

모바일 디바이스를 위한 외부 메모리 슬롯 기반의 다기종 운영체제 플랫폼

주영선
(주)여의시스템 연구소
e-mail : jys1001@yoisys.com

The Multi-operating System Platform based on External Memory Slots for Mobile Device

Young Sun Joo
Research and Development Center, Yoi System Co., Ltd.

요 약

모바일 디바이스는 PDA, 휴대전화 등 고정된 역할을 수행하는 임베디드 개념에서 스마트폰, 태블릿 PC 등 다양한 역할을 동시에 처리하는 스마트 개념으로 진화하고 있다. 이러한 변화에 따라, 모바일 디바이스는 가정용 및 일반 상업용에 국한되지 않고, 산업용 유무선 원격 컨트롤러, 디지털 정보 안내 시스템 등 산업 현장에서도 관심과 역할이 증대되고 있으며 적용분야를 점차 확대하고 있다. 기존의 모바일 디바이스는 내부 메모리에 단일한 운영체제를 탑재하여 다양한 운영체제를 사용하고자 하는 사용자의 요구를 충족시키지 못하고 다른 운영체제 사용을 위한 추가적인 가상화 프로그램 사용으로 인해 제한된 자원을 소모하게 만들며 사용자 데이터는 내부 메모리 혹은 단일한 외부 메모리에 저장하여 디바이스 분실이나 파손시에 데이터의 손실과 개인정보의 유출 등의 문제를 가지고 있다. 본 논문은 모바일 디바이스의 외부 메모리 슬롯을 증설하고 운영체제를 외부 메모리에 탑재하여 외부 메모리 교체를 통한 다기종 운영체제의 사용과 필요시 추가적인 데이터 저장 용도의 외부 메모리를 사용할 수 있는 모바일 플랫폼을 제안한다.

1. 서론

모바일 디바이스는 언제, 어디서나, 네트워크에 연결 가능한 컴퓨팅을 지향하는 유비쿼터스 컴퓨팅 개념에 현존하는 가장 적합한 디바이스이다. 모바일 디바이스는 이동하는 환경에 특화된 휴대성과 무선 네트워크가 존재하는 영역에서 네트워크에 연결하여 무선 데이터를 주고 받을 수 있는 접속성을 특징으로 한다[3]. 반복 작업과 간단한 기능을 제공하던 모바일 디바이스는 점차 복잡한 작업과 기능이 요구되면서 하드웨어 사양이 높아짐과 동시에 이를 제어하는 모바일 운영체제가 탑재되어 한정된 자원을 최적화하고 고도화된 기능을 수행할 수 있게 되었다. 다양한 기능 요구와 필요에 의해 모바일 운영체제는 다양한 종류가 개발되어 사용되고 있으며, 최근에는 스마트폰의 등장에 따라 구글의 안드로이드, 애플의 iOS 등이 선호되고 있으며[1] 풍부한 사용자 경험을 축적하고 있다.

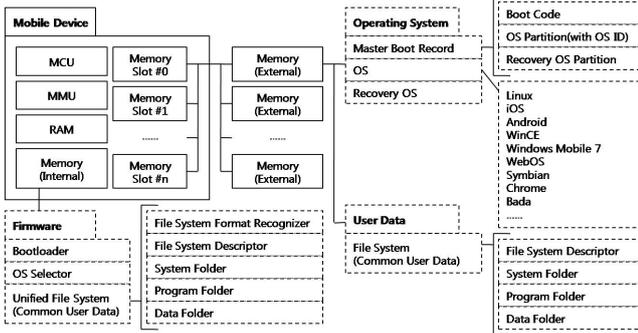
다양한 모바일 디바이스의 진화 양상과는 다르게 실제 사용에 있어서는 상당한 제약이 있다. 기존의 모바일 디바이스는 내부 메모리에 단일한 운영체제를 탑재하고 있어 다른 종류의 운영체제 사용을 위해서는 디바이스를 교체해야 하고 사용자 데이터는 별도로 백업을 받아 교체하고자 하는 디바이스의 메모리에 복사해야 하는 번거로움이 있으며, 외부 메모리 슬롯이 단일하여 확장성의 제한도 있

