스마트 카드를 위한 칩 운영체제 설계

*남궁우, **조상영

한국외국어대학교

*milkcow1234@naver.com, **sycho@hufs.ac.kr

Design of chip operating system for smart card

Woo Namgoong, Sang-Young Cho

Hankook University of Foreign Studies

요 약

최근 고성능 스마트 카드가 모바일 폰, 자바 카드, 전자 여권, 은행 카드용으로 여러 회사에서 다양하게 출시되고 있다. 이러한 스마트 카드는 내부의 다양한 응용 프로그램을 수행하기 위한 칩 운영체제를 가지고 있다. 본 논문은 다운로드 가능한 CAS 시스템에 특화된 칩 운영체제 설계에 대해 기술한다. 또한 칩 운영체제를 구현하기 위한 개발 환경으로 가상 개발 환경에 기초한 스마트 개발 환경 구현에 대해 기술하며 다양한 스마트 카드 응용 프로그램 개발을 위한 가상 개발 환경에 대해 논의한다.

1. 서론

스마트 카드는 데이터를 처리할 수 있는 내장된 집적 회로를 가지고 있는 포켓 크기의 카드이다. 최근 모바일 폰, 자바 카드, 전자 여권, 은행 카드, 교통 카드, 또는 전자 상거래 응용 등 스마트 카드의 분야가 확대되면서 여러 회사의 다양한 고성능 스마트 카드가 출시되고 있다[1]. 기존 셋탑 박스의 CAS (Conditional Access System)는 스마트 카드에 기반하여 구현이 되고 있으며 대상과 동작이 고정되어 있는 구조를 갖는다. DCAS (Downloadable CAS)는 소프트웨어직인 접근 방법으로 DRM (Digital Rights Management)을 제어할 조건부 접근 클라이언트를 안전하게 다운로드하고 운영하기 위하여 CableLabs에 의하여 제안되었다. DCAS는 기존 CAS에 비하여 다양한 프로그램 공급자를 지원할 수 있으며 다양한 가변적인 상황을 지원할 수 있는 유연한 구조이다[2]. 이러한 DCAS 응용을 구축하기 위해서는 스마트 카드가 다수개의 클라이언트를 동작시켜야 하기 때문에 칩 운영체제 (COS: Chip OS)가 필요하다.

본 논문에서는 DCAS 시스템에 특화된 COS의 설계 및 구현에 대하여 기술한다. 또한 칩 운영체제를 구현하기 위한 개발 환경으로 가상 개발 환경에 기초한 스마트 개발 환경 구현에 대해 기술하며 다양한 스마트 카드 응용 프로그램 개발을 위한 가상 개발 환경에 대해 논의한다.

2. DCAS 시스템

그림 1은 본 논문에서 가정하고 있는 스마트 카드와 카드 외부의 스마트 카드 제어부를 도시하고 있다. SM (Secure Micro)이라 불리는 스마트 카드와 외부는 ISO 7816 인터페이스를 통해 통신하며 SM 드라이버와 DCAS Monitor가 APDU 단계의 통신을 수행한다.

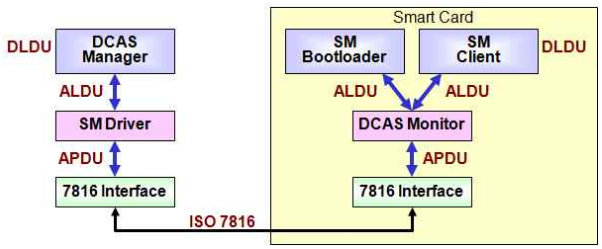

그림 1. DCAS 내의 스마트 카드 구조

SM 클라이언트들은 DRM을 위한 키 관리가 담당한다. SM 부트로더는 스마트 카드를 초기화 시키며, 암호화, 클라이언트, 메모리에 대한 관리 기능을 갖는다. DCAS Monitor는 DCAS Manager가 전달하는 메시지를 내부의 SM 부트로더 또는 SM 클라이언트로 전달하고 처리 결과를 다시 반환하며 스마트 카드 내부 전체를 관리한다. DCAS Manager 메시지는 클라이언트의 다운로드, 수행, 삭제 명령들을 포함하고 있으며 명령어에 따라 클라이언트를 담당하는 쓰레드가 활성화되어 동작되어야 한다.

