무선인터넷 플랫폼에서 다중 응용프로그램 수행을 위한 스케줄러 설계

김연수*, 강민철, 유용덕, 최훈
충남대학교 컴퓨터공학과
e-mail : {kimys, mckang, ydyu, hchoi}@ce.cnu.ac.kr

Design of Multi-Task Scheduler for a Wireless Internet Platform

Youn-Soo Kim*, Min-Cheol Kang, Yong-Duck Yu, Hoon Choi
Dept. of Computer Engineering, Chungnam National University

요 약

위피(WIPI: Wireless Internet Platform for Interoperability)는 국내 무선인터넷 플랫폼의 표준 규격이다. 따라서 위피 컨텐츠는 위피 플랫폼을 탑재한 어떠한 장치에서도 변경 없이 그대로 실행될 수 있다. 위피 플랫폼은 위피 응용프로그램이 정해진 라이프사이클에 따라 동작하기 때문에 일반적인 운영체제에서 응용프로그램을 멀티태스킹 하는 것과는 다른 방식으로 멀티태스킹을 지원한다. 본 연구팀은 위피 에뮬레이터를 개발하였고, 쓰레드를 사용하여 멀티태스킹을 가능하게 하는 스케줄러를 구현하였다. 본 논문에서는 본 연구팀이 개발한 위피 에뮬레이터를 소개하고, 쓰레드를 사용하여 구현한 에뮬레이터 스케줄러의 단점을 개선하는 새로운 스케줄링 방법을 제안한다.

1. 서론

국내 무선인터넷 시장은 최근 5 년 동안 급속한 성장을 이루었다. 음성통화서비스의 이용률이 감소되어 감에 따라 이동통신사들은 무선인터넷으로 사업영역을 확장하고 있고, 많은 무선인터넷 컨텐츠가 개발되어 성장이 가속화 되어 가는 추세이다.

무선인터넷 플랫폼이란 소형 휴대단말기에 내장되어 마치 컴퓨터의 운영체제(operating system)처럼 응용프로그램을 실행해 주는 소프트웨어를 말한다. 국내/외 무선인터넷 환경은 이동통신사 별로 각기 다른 무선인터넷 플랫폼을 사용하기 때문에 응용프로그램이 각자 자신의 실행 환경에서만 동작하도록 되어 있어서 다양한 무선인터넷 응용프로그램 및 컨텐츠의 상호 호환이 되지 않는다. 그로 인해 컨텐츠 개발자들은 여러 종류의 무선인터넷 플랫폼에 맞추어 하나의 컨텐츠를 여러 버전으로 만들어 재배포 해야 하는 중복투자가 불가피하였고, 사용자들은 원하는 서비스를 이동통신사가 달라서 이용하지 못하는 경우도 발생하였다.

이러한 불합리성을 극복하기 위해서 세계적으로 무선인터넷 플랫폼의 표준화 노력이 이루어지고 있으며, 국내에서는 한국무선인터넷표준화포럼(KWISF: Korea Wireless Internet Standardization Forum)이 위피를 제정하였다[1][2]. 위피는 국내 무선인터넷 플랫폼의 표준 규격이며, 한국정보통신기술협회(TTA: Telecommunication Technology Association)를 통해 단체 표준으로 채택되었다. 표준 규격이 제정됨에 따라 이에 맞추어 개발된 위피 컨텐츠는 이동통신사나 단말기의 종류에 관계없이 위피 플랫폼이 탑재된 어떠한 장치에서도 변경 없이 사용할 수 있다.

* 본 연구는 지역전략산업 석박사 연구인력 양성 사업의 지원으로 수행되었음

위피 플랫폼은 여러 개의 응용프로그램을 메모리에 적재하고 동시에 실행 가능해야 하며, 매 순간 실행 가능한 가장 높은 우선 순위의 응용프로그램을 실행할 수 있어야 한다[1][2]. 따라서 우선 순위에 따라 어떤 응용프로그램을 실행할 것인지 조정할 수 있는 스케줄러가 필요하다.

