

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3769019号
(P3769019)

(45) 発行日 平成18年4月19日(2006.4.19)

(24) 登録日 平成18年2月10日(2006.2.10)

(51) Int. Cl. F I
G06F 12/16 (2006.01) G O 6 F 12/16 3 1 O M
G06F 12/00 (2006.01) G O 6 F 12/00 5 3 1 M

請求項の数 4 (全 17 頁)

<p>(21) 出願番号 特願平8-528191 (86) (22) 出願日 平成8年3月18日(1996.3.18) (65) 公表番号 特表平11-502339 (43) 公表日 平成11年2月23日(1999.2.23) (86) 国際出願番号 PCT/GB1996/000651 (87) 国際公開番号 W01996/029650 (87) 国際公開日 平成8年9月26日(1996.9.26) 審査請求日 平成14年1月24日(2002.1.24) (31) 優先権主張番号 9505939.0 (32) 優先日 平成7年3月23日(1995.3.23) (33) 優先権主張国 英国(GB) (31) 優先権主張番号 08/465,925 (32) 優先日 平成7年6月6日(1995.6.6) (33) 優先権主張国 米国(US)</p>	<p>(73) 特許権者 シャイアン アドヴァンスト テクノロジ ー リミテッド イギリス サマーセット ティーエイ21 9ジェイイー ウェーリングトン チェ ルストーン ビジネス パーク サマーフイ ールド ウェイ (番地なし) (74) 代理人 弁理士 中村 稔 (74) 代理人 弁理士 大塚 文昭 (74) 代理人 弁理士 穴戸 嘉一 (74) 代理人 弁理士 竹内 英人</p>
---	--

最終頁に続く

(54) 【発明の名称】 ファイルを開いた状態で作動可能なコンピュータバックアップシステム

(57) 【特許請求の範囲】

【請求項1】

命令メモリと、中央処理装置と、記憶手段とを含むマイクロプロセッサベースコンピュータを操作する方法であって、

(1-1) コールが書き込みコールか否かを判断し、

(1-2) 前記コールが書き込みコールであると判断した場合に、前記書き込みコールが遮蔽されているか否かを判断し、

(1-3) 前記書き込みコールが遮蔽されていないと判断された場合、オリジナルのデータ領域(24)にデータを書き込み、

(1-4) ステップ(1-2)において、前記書き込みコールが遮蔽されていると判断された場合、デルタ・マッピング・テーブル(26)をチェックして、ファイル内の指定されたレコード又は部分に関して規定された、前記オリジナル・ファイル(24)のそのエリアに対して、書き込みが以前に為されたか否かを判断し(28)、

(1-5) 以前に書き込みが為されたと判断された場合、以前に、デルタ・マッピング・テーブル内のオリジナル・ファイル(24)のその部分にマップされたデルタ領域(30)のその部分に、新しいデータ(32)をオーバーライトし、

(1-6) ステップ(1-4)において、以前に書き込みが為されていないと判断された場合、前記テーブルに書き込みを追加するとともに、データ(34)のその断片に、前記デルタ領域の新しい領域を割当て、

(1-7) 前記データを、新しいデータ領域(36)に書き込む、

10

20

ステップを含む方法。

【請求項 2】

請求項 1 の方法によって書き込まれたデータを読み出すための方法であって、

(2 - 1) コールが読み出しコールか否か判断し、

(2 - 2) コールが読み出しコールであると判断した場合に、前記読み出しコールが遮蔽されているか否か判断し (38)、

(2 - 3) 読み出しコールが遮蔽されていないと判断した場合に、オリジナル・データ領域からデータ (40) を読み出し (24)、

(2 - 4) ステップ (2 - 2) において、読み出しコールが遮蔽されていると判断した場合に、コールがバックアップ・アプリケーション・プログラムから来たか否かをチェックし (42)、

(2 - 5) ステップ (2 - 4) において、コールが、バックアップ・アプリケーション・プログラムから来たか判断した場合に、オリジナル・データ領域 (24) からデータを読み出し (40)、

(2 - 6) ステップ (2 - 4) において、コールが、バックアップ・アプリケーションから来ていないと判断した場合に、デルタ・マッピング・テーブルをチェックして、読み出しコールの領域に対応するオリジナル・ファイルの領域 (24) への書き込みが、デルタ領域に転送されたか判断し (44)、

(2 - 7) ステップ (2 - 6) において、読み出しコールの領域に対応するオリジナル・ファイル (24) の領域への書き込みが、デルタ領域に転送されていないと判断した場合には (46)、オリジナル・データ領域 (24) からデータを読み出し (40)、

(2 - 8) ステップ (2 - 6) において、読み出しコールの領域に対応するオリジナル・ファイル (24) の領域への書き込みが、デルタ領域に転送されたと判断した場合には (46)、デルタ領域 (30) の正しい部分からデータを読み出す (48)、

ステップを含む方法。

【請求項 3】

命令メモリと、中央処理装置と、記憶手段とを含むマイクロプロセッサベースコンピュータを操作する方法であって、

(3 - 1) コールが書き込みコールか否かを判断し、

(3 - 2) 前記コールが書き込みコールであると判断した場合に、前記書き込みコールが遮蔽されているか否かを判断し (20)、

(3 - 3) 前記書き込みコールが遮蔽されていないと判断された場合、オリジナルのデータ領域 (24) にデータを書き込み、

(3 - 4) ステップ (3 - 2) において、前記書き込みコールが遮蔽されていると判断された場合、デルタ・マッピング・テーブル (26) をチェックして、ファイル内の指定されたレコード又は部分に関して規定された、前記オリジナル・ファイル (24) のそのエリアに対して、書き込みが以前に為されたか否かを判断し (28)、

(3 - 5) 以前に書き込みが為されたと判断された場合、オリジナル・データ領域 (24) を、新しいデータでオーバーライトし (66)、

(3 - 6) ステップ (3 - 4) において、以前に書き込みが為されていないと判断された場合、オリジナル・データ領域からデータを読み出し (62)、前記テーブルに書き込みを追加するとともに (34)、デルタ領域及びオリジナル・データ領域に、オリジナル・データ領域から読み出された (62) データを書き込む (64)、

ステップを含む方法。

【請求項 4】

請求項 3 の方法によって書き込まれたデータを読み出すための方法であって、

(4 - 1) コールが読み出しコールか否か判断し、

(4 - 2) コールが読み出しコールであると判断した場合に、前記読み出しコールが遮蔽されているか否か判断し (38)、

(4 - 3) 読み出しコールが遮蔽されていないと判断した場合に、オリジナル・データ領域

10

20

30

40

50

からデータ(40)を読み出し(24)、

(4-4)ステップ(4-2)において、読み出しコールが遮蔽されていると判断した場合に、コールがバックアップ・アプリケーション・プログラムから来たか否かをチェックし(60)、

