



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2016년07월11일  
 (11) 등록번호 10-1638658  
 (24) 등록일자 2016년07월05일

- (51) 국제특허분류(Int. Cl.)  
 G06F 9/44 (2006.01) G06F 17/30 (2006.01)  
 G06F 9/445 (2006.01)
- (21) 출원번호 10-2012-7004884
- (22) 출원일자(국제) 2010년08월27일  
 심사청구일자 2015년03월13일
- (85) 번역문제출일자 2012년02월24일
- (65) 공개번호 10-2012-0104161
- (43) 공개일자 2012년09월20일
- (86) 국제출원번호 PCT/US2010/046905
- (87) 국제공개번호 WO 2011/031537  
 국제공개일자 2011년03월17일
- (30) 우선권주장  
 12/549,294 2009년08월27일 미국(US)
- (56) 선행기술조사문헌  
 US20070005661 A1\*  
 \*는 심사관에 의하여 인용된 문헌

- (73) 특허권자  
 휴렛-팩커드 디벨롭먼트 컴퍼니, 엘.피.  
 미국 텍사스주 77070 휴스턴 콤팩트 센터 드라이브  
 웨스트 11445
- (72) 발명자  
 가우라브 방가  
 미국 캘리포니아주 95035 밀피타스 머피 랜치 로드 915  
 브랜리 리차드  
 미국 캘리포니아주 95035 밀피타스 머피 랜치 로드 915  
 마드하바 딜리프 벤카타 라오  
 미국 캘리포니아주 95035 밀피타스 머피 랜치 로드 915
- (74) 대리인  
 제일특허법인

전체 청구항 수 : 총 18 항

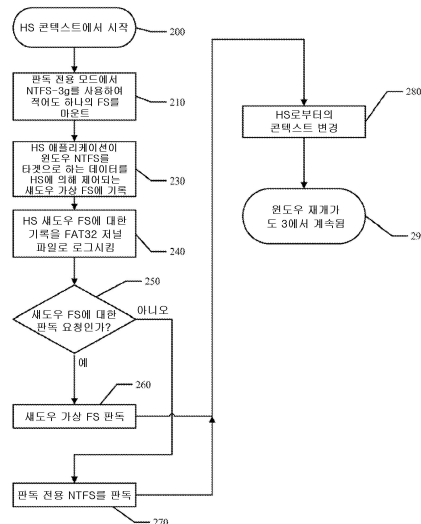
심사관 : 권현수

**(54) 발명의 명칭 이중 운영 시스템을 위한 파일 시스템**

**(57) 요약**

컴퓨팅 장치 내에서 또는 단일 컴퓨터 작동 세션 또는 콘텍스트 내에서 둘 이상의 운영 시스템(O/S) 및/또는 유사한 것들 사이의 공유 파일 시스템들을 관리, 활성화 및 제어를 위한 방법들, 시스템들, 장치들 및 프로그램 제품들이 개시된다. O/S들 중 적어도 하나가 다른 O/S의 존재를 고려하기 위한 기능들을 갖지 않는 곳에서도 파일 시스템들의 저널링 및 재동기를 위한 제공이 이루어진다.

**대표도**



## 명세서

### 청구범위

#### 청구항 1

제 1 O/S(운영 시스템)을 메모리로 로드하는 단계와,  
제 2 O/S(운영 시스템)을 상기 메모리로 로드하는 단계와,  
상기 제 2 O/S에 의해 제어되는 제 1 FS(파일 시스템)을 마운트하는 단계와,  
상기 제 1 O/S에 의한 판독 전용 액세스를 위해 상기 제 1 FS를 마운트하는 단계와,  
상기 제 1 O/S에 의해 제어되는 애플리케이션 프로그램으로부터 기록된 복수의 데이터를 상기 제 1 O/S에 의해 제어되는 저널(journal)로 재지향(redirecting)시키는 단계와,  
상기 제 2 O/S의 실행을 재개하는 단계와,  
상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS로 상기 저널을 적용하는 단계와,  
상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS로 상기 저널을 적용하는 단계의 완료를 계류(pending)시키는 상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS에 대한 트랜잭션을 연기하는 단계를 포함하는 컴퓨터 작동 방법.

#### 청구항 2

제 1 항에 있어서,  
상기 제 1 FS는 신기술 파일 시스템(New Technology File System;NTFS)인 컴퓨터 작동 방법.

#### 청구항 3

제 1 항에 있어서,  
상기 재개는 고급 구성 및 전원 인터페이스(Advanced Configuration and Power Interface;ACPI) 상태 변경에 따르는 컴퓨터 작동 방법.

#### 청구항 4

제 1 항에 있어서,  
상기 저널은 새도우 가상 파일 시스템(shadow virtual file system)에 저장된 컴퓨터 작동 방법.

#### 청구항 5

제 1 항에 있어서,  
상기 저널 및 상기 제 1 FS의 유니온으로부터, 복수의 판독 데이터를 상기 제 1 O/S에 의해 제어되는 상기 애플

리케이션 프로그램으로 재지향시키는 단계를 더 포함하는 컴퓨터 작동 방법.

#### 청구항 6

제 1 항에 있어서,  
상기 제 1 FS로 상기 저널을 적용하는 단계는,  
상기 제 2 O/S의 콘텍스트에서 상기 저널에 대해 일관성 체크 또는 유효성 체크를 수행하는 단계를 더 포함하는 컴퓨터 작동 방법.

#### 청구항 7

삭제

#### 청구항 8

제 1 항에 있어서,  
상기 저널은 제 2 FS 내에서 구성되는  
컴퓨터 작동 방법.

#### 청구항 9

제 8 항에 있어서,  
상기 제 1 FS의 종류는 NTFS이고 상기 제 2 FS의 종류는 32비트 파일 할당 테이블(File Allocation Table 32 bit; FAT32)인  
컴퓨터 작동 방법.

#### 청구항 10

제 1 항에 있어서,  
콘텍스트가 복수의 ACPI 상태 변경에 따라 상기 제 1 O/S 및 상기 제 2 O/S 사이를 계속적으로 교번하는  
컴퓨터 작동 방법.

#### 청구항 11

내부에 인코딩된 명령어를 갖는 적어도 하나의 컴퓨터 판독가능 저장 매체로서, 상기 명령어는 적어도 하나의 프로세서에 의해 실행될 때 상기 적어도 하나의 프로세서로 하여금,  
제 1 O/S(운영 시스템)을 메모리로 로드하고,  
제 2 O/S(운영 시스템)을 상기 메모리로 로드하고,  
상기 제 1 O/S에 의한 판독전용 액세스를 위해, 상기 제 2 O/S에 의해 제어되는 제 1 FS(파일 시스템)을 마운트 하고,  
상기 제 1 O/S에 의해 제어되는 애플리케이션 프로그램으로부터 기록된 데이터를 상기 제 1 O/S에 의해 제어되는 저널로 재지향시키고,

상기 제 2 O/S의 실행을 재개하고,

상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS로 상기 저널을 적용하고,

상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS로 상기 저널을 적용하는 것의 완료를 계류시키는 상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS에 대한 트랜잭션을 연기하도록 하는

컴퓨터 판독가능 저장 매체.

#### 청구항 12

제 11 항에 있어서,

상기 제 1 FS는 신기술 파일 시스템(New Technology File System;NTFS)인

컴퓨터 판독가능 저장 매체.

#### 청구항 13

제 11 항에 있어서,

상기 저널은 섀도우 가상 파일 시스템(shadow virtual file system)에 저장된

컴퓨터 판독가능 저장 매체.

#### 청구항 14

삭제

#### 청구항 15

삭제

#### 청구항 16

삭제

#### 청구항 17

컨트롤러와,

내부에 인코딩된 명령어를 갖는 컴퓨터 판독가능 저장 매체를 포함하는 전자 디바이스로서,

상기 명령어는 상기 컨트롤러에 의해 실행될 때 상기 컨트롤러로 하여금,

제 1 O/S(운영 시스템)을 메모리로 로드하고,

제 2 O/S(운영 시스템)을 상기 메모리로 로드하고,

상기 제 1 O/S에 의한 판독 전용 액세스를 위해, 상기 제 2 O/S에 의해 제어되는 제 1 FS(파일 시스템)을 마운트하고,

상기 제 1 O/S에 의해 제어되는 애플리케이션 프로그램으로부터 기록된 데이터를 상기 제 1 O/S에 의해 제어되는 저널로 재지향시키고,

상기 제 2 O/S의 실행을 재개하고,

상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS로 상기 저널을 적용하고,

상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS로 상기 저널을 적용하는 것의 완료를 계류시키는 상기 제 2 O/S의 콘텍스트에서 상기 제 1 FS에 대한 트랜잭션을 연기하도록 하는

전자 디바이스.