다. 사용자는 다양한 운영체제의 사용을 요구하고 있으며 사용자 데이터를 번거롭게 복사해서 디바이스를 교체하고자 하지 않는다. 본 논문은 이러한 기존 모바일 디바이스의 사용성의 제약을 개선하고자 외부 메모리 슬롯의 증설과 외부 메모리에 운영체제를 탑재하는 다기종 운영체제 플랫폼을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 플랫폼의 구성, 3장에서는 플랫폼의 동작원리를 설명하고, 4장에서는 결론을 기술한다.

2. 플랫폼 구성

본 논문에서 제안하는 플랫폼은 외부 메모리 슬롯을 증설하고 모바일 운영체제를 외부 메모리 슬롯에 장착된 메모리에 탑재하여 외부 메모리 교체만으로 자유롭게 운영체제를 변경할 수 있다. 사용자 데이터는 운영체제가 탑재된 외부 메모리 혹은 다른 외부 메모리에 저장할 수 있다. 플랫폼의 구성은 (그림 1)과 같으며, 실선은 물리적인 구성을 나타내고 점선은 논리적인 구성을 나타낸다.

플랫폼의 하드웨어 구성은 CPU 역할을 하는 MCU, 메모리 관리를 하는 MMU, 메인 메모리인 RAM, 내부 메모리인 Memory(Internal), 외부 메모리를 장착할 수 있는 메모리 슬롯인 Memory Slot으로 구성되며 다수의 외부 메모리 슬롯을 사용한다.



(그림 1) 플랫폼 구성도

내부 메모리는 펌웨어가 탑재되며 디바이스의 초기 부팅을 담당하는 Bootloader, 감지된 단일 혹은 다수의 운영 체제를 선택하도록 해주는 OS Selector, 통합된 파일 시스템의 관리를 제공하는 Unified File System으로 구성된다. Unified File System은 파일 시스템 관리 핸들러의 역할을 하며, 다양한 파일 시스템 포맷을 인식하여 테이블에 저장하는 File System Format Recognizer, 파일 시스템에 대한 설명이 담긴 File System Descriptor와 필요시 선택적으로 System Folder, Program Folder, Data Folder를 설정하여 사용할 수 있도록 구성한다. File System Format Recognizer는 사용자에게 의해 선택되어 구동하는 운영체제에 연결된 다양한 파일 시스템을 등록하고 디바이스 종료시 테이블에서 삭제한다. 급작스런 정전 사태에 의해 테이블에서 삭제되지 못한 파일 시스템은 다음 부팅시 체크하여 삭제하는 과정을 거쳐 시스템 안정성을 유지한다.

외부 메모리는 운영체제 혹은 데이터 저장 용도로 사용하며 탑재되는 운영체제는 모든 모바일 운영체제를 대상으로 한다. 운영체제 탑재시 디스크에 대한 정보가 담긴 MBR, OS 영역, Recovery OS 영역으로 구성된다. 데이터 저장 용도로 사용시 원하는 포맷의 파일 시스템이 탑재된다. MBR은 해당 운영체제의 부팅을 위한 Boot Code, 운영체제 식별을 위한 OS ID가 포함된 OS Partition, 복구 영역 지정을 위한 Recovery OS Partition으로 구성된다.

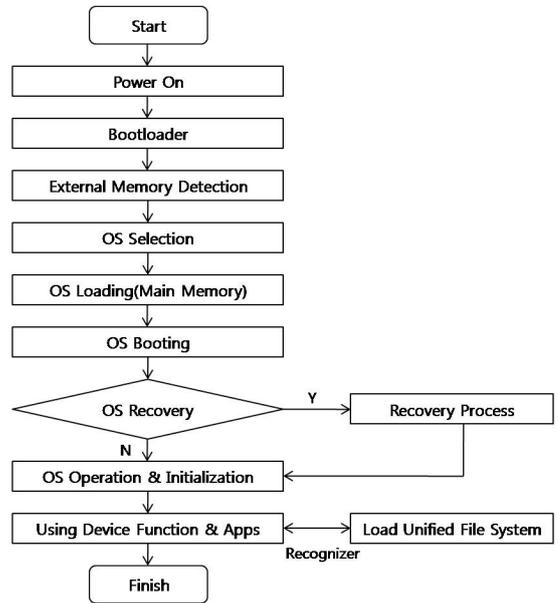
3. 동작 원리

본 장에서는 시스템의 전체 동작원리 및 주요 구성 요소의 동작원리를 설명한다.

