



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0012388  
(43) 공개일자 2016년02월03일

- |  |  |
|--|--|
| <p>(51) 국제특허분류(Int. Cl.)<br/>G06F 12/00 (2016.01)</p> <p>(21) 출원번호 10-2014-0093725</p> <p>(22) 출원일자 2014년07월24일<br/>심사청구일자 2014년07월24일</p> | <p>(71) 출원인<br/>성균관대학교산학협력단<br/>경기도 수원시 장안구 서부로 2066, 성균관대학교<br/>내 (천천동)</p> <p>(72) 발명자<br/>박대준<br/>경기도 수원시 권선구 고산로6번길 45, 102동 90<br/>3호 (고색동, 대원아파트)</p> <p>신동균<br/>서울특별시 강남구 선릉로 221, 103동 705호 (도<br/>곡동, 도곡렉슬아파트)</p> <p>(74) 대리인<br/>특허법인엠에이피에스</p> |
|--|--|

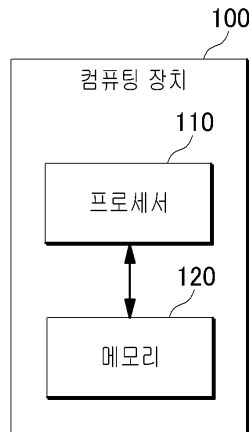
전체 청구항 수 : 총 11 항

(54) 발명의 명칭 파일 단위 순서 모드 저널링 기법을 이용한 fsync 시스템 호출 처리 장치 및 방법

(57) 요약

본 발명의 일 실시예는 컴퓨터 장치에 있어서 순서 모드 저널링을 지원하는 파일 시스템을 운영하는 프로그램이 탑재된 메모리 및 메모리에 저장된 프로그램을 운영하는 프로세서를 포함할 수 있다. 이때, 프로세서는 프로그램의 실행에 따라, 하나 이상의 파일이 포함된 트랜잭션에 대하여 순서 모드 저널링을 실행하면, 메모리의 데이터 영역에 트랜잭션을 저장하고, 메모리의 저널 영역에 트랜잭션의 메타데이터를 포함하는 일반 저널을 저장하여 저널링을 수행할 수 있다. 또한, 프로세서는 프로그램의 실행에 따라, 트랜잭션에 포함된 하나의 파일에 대한 fsync 시스템 호출이 발생하면, 파일을 메모리의 데이터 영역에 저장하고, 메모리의 저널 영역에 파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행할 수 있다.

대표도 - 도2



이 발명을 지원한 국가연구개발사업  
과제고유번호 1711006584  
부처명 미래창조과학부  
연구관리전문기관 한국산업기술평가관리원  
연구사업명 SW컴퓨팅산업원천기술개발사업  
연구과제명 스마트TV 2.0 소프트웨어 플랫폼  
기여율 1/1  
주관기관 성균관대학교 산학협력단  
연구기간 2013.12.01 ~ 2014.11.30

---

## 명세서

### 청구범위

#### 청구항 1

컴퓨팅 장치에 있어서,

순서 모드 저널링을 지원하는 파일 시스템을 운영하는 프로그램이 탑재된 메모리 및

상기 메모리에 저장된 프로그램을 운영하는 프로세서를 포함하되,

상기 프로세서는 상기 프로그램의 실행에 따라, 하나 이상의 파일이 포함된 트랜잭션에 대하여 순서 모드 저널링을 실행하면, 상기 메모리의 데이터 영역에 상기 트랜잭션을 저장하고, 상기 메모리의 저널 영역에 상기 트랜잭션의 메타데이터를 포함하는 일반 저널을 저장하여 저널링을 수행하고,

상기 프로세서는 상기 프로그램의 실행에 따라, 상기 트랜잭션에 포함된 하나의 파일에 대한 fsync 시스템 호출이 발생하면, 상기 파일을 상기 메모리의 데이터 영역에 저장하고, 상기 메모리의 저널 영역에 상기 파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행하는,

컴퓨팅 장치.

#### 청구항 2

제1항에 있어서,

상기 저널 영역의 일반 저널 및 fsync 저널은 추가 쓰기 방식으로 저장되는,

컴퓨팅 장치.

#### 청구항 3

제1항에 있어서,

상기 fsync 저널은 fsync 저널 헤더 블록을 포함하고,

상기 프로세서는 상기 프로그램의 실행에 따라, 상기 fsync 저널 헤더 블록에 상기 파일의 아이노드 정보 및 상기 파일의 이름 정보를 기록하는,

컴퓨팅 장치.

#### 청구항 4

제1항에 있어서,

상기 프로세서는 상기 프로그램의 실행에 따라, 파일 시스템을 복구하기 위하여 상기 저널 영역에 마지막 추가된 저널의 종류를 분석하고,

상기 저널의 종류가 일반 저널인 경우에는 일반 저널의 메타데이터를 이용하여 상기 파일 시스템을 복구하고,

상기 저널의 종류가 fsync 저널인 경우에는 fsync 저널의 메타데이터를 이용하여 상기 파일 시스템을 복구하는,

컴퓨팅 장치.

#### 청구항 5

제4항에 있어서,

상기 프로세서는 상기 프로그램의 실행에 따라 상기 저널의 종류가 fsync 저널인 경우,

상기 파일 시스템을 복구한 후, fsync 저널에 기록된 메타데이터를 이용하여 상기 메모리에 기록된 메타데이터를 수정하는,

컴퓨팅 장치.

**청구항 6**

파일 시스템의 순서 모드 저널링 방법에 있어서,

하나 이상의 파일이 포함된 트랜잭션에 대하여 순서모드 저널링을 실행하는 단계;

메모리의 데이터 영역에 상기 트랜잭션을 저장하는 단계; 및

상기 메모리의 저널 영역에 상기 트랜잭션의 메타데이터를 포함하는 일반 저널을 저장하여 저널링을 수행하는 단계;를 포함하되,

fsync 시스템 호출이 발생하면, 상기 fsync 시스템 호출에 따라 파일 단위로 저널링을 수행하는,

순서 모드 저널링 방법.

