

# YAFFS 기반의 데이터 공유를 통한 대용량 멀티미디어 파일 고속 편집 저장 기법

정승완\*, 남영진\*\*, 서대화\*

\*경북대학교 전자전기컴퓨터학부

\*\*대구대학교 컴퓨터·IT공학부

e-mail:tmdrod@ee.knu.ac.kr

## A Fast Editing/Writing Technique for Large-scale Multimedia Files with Data Sharing based on YAFFS

Seung Wan Jung\*, Young Jin Nam\*\*, Dae Wha Seo\*

\*School of Electrical Eng. & Computer Science, Kyungpook National University

\*\*School of Computer & Information Technology, Daegu University

### 요 약

최근 디지털 기술의 발달로 인해 디지털 캠코더, 디지털 카메라 등의 휴대용 멀티미디어 기기가 증가하고 있고, 이러한 휴대용 장치를 위한 저장 장치로 낸드 플래시가 많이 사용된다. 이러한 장치는 고화질 미디어 콘텐츠 녹화 기능을 제공하고, 녹화된 영상물은 대용량 파일 형태로 저장된다. 대용량 파일의 필요한 부분만을 편집하여 재 저장하기 위해서는 새로운 기법이 요구된다. 본 논문은 낸드 플래시 메모리 전용 파일 시스템인 YAFFS를 사용하는 멀티미디어 휴대 장치에서 멀티미디어 파일을 고속으로 편집하고 동시에 필요한 저장 공간 소모를 최소화 하는 기법을 제안한다. 동영상 파일 편집 후 저장에 있어서 현재의 낸드 플래시 파일 시스템들은 편집 내용을 빈 공간에 새로 저장하기 때문에, 대용량의 파일일수록 많은 시간과 저장 공간이 소모 된다. 본 논문에서 제안하는 기법은 동영상 편집 후 파일 간에 중복되는 데이터를 새로 저장하지 않고 공유하도록 하면서 소모되는 시간과 공간을 최소화 한다. 이를 위해 파일 간 공유 되는 데이터를 관리하는 페이지 공유 맵을 설계하고 이를 이용한 동영상 고속 편집 저장 기법을 제시한다.

### 1. 서론

최근 컴퓨터의 이동성이 강조되면서 PDA, PMP, 디지털 캠코더, 디지털 카메라 등의 휴대용 멀티미디어 기기가 증가하고 있다. 이러한 휴대용 장치를 위한 저장 장치로는 플래시 메모리와 하드 디스크가 이용된다. 하드 디스크는 단위 용량 당 단가가 싸고, 대용량화가 가능하다. 하지만 크기가 크고, 전력 소모가 많으며, 외부 충격에 약하기 때문에 휴대용 기기의 저장 장치로는 적합하지 않다. 이로 인해, 크기와 소비 전력이 작고 외부 충격에 강한 플래시 메모리의 사용이 증가하고 있다. 플래시 메모리는 메모리 카드나 USB 메모리의 형태로 상품화되고 있다. 또한 소형의 이동 저장장치로서 뿐만 아니라 SSD(Solid State Disk) 형태로 개발되어 일반적인 컴퓨터 시스템에서 하드 디스크를 대체할 수 있는 잠재력을 가지고 있다[1].

최근 멀티미디어 장치는 SD, HD급의 고화질 미디어 콘텐츠를 보급한다. 이런 SD, HD급의 고화질 미디어 콘텐츠는 SD급 화질의 5Mbit/s, HD급 화질의 경우 15Mbit/s의 용량을 가진다[2,3]. 이 경우 저장된 멀티미디어 파일은 대용량 파일의 형태로 저장된다.

현재 멀티미디어 장치들은 PVR(Personal Video Recorder) 기능을 통해 영상을 녹화하여 저장할 수 있고 장치 내에서 재생할 수 있다. 또한, PVR은 그 기능을 확대해 녹화물에 대해 원하는 부분만을 선택적으로 저장할 수 있는 편집 기능을 제공한다. 사용자는 편집 기능을 이용하여 영상물에서 광고 등 불필요한 부분을 삭제하여 원하는 영상물을 만들 수 있다. 이와 같은 편집 기술은 미디어 콘텐츠의 유통 및 관리에 있어 필수적이며, UCC의 보편화 등으로 인해 그 필요성이 점차 증가하고 있다.

