



(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl.	(45) 공고일자	2007년02월15일
G11B 7/004 (2006.01)	(11) 등록번호	10-0683058
	(24) 등록일자	2007년02월08일

(21) 출원번호	10-2005-7001443(분할)	(65) 공개번호	10-2005-0026005
(22) 출원일자	2005년01월26일	(43) 공개일자	2005년03월14일
심사청구일자	2006년01월05일		
번역문 제출일자	2005년01월26일		
(62) 원출원	특허10-2002-7011315		
	원출원일자 : 2002년08월28일	심사청구일자	2002년08월28일
(86) 국제출원번호	PCT/JP2001/001309	(87) 국제공개번호	WO 2001/63419
국제출원일자	2001년02월22일	국제공개일자	2001년08월30일

(81) 지정국 국내특허 : 캐나다, 중국, 대한민국, 미국,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 사이프러스, 독일, 덴마크, 스페인, 핀란드, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투갈, 스웨덴, 터키,

(30) 우선권주장 JP-P-2000-00050428 2000년02월28일 일본(JP)

(73) 특허권자 샤프 가부시기가이샤
일본 오사카후 오사카시 아베노구 나가이게쵸 22방 22고

(72) 발명자 이와노유리
일본 치바 266-0005 치바시 미도리쿠 혼다초 2-24-7-에이217

이케다나츠코
일본 치바 262-0033 치바시 하나미가와쿠 마쿠하리혼고 6-13-18-212

키야마지로
일본 치바 274-0825 후나바시시 마에바라니시 2-31-21-206

니시무라모토허데
일본 후쿠오카 807-0844 키타큐슈시 야와타니시쿠 카스가다이 5-15-22

야마무라히로유키
일본 치바 치바시 미도리쿠 오유미노 2-10-1-에프201

야마구치타카요시
일본 치바 270-0120 나가레야마시 니시하츠이시 6-829-33

(74) 대리인 백덕열
이태희

심사관 : 이백수

전체 청구항 수 : 총 4 항

(54) 파일관리방법

(57) 요약

1개의 기록매체에 있어서, 복수의 애플리케이션으로부터의 액세스를 행하는 경우, 통상 복수의 영역으로 분할한다. 이 영역분할을 파티션에 의해 분할관리한 경우, 파티션의 사이즈를 용이하게 변경할 수 없기 때문에, 기록매체를 효과적으로 사용하는 것이 가능하지 않다는 문제가 있었다.

기록매체를 복수의 영역으로 분할하고, 그 영역마다, 영역의 빈 공간을 더미데이터로서 기입하고, 해당 더미데이터는 특정한 애플리케이션 또는 특정한 파일밖에 재기입할 수 없도록 구성함으로써, 복수의 영역분할 및 사이즈변경을 용이하게 한다.

대표도

도 2

특허청구의 범위

청구항 1.

삭제

청구항 2.

삭제

청구항 3.

삭제

청구항 4.

삭제

청구항 5.

삭제

청구항 6.

기록매체와 상기 기록매체로의 데이터의 입출력을 관리하는 기록제어수단을 구비하는 기록장치에서, 기록매체에 하나 이상의 영역을 확보하고 각 영역의 빈 공간을 각각 더미 파일에 의해 관리하는 파일관리방법에 있어서,

해당 영역에 파일을 기입하는 경우에는, 해당 영역의 더미 파일상에 파일을 기입하고, 더미 파일의 사이즈를 상기 영역의 빈 공간의 사이즈로 갱신하고,

각 영역의 최후단은 항상 더미블록으로 구성되는 것을 특징으로 하는 파일관리방법.

청구항 7.

기록매체와 상기 기록매체로의 데이터의 입출력을 관리하는 기록제어수단을 구비하는 기록장치에서, 기록매체에 하나 이상의 영역을 확보하고 각 영역의 빈 공간을 각각 더미 파일에 의해 관리하는 파일관리방법에 있어서,

해당 영역에 파일을 기입하는 경우에는, 해당 영역의 더미 파일상에 파일을 기입하고, 더미 파일의 사이즈를 상기 영역의 빈 공간의 사이즈로 갱신하고,

각 영역의 최선단 및 최후단은 항상 더미블록으로 구성되는 것을 특징으로 하는 파일관리방법.

청구항 8.

제6항 또는 제7항에 있어서, 상기 각 더미 파일은, 소정의 애플리케이션으로부터의 기입만을 허용하는 것을 특징으로 하는 파일관리방법.

청구항 9.

제6항 또는 제7항에 있어서, 상기 각 더미 파일은, 소정 타입의 파일만의 기입을 허용하는 것을 특징으로 하는 파일관리방법.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은, 1개의 기록매체에 복수의 애플리케이션으로부터의 파일의 기입을 행하는 경우에 있어서, 기록매체를 복수의 영역으로 분할하여 이용하는 파일관리방법에 관한 것이다.

PC 용도나 AV 용도 등의 상이한 플랫폼(platform)에 있어서, 공통 이용할 수 있는 범용의 디스크매체가 존재하는 것은, 유저(user)에게 있어서 큰 메리트가 있다. 예컨대, AV 디스크 레코더 등으로 기록된 디스크를, PC에 접속된 디스크 드라이브에서 액세스할 수 있거나, 그 반대의 액세스를 용이하게 할 수 있는 것은, AV 디스크 레코더 등에 의해 기록된 AV 데이터를 PC에서 액세스하거나 편집할 수 있고, 편집결과 등을 또한 AV 디스크 레코더 등에 의해 용이하게 재생할 수 있는 것을 의미한다. 또한 1개의 디스크에, AV 데이터를 기록하거나, PC의 애플리케이션 소프트웨어를 넣으면, AV 용도와 PC 용도로 공통하여 이용하는 것도 고려될 수 있다.

그러나, AV 용도로 기록되는 데이터와 PC 용도로 기록되는 데이터의 성질은 상이한 것이다. 예컨대, 디스크에 기록된 AV 데이터를 재생하는 경우, AV 데이터를 디스크로부터 독출하고, 정해진 시간에 디스플레이에 표시를 행하지 않으면 안된다. 정해진 시간에 표시를 행할 수 없는 경우는, 재생화면이 도중에 끊겨 재생화면이 이상하게 되는 것을 의미하며, 허용되는 것이 아니다.

디스크매체는 랜덤 액세스성이 우수하기 때문에, 일련의 데이터이라도 디스크상에 있어서 연속적으로 배치되어 있을 필요는 없고, 디스크상의 빈 공간을 효과적으로 이용하여, 분단하여 기록되어 있더라도 상관없다. 예컨대 PC 용도의 경우, 문장 데이터 파일이 디스크상에서 분단하여 기록되어 있었다고 해도, 그 파일을 독출할 때에 각 분단점에서 씨크(seek)나 트랙 점프가 발생하고, 그 기간 디스크로부터의 데이터의 독출을 중단하지만, 연속적으로 데이터가 독출되는 경우와 비교하여 데이터의 독출 시간이 다소 걸리는 정도로, 기능면에서의 문제는 전혀 없다.

그러나, AV 용도에 있어서는 전술한 바와 같이, 재생하고자 하는 데이터가 디스크상에서 분단하여 기록되어 있으면, 각 분단점에서 데이터의 독출의 중단이 발생하여, 문제가 발생할 가능성이 있다. 일반적으로, 디스크로부터 독출된 AV 데이터는 일단 버퍼 메모리에 어느 정도 저장되어, 씨크나 트랙 점프 등의 데이터 독출의 중단이 발생하였다고 해도, 버퍼 메모리

에 저장된 데이터로 보충함으로써, 즉 재생화면이 도중에 끊기는 것을 막고 있다. 그러나, 버퍼 메모리에 의해 재생화면이 도중에 끊기는 것을 흡수하는 것도, 번잡하게 씨크나 트랙 점프가 발생하면 대응할 수 없게 된다. 이에 의하여, AV 데이터를 디스크에 기록하는 경우는, 데이터의 독출 중단이 발생하는 씨크나 트랙 점프가 발생하지 않는, 데이터의 연속적인 기록이 요망된다.

여기서, 1개의 디스크를 AV 용도와 PC 용도로 공용으로 이용하는 것을 고려하면, 각각의 용도의 데이터를, 각각 상이한 영역에 기록하고 싶다고 하는 요구가 발생한다. 이는, 연속적으로 데이터를 기록하고 싶은 AV 데이터가 기록되어 있는 디스크영역에, AV 데이터와 비교한 경우, 데이터량이 훨씬 작은 PC 용도의 데이터가 수월하게 기록되어 가면, AV 데이터의 연속기록을 방해하는 요인으로 되어, 경우에 따라서는 AV 데이터의 기록재생에 지장을 줄 가능성이 있기 때문이다.

또한, AV 용도이더라도, 기록되는 데이터는 AV 데이터뿐만 아니라, 그들을 재생하기 위한 관리정보 파일이나, 정지화상 등 여러 가지 종류의 데이터가 존재한다. 동일용도이더라도 성질이 상이한 데이터를 기록하기 때문에, 기록되는 데이터의 종류에 따라 영역을 나눠 관리하고 싶다고 하는 요구가 발생한다. 예컨대, 논리파일시스템의 관리정보를 기록하는 영역, AV 데이터를 재생하기 위한 관리정보 등을 기록하는 영역, AV 데이터 자체를 기록하는 영역, 정지화상을 기록하는 영역이 고려될 수 있다.

1개의 디스크를 복수의 영역으로 나눠 사용하기 위해, 종래부터, 이하와 같은 방법이 있다. 우선, 첫째, 영역을 명확히 나눈다고 하는 의미에서는, 논리파일시스템의 파티션 기능이 존재한다. 예컨대 AV 용도와 PC 용도용의 파티션을 정의함으로써, 각각의 전용영역을 제공하는 것이 가능해진다.

도16에 도시된 예에서는, 디스크를 파티션 1, 파티션 2, 파티션 3의 3개로 분할하고 있는 모양을 도시하고 있다.

