

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3607153号

(P3607153)

(45) 発行日 平成17年1月5日(2005.1.5)

(24) 登録日 平成16年10月15日(2004.10.15)

(51) Int. Cl.⁷

F I

G 0 6 F 12/00

G 0 6 F 12/00 5 0 1 H

G 0 6 F 3/06

G 0 6 F 3/06 3 0 1 J

G 1 1 B 20/12

G 1 1 B 20/12

G 1 1 B 27/00

G 1 1 B 27/00 D

請求項の数 16 (全 17 頁)

(21) 出願番号 特願2000-50428 (P2000-50428)
 (22) 出願日 平成12年2月28日(2000.2.28)
 (65) 公開番号 特開2001-243096 (P2001-243096A)
 (43) 公開日 平成13年9月7日(2001.9.7)
 審査請求日 平成16年2月16日(2004.2.16)

早期審査対象出願

(73) 特許権者 000005049
 シャープ株式会社
 大阪府大阪市阿倍野区長池町2番2号
 (74) 代理人 100112335
 弁理士 藤本 英介
 (74) 代理人 100102277
 弁理士 佐々木 晴康
 (74) 代理人 100103296
 弁理士 小池 隆彌
 (74) 代理人 100073667
 弁理士 木下 雅晴
 (72) 発明者 岩野 裕利
 大阪府大阪市阿倍野区長池町2番2号
 シャープ株式会社内

最終頁に続く

(54) 【発明の名称】 ファイル管理方法及び装置

(57) 【特許請求の範囲】

【請求項1】

複数アプリケーションにより使用可能な記憶媒体に関して、該記録媒体に記録する実データを格納するためのデータ領域と該実データに関する管理情報を格納するための領域とを確保し、該実データをファイルとして管理するファイル管理方法であって、
 第1のアプリケーションにより、前記記録媒体内に第1の実データに関する第1の管理情報を格納するための第1の領域を確保する工程と、
 前記第1のアプリケーションにより、前記第1の領域中の未使用領域をダミーファイルとして設定する工程と、を有し、
 前記ダミーファイルは、
 前記第1のアプリケーションによる前記データ領域への前記第1の実データの書込時には使用済み領域を意味し、該第1のアプリケーションによる該第1の実データに関する前記第1の管理情報の書込時には空き領域を意味し、且つ、
 前記第1のアプリケーション以外の第2のアプリケーションによる前記記憶媒体の使用時には使用済み領域を意味するファイル名によって規定されていることを特徴とするファイル管理方法。

【請求項2】

前記管理情報を格納する領域が複数確保され、前記ダミーファイルのファイル名が該管理情報を格納する前記領域毎に異なることを特徴とする請求項1に記載のファイル管理方法。

。

【請求項 3】

前記第 1 の管理情報は、前記第 1 の実データを再生するためのアプリケーション管理情報であることを特徴とする請求項 1 に記載のファイル管理方法。

【請求項 4】

前記第 1 の管理情報は、前記第 1 の実データを再生するためのアプリケーション管理情報のファイルのディレクトリ階層を管理するファイルシステム管理情報であることを特徴とする請求項 1 に記載のファイル管理方法。

【請求項 5】

前記第 1 の管理情報は、前記第 1 の実データのファイルのディレクトリ階層を管理するファイルシステム管理情報であることを特徴とする請求項 1 に記載のファイル管理方法。

10

【請求項 6】

前記第 1 の管理情報は、前記第 1 の実データのファイルと該第 1 の実データを再生するためのアプリケーション管理情報のファイルのディレクトリ階層を管理するファイルシステム管理情報であることを特徴とする請求項 1 に記載のファイル管理方法。

【請求項 7】

前記ダミーファイルのファイル管理情報は、前記第 1 の領域内に存在することを特徴とする請求項 1 に記載のファイル管理方法。

【請求項 8】

前記第 1 の管理情報の書込要求が発生した場合に、前記ダミーファイルのサイズと該書込要求のある前記第 1 の管理情報のサイズとを比較する工程と、
前記第 1 の管理情報を書き込むために必要な前記ダミーファイルのサイズが足りない場合には、前記第 1 の領域の隣接未使用領域を該ダミーファイルに含めることで該ダミーファイルを拡張する工程と、
を有することを特徴とする請求項 1 に記載のファイル管理方法。

20

【請求項 9】

前記第 1 の管理情報の書込要求が発生した場合に、前記ダミーファイルのサイズと該書込要求のある前記第 1 の管理情報のサイズとを比較する工程と、
前記第 1 の管理情報を書き込むために必要な前記ダミーファイルのサイズが足りない場合には、前記第 1 の領域の非隣接未使用領域を非隣接ダミーファイルとして設定する工程と、
を有し、
前記非隣接ダミーファイルのファイル名には、前記ダミーファイルと関連付けられたファイル名を規定することを特徴とする請求項 1 に記載のファイル管理方法。

30

【請求項 10】

前記隣接未使用領域は、前記第 1 の実データ又は第 2 の管理情報を移動させることで作成されることを特徴とする請求項 8 に記載のファイル管理方法。

【請求項 11】

前記第 1 の領域の後端は、常に前記ダミーファイルで構成されることを特徴とする請求項 1 に記載のファイル管理方法。

【請求項 12】

前記第 1 の領域の先端及び後端は、常に前記ダミーファイルで構成されることを特徴とする請求項 1 に記載のファイル管理方法。

40

【請求項 13】

前記ダミーファイルのファイル管理情報には、前記第 2 のアプリケーションに対して隠しファイル又は書き込み禁止ファイルを意味する属性を有することを特徴とする請求項 1 に記載のファイル管理方法。

【請求項 14】

前記第 2 のアプリケーションは、PC アプリケーションであることを特徴とする請求項 13 に記載のファイル管理方法。

【請求項 15】

請求項 1 に記載のファイル管理方法を実行するファイル管理装置。

50

【請求項 16】

前記ダミーファイルを複数識別可能であることを特徴とする請求項 15 に記載のファイル管理装置。

【発明の詳細な説明】

【0001】

本発明は、一つの記録媒体に複数のアプリケーションからのファイルの書き込みを行う場合において、記録媒体を複数の領域に分割して利用するファイル管理方法及び装置に関する。

【0002】

【従来の技術】

PC用途やAV用途などの異なったプラットフォームにおいて、共通利用できる汎用のディスク媒体が存在する事は、ユーザにとって大きなメリットがある。例えば、AVディスクレコーダなどで記録されたディスクが、PCに接続したディスクドライブでアクセスできたり、その逆のアクセスが容易に行える事は、AVディスクレコーダで記録したAVデータをPCでアクセスしたり編集したりする事ができ、編集結果などをまたAVディスクレコーダなどで容易に再生できたりする事を意味する。また1つのディスクに、AVデータを記録したり、PCのアプリケーションソフトを入れたりすると、AV用途とPC用途とで共有して利用する事も考えられる。