**청구항 18**

제 17 항에 있어서,

상기 제 1 FS는 신기술 파일 시스템(New Technology File System;NTFS)인 전자 디바이스.

**청구항 19**

제 17 항에 있어서,

상기 재개는 고급 구성 및 전원 인터페이스(Advanced Configuration and Power Interface;ACPI) 상태 변경에 따르는

전자 디바이스.

**청구항 20**

제 17 항에 있어서,

상기 저널은 새도우 가상 파일 시스템(shadow virtual file system)에 저장된 전자 디바이스.

**청구항 21**

제 11 항에 있어서,

상기 제 2 O/S의 실행을 재개하는 것은, 상기 제 2 O/S의 실행을 대기휴지 상태(quiet state)에서 활성 상태(active state)로 재개하는 것을 포함하는

컴퓨터 판독가능 저장 매체.

**청구항 22**

내부에 인코딩된 명령어를 갖는 적어도 하나의 컴퓨터 판독가능 저장 매체로서, 상기 명령어는 적어도 하나의 프로세서에 의해 실행될 때 상기 적어도 하나의 프로세서로 하여금,

제 1 운영 시스템 및 제 2 운영 시스템을 메모리로 로드하고,

상기 제 1 운영 시스템에 의한 판독전용 액세스를 위해, 상기 제 2 운영 시스템에 의해 제어되는 파일 시스템을 마운트하고,

상기 제 1 운영 시스템 의해 제어되는 애플리케이션 프로그램에 의해 기록된 데이터를 상기 제 1 운영 시스템과 연관된 저널로 재지향시키고,

상기 제 2 운영 시스템의 실행을 재개하고,

상기 제 2 운영 시스템의 콘텍스트에서 상기 파일 시스템으로 상기 저널을 적용하고,

상기 제 2 운영 시스템의 콘텍스트에서 상기 파일 시스템으로 상기 저널을 적용하는 것의 완료를 계류시키는, 상기 제 2 운영 시스템과 연관된 프로세서에 의한, 상기 파일 시스템의 파일에 대한 액세스를 연기하도록 하는

컴퓨터 판독가능 저장 매체.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 전반적으로 유사한 구조를 공유하는 개인용 컴퓨터 및 디바이스에 관한 것이고, 보다 구체적으로, 컴퓨팅 장치 내에서 또는 단일 컴퓨터 작동 세션 또는 콘텍스트 내에서 둘 이상의 운영 시스템(Operating Systems;O/S) 및/또는 이와 유사한 것 간에 공유하는 파일 시스템들을 관리, 활성화 및 제어하는 시스템 및 대응 방법에 관한 것이다.

**배경 기술**

[0002] 현대에는, 소위 랩탑과 노트북 컴퓨터를 포함한 개인용 컴퓨터(PC)의 사용이 점차 일반적이고 컴퓨터 자체는 전력과 복잡성이 증가되었으며 발열 전력은 감소되었다.

[0003] 예전 PC는 단일의, 그리고 매우 중요한, 제어 소프트웨어, 일반적으로 윈도우 운영 시스템(O/S), 예를 들어 마이크로소프트® 비스타®(상업 제품) 또는 GNOME(공개 소스 프로그램)을 사용한 리눅스®를 구비하였다.

[0004] 상업적인 윈도우® 제품 및 공개 소스 O/S 각각은 구별되는 장점을 갖고, 점차 랩탑 컴퓨터의 구매자들은 양쪽 모두를 갖고 사용하기를 원한다. 단일 컴퓨터상에서 다수의 O/S 프로그램 제품을 동시(simultaneous)에 제공하는 것은 다중 부트(multi-boot), ACPI가 지원되는 로딩 및 관련 하드웨어를 지원하는 가상 하이퍼바이저 프로그램(virtualization hypervisor programs)을 포함하여 다양한 도전과제(challenges)를 제기한다.

[0005] O/S를 거치는, 데이터 공유, 따라서 파일 시스템(File Systems;FS) 공유에 관하여 다른 이슈가 발생한다. 이는 특히 단일 운영 세션 내에서 다수의 O/S가 효과적 동시성으로 또는 계속적이고 대안적으로 작동하는 문제점이 있다. 통상적으로, PC에 대해, 운영 세션은 파워 온 셀프 테스트(Power-On Self Test;POST)부터 규칙적인 O/S 셧다운(shutdown), 및 일반적으로 파워 오프 시퀀스(power-off sequence)까지에 이른다. 다수의 엔티티들(실행 스레드(execution threads), 프로세서 코어(processor cores), 분산 프로세서(distributed processors) 등)가 데이터 파일 리소스를 효과적으로 공유할 필요가 있는 컴퓨팅에 있어서 뮤텍스 리소스 로킹(Mutexes resource locking) 및 유사 기술들은 잘 알려져 있다. 컴퓨팅에 있어서 이러한 상황은 보통 규칙적인 리소스 공유, 또는 적어도 규칙적인 리소스 할당을 보장하기 위해 협업 소프트웨어 프로시저(co-operative software procedures)에 의해 해결되었다. 하지만 (설계에 의해) 다른 O/S의 인식하지 못할 경우 비공개(closed) 소스 O/S는 다른 O/S와의 리소스 공유를 용이하게 하는 것을 기대할 수 없다.

[0006] 파일이 동시 발생적이 되거나 둘 이상의 O/S 사이 교대 가능성을 갖는 곳에서 보다 덜 일반적인 솔루션이 필요해지고 본 발명은 적어도 하나의 이러한 상황을 해결한다. 특히, 비공개 소스 O/S가 고급의 또는 복잡한 데이터 처리 기능, 예를 들어 저널라이징(journalizing) 또는 체크포인트/롤백(checkpoint/rollback)을 구현하는 곳에서, 다른 O/S에 대해 잘 동작하지 않게 될 수 있다(또는 적어도 잘 이해되는 방식으로 동작하지 않는 것). 본 발명은 특히 하나 이상의 일반적인 상황을 해결한다.

[0007] 이전의 솔루션은 각각의 O/S가 소유한 다른 O/S로부터의 판독전용(read-only;RO)으로서 간주되는 파일 시스템(File System;FS)을 완전히 제어하도록 하는 것을 포함하였다. 이는 기능에 있어서 과도하게 제한적일 뿐만 아니라, FS가 이상하게 중간 세션으로 남겨나 판독기록(Read-Write;RW)을 위해 FS를 개방하는 O/S로 문서화되지 않은 상태인 경우 일관된 결과를 생성하지 않을 수도 있다. 일부 경우에서, 이러한 FS는, 심지어 RO에 대해, RW를 액세스시키는 O/S의 콘텍스트 외부에, 탑재가능하지 않은(un-mountable)(또는 손상된) 것으로 검출될 수도 있다.

[0008] 이전에 개발된 솔루션을 능가하는 본 발명의 실시예의 중요한 장점은 단일 운영 세션 내에서 O/S 콘텍스트의 스위치 사이 파일 시스템을 셧다운 및 재시작할 필요 없이 하나 이상의 O/S 전후로 실행을 전환하도록 가능해진다 는 것이다.

**발명의 내용**

[0009] 본 발명은 컴퓨터를 작동시키기 위한 방법 및 이 방법을 구현한 장치를 또한 제공한다. 또한 본 발명을 이용하

기 위한 프로그램 제품 및 다른 수단이 제시된다.

[0010] 본 발명의 양상에 따라, 본 발명의 실시예는 제 2 O/S에 의해 제어되는 제 1 파일 시스템(FS)을 마운트(mounting)하여 두 O/S를 로드하는 것을 제공하고, 제 1 O/S에 의해 제어되는 저널(journal)을 마운트하여, 제 1 O/S에 의한 관독전용 액세스를 제공할 수 있다. 또한, 제 1 O/S 하에서 애플리케이션 프로그램으로부터 기록된 데이터인, 저널로 재지향(redirecting)시키고, 그 후 제 2 O/S의 실행을 재개하며 제 2 O/S의 콘텍스트 내에서 제 1 FS에 저널을 적용한다.

