

# NAND 플래시 메모리를 위한 저널링 FAT 파일시스템

강창호, 육동훈, 조상영  
한국의국어대학교 컴퓨터공학과  
e-mail:seasonyun@hufs.ac.kr

## Journaling FAT File System for NAND Flash Memory

Chang-Ho Kang, Dong-Hoon Yook, Sang-Young Cho  
Computer Science Engineering, HanKuk University of Foreign Studies

### 요 약

FAT 파일시스템은 거의 모든 운영체제에서 지원하기 때문에 임베디드 환경과 이동식 저장장치에서 NAND 플래시 메모리의 파일시스템으로 널리 사용되고 있다. 그러나 FAT 파일시스템은 하드디스크 특성에 맞게 설계되어 NAND 메모리의 특성에 맞지 않는 부분들이 있다. 이로 인한 파일 시스템의 신뢰성 문제를 해결하기 위하여 새로운 FAT 파일시스템 저널링 기능을 제안한다. 제안된 기능은 WindowsCE 환경에서 구현되었으며 동작을 다양한 크래시 환경에서 검사하였다.

### 1. 서 론

NAND 플래시 메모리는 비휘발성 메모리로서 페이지단위로만 접근할 수 있는 대신 집적도가 높고 단가가 낮기 때문에 기존의 저장매체를 대체해 나가고 있다. 현재 NAND 플래시 메모리[1]의 파일시스템은 호환성의 문제로 인해서 FAT 파일시스템이 많이 쓰여 지고 있다.

FAT 파일시스템[2]은 초기에 마이크로소프트사에서 플로피 디스크를 위한 파일시스템으로 설계되었으나 현재는 업계 표준처럼 사용되며 이러한 호환성 때문에 NAND 플래시 메모리를 쓴 임베디드 환경과 이동식 저장매체 등에서도 FAT 파일시스템을 표준처럼 사용하고 있다. 하지만 FAT 파일시스템은 NAND 플래시 메모리의 쓰기 전 소거(erase-before-write), 페이지 단위의 읽기-쓰기, 블록 단위의 지우기의 특성이 전혀 반영되어 있지 않다. 따라서 기존의 FAT 파일시스템을 쓰기 위해서 FTL(Flash Translation Layer)[3]이라는 계층을 두어 NAND 플래시 메모리와의 연동을 가능케 하고 있다.

FTL 계층은 이와 같은 변환을 위한 기능, NAND 플래시의 낮은 쓰기 수명을 보완하기 위한 wear-leveling 기능, 뿐만 아니라 NAND 플래시의 낮은 신뢰성을 보완하기 위한 페이지단위의 저널링 기능을 포함하고 있다. 이러한 FTL의 이용으로 기존의 FAT 파일시스템을 그대로 사용할 수 있다. 그러나 FTL만으로 FAT 파일시스템의 신뢰성이 보장되지는 않으며 FAT 파일시스템 단에서 파일시스템에 필요한 오퍼레이션 단위의 저널링이 필요하다. 본 논문에서는 NAND 플래시 메모리의 특성에 맞게 FAT 파일시스템의 신뢰성을 높일 수 있는 저널링 기법을 제안한다.

2장은 관련 연구에 대해 설명하고, 3장은 설계와 구현에 관한 내용을 다루고 4장은 구현 시스템의 검사에 대하여 마지막으로 5장에서 결론을 맺는다.

### 2. 관련 연구

#### 2.1 Journaling File System

저널링 기법이란 데이터베이스의 기본적인 특징들을 구현하기 위한 방법 중의 하나로서 데이터를 업데이트할 때 원본 영역에 바로 쓰는 것이 아니라 일종의 로그를 기록한 뒤 최종적으로 적용을 하거나 취소할 하는 방법이다. 이러한 저널링을 이용하면 시스템 크래시가 발생하더라도 그 위치가 어디인지 바로 파악할 수 있기 때문에 파일시스템 전체를 검사할 필요가 없어지게 되고 일관성 검사의 시간을 단시간내에 가능하게 한다.

이러한 저널링 기법을 이용한 다양한 파일시스템이 존재한다[4,5,6].

#### 2.2 FAT File System

FAT 파일시스템의 구조는 Reserved 영역, FAT 영역, Data 영역으로 구성된다[2].