멀티미디어 파일 편집과정은 그림 1에서와 같이 일반적으로 편집 과정과 저장 과정으로 나누어질 수 있다. 편집 과정은 원본 멀티미디어 파일로부터 삭제할 부분을 선택한 후에 원본 파일에 저장할지 혹은 새로운 파일에 저장할지를 결정한다. 그림의 위쪽 부분은 원본 파일에서 C와 D부분을 삭제하고 동일한 원본 파일에 저장하는 과정을 보여주며, 아래쪽 부분은 C와 D부분을 삭제한 후에 새로운 파일에 저장하는 예를 보여주고 있다. 저장 과정은 파일 시스템을 통해 편집 과정에서 삭제되고 남은 부분만을 저장 장치에 순차적으로 저장하며, 대용량 멀티미디어 파일일 경우 일반적으로 매우 긴 저장시간을 필요로 한다. 이러한 편집 후 저장 작업을 수행할 때에 기존 파일 시스템은 효율적인 저장 기법을 제공하지 못한다.

▪ 본 연구는 2008년도 경북대학교 BK21사업과 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2008-C1090-0801-0045)

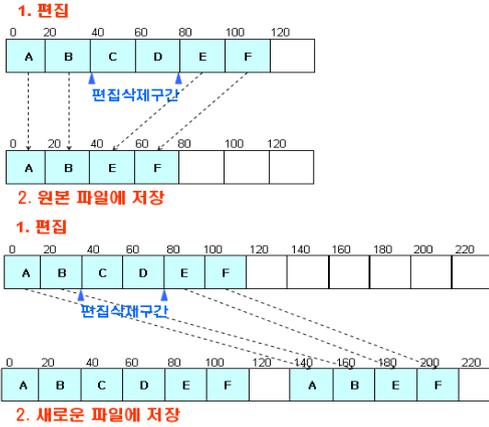


그림 1. 멀티미디어 파일 편집 과정

본 논문에서는 편집된 대용량 멀티미디어 파일을 YAFFS[2] 통하여 저장할 때 발생하는 성능상의 문제점을 극복할 수 있는 기법을 제시한다. 본 기법은 저장 시 대부분의 시간을 소모하는 데이터 I/O 시간을 줄인다. 또한 편집한 내용을 새로운 파일에 저장할 때 데이터 공유를 통해 불필요한 저장 공간 소모와 저장 시간을 줄인다. 이를 위해 메타데이터 수정을 통한 고속 저장 기법을 제시하고 공유 페이지 관리를 위한 페이지 공유 맵을 설계한다. 본 논문은 다음과 같이 구성되어 있다. 2장에서는 관련 연구를 소개하고 3장에서는 본 논문의 기본 알고리즘인 페이지 순서 재정렬 기법, 페이지 직접 사상 파일 생성 및 페이지 공유 맵 생성에 대해 설명한다. 마지막으로 4장에서는 결론 및 향후 연구에 대해 설명한다.

**2. 관련 연구**

본 논문의 기반이 되는 낸드 플래시 전용 파일 시스템인 YAFFS는 그림 2와 같이 파일 시스템 마운트 시 모든 낸드 플래시의 스페어 영역을 읽어 메인 메모리에 데이터 구조를 만든다. yaffs\_tnode는 파일의 빠른 접근을 가능하게 해주며 yaffs\_object는 디렉토리 구조를 만든다.