[0011] 본 발명을 구현함으로써 제공되는 또는 야기하는 장점 및/또는 기능은 두 O/S가 교대로 하나는 일시 중지되고 다른 하나는 활성화됨으로써 파일이 단일 파일 시스템 내에서 사용될 수 있다는 것이다.

**도면의 간단한 설명**

[0012] 본 발명의 기술된 관련 이점들과 특징들은 다음 도면들과 함께 취해지는 발명의 구체적인 내용을 검토할 시에 쉽게 이해되어 알 수 있을 것이며, 본원의 일부분에 통합되어 이를 구성하고, 발명의 실시예를 도시하여 숫자와 같은 것들은 구성요소와 같은 것을 나타내는 도면들에 있어서,

도 1은 본 발명에 따라 보안 기능을 구현하도록 구성된 전자 디바이스의 개략적인 블록도이다.

도 2 및 도 3은 본 발명의 실시예를 구현하도록 수행되는 일부 단계의 개요를 도시하는 흐름도이다.

도 4는 본 발명의 실시예의 소프트웨어 구성요소들 사이의 관계를 도시하는 도면이다.

도 5는 본 발명의 예시적인 실시예가 어떤 방법으로 컴퓨터 매체 또는 매체들 상에 인코딩될 수 있는지를 도시하는 도면이다.

도 6은 본 발명의 예시적인 실시예가 어떤 방법으로 전자파를 사용하여 인코딩, 전송, 수신 및 디코딩될 수 있는지를 도시하는 도면이다.

**발명을 실시하기 위한 구체적인 내용**

[0013] 이 도면에 도시된 다수의 구성요소들은 완전한 본 발명의 개시를 가능하게 하는 당업자에게 제공하기 위하여 제시된다. 본 명세서를 모호하게 하거나 또는 여기에 제공된 본 발명의 신규성 및 주요 이점들을 제거하거나 축소시키지 않도록 알려진 구성요소의 설명은 이 구체적인 내용에 포함되어있지 않다.

[0014] 이제 본 발명의 예시적 구현에는 도해들과 관련하여 설명될 것이다. 도 1은 본 발명에 따라 보안 기능을 구현하도록 구성된 전자 디바이스의 개략적인 블록 다이어그램이다.

[0015] 예시적인 구현예에서, 전자 디바이스(10)는 개인용 컴퓨터, 예를 들어 데스크탑 컴퓨터, 랩탑 컴퓨터, 태블릿 PC 또는 다른 적합한 컴퓨팅 디바이스로서 구현될 수 있다. 내용은 개인용 컴퓨터의 동작의 개요를 서술하지만, 전자 디바이스가, PDA, 무선 통신 디바이스, 예를 들어 셀룰러 전화기, 내장형 컨트롤러 또는 디바이스, 예를 들어 셋톱박스(set top boxes), 프린팅 디바이스 또는 다른 적합한 디바이스 또는 그것의 조합으로 구현되고, 본 발명으로 동작시키거나 상호운영에 적합하다는 것은 당업자에 의해 이해될 것이다.

[0016] 전자 디바이스(10)는 전자 디바이스(10)의 전체적인 동작을 제어하도록 구성된 적어도 하나의 프로세서 또는 중앙 처리 장치(CPU)(12)를 포함한다. 비슷한 컨트롤러들 또는 마이크로프로세서 장치(Microprocessor Unit;MPU)들이 일반적이다. 프로세서(12)는 일반적으로 프론트사이드 버스(Front-Side Bus;FSB)와 같은 버스(13)를 통해 노스브리지 칩(Northbridge chip)과 같은 버스 컨트롤러(14)에 연결된다. 버스 컨트롤러(14)는 일반적으로 랜덤 액세스 메모리(Random Access Memory;RAM)와 같은 관독 기록 시스템 메모리(16)를 위한 인터페이스를 제공한다.

[0017] 보통 동작 중에, CPU(12)는 시스템 메모리(16)로부터 컴퓨터 명령어(또한 컴퓨터 명령 코드로도 칭해지며 도 1에 도시되어있지 않음)를 인출한다. 그 후, CPU는 인출된 컴퓨터 명령어들을 해석하고, 그렇게 함으로써 전자 디바이스(10)의 동작을 제어하기 위해 명령어들을 해석하도록 동작한다. 일반적으로 운영 시스템(OS) 코드 및 다른 소프트웨어를 포함하는 CPU, 시스템 메모리 및 컴퓨터 명령어의 이러한 사용은 컴퓨팅 분야에서 알려져 있다.

[0018] 버스 컨트롤러(14)는 또한 시스템 버스(18), 예를 들어 일반 인텔® 방식 구현물 내 직접 매체 인터페이스(Direct Media Interface;DMI)에 연결되어있다. DMI(18)는 인텔® 입/출력 컨트롤러 허브 타입 8(ICH8)(24) 칩

과 같은 소위 사우스브리지 칩(Southbridge chip)이다.