제2 방법으로서, 파티션 기능을 이용하지 않는 경우는, AV 용도와 PC 용도의 영역을 논리파일시스템에서는 관리하지 않고, 실장레벨의 애플리케이션층의 관리정보로서 영역을 관리하는 것이 고려될 수 있다. 예컨대, 각각의 영역의 위치정보를 기록한 관리정보 파일을 디스크에 기록하여, 그 파일을 독출하는 것으로 영역의 위치정보를 파악하는 것이 가능해진다. 도 17에 도시된 예에서는, AREA.DAT라는 파일에 영역 1, 영역 2, 영역 3의 위치정보가 기록되어 있다. 이에 의해, AREA.DAT라는 파일을 해석할 수 있는 애플리케이션만이 영역의 범위를 파악하는 것이 가능해진다.

상기한 제1 방법과는 달리, 여기서의 영역 1~3은 논리파일시스템상에서는 1개의 파티션으로 구성되고, 그 속에서, 영역으로 구분되어 있는 것으로 된다.

셋째, 1개의 파일이나 데이터를 기록하는 영역을 예약하는 용도로, 실제로 데이터를 기록하지 않고 그 영역이 마치 사용되고 있는 것처럼 더미(dummy) 파일을 작성하는 것도 고려될 수 있다. 도18에 있는 영역내에 기록되는 파일이 각각 사용할 예정인 영역을 DUMMY1.DAT, DUMMY2.DAT, DUMMY3.DAT라는 3개의 더미 파일에 의해 확보하고 있는 것이다.

이와 같이 더미데이터로서, 미리 기록되는 데이터와 동일한 사이즈의 데이터를 기입해 놓고, 소정의 데이터 타입의 파일을 기입할 때, 그 더미데이터를 삭제하여, 그 위치에 파일을 기입함으로써, 데이터 타입마다 파일을 연속하여 기입할 수 있다.

발명이 이루고자 하는 기술적 과제

상기한 제1 방법에서는, AV 용도와 PC 용도의 영역을 각각 파티션 기능에 의해 나눠 각각의 전용영역을 제공하는 것이 가능하다. 그렇지만, 일반적으로 파티션은, 디스크를 초기화할 때에 파티션의 수나 크기 등을 정하여 작성하는 것으로, 사용중에 파티션의 구성을 변경하는 것은 용이하게 행할 수 있는 것이 아니다. 이는, 파티션마다 독립한 논리적인 어드레스나, 빈 공간의 관리정보 등을 가지고 있는 것이 일반적이기 때문에, 구성을 변경하는 경우는, 많은 관리정보의 재구성을 행할 필요가 있기 때문이다. 또한, 유저는 초기화시에 AV 용도와 PC 용도에 각각 어느 만큼의 영역을 확보할지를 정하지 않으면 안된다. 그러나, 사용해 가는 과정에서, 어느 쪽의 영역이 부족하게 되는 것이 고려될 수 있다.

전술한 바와 같이, 파티션의 크기 등을 변경하는 것이 용이하지 않기 때문에, 예컨대, PC 용도의 영역에 빈 공간이 남아 있었다고 해도, AV 용도의 영역이 가득 찬 시점에서 그 이상 AV 데이터를 기록하는 것이 불가능하게 되어, 디스크를 효과적으로 이용할 수 없는 문제가 있다. 또한, 동일한 AV 용도에 있어서도, 논리파일시스템의 관리정보를 기록하는 영역, AV 데이터를 재생하기 위한 관리정보 등을 기록하는 영역, AV 데이터를 기록하는 영역, 정지화상을 기록하는 영역과 같이 복수의 영역으로 나눠 관리를 하고 싶은 경우도 있어, 이들의 영역을 각각 파티션으로 관리를 하면, 파티션의 수가 많아지므로, 1개의 파티션이 가득 찬 경우에, 디스크가 비어 있었다고 해도, 그 이상 데이터 등을 기록할 수 없게 될 가능성이 그만큼 높아지게 되는 문제가 있다.

제2 방법과 같이, 영역을, 논리파일시스템에 의해 제공되는 파티션 기능으로서는 관리하지 않고, 실장레벨의 애플리케이션층의 관리정보로서 관리한 경우, 애플리케이션에 의해 영역을 관리하고 있기 때문에, 상정하지 않고 있는 애플리케이션에 있어서는, 이 영역의 위치정보가 기록되어 있는 관리정보는 아무런 의미를 가지지 않는다. 각각 확보된 영역은, 논리파일시스템 레벨에서 확보되어 있는 것은 아니기 때문에, 상정하는 애플리케이션에 의해서만 그 디스크를 사용하는 경우는 문제없지만, 상정 외의 애플리케이션으로부터 액세스된 경우는, 영역을 인식할 수 없기 때문에, 확보된 영역내에 상정 외의 데이터가 기록될 가능성이 있어, 문제가 있다.

또한, 제3 방법과 같이, 1개의 파일이나 데이터를 기록하는 영역을 예약하기 위해 더미 파일을 이용함으로써, 영역을 확보하는 것이 가능하다. 이 방법에 있어서는, 예컨대 관리정보를 기록하기 위한 연속영역을 확보하고 싶은 경우, 이후 기록되는 파일의 크기, 수를 예측하여, 그와 동일한 사이즈, 수의 더미 파일을 기록하는 것으로 영역을 확보하지 않으면 안된다. 이에 의하여, 기본적으로는 파일의 상정되는 최대 사이즈로 더미 파일을 작성하게 되어, 실제 기입되는 파일이 더미 파일의 사이즈보다 작은 경우, 낭비가 많이 발생한다고 하는 문제가 있다.

도18에 도시된 영역 1에 있어서, 기입되는 파일의 사이즈가 각각 1 M Byte라고 상정한 경우, 1 M Byte의 사이즈의 더미 파일이 작성되게 된다. 여기에 0.8 M Byte의 파일을 기입하는 경우는, 우선 최초의 더미 파일(DUMMY1.DAT)의 위치에 0.8 M Byte의 데이터가 기입되고, 다음에 0.9 M Byte의 데이터가 기입되는 경우는, 다음 더미 파일(DUMMY2.DAT)의 위치에 기입되기 때문에, 0.1 M Byte의 낭비된 부분이 발생하게 된다. 또한, 기입되는 파일의 사이즈를 예측할 수 없는(최대 사이즈를 기대할 수 없는) 경우에는, 대응할 수 없다고 하는 문제가 있다.

그래서, 본원 발명은 상기 3개의 방법에 있어서의 과제를 해결하는 것으로, 논리파일시스템에 의한 파티션의 구분을 행하지 않고, 소정의 영역내에, 소망의 데이터 이외가 기입되는 것을 방지하는 것이 가능해진다.

발명의 구성

본 발명의 제1 요지는, 기록매체와 상기 기록매체로의 데이터의 입출력을 관리하는 기록제어수단을 구비하는 기록장치에서, 기록매체에 하나 이상의 영역을 확보하고 각 영역의 빈 공간을 각각 더미 파일에 의해 관리하는 파일관리방법에 있어서, 해당 영역에 파일을 기입하는 경우에는, 해당 영역의 더미 파일상에 파일을 기입하고, 더미 파일의 사이즈를 상기 영역의 빈 공간의 사이즈로 갱신하는 것이다.

본 발명의 제2 요지는, 상기 각 더미 파일이, 소정의 애플리케이션으로부터의 기입만을 허용하는 것이다.

본 발명의 제3 요지는, 상기 각 더미 파일이, 소정 타입의 파일만의 기입을 허용하는 것이다.

본 발명의 제4 요지는, 각 영역의 최후단(最後端)이 항상 더미 블록으로 구성되는 것이다.

본 발명의 제5 요지는, 각 영역의 최선단(最先端) 및 최후단이 항상 더미블록으로 구성되는 것이다.

본 발명에 의하면, 확보된 연속영역내의 미사용 부분을 관리하도록 더미 파일을 작성함으로써, 이 영역을 사용하는 애플리케이션 이외에 있어서, 이 영역내에는 모두 사용되고 있는 것으로서 취급되어, 데이터가 기입되는 것을 억제하는 것이 가능해진다.

또한, 더미 파일을 규정한 애플리케이션에 있어서는, 더미 파일의 파일 사이즈나 위치정보를 취득함으로써, 확보된 영역내의 미사용 개소(箇所)의 크기나 위치를 용이하게 파악하는 것이 가능해져 영역내에서의 파일작성이나 갱신을 용이하게 행할 수 있게 된다.

또한, 더미 파일을 이해할 수 있는 애플리케이션이 복수의 메이커 등에 의해 실장되어 있는 경우, 확보되는 연속영역의 크기가 반드시 동일한 것에 한정되지는 않는다. 이 경우에도, 더미 파일의 정보를 참조하는 것만으로, 미사용 부분을 파악할 수 있기 때문에 메이커간의 호환을 취하는 것도 가능해진다.

또한, 논리파일시스템의 기능인 파티션 기능을 이용하고 있지 않기 때문에, 파티션의 크기 등을 변경하는 것과 비교하여, 더미 파일의 갱신이라고 하는 단순한 처리만으로 영역의 크기를 변경하는 것이 가능해진다. 확보된 연속영역의 범위를 알 필요가 있는 경우는, 필요에 따라 연속영역의 최후미(最後尾) 또는 선두와 최후미의 양방의 논리블록을 더미 파일로 관리함으로써, 관리하고 있는 연속영역의 범위를 용이하게 파악하는 것이 가능해진다.

이하, 본 발명의 영역관리방법에 관한 실시예에 관해서, 도면과 함께 상세에 관해 설명한다. 본 실시예는, 디스크장치로서, AV 기록재생을 목적으로 하는 디스크를 사용한 휴대형 비디오 카메라나 비디오 팩(deck), PC에 접속된 외부기억장치 등을 상정한 것이다. 디스크매체는, 리무버블(removable) 디스크가 바람직하지만, 하드디스크 등의 설치형이더라도 상관없다. 또한, 설명의 편의상 디스크에 이용되는 논리파일시스템으로서 OSTA(Optical Storage Technology Association)의 규격인 UDF(Universal Disk Format)를 상정하지만, 그 밖의 범용논리 파일시스템이더라도 상관없다.