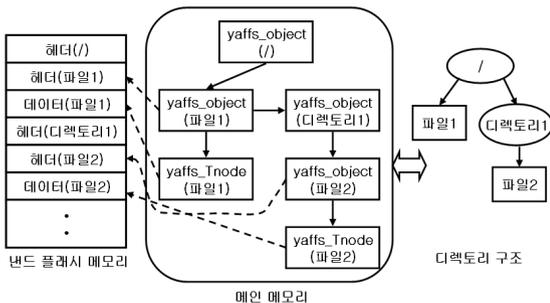


그림 2. YAFFS 메인 메모리 자료 구조

YAFFS는 메타데이터와 파일 데이터와의 연관 관계를 그림 3.(a)와 같이 역사상(inverse mapping) 방법으로 표현한다. 역사는 실제 파일 데이터가 파일의 정보를 담고 있는 헤더를 가르키도록하여 파일과 메타데이터의 관계를 표현하는 방식이다[3]. YAFFS는 낸드 플래시에 존재하는 스페어 영역에 파일에 대한 정보(파일 ID)와 파일에서 해당 데이터페이지의 논리적 순서 정보(ChunkID)를 가지고 있다. 낸드 플래시 상에서 페이지 단위 삽입, 삭제 시 역사상 방법은 파일의 모든 데이터를 새로써야 하는 불편함이 있다. YAFFS에서 사용하는 역사상과는 반대로 메타데이터에서 실제 파일 데이터를 가르키는 직접 사상이 있다. 직접 사상은 그림 3.(b)와 같이 표현되며 Ext3 파일 시스템[4]에서 사용한다. 직접 사상은 파일 정보를 담고 있는 아이노드에서 파일의 데이터 블록을 가르킨다. 직접 사상은 데이터 블록 단위 삽입, 삭제 시 전체 데이터를 새로 쓰기 않고 메타데이터 수정만을 통해 편집할 수 있는 구조를 가진다.

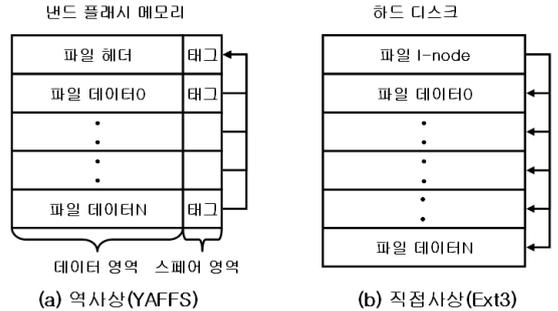


그림 3. 파일 메타데이터와 데이터간의 연관 관계

**3. 제안 기법**

제안 기법은 그림 4에서 대용량 편집 파일 고속 저장 엔진이며, YAFFS와 통합되어 동작한다. 대용량 편집 파일 고속 저장 엔진은 페이지 순서 재정렬, 페이지 직접 사상 파일 생성, 페이지 공유 맵 생성 및 관리 3가지 모듈로 구성되어 있다. 이러한 모듈들은 대용량 파일 편집 저장할 때와 파일 시스템을 마운트할 때 YAFFS랑 연계하여 동작한다.

제안 기법은 편집한 내용을 원본 파일에 덮어쓸지 새로운 파일에 저장할 지에 따라 동작 방식에 차이가 있다. 원본 파일에 덮어쓸 경우, 페이지 순서 재정렬 모듈을 실행하고, 새로운 파일에 저장할 경우, 데이터 페이지 공유를 위해 페이지 직접 사상 파일 생성 모듈을 실행한다. 각 경우에 따라 실행하는 모듈이 다르지만, 실제 데이터의 I/O 없이 메타데이터와 같은 정보를 생성 및 수정해줌으로써 빠른 편집이 가능하다는 공통점을 가진다. 페이지 공유 맵을 통해 각 페이지의 공유 횟수 정보를 관리한다.