- [0019] 일반적인 구현예에서, 사우스브리지 칩(24)은 본 발명의 실시예에서 사용되는 다양한 파일 시스템(FS)을 제공할 수 있는 디스크 스토리지 서브시스템(66)에 번갈아 연결되는 퍼리퍼럴 컴포넌트 인터커넥트(Peripheral Component Interconnect; PCI) 버스(22)에 연결된다. 이것으로 도 1의 논의를 마친다.
- [0020] 피닉스® 테크놀로지스 리미티드의 하이퍼스페이스™ 제품군은 다양한 방식으로 단일 컴퓨터 상에 다수의 O/S 프로그램 제품 제공을 가능하게 한다. 하이퍼스페이스™ 제품은, 특히 이중 부트(Dual Boot), 이중 재개(Dual Resume), 또는 가상 머신(Virtual Machines) - 후자는 오늘날 랩탑 컴퓨터 제품에서 보편적으로 발견되는 것이 아닌 하드웨어 지원을 필요로 함 - 의 기술들을 사용하여 여러 변형들로 출시된다. 하이퍼스페이스™ 제품의 이중 부트 버전은 동시 로드를 제공하지 않지만 제 2 O/S가 로드되고 PC를 제어하는 것이 가능하게 되기 이전에 제 1 O/S가 섯다운되는 것이 필요하다.
- [0021] 하이퍼스페이스™ 제품에 의해 지향된 바와 같이, 하이퍼스페이스™ 제품의 이중 개시 버전은, 예를 들어, 간소화된(streamlined) 리눅스® 유사 O/S 및 마이크로소프트® 코포레이션의 제품 중 하나와 같은 비공개 소스의 풍부한 기능을 갖춘 O/S인, 제 1 O/S 및 제 2 O/S의 동시 로드를 용이하게 한다. 기본적으로, 하이퍼스페이스™ 제품의 이중 개시 버전은, 두 O/S가 동시에 로드되지만, 임의의 순간에 사용자 레벨에서 활성화되도록 오직 하나만 섯택된다. O/S 사이의 스위칭은, 명칭이 제시하는 바와 같이, 하나의 O/S를 대기 휴지(quiescent) 상태로 되게 하고 다른 O/S를 이의 이전 대기 휴지 상태에서부터 더 활성화된 상태로 "재개"(복귀)시키는 것을 포함한다. 따라서, 본 발명의 설명된 실시예에서, 오직 하나의 O/S는 언제든지 애플리케이션 레벨에서 활성화된다.
- [0022] 하이퍼스페이스™ 제품의 이중 재개 버전은 두 대상 O/S 사이의 제한된 파일 공유를 제공한다. 하이퍼스페이스™ 제품의 리눅스 유사 O/S 측은 일반적으로 단지HS(하이퍼스페이스™의 축약)로 지칭되고, 반면에, 마이크로소프트® 코포레이션의 임의의 현대 O/S 제품 중 섯택된 하나는 컴퓨팅 기술분야에서 일반적으로 사용되는 총칭으로서 "윈도우"로 일반적으로 칭해진다. 그렇지만 윈도우®는 또한 마이크로소프트® 코포레이션의 승인된 등록 마크이며, MIT(메사추세츠 공과 대학)의 엑스-윈도우(X-Windows)는 공개 소스 프로그램 제품이다.
- [0023] 다양한 하이퍼스페이스™ 펌웨어/소프트웨어 제품의 주요 기능은 당연히 FS(그러므로 따라서 파일)가 윈도우® 콘텍스트와 동시 HS(하이퍼스페이스™의 리눅스 유사 O/S) 콘텍스트 사이에서 공유되는 것을 가능하게 한다. 특정 실시예의 다음의 설명에서, NTFS, FAT32, 윈도우 PSA, ext3(및 본 기술분야에서 알려진 모든 것 등)와 같이, 구현되어진 특정 파일 시스템은 예시적인 FS의 가능한 설명을 제공하는 것으로서 간주되어야한다. 본 발명은 일반적으로 설명된 이들 특정 O/S에 제한되지 않고, 이들은 단지 전형적인 예시로서 사용되는 것이다.
- [0024] 베어본(bare-bones) 및 (보통의 리눅스® 배포로부터 차용된) 거의 허용가능하며 기본적인 실시예는 하이퍼스페이스™ 콘텍스트 내에서부터, C:, D: 등으로 지정된 스토리지 볼륨을 마운트하고 액세스하기 위해 수정되지 않은 신기술 파일 시스템-3세대(New Technology File System, third generation; NTFS-3g) 드라이버 사용을 이용한다. 이중 부트 기술을 사용하여, 이 유형의 실시예가, 윈도우에 의해 생성된(그리고 유지되는) FS의 (리눅스 유사 O/S로부터의) 안전한 판독가능 액세스를 가능하게 하며 이는 거의 허용가능하다. 이중 재개 제품에서 이러한 제한은 대부분의 사용자들에게 전적으로 허용가능하지는 않을 것이다.
- [0025] 일반적으로, FS에 기록하는 것은 NTFS-3g 드라이버에 의해 지원되지만, 이 기능을 사용하는 추가 예방의 그리고 교정의 수단 없이는 MS NNTFS VSS(마이크로소프트® 인코포레이트의 볼륨 스냅샷 서비스의 NTFS) 및 데이터 저널라이징 기능과 같은 고급 기능의 NTFS-3g에 의한 불완전 구현으로부터 야기하는 문제점으로 이어질 수 있다(그리고 이어질 것 같다). MS NTFS VSS는 본 기술분야에서 잘 알려져있다.
- [0026] 상기의 어려움을 해결하기 위한 기회를 취하는 부분은 대상 볼륨의 제한된 부분 집합에만 완전한 판독-기록 액세스를 제공하기 위해서 NTFS-3g 내에서의 판독기록 액세스를 제한하는 것을 포함할 수 있다. 그렇게 함으로써 복잡성의 범위를 제한하고, 반면 이러한 파일에게 완전한 NTFS-3g 액세스를 제공하는 것은 판독-기록 액세스가 사용자에게 가치있는 것이라고 대부분 여겨진다. 이 방식으로 범위를 제한하는 것은 소프트웨어가 최소한으로 유지되는 것을 가능하게 하고 따라서 초과 리소스를 소비하는 것의 위험을 회피한다.
- [0027] 본 발명의 실시예에서 (리눅스®의 변형을 실행시키는) HS 측으로부터의 윈도우® 파일의 판독기록 액세스는 FS의 작은, 제어된 일부분에 제한된다. 일 특정 실시예에서 "내 문서 폴더"로의 판독-기록 액세스가 가능하다. "내 문서 폴더"는 마이크로소프트® 소프트웨어 제품의 일반적이고 잘 알려진 기능이다.
- [0028] 문제를 야기하지 않고 모든 파일에 대해 판독전용 액세스가 일반적으로 제공될 수 있지만, 제 2 O/S로 제어를 전달할 때 제 1 O/S의 FS가 정리된 상태가 되도록 보장하기 위해 단계가 취해지는 것이 필요할 수 있다. 설명된



실시예들은 O/S로부터 FS(특히 NTFS)에 액세스하는 것과 관련된 문제를 완화시키며 반면 다른 O/S는 중지되기는 하지만 애플리케이션 레벨에서 계류(pending)되는 동작을 가질 수 있다. 이들 효과는 당업자에게 명백할 것이다.

- [0029] 따라서, 예시적인 하이퍼스페이스™ 이중 재개 프로그램 제품에서, 파일 공유의 문제점은 도전과제이다. 애플리케이션 레벨 프로그램을 동시에 라기보다는 이와 다르게 실행하지 않는다 해도 윈도우 및 HS가 모두 활성화되는 상황이 발생한다. 이는 예를 들어 하나의 O/S가 고급 구성 및 전원 인터페이스(Advanced Configuration and Power Interface;ACPI) 상태 S0를 참조하고 다른 하나는 스스로 상태 S3에서 지연되는 것을 고려할 때 발생할 수 있다. 이 경우에 양쪽 모두는 동시에 마운트된 윈도우 소유 NTFS 파일 시스템을 구비할 수 있고, 그동안 마운트하는 OS 중 하나는 효과적으로 지연될 수 있다. 일관성을 보장하도록 취해진 특정 수단이 없이, 예를 들어 윈도우®가 지연되는 기간 동안 HS로부터의 NTFS에 대한 기록이 있을 때, 판독되는 스테일 데이터(stale date) 및/또는 손상된 윈도우 파일 시스템의 위험이 발생한다.
- [0030] 특히, 다음의 이슈가 핵심적이며 본 발명의 실시예에서 전체 또는 상당한 부분에서 해결된다.
- [0031] (A) 파일 시스템 또는 볼륨 내에서의 파일의 '온 디스크(on disk)' 버전은 '아직 저장되지 않은' 또는 동일 파일의 메모리 버전에서 캐싱된(cached) 것과는 상이할 수 있다. 이는 이분된(bifurcated) 이슈 - 둘 중 하나 또는 임의의 O/S는 파일 또는 FS 볼륨을 소유하고 제어할 수 있고 따라서 해결해야 할 다른 O/S에 대한 이슈를 생성할 수 있음 - 이다.
- [0032] (B) NTFS-3g 드라이버는 완전하게 탈착(dismounted)되지 않은 NTFS 볼륨을 마운트하는 것을 거부할 수 있다.
- [0033] (C) HS 이중 재개 유형 제품을 사용하여, 윈도우 파일은 HS에 대한 콘텍스트를 스위칭하는 시간에 편집을 위해 열린 상태로 유지될 수 있다.
- [0034] (D) 윈도우 기반 NTFS 구현은 NTFS-3g에 부재하는 향상된 기능을 포함할 수 있다.
- [0035] (I) 포인트 및 VSS 재저장, (II) 파일 인덱싱, (III) 오디오 시각(Audio-visual;AV) 패키지에 의한 필터 드라이버의 사용, 및 (IV) 재저장 기능(facilities)을 사용한 백업.
- [0036] 도 2를 참조하면, 본 발명의 실시예는, 예를 들어 리눅스® 기반 HS, O/S의 콘텍스트에서, 참조부호 200에서 시작하는 것으로서 도시되고, 다수의 동작이 취해진다. 참조부호 210에서, 판독전용 모드로 HS는 NTFS-3g를 사용하여 FS를 마운트한다. 이는 NTFS-3g 소프트웨어 서브시스템에 의해 마운트되도록 선택된 볼륨에서 하나 이상의 (임의의 또는 모든) 윈도우 기반 FS에 대해 HS로부터의 액세스를 제공할 수 있다. 또한, 윈도우 O/S는 컴퓨터 시스템에서 활성 하이퍼스페이스™ 콘텍스트 및 O/S의 존재에 의해 영향을 받지 않는, 통상적인 방식으로 다양한 FS에 기록할 수 있다.
- [0037] 참조부호 230에서, 윈도우®에 속하는 임의의 NTFS 내에서 HS 애플리케이션으로부터 파일로 기록된 임의의 데이터 및 모든 데이터는 HS에 의해 제어되고 소유된 새도우 가상(Shadow Virtual) FS에 기록된다. 새도우 가상 FS에 대한 공간의 볼륨(디스크) 할당은 HS를 근원으로 하는 FS 내에서부터 제공될 수 있다.
- [0038] 참조부호 240에서, HS내에서부터 새도우 가상 FS로의 기록은 또한 저널 파일에 로그될 수 있다. 저널은 예를 들어 매 기록마다 또는 매 수초 정책에 의해 주기적으로 또는 비교가능한 기준에 따라 디스크(집적 드라이브 전자(Integrated Drive Electronics;IDE)디스크를 위한 캐시)에 동기 될 수 있다. 저널 파일은 볼륨 상에서 전용된 데이터 영역 내에서 구현될 수 있고, 저널 파일은 또한 윈도우® O/S 또는 콘텍스트에서 작동하는 시스템 프로그램으로부터 액세스 될 수 있다. 예를 들어, 전용의 확장된 디스크 파티션에서 32비트 파일 할당 테이블(32 bit File Allocation Table;FAT32) FS가 사용될 수 있고, 또는 특허등록 포맷의 저널 영역이 숨겨진, 이동불가능한, 인접한 윈도우® 스크래치 파일의 최상에서 구현될 수 있다.
- [0039] 참조부호 250에서, HS 콘텍스트에서 서비스될 판독 요청이 수신되며, 이는 요청된 데이터가 이미 그곳에 위치되어 있는지를 판정하기 위해 가상 새도우 FS 상에서 체크가 수행되도록 한다. 그렇지 않다면, 데이터 판독 요청은 새도우 가상 FS로부터 부재하는 요청된 파일에 대한 것이고, 또는 요청된 파일은 발견되지만 특정 기록은 발견되지 않는다. 참조부호 260에서, 새도우 FS를 판독함으로써 요청이 서비스될 수 있다면, 이 판독은 수행된다. 반면, 참조번호 270에서 판독전용 NTFS가 판독되고 파일 및 데이터는 그곳에서부터 위치된다.
- [0040] 참조번호 280에서는 HS 콘텍스트로부터 변경이 될 것이고, 참조번호 290에서는 도 3에, 윈도우 콘텍스트에서 실행이 계속될 것이다.