일반적인 디스크장치의 구성을 도1에 도시한다. 데이터입출력부(1)는 카메라 등으로부터 입력되는 영상신호를 취입하거나, 재생되는 데이터를 모니터 등에 출력하기도 한다. 데이터처리부(2)는, 예컨대 MPEG 부호를 인코딩하거나, 디코딩하기도 하는 신호처리 등을 행하는 처리부이며, 처리된 데이터는 메모리(3)에 저장된다. 데이터를 기록하는 경우는, 디스크 제어부(5)에 있어서, 디스크(6)를 제어하여 디스크상의 원하는 개소에 데이터가 기록되고, 재생시에는, 디스크(6)를 제어하여 디스크상의 원하는 개소에서부터 데이터가 독출되어 메모리(3)에 저장된다. 각 처리부는 시스템제어부(4)에 의해 제어된다.

이러한 디스크장치에 있어서, 디스크의 영역을 분할하여 관리하고 싶은 경우는, 논리파일시스템에 있어서의 파티션 기능을 이용하는 것이 일반적이다. 그러나, 전술한 바와 같이, 일단 파티션을 구분하면 파티션의 크기를 변경하거나, 어떤 파티션을 다른 파티션과 연결시켜 크기를 확대하는 것을 용이하게 행할 수 없다. 본 실시예에서는 파티션 기능에 의한 영역의 분할은 행하지 않기 때문에, 도2에 도시된 바와 같이 디스크의 유저영역 전역을 관리하도록 1개의 파티션을 작성한다. 작성된 파티션내에, 어떤 특정한 정보를 기록하기 위한 영역을 정의하는 것으로 한다. 파티션 기능을 이용하지 않는 것과, 범용 파일시스템인 UDF를 이용하는 것 때문에, 1개의 파티션내에 복수의 영역을 관리하는 것 같은 논리파일시스템 레벨의 관리정보는 존재하지 않는다. 여기서 중요하게 되어 오는 것이, 1개의 디스크를 복수의 플랫폼에서 사용할 가능성이 있는 것이다. 예컨대, AV 용도로 기록된 디스크를 PC의 드라이브에서 액세스하는 것이 고려될 수 있다. 따라서, AV 애플리케이션(제1 애플리케이션)이라고 하는 특정한 애플리케이션에 의해서만 이해할 수 있는 것 같은 정보(더미파일)에서의 영역 확보[실데이터영역, 제1 영역(관리정보영역)]만으로는 불충분하다. 요컨대, 어떤 특정한 정보(실데이터, 관리정보)를 기록하기 위한 영역[실데이터영역, 제1 영역(관리정보영역)]을 정의한다고 하는 것은, AV 용도(제1 애플리케이션)에 있어서 데이터의 종류(실데이터, 제1 관리정보)에 따른 기록영역[실데이터영역, 제1 영역(관리정보영역)]을 관리하는 것뿐 아니라, PC 애플리케이션(제2 애플리케이션) 등으로부터의 액세스가 있었다고 해도, 확보된 영역으로의 기입을 억제하지 않으면 안되는 것을 의미한다. 이를 실현하기 위해서는, 논리파일시스템으로부터 본 경우에 확보된 영역이 이미 데이터가 기록되어 있고, 사용되고 있는 것으로서 보이지 않으면 안된다.

그래서, 본 발명에서는 임의의 수의 어떤 특정한 정보를 기록하기 위한 연속영역을 확보하기 위해서, 영역내에 데이터가 기록되어 있지 않은 미사용 부분을 관리하는 형태로 더미 파일을 작성하는 것으로 한다. 요컨대, 영역내의 미사용 부분에 더미 파일이 관리하는 디스크상의 데이터가 마치 기입되어 있는 것처럼 함을 의미한다.

UDF에서는, 볼륨 레벨의 관리정보와 파일시스템 레벨의 관리정보로 분리되고, 여기서는 관련되는 파일시스템 레벨의 관리정보에 관해 간단히 설명한다. 볼륨 레벨의 관리정보에 의해 정의되는 파티션내에서, 빈 공간을 관리하는 스페이스 비트맵 디스크립터(Space Bitmap Descriptor)와, 파티션내의 디렉토리 구조의 기본정보나 루트(ROOT) 디렉토리를 관리하는 파일 엔트리(File Entry)로의 포인터 정보를 포함하는 파일 세트 디스크립터(File Set Descriptor), 파일 세트 디스크립터가 종료한 것을 나타내는 터미네이팅 디스크립터(Terminating Descriptor), 루트 디렉토리의 파일 엔트리가 파일시스템 레벨의 관리정보로서 기본적으로 기록된다.

스페이스 비트맵 디스크립터란, 파티션내에서 관리되는 모든 논리블록에 대해 1 bit를 각각 할당하여, 대응하는 논리블록이 이용되고 있는지 이용되고 있지 않은지를, bit 정보가 0인지 1인지를 판단하는 것이다. 여기서의 논리블록이란, 파일시스템에 있어서의 최소 액세스 단위이고, UDF 파티션내에서 각 논리블록에 대하여 승순(昇順)으로 논리블록번호가 부가된다.

디렉토리가 정의되면, 디렉토리를 관리하는 파일 엔트리가 디렉토리내에 포함되는 파일이나 디렉토리에 대응하는 파일 아이덴티파이어 디스크립터(File Identifier Descriptor)의 집합의 기록위치를 관리하게 된다. 파일 아이덴티파이어 디스크립터에는, 파일이나 디렉토리의 이름이나 속성정보에 대응하는 파일 엔트리가 기록되어 있는 어드레스정보가 저장된다. 파일 엔트리에는, 날짜 등의 속성정보와 이 파일(파일 엔트리)이 관리하고 있는 디스크상의 데이터의 위치정보, 또는 디렉토리의 경우에는, 디렉토리정보(파일 아이덴티파이어 디스크립터의 경우)가 기록되어 있는 위치정보를 관리하고 있다. 이와 같이, 파일 엔트리와 파일 아이덴티파이어 디스크립터를 트래킹(tracking)해 가는 것에 의해 디렉토리 계층을 파악하는

것이 가능해진다. 또한 이들의 정보는, UDF 파일시스템의 관리정보이고, 통상은 파일시스템의 드라이버가 이용할 뿐으로, 통상은 유저에서 보이지 않는 정보로 된다. 도4에 UDF에 있어서의 디렉토리 계층을 표현하기 위한 파일 엔트리와 파일 아이덴티파이어 디스크립터의 관계를 도시한다.

더미 파일에 의한 영역확보의 1개의 예로서, 도2에, 임의의 수의 어떤 특정한 정보를 기록하기 위한 연속영역으로서 영역 1,2,3을 정의하고, 연속영역 2에 파일인 FILE1.DAT, FILE2.DAT, FILE3.DAT가 기록되어 있는 모양을 도시한다. 실제의 디스크에서는, FILE1.DAT에 대응하는 UDF의 관리정보인 FE(파일 엔트리)와 데이터체인 파일 1이 기록되어 있다. 이 예에서는 파일의 관리정보인 FE(파일 엔트리)로의 포인터정보를 포함하는 파일 아이덴티파이어 디스크립터는 예컨대 영역 1 내에 기록되어 있는 것으로 하여 도시하지 않는다.

스크상의 위치정보는, 개시논리 블록번호와 논리 블록수를 나타내는 얼로케이션 디스크립터(Allocation Descriptor)의 집합에 의해 관리된다. 요컨대, 1개의 파일(파일 엔트리)에 의해 관리되는 데이터가 디스크상에서 연속적으로 기록되어 있는 경우에 얼로케이션 디스크립터는 1개, 2개로 분단하여 기록되어 있는 경우는 얼로케이션 디스크립터 2개에 의해서 파일의 위치정보를 관리하게 된다.

FILE1.DAT와 마찬가지로 FILE2.DAT, FILE3.DAT에 대응하는 FE(파일 엔트리)와 대응하는 실제 데이터가 확보되어 있는 연속영역에 기록되어 있다. 여기서, 확보된 연속영역내에 데이터가 기록되어 있지 않은 미사용 부분에, 더미 파일 DUMMY1.DAT를 작성한다. 요컨대 미사용 부분에 데이터가 마치 기입되어 있는 것과 같이 더미 파일 DUMMY1.DAT를 작성하여, FE(파일 엔트리)의 얼로케이션 디스크립터에 의해 관리하는 것이다. 또한, UDF 파일시스템의 관리정보로 빈 공간을 관리하는 스페이스 비트맵 디스크립터에 관해서도 더미 파일 DUMMY1.DAT에 의해 관리되고 있는 부분은 이용되고 있는 것으로서 취급된다.

또한, 사용과정에서, 예컨대 도3에 있어서의 FILE2.DAT가 삭제된 경우, 통상은 FILE2.DAT에 대응하는 FE(파일 엔트리)와 데이터가 기록되어 있던 개소는 스페이스 비트맵 디스크립터에 의해 미사용 부분이라고 하여, 해방된다. 그러나 본 발명에서는, 도면에 도시된 바와 같이 삭제된 부분은 해방되지 않고, 전술한 더미 파일 DUMMY1.DAT가 관리하는 부분으로서 취급된다. 이 때, DUMMY1.DAT의 FE(파일 엔트리)내의 얼로케이션 디스크립터를 갱신하여, DUMMY1.DAT가 2개의 분단으로 구성되는 파일로 하게 된다.

더미 파일을 작성할 때, 확보된 연속영역을 이용하는 파일시스템의 드라이버가 이해할 수 있는 더미 파일의 이름을 규정하여 놓는다. 도면의 예에서는, 확보된 연속영역내에서의 더미 파일이 DUMMY1.DAT로서 규정되어 있다. 요컨대, 본 발명에 대응하는 파일시스템의 드라이버로부터 보면, 이 DUMMY1.DAT는 빈 공간으로 이해된다.