(4-5)ステップ(4-4)において、コールが、バックアップ・アプリケーション・プログラムから来ていないと判断した場合に、オリジナル・データ領域(24)からデータを読み出し(40)、

(4-6)ステップ(4-4)において、コールが、バックアップ・アプリケーションから来たか判断した場合に、デルタ・マッピング・テーブルをチェックして、ファイル内の指定された、前記オリジナル・ファイル(24)のそのエリアに対して、書き込みが以前に

10

為されたか否かを判断し(46)、

(4-7)ステップ(4-6)において、ファイル内の指定された、前記オリジナル・ファイル(24)のそのエリアに対して、書き込みが以前に為されていないと判断した場合には(46)、オリジナル・データ領域(24)からデータを読み出し(40)、

(4-8)ステップ(4-6)において、ファイル内の指定された、前記オリジナル・ファイル(24)のそのエリアに対して、書き込みが以前に為されたか判断した場合には(46)、デルタ領域(30)の正しい部分からデータを読み出す(48)、

ステップを含む方法。

【発明の詳細な説明】

〔背景技術〕

20

この発明はコンピュータシステムを作動させる方法に関し、特に、記憶装置が故障したり、或いは、データが汚染され、失われ、又は、コンピュータウイルスによって感染したりすることに対してガードするために、記憶装置にストアされたデータのバックアップコピーを作成する方法に関する。本発明は特に、パーソナルコンピュータに関してバックアップを備えることに関する。

ハードディスク即ち固定ディスクのような記憶装置を備える従来の簡単な単一ユーザ用コンピュータシステムでは、たった1つのアプリケーションプログラムが何時でも記憶装置にアクセスすることができる。バックアップアプリケーションが終了するまで、他のアプリケーションは作動することができない環境なので、記憶装置によってストアされたデータをバックアップするタスクは、バックアップ操作を実行しながら、記憶装置を排他的に

30

使用するアプリケーションプログラムによって実行される。

いくつかのアプリケーションを同時に操作することができるマルチタスク環境と、ユーザが幾らかのリソースを共有することができるネットワークシステムとを導入することにより、1つ以上のアプリケーションが同じ記憶装置に同時にアクセスする状況が起こる。この状況が生じたとき、アプリケーション間に競合の可能性がある。例えば、あるアプリケーションが、別のアプリケーションによって現在使用されているファイルを変更又は削除することがあり得る。このことは混乱及びもしかすると記憶装置の破損をもたらす、その結果、データを失うことになる。

「ファイルロック」及び「レコードロック」として知られる方法が、ファイルロックの場合では、1つ以上のアプリケーションがファイルに書き込むことを、レコードロックの場合では、2つのアプリケーションが所定のファイルの同じ領域を変更することを回避するために、オペレーティングシステムに導入されてきた。これらの方法は、特に各アプリケーションが共有されているファイルに小さな変更をそれぞれ行うだけであり、その結果、ロックされているファイル又はレコードにアクセスしたい他のアプリケーションを著しく遅延させないならば、ほとんどのアプリケーションに関して十分に高性能のものとなる。しかしながら、記憶装置が共有されているコンピュータシステムにおいて、データを定期的にバックアップするためのアプリケーションは、特別な場合である。

40

ファイルはアプリケーションプログラムと組み合わせられるであろうから、バックアップアプリケーションとオペレーティングシステムのどちらもファイルの内部構造の知識を有することはなく、従って、バックアップをとることができるデータの最小単位は単一のファ

50

イルである。

更に、ファイルのバックアップコピーは、特定の時点でのオリジナルの正確なイメージであることが必要である。しかし、ファイルをコピーすることは瞬間的なプロセスではなく、コピーをするのにかかる時間はファイルのサイズとコピー動作の速度に依存する。このことは、コピープロセス中にファイルを変更することができる他のアプリケーションがないことをバックアップアプリケーションが確信していなければ問題をもちたらず。

ファイルをコピーするために、バックアップアプリケーションは転送バッファとしてランダムアクセスメモリ(RAM)の領域を割り当て、ソースファイルのブロックをバッファに交互に読み込むことによって続けられ、次いで、バッファコンテンツを2次記憶装置に書き出す。ファイルが非常に小さくなければ、転送バッファは幾らかの時間でいっぱいになっ

10

てしまい、コピー動作はファイルのコピーを完全にさせる種々の読み込みオペレーションと書き込みオペレーションを要求する。コピープロセス中、変更されるべきファイルを許すことにより生じる潜在的な問題は、一連の10個のブロックとしてコピーされるファイルを例として考慮することによって説明しうる。ブロック1乃至5が、既にコピーされた後にブロック3及び8に対応するファイルへの変更を所望するトランザクションによってファイルが変更されうるならば、トランザクションが発生する前に既にコピーされてしまっているため、コピープロセスはブロック8への変更を引き続きコピーするが、ブロック3への変更はコピーしない。バックアップコピーを「所有する」アプリケーションがファイルを破損しているものとおそらくみなすので、バックアップコピーは従って、バックアップコピーを全く役に立たなくさせうる部分的なトランザクションだけを含む。

20

本発明者等の米国特許第5,086,502号は、複製された書き込みオペレーションを2次記憶装置に作ることによって行わせる各書き込みオペレーションをバックアップする方法を開示する。この方法に加えて、共有された記憶装置を有するコンピュータシステムでファイルをバックアップする他の2つの方法がある：

(i) バックアップ中にファイルをロックさせる。これは他のユーザが他のユーザの書き込みアクセスを拒絶することによってファイルを変更しないようにする。別のユーザが既に書き込みアクセスをしているならば、バックアップアプリケーションは、そのファイルをスキップしなければならないか、若しくは、排他的な書き込みアクセスが得られるまで待たなければならないかのいずれかである。

30

(ii) ファイルへの変更を検出する。ファイルをコピーした後、バックアップアプリケーションは、コピーオペレーション中に変更がなされることがたとえあるとしても、チェックするためにオペレーティングシステムを使用することができる。変更がなされたならば、次いで、コピーは繰り返されなければならないが、即ちそのファイルをスキップする。

多くのファイル(特にデータベースファイル)が絶えず使用されているので、後者の2つの方法のいずれも十分な解決策を提供しない。更に、2つの方法は、バックアップが行われている間、ユーザが有効にロックアウトされることを要求する。更に、2次記憶装置とオペレーティングシステムの少しの組み合わせは、15MB/分より速い転送速度を維持することができ、従って大きなファイルをコピーするのにかかる時間は重要である。