- [0041] 이제 도 3을 참조하면, 참조번호 300은 윈도우 콘텍스트(또는 마이크로소프트® 윈도우® 리소스 파티션의 재부트 시에)에서 실행으로 재개의 시작이다.
- [0042] 참조번호 310에서, 윈도우 서비스는(통상적으로 애플리케이션 데몬의 형태로) 상기된 바와 같이 HS 하에서 생성된 저널 파일을 일관성 검사(유효성 검사) 할 수 있다.
- [0043] 참조번호 320에서, 윈도우 애플리케이션(익스플로러 프로그램과 같은)으로 사전에 인스톨된 필터(플러그인) 프로그램 및/또는 사전에 인스톨된 FS 필터 드라이버는 윈도우 프로세스로부터 특정 파일로의 요청된 액세스를 보류(즉, 연기 또는 지연)할 수 있다. 영향을 받는 파일은 저널 파일을 적용함으로써 수정되도록 현재 대상이 되는 파일이다. (이하에서 설명된 바와 같이) 이 액세스는 HS 기록 저널이 통합되고 완전하게 적용된 이후가 될 때까지 보류된다.
- [0044] 그 후 참조번호 325에서, 서비스가 저널을 NTFS 파일에 적용하고 그 후, 참조번호 330에서, (예를 들어 모두 제로인 데이터를 기록함으로써)저널 콘텍스트를 지울 수 있으며 참조번호 340에서 가상 새도우 FS는 비어있는 것으로서 마크된다(또는 그렇지않으면 적용되어야하는 저널 파일을 더이상 포함하지 않음으로써 적용가능하지 않다). 저널 파일의 적용은 기본적으로 새도우 카피의 캐치업(catch-up)에 도달하고 나중의 액세스(예를 들어 윈도우 내에서부터 될 수 있고 새도잉을 인식하지 못할 수 있다)가 오직 유효하고 새로운(즉, 스테일이 아닌) 데이터를 수신할 것을 보장한다.
- [0045] 참조번호 350에서, 단계 320에서 시행된 액세스의 보류 또는 연기가 해제된다. 이는 재개된 O/S 콘텍스트 내의 계류(pending) 전달이 진행되는 것을 가능하게 하고, 빌드업(built up) 되었어야했을 수 있는 임의의 백로그는 해제된다. 참조번호 390에서 동작의 시퀀스가 완료된다.
- [0046] 도 4는 본 발명의 실시예의 소프트웨어 구성요소 사이의 관계를 도시하는 도면이다. 저레벨에서 BIOS 지원을 사용하여, ACPI(490)을 통한 하이퍼스페이스™ 이중 재개 지원이 존재한다. 이외에 윈도우® O/S 콘텍스트(400) 및 하이퍼스페이스™ O/S 콘텍스트(450), 두가지 O/S 콘텍스트가 존재한다.
- [0047] 윈도우® O/S 콘텍스트(400)는 구성요소 사이에서 분할되어 그래픽 사용자 인터페이스(Graphical User Interface;GUI) 또는 데몬 애플리케이션(410) 중 하나로서 구동하고, 대안적으로는 플러그인(plug-ins)으로서 기능하는 프로그램으로 구동한다.
- [0048] 하이퍼스페이스™ O/S 콘텍스트(450)는 사용자 공간(455)과 커널 공간(470) 사이에서 분할된다.
- [0049] 이제 본 발명의 실시예를 형성하는 다양한 소프트웨어 구성요소가 도 4 전반을 참조하여 설명될 것이다. 윈도우 측(400)에서의 설계는 다수의 구성요소, (A) NTFS 파일 필터(440), (B) FS 전이 관리자(420), 및 (C) FAT 파일 필터(450)로 구성된다.
- [0050] NTFS 파일 필터(440)는 파일 시스템 드라이버로 부트 중 윈도우 파일 시스템 스택에 부착하고 윈도우 O/S의 FS 드라이버로의 모든 콜(calls)을 모니터한다. 이는 파일 공유에서의 FS 전이 관리자를 보조하는 것에 있어서 중요한 역할을 한다.
- [0051] 윈도우에서 HS로의 스위치 동안, NTFS 파일 필터(440)는 FS 전이 관리자(420)로부터의 통지 시에 디스크 캐시를 플러시(flush)한다. NTFS 파일 필터(440)는 파일 핸들에 기초하여 IRP(I/O 요청 패킷)을 큐잉(queueing up)함으로써 모든 FS I/O(입력-출력 작동)가 FS에 접근하는 것을 방해한다. 이 방해는 FS 전이 관리자(420)로부터의 통지 시에 응답하여 시작되고 중지된다. 파일 필터는 오직 판독 및 기록 IRP를 방해하지만 PNP IRP(플러그 앤 플레이 IRP), 파워 IRP, 윈도우 관리 도구(Windows Management Instrumentation IRP; WMI IRP)와 같은 다른 IRP를 가능하게 한다.
- [0052] NTFS 파일 필터(440)는 파일이 두 O/S를 거쳐 공유될 시에 강제적 오픈락(OpLock) 기능을 구현하는데 있어서 중요한 역할을 한다. 오픈락은 다수의 판독기 및 하나의 리코더(recorder) 메커니즘을 구현한다. 오픈락은 하나의 O/S에서 오픈되어있는 파일이 다른 O/S에서의 편집을 위해 오픈되는 경우에 데이터의 무결성(integrity)을 보장한다. 애플리케이션이 공유된 파티션에서 파일을 오픈하도록 요청할 때, NTFS 파일 필터(440)는 IRP를 트랩(trap)하고 로그 파일에서 파일을 룩업(look up)하려고 시도한다. 요청된 파일에 대한 엔트리가 없으면 요청이 허용된다. 그러나 엔트리가 있으면 파일 오픈에 대한 개별 애플리케이션의 요청이 거절된다. 이 기술은 FS가 순차적인 방식으로 O/S를 거쳐 공유되는 것이 가능하도록 하며 반면 (단지 동일 FS가 아닌) 동일한 실제 파일을 동시에 업데이트하도록 시도하는 두 애플리케이션의 통상적인 상황에서는 파일 로킹을 아직도 이행(honoring)한다.
- [0053] 일 실시예에서, (HS에) 유니온FS 필터 드라이버(482)를 사용하여 로그 파일이 구현되며 이는 HS(즉, Dom0)에서

실행하는 애플리케이션이 파일 핸들을 오픈할 때마다 로그 파일에 엔트리를 기록(또는 업데이트)한다. 유니온FS는 리눅스®에서 대중화된 기능이고 본 기술 분야에서 잘 알려져 있다. 파일 핸들에 대한 모든 참조(reference)가 클로즈될 때 관련 로그 엔트리는 로그 파일로부터 삭제된다. (HS 측에서) 유니온FS 드라이버만이 로그 및 (통상적으로 윈도우를 구동시키는) 다른 도메인을 업데이트시키고 각각의 NTFS 파일 필터는 단지 관독만 한다(즉, 이는 관독전용 액세스를 갖는다).