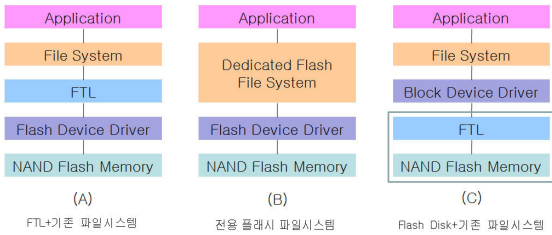
Reserved 영역은 미래를 위해 예약해 놓은 영역으로서 보통 FAT16의 경우 1섹터를 FAT32인 경우에는 32섹터를 할당하며 임의적으로 늘릴 수 있다. 이 영역에 부팅을 위한 기계어와 FAT 파일시스템의 여러 설정값들이 있는 부트 섹터(Boot Sector)가 오게 된다.

FAT 영역은 클러스터들을 관리하는 테이블이 모여 있는 공간이다. FAT 영역을 통해서 어떤 클러스터가 비어 있는지, 어떤 파일에 어떤 클러스터가 연결되어 있는지를 알 수 있게 되며 FAT1, FAT2로 두개의 클론 형태를 취하고 있다.

Data 영역은 파일 또는 디렉토리가 저장되어 있다. 이영역들은 클러스터라고 불리는 논리적인 단위로 읽기/쓰기가 된다. 클러스터는 섹터(Sector)들의 집합으로써 FAT 파일시스템 포맷시 저장장치의 용량에 의해서 결정된다.

### 2.3 플래시 메모리 관리

플래시 메모리 관리는 (그림 1)의 (A),(B),(C)의 구조[7]로 이루어 질 수 있다.



(그림 1) 플래시 메모리 관리

(A)의 경우에는 플래시 메모리 칩을 보드상에 장착하고 플래시 변환계층인 FTL을 모듈로 가지고 기존의 파일 시스템을 쓰는 경우이고, (B)의 경우에는 (A)와 같이 플래시 메모리 칩을 보드상에 장착하고 FTL과 파일시스템이 합쳐진 전용 플래시 파일 시스템이 쓰인 경우이다. (C)의 경우 FTL이 플래시 저장장치 내에 펌웨어 형태로 내장되어 기존의 파일 시스템을 쓴 경우이다.

본 논문에서는 (A), (C)의 경우 처럼 기존의 파일 시스템을 그대로 쓰게 되는 경우에 대해서 기존 FAT 파일시스템을 개발된 저널링 FAT 파일시스템으로 교체하는 방식으로 개발되었다. 물론 (A),(C)의 경우 (그림 4) 처럼 FTL단에 저널링이 있고 FAT파일시스템에도 저널링이 있기 때문에 오버헤드가 발생하지만 기존에 나와있는 FTL 모듈 혹은 FTL + NAND플래시를 그대로 사용할 수 있다는 장점을 가질 수 있다.

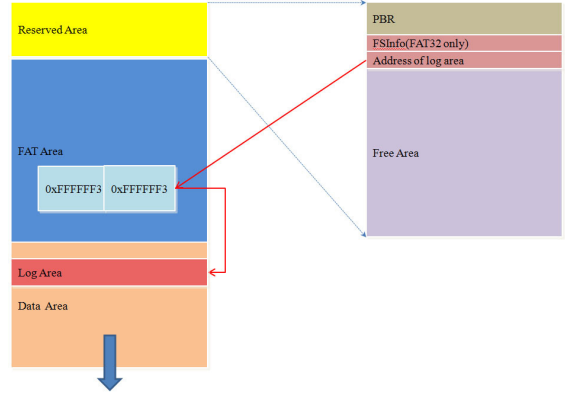
## 3. 설계 및 구현

### 3.1 로그 기록 방식

본 논문에서는 저널링 기능이 없는 기존의 FAT 파일시스템과의 호환성을 유지하면서 섹터단위의 접근을 통해 빠르게 로그를 기록할 수 있는 동적인 로그 기록 방식을 제안한다.