2、3のアプリケーションプログラムは、共有された環境でそれら自身のファイルを安全にバックアップする方法を提供することによって問題をアドレス指定するが、それぞれのプロシージャは異なる。全ての記憶装置のバックアップをとるために、それゆえバックアップアプリケーションは、各々異なった組み込み手順作業の仕方を知っている必要があり、或いは、適当な時点で手動の介入に頼らなければならない。

40

従って、バックアップアプリケーションが全体のファイルのバックアップをとる必要があり、且つ、これはバックアッププロセス中に別のアプリケーションプログラムによって変更されないようにしなければならないので、先のバックアップ方法の使用は共有された記憶装置を有するコンピュータシステムに遅延をもちたらずであろう。このことは、大きなファイルがバックアップされる間、特に正しい。

Wallsの米国特許第5,163,148号は、ファイルのバックアップが始まる前に、該ファイルは

50

無矛盾の状態にされ、例えば、複数のオペレーションがトランザクションに関連して要求されているならば、次いで、これらのすべてのオペレーションは完了するという、バックアップシステムを説明している。バックアップされているファイルの一部が更新を要求するならば、オリジナルファイルのその部分は更新が行われる前に、ディスクの別の部分にコピーされる。バックアッププログラムは、ディスクのほかの場所にコピーされ且つ保持されているオリジナルの部分によって更新されるそれらの部分を置換することによってファイルをバックアップする。そのファイルによって占有されたディスクのトラック及びセクタのリストを正常に保持するディスクエクステンツマップは、コピーされたオリジナルのファイルの部分のトラック及びセクタロケーションを、それらがコピーされる領域を占有する更新された部分のトラック及びセクタロケーションに関係付ける用語索引を含むように拡張される。

10

国際公開W095/19599として1995年7月20日に公開された本発明者等の国際特許出願PCT/GB94/01581は、バックアッププロセスの実行中、記憶装置のデータに対する変化の記録を保全するためのプロセスを含むバックアッププロセスをも備える、種々のアプリケーションプログラム間で共有された記憶装置を有するコンピュータ操作方法を説明する。その記録は、バックアッププロセスの実行中、バックアッププロセスは変更しないことによってデータが確認されることを保証するのに使用され、一方、他のプロセスが記憶装置を正常に更新し続けることを可能にし、それによってファイルロック技術に関連する遅延を回避する。追加のオペレーションは、コンピュータの通常の使用が変化しないような仕方のマイクロコンピュータのオペレーティングシステムインターフェースのレベルで実行されている。その方法では、バックアッププロセスはスタート信号及びストップ信号を発行するため、及び、オリジナルデータを記憶装置即ちハードディスクから読み出すために少なくとも1つのバックアップ読み込み操作を実行するために操作可能である。中央演算装置(CPU)は、データを記憶装置の1次領域に定期的書き込むために少なくとも1つの通常の書き込み操作、及び、データを記憶装置から読み出すために少なくとも1つの通常の読み出し操作を実行する。

20

前記スタート信号を受信した後、遮断操作が、書き込み操作を遮断するため、及び、書き込み操作によってさもなければ上書きされうるオリジナルデータを保存するために、記憶装置の2次記憶領域にデータを書き込むために実行される。2次領域に書き込まれるファイル部分の記録は保持され、バックアップ及び通常の読み込み操作のうちの1つを実行するとき、記録は記録に従って1次及び2次記憶領域の1つからデータを読み出すために応答司令信号が送られる。このようにして、バックアッププロセスは、スタート信号が発行される前にバックアッププロセスがすぐになされる状況でオリジナルデータを備える。しかしながら、この例では、記録はディスクのトラック及びセクタの観点からではなく、ファイル記録の観点からファイル部分を定義する。このことは、記録がディスクロケーション、例えばトラック及びセクタに関して保持されるならば、生ずるかもしれない問題を回避し、所定の利益を提供する。

30

記録は、好ましくは、1次データがストアされているデルタ領域と、1次データの実際の位置とその意図された位置との間のマッピングリファレンスを有する。変形実施形態では、2次データはデルタ領域に記録されてよく、他のアプリケーションは記憶装置の意図された位置に書き込みを続けることができる。

40

デルタ領域は好ましくは、幾らかの記憶装置、例えばハードディスクの異なる領域である。要求された追加のオペレーションは、コンピュータシステムのオペレーティングシステムインターフェースレベルで有利にストアされ且つ操作される。ソフトウェアがオペレーティングシステムレベルでストアされるならば、それはアプリケーションソフトウェアに影響をおよぼさないようにユーザに対して透過的に作ることができる。

このバックアップレコーディングは、種々の記憶装置、例えばテープドライブ、光ディスク、又は、基本記憶メディアの別の領域にストアすることができる。一旦バックアップが行われたならば、在来の形でリストアすることができる。

〔発明の概要〕

50

本発明は、前述の方法に適用可能な改良を提供する。まず、システムが、実行することができる他のアプリケーションから始まる読み込み要求からのバックアッププログラムから始まる読み込み要求を識別することができるということが必要条件である。

第1の態様では、本発明は、命令メモリと、中央処理装置と、記憶手段とを含むマイクロプロセッサベースコンピュータを操作する方法を提供する。この方法は、開始信号及び停止信号を発生し、記憶手段からオリジナルデータを読み込むために少なくとも1つのバックアップ読み込み命令を発生するように作動可能な、コンピュータシステムで走る、バックアッププロセスを設け、前記メモリにストアされた第1の命令を中央処理装置に与えて、該中央処理装置に、少なくとも1つの通常の読み込み操作を行わせることによって、記憶手段からデータを読み込み、該記憶手段の1次記憶領域及び前記記憶手段の2次記憶領域における対応する領域を関係付けるために、前記記憶手段に書き込まれたデータの記録を維持し、前記開始信号を受け取った後、読み込み命令を受け取ると同時に、前記読み込み命令がバックアップ読み込み操作又は通常の読み込み操作に關係するかどうかを判断し、前記バックアップ及び通常の読み込み操作のうちの一つを行うと同時に、前記記録に従って前記1次及び2次記憶領域のうちの一つからデータを読み込むために前記記録をチェックし、他の前記バックアップ及び通常の読み込み操作を行うと同時に、前記1次記憶領域からデータを読み込む段階、を含み、読み込み命令がバックアップ読み込み操作又は通常の読み込み操作に關係するかどうか判断する段階は、前記読み込み命令が前記読み込み命令を開始するプログラムの識別を判断するように始まる、メモリの領域で調べる段階を含む。