[0054] 당연한 결과로서 유사하게, NTFS 파일 필터(440)는 또한 제 2 로그 파일에서 모든 오픈 파일 엔트리에 로그한다. 이 로그 파일은 특업을 위해 유니온FS 필터(482)에 의해 사용되며 이는 HS에서의 애플리케이션이 편집을 위해 파일을 오픈하는 것을 가능하게 한다. 따라서, 일 실시예에서, 제 2 로그에서의 개별 엔트리의 존재는 파일이 HS에서 오픈되는 것을 방지하도록 동작한다.

[0055] 로거(logger)는 읍록을 구현하는데 사용되므로 - (많은 관독기, 하나의 리코더가 허용됨) 두 O/S 상에서 개별 로거는 애플리케이션이 관독전용 액세스를 위해 파일을 오픈하는 상황에서 파일 상세를 차단시키거나 로그시키지 않는다. 또한, 두 분리 로거를 구비하는 것의 원하는 결과는 RW(관독-기록)모드에서 두 O/S가 (로그가 존재하는) 동일 파티션을 마운트(mount)하는 것을 막는다. 따라서, "윈도우 로그 파일"은 공유된 볼륨에서 존재할 수 있어서 윈도우 필터 드라이버는 관독 및 기록을 할 수 있지만, 이는 HS로부터의 관독전용이다. 역으로, HS 로그는 새도우 볼륨에서 존재하며 이 볼륨은 HS가 업데이트하거나 편집할 수 있지만 다른 도메인(통상적으로 윈도우 측)에게 관독전용 볼륨 또는 FS로서 간주된다.

[0056] FS 전이 관리자(420)는 윈도우 애플리케이션 및 바탕화면(background)에서 구동하는 서비스이다. 서비스 애플리케이션은 사용자에게 임의의 팝업 메시지를 주는 윈도우 애플리케이션을 론칭(launch)한다. 서비스 애플리케이션은 통상적으로 그래픽 디스플레이 인터페이스(Graphical display interface;GDI) 기능에 액세스하도록 설계되지 않으며 직접 이렇게 하려는 어떠한 시도도 실행 중 실패할 것이다. 서비스 애플리케이션은 서비스 제어 관리자(service control manager;SCM - 잘 알려진 윈도우 컴포넌트)로부터의 통지를 듣고 또한 대기하며 통지 이벤트를 재개한다.

[0057] FS전이 관리자(420)는 다수의 역할을 수행하는 중요한 구성요소이다 : (A) 로그온 단계 중 자체를 론칭한다. 이는 윈도우 패키지 시스템 영역(packaged system area;PSA) 파티션을 마운트하고 이를 사용자 레벨로부터 숨긴다. (B) 윈도우가 활성화될 때마다, FS전이 관리자(420)는 ACPI 시스템 상태를 변경하기 위한 트리거를 듣고 수신시에는, NTFS 파일 필터(440)가 디스크 캐시를 플러시하고 I/O를 임시로 차단하도록 통지한다. (C) 윈도우가 대기상태로부터 재개할 때마다, 애플리케이션은 웨이크업(wake up) 통지를 듣고 수신시에는, 새도우 볼륨으로부터 공유된 볼륨 파티션으로 카피를 수행한다. 카피가 완료되면, 이는 필터 드라이버가 계류(pending) IRP를 차단해제하도록 통지한다. (D) FS 전이 관리자(420)는 또한 대기상태로부터 재개할 시에 윈도우 PSA 볼륨을 마운트한다. 본 발명의 실시예에서 새도우 볼륨은 윈도우 PSA에 있다.

[0058] FAT32 파일 필터(450)는 FAT32 파일 시스템의 스택에 위치한 필터 드라이버이다. FAT32는 본 기술분야에서 잘 알려져 있다. 이 설계에서 윈도우 PSA는 새도우 파티션으로서 사용되고 이는 FAT32 파티션이다. FS 전이 관리자(420)는 파티션을 마운트하고 필요한 카피 작동을 수행한다. 마운트되면, 윈도우는 파티션을 사용자에게 사용가능하게 만들고 사용자는 필요로 하는 만큼 이를 활용할 수 있다. 윈도우 PSA 상에서 (잠재적으로 문제가 있는) 임의의 이러한 작동을 회피하기 위해, 설계에 의한 FAT32 파일 필터(450)는 윈도우 PSA로 어드레스된 I/O를 차단한다.

[0059] 이제 하이퍼스페이스™ 측(450) 상에서의 설계로 돌아가서, 사용자 공간(455)에는, 다양한 구성요소들이 있다 : (D) HS 전이 관리자(460), (E) HS 전이 관리자 UI(사용자 인터페이스)(465), 및 (F) 스크립트 지원(470). 커널 공간(470)에는 파일 필터, 가상 파일 시스템(Virtual file system;VFS)(480) 및 유니온FS(482)가 있다. 이들 파일 필터는 다양한 종류의 FS(파일 시스템), "ext3"(484)(디폴트 리눅스® FS), NTFS(486)및 FAT32(488)와 작동한다.

[0060] HS 전이 관리자(460)는 NTFS RO(관독전용) 파티션에 대해 제어된 I/O를 관리한다. 업데이트한 디스크, 즉, 임의의 RO 작동을 필요로 하지 않는 트랜잭션이 존재할 때마다, HS 전이 관리자(460)는 적절한 NTFS 파티션으로부터 파일을 관독하고 이를 RO 모드에서 오픈한다. 임의의 트랜잭션이 업데이트한 디스크를 필요로 할 때 HS 전이 관리자(460)는 RO 모드에서 NTFS 파티션으로부터 파일을 오픈하지만, 이 경우에 I/O 전달(기록 또는 재기록)은 새도우 볼륨으로 지향되며 이 볼륨은 NTFS 파티션 자체로 지향되는 I/O 전달 대신 RW(관독기록)모드에서 마운트된다.