【0003】

しかし、AV用途で記録されるデータとPC用途で記録されるデータの性質は異なるものである。例えば、ディスクに記録されたAVデータを再生する場合、AVデータをディスクから読み出し、決まった時間にディスプレイに表示をしなければならない。決まった時間に表示ができない場合は、再生画面が途切れ再生画面がおかしくなる事を意味し、許容されるものではない。

【0004】

ディスク媒体はランダムアクセス性に優れている為、一連のデータであってもディスク上において連続的に配置されている必要はなく、ディスク上の空き領域を有効に利用し、分断して記録されていても構わない。例えばPC用途の場合、文章データファイルがディスク上で分断して記録されていたとしても、そのファイルを読み出す際に各分断点においてシークやトラックジャンプが発生し、その期間ディスクからのデータの読み出しが中断し、連続的にデータが読み出せる場合と比較してデータの読み出し時間が多少かかる程度で、機能面での問題は全く無い。

【0005】

しかし、AV用途においては前述した通り、再生しようとするデータがディスク上で分断して記録されていると、各分断点でデータの読み出しの中断が発生し、問題が発生する可能性がある。一般的に、ディスクから読み出されたAVデータは一度バッファメモリにある程度貯蓄され、シークやトラックジャンプなどのデータ読み出しの中断が発生したとしても、バッファメモリに貯蓄されたデータで補う事により、即再生画面が途切れる事を防いでいる。しかし、バッファメモリによって再生画面の途切れを吸収するのも、繁雑にシークやトラックジャンプが発生すると対応しきれなくなってしまう。よって、AVデータをディスクに記録する場合は、データの読み出し中断が発生するシークやトラックジャンプが発生しない、データの連続的な記録が望まれる事になる。

【0006】

ここで、1つのディスクをAV用途とPC用途とで共用で利用する事を考えると、それぞれの用途のデータを、それぞれ異なる領域に記録したいという要求が発生する。これは、連続的にデータを記録したいAVデータが記録されているディスク領域に、AVデータと比較した場合、データ量が遥かに小さいPC用途のデータが無造作に記録されていくと、AVデータの連続記録を妨げる要因となり、場合によってはAVデータの記録再生に支障を与える可能性があるからである。

【0007】

また、A V用途であっても、記録するデータはA Vデータだけではなく、それらを再生するための管理情報ファイルや、静止画像など色々な種類のデータが存在する。同一用途であっても性質の異なるデータを記録するので、記録するデータの種類に応じて領域を分けて管理したいという要求が発生する。例えば、論理ファイルシステムの管理情報を記録する領域、A Vデータを再生するための管理情報などを記録する領域、A Vデータ自体を記録する領域、静止画を記録する領域と言ったものが考えられる。

【0008】

1つのディスクを複数の領域に分けて使用するために、従来から、以下のような方法がある。まず、第1に、領域を明確に分けるという意味では、論理ファイルシステムのパーティション機能が存在する。例えばA V用途とP C用途用のパーティションを定義する事によって、それぞれの専用領域を設ける事が可能となる。

10

【0009】

図16に示す例では、ディスクをパーティション1、パーティション2、パーティション3と3つに分割している様子を示している。

【0010】

第2の方法として、パーティション機能を用いない場合は、A V用途とP C用途の領域を論理ファイルシステムでは管理せずに、実装レベルのアプリケーション層の管理情報として領域を管理する事が考えられる。例えば、それぞれの領域の位置情報を記録した管理情報ファイルをディスクに記録し、そのファイルを読み込む事で領域の位置情報を把握する事が可能となる。図17に示す例では、AREA.DATというファイルに領域1、領域2、領域3の位置情報が記録されている。よって、AREA.DATというファイルを解釈できるアプリケーションのみが領域の範囲を把握する事が可能となる。

20

【0011】

上記した第1の方法とは異なり、ここでの領域1～3は論理ファイルシステム上では1つのパーティションで構成され、その中で、領域に区分されていることになる。

【0012】

第3に、1つのファイルやデータを記録する領域を予約すると言った用途で、実際にデータを記録する事なくその領域があたかも使われているかのようにダミーファイルを作成する事も考えられる。図18にある領域内に記録されるファイルがそれぞれの使用する予定である領域をDUMMY1.DAT、DUMMY2.DAT、DUMMY3.DATという3つのダミーファイルによって確保しているものである。

30

【0013】

このようにダミーデータとして、予め記録されるデータと同じサイズのデータを書き込んでおき、所定のデータ種別のファイルを書き込む際に、そのダミーデータを削除して、その位置にファイルを書き込むことによって、データ種別ごとにファイルを連続して書き込むことができる。

【0014】

【発明が解決しようとする課題】

上記した第1の方法では、A V用途とP C用途の領域をそれぞれパーティション機能によって分けそれぞれの専用領域を設ける事ができる。しかしながら、一般的にパーティションは、ディスクを初期化する際にパーティションの数や大きさなどを決めて作成するものであり、使用中にパーティションの構成を変更する事は容易に行なえるものではない。これは、パーティション毎に独立した論理的なアドレスや、空き領域の管理情報などを持っているのが一般的なので、構成を変更する場合は、多くの管理情報の再構成を行う必要があるからである。また、ユーザは初期化時にA V用途とP C用途にそれぞれどれぐらいの領域を確保するかを決めなければならない。しかし、使用していく過程で、どちらかの領域が足りなくなってしまう事が考えられる。

40

【0015】

前述の通り、パーティションの大きさなどを変更するのが容易ではないため、例えば、P C用途の領域に空き領域が残っていたとしても、A V用途の領域が一杯になってしまった

50

時点でそれ以上ＡＶデータを記録する事ができなくなり、ディスクを有効に利用する事ができず問題がある。また、同じＡＶ用途においても、論理ファイルシステムの管理情報を記録する領域、ＡＶデータを再生するための管理情報などが記録する領域、ＡＶデータを記録する領域、静止画を記録する領域と言ったように複数の領域に分けて管理を行ないたい場合もあり、これらの領域をそれぞれパーティションで管理を行なうと、パーティションの数が多いため、１つのパーティションが一杯になった場合に、ディスクが空いていたとしても、それ以上データ等が記録できなくなる可能性がそれだけ高くなり問題がある。

【００１６】

第２の方法のように、領域を、論理ファイルシステムで提供するパーティション機能では管理せず、実装レベルのアプリケーション層の管理情報として管理した場合、アプリケーションで領域を管理しているため、想定していないアプリケーションにとっては、この領域の位置情報が記録されている管理情報は何ら意味を持たない。それぞれ確保された領域は、論理ファイルシステムレベルで確保されているわけではないため、想定するアプリケーションのみでそのディスクを使用する場合は問題無いが、想定外のアプリケーションからアクセスされた場合は、領域を認識することができないため、確保した領域内に想定外のデータが記録される可能性があり、問題がある。

