

## 안드로이드 플랫폼 상의 PosFFS2를 위한 스냅샷 지원 파일 브라우저 설계 및 구현

신재복<sup>○</sup> 성백재 박찬익  
 포항공과대학교 컴퓨터공학과  
 {zstormx, jays, [cipark](mailto:cipark@postech.ac.kr)}@postech.ac.kr

### Design and Implementation of File Browser Supporting Snapshot on Android Platform with PosFFS2

Jaebok Shin<sup>○</sup> Baegjae Sung Chanik Park  
 Dept. of Computer Science and Engineering, Pohang University of Science and Technology

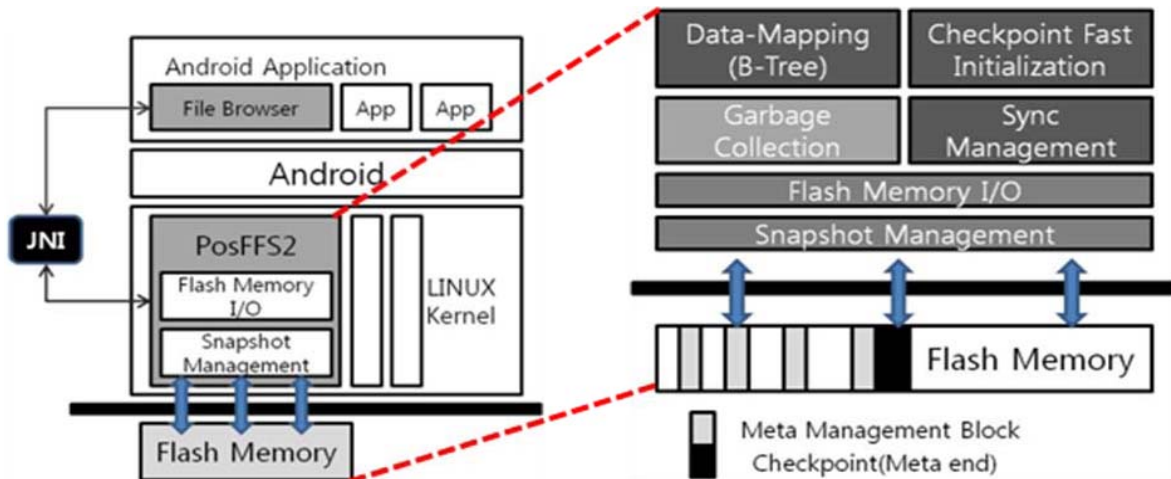
모바일 컴퓨팅 기술의 발달로 PDA, 스마트 폰 등과 같은 이동 단말기 등은 다양한 멀티미디어 데이터를 처리할 수 있도록 고성능화 되고 있으며 개인 컴퓨터 환경과 유사한 구조를 가지게 되었다. 이에 따라 빈번한 데이터의 수정과 삭제가 일어나게 되고 때로는 사용자의 실수로 인한 파일 변경 및 파일 내용의 삭제가 발생할 수 있다. 심지어 악의적인 프로그램에 의하여 시스템 파일의 변경되어 시스템에 악영향을 줄 수도 있다. 이들 여러가지 경우는 데이터 손실등 사용자 피해를 야기하므로 이를 방지하기 위한 효율적인 데이터 백업 및 복구 기법이 그 중요성을 더해가고 있다.

본 논문에서는 안드로이드를 탑재하고 저장장치로 NAND 플래시 메모리를 사용하는 모바일 기기에서의 데이터 복구와 효율적인 데이터 백업을 위하여 PosFFS2[1] 플래시 전용 파일 시스템과 안드로이드 파일 브라우저를 사용하는 프레임워크를 제안한다. PosFFS2 는 NAND 플래시 메모리의 Out-Place 업데이트 연산의 특징과 로그 구조기반(Log - Structured) 파일 시스템(LFS) 기법을 이용하여 최소한의 오버헤드를 갖는 스냅샷 지원 파일 시스템이다. 이 스냅샷 기능을 이용하여 UNDO 기능을 추가한 파일 브라우저로 사용자가 실수로 지운 데이터 파일을 복원하거나 전체적인 파일 시스템을 복구하는 것이 가능하다.

기존 로그 기반 파일 시스템은 파일 시스템 초기화 시 플래시 메모리 전체를 검색하여 파일 시스템에 필요한 초기화 정보를 재 구성하는데 시간을 소모하므로 파일 시스템 초기화에 필요한 정보를 별도로 유지하고 있는 초기화 영역에 기록하고 파일 시스템 초기화 시 이 초기화 영역만 읽도록 함으로써 초기화 시간을 단축시킬 필요가 있다[2]. PosFFS2는 이 초기화 영역을 메타 관리 블록으로 정의하고 별도로 존재하는 초기화 영역에 각 파일 매핑 정보를 할당하는 것이 아니라 데이터 영역에서 해당 블록을 할당 받아 초기화 영역으로 사용하도록 하였다. 또한 파일의 개수가 증가함에 따라 각 파일의 메타 관리 블록이 증가하여 넓은 검색 범위 때문에 초기화 성능이 저하 되는 결과를 가져온다. 따라서 PosFFS2에서는 Checkpoint 기법을 도입하여 파일시스템 해제 시 현재 플래시 메모리 상태를 메타 엔드 블록으로 할당하고 저장한다. 파일 시스템 초기화 시에 필요한 정보는 이 메타 엔드 블록을 통해서 읽어 들이고, 파일 매핑 정보는 초기화 이후 메타 관리 블록을 통해서 시스템 메모리에 로딩한다. 아래 그림의 오른쪽에 PosFFS2의 구조를 나타내었다.