- [0061] 본 발명의 일 실시예에서, HS 파일 필터 모듈은 유니온FS(482)와 관련하여 동작하는 VFS(480)을 위한 파일 필터의 부분이 될 수 있다. 사용자 공간에서 애플리케이션이 NTFS 파티션에 대해 직접 발생한 기록 I/O의 뷰를 제시할지라도 유니온FS(482)는 두 마운트 포인트를 통합하고 새도우 볼륨(488)으로의 I/O 기록을 수행하는 것을 처리하는 파일 시스템의 종류이다. 데이터가 최근 기록 작동에 의해 변경될 때마다, 유니온FS의 사용은 판독 작동을 제공하며 이는 (원본 및 변경되지 않은 데이터보다는) 새도우 볼륨 상에서 편집된 데이터로부터 적절하고, 특별하게 이 데이터를 페치(fetch)할 것이다.
- [0062] VFS(480)를 위한 파일 필터는, HS 파일 필터 모듈 기능을 제공하는 것 뿐만아니라, 예를 들면 ext3(484) 및 NTFS(486)로의 직접 일반적인 파일 액세스를 제공한다.
- [0063] HS 전이 관리자(460)는 파일의 상태에 대한 정보를 판독하고 이는 (도 4에서는 도시되지 않은) 메타데이터 파일로부터 기록 I/O를 위해 오픈된다. 메타데이터 파일은 상술된 로그 파일의 기능으로서 구현될 수 있고 또는 이는 개별적으로 생성되고 유지될 수 있다. 메타데이터 파일은 RO 모드에서 마운트되는 NTFS 파티션에 존재한다. (윈도우에서) NTFS 파일 필터는 윈도우에서 편집되거나 업데이트되는 임의의 적절한(잠재적으로 공유가능한) 파일에 관한 정보를 갖는 이 메타데이터 파일을 업데이트한다. 윈도우 시스템 파일과 같은, 일부 파일은 의도적으로 불포함될 수 있어서 HS 애플리케이션 측에 대해 알려지지 않은 상태를 유지한다.
- [0064] 메타데이터 파일에 포함된 정보는 파일이름, NTFS 파티션에서의 위치, 현상태, 파일 기술어(descriptor) 참조번호 및 그 이상을 포함한다. 파일이 HS에서 기록허용 모드로 오픈되고 HS 전이 관리자(460)가 메타데이터 파일로부터 이 파일이 "편집을 위해 윈도우에서 오픈된" 상태에 있다는 것을 발견하면, HS 전이 관리자(460)는 HS 전이 관리자 UI(465)가 적절한 메시지를 디스플레이하도록 신호를 보내고 파일 오픈이 불허된다.
- [0065] 유사하게, HS 전이 관리자(460)은 새도우 볼륨에서 메타데이터 파일을 유지관리한다. 구조 내부는 윈도우에 의해 업데이트 된 NTFS 파티션에서 존재하는 메타데이터 파일과 유사하다. HS 전이 관리자(460)는 임의의 HS 애플리케이션으로부터의 요청이 NTFS 파티션에 속하는 기록(writing)을 위해 파일을 오픈하기 위해 도착할 때마다 이 메타데이터 파일을 업데이트하며 이는 윈도우에서 이미 편집된 것은 아니다. HS에서 편집되는 파일이 윈도우를 향해 콘텍스트를 스위칭하기 이전에 클로즈되면, 이 파일에 대한 엔트리는 메타데이터 파일로부터 제거된다. 따라서 이는 이 파일을 편집하기 위해 오픈할 수 있다는 것을 윈도우에 나타낸다. 메타데이터를 따라, 실제 파일 자체는 새도우 볼륨에서 저장된다. 콘텍스트가 윈도우로 스위치 될 때마다, 이 파일은 윈도우에 의해 NTFS 파티션 상에서 이의 실제 위치로 카피된다. 특히 이 상황은 HS 콘텍스트에서 신규 파일이 생성되고 윈도우 측에 대해 사용가능하게 만들어지는 곳에서 적용할 수 있다. 이는 또한 저널이 간단하게 전체 파일인 상황과 동등할 수 있으며(또는 간주될 수 있으며), 어떻게 이것이 구현되는지에 관한 실시예이다.
- [0066] HS 파일 필터는 리눅스 스타일 유니온FS(482) 드라이버에 의해 구현된다. 유니온FS는 VFS와 NTFS-3G 드라이버 사이에 위치한 파일 시스템의 종류이다.
- [0067] 유니온FS 는 본 기술분야에서 잘 알려져 있다. 기능적으로, 유니온FS(482)는 여러 파일 시스템(484, 486, 488)의 최상에서 또는 종종 단일 파일 시스템 내 다수의 디렉토리 상에 동시에 레이어(layer)한다. 유니온FS(482)는 하나의 단일 마운트 포인트(mount point)로 여러 디렉토리를 오버레이(overlays)시킨다. 첫 번째로 이는 최상 브랜치 상의 파일에 액세스하려고 시도하고 성공적이지 못하면 더 낮은 레벨 브랜치 상에서 계속된다. 사용자가 RO인 더 낮은 레벨 브랜치 상에서 파일을 수정하려고 시도하면, 그 후 파일이 더 높은 레벨의 판독기록 브랜치로 카피된다. 유니온FS(482)는 커널로 파일시스템 인터페이스를 제시하고, 차례대로 유니온FS는 자체를 커널의 VFS로서 이것이 스택된 파일 시스템으로 제시한다. 그러한 이유로 유니온FS는 파일시스템 뷰를 커널로 제시하고, 이는 사용자 레벨 애플리케이션에 의해 이용될 수 있다. 유니온FS는 더 낮은 레벨의 파일시스템을 향한 작동을 인터셉트하고, 따라서 이는 통합된 뷰를 제시하기 위해 작동을 변경할 수 있다.
- [0068] 본 발명의 실시예에서, 두 파일시스템은 NTFS(판독전용) 및 FAT32 새도우 볼륨(판독기록)이 될 수 있다. 사용자(사용자 모드 애플리케이션)는 새도우 볼륨으로의 액세스가 허용되지 않지만, 예외로서 HS 전이 관리자(460)는 허용된다. FAT32 새도우 볼륨(488)은 NTFS 파티션(486)에서 생성되도록 시도된(sought) 메타데이터 파일 및 신규 파일을 유지관리하는데 사용될 수 있다.
- [0069] HS 전이 관리자(460)는 유니온FS-FUSE 드라이버(사용자 공간에서의 유니온FS 파일 시스템)또는 대안적으로 유니온FS 커널 공간 파일시스템 드라이버 중 하나로서 구현될 수 있다. 유니온FS-FUSE 드라이버는 도 4에서 도시되지 않지만 본 기술분야에서 잘 알려져 있다.
- [0070] 스크립트 지원(468)은 유니온FS 커널 모듈을 로드하도록 커스터마이징된(customized) 스타트업(Startup) 스크립

트를 제공한다. 유니온FS-FUSE가 사용되면, 스크립트는 FUSE 커널 모듈을 로드하고 유니온FS-FUSE를 호출 (invoke)해야만한다. 그러면 (유니온FS-FUSE가 사용되는지 여부와 상관없이) HS 전이 관리자 데몬(460) 및 HS 전이 관리자-UI(465)가 시작된다.

[0071] 도 5와 관련하여, 전자 디바이스(10)로 통합될 컴퓨터 명령어가 제조된 펌웨어 및/또는 소프트웨어 컴퓨터 제품(510)으로서 배포될 수 있으며 이는, 예를 들어, 스토리지 리코더(520)를 사용함으로써 리코딩된 명령어를 갖는 다양한 가능 매체(530)를 사용한다. 본 발명을 활용(deploy)한 제품만큼 복잡한 제품에서는 종종, 배포 및 관련 제품 제조 양쪽 모두에서 하나 이상의 매체가 사용될 수 있다. 명료성을 위해 오직 하나의 매체가 도 5에서 도시되었지만 하나 이상의 매체가 사용될 수 있고 단일 컴퓨터 제품은 복수의 매체 사이에서 분할될 수 있다.

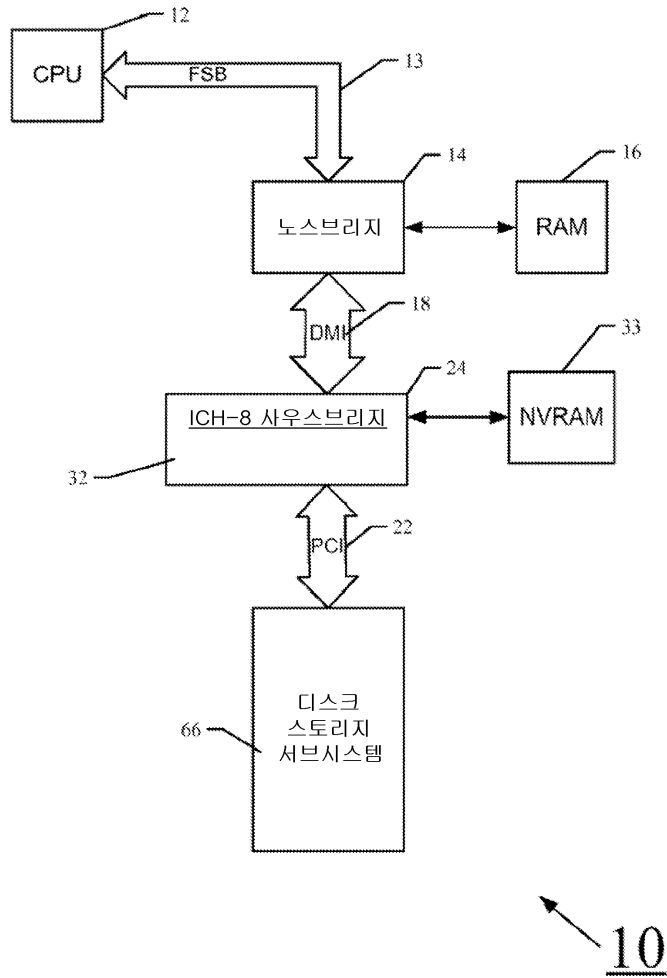
[0072] 도 6은 본 발명의 예시적인 실시예가 전자파를 사용하여 어떻게 인코딩, 전송, 수신 및 디코딩될 수 있는지를 도시한다.