【００１７】

また、第３の方法のように、１つのファイルやデータを記録する領域を予約するためにダミーファイルを使う事により、領域を確保する事が可能である。この方法においては、例えば管理情報を記録するための連続領域を確保したい場合、今後記録されるファイルの大きさ、数を予測し、それと同じサイズ、数のダミーファイルを記録することで領域を確保しなければならない。よって、基本的にはファイルの想定される最大サイズでダミーファイルを作成することになり、実際書き込まれるファイルがダミーファイルのサイズより小さい場合、無駄が多く発生するという問題がある。

【００１８】

図１８に示す領域１において、書き込まれるファイルのサイズがそれぞれ１Ｍであると想定した場合、１Ｍのサイズのダミーファイルが作成されることになる。ここに０．８Ｍのファイルを書き込む場合は、まず最初のダミーファイル（ＤＵＭＭＹ１．ＤＡＴ）の位置に０．８Ｍのデータが書き込まれ、次に０．９Ｍのデータが書き込まれる場合は、次のダミーファイルの位置（ＤＵＭＭＹ２．ＤＡＴ）の位置に書き込まれるために、０．１Ｍの無駄な部分が発生するということになる。また、書き込まれるファイルのサイズが予測できない（最大サイズが見込めない）場合には、対応できないという問題がある。

【００１９】

そこで、本願発明は上記３つの方法における課題を解決するものであり、論理ファイルシステムによるパーティションの区分を行うことなく、所定の領域内に、所望のデータ以外が書き込まれることを防止することが可能となる。

【００２０】

【課題を解決するための手段】

本発明は、記録媒体と該記録媒体へのデータの入出力を管理する記録制御手段とを備える記録装置において、記録媒体において少なくとも１以上の領域を確保し、各領域の空き領域をそれぞれダミーファイルで管理する方法であり、当該領域にファイルを書き込む場合には、当該領域のダミーファイル上にファイルを書き込み、ダミーファイルのサイズを該領域の空き領域のサイズに更新することで上記課題を解決するものである。

【００２１】

【発明の実施の形態】

以下、本発明の領域管理方法に関する実施形態について、図面とともに詳細について説明する。本実施例は、ディスク装置として、ＡＶ記録再生を目的としたディスクを用いた携帯型のビデオカメラやビデオデッキ、ＰＣに接続された外部記憶装置などを想定するものである。ディスク媒体は、リムーバブルディスクが好ましいが、ハードディスクなどの据え付け型であっても構わない。また、説明の都合上ディスクに用いる論理ファイルシステ

10

20

30

40

50

ムとしてOSTA(Optical Storage Technology Association)の規格であるUDF(Universal Disk Format)を想定するが、その他の汎用論理ファイルシステムであっても構わない。

【0022】

一般的なディスク装置の構成を図1に示す。データ入力出力部1はカメラなどから入力される映像信号を取り込んだり、再生するデータをモニタ等に出力したりする。データ処理部2は、例えばMPEG符号をエンコードしたり、デコードしたりする信号処理等を行う処理部であり、処理されたデータはメモリ3に格納される。データを記録する場合は、ディスク制御部5において、ディスク6を制御しディスク上の目的の箇所にデータが記録され、再生時には、ディスク6を制御しディスク上の目的の箇所からデータが読み出されてメモリ3に格納される。各処理部はシステム制御部4によって制御される。

10

【0023】

このようなディスク装置において、ディスクの領域を分割して管理したい場合は、論理ファイルシステムにおけるパーティション機能を用いるのが一般的である。しかし、前述したように、一度パーティションを区切ってしまうとパーティションの大きさを変更したり、あるパーティションを他のパーティションと連結させて大きさを拡大すると言った事は容易に行なえない。本実施例ではパーティション機能によつての領域の分割は行なわないので、図2に示すようにディスクのユーザ領域全域を管理するように1つのパーティションを作成する。作成したパーティション内に、ある特定の情報を記録するための領域を定義するものとする。パーティション機能を利用しない事と、汎用ファイルシステムであるUDFを用いるため、1つのパーティション内に複数の領域を管理するような論理ファイルシステムレベルの管理情報は存在しない。ここで重要になってくるのが、1つのディスクを複数のプラットフォームで使用する可能性がある事である。例えば、AV用途で記録されたディスクをPCのドライブでアクセスしたりする事が考えられる。従つて、AVアプリケーションと言つた特定のアプリケーションのみで理解できるような情報での領域確保だけでは不十分である。つまり、ある特定の情報を記録するための領域を定義するという事は、AV用途においてデータの種別に応じた記録領域を管理するだけでなく、PCアプリケーションなどからのアクセスがあつたとしても、確保された領域への書き込みを抑制しなければならない事を意味する。これを実現するためには、確保された領域が論理ファイルシステムから見た場合に既にデータが記録されており、使われているものとして見えなければならない。

20

30

【0024】

そこで、本発明では任意の数のある特定の情報を記録するための連続領域を確保するために、領域内においてデータが記録されておらず未使用な部分を管理する形でダミーファイルを作成するものとする。つまり、領域内の未使用部分にダミーファイルが管理するディスク上のデータがあたかも書き込まれているかのようにするわけである。

【0025】

UDFでは、ボリュームレベルの管理情報とファイルシステムレベルの管理情報に分かれ、ここでは関連するファイルシステムレベルの管理情報について簡単に説明する。ボリュームレベルの管理情報によって定義されるパーティション内において、空き領域を管理するSpace Bitmap Descriptorと、パーティション内のディレクトリ構造の基本情報やROOTディレクトリを管理するFile Entryへのポインタ情報を含むFile Set Descriptor、File Set Descriptorが終了した事を示すTerminating Descriptor、ROOTディレクトリのFile Entryがファイルシステムレベルの管理情報として基本的に記録される。

40

【0026】

Space Bitmap Descriptorとは、パーティション内で管理する全ての論理ブロックに対して1 bitをそれぞれ割り当て、対応する論理ブロックが使われているか使われていないかを、bit情報が0か1かで判断するものである。ここでの