10

20

本発明の第2の別の態様では、コンピュータネットワークにおいて、命令メモリと、中央処理装置と、記憶手段とを含む、マイクロプロセッサベースコンピュータシステムを操作する方法が提供される。この方法は、開始信号及び停止信号を発生し、前記記憶手段からオリジナルデータを読み込むために少なくとも1つのバックアップ読み込み命令を発生するように作動可能な、前記コンピュータシステムで走る、バックアッププロセスを設け、前記メモリにストアされた第1の命令を中央処理装置に与えて、該中央処理装置に、少なくとも1つの通常の読み込み操作を行わせることによって記憶手段からデータを読み込み、前記記憶手段の1次記憶領域及び前記記憶手段の2次記憶領域における対応する領域を関係付けるために、前記記憶手段に書き込まれたデータの記録を維持し、前記開始信号を受け取った後、読み込み命令を受け取ると同時に、前記読み込み命令がバックアップ読み込み操作又は通常の読み込み操作に關係するかどうかを判断し、前記バックアップ及び通常の読み込み操作のうちの一つを行うと同時に、前記記録に従って前記1次及び2次記憶領域のうちの一つからデータを読み込むために前記記録をチェックし、他のバックアップ及び通常の読み込み操作を行うと同時に、前記1次記憶領域からデータを読み込む段階を含むものであって、バックアッププロセスに一意的なユーザ識別子を割り当てること、及び、読み込み命令がバックアップ読み込み操作又は通常の読み込み操作に關係するかどうか判断する段階で、読み込み操作が前記一意的なユーザ識別子を備えるユーザによって要求されるかどうかを判断することからなる改良がなされる。

30

従来のアプリケーションのメソッドの潜在的な問題は、特定のファイルが別のアプリケーションに対して開いていると、バックアッププロセスによる該ファイルへのアクセスが、オペレーティングシステムによって拒否されるので、所望のバックアップ操作をそのファイルに行うことができないということである。

40

本発明の前述の態様と関連して用いられるのが好ましい、本発明の第3の態様では、命令メモリと、中央処理装置と、記憶手段とを含むマイクロプロセッサベースコンピュータシステムを操作する方法が提供される。この方法は、開始信号及び停止信号を発生し、前記記憶手段からオリジナルデータを読み込むために少なくとも1つのバックアップ読み込み命令を発生するように作動可能な該コンピュータシステムで走る、バックアッププロセスを設け、前記メモリにストアされた命令を中央処理装置に与えて、該中央処理装置に少なくとも1つの通常の読み込み操作を行わせて、記憶手段からデータを読み込み、通常の読み込み操作によって読み込まれるべきファイルを開け、他の読み込み操作によって前記フ

50

ファイルへのアクセスを拒絶するファイルオープニング手段を設けること、を含み、前記ファイルオープニング手段の前記拒絶機能が、バックアップ読み込み操作及び通常の読み込み操作によって前記ファイルに同時に読み込みアクセスさせるのを不能にする、前記バックアッププロセスと関連する手段を提供することからなる改良を含む。

最後に、従来のアプリケーションのバックアップ方法は、バックアップが作動中にバックアップされているファイルに変更を加えることに対処することができるが、本発明者は、ファイルが、バックアップ操作が実際に始まる時に、書き込みがなされているプロセスにあるならば、問題があることを認識した。

第4の態様では、本発明は、命令メモリと、中央処理装置と、記憶手段とを含むマイクロプロセッサベースコンピュータシステムを操作する方法を提供する。該方法は、開始信号及び停止信号を発生し、前記記憶手段からオリジナルデータを読み込むために少なくとも1つのバックアップ読み込み操作を行うように作動可能な、前記コンピュータシステムで走る、バックアッププロセスを提供し、前記メモリにストアされた命令を中央処理装置に与えて、該中央処理装置に少なくとも1つの通常の書き込み操作を行わせて、記憶手段にデータを書き込み、バックアッププロセスが、前記命令によってアクセスするために開いているファイルを操作することができるようになっており、バックアップされるべきファイルへの一連の書き込み操作が完了するまで、開始信号の発生を遅延させる段階からなるか

いりようがなされている。

本発明を図面に関する方法の例によって以下に説明する。

【図面の簡単な説明】 20

図1は、パーソナルコンピュータとカートリッジテープドライブのブロック図である。図2A及び図2Bは、それぞれ、在来のコンピュータ及び本発明の方法に従って操作するように変更されたコンピュータのオペレーティングシステムレベルのシステムメモリマップを説明する図である。

図3は、バックアップ操作においてデータを記録するための方法のフローチャートである。

図4は、図3の方法の第2実施形態のフローチャートである。

図5は、図3及び図4に関して説明された実施形態に対する改良を説明するフローチャートである。

〔発明を実施する最善の形態〕 30

図面の図1を参照すると、パーソナルコンピュータ10は、中央処理装置12と、ランダムアクセスメモリ14と、ハードディスクドライブ16の形をとった不揮発性記憶手段とを典型的に備える。この例では、記憶手段のバックアップを取る目的のために、テープユニット18がコンピュータ10に取り付けられている。

バックアッププログラムアプリケーションの部分は、ディスクドライブ16とコンピュータ10の他の部分との間のデータの流れを制御するために、それぞれディスクドライブ16から及びディスクドライブ16に正常に読み込み及び書き込み操作を遮るように作動するレコーディングプロセスとして形成される。

ファイルのバックアップコピーをとり始める前に、レコーディングプロセスは「デルタ」領域として知られるディスク16の別の領域に全てのディスク書き込み活動を転送するためにバックアップアプリケーションによって付勢される。デルタマッピングテーブルは、デルタ領域に転送されているすべてのディスク書き込み操作の記録を提供するために保持される。このテーブルは、各ディスク書き込み操作と関係するデータのディスク16の意図された宛先と、デルタ領域におけるその実際の記憶場所との間のマッピングを保持する。続いて起こる書き込み操作が、既にテーブルにおけるエントリのサブジェクトである領域に生じるならば、デルタ領域の古いデータは新しいデータによって上書きされるのを許す。この方法では、全ての変更がデルタ領域に転送されるので、ディスクの主な部分のデータはバックアップコピーが取られている間、変更しない。ディスク読み込み操作がバックアップアプリケーションによって要求されているとき、それは、バックアップアプリケーションに返されるデータがディスクの主な部分のオリジナルデータ領域から来るように不変に 50

通される。しかしながら、ディスク読み込み操作が別のアプリケーションによって要求されているとき、レコーディングプロセスは、その領域へのディスク書き込みが生じているかどうかを確認するためにテーブルをチェックするようにはじめに作動し、その理由は、そのプロセスは書き込み操作を遮り始めるからであり、もしそうならば、読み込み操作はデルタ領域の対応する位置に転送される。書き込み操作遮蔽が始まっているので、書き込み操作が生じていなければ、即ち、データのピースが変更されていないならば、次いで、読み込み操作は変効無く通過する。

この方法では、ディスク16にストアされたデータへの変更は、バックアップアプリケーションの他の全てのアプリケーションによって正常に見ることができる。バックアップアプリケーションは、読み込み及び書き込み操作を遮断するレコーディングプロセスを停止するために命令が発行されるまで、変更しないデータを見る。