[0073] 도 6과 관련하여, 부가적으로 특히 인터넷 사용에서의 증가 때문에, 컴퓨터 제품(610)은 파동으로서 변조된 신호로 이를 인코딩함으로써 배포될 수 있다. 그 후 최종 파형은 전송기(640)에 의해 전송될 수 있고 유형의 변조된 전자 반송파로서 전파될 수 있으며 수신기(660)에 의해 수신될 수 있다. 수신시 이들은 복조 될 수 있고 메모리 또는 다른 스토리지 디바이스에서 추가 버전 또는 컴퓨터 제품(611)의 카피로 디코딩되는 신호는 제 2 전자 디바이스(11)의 부분이며 통상적으로 전자 디바이스(10)와 유사한 성질이다.

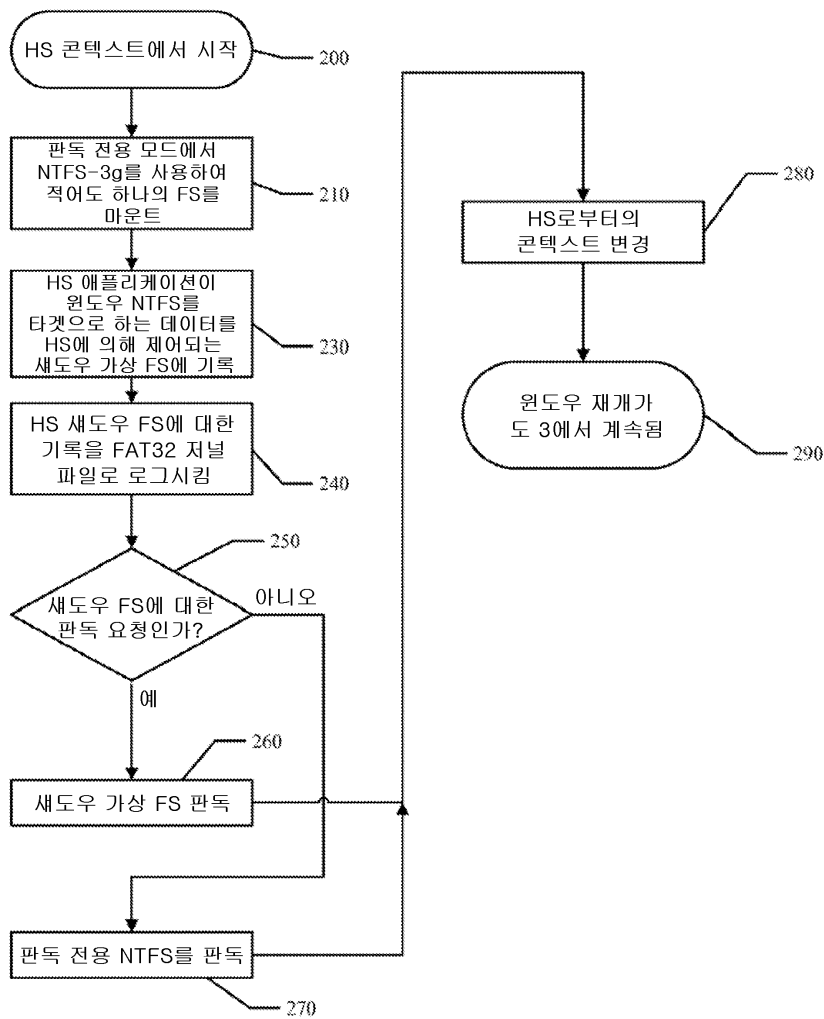
[0074] 다른 토폴로지(topologies) 및/또는 디바이스는 또한 본 발명의 대체 실시예를 구성하는데 사용될 수 있다. 상술된 실시예는 제한하는 것이기보다는 예시적인 것이며 본 발명의 경계는 청구항으로부터 결정된다. 비록 본 발명의 선호되는 실시예가 위에서 자세하게 설명되었지만, 본 기술분야의 당업자에게 나타날 수 있는 본 명세서에서 교시된 기본적인 발명의 개념의 많은 변형 및/또는 수정은 여전히 첨부된 청구항에서 정의된 바와 같은, 본 발명의 사상 및 범위 내에 속하게 된다는 것이 명백하게 이해될 것이다.

도면

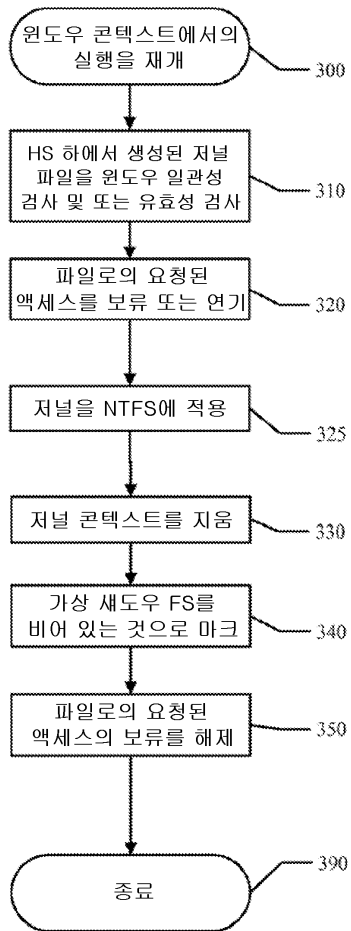
도면1



도면2

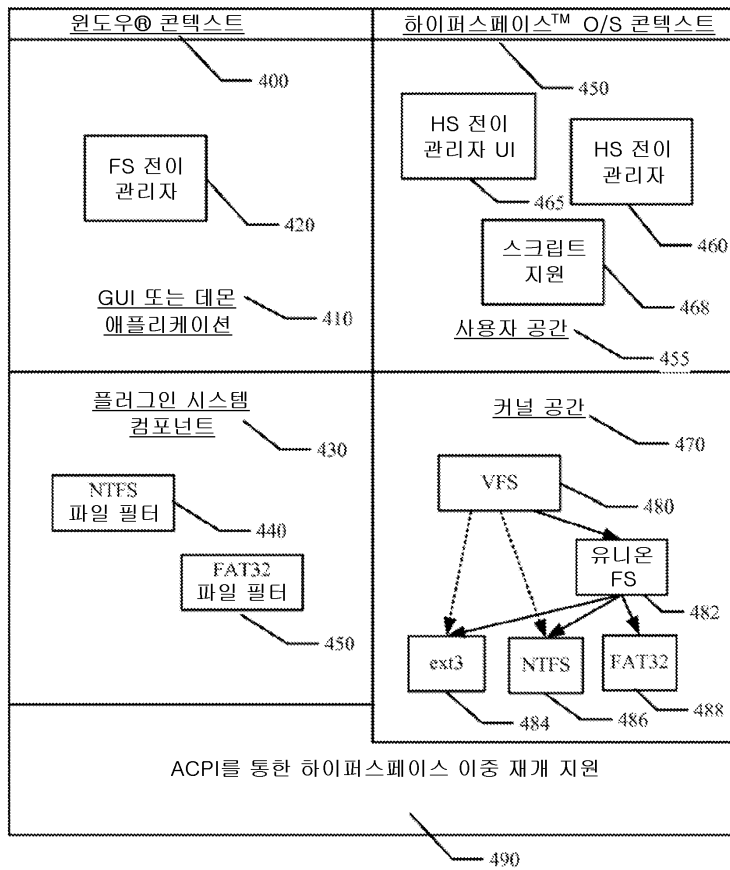


도면3

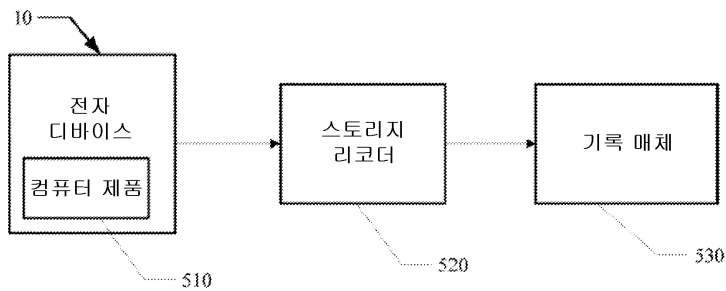




도면4



도면5



도면6