이와 같이 항상, 확보된 연속영역내의 미사용 부분을 관리하도록 더미 파일을 작성함으로써, 이 영역을 사용하는 애플리케이션 이외에 있어서, 이 영역내는 항상 모두 사용되고 있는 것으로서 취급되어, 데이터가 기입되지 않는다. 또한, 다른 애플리케이션으로부터의 데이터의 기입을 억제할 뿐 아니라, 더미 파일을 규정한 파일시스템의 드라이버에 있어서는, 더미 파일의 정보를 취득함으로써, 확보된 영역내의 미사용 부분의 크기나 위치를 용이하게 파악하는 것이 가능하다고 하는 효과도 있다. 예컨대, 미사용 부분의 크기는, 더미 파일의 파일 사이즈를 참조하면 알 수 있고, 위치정보는 더미 파일의 FE(파일 엔트리)내의 얼로케이션 디스크립터를 참조함으로써 용이하게 파악하는 것이 가능하다. 요컨대, 더미 파일이 그대로, 확보된 영역내의 미사용 부분이 됨을 의미한다.

또한, 더미 파일을 이해할 수 있는 파일시스템의 드라이버가 복수의 메이커에 의해 실장되어 있는 경우, 각 메이커간에 확보되는 연속영역의 크기가 반드시 동일한 것에 한정되지는 않는다. 이 경우에도, 더미 파일의 정보를 참조하는 것만으로, 확보되어 있는 영역내의 미사용 부분을 파악할 수 있기 때문에 영역의 크기를 규정하지 않더라도 메이커간의 호환을 취하는 것도 가능해진다. 또한, 필요에 따라서, 영역내에 기록되어야 되는 파일과 더미 파일의 정보로부터, 영역의 크기를 파악하는 것이 가능해진다.

여기서, 신규로 특정한 정보를 기록하기 위한 소정 사이즈의 연속영역을 확보하는 경우의 처리에 관한 상세를 도5에 나타낸 플로우차트를 사용하여 설명한다. 스텝 S1에 있어서, 연속영역을 확보하는 요구가 발생하면, 스텝 S2에 있어서, UDF 논리파일시스템의 관리정보인 스페이스 비트맵(Space Bitmap) 정보를 참조함으로써 디스크상의 빈 공간을 파악한다. 스텝 S3에 있어서, 확보하고자 하는 연속영역을 디스크상의 임의의 개소에 확보할 수 없는 경우는, 스텝 S5에 있어서 여러 처리를 행하여, 처리를 종료한다. 또한, 스텝 S3에 있어서, 확보하고자 하는 연속영역을 디스크상의 임의의 개소에 확보할 수 있는 경우는, 스텝 S4에 있어서, 확보되는 연속영역을 관리하는 형태로 더미 파일을 작성하여 처리를 종료한다. 구체적인

으로는 더미 파일이 작성되는 디렉토리의 디렉토리 정보에 더미 파일의 파일 엔트리로의 포인터정보인, 파일 아이덴티파이어 디스크립터를 추가하고, 더미 파일의 데이터의 위치정보나 속성정보를 관리하는 파일 엔트리를 기록한다. 이 때, 파일 엔트리내의 오프셋 디스크립터에, 확보된 영역의 위치정보인 개시논리 블록번호와 논리블록수를 기록한다.

여기서, 실제로 파일작성요구가 발생한 경우의 처리에 관해서 설명한다. 파일작성요구가 있는 경우, 그 파일을 어떤 영역에 기입하는 것이 가능한지를 판단한다. 예컨대, 도2에 도시된 바와 같이 영역 1, 영역 2, 영역 3 ...으로 영역을 분할하고 있는 경우에, 임의의 AV 애플리케이션이 파일의 기입 요구를 한 경우, 영역 1에 기입하도록 대응시킨다. 이 판단은, 예컨대 AV 애플리케이션이 DUMMY1.DAT의 위치에 기입하도록, 파일시스템의 드라이버에 지시함으로써 행해진다. 예컨대, AV 애플리케이션 이외의 PC 애플리케이션 등은, DUMMY1.DAT상에 기입하지 않고, 다른 영역 등에 기입되게 된다.

또한, 용도마다 파일종별에 따라서 기입 영역(더미데이터에 대응)을 정의하여 놓음으로써, 예컨대 비디오 데이터는 DUMMY1.DAT상에 기입되는 것으로 하고, 오디오(audio) 데이터는 DUMMY2.DAT상에 기입되도록 제어하는 것도 가능하다.

요컨대, 소정 타입의 파일을 소정의 영역에 기입하는 것이 가능해진다.

다음에, 기입 영역을 결정한 후, 더미 파일에 의해 확보된 영역내에 파일을 작성하는 경우의 처리에 관한 상세를 도6에 나타낸 플로우챗트를 사용하여 설명한다.

스텝 S10에 있어서 파일작성요구가 발생하면, 스텝 S11에 있어서 더미 파일의 정보를 취득한다. 구체적으로는, 더미 파일로서 규정된 파일명으로부터, 대응하는 파일 엔트리를 참조하여 그 정보로부터 더미 파일의 파일 사이즈나 위치정보를 파악한다. 스텝 S12에 있어서, 더미 파일의 파일 사이즈, 요컨대 확보되어 있는 영역내의 빈 사이즈가 작성하려고 하고 있는 파일의 사이즈보다 큰지 어떤지를 판정한다. 더미 파일의 사이즈가 작성하려고 하고 있는 파일의 사이즈보다 작은 경우는, 확보되어 있는 영역에 파일을 작성할 스페이스가 없기 때문에, 스텝 S15에 있어서 에러 처리를 행하여, 처리를 종료한다. 스텝 S12에 있어서, 더미 파일의 사이즈가 작성하려고 하고 있는 파일의 사이즈보다 큰 경우는, 확보되어 있는 영역에 파일을 작성할 수 있기 때문에, 스텝 S13에 있어서, 더미 파일이 관리하는 빈 부분에 파일을 작성한다.

구체적으로는, 작성되는 파일이 포함되는 디렉토리의 디렉토리정보에, 파일을 관리하는 파일 엔트리로의 포인터 정보인 파일 아이덴티파이어 디스크립터를 추가하고, 파일의 데이터의 위치정보나 속성정보를 관리하는 파일 엔트리를 기록한다. 스텝 S14에 있어서, 더미 파일의 파일 엔트리를 갱신하여 처리를 종료한다. 이 때, 파일 엔트리내의 오프셋 디스크립터 정보로부터, 신규작성한 파일에 할당된 부분을 빼는 형태로 갱신한다.

더미 파일에 의해 확보된 영역내에 존재하는 파일의 사이즈의 변경을 수반하는 갱신이 발생하는 경우의 처리에 관한 상세를 도7에 나타낸 플로우챗트를 사용하여 설명한다.

스텝 S20에 있어서, 파일 사이즈 변경을 수반하는 파일갱신요구가 발생하면, 스텝 S21에 있어서 파일의 사이즈가 커지는 것인지 작아지는 것인지를 판정한다. 파일의 사이즈가 축소되는 경우는, 스텝 S28에 있어서, 축소되는 파일의 파일 엔트리의 오프셋 디스크립터를 축소된 데이터를 관리하도록 갱신한다. 스텝 S29에 있어서 더미 파일의 파일 엔트리의 오프셋 디스크립터에 축소된 파일의 개소를 추가하는 형태로 갱신하여 처리를 종료한다. 사이즈가 확장되는 경우는, 스텝 S22에 있어서 더미 파일의 정보를 취득하여, 확보되어 있는 영역내의 빈 부분을 파악한다. 스텝 S23에 있어서, 더미 파일의 사이즈 요컨대 확보되어 있는 영역내의 미사용 스페이스의 크기와 새롭게 파일로서 확장되는 데이터 사이즈의 비교를 행한다. 더미 파일의 사이즈쪽이 큰 경우, 요컨대 확보되어 있는 영역내에 충분한 빈 부분이 있는 경우, 스텝 S24에 있어서, 더미 파일에 의해 관리되는 미사용 스페이스에 확장되는 데이터를 기록한다. 스텝 S25에 있어서, 확장된 파일의 파일 엔트리의 오프셋 디스크립터를 갱신한다. 스텝 S26에 있어서 더미 파일의 파일 엔트리의 오프셋 디스크립터를 갱신하여 처리를 종료한다. 스텝 S23에 있어서, 더미 파일의 사이즈보다 확장되는 데이터의 사이즈가 큰 경우, 요컨대 확장되는 만큼의 빈 스페이스가 남아 있지 않은 경우는, 스텝 S27에 있어서 에러 처리를 행하여 처리를 종료한다.

더미 파일에 의해 관리된 영역내에 존재하는 파일을 삭제하는 경우의 처리에 관한 상세를 도8에 나타낸 플로우챗트를 사용하여 설명한다. 스텝 S30에 있어서, 파일삭제요구가 발생하면, 스텝 S31에 있어서 파일의 삭제처리를 행한다. 구체적으로는, 삭제되는 파일이 포함되어 있는 디렉토리의 디렉토리 정보로부터 대응하는 파일 아이덴티파이어 디스크립터를 삭제한다. 스텝 S32에 있어서 더미 파일의 파일 엔트리를 갱신하여 처리를 종료한다. 구체적으로는, 삭제된 파일이 기록되어 있던 부분을 관리하도록, 더미 파일의 오프셋 디스크립터를 갱신한다.

더미 파일에 의해 확보된 영역이 부족하게 됨으로써, 영역의 크기를 확장하고 싶은 경우의 처리에 관한 상세를 도9에 나타낸 플로우차트를 사용하여 설명한다.