본 연구팀은 위피 컨텐츠를 쉽고 빠르게 개발할 수 있도록 하는 위피 에뮬레이터를 개발하였고, 쓰레드(thread)를 사용한 스케줄러를 구현하여 적용하였다. 그러나 쓰레드 기반의 스케줄러는 쓰레드간 문맥교환(context switching)으로 인한 오버헤드가 있고, 쓰레드를 지원하지 않는 단말기 운영체제에는 적용하기가 힘들다는 단점을 가진다. 본 논문에서는 이러한 단점을 보완하여, 쓰레드를 지원하지 않는 단말기 운영체제에서도 쉽게 적용 가능한 경량 스케줄러를 제안하였다. 본 논문의 2 장에서는 본 연구팀이 개발한 위피 에뮬레이터를 소개하고, 에뮬레이터에서 사용하는 쓰레드 기반 스케줄러의 소개 및 장단점을 살펴본다. 3 장에서는 새롭게 제안하는 스케줄러를 설명하고, 4 장에서는 결론을 맺고 향후 연구 방향에 대해서 기술한다.

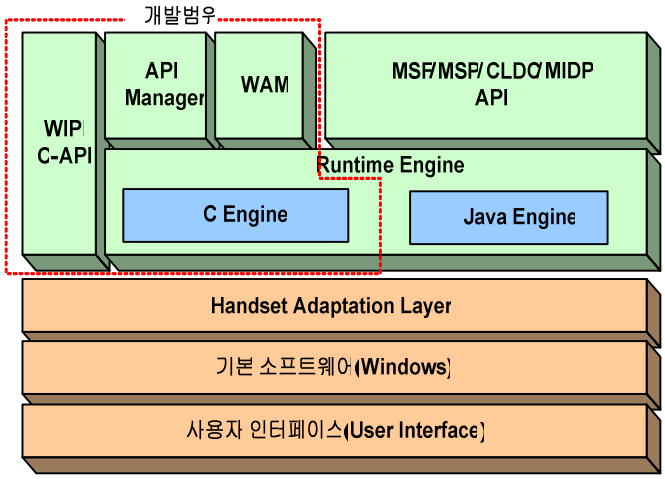
2. 위피

2.1 위피 Clet 에뮬레이터

위피 응용프로그램을 개발하고 검증하는데 있어서 좀 더 쉽고 효율적인 방법은 일반 데스크탑 환경에서 응용프로그램을 작성하고 실행해 보는 것이다. 이를 위해서는 실제 단말기에서 위피 응용프로그램이 실행되는 것과 동일한 결과를 흉내 낼 수 있는 에뮬레이터 프로그램이 필요하다. 본 연구팀은 위피의 표준 규격 버전 1.2 에서 C API 를 지원하는 마이크로소프트 윈도우 기반의 위피 Clet 에뮬레이터를 개발하였다. Clet 은 위피의 C API 와 JAVA API 중 C API 를 사용하여 만들어진 위피 응용프로그램을 말한다.

위피 Clet 에뮬레이터는 단말기의 하드웨어 역할을 하는 GUI 와 윈도우용 HAL(Handset Adaptation Layer), 그리고 위피 플랫폼으로 구성되어 있다. 위피 플랫폼은 메모리 관리자, 스케줄러, 이벤트 핸들러, 링커 및 로더 등 위피 응용프로그램을 실행하는데 꼭 필요한 핵심 부분을 담고 있는 C Engine 과 C API, API 관리자, WIPI 응용프로그램 관리자들로 이루어져 있다[1][3] ([그림 1]).