10

バックアップコピー操作が終了したとき、レコーディングプロセスはコンピュータを通常の操作に戻すために付勢される。この復元段階は、デルタ領域にストアされた変更を、テーブルにストアされたマッピングを使用するディスク16の主部分の適当なオリジナルデータ領域に適用することによって達成される。デルタ変更が適用されている間、ディスク書き込み操作は上述したように遮蔽され続ける。オリジナルデータ領域に適用されていない書き込み操作がもはやないとき、書き込み操作の遮蔽は停止し、システムは従来通り作動する。

更なる改良として、バックアップコピープロセスが進行する範囲までレコードが入らないようにされているならば、ディスクのその部分がバックアップされた後に、ソフトウェアによってディスク16の主部分への書き込み操作は変更せずに通過する、というのは、データが変更するならば、もはや問題ではないからである。これはデルタマッピングテーブルのサイズを減少させる。また、その操作が差し向けられるディスクの主部分の領域が既にマッピングテーブルにおけるサブジェクトのエントリであるならば、次いで、そのエントリは簡単に削除することができ、それによって、復元段階での所要時間を減少させる。

20

本発明は、IBM互換パーソナルコンピュータ、即ち、インテルによって製造された8086, 8088, 80286, 80386, 80486及びペンティアムタイプのプロセッサチップ、及び、システムに基づいた同様なマイクロプロセッサをそれらの中央処理装置(CPU)として使用する大多数のパーソナルコンピュータに特に適用することができる。かかるコンピュータでは、プログラムがファイルにアクセスしたいときはいつでも、それはデータをディスクに書き込む標準的なルーチンを呼び出す。DOSオペレーティングシステムの場合では割り込み21 hex (INT 21h)として知られているこのルーチンは、オペレーティングシステムの不可欠な部分である。割り込みハンドラルーチンとして知られているルーチンによって実行される動作は、エントリのルーチンに通されるパラメータに依存する。このルーチンを、システムメモリマップ中のオペレーティングシステムの部分を形成するものとして図2Aに示し、INT 21hエントリポイントは矢印によって示されている。本発明における好ましい方法を実行するために、追加のプログラムコードが図2Bに示したようにオペレーティングシステムインターフェースレベルで加えられる。実際に、DOS環境では、これをCONFIG.SISファイルを使用するデバイスドライバとしてコンピュータ内にロードすることができる。

30

40

追加されたソフトウェアは、別の命令のセットによって更新されているデータを書き込むための命令の効果の有する。同様な技術を、ディスクからデータを読み出すための命令を遮蔽するのに使用することができ、且つ、これをまた別の命令のセットと置換するのに使用することができる。

別の命令のセットによって提供された関連した一連の操作を図3のフローチャートに示す。INT 21hコールの受信で、プログラムは最初に、コールが書き込みコールか否か判断する。もしそうならば、プログラムは、書き込みコールを遮蔽することになっているかどうかを見るためにステップ20でチェックする。もしそうでなければ、データはオリジナルデータ領域24に書き込まれる(ステップ22)。

書き込みコールが遮蔽されているならば、次いで、プログラムは、書き込みがオリジナル

50

ファイル 2 4 のその領域に対して前になされているかどうか見るためにデルタマッピングテーブルをチェックし (ステップ 2 6)、ファイルにおける指定されたレコード及び部分の観点から定義される。

書き込みが以前に行われているならば (ステップ 2 8)、次いで、デルタマッピングテーブルにおいてオリジナルファイル 2 4 のその部分に以前にマップされたデルタ領域 3 0 のかかる部分は新しいデータで上書きされる (ステップ 3 2)。

書き込みコールが遮蔽されているが、書き込みがオリジナルファイル 2 4 のその部分に対して以前に行われていないならば、次いで、書き込みはテーブルに加えられ、デルタ領域の新しい領域はデータの該ピースに割り当てられる (ステップ 3 4)。次いで、データは新しいデルタ領域に書き込まれる (ステップ 3 6)。

10

コールが読み込みコールならば、プログラムは、読み込みコールを遮蔽することになっているかどうかをステップ 3 8 においてチェックする。もしそうでなければ、次いで、データはオリジナルファイル 2 4 から読み込まれる (ステップ 4 0)。

プログラムが読み出しコールを遮蔽するならば、次いで、それはコールがバックアップアプリケーションプログラムから来たかどうか見るためにステップ 4 2 ではじめにチェックする。コールがバックアップアプリケーションから来ているならば、プログラムはステップ 4 0 に通過し、データはオリジナルファイル 2 4 から読み込まれる。

読み出しコールがバックアップアプリケーションから来ていなければ、プログラムは、オリジナルファイル 2 4 の領域への書き込みが読み出しコールのものに対応するかどうか調べるために、デルタマッピングテーブルをチェックし、デルタ領域に転送する (ステップ 4 4)。それが転送されていなければ (ステップ 4 6)、プログラムはステップ 4 0 に戻り、データはオリジナルファイル 2 4 から読み込まれる。読み込みが転送されているならば、データはデルタ領域 3 0 の正しい部分から読み込まれる (ステップ 4 8)。

20

INT 21h コールが書き込みコールでも読み込みコールでもないならば、次いで、それは通常の INT 21h 処理ルーチンに戻される。

本発明の第 2 の実施形態では、読み込み及び書き込み操作が、レコーディングプロセス中に遮蔽されているとき、その代わりにデルタ領域の変更されたデータをストアし、ファイルの特定の領域にはじめて書き込みがなされたとき、オリジナルデータはデルタ領域にストアされ、且つ、オリジナルファイルは変更されるようになる。デルタエリアにストアされるオリジナルファイルのその領域のオリジナルコンテンツを示すエントリがマッピング

30

テーブルになされる。ファイルの同じ領域に続いて起こるどんな書き込みも、次いで、レコーディングプロセスによって無視される。マッピングテーブルが、オリジナルファイルの領域が上書きされていることを示しているならば、バックアップアプリケーションからのリードリクエストは次いで、デルタ領域に転送される。

図 4 では、第 2 の実施形態がフローチャートで示されている。該チャートは同じ参照番号が同じ動作を示すステップを備えた図 3 のものと似ている。

2 つの図 (3 及び 4) の間の相違は以下の通りである：
ステップ 6 0 では、読み込み遮蔽がアクティブ (ステップ 3 8) であり且つ読み込みがバックアップアプリケーションからのものでないならば、データはオリジナルデータ領域から読み込まれる。これは図 3 におけるステップ 4 2 の逆である。

40

最初に上書きされる前にオリジナルデータ領域からデータを読み込むエキストラステップ 6 2 はステップ 2 8 の後に挿入されている。

図 3 のステップ 3 6 は、ステップ 6 2 でのデータ読み込みのデルタ領域に追加の書き込みを取り入れる新しいステップ 6 4 を形成するために修正されている。