스텝 S40에 있어서, 영역확장요구가 발생하면, 스텝 S41에 있어서 스페이스 비트맵으로부터 디스크의 빈 스페이스를 파악한다. 스텝 S42에 있어서, 확장되는 영역의 인접 개소에 확장하고 싶은 만큼의 연속영역이 존재하는지 어떤지를 판정한다. 존재하는 경우는 스텝 S44에 있어서 더미 파일의 파일 엔트리를 그 확장영역도 포함시키는 형태로 갱신한다. 인접 개소에 확보하고 싶은 만큼의 연속영역이 없는 경우는, 도9에 도시된 연속영역 확보처리를 행하여 처리를 종료한다. 이 경우는, 확장되는 영역이 기존의 영역과 연속적이지는 않게 된다. 만일 연속영역이 아니면 안되는 경우는, 스텝 S43에 있어서 연속영역 확보처리를 행하지 않고, 에러처리를 행하여 처리를 종료하더라도 상관없다. 도10에 임의의 영역을 확장한 경우의 예를 도시한다. 이 예에서는, 영역 1에 FILE1.DAT, FILE2.DAT, FILE3.DAT를 기록하여 빈 공간을 없애고, 인접한 개소를 이용하여 영역을 확장하려고 하였지만, 이미 인접영역 2가 확보되어 있던 관계로, 디스크상의 다른 연속영역을 확보한 예이다. 영역 1의 더미 파일은 DUMMY1.DAT라는 파일명으로, 영역 1의 확장영역은 DUMMY2.DAT라는 더미 파일에 의해 빈 스페이스가 관리되고 있다. 여기서, 파일명이 규정되어 있기 때문에, DUMMY2.DAT가 존재하는 경우는 디스크상의 비인접 개소에 확장된 연속영역이 있는 것을 파악하는 것이 가능하게 된다. 또한, 기존의 연속영역을 확장하고 싶지만 인접 개소가 사용되고 있기 때문에 연속적으로 확장할 수 없는 경우, 인접하는 데이터를 디스크상의 다른 개소로 이동시켜, 영역을 비워서 연속적인 확장을 행하더라도 좋다.

이와 같이, 논리파일시스템의 기능인 파티션 기능을 이용하고 있지 않기 때문에, 파티션의 크기 등을 변경하는 경우와 비교하여, 더미 파일의 갱신만으로 영역의 크기를 변경하는 것이 가능해지기 때문에, 훨씬 단순한 처리로 행할 수 있다.

또, 정의되는 연속영역에서 더미파일로 빈 스페이스를 관리할 때, 도11에 도시된 바와 같이 반드시 연속영역의 최후의 논리블록을 더미 파일에 의해 관리하지 않으면 안된다고 하는 제한을 부가하더라도 좋다. 요컨대, 연속영역에서 최후의 1 논리블록은, 더미 파일이 관리하는 스페이스로서 항상 확보된다. 이에 의하여, 연속영역으로서 확보하고 싶은 크기에, 1 논리블록과 더미 파일을 관리하는 파일 엔트리를 기록하는 1 논리블록의 합계 2 논리블록을 여분으로 확보하여 놓게 된다. 이에 의해서, 더미 파일 DUMMY1.DAT를 관리하는 파일 엔트리내의 오프셋 디스크립터중에서, 가장 큰 논리블록번호가 확보되어 있는 연속영역의 최후미로 된다. 이와 같이 구성함으로써, 연속영역의 최후까지 데이터 파일이 기록된 경우에, 연속영역의 경계를 모르게 되는 것을 방지하여, 연속영역의 경계의 파악을 용이하게 하는 것이 가능해진다.

또한 확보되는 연속영역내에 기록되는 파일이 결정되어 있는 경우는, 그들의 파일 엔트리내의 오프셋 디스크립터와 더미 파일의 정보로부터 연속영역의 범위를 파악하는 것이 가능하지만, 그 연속영역에 기록되는 파일명 등이 결정되어 있지 않은 경우는, 파일명으로부터 어떤 연속영역에 기록되어 있는 데이터인지를 판단하는 것이 가능하지 않기 때문에, 더미 파일에 의해서 빈 스페이스를 파악하는 것은 가능하지만, 확보된 연속영역의 범위를 파악하는 것은 할 수 없다. 이에 의하여 이러한 경우, 도12에 도시된 바와 같이 반드시 연속영역의 선두와 최후미의 논리블록을 더미 파일에 의해 관리하지 않으면 안된다고 하는 제한을 부가하더라도 좋다. 요컨대, 연속영역에서 선두와 최후미의 1 논리블록은, 더미 파일이 관리하는 스페이스로서 항상 확보된다. 이에 의하여, 연속영역으로서 확보하고 싶은 크기에 여분으로 2 논리블록과 더미 파일의 파일 엔트리를 기록하는 1 논리블록의 합계 3 논리블록을 확보하여 놓게 된다.

또는, 도13에 도시된 바와 같이, 확보되는 연속영역의 선두는 더미 파일을 관리하는 파일 엔트리가 아니면 안되고, 최후미의 1 논리블록은 더미 파일이 관리하는 스페이스로서 확보되도록 하더라도 상관없다.

이와 같이 구성함으로써, 더미 파일 DUMMY1.DAT를 관리하는 파일 엔트리의 오프셋 디스크립터중에서 가장 작은 논리블록번호 또는 더미 파일의 파일 엔트리 자체가 기록되어 있는 논리블록번호가 확보되어 있는 연속영역의 선두위치로 되고, 가장 큰 논리블록번호가 반대로 연속영역의 최후미의 위치로 되어, 관리되고 있는 연속영역의 범위를 용이하게 파악하는 것이 가능해진다.

여기서, 본 발명의 실제의 적용예에 관해서 도14를 사용하여 설명한다. 이 도면에서는, AV 데이터를 기록하는 AV 용도에 있어서의 영역관리의 일례이다. 우선, 영역으로서 파일시스템 관리정보영역, 애플리케이션 관리정보영역, 데이터영역으로 분할하여 관리하는 것으로 한다. 파일시스템 관리정보영역에는, UDF 파일시스템에 있어서의 기본관리정보인, 스페이스 비트맵 디스크립터(SBD), 파일 세트 디스크립터(FSD), 터미네이팅 디스크립터(TD)와 AV 애플리케이션에 의해 규정되는 디렉토리의 파일 엔트리(FE)와, 디렉토리내에 정의되는 파일이나 디렉토리를 관리하는 파일 엔트리로의 포인터 정보인, 파일 아이덴티파이어 디스크립터(FID)를 기록하는 것으로 한다. 이 도면의 예에서는, 루트 디렉토리, AV 데이터의 관리정

보를 기록하는 AVMAN 디렉토리, AV 데이터를 기록하는 AVDAT 디렉토리를 관리하는 파일 엔트리와, 각각의 디렉토리 내에서 정의되는 파일의 파일 아이덴티파이어 디스크립터(FID)의 집합이 기록되어 있다. 그리고, 더미 파일 LOGI.DMY가 파일시스템 관리정보영역에서의 미사용 부분을 관리하고 있다.

애플리케이션 관리정보영역은, AV 데이터를 재생하기 위한 관리정보 등의 AV 애플리케이션의 관리정보가 기록되는 영역이다. 이 영역에는, AV 데이터의 관리정보인, AVINFO1.DAT 및 AVINFO2.DAT가 기록되어 있다. 디스크에는 각각에 대응하는 파일 엔트리와 데이터의 실체가 기록되어 있다. 파일시스템 관리정보영역과 마찬가지로, 더미 파일 APPL.DMY가 애플리케이션 관리정보영역에서의 미사용 부분을 관리하고 있다. 이러한 구성의 디스크를 유저가 액세스하면, 도15에 도시된 바와 같은 디렉토리계층으로서 보이게 된다.

데이터영역에는, AV 데이터인 AV1.DAT 및 AV2.DAT가 기록되어 있고, 디스크에는 각각에 대응하는 파일 엔트리와 그 실체가 기록되어 있다. 이 도면의 예에서는, 데이터영역에 관해서는, 더미 파일에 의한 데이터영역의 영역확보는 하고 있지 않지만, 필요에 따라서 더미 파일을 정의하여 영역을 확보하더라도 상관없다.

이와 같이, 파일시스템 관리정보영역의 미사용 개소는, 더미 파일 LOGI.DMY에 의해서 관리되고, 애플리케이션 관리정보 영역의 미사용 개소는 더미 파일 APPL.DMY에 의해서 관리되고 있고, 이들의 영역을 사용하는 AV 애플리케이션에 있어서는, 각각 대응하는 영역의 빈 부분을 관리하고 있는 것으로서 취급된다. 예컨대 이러한 상황의 디스크를 PC에 접속된 디스크 드라이브에서 액세스한 경우에는, 파일시스템 관리정보영역과 애플리케이션 관리정보영역에는, 빈 스페이스가 없는 상태로 파일이 기록되어 있는 것과 같이 취급되기 때문에, 영역이 확보되어 있는 것과 등가가 된다. 이에 의하여, 이들의 영역에 데이터가 기입되지 않게 된다.

또한, 이 때 PC 애플리케이션으로부터 AV 애플리케이션에서 사용되고 있는 디렉토리나 AV 데이터의 관리정보 등이 삭제되어서는 곤란하기 때문에, 이들의 파일이나 디렉토리의 속성으로서 기입 금지를 설정하더라도 좋다. 요컨대, PC 애플리케이션은 데이터영역의 빈 공간 밖에 기입할 수 없는 것으로 된다. 또한, 당연 더미 파일 자체가 삭제되더라도 곤란하기 때문에, 마찬가지로 기입 금지속성을 설정하더라도 좋다. 또한 더미 파일은 PC 애플리케이션으로부터 보이지 않도록 하기 위해서, 숨김(hidden) 파일속성을 설정하더라도 좋다. 만약 더미 파일이 잘못하여 삭제된 경우에는, 현재 정의되어 있는 파일이나 디렉토리의 논리파일시스템의 관리정보와 빈 공간을 관리하고 있는 스페이스 비트맵 디스크립터의 정보로부터 더미 파일을 재구축하는 것도 가능하다.