**청구항 7**

제 6항에 있어서,

상기 fsync 시스템 호출에 따른 파일 단위 저널링은

상기 파일을 상기 메모리의 데이터 영역에 기록하는 단계; 및

상기 메모리의 저널 영역에 상기 파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행하는 단계;를 포함하는,

순서 모드 저널링 방법.

**청구항 8**

제7항에 있어서,

상기 fsync 저널은 fsync 저널 헤더 블록을 포함하고,

상기 메모리의 저널 영역에 상기 파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행하는 단계는 상기 fsync 저널 헤더 블록에 상기 파일의 아이노드 정보 및 상기 파일의 이름 정보를 기록하는,

순서 모드 저널링 방법.

**청구항 9**

제7항에 있어서,

상기 메모리의 저널 영역은 일반 저널 및 fsync 저널을 추가 쓰기 방식으로 저장하는,

순서 모드 저널링 방법.

**청구항 10**

순서 모드 저널링 기반 파일 시스템 복구 방법에 있어서,

메모리의 저널 영역에 마지막 저장된 저널의 종류를 확인하는 단계;

상기 저널의 종류가 일반 저널인 경우에는 상기 일반 저널의 메타데이터를 이용하여 파일 시스템을 복구하는 단계; 및

상기 저널의 종류가 fsync 저널인 경우에는 fsync 저널의 메타데이터를 이용하여 파일 시스템을 복구하는 단계;를 포함하는,

파일 시스템 복구 방법.

**청구항 11**

제10항에 있어서,

상기 fsync 저널의 메타데이터를 이용하여 파일 시스템을 복구하는 단계는,

상기 파일 시스템을 복구한 이후, fsync 저널에 기록된 메타데이터를 이용하여 메모리에 기록된 메타데이터를 수정하는 단계;를 포함하는,

파일 시스템 복구 방법.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 파일 단위 순서 모드 저널링 기법을 이용한 fsync 시스템 호출 처리 방법 및 장치에 관한 것이다.

**배경 기술**

[0002] 파일 시스템(file system)은 운영 체제(operating system)에 탑재되어 데스크 톱 컴퓨터(desktop computer), 노트북 컴퓨터(notebook computer), 스마트 폰(smartphone) 및 태블릿 PC(tablet PC) 등 컴퓨팅 장치의 저장 장치(storage device)에 접근하고 파일을 읽기, 쓰기 및 관리를 수행한다. 최근 많은 컴퓨팅 장치에서 이용하는 파일 시스템은 EXT4(extended file system 4)이다. EXT4는 EXT3(extended file system 3) 파일 시스템의 향상된 버전으로 일관성(consistency)과 지속성(continuity)을 보장하는 저널링(journaling)을 제공한다.

[0003] 저널링은 컴퓨팅 장치에서 시스템 장애 또는 전원 오류 등이 발생하여 갑작스럽게 시스템이 종료되었을 때, 파일 시스템을 빠르게 복구할 수 있는 기법이다. EXT4에서 저널링은 주기적으로 저장 장치의 저널 로그(journal log)를 저장 장치의 미리 예약된 영역인 저널 영역(journal area)에 기록한다.

[0004] EXT4의 저널링 방법은 쓰기 저장 모드(write-back mode), 순서 모드(ordered mode), 데이터 모드(data mode)가 있다. 순서 모드 저널링 방법은 EXT4의 저널링 방법의 기본 옵션이다. 순서 모드 저널링 방법은 파일 시스템의 일관성을 유지하기 위하여 업데이트가 필요한 모든 데이터가 기록된 이후 메타데이터(meta-data)만 저널 영역에 기록한다.

[0005] 일반적으로 EXT4의 순서 모드 저널링 방법은 백 그라운드에서 실행되는 저널 스레드(journal thread)를 이용한다. 그러므로 저장할 데이터 양이 많아 응답 시간(response time)이 긴 파일을 저장하더라도 큰 문제가 되지 않는다. 그러나 fsync 시스템 호출(system call)은 기존 순서 모드 저널링 방법과 달리 파일 저장이 백 그라운드로 실행되지 않는다. 그러므로 저널링의 응답 시간이 중요한 문제가 될 수 있다.

[0006] Fsync 시스템 호출은 리눅스(Linux)의 시스템 호출 함수(system call function) 중 하나로 사용자가 지정한 파일의 변경 내용이 저장 장치에 확실히 기록되었는지를 보장받기 위한 함수이다. EXT4에서 fsync 시스템 호출은 저널링 스레드를 이용하여 트랜잭션(transaction) 안의 수정된 하나 이상의 데이터와 메타데이터를 메모리의 데이터 영역과 저널 영역에 기록한다. 이때, 트랜잭션은 파일 시스템의 업데이트를 모은 것이다. 즉, 트랜잭션은 가장 최근 파일 시스템이 변경된 이후 파일 연산에 따른 파일의 수정 데이터 및 메타데이터를 포함할 수 있다. 그러므로 트랜잭션에는 fsync 시스템 호출에 포함되지 않은 파일이 포함될 수도 있다. 만약, fsync 시스템 호출에서 요청하지 않은 파일의 변경한 부분이 많은 경우에는 fsync 시스템 호출의 수행 시간이 증가할 수 있다.

[0007] 이러한 문제를 해결하기 위한 기존 발명은 다음과 같다.

[0008] 한국 공개특허공보 제2005-0052016호(발명의 명칭: “파일 시스템의 메타 데이터 회복을 위한 로깅과 회복 방법 및 장치”)는 트랜잭션 단위로 메타데이터를 변경하고, 변경한 메타데이터의 로그를 전역 로그 버퍼 단위로 저장하는 발명을 개시하고 있다. 이 발명은 메타데이터 로그를 이용하여 버퍼에 적재된 메타데이터의 변경 여부를 확인하고, 변경된 경우 디스크로 저장한다.