ステップ 3 2 は、もはやデルタ領域を上書きするためにデータを転送しないが、その代わりにオリジナルデータ領域にデータを書き込む新しいステップ 6 6 を形成するために修正されている。

各ファイル書き込みリクエストが第 1 の方法の単一の転送書き込みではなく、読み込みと 2 つの書き込みからなるので、この第 2 の方法は、第 1 の方法よりも遅い。しかしながら、第 2 の方法は、いったんバックアップが完了したならば、オリジナルファイルに適用さ

50

れうるデルタ領域にストアされた変化を要求しない。デルタマッピングテーブル、及び、デルタ領域にストアされたデータは簡単に放棄されうる。更にこの違いのため、デルタ変更がオリジナルデータに対して十分に適用されていないとき、準備は（電源故障のような）致命的な状況から回復させるために必要でない。

上述した方法がバックアップアプリケーションの一部分を形成する場合には、読み込み操作の遮蔽を、バックアップアプリケーション内で起こすことができ、オペレーティングシステムルーチンの使用を含む必要がない。

図3及び図4に関して説明されたようにシステムの詳細な実施は当業者の能力の範囲内にあり、それについての十分な説明は必要ない。MS.DOSで実施するならば、読者は必要ならば、背景材料に関してIBM DOSテクニカルリファレンスマニュアルを参照されたい。

本発明は、オリジナルオペレーティングシステムコードの不可欠な部分であることによって、又は、デバイスドライバの形でオリジナルオペレーティングシステムコードまでの拡張部分を形成することによって、オペレーティングシステム自身の一部として実施されるのが好ましい。インテル80386（及びそれ以上の）CPUの場合では、本発明は、「Ring 0」として知られる最も高い特権レベルで作動し、システムの全てのエレメントへのアクセスが無制限である。

しかし、本発明は操作するためにいかなる特定のCPUアーキテクチャも当てにしない。それは、例えば68000及びPowerPCシリーズCPUを使用するMacintoshコンピュータのような他のシステムでも容易に実施することができる。

本発明はマルチユーザ環境で使用することができる。この場合では、本発明がファイルアクセスコントロールを備えた環境で使用されるのが好ましい。本発明がまるで現れていないかのように本発明がなされるとき、システムのリマインダが効果的に完全に作動する間、本発明は、正しいバックアップコピーがなされるのを保証する。ファイルアクセスコントロールがないときには、本発明が使用されていようと使用されていないとにかかわらず、ファイル破損の危険がある。

本発明は、例えば、大きく且つレコードに分割されることができるdBaseのようなデータベースによって作られたファイルに特に適用することができる。更新されたレコードはテーブルに記録され、レコードの更新されたバージョン及びオリジナルのバージョンの双方は、バックアップが完了するまで維持される。テーブルは、例えば、更新されたデータがスタートするファイルにスターティングポイントと、更新されたデータの量に対応する長さ値とを含む。別の実施形態では、テーブルは、更新されたレコードの識別性の観点から識別子（identification）を含む。

マルチユーザ又はネットワーク状況では、コンピュータ環境が、バックアップを実行するのを望む特定のユーザの識別を可能にするとき、メソッドはまた、ユーザを認識し、上述の仕方でのそのユーザのファイルをバックアップするだけのステップを含みうる。この方法では、全てのデータトランザクションを遮蔽する必要はなく、そのユーザに関するデータトランザクションを遮蔽するだけでよい。

リクエストソース識別

開いているファイルで操作することができる上述したバックアップ方法は、バックアップアプリケーションからのリードリクエスト発信と他のソース即ち通常のアプリケーションからのそれらの発信との間を識別する能力を当てにしている。これは図3のステップ42又は図4のステップ60に現れている。

オープンファイル方法がバックアップアプリケーションの不可欠な部分として提供されているならば、バックアップアプリケーションはそれが発行されるのを要求することが分かるので、リードリクエストのソースは容易に判断される。しかし例えば、追加のコードによって代わりに実施される、標準の規格品パッケージの形態で、又は、さもなければオープンファイルメソッドを収容するために改めることができない形態で、バックアップアプリケーションが既に存在しているならば問題は生ずる。

第1の改良によれば、バックアップアプリケーションからのリクエストは、リードリクエストを開始するコードの領域を識別するためにイベントの連鎖をトレースする（即ち、後

10

20

30

40

50

方又は「上方」を見る) ことによって他のリクエストから識別される。かかる方法の以下に示す例では、MS.DOSオペレーティングシステムは具体例の方法によって再び使用されるが、しかし、同様な技術を他のオペレーティングシステム環境に適用することができる。リードリクエストはINT 21hコールによって開始され、割り込みハンドラルーチンに通常は直接行く。割り込みハンドラルーチンは一般的には、コーラの知識を有していない。それは特定の関数を実行するために単に教えられ、次いでコーラに戻される。DOS環境で使用される上述したインテルのCPUでは、CPUはコーラのリターンアドレスを「スタック」に自動的に設置し、割り込みルーチンの第1の命令にジャンプする。関数が実行されているとき、割り込みハンドラはIRET(割り込みリターン)命令を発行し、CPUによってスタックからのリターンアドレスを検索させ、リターンアドレスで実行しつづける。従って、割り込みハンドラはコーラの知識を必要としない。

10

図3及び図4に関して上述したシステムでは、INT 21h割り込みルーチンは別の命令のセットによって取り替えられる。前述した関数に加えて、これらの命令はリクエストのソースを追加的に求めるようにされており、次に述べる通りである。

この改良では、割り込みハンドラはリターンアドレスを決定するためにスタックを調べる。インテルの80X86シリーズCPUの場合では、コードオフセット及びコーラのセグメントは、スタックのトップに、即ち、SS:[SP]及びSS:[SP+2]にそれぞれ第1及び第2のワードとしてストアされる。この改良のMS.DOSの実施に影響を与えるステップの概略は、この明細書に付録として与えられる。

次いで、ルーチンは、スタックによって定義されたものとしてコールのソースを調べることができ、たくさんの可能な方法においてコールをさせるプログラムを決定し、以下に例を示す：

20

(i) 規格品のプログラムの各バージョンにおけるコードが同じなので、コーラが知られたバックアッププログラムかどうかを判断するために、リターンアドレスのまわりのコードの種々の特徴を調べることができる。リターンアドレスのまわりのコードは、コールがバックアッププログラムによってなされることを示す一致があるか否かを判断するために使用されているバックアッププログラムを期待するコードと比較される。

