 컴퓨터와는 달리 운영체제의 종류도 매우 다양하다. 본 논문에서는 이러한 여러 응용에서의 실시간 임베디드 운영체제 연구 개발 동향에 대하여 알아 보았다. 그 중에서도 특별히 주목할만한 것은 실시간 임베디드 리눅스가 활발히 연구 개발되고 있으며, 상용 시스템에도 널리 활용되고 있다는 점과, 자동차나 항공기, 그리고 센서 네트워크와 같이 특수 응용을 목적으로 한 전용 실시간 운영체제도 나름대로의 영역을 확보하고 왕성하게 연구 개발되고 있다는 점이다.

참 고 문 헌

- [1] D. Milojevic and et al., "Embedded Systems," IEEE Concurrency, pp.80-90, 2000.
- [2] J. A. Stankovic, "Distributed Real-Time Computing : The Next Generation," J. of the Society of Instrument and Control Engineers of Japan, 31(7), pp.726-736, 1992.
- [3] M. Barabanov, A Linux-based Real-Time Operating System, Master Thesis, NMIMT, 1997.
- [4] OSEK/VDX Operating System Specification 2.2, 2001.
- [5] K. M. Zuberi and et al., "EMERALDS-OSEK : A Small Real-Time Operating System for Automatic Control and Monitoring," SAE Technical Paper Series 1999-01-1102.
- [6] "Design Guidance for Integrated Modular Avionics," ARINC Report 651, Aeronautical Radio Inc, Annapolis, MD, Nov. 1991.
- [7] "Avionics Application Software Standard Interface," ARINC Report 653, Aeronautical Radio Inc, Annapolis, MD, Jan. 1997.
- [8] D. Kim, Y.-H. Lee, M. Younis, "SPIRIT-mKernel for Strongly Partitioned Real-Time Systems," IEEE Real-Time Computing Systems and Applications, pp.73-80, Dec. 2000.
- [9] D. K. Y. Yau, X. Chen, "Resource Management in Software Programmable Router Operating Systems," IEEE J. on selected areas in communications, vol. 19, No. 3, Mar. 2001.
- [10] J. Hill, A Software Architecture Supporting Networked Sensors, Master Thesis, UCB, Dec. 2000.

저자 소개



金大永

1990년 2월 부산대학교 전자계산학과/학사, 1992년 2월 부산대학교 전자계산학과/전산학 석사, 2001년 8월 University of Florida/컴퓨터공학 박사, 1992년 1월~2001년 7월: 한국전자통신연구원/네트워크연구소 연구원, 1999년 5월~1999년 8월: AlliedSignal Aerospace 연구소/방문연구원, 2001년 1월~2001년 8월: Arizona State University/컴퓨터공학과 방문연구원, 2001년 9월~2002년 1월: Arizona State University/컴퓨터공학과 조교수(연구), 2002년 1월~현재: 한국정보통신대학교/공학부 전임강사, <주관심 분야: 실시간 임베디드 시스템, 센서 네트워크, 무인항공로봇, 무선 모바일 컴퓨팅>



都允美

1989년 2월 경북대학교 전자공학과/학사, 1991년 2월 경북대학교 전자공학과/전자공학 석사, 2002년 9월 현재 University of Florida/컴퓨터공학 박사수료, 1991년 1월~1997년 7월: 한국전자통신연구원/네트워크연구소 연구원, 2001년 1월~2002년 9월 현재: Arizona State University/컴퓨터공학과 방문연구원, <주관심 분야: 실시간 임베디드 시스템, 저전력 스케줄링>



金周泓

2001년 2월 성균관대학교 전기전자 컴퓨터공학부/학사, 2002년 9월 현재 한국정보통신대학원대학교/공학부 석박사 통합과정, 2000년 12월~2002년 2월 이스텔 시스템즈(주)/광통신연구소 연구원, <주관심 분야: 실시간 임베디드 시스템, 네트워크 프로세서, 고속인터넷정합기술>



趙惠榮

2000년 2월 부산대학교 전자계산학과/학사, 2002년 9월 현재 한국정보통신대학원대학교/공학부 석사과정, 2000년 2월~2001년 3월: 한국통신하이텔(주), <주관심 분야: 실시간 임베디드 시스템, 네트워크 프로세서, 인터넷 보안>



成鍾宇

2002년 8월 성균관대학교 전기전자 컴퓨터공학부/학사, 2002년 9월 현재 한국정보통신대학원대학교/공학부 석사과정, 2002년 7월~2002년 8월: 한국정보통신대학원대학교/외부연구원, <주관심 분야: 실시간 임베디드 시스템, 멀티미디어 스트리밍, 콘텐츠 분배 네트워크>