

L4 마이크로커널 기반 모바일 미들웨어의 설계

이상민, 신동렬
성균관대학교 이동통신공학과
e-mail : smlee978@ece.skku.ac.kr, drshin@ece.skku.ac.kr

Design of Lightweight Mobile Middleware Based on L4 Microkernel

Sang-Min Lee, Dong-Ryul Shin
Department of Mobile Communications Engineering
Sungkyunkwan University

Abstract

Due to the rapid development of the hardware divisions, varied and low-price mobile devices were provided to not only end-users but also widespread industry. However, in the software divisions, the characteristics of the software such as closed and differentiation causing restrictions of development. Therefore, it requires high costs and long times in the development process. In this paper, we introduce the *Mobile Platform Middleware based on L4 Microkernel* which provides linux-like environment and services for other mobile application platforms such as BREW and WIPI to facilitate easily development and migration of applications for mobile devices.

I. 서론

최근 휴대폰을 비롯한 다양한 임베디드 시스템들의 질적, 양적 발전으로 인해 보다 많은 사용자들 뿐만 아니라 산업 전반에 걸쳐 모바일 기기들의 활용이 확산되고 있다. 그러나 이들을 구동시키는 임베디드 소프트웨어는 개발 주체의 차별성과 소스코드의 기밀성

유지를 위한 폐쇄적인 개발 환경으로 인해 개발자의 접근이 용이하지 않고 하드웨어와 RTOS의 동작에 대한 깊은 이해를 필요로 하는 등 시스템 개발 과정에서 많은 어려움이 뒤따르고 있다. 특히 기존 휴대폰 S/W의 경우 대부분 운영체제의 소스 코드에 직접 포팅되는 구현 방법으로 인해 S/W의 문제점 발생 시 시스템 전체의 치명적인 오류로 이어지기 쉬우며, S/W 문제점 해결을 위한 디버깅과 유지 보수가 어렵다.

따라서, 본 논문에서는 최근 연구와 응용기술의 개발이 활발해 지고 있는 L4 마이크로 커널[2]을 이용하여, 저용량임에도 효율성과 안정성을 동시에 요구하는 임베디드 시스템의 특성을 반영하고, 리눅스와 기존 무선 인터넷 플랫폼(WIPI[4], BREW[5]) 등 다양한 플랫폼에서 개발된 어플리케이션들을 복잡한 포팅 작업 없이 쉽게 재사용할 수 있는 모바일 미들웨어를 설계하고 그 발전 방향을 논의한다.

II. 관련 연구

2.1 L4 마이크로 커널

L4 마이크로 커널은 Jochen Liedke 교수 연구진에 의해 개발되었으며, 성능 향상 여부가 논란이 되었던 기존의 마이크로 커널을 구현 방식의 변경을 통해 그 성능을 획기적으로 개선하여 그 발전 가능성이 확인되었다.[1] 이후 안정성과 가상화 등 여러 부문에 걸쳐

후속 연구들이 진행되고 있으며, 호주의 NITCA 산하 ERTOS 팀 등에서 임베디드 L4 커널 및 L4커널 기반 리눅스 운영체제[3]의 개발을 활발히 진행하고 있다.

2.2 무선 인터넷 플랫폼

WIPI 와 BREW는 서로 유사한 형태를 가지는 무선 인터넷 플랫폼으로서 어플리케이션 개발자에게 제공되는 SDK 에 포함된 API 들을 이용하여 개발된 어플리케이션을 동일한 플랫폼이 탑재된 어떠한 단말에서도 실행 가능하도록 하는 일종의 가상머신이다. 이를 위해 RTOS 상에서 실행되는 특정 Task가 주기적으로 런타임 엔진을 호출하여, 어플리케이션을 실행하거나 제어할 때 필요한 자원(CPU Time)을 할당하고, 어플리케이션과 RTOS 사이의 인터페이스 계층을 통해 실제 단말의 Native 코드를 실행하는 구조로 되어 있다.

최근에는 에뮬레이터와 같은 개발 환경의 편의성 및 예외 처리의 용이성으로 인해 무선인터넷 어플리케이션에만 국한하지 않고 단말의 UI 및 기본 어플리케이션으로 그 구현 범위가 확장되고 있다.