50

論理ブロックとは、ファイルシステムにおける最小アクセス単位であり、UDFパーティション内において各論理ブロックに対して昇順に論理ブロック番号が付加される。

【0027】

ディレクトリが定義されると、ディレクトリを管理するFile Entryがディレクトリ内に含まれるファイルやディレクトリに対応するFile Identifier Descriptorの集合の記録位置を管理する事になる。File Identifier Descriptorには、ファイルやディレクトリの名前や属性情報と対応するFile Entryが記録されているアドレス情報が格納される。File Entryには、日付などの属性情報とこのファイル(File Entry)が管理しているディスク上のデータの位置情報、あるいはディレクトリの場合には、ディレクトリ情報(10 File Identifier Descriptorの集合)が記録されている位置情報を管理している。このように、File EntryとFile Identifier Descriptorを辿って行く事によってディレクトリ階層を把握する事が可能となる。またこれらの情報は、UDFファイルシステムの管理情報であり、通常はデバイスドライバが利用するだけで、通常はユーザから見えない情報となる。図4にUDFにおけるディレクトリ階層を表現するためのFile EntryとFile Identifier Descriptorの関係を示す。

【0028】

ダミーファイルによる領域確保の1つの例として、図2に、任意の数のある特定の情報を記録するための連続領域として領域1、2、3を定義し、連続領域2にファイルであるFILE1.DAT、FILE2.DAT、FILE3.DATが記録されている様子を示す。実際のディスクでは、FILE1.DATに対応するUDFの管理情報であるFE(20 File Entry)とデータ自体であるFile1が記録されている。この例ではファイルの管理情報であるFE(File Entry)へのポインタ情報を含むFile Identifier Descriptorは例えば領域1の中に記録されているものとして図示しない。

【0029】

ディスク上の位置情報は、開始論理ブロック番号と論理ブロック数を示すAllocation Descriptorの集合で管理される。つまり、1つのファイル(File Entry)で管理するデータがディスク上で連続的に記録されている場合、Allocation Descriptorは1つ、2つに分断して記録されている場合はAllocation Descriptor 2つによってファイルの位置情報を管理する30 事になる。

【0030】

FILE1.DATと同様にFILE2.DAT、FILE3.DATに対応するFE(File Entry)と対応する実データが確保されている連続領域に記録されている。ここで、確保した連続領域内でデータが記録されていない未使用な部分に、ダミーファイルDUMMY1.DATを作成する。つまり未使用部分にデータがあたかも書き込まれているようにダミーファイルDUMMY1.DATを作成し、FE(File Entry)のAllocation Descriptorで管理するわけである。また、UDFファイルシステムの管理情報で空き領域を管理するSpace Bitmap Descriptorに関してダミーファイルDUMMY1.DATの管理している部分は使40 われているものとして扱う。

【0031】

また、使用過程において、例えば図3におけるFILE2.DATが削除された場合、通常はFILE2.DATに対応するFE(File Entry)とデータが記録されていた箇所はSpace Bitmap Descriptorによって未使用な部分であると50 し、解放される。しかし本発明では、図に示すように削除された部分は解放せずに、前述したダミーファイルDUMMY1.DATが管理する部分として扱う。この際、DUMMY1.DATのFE(File Entry)内のAllocation Desc

riptorを更新して、DUMMY1.DATが2つの分断から構成されるファイルとする事になる。

【0032】

ダミーファイルを作成するのにあたって、確保した連続領域を利用するアプリケーションが理解できるダミーファイルの名前を規定しておく。図の例では、確保した連続領域内のダミーファイルはDUMMY1.DATとして規定されている。つまり、あるアプリケーションからみると、このDUMMY1.DATは空き領域であると理解するのである。

【0033】

このように常に、確保した連続領域内の未使用部分を管理するようにダミーファイルを作成する事によって、この領域を使用するアプリケーション以外にとって、この領域内は常に全て使用されているものとして扱われ、データが書き込まれる事はない。また、他のアプリケーションからのデータの書き込みを抑制するだけではなく、ダミーファイルを規定したアプリケーションにとっては、ダミーファイルの情報を取得する事によって、容易に確保した領域内の未使用部分の大きさや位置を把握する事ができるという効果もある。例えば、未使用部分の大きさは、ダミーファイルのファイルサイズを参照すればわかり、位置情報はダミーファイルのFE (File Entry)内のAllocation Descriptorを参照する事によって容易に把握する事が可能である。つまり、ダミーファイルがそのまま、確保した領域内の未使用部分になるわけである。

【0034】

また、ダミーファイルを理解できるアプリケーションが複数のメーカーによって実装されている場合、各メーカー間で必ずしも確保する連続領域の大きさが同一であるとは限らない。この場合にも、ダミーファイルの情報を参照するだけで、確保している領域内の未使用部分を把握できるので領域の大きさを規定しなくてもメーカー間の互換も取る事が可能となる。また、必要に応じて、領域内に記録されるべきファイルとダミーファイルの情報から、領域の大きさを把握する事が可能となる。

【0035】

ここで、新規に特定の情報を記録するための所定サイズの連続領域を確保する場合の処理について詳細を図5に示すフローチャートを用いて説明する。ステップS1において、連続領域を確保する要求が発生すると、ステップS2において、UDF論理ファイルシステムの管理情報であるSpace Bitmap情報を参照する事によってディスク上の空き領域を把握する。ステップS3において、確保しようとする連続領域がディスク上の任意の箇所に確保できない場合は、ステップS5においてエラー処理を行ない処理を終了する。また、ステップS3において、確保しようとする連続領域がディスク上の任意の箇所に確保できる場合は、ステップS4において、確保する連続領域を管理する形でダミーファイルを作成し処理を終了する。具体的にはダミーファイルが作成されるディレクトリのディレクトリ情報にダミーファイルのFile Entryへのポインタ情報である、File Identifier Descriptorを追加し、ダミーファイルのデータの位置情報や属性情報を管理するFile Entryを記録する。この際、File Entry内のAllocation Descriptorに、確保した領域の位置情報である開始論理ブロック番号と論理ブロック数を記録する。

【0036】

ここで、実際にファイル作成要求が発生した場合の処理について説明する。ファイル作成要求があった場合、そのファイルをどの領域に書き込むことが可能であるかを判断する。例えば、図2に示すように領域1、領域2、領域3...と領域を分割している場合に、あるAVアプリケーションがファイルの書き込み要求をした場合、領域1に書き込むように対応づけられているとする。この判断は、例えばAVアプリケーションはDUMMY1.DATの位置に書き込むように、AVアプリケーション自体に定義しておくことによって行われる。例えば、AVアプリケーション以外のPCアプリケーションなどは、DUMMY1.DAT上に書き込むことはなく、他の領域などに書き込むことになる。

【0037】

10

20

30

40

50

また、各アプリケーションにファイル種別に応じて書き込む領域（ダミーデータとの対応）を定義しておくことにより、例えばvideoデータはDUMMY1.DAT上に書き込むこととし、audioデータはDUMMY2.DAT上に書き込むように制御することも可能である。

