

Process-Based 운영체제 기반 Embedded S/W 의 Thread-Based 운영체제 기반으로의 변환을 위한 OS Abstraction Layer 의 구현

이종인, 신정민, 한국현, 권재욱
삼성전자 디지털 미디어 연구소 SW Platform Lab

Implementation of OSAL to Migrate Embedded S/W from Process-Based System to Thread-Based System

*Jong In Lee, Jungmin Shin, Kuk-Hyun Han, Jaewook Kwon
Software Platform Lab Digital Media R&D Center Samsung Electronics
*chadlee@samsung.com

요 약

정보 가진 분야에 있어서 급속한 기술 발전으로 인해 하루가 다르게 새로운 기능이 추가됨에 따라 embedded system S/W 의 복잡도 또한 증가하고 있고 이를 개발하고 유지보수 하는데 있어서도 막대한 비용과 노력이 요구 되고 있다. 이로 인해 embedded S/W 에도 재사용성의 요구가 높아지고 있지만 embedded system 의 경우 제품의 특징에 따라 system 의 process model 을 변경해야 하는 경우가 있는데 이러한 경우 OS 의 변경으로 인해 S/W 의 재사용성이 크게 낮아진다. 즉, process-based model 에서 thread-based model 로 변경하는 경우 OS 변경으로 인해 전체 S/W 의 변경이 요구되어 개발의 효율이 떨어지게 되는 것이다. 이에 본 논문에서는 OS 변경으로 인한 전체 S/W 의 변경을 최소화 하고자 OS abstraction layer 를 도입하고 process-based model 에서 thread-model 로의 변환 시 발생하는 변경 요소인 process 와 thread 간의 변환에 따른 memory 및 scheduling 관련 사항, IPC (Inter-Process Communication) 관련 사항, device I/O 관련 사항 중, 상위 S/W 의 재사용성에 직접적인 영향을 미치는 IPC 에 대해 OSAL 을 통한 process/thread model 공통 interface 를 제안하고자 한다. 특히, 고려해야 할 사항이 많은 components 인 semaphore, shared memory, socket communication 등의 OSAL 구현 방법을 Linux to VxWorks 의 경우에 대해 제안 하고 이를 이용한 DTV 시스템의 구현을 통해 타당성을 검증한다.

I. 서론

임베디드 시스템의 기능이 다양해지고 규모가 커짐에 따라 운영체제(OS)의 사용은 필수적이 되었고, OS 선정 또한 시스템의 특성에 적합하게 다양화되고 있다[1]. 이에 임베디드 시스템 구현에 있어서 OS 선정은 시스템 구현의 시작일 뿐만 아니라 시스템의 성능을 결정하는 중요한 요소이다. 하지만 각 시스템 및 OS 가 갖는 특성으로 인해, 특정 시스템에 특정 OS 가 적합하다는 식의 일률적인 규칙을 적용하기는 어렵다. 이로 인해 프로세스 모델을 변경해야 하는 경우가 발생하게 되는데 이 때 OS 변경으로 인해 전체 S/W 의 재사용성이 크게 떨어지게 된다. 프로세스 기반 모델은 응용 프로그램이 process 의 단위로 수행되며 각 process 들은 서로의 메모리가 보호되어 있어서 application 개발이나 모듈의 추가, 변경이 용이하고, 안정된 시스템의 개발이 가능하여 대규모 시스템의 개발에 적합한 특징을 갖는데[2], 반면 쓰레드 기반 모델은 응용 S/W 즉, thread 들이 address 공간을 공유하는 방식으로 공통의 작업영역(memory)을 자유롭게 접근할 수 있을 뿐 아니라 OS 의 크기가 작고, 구현이 용이하고 빠르다는 특징을 갖기 때문이다[3]. 특히 DTV 와 같은

경우는 임베디드 시스템의 규모가 커지고, 각 CE(Consumer Electronics) 제품 군의 특성이 복잡해짐에 따라 다양한 OS 의 적용이 시도되게 되는데, 이러한 경우 OS 의 변경으로 인하여 수정이 필요 없는 응용 프로그램 등을 포함한 전체 S/W 의 변경 및 재구현이 불가피한 경우가 발생한다[4]. 이를 방지하기 위해서는 OS 변경에 따른 수정 부분 최대한 abstraction 하여 빠른 시간 내에 시스템의 재구성이 가능하도록 하는 embedded S/W 구조가 필요하게 된다.

본 논문에서는 특히 상위 S/W 의 변경이 많이 요구되는 process-model 에서 thread-based model 로의 변환 시 필요한 OSAL 을 구현하는 데 있어서 고려 요소인 process-thread 변환에 따른 memory 및 scheduling 관련 사항, IPC(Inter-Process Communication) 관련 사항, device I/O 관련 사항 가운데 상위 S/W 의 재사용을 낮추는 직접적인 영향을 갖는 IPC 에 대해 OSAL 을 구현하는데 있어서 OSAL 을 통한 process/thread model 공통 interface 를 제안하고자 한다. 특히, semaphore, shared memory, socket communication 등 process model 변환 시 고려해야 할 사항이 많은 components 의 OSAL 구현 방법을 제안하고 Linux to VxWorks 의 경우에 대