[그림 1]에서 사용자 인터페이스(User Interface)는 윈도우용 프로그램으로서 하드웨어와 같은 역할을 한다. 사용자 인터페이스는 사용자로부터 입력을 받거나 결과값을 LCD 창을 통하여 사용자에게 보여준다. 앞서 설명한 바와 같이 본 에뮬레이터는 일반 데스크탑 컴퓨터에서 구동되기 때문에 일반적인 운영체제인 마이크로소프트 윈도우를 단말기 기본 소프트웨어로 사용하였다. HAL 은 단말기 기본 소프트웨어에서 구동되어야 하므로 역시 윈도우용으로 포팅된 것을 사용하였다. HAL 의 상위에는 위피 플랫폼이 있다. 본 에뮬레이터는 Java API 는 지원하지 않으며 C API 만 지원한다.

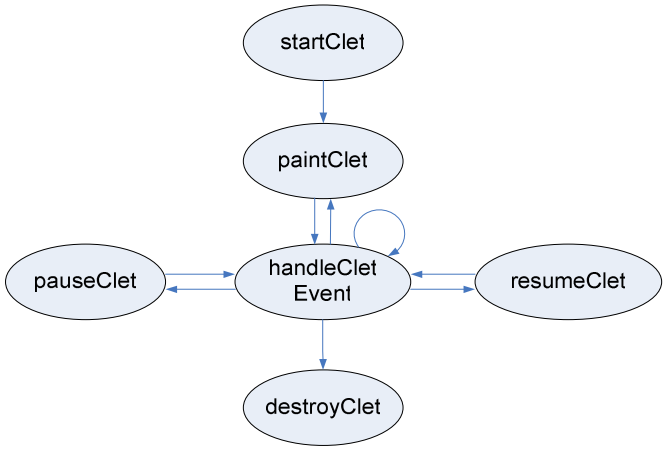


[그림 1] 위피 Clet 에뮬레이터 구조도

에뮬레이터를 실행시키면 단말기 운영체제가 먼저 사용자 인터페이스를 구동하고 위피 플랫폼을 시작한다. 위피 플랫폼은 HAL 과 C API 를 초기화하고 위피 응용프로그램을 실행시킬 준비를 마친다. 위피 응용프로그램은 위피 Clet 에뮬레이터와 별도로 작성되어 플랫폼에서 실행되어야 하기 때문에 동적 링킹 및 로딩을 위해 윈도우의 DLL 형태로 작성하여 사용한다.

2.2 위피 Clet 에뮬레이터에서의 응용프로그램 실행

startClet, paintClet, handleEventClet, pauseClet, resumeClet 함수는 Clet 이 구현해야 하는 함수이며, 플랫폼은 이벤트에 따라 해당 함수를 불러준다[3][4]. [그림 2]는 Clet 의 라이프사이클(lifecycle)을 도시한 것이다.



[그림 2] 위피 Clet 의 라이프사이클

Clet 이 처음 실행되면 플랫폼은 개발자에 의해 Clet 에 구현된 startClet 과 paintClet 을 차례대로 실행한다.

그리고 나서 Clet 은 자신에게 전달될 이벤트를 기다린다. 이벤트는 최초에 플랫폼으로 전달되는데 전달된 이벤트 중 Clet 이 처리해야 하는 일반적인 이벤트(예: key event, timer event 등)는 handleCletEvent 에서 처리가 되며, Clet 을 중지/재개 시키거나 종료시키는 등의 플랫폼 수준의 이벤트는 플랫폼이 pauseClet/resumeClet, destroyClet 을 호출하고 나서 Clet 자체를 중지/재개 또는 종료 시키는 것으로 처리된다([그림 2]).

위피에서는 응용프로그램이 일반 운영체제에서의 멀티태스킹처럼 다수의 프로그램이 동시에 실행되는 형태가 아니라 라이프사이클에 따라 실행되면서 다중 응용프로그램 수행을 지원하는 방식을 취한다.