3. DCAS를 위한 Chip OS 설계

COS의 주요 기능은 DCAS Monitor와 클라이언트의 동작을 위한 쓰레드와 호스트 프로그램, DCAS Monitor, 클라이언트 사이의 통신을 지원하여야 한다. 우리는 스마트 카드 내에서 최대 3개의 클라이언트가 동작할 수 있다고 가정을 하여 DCAS Monitor 포함 최대 4개의 쓰레드를 지원한다. 클라이언트는 CAS, DRM, ASD 형태가

가능하며 각 클라이언트는 COS 구현에 독립적으로 제작될 수 있다고 가정한다. 즉, 각 클라이언트는 스마트 카드 동작 중에 동적으로 다운로드되고 수행될 수 있어야 한다. 쓰레드는 NOUSE, CREATED, READY, RUN 등의 상태를 가지며 클라이언트 쓰레드 사이에서는 문맥전환이 발생하지 않도록 하여 클라이언트들 사이의 보안이 보장되도록 하였다. 쓰레드 사이의 통신을 위하여 비동기적으로 동작하는 Send와 Receive를 제공하며 이를 이용하여 보다 상위 단계의 통신 프리미티브를 구축할 수 있게 하였다.

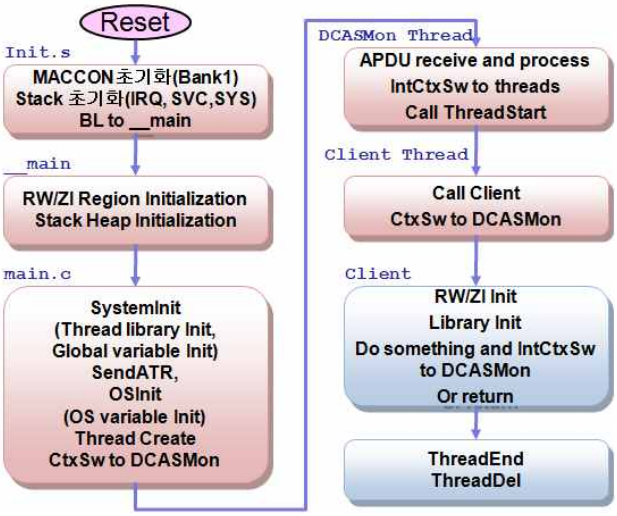

그림 2. DCAS 내의 스마트 카드 구조

그림 2는 스마트 카드 내 초기화와 COS 동작을 도시하고 있다. 스마트 카드는 ARM 사의 SC100 코어를 사용한다고 가정하고 내부적으로 IRQ, SVC, SYS 모드만 사용하기 때문에 이에 관련된 스택 초기화와 실행 데이터 초기화 그리고 COS 초기화를 수행하고 Monitor와 클라이언트를 담당하는 쓰레드를 생성하고 DCASMon 쓰레드로 전환한다. DCASMon은 호스트로부터 계속적으로 APDU 명령을 받아 관련 처리를 하거나 해당 클라이언트를 위한 쓰레드로 문맥전환한다. 클라이언트 쓰레드는 명령에 따라 해당 클라이언트를 실행하고 다시 DCASMon으로 문맥전환한다. 각 클라이언트는 동적으로 실행되어야 하기 때문에 마치 시스템이 부팅하듯이 클라이언트를 부팅하여 동작하도록 한다.