FAT 파일시스템은 FAT Area에 클러스터의 사용 여부를 표시하는데 FAT32의 경우 0x0000002~0xFFFFFEF의 비트가 사용하고 있는 클러스터를 나타낸다. 이를 제외한 다른 비트들은 예약이나, 불량, 파일의 마지막을 알리는 역할로 쓰인다. 따라서 본 논문에서 제안하는 로그 기록 방식은 비어있는 클러스터 영역을 FAT Area에 예약으로 표시하고 해당

영역에 로그를 기록하는 방식이다. (그림 2)는 이를 표현한 그림으로써 FAT Area에 로그 기록으로 사용할 클러스터를 0xFFFFF3으로 예약하고 해당 시작 클러스터를 Reserved Area의 FSInfo영역에 저장하여 저장장치 마운트시 로그가 기록될 클러스터를 알 수 있도록 하였다.



(그림 2) 로그 기록 구조

이 로그기록 방식은 동적으로 동작하게 되는데 기존의 저장장치에 로그영역이 없을 경우 저장장치를 스캔하여 로그를 기록할 적당한 위치를 선택하여 해당 클러스터 영역을 로그영역으로 사용되게 되고, 이미 로그영역이 있을 경우에는 해당 로그 영역에 로그를 기록하기 된다.

로그는 순환하는 구조를 가지고 있고 로그의 단위는 페이지단위로 되어 있다. 이는 로그를 기록할 때 NAND 플래시 메모리의 특성을 고려해 오버헤드를 줄여준다.

### 3.2 저널링 방식

저널링 방식에는 메타 데이터에 대해서만 저널링 하는 방식과 메타 데이터와 일반 데이터 모두에 대해서 저널링 하는 방식이 있다. 일반 데이터까지 저널링 할 경우 방대한 양의 로그가 필요하기 때문에 본 논문에서는 메타 데이터에 대해서만 저널링 하고 있다. NAND 플래시 메모리의 환경에서 일반적인 크래시 상황은 power failure이다. 메타 데이터 혹은 데이터를 기록 중에 전원이 차단되면 메타 데이터는 기록이 되었는데 실제 데이터는 기록되지 않아 일종의 손상된 파일이 만들어지게 된다. 따라서 본 논문에서는 메타 데이터를 기록 혹은 변경하기 전에 해당 오퍼레이션에 대한 로그를 남기고 해당 오퍼레이션이 정상적으로 끝났을 때 OK 로그를 남겨서 로그와 OK 로그 사이에 크래시 상황이 생겼을 경우 해당 로그를 가지고 일관성 문제를 해결해 나갈 수 있도록 하였다.

FAT 파일시스템에서 메타 데이터는 클러스터의 사용을 알려주는 FAT Area와 파일과 디렉토리를 기록하는 디렉토리 엔트리 이렇게 두 부분이다.

FAT 파일시스템의 오퍼레이션중 디렉토리 생성하는 오퍼레이션을 예로 들면 먼저 생성되는 디렉토리에 클러스터가 할당되게 된다. 따라서 FAT Area의 업데이트가 일어나게 되고 그 다음 디렉토리에 대한 엔트리 정보를 디렉토리 엔트리에 등록하게 되어서 디렉토리 엔트리의 업데이트가 일어난다.

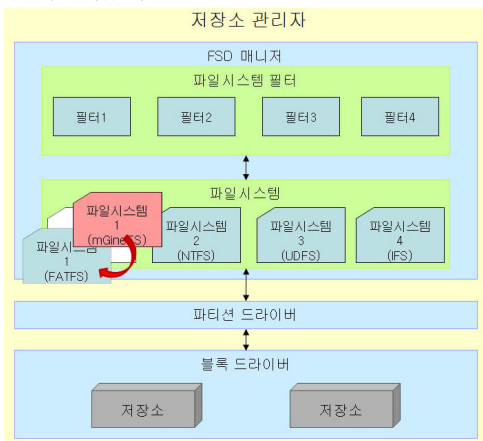
3.3 리커버리 방식

리커버리 방식에는 Redo 리커버리와 Undo 리커버리가 있다. Redo 리커버리 방식은 로그에 기록된 내용을 실제 저장장치에 적용 시키는 방식이고 Undo 리커버리는 OK 로그가 없으면 로그 기록 이전 상태로 되돌리는 방식이다.