발명의 효과

이상과 같이, 본 발명에 따른 파일관리방법은, 논리파일시스템에 의한 파티션의 구분을 하지 않고, 소정의 영역내에, 소정의 데이터 이외를 기입할 수 있기 때문에, PC 용도나 AV 용도 등의 상이한 플랫폼에 있어서, 공통이용할 수 있는 범용의 디스크매체에 이용할 수 있다.

도면의 간단한 설명

도1은, 본 발명에 있어서의 일 실시예인 디스크장치의 구성을 도시하는 블록도이고,

도2는, 본 발명의 영역관리방법의 실시예에 있어서 더미 파일에 의해 영역을 확보하고 있는 모양을 도시하는 설명도이고,

도3은, 본 발명의 영역관리방법의 실시예에 있어서 파일이 삭제되었을 때의 더미 파일의 모양을 도시하는 설명도이고,

도4는, 본 발명의 영역관리방법의 실시예에 있어서 UDF 관리정보의 관계를 도시하는 설명도이고,

도5는, 본 발명의 영역관리방법의 형태에 있어서 연속영역을 확보하는 처리의 흐름을 도시하는 플로우차트이고,

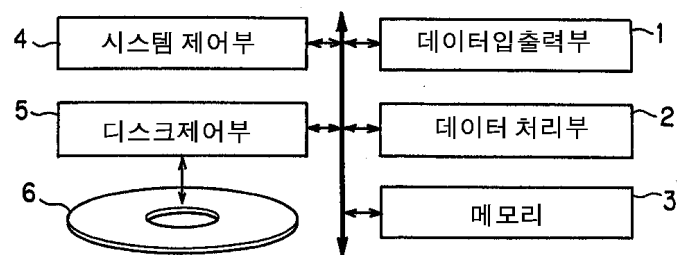
도6은, 본 발명의 영역관리방법의 실시예에 있어서 확보된 연속영역내에 파일을 작성할 때의 처리의 흐름을 도시하는 플로우차트이고,

도7은, 본 발명의 영역관리방법의 실시예에 있어서 확보된 연속영역내의 파일의 크기의 변경을 수반하는 갱신이 있는 경우의 처리의 흐름을 도시하는 플로우차트이고,

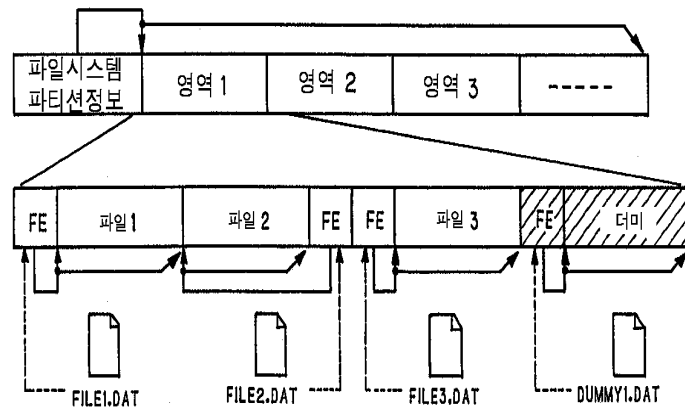
- 도8은, 본 발명의 영역관리방법의 실시예에 있어서 확보된 연속영역내의 파일을 삭제할 때의 처리의 흐름을 도시하는 플로우차트이고,
- 도9는, 본 발명의 영역관리방법의 실시예에 있어서 확보된 연속영역을 확장할 때의 처리의 흐름을 도시하는 플로우차트이고,
- 도10은, 본 발명의 영역관리방법의 실시예에 있어서 확보된 연속영역을 확장할 때의 예를 도시하는 설명도이고,
- 도11은, 본 발명의 영역관리방법의 실시예에 있어서 영역내의 더미 파일이 반드시 최후의 논리블록을 관리하는 모양을 도시하는 설명도이고,
- 도12는, 본 발명의 영역관리방법의 실시예에 있어서 영역내의 더미 파일이 반드시 선두와 최후의 논리블록을 관리하는 모양을 도시하는 설명도이고,
- 도13은, 본 발명의 영역관리방법의 실시예에 있어서 영역내의 더미 파일이 반드시 최후의 논리블록을 관리하고, 더미 파일의 파일 엔트리가 최초의 논리블록에 배치되는 모양을 도시하는 설명도이고,
- 도14는, 본 발명의 영역관리방법의 실시예에 있어서의 실제의 적용예를 도시하는 설명도이고,
- 도15는, 본 발명의 영역관리방법의 실시예에 있어서 도12의 예의 디렉토리 계층을 도시하는 설명도이고,
- 도16은, 종래 기술에 있어서의 파티션에 의한 영역확보의 모양의 설명도이고,
- 도17은, 종래 기술에 있어서의 영역의 위치정보를 기록하는 파일에 의한 영역확보의 모양의 설명도이고,
- 도18은, 종래 기술에 있어서의 파일단위에서의 더미 파일에 의한 영역확보의 모양의 설명도이다.