[0009] 한국 등록특허공보 제0453228호(발명의 명칭: “공유 디스크 파일 시스템의 저널링 및 회복 방법”)는 메타데이터 로그를 이용하여 버퍼에 적재된 메타데이터의 변경 여부를 확인하고 변경된 경우 저장 장치로 저장하는 발명을 개시하고 있다.

**발명의 내용**

**해결하려는 과제**

[0010] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, 본 발명의 일 실시예는 파일 단위 순서 모드 저널링 기법을 이용하여 fsync 시스템 호출 시 수행 시간을 단축하는 방법 및 장치를 제공하는데 그 목적이 있다.

[0011] 다만, 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

**과제의 해결 수단**

[0012] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 일 실시예에 따른 컴퓨터 장치는 순서 모드 저널링을 지원하는 파일 시스템을 운영하는 프로그램이 탑재된 메모리 및 메모리에 저장된 프로그램을 운영하는 프로세서를 포함할 수 있다. 이때, 프로세서는 프로그램의 실행에 따라, 하나 이상의 파일이 포함된 트랜잭션에 대하여 순서 모드 저널링을 실행하면, 메모리의 데이터 영역에 트랜잭션을 저장하고, 메모리의 저널 영역에 트랜잭션의 메타데이터를 포함하는 일반 저널을 저장하여 저널링을 수행할 수 있다. 또한, 프로세서는 프로그램의 실행에 따라, 트랜잭션에 포함된 하나의 파일에 대한 fsync 시스템 호출이 발생하면, 파일을 메모리의 데이터 영역에 저장하고, 메모리의 저널 영역에 파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행할 수 있다.

[0013] 또한, 본 발명의 일 실시예에 따른 파일 시스템의 순서 모드 저널링 방법은 하나 이상의 파일이 포함된 트랜잭션에 대하여 순서모드 저널링을 실행하는 단계; 메모리의 데이터 영역에 트랜잭션을 저장하는 단계; 및 메모리의 저널 영역에 트랜잭션의 메타데이터를 포함하는 일반 저널을 저장하여 저널링을 수행하는 단계;를 포함할 수 있다. 이때, fsync 시스템 호출이 발생하면, fsync 시스템 호출에 따라 파일 단위로 저널링을 수행할 수 있다.

[0014] 또한, 순서 모드 저널링 기반 파일 시스템 복구 방법은 메모리의 저널 영역에 마지막 저장된 저널의 종류를 확인하는 단계; 저널의 종류가 일반 저널인 경우에는 일반 저널의 메타데이터를 이용하여 파일 시스템을 복구하는 단계; 및 저널의 종류가 fsync 저널인 경우에는 fsync 저널의 메타데이터를 이용하여 파일 시스템을 복구하는 단계;를 포함할 수 있다.

**발명의 효과**

[0015] 전술한 과제 해결 수단 중 어느 하나에 의하면, 본 발명의 일 실시예는 파일 단위 순서 모드 저널링 기법을 이용한 fsync 시스템 호출 처리 방법 및 장치를 제공할 수 있으며, 이로 말미암아 관련 사업 전반에서 다양한 파급 효과를 기대할 수 있다.

[0016] 또한, 본 발명의 일 실시예는 fsync 시스템 호출 실행 시, 파일 단위 순서 모드 저널링을 수행하므로 기존 트랜잭션 단위 순서 모드 저널링에 비하여 소요 시간을 줄일 수 있다. 본 발명의 일 실시예는 파일 시스템 복구 시 fsync 시스템 호출 이후 순서 모드 저널링을 수행하지 않은 경우에만 fsync 저널을 이용하여 복구를 수행하므로, 파일 시스템의 일관성을 유지할 수 있다. 이를 통하여 fsync 시스템 호출을 자주 사용하는 데이터 베이스 및 애플리케이션 등의 성능 저하를 방지할 수 있다.

**도면의 간단한 설명**

- [0017] 도 1은 기존 EXT4에서의 순서 모드 저널링을 이용한 fsync 시스템 호출 처리 과정을 설명하기 위한 예시도이다.
- 도 2는 본 발명의 일 실시예에 따른 컴퓨팅 장치를 개략적으로 도시한 구성도이다.
- 도 3은 본 발명의 일 실시예에 따른 일반 저널을 개략적으로 도시한 구성도이다.
- 도 4는 본 발명의 일 실시예에 따른 fsync 저널을 개략적으로 도시한 구성도이다.
- 도 5는 본 발명의 일 실시예에 따른 순서 모드 저널링 방법의 순서도이다.
- 도 6은 본 발명의 일 실시예에 따른 순서 모드 저널링을 이용한 fsync 시스템 호출 처리 방법의 순서도이다.
- 도 7은 본 발명의 일 실시예에 따른 순서 모드 저널링을 이용한 파일 시스템 복구 방법의 순서도이다.