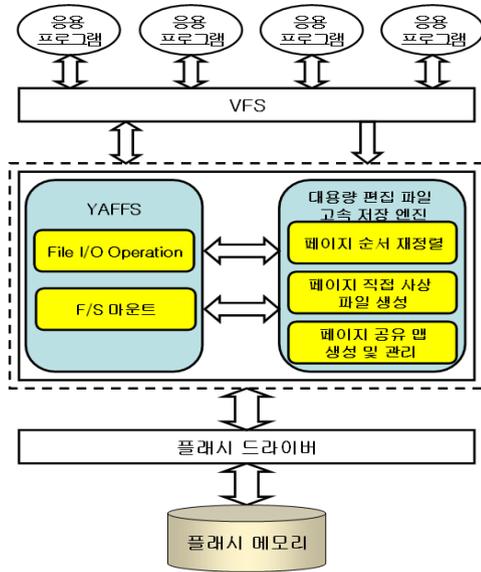


그림 4. 대용량 편집 파일을 위한 제안 기법 구조

### 3.1 페이지 순서 재정렬

YAFFS는 파일 편집 시 플래시 특성상 덮어쓰기가 불가능하기 때문에 모든 데이터 내용을 새로 쓴다. 이 때 편집하고자 하는 파일의 데이터양이 많을수록 많은 시간이 걸린다. 이와 같은 문제를 해결하기 위해 편집한 내용을 원본 파일에 덮어쓸 경우 실제 데이터 I/O 없이 빠른 편집 저장을 가능하게 하는 페이지 순서 재정렬 모듈이 필요하다. YAFFS는 플래시 메모리에 파일 헤더와 파일 데이터를 적고 메인 메모리에 빠른 파일 접근을 위해 `yaffs_tnode` 구조를 만든다. 실제 파일 접근은 `yaffs_tnode`를 통해 이루어지므로 편집된 내용에 대해 `yaffs_tnode`의 페이지 주소를 수정한다. 또한 편집으로 인해 파일의 크기와 같은 메타 정보 수정이 필요하다. 이를 위해 수정된 파

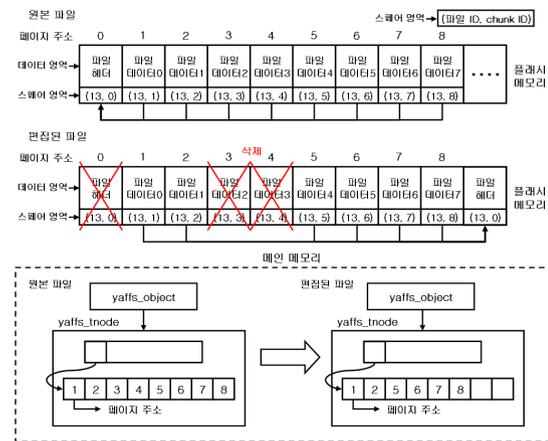


그림 5. 페이지 순서 재정렬 동작 예

일 헤더를 플래시 메모리에 새로 기록한다. 그림 5에 동작 예가 나타나있다.

페이지 순서 재정렬 작업을 하게 되면 삭제된 데이터로 인해 플래시 메모리 스페어 영역의 ChunkID 중 빠지는 것이 생긴다. 이 경우 파일 시스템 마운트 시 이를 고려해 파일 페이지 순서를 재정렬할 수 있는 작업도 필요하며, 이는 YAFFS 파일 시스템 마운트 부분을 수정한다.

### 3.2 페이지 직접 사상 파일 생성

YAFFS는 파일 편집 시 편집한 내용을 새로운 파일에 저장할 경우, 모든 내용을 플래시 메모리에 새로 저장한다. 이 때 데이터 I/O가 많이 발생하게 되므로 많은 시간이 소모된다. 이와 같은 문제를 해결하기 위해 제안 기법에서는 편집된 내용을 새로운 파일에 저장할 경우, 페이지 직접 사상 파일 생성 모듈을 실행한다.

그림 6은 편집한 내용을 새로운 파일에 저장할 때 YAFFS와 제안 기법의 동작을 나타낸다. 그림 6에서와 같이 제안 기법은 새롭게 생성된 파일의 파일 데이터 영역에 페이지 주소를 기록한다. 이 경우 기록되는 페이지 주소는 모두 원본 파일에 속하는 파일 데이터이며 공유하여 사용한다. 이와 같은 파일을 페이지 직접 사상 파일이라 한다. 새로운 파일은 기존의 YAFFS 파일 구조와 다르므로 마운트 시 구분하여 별도의 작업을 수행하기 위해 파일 헤더의 타입(type) 필드값을 특별한 값으로 정한다.