도면

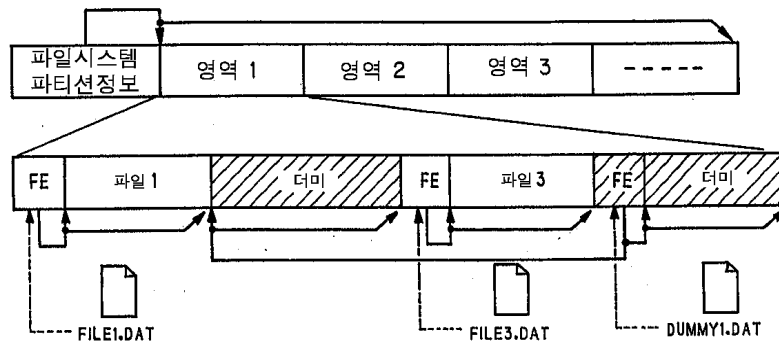
도면1



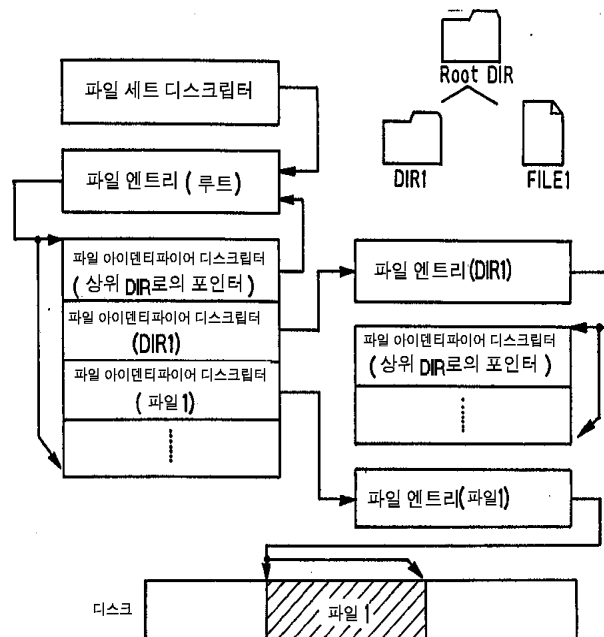
도면2



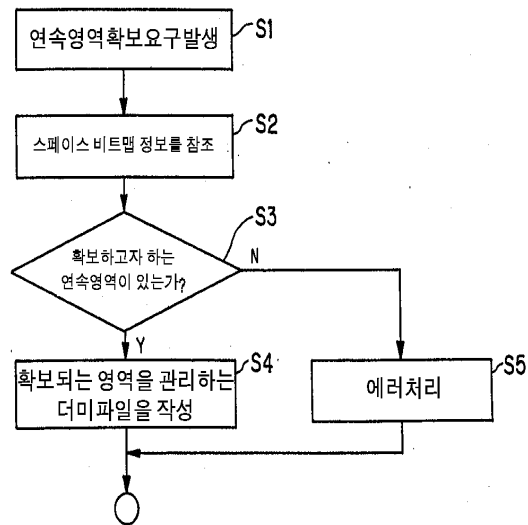
도면3



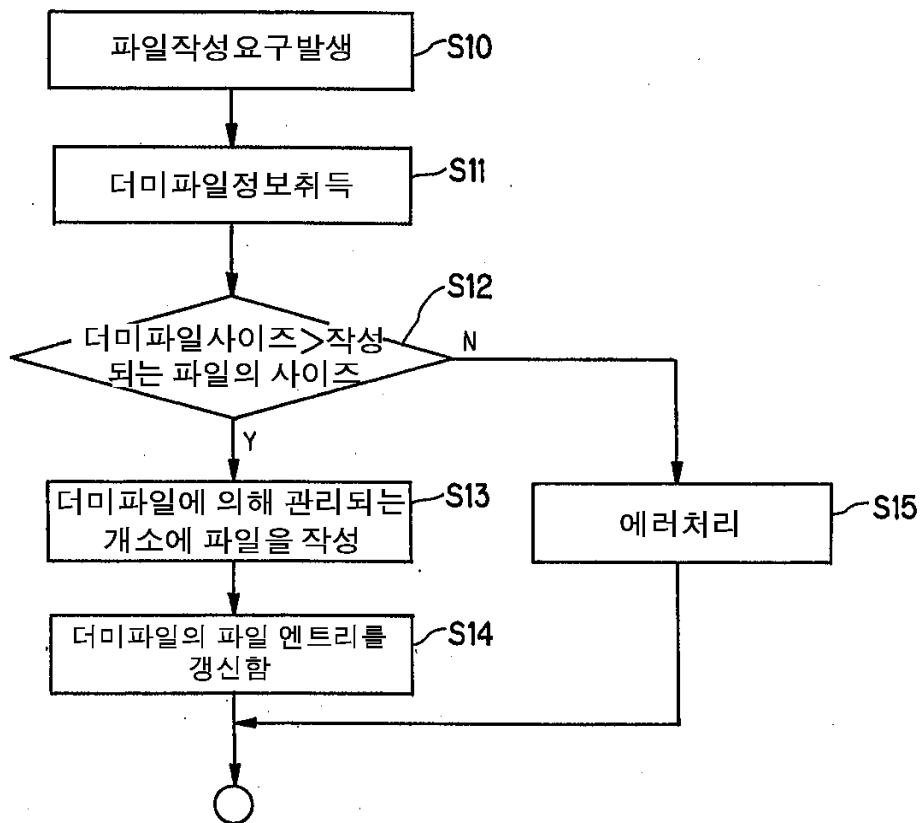
도면4



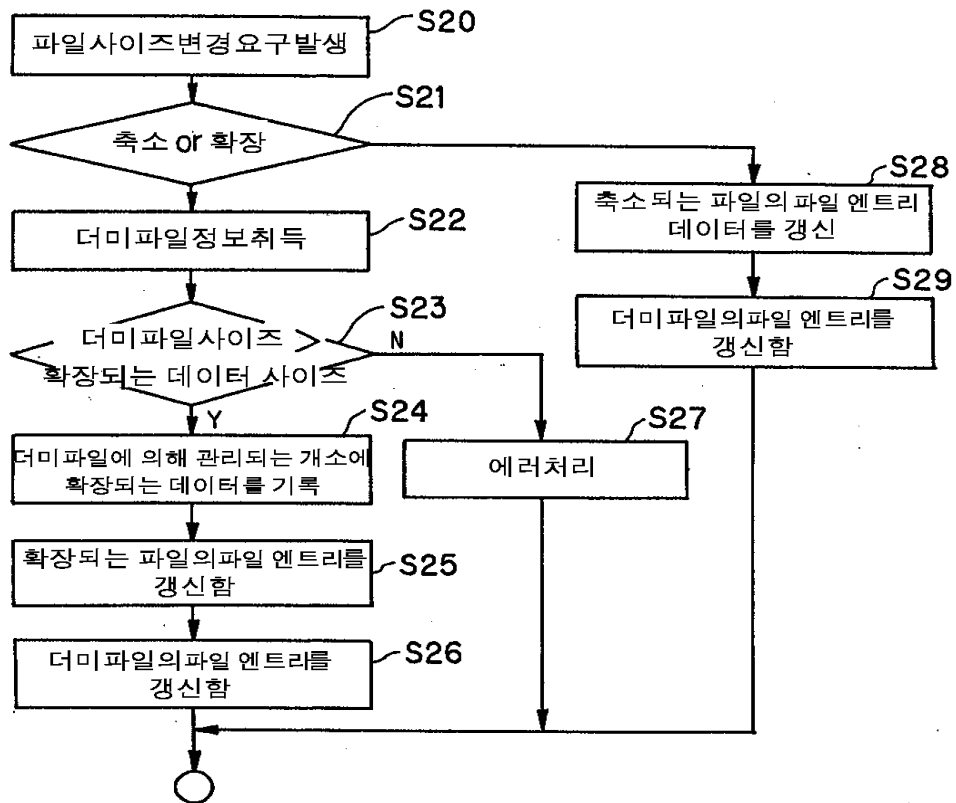
도면5



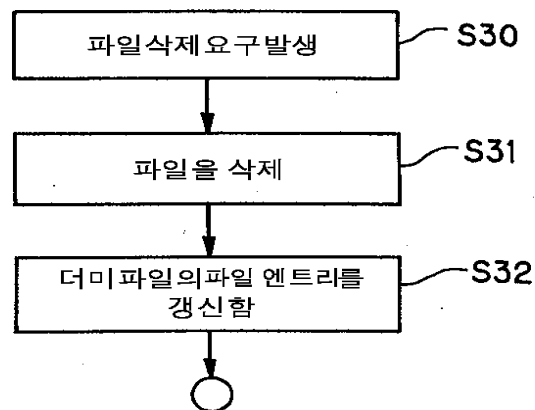
도면6



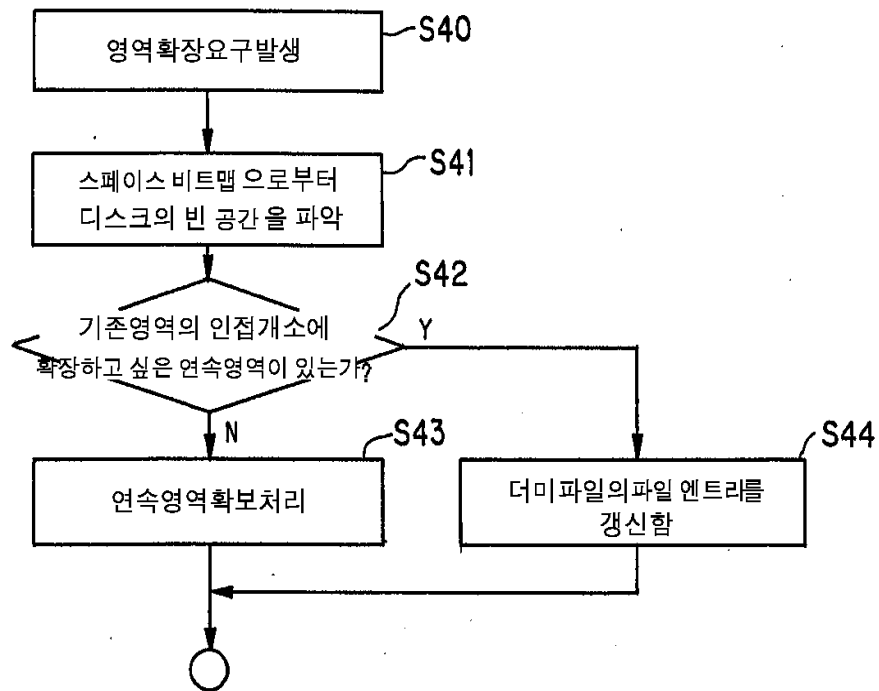
도면7



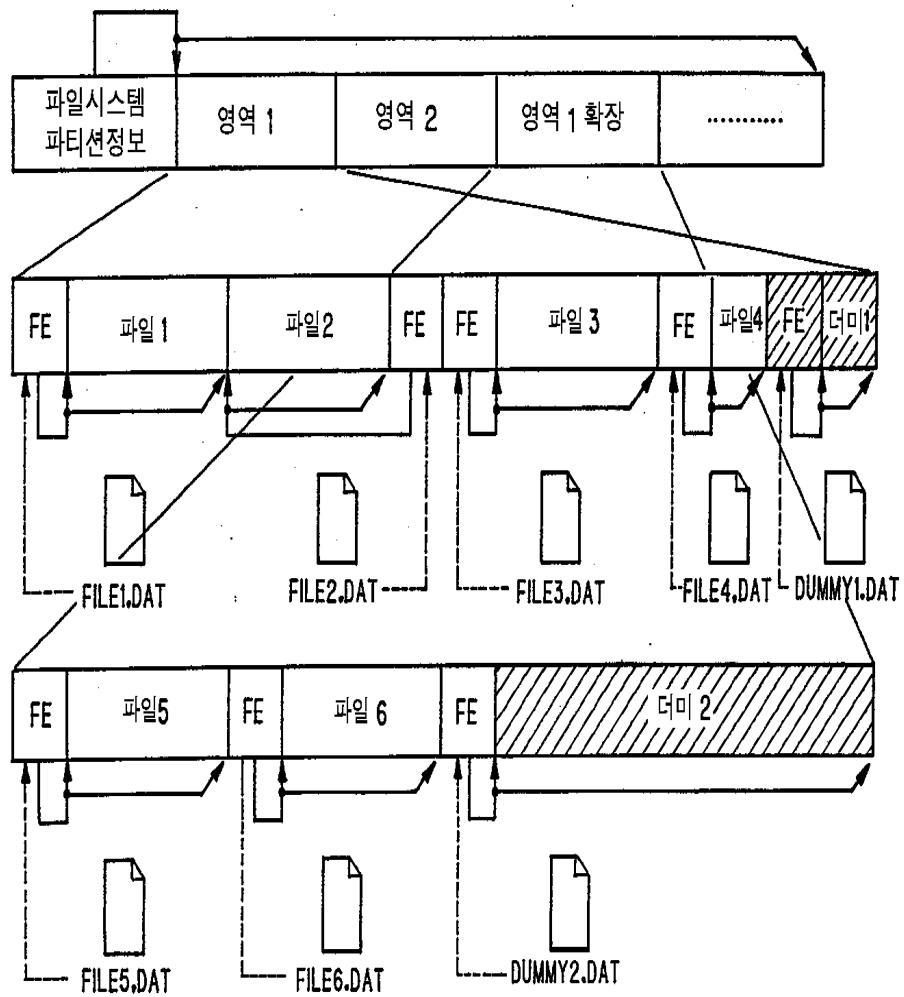