**발명을 실시하기 위한 구체적인 내용**

[0018] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며

여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

- [0019] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0020] 파일 시스템은 컴퓨팅 장치에 연결된 저장 장치 및 데이터베이스(database) 등에 파일이나 데이터를 보관하고 관리하는 체계를 의미한다. 예를 들어, 파일 시스템은 FAT32(file allocation table 32), NTFS(new technology file system), EXT3 및 EXT4 등이 포함될 수 있다.
- [0021] 최근 많은 컴퓨팅 장치에서 이용하는 파일 시스템은 EXT4이다. EXT4는 안드로이드(android)나 리눅스(Linux) 등의 운영 체제에서 주로 사용하는 파일 시스템으로 일관성과 지속성을 보장하는 저널링을 제공한다.
- [0022] 저널링이란 주기적으로 파일 시스템의 변경 사항을 저장 장치에 기록하여 시스템의 장애 또는, 전원 오류 등에 의해 갑작스럽게 시스템이 종료되었을 때, 빠르게 파일 시스템을 복구하는 방법이다. EXT4에서 지원하는 저널링 방법은 쓰기 저장 모드, 순서 모드, 데이터 모드가 있다. EXT4에서 사용하는 대표적인 저널링 방법은 순서 모드이다.
- [0023] 순서 모드 저널링 방법은 파일 시스템의 일관성을 유지하기 위하여 트랜잭션에 포함된 업데이트가 필요한 하나 이상의 데이터를 저장 장치의 데이터 영역에 기록한 이후, 저장 장치의 저널 영역에 메타데이터를 기록한다. 이때, 기존 일반적인 순서 모드 저널링 방법은 백그라운드로 실행되기 때문에 트랜잭션에 저장하는데 소요 시간이 긴 데이터가 포함되어 있더라도 큰 문제가 되지 않는다. 그러나 기존 일반적인 순서 모드 저널링과 달리 백그라운드에서 실행되지 않는 fsync 시스템 호출을 사용하는 경우에는 저널링의 응답 시간이 중요한 문제가 된다.
- [0024] 이러한 문제를 해결하기 위하여, 본 출원인은 트랜잭션에서 fsync 시스템 호출에 해당하는 파일과 메타데이터를 추출하여 파일 단위 저널링을 수행하는 방법을 제안하였다.
- [0025] 다음은 도 1을 이용하여 기존 EXT4에서의 순서 모드 저널링을 이용한 fsync 시스템 호출 처리 과정을 설명한다.
- [0026] 도 1은 기존 EXT4에서의 순서 모드 저널링을 이용한 fsync 시스템 호출 처리 과정을 설명하기 위한 예시도이다.
- [0027] 도 1의 저장 장치는 하나의 저널 영역 및 하나의 데이터 영역(data area)을 포함하고 있다. 예를 들어, 데이터 영역은 BG1, BG2, BG3 및 BG4라고 표기된 4개의 블록 그룹으로 구성될 수 있다. 도 1의 트랜잭션은 데이터가 수정되었으나, 아직 저장 장치에 저장하지 못한 4개의 파일과 그 파일과 연관된 8개의 메타데이터가 존재한다. 메타데이터는 수정된 파일에 대한 아이노드 테이블(inode table), 블록 비트맵(block bitmap), 아이노드 비트맵(inode bitmap) 및 그룹 디스크립터 테이블(group descriptor table) 등을 포함할 수 있다. 아이노드 테이블은 파일의 정보를 기록하고, 블록 비트맵은 파일의 할당 여부를 기록하며, 블록 비트맵은 블록 그룹 내의 데이터 블록 할당 여부를 기록하는 메타데이터이다.
- [0028] 기존 일반적인 순서 모드 저널링은 파일 3에 대하여 fsync 시스템 호출이 발생하면, 저널링 스레드를 호출하여 트랜잭션에 대한 저널링을 수행한다. Fsync 시스템 호출은 DBMS(database management system)에 저장되는 데이터나 애플리케이션의 XML 설정 파일과 같이 자주 업데이트되며, 특정 시점에 저장 장치에 저장되는 것이 보장되어야 하는 파일에 사용한다. Fsync 시스템 호출은 백그라운드로 실행되지 않는다. 그러므로 fsync 시스템 호출은 트랜잭션에 존재하는 모든 파일의 수정된 데이터가 데이터 영역에 저장될 때까지 기다리게 된다. 즉, 도 1에서는 fsync 시스템 호출을 실행하는 파일 3뿐만 아니라 파일 1, 파일 2 및 파일 4의 수정된 데이터가 모두 데이터 영역에 저장될 때까지 기다리게 된다. 모든 파일의 수정된 데이터가 데이터 영역에 쓰인 이후, 트랜잭션에 존재하는 메타데이터를 저널 영역에 기록하게 된다.
- [0029] 이렇게 기존 순서 모드 저널링은 파일 3과 관계없는 파일까지 데이터 영역에 저장해야 하므로 fsync 시스템 호출의 응답 시간이 증가하게 된다. 특히, 파일 3과 관계없는 파일의 크기가 큰 경우 fsync 시스템 호출의 응답 시간은 더욱 길어지게 된다.
- [0030] 이러한 문제를 해결하기 위하여 트랜잭션에서 fsync 시스템 호출을 수행하는 파일을 분리하여 파일 단위 순서 모드 저널링을 수행할 수 있다. 이렇게 파일 단위로 순서 모드 저널링을 수행하면, 기존 순서 모드 저널링에 비하여 fsync 시스템 호출의 소요 시간을 줄일 수 있다. 그러나 트랜잭션의 메타데이터는 메타데이터의 특성상

해당 파일이 아닌 다른 파일의 파일 연산으로 변경된 결과도 반영되게 된다. 그러므로 파일 단위 순서 모드 저널링을 사용하는 파일 시스템은 불일치(inconsistency) 상태가 될 수 있다.