하나의 응용프로그램은 로딩되어 실행되다가 다른 응용프로그램이 실행될 필요가 있을 때 잠시 중지되고(pauseClet), 새롭게 실행되어야 하는 응용프로그램은 라이프사이클에 따라 실행된다. 즉, 어느 응용프로그램이 실행되려면 반드시 현재 실행중인 응용프로그램이 적절한 절차를 밟아서 중지되거나 종료된 후여야 한다. 따라서 위피 응용프로그램은 특별한 스케줄링 알고리즘은 필요하지 않고, 하나의 응용프로그램이 실행되는 도중에 다른 응용프로그램을 실행하겠다는 이벤트를 받게 되면, 현재 실행중인 응용프로그램을 중지시키고 새로운 응용프로그램을 실행하는 방식으로 다중 응용프로그램을 지원한다. 본 연구팀은 이러한 기능을 구현하기 위해서 쓰레드를 이용하였다. Clet 쓰레드는 위피의 라이프사이클을 따라서 실행되며, 플랫폼은 이벤트를 받았을 때 Clet 쓰레드 자체를 중지/재개 또는 종료 시키는 구조이다.

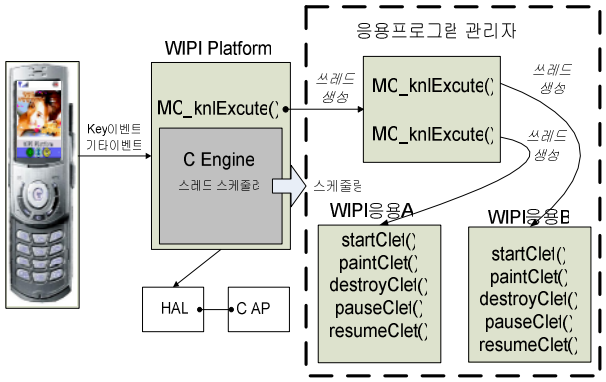
2.3 위피 Clet 에뮬레이터의 제약사항

본 에뮬레이터는 윈도우 운영체제를 기반 소프트웨어로 사용하는 것을 기본 전제로 하고, 위피 응용프로그램이 위피 플랫폼을 탑재한 단말기에서 실행될 때와 동일한 결과를 보여주는 것을 목적으로 하여 만들어 졌다. 따라서 윈도우 운영체제용 HAL 을 사용하고 있으며, 위피의 구조상 HAL 의 상위 계층인 위피 플랫폼은 HAL 을 통해서만 기반소프트웨어(OS)를 이용할 수 있어야 한다. 그러나 구현 목적상 최소한의 범위 내에서 HAL 을 이용하지 않고 직접 기반 소프트웨어를 이용하였다. 예를 들면, 위피 응용프로그램을 위한 Clet 쓰레드 생성 및 관리나 응용프로그램의 동적 링킹 및 로딩 등이다.

2.4 쓰레드 방식의 스케줄러

2.2 절에서 설명한 바와 같이 기존에 구현된 위피 플랫폼은 응용프로그램을 실행하기 위해 Clet 쓰레드를 생성한다.

에뮬레이터가 실행되면, 위피 플랫폼은 응용프로그램 관리자를 실행시키기 위해 쓰레드를 생성한다. 어플리케이션 관리자는 최초 실행되는 프로그램으로서 최상위의 부모프로그램이다. 이것은 다른 위피 응용프로그램을 실행, 중지, 종료하는 기능과 응용프로그램을 설치, 삭제하는 기능을 가진다. 응용프로그램 관리자가 자식 프로그램인 위피 응용프로그램을 실행시키면, 플랫폼에서 새로운 쓰레드가 생성되어 실행된다. 위피 응용프로그램들은 각각의 쓰레드 내에서 각자의 라이프사이클을 따라 실행된다. 이때 위피 플랫폼은 생성된 응용프로그램 쓰레드를 모두 관리하면서, 이벤트에 따라 어떤 응용프로그램이 실행되어야 할지를 결정한다. 예를 들어 위피 응용프로그램 B 가 실행되어야 할 경우 응용프로그램 A 의 pauseClet 이 실행되고, 플랫폼은 A 의 쓰레드를 중지 시킨다. 그리고 나서 B 의 쓰레드를 다시 재개한다.