도면8



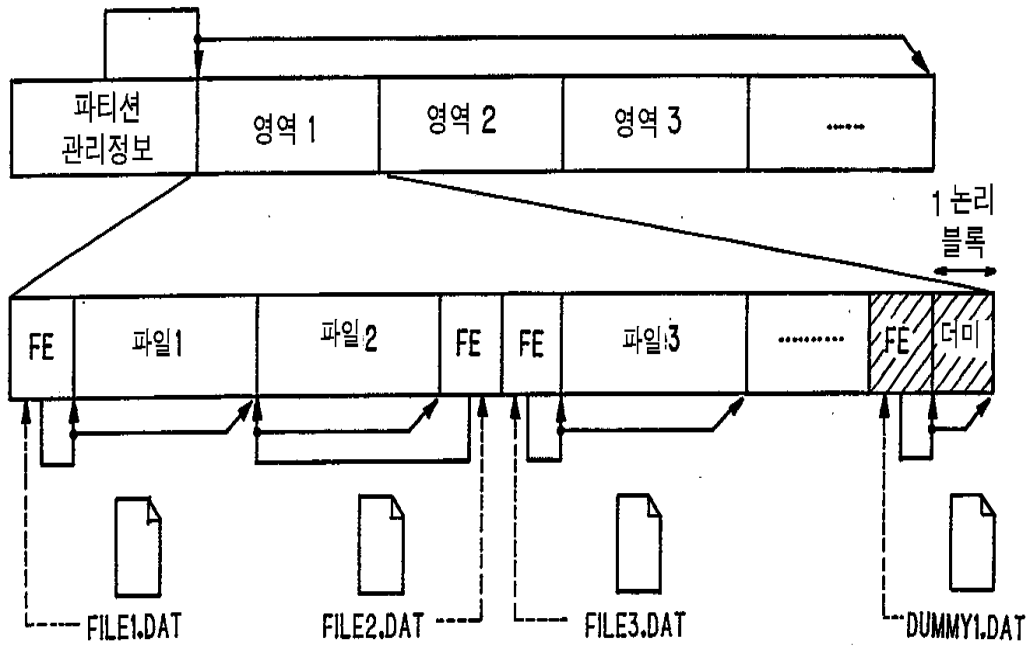
도면9



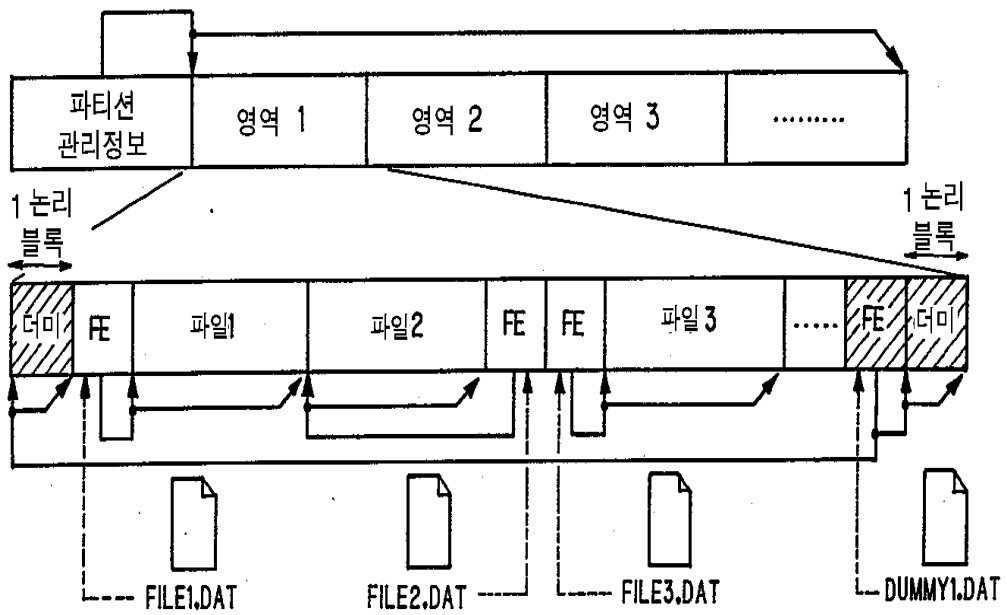
도면10



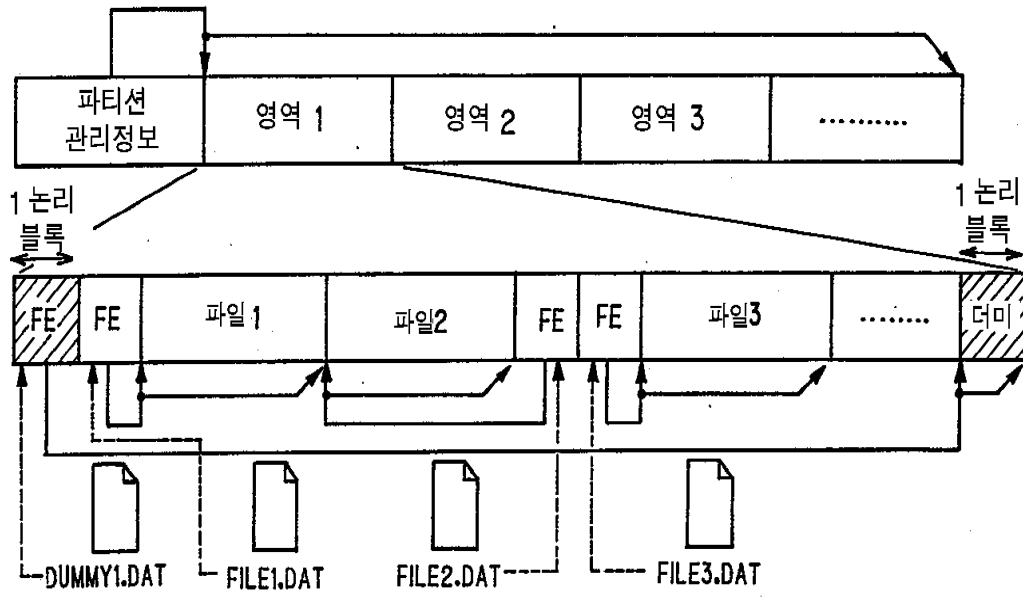
도면11



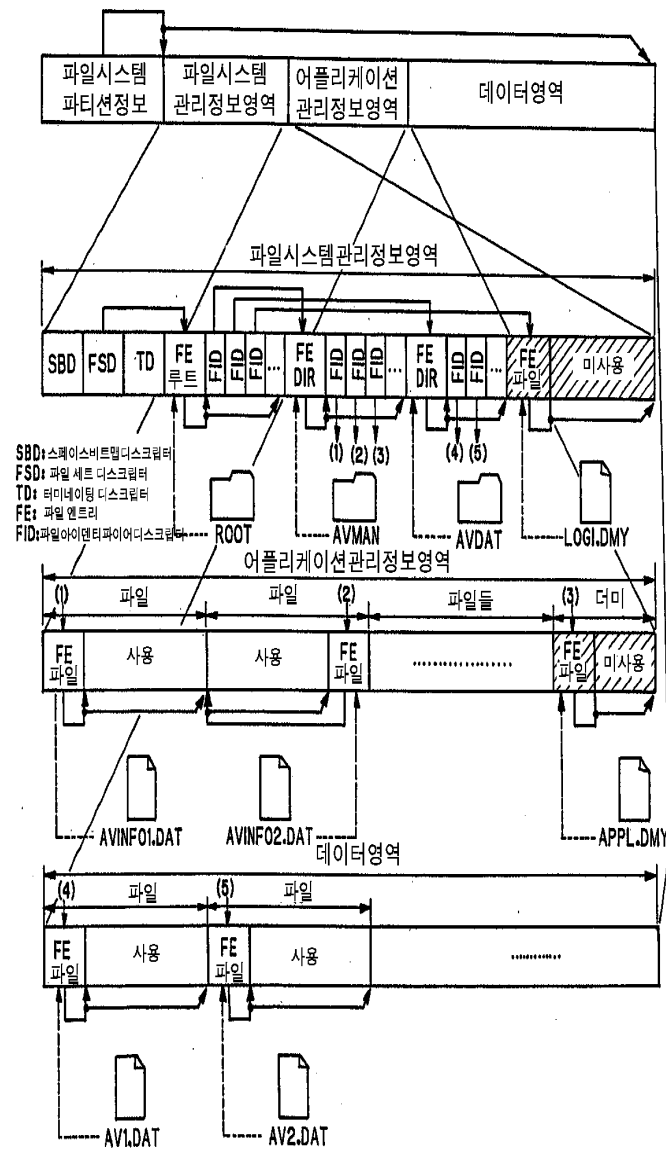
도면12



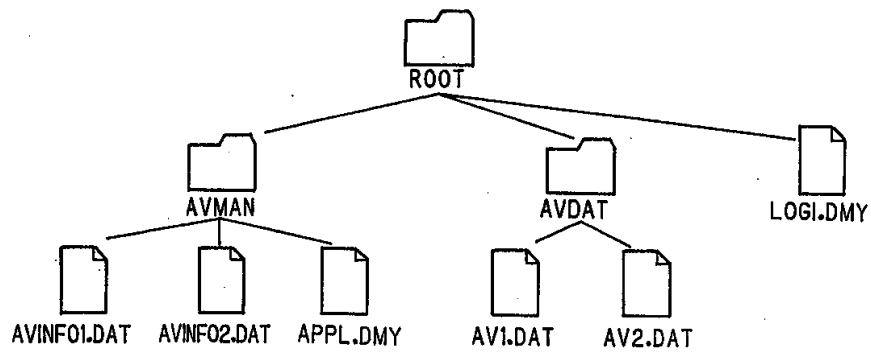
도면13



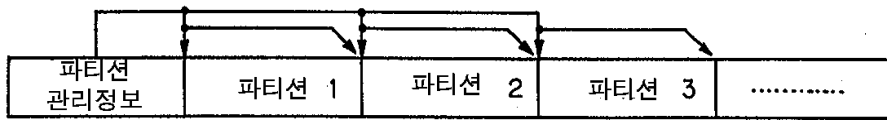
도면14



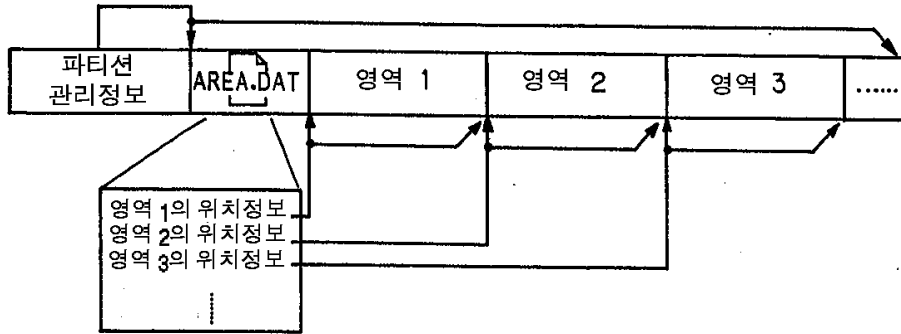
도면15



도면16



도면17



도면18