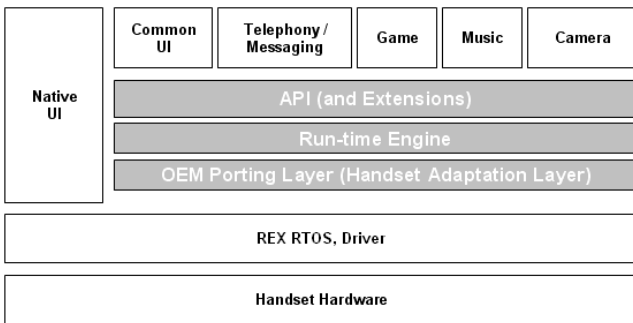


그림 1. 무선 인터넷 플랫폼(BREW, WIPI) 구조

III. 시스템 아키텍처

본 논문에서 제안하는 모바일 미들웨어의 구조는 그림 2와 같다. L4 마이크로 커널을 기반으로 하여, 커널 상위 계층에 기본적인 Linux OS 의 기능을 제공하는 서버와 장치 드라이버들로 구성된 미들웨어 계층을 구현하여, 기존 휴대단말의 RTOS 를 대체하고 이를 통해 C/C++ 로 구현된 리눅스 Native 어플리케이션들을 실행할 수 있는 환경을 제공한다. 또한, WIPI 와 BREW 의 OEM Porting Layer(Handset Adaptation Layer) 를 포함하여, 이와 연동하는 런타임 엔진 프로세스들을 통해 WIPI 와 BREW 어플리케이션들을 동시에 실행할 수 있게 된다.

구현된 커널과 미들웨어 S/W 및 어플리케이션들은 ARM926EJ-S CPU 기반의 Qualcomm MSM6100 칩셋을 탑재한 단말기에서 실행할 수 있도록 ARM 컴파

일러를 이용하여 컴파일하고, 이를 단말의 커널 영역과 파일시스템 영역에 각각 다운로드하여 실행한다.

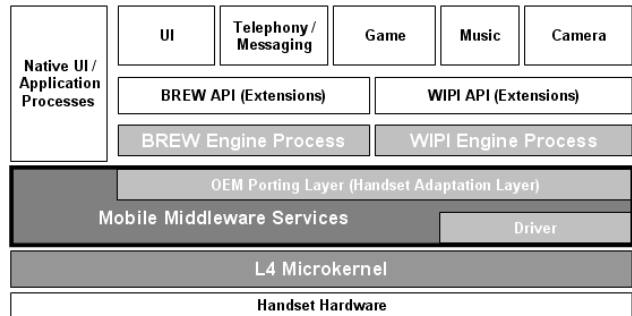


그림 2. L4 마이크로커널 기반 모바일 미들웨어 구조

IV. 결론 및 향후 연구 방향

본 논문에서는 리눅스, WIPI 및 BREW 플랫폼 상에서 개발된 어플리케이션을 별도의 포팅 과정 없이 재사용할 수 있는 환경의 구축을 목표로 휴대단말기나 사용자 인터페이스(UI)를 가지는 임베디드 기기용 L4 마이크로 커널 기반의 모바일 미들웨어를 설계하였다.

그러나 이러한 설계가 단순한 UI 나 어플리케이션들의 실행환경에 그쳐, 실제 휴대 단말에서 가장 중요한 기능인 통신 기능을 반영하지 못하고 있다. 따라서, 통신 기능을 효율적으로 처리할 수 있는 드라이버의 포팅과 이를 이용하는 어플리케이션의 설계가 필요하다.

또한 아직까지 상용화가 본격화되지 않은 L4 마이크로커널이 기존의 RTOS를 대체할 수 있도록 성능 향상과 안정성의 확보가 지속적으로 이루어져야 한다.

참고문헌

- [1] Hermann Härtig, Michael Hohmuth, Jochen Liedtke, Sebastian Schönberg, and Jean Wolter. The performance of μ -kernel-based systems. In *Proceedings of the 16th ACM Symposium on OS Principles(SOSP)*, pages 66 - 77, St. Malo, France, October 1997.
- [2] L4Ka Team. L4Ka::Pistachio microkernel. <http://l4ka.org/projects/pistachio>
- [3] National ICT Australia. NICTA::Pistachio-embedded kernel. <http://www.ertos.nicta.com.au/software>
- [4] 한국 무선 인터넷 표준화 포럼. 모바일 표준 플랫폼(WIPI) 규격 2.0.1, 2004
- [5] Qualcomm Inc. *BREWapi™ OEM Porting Guide for MSM™ Platforms*, 2003