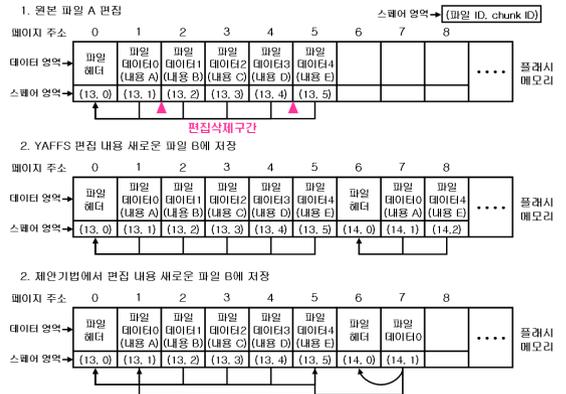


그림 6. 편집한 내용 새로운 파일 저장 시 동작 비교

### 3.3 페이지 공유 맵 생성 및 관리

제안 기법은 다수의 파일이 데이터를 공유하여 사용한다. 이 경우, 파일 삭제 시 다른 파일에서 사용하는 공유 데이터를 삭제할 수도 있기 때문에 플래시의 데이터 저장 단위인 페이지 단위로 공유 데이터 여부를 판단할 수 있는 관리 방법이 필요하다. 이를 위해 제안 기법에서는 페이지 공유 맵을 사용한다. 페이지 공유 맵의 생성 및 관리는 본 모듈에서 수행한다.

페이지 공유 맵은 그림 7과 같이 파일 시스템 마운트

시 플래시 스페어 영역을 스캔하여 메인 메모리에 생성된다. 전체 플래시 메모리 페이지의 공유 횟수 정보를 담고 있다. 페이지 공유 맵 관리는 파일 삭제 시 해당 페이지에 대해 공유 횟수 값을 1 감소시키고, 편집한 내용을 새로운 파일에 저장하기 작업을 수행하면, 새로운 파일에 속한 페이지에 대해 공유 횟수 값을 1 증가시킨다.

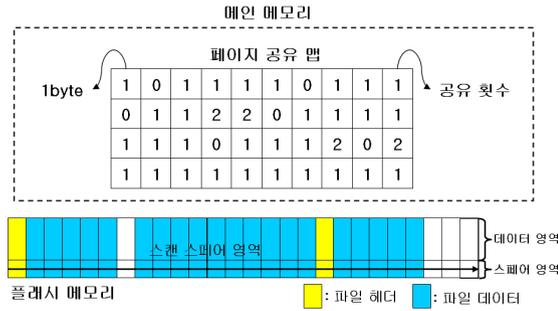
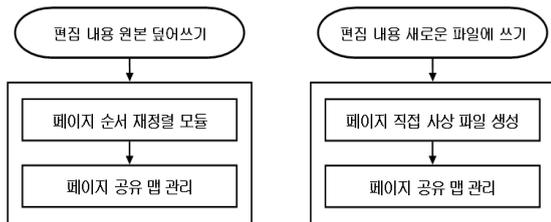


그림 7. 페이지 공유 맵 생성

### 3.4 제안 기법을 이용한 파일 저장

제안 기법을 이용하여 편집한 내용을 원본 파일에 덮어 쓸 때와 새로운 파일에 저장할 때의 동작 과정은 그림 8과 같다. 첫째, 원본 파일에 덮어쓸 경우 앞서 3.2절에서 설명한 페이지 순서 재정렬 모듈을 통해 불필요한 파일 데이터에 대해 플래시 메모리에서 해당 페이지를 더티(dirty)로 표시한다. 그리고 해당 파일의 yaffs\_tnode를 수정한다. 그리고 페이지 공유 맵 관리 모듈을 통해 삭제되는 페이지에 대해 페이지 공유 맵에서 해당 페이지 값을 1 감소시킨다.

