

L4 커널 기반에서 운영체제 가상화에 대한 설계

김동근, 신동렬
성균관대학교 정보통신공학부
e-mail : bigbang7@ece.skku.ac.kr, drshin@ece.skku.ac.kr

Design of Virtualization of Operating Systems on L4 Kernel

Dong-Geun Kim, Dong-Ryeol Shin
School of Information and Communication Engineering
Sungkyunkwan University

Abstract

Virtualization of operating systems have already been developed and commercialized in the enterprise computing area. As the computing power of the embedded systems is growing, it is regarded that the virtualization is the important research area. The virtualization is typically established by the micro kernel. L4 kernel is the one example of the micro kernel. In this paper, we propose the architecture for virtualization of Linux over the L4 kernel.

I. 서론

현재 엔터프라이즈 분야에서는 운영체제 가상화를 이용한 시스템이 이미 사용되고 있다. VMware[1] 등이 이러한 시스템의 대표적인 예라고 할 수 있다. 임베디드 시스템이 계속 급격히 발전하고 있고, 여기에 사용되는 소프트웨어도 복잡해지고 있다. 임베디드 시스템의 한 형태로 스마트폰을 생각할 수 있다. 스마트폰은 기본적으로 전화 통화 기능을 제공해야 하고, 또한 사용자의 용도에 맞게 응용 프로그램을 설치해서 사용할 수 있는 환경이기 때문에 여기에 사용되는 운

영체제로 리눅스(Linux), 윈도우(Windows), 혹은 심비안 (Symbian)[2] 등이 적합하다. 이러한 운영체제는 공개된 API를 제공하기 때문에 응용프로그램 개발자에 대해서 편리성을 제공한다는 장점이 있다. 대부분의 휴대전화는 전화 통화에 관련된 부분에 대해서 실시간성을 보장해야 하기 때문에, RTOS (Real-Time Operating System)로 작성된 프로토콜 스택을 가지게 된다. 스마트폰 시스템이 듀얼 코어 혹은 싱글 코어로 구성되어 있는 경우에, RTOS로 되어 있는 프로그램과 리눅스와 같은 일반 운영체제 상에서 작성된 응용프로그램 간에 공존해서 동작시켜야할 필요성이 제기된다. 이러한 요구를 만족시키기 위하여 성능이 보증된 마이크로 커널 기반에서 운영체제를 가상화한다. 그 중에서 경량화 되어 있고, 성능이 검증되어 있는 L4[3]를 기반으로 하여 운영체제를 가상화할 수 있다. 이 논문에서는 L4 커널을 기반으로 하여 리눅스를 가상화하기 위해서 필요한 기법에 대해서 제안한다.

II. 본론

2.1 L4

L4가 나오기 이전에도 마이크로 커널이 많이 존재하였다. 여기에는 Mach[4] 등이 포함되는데, 이 커널은 모노리딕(monolithic) 커널보다 부족한 성능을 보여 실제 사용되기에는 어려움이 따랐다. 마이크로 커널의

이론에 있어서 문제가 없으며, 설계와 구현의 잘못으로 인해서 성능 저하를 가져왔다는 것을 [5]의 논문에서 입증하였다. 그리고 저자는 충분한 성능을 제공하는 커널인 L4커널을 제시하였다. 이 커널은 마이크로 커널의 정신에 입각하여 최소한의 서비스를 제공한다. 여기에는 쓰레드(thread), 주소공간(address space), 프로세스간 통신(Inter Process Communication), 쓰레드에 대한 고유 식별자 (Unique Identifier) 등이 있다. 기존의 운영체제가 제공하던 서비스들은 (디바이스 드라이버 등)은 사용자 모드에서 동작하는 서버로써 동작하도록 한다. L4를 기반으로 하는 운영체제는 여러 개의 서버가 존재하고 다른 쓰레드가 서버에 IPC를 통해서 서비스를 요청하는 방식을 가진다. 결국 IPC의 성능 향상이 L4에서 기존의 운영체제와 비슷한 성능을 갖는 가상화된 운영체제를 구축할 수 있게 한다.

2.2 리눅스의 가상화

일반적인 운영체제는 하드웨어와 직접적으로 연관되어서 그것을 관리하는 역할을 한다. 반면에 가상화된 운영체제는 사용자 모드에서 동작하게 되고, 마이크로 커널이 제공하는 인터페이스를 사용하여 간접적으로 하드웨어에 대한 관리를 하게 된다. 따라서 새로운 인터페이스에 맞게 운영체제의 변경이 필요로 하게 된다.