【0038】

つまり、所定のアプリケーションを所定の領域に書き込む、或いは所定の種別のファイルを所定の領域に書き込むことが可能となる。

【0039】

次に、書き込む領域が決定した後、ダミーファイルによって確保された領域内にファイルを作成する場合の処理について詳細を図6に示すフローチャートを用いて説明する。

10

【0040】

ステップS10においてファイル作成要求が発生すると、ステップS11においてダミーファイルの情報を取得する。具体的には、ダミーファイルとして規定されたファイル名より、対応するFile Entryを参照しその情報よりダミーファイルのファイルサイズや位置情報を把握する。ステップS12において、ダミーファイルのファイルサイズ、つまり確保した領域中の空きサイズが作成しようとしているファイルのサイズより大きいかどうかを判定する。ダミーファイルのサイズが作成しようとしているファイルのサイズより小さい場合は、確保された領域にファイルを作成するスペースがないので、ステップS15においてエラー処理を行ない、処理を終了する。ステップS12において、ダミーファイルのサイズが作成しようとしているファイルのサイズより大きい場合は、確保された領域にファイルが作成できるので、ステップS13において、ダミーファイルが管理する空き部分にファイルを作成する。

20

【0041】

具体的には、作成するファイルが含まれるディレクトリのディレクトリ情報に、ファイルを管理するFile Entryへのポインタ情報であるFile Identifier Descriptorを追加し、ファイルのデータの位置情報や属性情報を管理するFile Entryを記録する。ステップS14において、ダミーファイルのFile Entryを更新し処理を終了する。この際、File Entry内のAllocation Descriptor情報から、新規作成したファイルに割り当てた部分を差し引く形で更新する。

30

【0042】

ダミーファイルによって確保された領域内に存在するファイルのサイズの変更を伴う更新が発生する場合の処理について詳細を図7に示すフローチャートを用いて説明する。

【0043】

ステップS20において、ファイルサイズ変更を伴うファイル更新要求が発生すると、ステップS21においてファイルのサイズが大きくなるのか小さくなるのかを判定する。ファイルのサイズが縮小される場合は、ステップS28において、縮小するファイルのFile EntryのAllocation Descriptorを縮小したデータを管理するように更新する。ステップS29においてダミーファイルのFile EntryのAllocation Descriptorに縮小したファイルの箇所を追加する形で更新し処理を終了する。サイズが拡張される場合は、ステップS22においてダミーファイルの情報を取得し、確保されている領域内の空き部分を把握する。ステップS23において、ダミーファイルのサイズつまり確保されている領域内の未使用スペースの大きさと新たにファイルとして拡張されるデータサイズの比較を行なう。ダミーファイルのサイズの方が大きい場合、つまり確保された領域内に十分な空き部分がある場合、ステップS24において、ダミーファイルの管理する未使用スペースに拡張するデータを記録する。ステップS25において、拡張したファイルのFile EntryのAllocation Descriptorを更新する。ステップS26においてダミーファイルのFile EntryのAllocation Descriptorを更新し処理を終了する。ステップS23において、ダミーファイルのサイズより拡張するデータのサイズが大

40

50

きい場合、つまり拡張するだけの空きスペースが残っていない場合は、ステップS27においてエラー処理を行ない処理を終了する。

【0044】

ダミーファイルによって確保された領域内に存在するファイルを削除する場合の処理について詳細を図8に示すフローチャートを用いて説明する。ステップS30において、ファイル削除要求が発生すると、ステップS31においてファイルの削除処理を行なう。具体的には、削除するファイルが含まれているディレクトリのディレクトリ情報から対応するFile Identifier Descriptorを削除する。ステップS32においてダミーファイルのFile Entryを更新し処理を終了する。具体的には、削除したファイルが記録されていた部分を管理するように、ダミーファイルのAllocation Descriptorを更新する。

10

【0045】

ダミーファイルによって確保された領域が足りなくなったりした事により、領域の大きさを拡張したい場合の処理について詳細を図9に示すフローチャートを用いて説明する。

【0046】

ステップS40において、領域拡張要求が発生すると、ステップS41においてSpace Bitmapよりディスクの空きスペースを把握する。ステップS42において、拡張する領域の隣接箇所に拡張したいだけの連続領域が存在するかどうかを判定する。存在する場合はステップS44においてダミーファイルのFile Entryをその拡張領域も含める形で更新する。隣接箇所に確保したいだけの連続領域が無い場合は、図5に示した連続領域確保処理を行ない処理を終了する。この場合は、拡張する領域が既存の領域と連続的では無くなる。もし連続領域でなければならない場合は、ステップS43において連続領域確保処理を行わず、エラー処理を行なって処理を終了しても構わない。図10にある領域を拡張した場合の例を示す。この例では、領域1にFILE1.DAT、FILE2.DAT、FILE3.DATを記録し空き領域が無くなり、隣接した箇所を利用して領域を拡張しようとしたが、既に隣接領域2が確保されていた関係で、ディスク上の他の連続領域を確保した例である。領域1のダミーファイルはDUMMY1.DATというファイル名で、領域1の拡張領域はDUMMY2.DATというダミーファイルによって空きスペースが管理されている。ここで、ファイル名が規定されているので、DUMMY2.DATが存在する場合はディスク上の非隣接箇所に拡張した連続領域がある事を把握する事が可能となる。また、既存の連続領域を拡張したい場合に、隣接箇所が使われているために連続的に拡張できない場合、隣接するデータをディスク上の他の箇所へ移動し、領域を空けて連続的な拡張を行っても良い。

20

30

【0047】

このように、論理ファイルシステムの機能であるパーティション機能を使っていないので、パーティションの大きさなどを変更する場合と比較して、ダミーファイルの更新だけで領域の大きさを変更することが可能となるので、遥かに単純な処理で行える。

【0048】

なお、定義する連続領域においてダミーファイルで空きスペースを管理する際、図11に示すように必ず連続領域の最後の論理ブロックをダミーファイルで管理しなければならないという制限を付けても良い。つまり、連続領域において最後の1論理ブロックは、ダミーファイルが管理するスペースとして常に確保する。よって、連続領域として確保したい大きさに、1論理ブロックとダミーファイルを管理するFile Entryを記録する1論理ブロックの計2論理ブロックを余分に確保しておく事になる。よって、ダミーファイルDUMMY1.DATを管理するFile Entry内のAllocation Descriptorの中で、一番大きい論理ブロック番号が確保している連続領域の最後尾という事になる。このように構成することによって、連続領域の最後までデータファイルが記録された場合に、連続領域の境界がわからなくなるというのを防止し、連続領域の境界の把握を容易に行うことが可能となる。