마운트시 로그 영역을 스캔하여 로그와 OK 로그의 짝을 비교하여 짝이 맞지 않을 경우 리커버리를 수행하게 된다. 즉 OK 로그가 없고 로그만 있는 경우이다. 이 경우 기록된 로그를 보고 로그에 있는 메타 데이터 내용과 실제 저장장치의 메타 데이터를 비교하여 서로 내용이 다를 경우 Undo를 하게 되고 내용이 같을 경우에는 OK 로그를 기록한다.

3.4 구현

본 논문의 구현은 Microsoft사의 Windows CE(이하 WINCE)의 파일시스템으로 진행되었다. WINCE의 Storage 관리 구조는 아래 (그림 3)와 같다. WINCE의 Storage 관리는 FSD매니저, 파일시스템, 파티션 드라이버, 블록 드라이버 4개의 레이어로 이루어져 있다. 여기서 파일시스템 레이어는 여러종류의 파일시스템을 가지고 있고 각 시스템에 맞는 파일시스템이 동작하도록 되어 있다. FAT 파일시스템의 경우 (그림 3)의 파일시스템1 (FATFS)가 담당하고 있는데 본 논문에서는 (그림 3)처럼 FATFS대신 저널링 기능이 추가된 mGineFS로 대체하여 동작하도록 개발되었다.



(그림 3) Windows CE의 Storage 관리 구조

4. 동작의 검사

본 논문은 실제 저널링에 대한 테스트를 하기 위한 특정 위치에서의 크래시 상황을 물리적으로 만들 수가 없어서 소프트웨어 상으로 크래시가 생길 수 있는 모든 위치에 대해서 테스트를 수행하였다.

이와 같이 크래시 상황이 생길 수 있는 위치에서 소프트웨어 적으로 크래시 상황을 만들어서 구현된 시스템을 검사하였다. 검사 결과는 <표 1>과 같다.

<표 1> 크래시 테스트(디렉토리 생성)

op	critical state	short file			long file				
		in sector	new sector	new cluster	in sector	over sector	over cluster	new sector	new cluster
entry가 sector를 넘어가지 않을때	log write	undo	undo	undo				undo	undo
	FAT update	undo	undo	undo	undo			undo	undo
	...' write	undo	undo	undo	undo			undo	undo
	sector clear	undo	undo	undo	undo			undo	undo
	entry update	redo	redo	redo	redo			redo	redo
	ok log write	non	non	non	non			non	non
mkdir	log write					undo	undo		
	entry update					undo	undo		
	FAT update					undo	undo		
	...' write					undo	undo		
	sector clear					undo	undo		
	log2 write					undo	undo		
	entry2 update					redo	redo		
	FAT2 update					redo	redo		
ok log write					non	non			

5. 결론

본 논문에서는 FAT 파일시스템을 NAND 플래시 메모리의 특성에 맞게 신뢰성을 추가한 저널링 FAT 파일시스템을 제안하였다. 제안된 저널링 FAT 파일시스템은 동적으로 로그 영역을 할당하여 기존 FAT 파일시스템에 영향을 미치지 않으면서 로그에 대한 빠른 접근이 가능하도록 하였다. 또한 로그의 크기를 NAND 플래시 메모리의 페이지 단위로 맞추어 로그 기록시 생길 수 있는 오버헤드를 줄였다. 이러한 로그 기록 환경을 이용하여 FAT 파일시스템에 저널링 기능을 추가하였고 파일 시스템의 신뢰성을 높였다.

참고 문헌

[1] Samsung Electronics, "NAND Flash Memory and SmartMedia Data Book", 2006  
 [2] Microsoft, "Microsoft Extensible Firmware

Initiative FAT32 File System Specification",  
2000

[3] intel, "Understanding the Flash Translation  
Layer(FTL) Specification", 1998

[4] Red hat, Whitepaper: Red Hat's New  
Journaling  
File-System:ext3,[http://www.redhat.com/support/w  
papers/redhat/ext3/index.html](http://www.redhat.com/support/wpapers/redhat/ext3/index.html)

[5] IBM, <http://oss.software.ibm.com/jfs>

[6] D. Woodhouse, "JFFS: The Journaling Flash  
File System", Ottawa Linux Symposium 2001,  
2001

[7] 김성관, "플래시 파일 시스템 기술 소개", 정보과  
학회지 제25권 제6호, p12, 2007.