- [0031] 본 발명의 일 실시예는 이러한 기존 순서 모드 저널링 방법의 문제점을 해결하고 fsync 시스템 호출 실행 시 파일 시스템의 일관성을 유지하면서 fsync 시스템 호출의 응답 시간을 줄일 수 있는 파일 단위 순서 모드 저널링 장치 및 방법을 제공하고자 한다.
- [0032] 다음은 본 발명의 일 실시예에 따른 파일 단위 순서 모드 저널링을 지원하는 컴퓨터 장치를 도 2, 도 3 및 도 4를 이용하여 설명한다.
- [0033] 도 2는 본 발명의 일 실시예에 따른 컴퓨팅 장치를 개략적으로 도시한 구성도이다. 도 3은 본 발명의 일 실시예에 따른 일반 저널을 개략적으로 도시한 구성도이다. 도 4는 본 발명의 일 실시예에 따른 fsync 저널을 개략적으로 도시한 구성도이다.
- [0034] 본 발명의 일 실시예에 따른 컴퓨팅 장치(100)는 파일 단위 순서 모드 저널링을 지원하는 파일 시스템을 운영하는 프로그램이 탑재된 저장 장치인 메모리(120) 및 메모리(120)에 저장된 프로그램을 운영하는 프로세서(110)를 포함할 수 있다.
- [0035] 컴퓨팅 장치(100)는 서버(server), 워크스테이션(workstation), 데스크톱 컴퓨터 및 노트북 컴퓨터 등 일반적인 컴퓨터와 스마트 폰, 태블릿 PC 및 스마트 TV(smart TV) 등 최신 스마트 디바이스(smart device) 등을 포함할 수 있다.
- [0036] 컴퓨팅 장치(100)의 메모리(120)는 전원이 공급되지 않아도 저장된 정보를 계속 유지하는 저장 장치를 통칭하는 것이다. 예를 들어, 콤팩트 플래시(compact flash; CF) 카드, SD(secure digital) 카드, 메모리 스틱(memory stick), 솔리드 스테이트 드라이브(solid-state drive; SSD) 및 마이크로(micro) SD 카드 등과 같은 낸드 플래시 메모리(NAND flash memory)나 하드 디스크 드라이브(hard disk drive; HDD) 등과 같은 마그네틱 컴퓨터 기억 장치 등을 포함할 수 있다.
- [0037] 메모리(120)에 저장된 프로그램은 본 발명의 일 실시예가 적용된 파일 시스템을 운영하는 운영 체제로서 동작하거나, 본 발명의 일 실시예가 적용된 파일 시스템이 탑재된 애플리케이션 형태로 구성될 수 있다.
- [0038] 또한, 메모리(120)는 기존 순서 모드 저널링을 지원하는 메모리와 동일하게 데이터 영역 및 저널 영역을 포함할 수 있다. 본 발명의 일 실시예에서 저널 영역은 일반 저널 혹은 fsync 저널을 저장할 수 있다. 또한, 데이터 영역은 데이터를 저장할 수 있다.
- [0039] 프로세서(110)는 프로그램의 실행에 따라, 하나 이상의 파일이 포함된 트랜잭션에 대하여 순서 모드 저널링을 실행하면, 메모리(120)의 데이터 영역에 트랜잭션을 저장하고, 메모리(120)의 저널 영역에 트랜잭션의 메타데이터를 포함하는 일반 저널을 저장하여 저널링을 수행할 수 있다.
- [0040] 또한, 프로세서(110)는 프로그램의 실행에 따라, 트랜잭션에 포함된 하나의 파일에 대한 fsync 시스템 호출이 발생하면, 파일을 메모리(120)의 데이터 영역 저장하고, 메모리(120)의 저널 영역에 fsync파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행할 수 있다.
- [0041] 즉, 본 발명의 일 실시예에 따른 순서 모드 저널링 방법은 기존 순서 모드 저널링 방법과 동일할 수 있다. 트랜잭션에 대하여 순서 모드 저널링을 수행하게 되면, 트랜잭션에 속하는 하나 이상의 파일에 대하여 저널링을 수행할 수 있다. 그러나 본 발명의 일 실시예에 따른 fsync 시스템 호출에서의 저널링 방법은 해당하는 파일에 대해서만 저널링을 수행하므로, 기존 순서 모드 저널링 방법과 상이할 수 있다. 이렇게 본 발명의 일 실시예는 fsync 시스템 호출이 발생하면, 해당하는 파일에 대하여만 저널링을 수행하므로 기존 순서 모드 저널링 방법에 비하여 소요 시간이 적게 걸린다는 장점이 있다.
- [0042] 이때, 일반 저널은 도 3과 같이 구성될 수 있으며, fsync 저널은 도 4와 같이 구성될 수 있다. 도 3에서처럼 일반 저널은 저널 디스크립터(journal descriptor), 블록 태그(block tag), 메타데이터 및 커밋 레코드(commit record)를 포함할 수 있다. 이와 달리 도 4의 fsync 저널은 저널 디스크립터와 함께 아이노드 정보, 파일 정보 등을 포함할 수 있다. Fsync 저널에 저장되는 아이노드 정보는 아이노드 번호와 아이노드 자료구조 등이 될 수 있다. 이러한 아이노드 정보는 해당 파일의 아이노드를 복구하기 위하여 사용될 수 있다. 파일 시스템을 복구할 때, 기존 아이노드 테이블에 해당하는 아이노드 자료구조를 기록하는 방식으로 복구할 수 있다.
- [0043] 본 발명의 일 실시예에서 일반 저널 및 fsync 저널은 메모리(120)의 저널 영역에 추가 쓰기(append) 방식을 이



용하여 저장될 수 있다. 그러므로 가장 최근에 저널링을 수행한 일반 저널이나 fsync 저널이 메모리(120)의 저널 영역에 가장 마지막으로 기록될 수 있다.

[0044] 또한, 본 발명의 일 실시예에 따른 fsync 저널은 fsync 저널 헤더 블록을 포함할 수 있다. Fsync 저널 헤더 블록은 도 4의 fsync 저널 헤더 블록과 같으며, 기존 저널 디스크립터, 아이노드 정보인 아이노드 번호, 아이노드 자료구조 및 파일 정보인 부모 아이노드 번호, 파일 이름 길이, 파일 이름 등이 포함될 수 있다.

[0045] 프로세서(110)는 프로그램의 실행에 따라, fsync 저널 헤더 블록에 파일의 아이노드 정보 및 파일의 이름을 기록할 수 있다.

[0046] 도 4에서 Fsync 저널의 fsync 저널 헤더 블록의 부모 아이노드 번호와 파일 이름 길이, 파일 이름 등은 해당 파일의 이름을 복구하기 위한 정보이다. 기존 아이노드 테이블에서 부모 아이노드를 찾아, 해당 파일 이름을 디렉터리 엔트리에 갱신하는 방식으로 복구를 수행할 수 있다.

[0047] 또한, 본 발명의 일 실시예를 따른 프로세서(110)는 프로그램의 실행에 따라, 파일 시스템을 복구하기 위하여 메모리(120)의 저널 영역에 마지막 추가된 저널의 종류를 분석할 수 있다. 만약, 마지막에 추가된 저널의 종류가 일반 저널인 경우, 일반 저널의 메타데이터를 이용하여 파일 시스템을 복구할 수 있다. 반대로, 마지막 추가된 저널의 종류가 fsync 저널인 경우에는 fsync 저널의 메타데이터를 이용하여 파일 시스템을 복구할 수 있다.