40

【0049】

50

また確保する連続領域内に記録されるファイルが決まっている場合は、それらのFile Entry内のAllocation Descriptorとダミーファイルの情報から連続領域の範囲を把握する事が可能であるが、その連続領域に記録されるファイル名などが決まっていない場合は、ファイル名からどの連続領域に記録されているデータであるかを判断することができないため、ダミーファイルによって空きスペースは把握する事が可能であるが、確保された連続領域の範囲を把握する事はできない。よってこのような場合、図12に示すように必ず連続領域の先頭と最後尾の論理ブロックをダミーファイルで管理しなければならないという制限を付けても良い。つまり、連続領域において先頭と最後尾の1論理ブロックは、ダミーファイルが管理するスペースとして常に確保する。よって、連続領域として確保したい大きさに余分に2論理ブロックとダミーファイルのFile Entryを記録する1論理ブロックの計3論理ブロックを確保しておく事になる。

10

【0050】

あるいは、図13に示すように、確保する連続領域の先頭はダミーファイルを管理するFile Entryでなければならず、最後尾の1論理ブロックはダミーファイルが管理するスペースとして確保するようにしても構わない。

このように構成することによって、ダミーファイルDUMMY1.DATを管理するFile EntryのAllocation Descriptorの中で一番小さい論理ブロック番号あるいはダミーファイルのFile Entry自体が記録されている論理ブロック番号が確保している連続領域の先頭位置となり、一番大きい論理ブロック番号が逆に連続領域の最後尾の位置となり、管理している連続領域の範囲を容易に把握する事が可能となる。

20

【0051】

ここで、本発明の実際の適用例について図14を用いて説明する。この図では、AVデータを記録するAV用途における領域管理の一例である。まず、領域としてファイルシステム管理情報領域、アプリケーション管理情報領域、データ領域に分割して管理するものとする。ファイルシステム管理情報領域には、UDFファイルシステムにおける基本管理情報である、Space Bitmap Descriptor (SBD)、File Set Descriptor (FSD)、Terminating Descriptor (TD)とAVアプリケーションによって規定されるディレクトリのFile Entry (FE)と、ディレクトリ内に定義されるファイルやディレクトリを管理するFile Entryへのポインタ情報である、File Identifier Descriptor (FID)を記録するものとする。この図の例では、ROOTディレクトリ、AVデータの管理情報を記録するAVMANディレクトリ、AVデータを記録するAVDATディレクトリを管理するFile Entryと、それぞれのディレクトリ内で定義されるファイルのFile Identifier Descriptor (FID)の集合が記録されている。そして、ダミーファイルLOGI.DMYがファイルシステム管理情報領域における未使用部分を管理している。

30

【0052】

アプリケーション管理情報領域は、AVデータを再生するための管理情報などのAVアプリケーションの管理情報が記録される領域である。この領域には、AVデータの管理情報である、AVINFO1.DATおよびAVINFO2.DATが記録されており、ディスクにはそれぞれに対応するFile Entryとデータの実体が記録されている。ファイルシステム管理情報領域と同様に、ダミーファイルAPPL.DMYがアプリケーション管理情報領域における未使用部分を管理している。このような構成のディスクをユーザがアクセスすると、図15に示すようなディレクトリ階層として見える事になる。

40

【0053】

データ領域には、AVデータであるAV1.DATおよびAV2.DATが記録されており、ディスクにはそれぞれに対応するFile Entryとその実体が記録されている。この図の例では、データ領域に関しては、ダミーファイルによるデータ領域の領域確保

50

は行なっていないが、必要に応じてダミーファイルを定義して領域を確保しても構わない。

【 0 0 5 4 】

このように、ファイルシステム管理情報領域の未使用箇所は、ダミーファイル LOGI . DMY によって管理され、アプリケーション管理情報領域の未使用箇所はダミーファイル APPL . DMY によって管理されており、これらの領域を使用する AV アプリケーションにとっては、それぞれ対応する領域の空き領域を管理しているものとして扱われる。例えばこのような状況のディスクを PC に接続されたディスクドライブでアクセスした場合には、ファイルシステム管理情報領域とアプリケーション管理情報領域には、空きスペースが無い状態でファイルが記録されているように扱われるため、領域が確保されているのと等価になる。よって、これらの領域にデータが書き込まれる事が無くなる。

10

【 0 0 5 5 】

また、この時 PC アプリケーションから AV アプリケーションで使用しているディレクトリや AV データの管理情報などが削除されては困るので、これらのファイルやディレクトリの属性として書き込み禁止を設定しても良い。つまり、PC アプリケーションはデータ領域の空き領域にしか書き込むことができないこととなる。また、当然ダミーファイル自体が削除されても困るので、同様に書き込み禁止属性を設定しても良い。またダミーファイルは PC アプリケーションから見えないようにするために、隠しファイル属性を設定しても良い。仮にダミーファイルが誤って消されてしまった場合には、現在定義されているファイルやディレクトリの論理ファイルシステムの管理情報と空き領域を管理している S

20

【 0 0 5 6 】

【 発明の効果 】

本発明によれば、確保した連続領域内の未使用部分を管理するようにダミーファイルを作成する事によって、この領域を使用するアプリケーション以外にとって、この領域内は全て使用されているものとして扱われ、データが書き込まれる事を抑制することが可能となる。

【 0 0 5 7 】

また、ダミーファイルを規定したアプリケーションにとっては、ダミーファイルのファイルサイズや位置情報を取得する事によって、容易に確保した領域内の未使用箇所の大きさや位置を把握する事が可能となり領域内でのファイル作成や更新が容易に行える事になる。

30

【 0 0 5 8 】

また、ダミーファイルを理解できるアプリケーションが複数のメーカー等によって実装されている場合、必ずしも確保する連続領域の大きさが同一であるとは限らない。この場合にも、ダミーファイルの情報を参照するだけで、未使用部分を把握できるのでメーカー間の互換も取る事が可能となる。

【 0 0 5 9 】

また、論理ファイルシステムの機能であるパーティション機能を使っていないので、パーティションの大きさなどを変更する事と比較して、ダミーファイルの更新といった単純な処理だけで領域の大きさを変更することが可能となる。確保した連続領域の範囲を知る必要がある場合は、必要に応じて連続領域の最後尾あるいは先頭と最後尾の両方の論理ブロックをダミーファイルで管理する事によって、容易に管理している連続領域の範囲を把握することが可能となる。