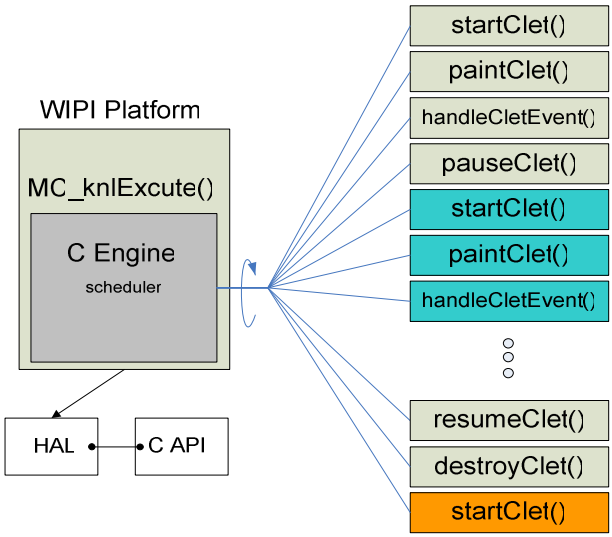


[그림 3] 쓰레드방식 스케줄러

[그림 3]에서 큰 사각형은 쓰레드를 나타낸다. 이 구조는 윈도우용 에뮬레이터에서는 큰 문제가 되지 않을 뿐만 아니라, 쓰레드 별로 응용프로그램을 관리하기 때문에 쉽게 구현이 가능하다. 그러나 다중 작업을 동시에 빠르게 실행할 수 있는 쓰레드의 장점은 위피의 응용프로그램에서는 필요가 없다. 또한 HAL API 에 쓰레드 기능이 정의되지 않았기 때문에 이러한 방식을 단말기에서 사용하려면 사용자 레벨의 쓰레드 라이브러리를 C Engine 내에 직접 구현해야만 한다. 이것은 REX 와 같이 쓰레드를 지원하지 않는 OS 에 위피 플랫폼을 포팅할 경우 큰 부담이 될 것이다. 위피 에뮬레이터 1.0 에서는 HAL 을 거치지 않고 윈도우의 쓰레드 기능을 직접 이용하였다. 그러나 시스템 의존적이라는 제약이 따른다. 따라서 쓰레드를 사용하지 않고 자체 구현한 효율적인 스케줄링 방법을 제시한다.

3. 경량 스케줄러

앞서 설명한 바와 같이 하나의 위피 응용프로그램이 실행되려면, 라이프 사이클에 따라 해당 함수가 호출되어 실행되기만 하면 된다. 따라서 일반 쓰레드를 쓰지 않고도 동일한 효과를 낼 수 있다. [그림 4]는 이러한 방법을 보여준다. 일반적인 쓰레드 방식의 스케줄러에서는 쓰레드가 실행되면서 자체적으로 Clet 의 함수를 호출하는 방식이었으나, 이것은 플랫폼의 C Engine 이 Clet 내의 함수들을 맡아서 관리하고 호출함으로써 스케줄링한다.