[0048] fsync 시스템 호출이 실행되면, 파일 단위로 저널링을 수행하기 때문에 fsync 시스템 호출이 실행하는 파일과 같은 트랜잭션에 속하며 메타데이터를 공유하는 파일에 대한 정보가 fsync 저널에 존재하지 않을 수 있다. 그러므로 fsync 시스템 호출 실행 후 순서 모드 저널링을 수행한 경우 fsync 저널을 이용하여 복구를 수행하면, 파일 시스템은 불일치상태가 될 수 있다.

[0049] 그러나 앞서 설명한 것과 같이 저널 영역은 추가 쓰기 방식으로 기록되게 때문에 가장 최근에 저널링을 수행한 저널이 가장 마지막에 추가될 수 있다. 그러므로 마지막 추가된 저널의 종류를 분석하여 가장 최근 수행된 저널링의 종류에 맞게 복구를 수행하여 파일 시스템의 일관성을 유지할 수 있다.

[0050] 즉, 본 발명의 일 실시예는 저널 영역에 마지막 저장된 저널이 fsync 저널인 경우 fsync 시스템 호출 이후 순서 모드 저널링을 수행하지 않은 것이므로 fsync 저널을 이용하여 파일 시스템을 복구를 수행할 수 있다. 만약, 마지막 저장된 저널이 일반 저널인 경우에는 fsync 시스템 호출 이후 순서 모드 저널링을 수행한 것이므로, 일반적인 복구과정과 동일한 방법으로 파일 시스템을 복구할 수 있다.

[0051] 이때, 본 발명의 일 실시예에 따른 프로세서(110)는 프로그램의 실행에 따라 저널의 종류가 fsync 저널인 경우에는 파일 시스템을 복구한 후, fsync 저널에 기록된 메타데이터를 이용하여 메모리(120)에 저장된 메타데이터를 수정할 수 있다. 이렇게 fsync 저널을 이용한 파일 복구 후, 메모리(120)에 저장된 메타데이터를 수정하므로 파일 시스템의 일관성을 유지할 수 있다.

[0052] 본 발명의 일 실시예에 따른 컴퓨터 장치는 fsync 시스템 호출 실행 시, 파일 단위 순서 모드 저널링을 수행하므로 기존 트랜잭션 단위 순서 모드 저널링에 비하여 소요 시간을 줄일 수 있다. 또한, 본 발명 일 실시예는 파일 시스템 복구 시 fsync 시스템 호출 이후 순서 모드 저널링을 수행하지 않은 경우에만 fsync 저널을 이용하여 복구를 수행하며, fsync 저널을 이용한 파일 시스템 복구가 수행된 후 메모리(120)에 저장된 메타데이터를 수정하여 파일 시스템의 일관성을 유지할 수 있다.

[0053] 다음은 본 발명의 일 실시예에 따른 파일 단위 순서 모드 저널링 방법을 도 5과 도 6을 이용하여 설명한다.

[0054] 도 5는 본 발명의 일 실시예에 따른 순서 모드 저널링 방법의 순서도이다. 도 6은 본 발명의 일 실시예에 따른 순서 모드 저널링을 이용한 fsync 시스템 호출 처리 방법의 순서도이다.

[0055] 본 발명의 일 실시예에 따른 파일 시스템의 순서 모드 저널링 방법은 컴퓨팅 장치(100)에서 하나 이상의 파일이 포함된 트랜잭션에 대하여 순서 모드 저널링을 실행할 수 있다(S600). 순서 모드 저널링을 수행하면, 먼저 메모리(120)의 데이터 영역에 트랜잭션을 저장하고(S610), 메모리(120)의 저널 영역에 트랜잭션의 메타데이터를 포함하는 일반 저널을 저장하여 저널링을 수행할 수 있다(S620). 이때, fsync 시스템 호출이 발생하면, fsync 시스템 호출에 따라 파일 단위로 저널링을 수행할 수 있다.

[0056] 본 발명의 일 실시예에 따른 파일 시스템의 순서 모드 저널링 방법은 기존 순서 모드 저널링 방법과 동일할 수 있다. 즉, 트랜잭션에 대하여 순서 모드 저널링을 수행하게 되면, 트랜잭션에 속하는 하나 이상의 파일에 대해

여 저널링을 수행할 수 있다.