리눅스를 가상화하기 위해서 인터럽트 처리, 시스템 콜 처리, 예외 처리, 페이지 테이블 관리 등을 새롭게 수정해야 한다. 인터럽트를 처리하는 방식은 CPU 아키텍처에 따라서 다르지만, 인터럽트 요청 번호에 해당하는 인터럽트 처리기가 호출되고 궁극적으로 인터럽트 서비스 루틴 (Interrupt Service Routine)이 호출되도록 매핑을 시켜서 해결한다. 이것을 가상화된 리눅스에서는 L4에서 제공하는 커널 API를 통해서 L4 내부에 있는 자료구조에 인터럽트 처리기를 등록해준다. L4는 인터럽트 요청 번호에 해당하는 각각의 인터럽트 쓰레드를 미리 초기화시킨 상태에서, 인터럽트가 발생할 경우 리눅스 서버의 인터럽트 처리 쓰레드에게 IPC로 변환된 인터럽트를 전달한다. 리눅스 서버의 인터럽트 처리 쓰레드는 IPC 메시지를 이용해서 해당 인터럽트 서비스 루틴을 호출한다. 예외 처리와 시스템 콜은 동일한 방식으로 처리된다. 사용자 프로세스에 의해서 시스템 콜 혹은 예외를 발생시키는 명령어가 실행될 경우, 이것은 L4 커널에서 먼저 발견되고 리눅스 서버에 있는 예외 처리 쓰레드에 IPC 형태로 전달되게 된다. 리눅스 서버의 예외 처리 쓰레드는 시스템 콜과 예외를 처리할 수 있어야 한다. 필요에 따라서 예외 처리 쓰레드는 사용자 프로세스와 IPC를 할 수도

있다. 리눅스 서버에 대한 수정을 최소화하기 위해서 페이지 테이블은 리눅스 서버에서 관리하는 테이블과 L4에서 관리하는 테이블을 두 개 가지도록 설계할 수 있다. 페이지 폴트 예외가 발생하면 이것을 리눅스 서버에서 예외 처리 쓰레드가 L4에 있는 페이지 관리 API를 호출해서 페이지 프레임을 할당받고 페이지 테이블을 수정한다. 이와 함께 리눅스 내부에서 페이지 테이블을 관리하기 위한 자료구조도 수정을 한다.

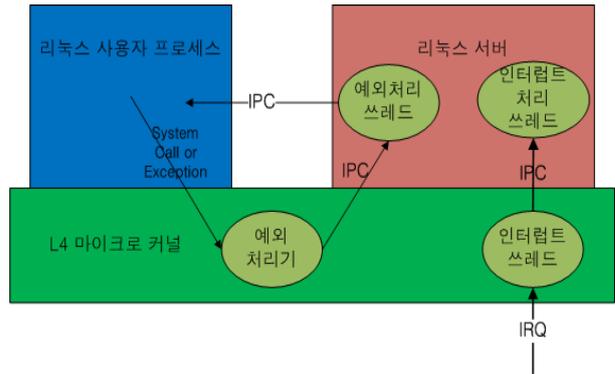


그림 1. 리눅스의 가상화 구조

III. 결론 및 향후 연구 방향

이미 L4 커널의 성능은 다른 논문에서 검증이 되어 있다. 이러한 커널 상에서 기존의 운영체제를 가상화하여 사용한다면, 가상화하지 않는 것에 견주는 성능을 가질 수 있게 될 것이다. 현재 임베디드 환경에서 운영체제 가상화에 대한 연구가 활발하게 진행되는 가운데, 이 논문에서는 리눅스를 L4 마이크로 커널 상에서 가상화하는데 필요한 구조를 제안하였다. 향후에는 이러한 개념을 바탕으로 하여 실제 리눅스 커널을 L4 커널 상에서 가상화하고, 이것을 가상화하지 않은 것과 성능 평가를 하는 연구를 진행할 것이다.

참고문헌

- [1] VMware, <http://www.vmware.com>
- [2] Symbian OS, www.symbian.com
- [3] J. Liedtke. Improving IPC by Kernel Design, Proc. 14th ACM Symposium on Operating System Principles
- [4] Mach, <http://www-2.cs.cmu.edu/afs/cs/project/mach/public/www/mach.html>
- [5] J. Liedtke. On μ -kernel construction, Proc. 15th ACM Symposium on Operating Systems Principles.