[그림 4] 경량 스케줄러

위피 응용프로그램이 실행되기 위해 로딩되면, 플랫폼은 그 응용프로그램을 실행하기 위한 정보들을 수집하고 라이프 사이클에 따라 Clet 내의 함수들을 차례로 실행한다. 하나의 응용프로그램이 실행되는 도중에 다른 응용프로그램이 실행되어야 할 경우 마찬가지로 그 응용프로그램을 실행하기 위한 정보들을 수집하고 라이프 사이클에 따라 함수들을 실행한다. 새로운 응용프로그램이 실행되면 이전에 실행되던 응용프로그램은 중지 되어야 하는데, 그 응용프로그램의 pauseClet 을 실행시키고, 특별히 context 를 저장하거나 할 필요가 없다. Clet 함수들이 실행되지 않으면 응용프로그램은 중지된 것처럼 보이며, 이때 새롭게 실행되는 응용프로그램이 사용자 인터페이스의 LCD 화면에 동작중인 결과를 보여주면 사용자에게는 마치 이전 응용프로그램이 중지되고 새로운 응용프로그램이 실행되는 것처럼 보인다. 이러한 방법으로 다중 응용프로그램 수행 기능을 제공할 수 있다. 물론 이전 응용프로그램의 정보는 C Engine 이 모두 가지고 있어야 하고, 나중에 다시 실행되어야 할 때 사용된다.

위에서 제시한 방법은 실행 중인 어떤 한 응용프로그램에서 무한루프를 돈다거나 오류가 생겨서 CPU 를 계속 점유하면 다른 응용프로그램이 실행될 수 없다는 단점이 있다. 즉, 비선점형(Non preemptive)방식의 운영체제에서처럼 하나의 응용프로그램이 비정상적인 방법으로 CPU 를 점유하면 시스템을 재부팅 해야 하는 것처럼 플랫폼을 재부팅 해야 하는 단점이 있다. 그러나 무선인터넷 응용프로그램은 사용자에게 제공되기 이전에 공인된 기관으로부터 검증을 받아 제공되므로[1] 응용프로그램의 불안정성으로 인해 발생하는 단점은 무시될 수 있으며, 쓰레드 방식의 스케줄러에 비하여 응용프로그램간의 문맥교환이 단순하기 때문에 보다 빠르며, 쓰레드 기능이 없는 운영체제에 포팅될 때 이식성이 좋다.

4. 결론

무선인터넷 플랫폼의 표준 규격이 없어서 다양한 종류의 플랫폼과 수많은 컨텐츠들이 복잡하게 혼재된 상황에서도 무선인터넷 시장은 꾸준한 성장을 계속하여왔다. 그러나 플랫폼에 따른 컨텐츠의 상호 호환성 결여는 문제점으로 인식되어 왔고, 국산 토종 플랫폼이 없다는 것 또한 위피라는 표준 플랫폼 규격을 만들게 된 중요한 동기가 되었다.

위피가 국내 표준 규격으로 자리잡게 됨에 따라 본 연구팀은 위피 컨텐츠를 쉽게 개발하고 검증할 수 있는 에뮬레이터 버전 1.0 을 구현하였다.

본 논문에서는 개발한 에뮬레이터 버전 1.0 의 핵심 모듈인 C Engine 의 스케줄러 구현 방식을 살펴보고, 기존 방식의 단점을 보완하여 빠르고 좀더 다양한 하드웨어에 쉽게 이식될 수 있는 새로운 스케줄링 방식을 제안하였다. 향후 연구는 제안한 스케줄러를 구현하고, 단말기를 통하여 검증한다.

참고문헌

[1] 모바일 표준 플랫폼 WIPI 1.2.1, KWISFS.K-05-001R3

[2] 모바일 표준 플랫폼 WIPI 2.0, KWISFS.K-05-002

[3] “윈도우용 Clet 에뮬레이터 개발”, ETRI 연구보고서, 충남대학교, 2004.8

[4] “휴대정보단말기용 웹 서비스 클라이언트 및 WIPI SDK”, ETRI 연구보고서, 충남대학교 2003.12

[5] www.kwisforum.org, 한국무선인터넷표준화포럼

[6] www.mobilejava.co.kr

[7] 유용덕, 김연수, 최훈, “동적 Upgrade 기능을 구비한 WIPI 개발환경 설계”, 한국통신학회 추계 종합 학술발표회 논문초록집, vol. 28, p.343, 2003.12