40

【 図面の簡単な説明 】

【 図 1 】 本発明における一実施形態であるディスク装置の構成を示すブロック図である。

【 図 2 】 本発明の領域管理方法の実施形態においてダミーファイルによって領域を確保している様子を示す説明図である。

【 図 3 】 本発明の領域管理方法の実施形態においてファイルが削除された際のダミーファ

50

イルの様子を示す説明図である。

【図４】本発明の領域管理方法の実施形態においてＵＤＦ管理情報の関係を示す説明図である。

【図５】本発明の領域管理方法の実施形態において連続領域を確保する処理の流れを示すフローチャートである。

【図６】本発明の領域管理方法の実施形態において確保した連続領域内にファイルを作成する際の処理の流れを示すフローチャートである。

【図７】本発明の領域管理方法の実施形態において確保した連続領域内のファイルの大きさの変更を伴う更新がある場合の処理の流れを示すフローチャートである。

【図８】本発明の領域管理方法の実施形態において確保した連続領域内のファイルを削除する際の処理の流れを示すフローチャートである。 10

【図９】本発明の領域管理方法の実施形態において確保した連続領域を拡張する際の処理の流れを示すフローチャートである。

【図１０】本発明の領域管理方法の実施形態において確保した連続領域を拡張する際の例を示す説明図である。

【図１１】本発明の領域管理方法の実施形態において領域内のダミーファイルが必ず最後の論理ブロックを管理する様子を示す説明図である。

【図１２】本発明の領域管理方法の実施形態において領域内のダミーファイルが必ず先頭と最後の論理ブロックを管理する様子を示す説明図である。

【図１３】本発明の領域管理方法の実施形態において領域内のダミーファイルが必ず最後の論理ブロックを管理し、ダミーファイルのファイルエントリが最初の論理ブロックに配置される様子を示す説明図である。 20

【図１４】本発明の領域管理方法の実施形態における実際の適用例を示す説明図である。

【図１５】本発明の領域管理方法の実施形態において図１２の例のディレクトリ階層を示す説明図である。

【図１６】従来技術におけるパーティションによる領域確保の様子の説明図である。

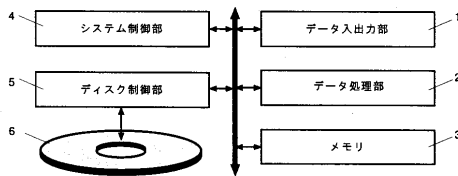
【図１７】従来技術における領域の位置情報を記録するファイルによる領域確保の様子の説明図である。