4. COS 개발을 위한 가상 개발 환경

3절에서 기술한 COS 또는 DCAS 응용 프로그램을 개발하기 위해서는 하드웨어 키트, 소프트웨어 개발 도구, 선택적으로 디버깅을 위한 에뮬레이터가 필요하다. 그러나 응용 프로그램을 개발하기 위하여 실제 하드웨어를 사용하는 것은 다음과 같은 단점이 있다. 첫째, 잘 보호된 하드웨어 키트라도 잦은 프로그래밍 및 플래시 메모리 프로그램을 통해 장애가 생길 수 있다. 둘째, 디버깅 시에 내부 상태 또는 응용 프로그램의 자세한 동작을 확인하는데 제한이 있다. 셋째, 프로그래밍과 디버깅 과정이 하드웨어 스위칭 전환 같은 동작을 필요로 하여 시간이 많이 소요된다. 마지막으로 보통 평가 키트와 개발 도구는 가격이 비싸기 때문에 팀 개발에 필요한 충분한 환경을 구축하기 어렵다. 이러한 단점을 해소하기 위하여 시뮬레이션 기반의 가상 개발 환경을 구축하였다.

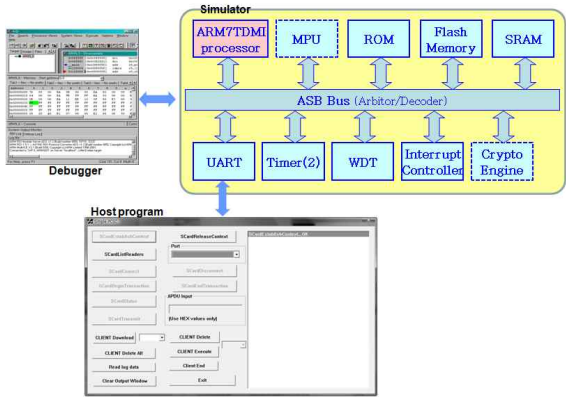

그림 3. 스마트 카드 개발을 위한 가상 개발 환경

그림 3은 ARMulator[3]를 기반으로 간단한 하드웨어 IP와 플래시, ROM, RAM 메모리 모델을 추가하여 구축한 가상 개발 환경을 도시하고 있다. ARMulator는 ARM 상의 ADS1.2에 포함되어 있으며 소프트웨어 개발과 디버깅이 시뮬레이션 환경에서 가능하다. ISO 7816 인터페이스와 호스트 에뮬레이션을 위한 호스트 프로그램을 개발하여 연결하였다. 대부분의 스마트 카드는 보안을 위한 암호화 엔진을 보조프로세서로 내장하고 있으며 MPU (Memory Protection Unit)을 가지고 있으나 현재 구성에서는 구현되지 않았다.

5. 결론

본 논문에서는 DCAS 응용을 위한 스마트 카드의 COS 설계에 대해 기술하였다. 설계된 COS는 최대 3개의 클라이언트를 지원하고 있으며 동적 로딩 및 실행이 가능한 구조를 가지고 있다. 쓰레드 간의 통신을 위한 비동기 Send, Receive 통신 프리미티브를 제공하고 있으며 호스트의 APDU 명령어를 DCASMon 쓰레드가 담당한다. COS는 스마트 카드의 자원 제약에 맞게 최소한의 기능으로 설계되었다. 설계된 COS의 구현 및 스마트 카드 응용 개발을 위한 시뮬레이션 기반의 가상 개발 환경이 구축되었다.

참고문헌

[1] W. Ranke and W. Effing, Smart Card Handbook 3rd Ed., John Wiley & Sons, 2003.

[2] Wikipedia: http://en.wikipedia.org/wiki/Downloadable_Conditional_Access_System.

[3] ARM Cop. *ARM Developer Suite 1.2 Debug Target Guide* Nov. 2001.

본 연구는 지식경제부 정보통신산업진흥원의 ITRC 사업에 지원을 받았음. (NIPA-2010-C1090-1031-0004)