둘째, 새로운 파일에 저장할 경우 앞서 3.3절에서 설명한 페이지 직접 사상 파일 생성 모듈을 통해 원본 파일과 파일 데이터를 공유하는 페이지 주소가 저장된 직접 사상 파일을 생성한다. 직접 사상 파일의 파일 헤더에 타입 필드는 구별되도록 특정한 값으로 한다. 그리고 직접 사상 파일이 포함하는 파일 데이터에 대해 페이지 맵에서 해당 페이지 공유 횟수를 1 증가시킨다.



(a) 원본 덮어쓰기 동작 (b) 새로운 파일에 쓰기 동작

그림 8. 파일 편집 시 동작 과정

### 3.5 토론

현재 낸드 플래시 전용 파일 시스템으로 널리 사용되는 YAFFS에서 동영상 편집저장 시 소모되는 시간을 측정하였다. 실험 환경은 리눅스 운영체제에서 520MHz ARM 프로세서, 64MB 메인 메모리 그리고 64MB 낸드 플래시 메모리를 사용했다. 16 MB 동영상 파일을 각각 3.2 MB, 8 MB, 12.8 MB에 해당하는 영역을 저장 구간으로 설정한 뒤, 편집저장을 실험하였다. 이에 걸리는 시간은 각각 5.61초, 14초, 22.43초이다. 이를 동영상 파일이 1 GB라고 가정하고 각각 200 MB, 500 MB, 800 MB를 저장 구간으로 편집한다면, 이에 걸리는 시간은 5분 36초, 14분, 22분 26초가 된다.

논문에서 제안하는 기법은 현재 위의 실험환경 상에 구현 중에 있으며, 다음과 같은 성능 예측이 가능하다. 일단 제안 기법은 편집 저장 시 대부분의 시간을 소모하는 데이터 I/O 시간을 줄일 수 있다. 페이지 순서 재정렬, 페이지 직접 사상 파일 생성과 같은 작업은 프로세서 연산을 사용하므로 소모되는 시간이 적다. 또한 데이터 페이지 공유를 통해 사용되는 공간도 줄일 수 있다.

### 4. 결론 및 향후 연구 과제

본 논문에서는 낸드 플래시 전용 파일 시스템인 YAFFS에서 대용량 파일 고속 편집 저장할 수 있는 방법에 대해 살펴보았다. 제안 기법은 파일 편집 저장 시 실제 데이터 I/O 없이 페이지 순서 재정렬 작업과 직접 사상 파일을 생성하여 데이터 페이지 공유를 가능하게 하며, 빠른 편집 저장을 제공한다. 페이지 공유를 가능하게 하기 위해 페이지 공유 맵을 생성하여 페이지의 공유 횟수를 따로 관리한다. 또한 파일 시스템 마운트 시 제안 기법으로 인해 달라지는 부분에 대해 수정된 마운트 작업을 필요로 한다. 저장 시 대부분의 시간을 소모하는 데이터 I/O가 없어지므로 많은 시간을 절약할 수 있을 것이다. 또한 페이지 공유를 통해 사용하는 저장 공간도 줄일 수 있다.

향후 연구 과제로는 본 제안 기법을 YAFFS 파일 시스템에 구현하는 것이고, 다양한 벤치마크를 사용하여 테스트해볼 예정이다.

### 참고문헌

[1] 배영현, “고성능 플래시 메모리 SSD(Solid State Disk) 설계 기술”, *정보과학회지*, July 2007.  
 [2] Aleph One Ltd, Embedded Debian, “Yaffs: A NAND-flash filesystem,” <http://www.aleph1.co.uk/yaffs>, 2002.  
 [3] 김기홍, 안성준, 최종무, 이동희, 노삼혁, “차세대 비휘발성 메모리를 이용한 플래시 파일 시스템의 개선”, *한국컴퓨터 종합 학술대회 논문집*, Vol. 33, No. 1, 2006.  
 [4] D. Bovet and M. Cesati, *Understanding the LINUX KERNEL*. 3rd edition, O’Reilly, pp. 738-774, 2006.