【図１８】従来技術におけるファイル単位でのダミーファイルによる領域確保の様子の説明図である。 30

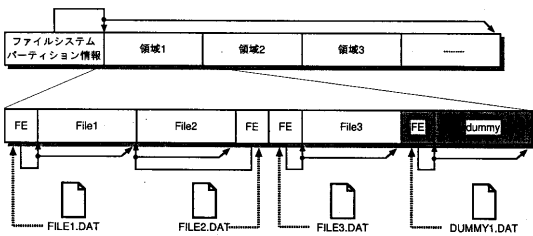
【符号の説明】

- 1 データ入出力部
- 2 データ処理部
- 3 メモリ
- 4 システム制御部
- 5 ディスク制御部
- 6 ディスク

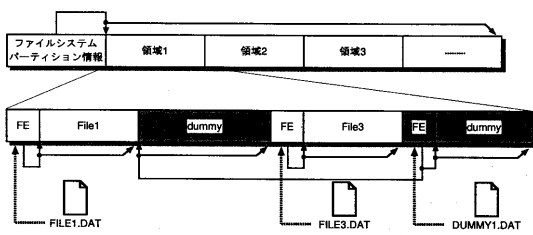
【図 1】



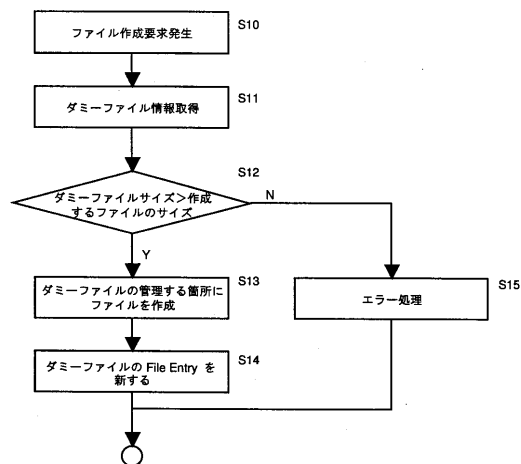
【図 2】



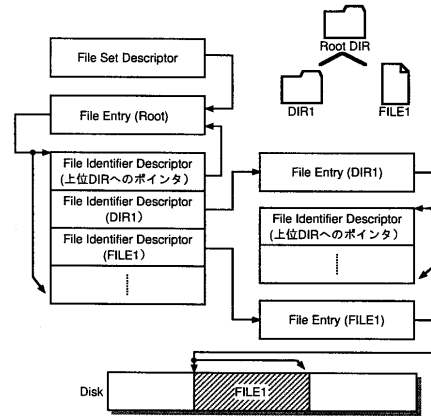
【図 3】



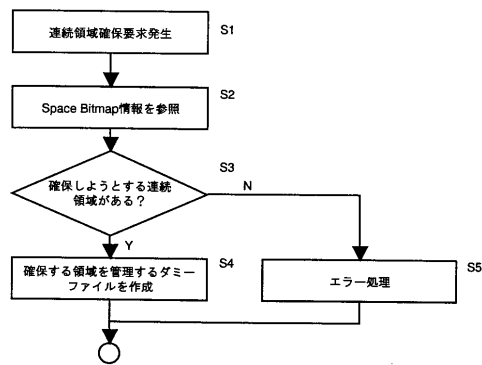
【図 6】



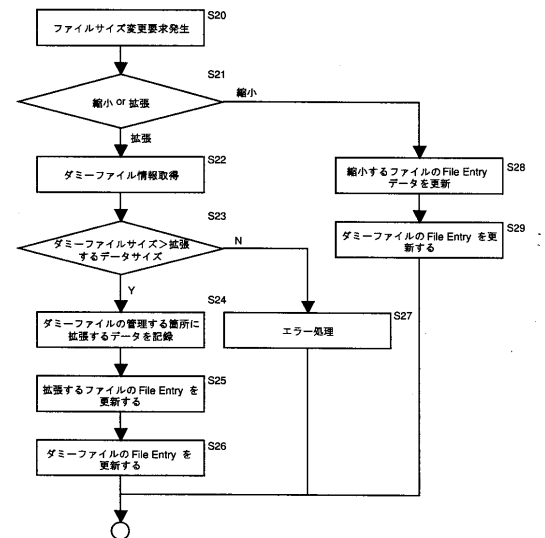
【図 4】



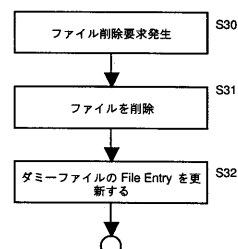
【図 5】



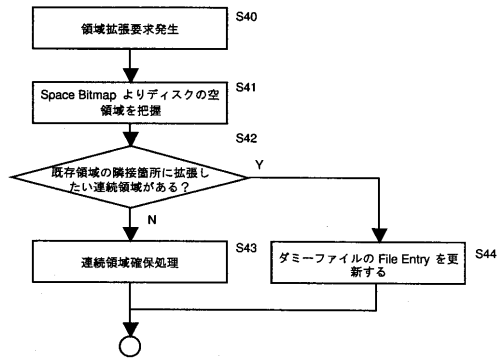
【図 7】



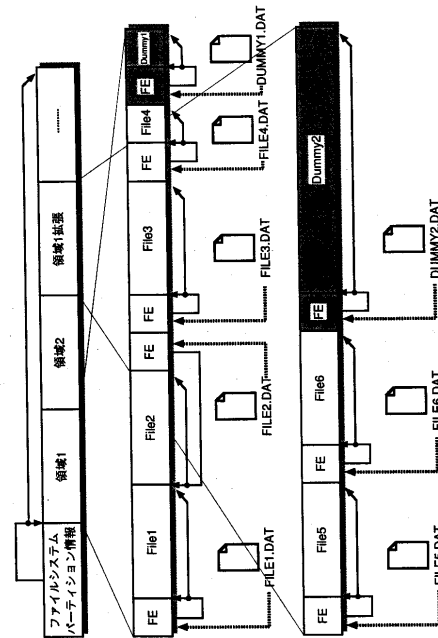
【図 8】



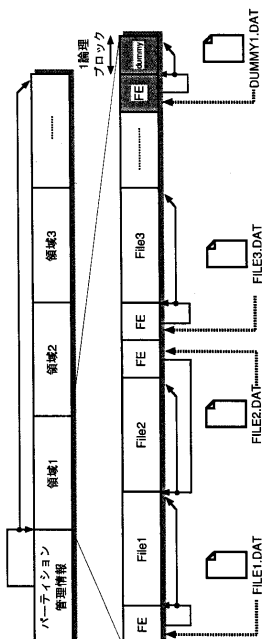
【図 9】



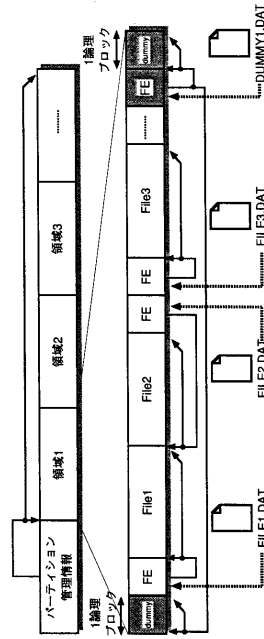
【図 10】



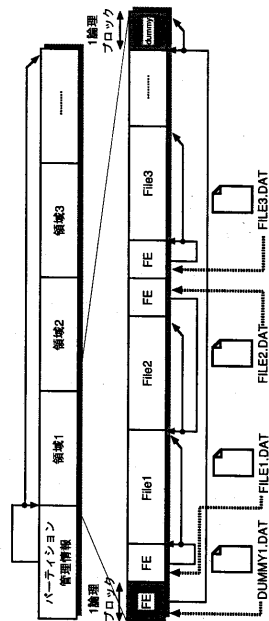
【図 11】



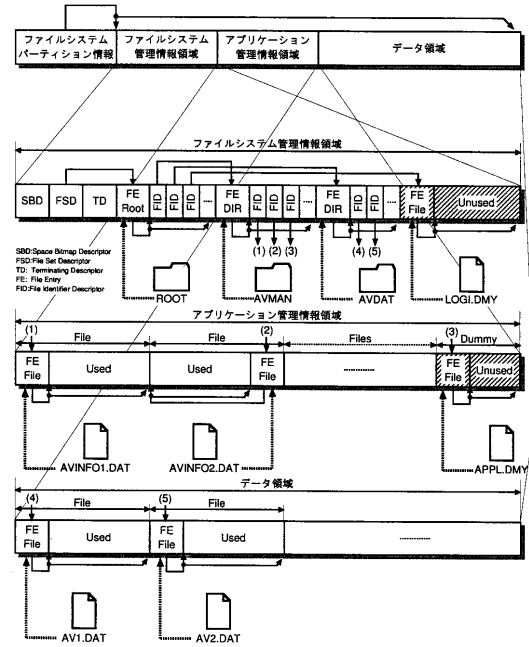
【図 12】



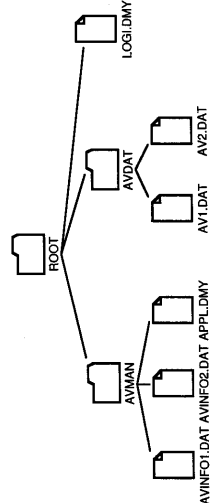
【図 13】



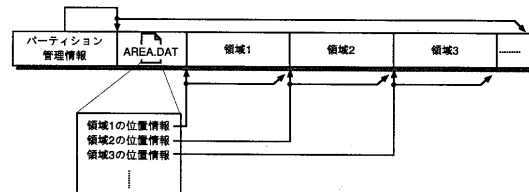
【図 14】



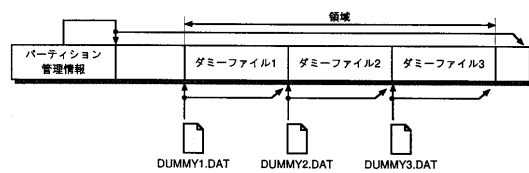
【図 15】



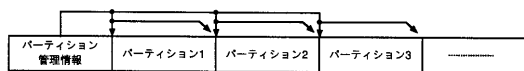
【図 17】



【図 18】



【図 16】



フロントページの続き

- (72)発明者 池田 奈津子
大阪府大阪市阿倍野区長池町 2 2 番 2 2 号 シャープ株式会社内
- (72)発明者 木山 次郎
大阪府大阪市阿倍野区長池町 2 2 番 2 2 号 シャープ株式会社内
- (72)発明者 西村 元秀
大阪府大阪市阿倍野区長池町 2 2 番 2 2 号 シャープ株式会社内
- (72)発明者 山村 博幸
大阪府大阪市阿倍野区長池町 2 2 番 2 2 号 シャープ株式会社内
- (72)発明者 山口 孝好
大阪府大阪市阿倍野区長池町 2 2 番 2 2 号 シャープ株式会社内

審査官 堀江 義隆

- (56)参考文献 特開平 0 8 - 0 9 5 8 3 5 (J P , A)
特開昭 6 3 - 1 6 3 6 4 2 (J P , A)
特開平 0 9 - 1 7 9 7 5 8 (J P , A)
特開平 1 0 - 2 8 3 2 3 1 (J P , A)

- (58)調査した分野(Int.Cl.⁷, D B 名)

G06F 12/00
G06F 3/06 - 3/08
G11B 20/12
G11B 27/00