PosFFS2 의 스냅샷 기법은 전체 파일 시스템 단위의 스냅샷 기법이 아닌 파일 단위 스냅샷 기법을 사용한다. PosFFS2 는 파일의 물리 주소와 논리 주소 간의 매핑을 위해 B-Tree 를 사용한 직접 매핑 구조를 사용한다. 매 파일 업데이트 연산으로 인하여 스냅샷이 발생하고 매핑을 업데이트할 때 업데이트 대상이 되는 노드에 체인 형태로 업데이트 이전 노드를 관리 함으로써 스냅샷 이미지를 관리한다. 이와 같은 구조를 사용 함으로써 하나의 매핑으로 스냅샷과 현재 이미지를 구분 할 수 있으며 업데이트 이전의 매핑을 유지하지 않아도 된다. 스냅샷의 현재 이미지와 이전 이미지를 검색하기 위해서는 트리 구조를 순회 하면서 이전 이미지와 현재 이미지를 대표하는 노드를 검색하고 이 검색한 노드를 재구성하는 방법을 사용한다.

아래 그림의 왼쪽에서 볼 수 있듯이 본 프레임워크는 데이터 저장소로서 플래시 메모리를 사용하고 안드로이드를 탑재한 모바일 아키텍처에서 플래시 변환 계층을 사용하지 않고 플래시 메모리 전용 파일 시스템으로 PosFFS2를 사용한다. 그리고 사용자 어플리케이션 계층에서는 플래시 메모리의 파일 관리를 지원하는 안드로이드 파일 브라우저를 구현하고 실행하는 구조를 나타내고 있다.



파일 브라우저는 플래시 메모리에 존재하는 폴더 및 파일을 탐색하고 파일을 실행하는 기능과 함께 파일의 삭제(Delete), 파일 업데이트를 되돌리거나 삭제된 파일의 복원(Undo)의 두 가지 추가기능을 제공한다. 이 두 가지 기능은 PosFFS2의 스냅샷 기능을 활용하여 사용자에게 효과적인 데이터 관리 방법을 지원한다. Delete 버튼을 누르면 파일 브라우저는 파일 삭제 모드로 바뀌어 이 후 선택되는 파일들은 목록에서 삭제된다. 삭제된 파일들은 Undo 버튼을 통하여 가장 최근에 삭제된 파일부터 차례대로 복원된다. 브라우저에서 Undo 기능을 실행하면 JNI를 통하여 최근에 지워진 파일에 대한 복구 시스템 콜을 호출한다..

Delete/Undo 기능은 안드로이드의 아래 계층인 리눅스 커널의 PosFFS2 파일 시스템에 의해서 동작한다. PosFFS2는 파일 시스템의 스냅샷을 파일 단위로 유지하고 있어 파일이 수정되거나 삭제되기 전의 파일 매핑 정보를 가지고 있다. 기본적으로 PosFFS2는 로그 기반 쓰기 방식을 사용하므로 지워진 파일의 내용은 플래시 메모리에 남아 있고 PosFFS2가 파일 매핑에 사용하는 B-Tree의 구조만 바뀐다. 따라서 파일 브라우저로부터 JNI를 통하여 호출된 Undo 시스템 콜을 받으면 업데이트 파일에 대해서는 매핑 트리 순회를 통하여 이전의 파일 스냅샷을 검색한 후 이전 매핑 트리로 복구하고, 삭제된 파일에 대해서는 파일에 대한 가장 최근의 매핑 트리를 저장하고 있는 메타 관리 블록의 주소를 스택에서 검색하고 valid 상태로 되돌려 파일을 사용 가능하게 한다.

삭제 및 업데이트된 파일의 메타 관리 블록의 주소는 파일 시스템 해제 시 파일시스템 초기화에 필요한 데이터를 저장하기 위해 할당하는 메타 엔드 블록에 추가한 파일 삭제 히스토리를 저장하는 히스토리 스택에 저장된다. 매 파일 삭제 및 업데이트 시 해당 파일의 매핑 트리 스냅샷에 대한 주소를 저장함으로써 JNI를 통한 Undo 시스템 콜이 발생할 때 차례대로 파일을 원래 상태로 복구할 수 있다. 이는 파일 업데이트를 되돌리기 위한 스냅샷만을 가지고 있는 PosFFS2를 삭제된 파일도 되돌릴 수 있도록 개선한 점이다.

### 참고문헌

- [1] 오용석, 박찬익 “ 임베디드 모바일 디바이스를 위한 스냅샷 지원 내장형 플래시 메모리 파일 시스템” 2009 대한임베디드공학회 추계학술대회
- [2] Junkil Ryu and Chanik Park, “ Fast Initialization and Memory Management Techniques for Log-based Flash Memory File Systems”. International Conference on Embedded Software and System 2007

### Acknowledgements

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2010-C1090-1031-0009)