1) 시스템

시스템의 전체 동작원리는 (그림2)와 같다. 디바이스에 전원 인가시 부트로더를 로딩하고 외부 메모리를 감지한다. 외부 메모리에 탑재된 운영체제를 선택하고 운영체제를 메인 메모리로 로딩한다. 운영체제를 부팅하면서 운영체제를 초기 상태로 복구할 것인지를 물어보고 복구를 선택하면 복구 프로세스가 진행되고 복구를 선택하지 않으면 바로 운영체제 부팅과 초기화를 진행한다. 디바이스의

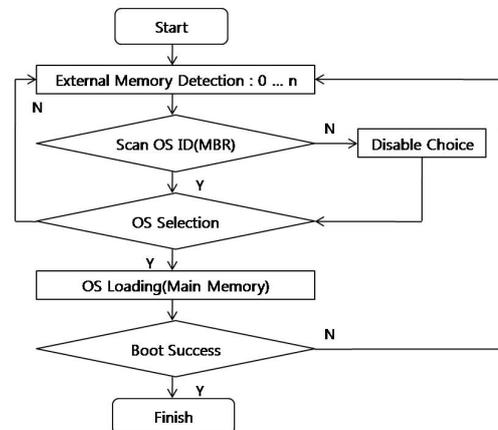
기능과 응용 프로그램을 사용하는데 운영체제가 탑재된 외부 메모리에 저장된 내용 이외에 내부 메모리나 다른 외부 메모리에 저장된 내용을 로드할 시에는 내부 메모리의 File System Format Recognizer를 사용한다.



(그림 2) 시스템 동작 원리

2) OS Selector

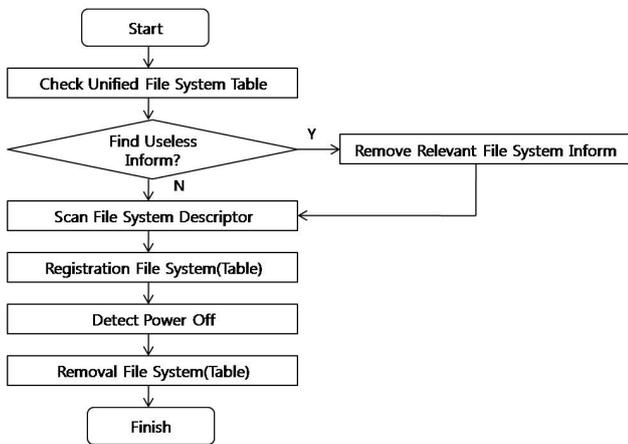
운영체제를 감지하고 선택하여 부팅하도록 역할하는 OS Selector의 동작원리는 (그림 3)과 같다. OS Selector는 복수개의 외부 메모리를 감지하고 각각의 외부 메모리마다 MBR에 포함된 OS ID를 스캔한다. 만약 스캔하여 OS ID가 없는 경우 해당 외부 메모리는 선택 불가 상태로 표시한다. OS ID가 있는 경우는 해당 운영체제를 표시하고 선택 가능으로 표시한다. 감지된 외부 메모리를 모두 스캔한 후에 사용자는 운영체제를 선택한다. OS ID를 발견하지 못했을 경우 외부 메모리를 다시 감지한다. 선택한 운영체제를 메인 메모리에 적재하여 부팅을 시도하고 부팅이 실패했을 경우에는 외부 메모리를 다시 감지한다.