- [0057] 그러나 본 발명의 일 실시예에 따른 파일 시스템의 fsync 시스템 호출 방법은 기존 순서 모드 저널링 방법과 다를 수 있다. 본 발명의 일 실시예에 따른 파일 시스템의 fsync 시스템 호출에 따른 파일 단위 저널링은 fsync 시스템 호출이 발생하면(S700) 해당하는 파일을 메모리(120)의 데이터 영역에 기록하고(S710), 메모리(120)의 저널 영역에 파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행할 수 있다(S720). 이렇게 본 발명의 일 실시예는 해당하는 파일에 대하여만 저널링을 수행하므로 fsync 시스템 호출 수행 시, 기존 순서 모드 저널링 방법보다 소요 시간이 적게 걸린다는 장점이 있다.
- [0058] 이때, 본 발명의 일 실시예에 따른 fsync 저널은 fsync 저널 헤더 블록을 포함할 수 있다. 그러므로 메모리(120)의 저널 영역에 fsync 시스템 호출을 수행하는 파일의 메타데이터를 포함하는 fsync 저널을 저장하여 저널링을 수행하기 위하여 fsync 저널 헤더 블록에 해당 파일의 아이노드 정보 및 파일의 이름 정보를 기록할 수 있다.
- [0059] 본 발명의 일 실시예에서 일반 저널 및 fsync 저널은 앞에서 설명한 것과 같이 각각 도 3 및 도 4와 같이 구성될 수 있다. 도 3에서처럼 일반 저널은 저널 디스크립터, 블록 태그, 메타데이터 및 커밋 레코드를 포함할 수 있다. 이와 달리 도 4의 fsync 저널은 저널 디스크립터와 함께 아이노드 정보, 파일 정보 등을 포함할 수 있다. Fsync 저널에 저장되는 아이노드 정보는 아이노드 번호와 아이노드 자료구조 등이 될 수 있다. 이러한 아이노드 정보는 해당 파일의 아이노드를 복구하기 위하여 사용될 수 있다. 파일 시스템을 복구할 때, 기존 아이노드 테이블에 해당하는 아이노드 자료구조를 기록하는 방식으로 복구할 수 있다.
- [0060] 또한, 본 발명의 일 실시예에 따른 fsync 저널은 fsync 저널 헤더 블록을 포함할 수 있다. Fsync 저널 헤더 블록은 앞에서 설명한 것과 같이 도 4의 fsync 저널 헤더 블록과 같으며, 기존 저널 디스크립터, 아이노드 정보인 아이노드 번호, 아이노드 자료구조 및 파일 정보인 부모 아이노드 번호, 파일 이름 및 파일 이름 길이 등이 포함될 수 있다. 부모 아이노드 번호와 파일 이름 및 파일 이름 길이 등의 정보는 파일 복구 시 해당 파일의 이름을 복구하기 위하여 사용될 수 있다.
- [0061] 본 발명의 일 실시예에서 메모리(120)의 저널 영역은 일반 저널 및 fsync 저널을 추가 쓰기(append) 방식을 이용하여 저장할 수 있다. 그러므로 가장 최근에 저널링을 수행한 일반 저널이나 fsync 저널이 메모리(120)의 저널 영역에 가장 마지막으로 기록될 수 있다.
- [0062] 다음은 본 발명의 일 실시예에 따른 파일 단위 순서 모드 저널링을 이용한 복구 방법을 도 7을 이용하여 설명한다.
- [0063] 도 7은 본 발명의 일 실시예에 따른 순서 모드 저널링을 이용한 파일 시스템 복구 방법의 순서도이다.
- [0064] 본 발명의 일 실시예에 따른 순서 모드 저널링 기반 파일 시스템 복구 방법은 파일 시스템에 대한 복구가 발생하면(S800) 메모리(120)의 저널 영역에 마지막 저장된 저널 종류를 확인할 수 있다(S810). 만약 메모리(120)에 마지막 저장된 저널의 종류가 일반 저널인 경우, 일반 저널의 메타데이터를 이용하여 파일 시스템을 복구할 수 있다(S820). 그러나 메모리(120)에 마지막 저장된 저널의 종류가 fsync 저널인 경우에는 fsync 저널의 메타데이터를 이용하여 파일 시스템을 복구할 수 있다(S830).
- [0065] 메타데이터 fsync 시스템 호출이 실행되면, 파일 단위로 저널링을 수행하기 때문에 메모리(120)의 저널 영역에 저장된 fsync 저널에 같은 트랜잭션에 속하며 메타데이터를 공유하는 파일에 대한 정보가 존재하지 않을 수 있다. 그러므로 fsync 시스템 호출 실행 이후 순서 모드 저널링을 수행한 경우에 fsync 저널을 이용하여 복구를 수행하면, 파일 시스템은 불일치상태가 될 수 있다.
- [0066] 그러나 본 발명의 일 실시예에서 메모리(120)의 저널 영역은 일반 저널 및 fsync 저널을 추가 쓰기(append) 방식을 이용하여 저장하기 때문에 가장 최근에 저널링을 수행한 일반 저널이나 fsync 저널이 메모리(120)의 저널 영역에 가장 마지막으로 기록될 수 있다. 즉, 저널 영역에 마지막 저장된 저널의 종류를 분석하면, 가장 최근 수행된 저널링의 종류를 알 수 있다.
- [0067] 즉, 복구를 수행하기 이전에 먼저 메모리(120)의 저널 영역에 마지막 저장된 저널의 종류를 분석하여 fsync 시스템 호출 이후 순서 모드 저널링이 수행하지 않은 경우에만 fsync 저널을 이용하여 복구를 수행한다면, 파일 시스템의 일관성을 유지할 수 있다.
- [0068] 또한, 본 발명의 일 실시예는 파일 시스템의 일관성을 유지하기 위하여 fsync 저널의 메타데이터를 이용하여 파일 시스템을 복구한 후, fsync 저널에 기록된 메타데이터를 이용하여 메모리(120)에 기록된 메타데이터를 수정

할 수 있다(S840).

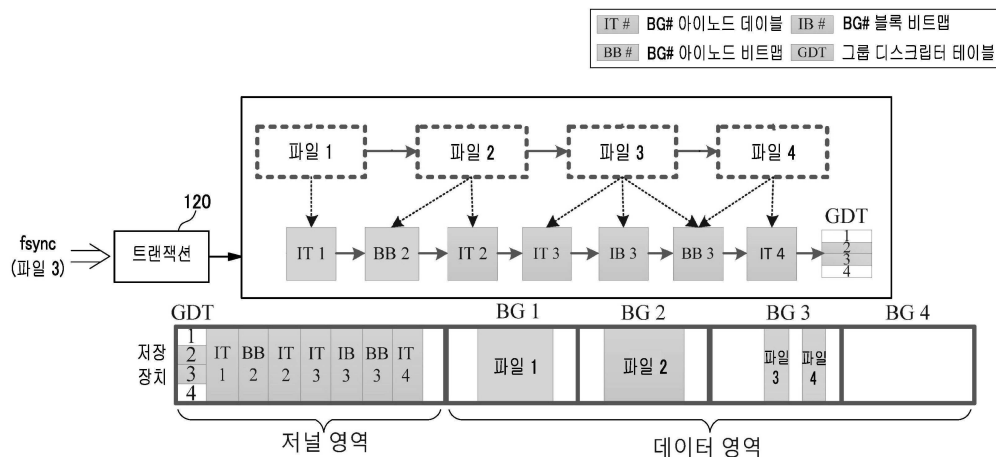
- [0069] 이렇게 본 발명의 일 실시예에 따른 파일 시스템 복구 방법은 파일 시스템 복구 시 fsync 시스템 호출 이후 순서 모드 저널링을 수행하지 않은 경우에만 fsync 저널을 이용하여 복구를 수행하며, fsync 저널을 이용한 파일 시스템 복구가 수행된 후 메모리(120)에 저장된 메타데이터를 수정하여 파일 시스템의 일관성을 유지할 수 있다.
- [0070] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행 가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스 될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독 가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 매커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.
- [0071] 본 발명의 방법 및 시스템은 특정 실시예와 관련하여 설명되었지만, 그것들의 구성 요소 또는 동작의 일부 또는 전부는 범용 하드웨어 아키텍처를 갖는 컴퓨터 시스템을 사용하여 구현될 수 있다.
- [0072] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.
- [0073] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