(ii) 変形実施形態では、各々ロードされたプログラムと関係するプログラムセグメントプレフィックス(PSP)及び環境ストリングを参照することによってプログラム名を設定することができる。スタックから、セグメントは知られ、PSPを読むことができる。このことにより、環境ストリングは、従ってプログラム名を特定することが可能になる。この技術はAndrew Schulman等によって執筆されたテキスト「Undocumented DOS」(1993年、第2版、Addison-Wesley、ISBN0-201-63287-X)に十分に説明されており、第7章特に356乃至359頁を参照されたい。

30

MS.DOS以外のオペレーティングシステムの場合では、コーラの識別は、接続番号、タスク識別番号、プロセス番号及び同様なものを含むスタックの他のパラメータから判断されうる。所定のオペレーティングシステムでは、広域変数はまた、プロセスの名前を設定するのに使用されうる。

この方法は、プログラム名、プロセス(又はスレッド)名のようなコーラのある知られた特徴を識別することに依存するか、若しくは、バイトの知られたシーケンスであってよいプログラムコードにおける何らかの「署名(signature)」を識別すること、又は、スタックを介してコーラプログラムをルックバックすることによって行う。

40

ネットワークサーバ環境において使用することができる第2の改良によれば、コーラプログラムの識別はバックアッププロセスに関する特定のユーザ名だけを予約することによって判断される。プログラムの識別を設定する代わりに、ユーザの名前(一意的なユーザ識別子)が識別され、予約されたユーザからの全てのリクエストをバックアッププログラムで作られたと仮定する。ファイルへのアクセスが認可される前に、全てのユーザがサーバにログインするために要求されるので、もちろんユーザ名が常にバックアップするために使用され、他のいかなる目的にも使用されないならば、これは有効な方法である。この方法は特定のバックアッププログラムの事前の知識を要求しない。

50

同様な方法で、バックアッププロセスに特有のプロセス又はタスク識別は、それがバックアッププロセスか読み込みをリクエストする別のアプリケーションかどうか判断するために応答司令信号を送りうる。同じ識別子(id)を備えた2つのタスクは同時には決して存在しないので、それらはどんな瞬間でもプロセスを一意的に識別する(しかし、タスク番号は再利用される)。

前述のことはリードリクエストに適應する際に説明されるが、書き込みリクエストに対する同じ手順にも適用することができる。バックアッププログラムが書き込みリクエストを発行して次いで適当な動作を起こすことができるイベントでは、バックアップアプリケーションが、書き込みリクエストを割り込みハンドラに通常は発行するはずがないので、それは幾らかのエラールーチンを通常は含む。

10

上述したように、追加の命令は、割り込みコールが読み込み(又は書き込み)リクエストであるかどうか判断するためにはじめに探し、もしそうならば、リクエストがバックアッププログラムによって或いは別のアプリケーションによって発行されるかどうか判断するために探す。原則として、これらの2つの操作の順序は逆にされてもよいが、コールの性質が調べられる前に、コールのソースが最初に求められる。

保証されたファイルアクセス

図3及び図4に関して説明されるオープンファイルバックアップ方法は、ファイルへの読み込みアクセスがバックアップアプリケーションによって常に得られることができると仮定する。特に、ファイルが別のアプリケーションによって排他的な読み込み/書き込みの使用のために常に開いているので、これは常に該場合ではなく、従ってアクセスはバックアップアプリケーションに拒絶されうる。

20

本発明者は、バックアップアプリケーションがファイルに決して書き込まないので、それがバックアップし、上述したオープンファイル方法は、それがバックアップしている間バックアップアプリケーションによって受け取られたファイルのコピーが変更しないことを保証するので、ファイルへの読み込みアクセスを有することはバックアップアプリケーションにとって実際に全く安全である。

バックアップ目的のためのアクセスは、以下のように提供することができる。オペレーティングシステムは一般的には、ファイルハンドル識別子としてかかる情報を含む全ての開いたファイルのテーブルと、カレントファイルオフセットポインタと、各開いたファイルのために維持されたアクセス権とを維持する。

30

MS.DOSオペレーティングシステムの場合では、ファイルはINT 21h関数を使用して開かれる。従って、ファイルオープンリクエストを、上述したような命令の同様な別のセットへの追加によって識別することができる。前のセクションで説明したリクエストソース識別技術を使用することによって、又は、バックアッププログラムからの直接的な命令によって、ファイルオープンリクエストがバックアップアプリケーションによってなされることを設定することができ、もしそうならば、たとえ通常は拒絶されるべきであるとしても、開いたファイルのテーブルを、ファイルアクセスが認可されることを保証するために一時的に操作することができる。いったんアクセスが認可されたならば、割り込みハンドラが戻る前に、テーブルはそれの前の状態にリストアされるが、この場合、他のアプリケーションからの一連のリクエストが通常のようにアクセスを拒絶するであろう。次いで、バックアップアプリケーションからの読み込みリクエストは満たされ、たとえオリジナルアプリケーションが信じたとしても、それはファイルの排他的な使用を有する。

40

MS.DOSでは、開いたファイルのテーブルはシステムファイルテーブル(SFT)として知られている。SFTは各開いたファイルに関するエントリを収容するテーブルである。各エントリはファイルアクセスを認可するアクセスモードを収容する。バックアップアプリケーションが開いたファイルへのアクセスを要求するとき、バックアッププログラムによってアクセスを通常は阻止されうるアクセス権が既に認可されていない影響を与えるために、適当なSFTエントリを一時的に変更することによって、必要とされる追加の命令はSFTを変更する。次いで、アクセスは許され、「ファイルハンドル」が割り当てられる。コーラに戻る前に、SFTエントリのアクセス権はそれらのオリジナルの状態に戻る。

50

システムファイルテーブル (SFT) は上で参照したテキスト「Undocumented DOS」に十分に説明されており、第8章465乃至469ページ及び490乃至494ページを参照されたい。特に、DOSオペレーティングシステムの場合では、バックアップアプリケーションへのアクセスを保証するためにINT 21h割り込みハンドラに要求されるステップを図5に示す。

図5を参照すると、INT 21hコールのようなファイルオープンコマンドを受信したとき、ステップ70では、前のセクションで説明した技術のうちの1つを使用して、リクエストのソースに関して判断を行う。もしそうならば、次いでステップ72で、アクセスが要求されているファイルが既に開いているかどうかに関して判断がなされる。これらのいずれの質問に対する答えが「いいえ」ならば、次いで、プログラムはステップ74の通常のINT 21hハンドラルーチンに戻る。両方の質問に対する答えが「はい」ならば、次いで、図5に記述された特別なコードを実行する。