(그림 3) OS Selector 동작 원리

3) File System Format Recognizer

File System Format Recognizer는 사용자에게 선택되어 구동되는 운영체제에서 사용하는 다양한 파일 시스템을 등록하여 사용하고, 디바이스 종료시에 삭제하는 과정을 통해 파일 시스템 간의 호환성을 높이는 역할을 담당하며, 동작원리는 (그림 4)와 같다. File System Format Recognizer는 사용하고 있는 File System을 등록하고 관리하는 Unified File System Table을 먼저 확인한다. 이 과정은 이전 디바이스 구동시 급작스런 정전 사태나 다른 요인에 의해 비정상적으로 종료되었을 경우에 Table에서 정상적인 삭제 과정이 이루어지지 않게 되므로, 이러한 문제를 인지하는 역할을 한다. 불필요한 파일 시스템 발견시 해당 파일 시스템 정보를 삭제한다. 다음으로 다수의 외부 메모리 내의 File System Descriptor를 스캔하여 발견된 파일 시스템을 Unified File System Table에 등록하고 디바이스 종료시에 등록했던 파일 시스템을 삭제한다.



(그림 4) File System Format Recognizer 동작 원리

4. 결론

본 논문은 모바일 디바이스를 위한 외부 메모리 슬롯 기반의 다중 운영체제 플랫폼을 제안한다. 기존의 모바일 디바이스는 단일한 운영체제 사용과 단일한 외부 메모리 슬롯 사용으로 인해 사용자가 다양한 운영체제 사용을 하기 위해서는 불가피하게 가상화 프로그램에 의존해야 한다. 이는 제한된 자원을 가진 모바일 디바이스의 비효율성을 초래하게 된다. 또한, 단일한 외부 메모리 슬롯의 사용은 사용자 데이터가 내부 메모리에 저장되거나 저장 매체의 수를 제한하게 되어 다양한 메모리에 데이터가 분산 저장될 수 있는 기회를 상대적으로 줄여, 디바이스 분실시 데이터 손실과 사용자 정보의 유출 가능성과 피해 규모가 훨씬 커질 가능성이 있다. 본 논문에서 제안한 모바일 플랫폼은 기존의 문제점을 개선하기 위해, 모바일 디바이스의 외부 메모리 슬롯을 증설하고 운영체제를 외부 메모리에 탑재하여 외부 메모리 교체를 통한 다중 운영체제의 사용과 필요시 추가적인 데이터 저장 용도로 외부 메모리를 사용할 수 있다. 플랫폼은 파일 시스템 관리 핸들러의

역할을 하는 Unified File System, 다양한 파일 시스템 포맷을 인식하여 테이블에 저장하고 관리하여 파일 시스템 간의 호환성과 시스템 안정성을 유지시켜 주는 File System Format Recognizer, 외부 메모리를 감지하고 원하는 운영체제를 선택하여 부팅할 수 있도록 해주는 OS Selector로 구성된다.

향후에는 본 논문에서 제안한 모바일 플랫폼을 클라우드 컴퓨팅과 결합하여 보안성과 시스템 자원의 효율성을 개선하겠다.

참고문헌

[1] 김도형, 류철, 이재호, 김선자, “스마트폰용 모바일 소프트웨어 플랫폼 동향”, 전자통신동향분석, Vol.25, No.3, pp.1-10, 2010.
 [2] 이광표, 이현수, 박문서, 김의준, “스마트 모바일 어플리케이션을 이용한 건설 자재 관리”, 한국건설관리학회논문집, Vol.12, No.4, pp.59-69, 2011.7.
 [3] 이호중, 라현정, 금창섭, 김수동, “서비스 기반 안드로이드 어플리케이션의 설계 및 구현 프로세스”, 정보처리학회논문지D, Vol.18D, No.4, pp.245-260, 2011.
 [4] 황재영, 정신일, 정연호, “스마트폰을 이용한 차량용 주행 모니터링 모듈 개발”, 한국정보통신학회논문지, Vol.15, No.9, pp.1903-1909, 2011.9.