**부호의 설명**

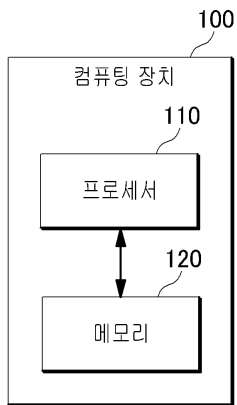
- [0074] 100: 컴퓨팅 장치
- 110: 프로세서
- 120: 메모리

**도면**

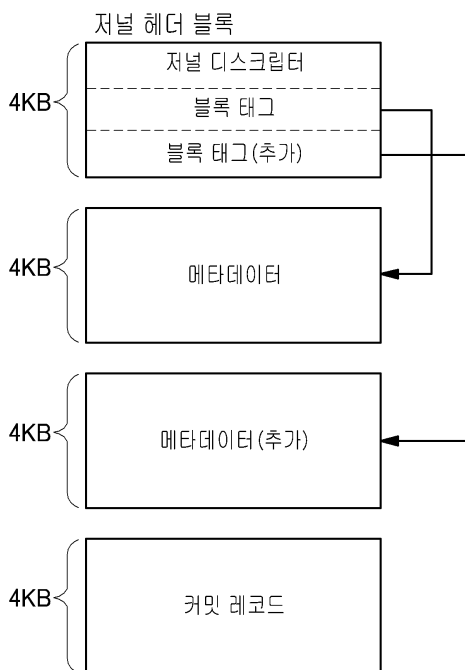
**도면1**



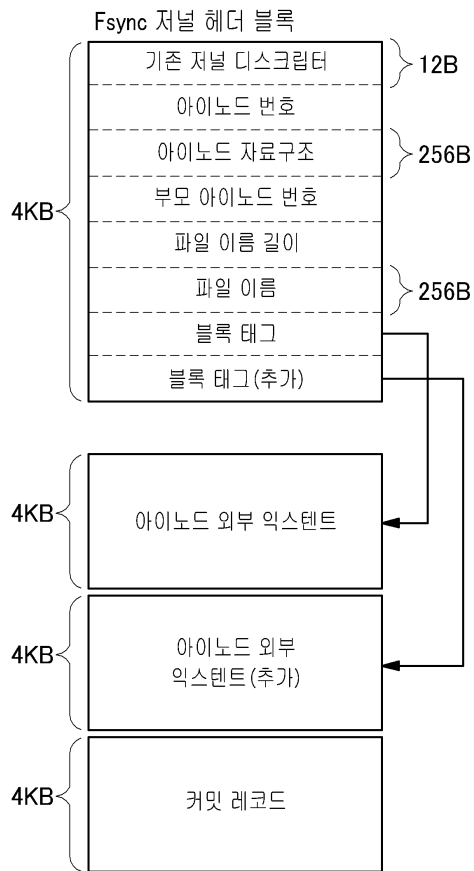
도면2



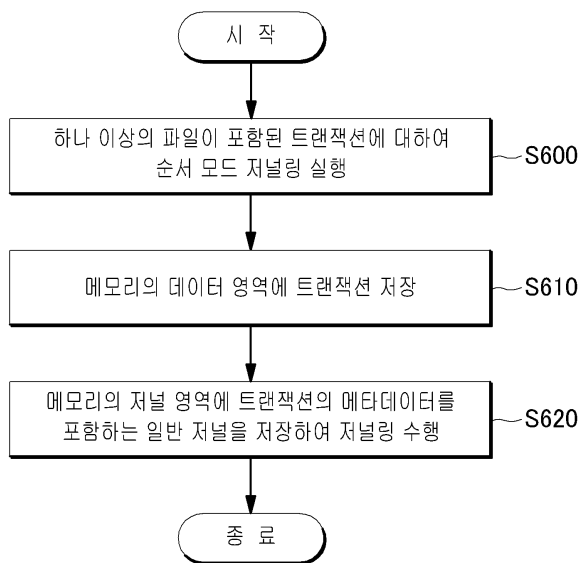
도면3



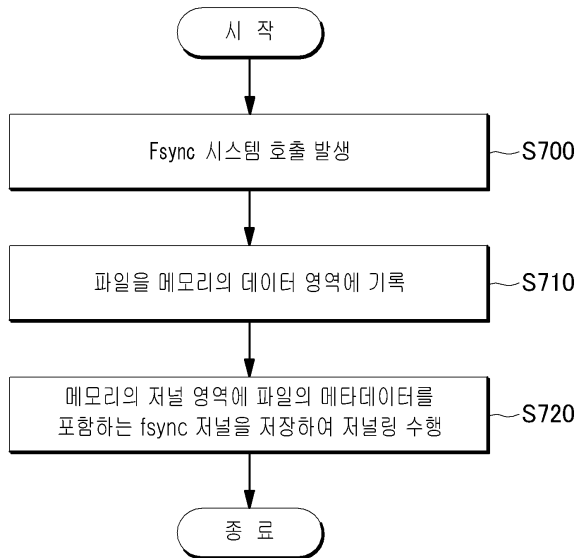
도면4



도면5



도면6



도면7