このプロセスは以下の通りである。SFTにおける各エントリは、プロセスをリクエストするのを許可したアクセスモードを含む(「アクセスモード属性」(Access Mode Attribute))。ステップ76では、SFTにおける適当なエントリの現在又はオリジナルアクセスモード属性をセーブする。ステップ78では、次いで、アクセスモード属性が「Deny None」状態に変更するので、効果的には、SFTは開いたファイルへの複数のアクセスを停止させないような無効の状態にする。今、通常のINT 21h割り込みハンドラを、通常の方法でバックアッププログラムに関するファイルを開くために、ステップ80で呼び出すことができる。バックアッププログラムがファイルで終了し、予めストアされたオリジナルアクセスモード属性がバックをSFTに入れる(ステップ82)。最終的に、実行はIRET命令を使用して呼び出しプログラムに戻される(ステップ84)。

付録に与えられたステップは、保証されたファイルアクセスを提供するこの環境が使用されていることを仮定する。

SFTを変更する正確な方法は重要ではなく、種々の方法を使用することができうる。好ましい方法は、共有ファイル「share_prev_sft」及び「share_net_machine」を操作する(Undocumented DOS、478頁参照)。これらのファイルを操作することによって、例えば、「deny read」値を「deny none」に変更することによって、アクセスはバックアッププログラムに対して許可されるようにされうる。変形実施形態として、MS.DOSの一部である、一部のシェアユーティリティは変更されうるので、アクセスは許可される(op. cit.、490乃至494ページ参照)。

同様の方法が、例えばノベルのNetWareのような、他のオペレーティングシステム環境において使用されうる。

「良好な(Good)」状況の確立

図3及び図4に関して上述したオープンファイルバックアップ方法は、バックアップのために開かれたファイルが、バックアップが進行中に変更されないことを保証する。しかし、バックアップが始まるとすぐに、ファイルが「良好な(Good)」状況になることを確立しない。例えば、別のアプリケーションが、一連の書込み操作を包含する命令を書くことができ、幾らかのものは実行され、幾らかのものは実行されない。バックアップメソッドが開いたファイルで作動するので、それゆえ、バックアップアプリケーションは、部分的な命令を含むファイルのコピーを受け取ることが可能である。

発明者は、トランザクションが進行していないことを判断する時間まで、バックアップアプリケーションからのファイルオープンリクエストを遅延させることによって、ファイルの「良好な状況」のバックアップコピーが保証されうることを理解していた。ほとんどのネットワークオペレーティングシステムは、同時に1つ以上のプロセスを実行するように設計されているマルチスレッドであるまさしくそれらの特性によっており、あるプロセスを完了させながら別のあるプロセスを遅延させる能力はかかるオペレーティングシステムの共通の機能であるので、これは容易に達成することができる。特に、ファイルが開いている間にバックアップすることができるならば、ユーザは通常バックアップオペレーションに左右されないので、バックアップオペレーションが短い間遅らせられるならば、一般的に問題はない。ファイルオープンリクエストにおける遅延は、図3及び図4に関して説

10

20

30

40

50

明したルーチンの実行をはじめると開始信号を遅延させることに相当する。

交流保全性を維持するための必須の機能が使用可能でない場合、第2の方法は適用可能である。次いで、以下のように、交流保全性を確立する別の方法を使用することができる。典型的には、トランザクションは、トランザクションの間に発生するより長い時間のギャップを伴って、比較的短い時間に亘って発生する。アプリケーションプログラムは、一般的には、高速で連続な単一のトランザクションを要求する全ての書き込みオペレーションを実行する、というのは、さもなければ、電源の故障又は同様の重大な問題のような危険により、システムが再起動するとき、ディスクデータは、破損ファイルとして明白である部分的なトランザクションを収容するからである。明らかに、トランザクションがディスクに書き込まれている期間が短ければ短いほど、(電源故障のような)無関係の致命的な問題がトランザクションが書き込まれている間に起こりにくい。継続的な書き込みオペレーションの間の時間を測定することによって、ファイルの交流状況に関して値判断を行うことができる。例えば、言ってみればぎりぎりの時間で起こる書き込みがなければ、アプリケーションが単一のトランザクションの書き込みを途中まで通すことは非常にありそうにない。不活性な期間をアプリケーションの特徴に経験的に適用することによって、交流保全性の非常に信頼性の高い標識を確立することができる。

従って、図3及び図4に関して例示した方法について、種々の説明した改良、すなわち、リクエストソース識別に関する種々の技術と、保証されたファイルアクセスを保証するための技術と、バックアップされるべきファイルがバックアップが行われる前に「良好」な状態にされることを保証するための技術と、がある。これらの種々の改良は、好ましくは組み合わせて使用される。しかし、それらは選択的及び独立して使用することができ、それらは同じアプリケーションに同時に全て実施されるべきことを必要としない。

本発明をDOSオペレーティングシステム、特にMS.DOSに関して主に説明した。しかし、本発明は作動するためにどんな特別なCPUアーキテクチャにもよらない。特に、DOSオペレーティングシステムに限定されず、ノベルNetWare、OS/2又はUNIX、及び、68000及びPowerPCシリーズのCPUを使用するマッキントッシュコンピュータでも実施することができる。

10

20

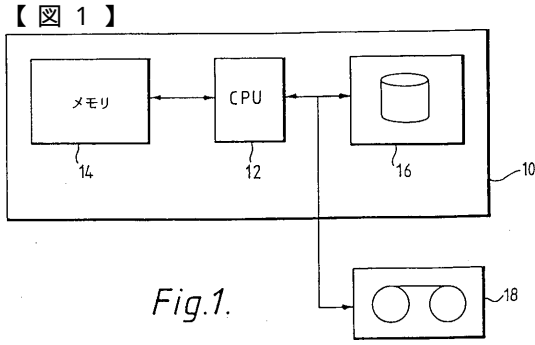


Fig.1.

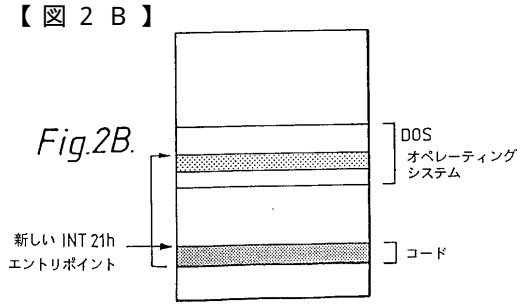


Fig.2B.

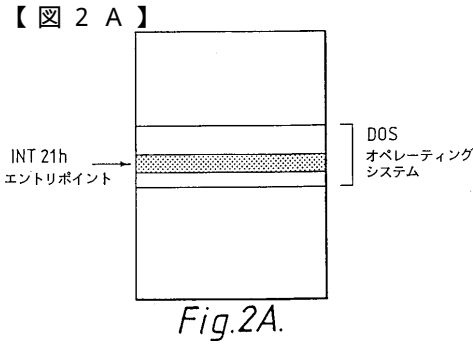


Fig.2A.

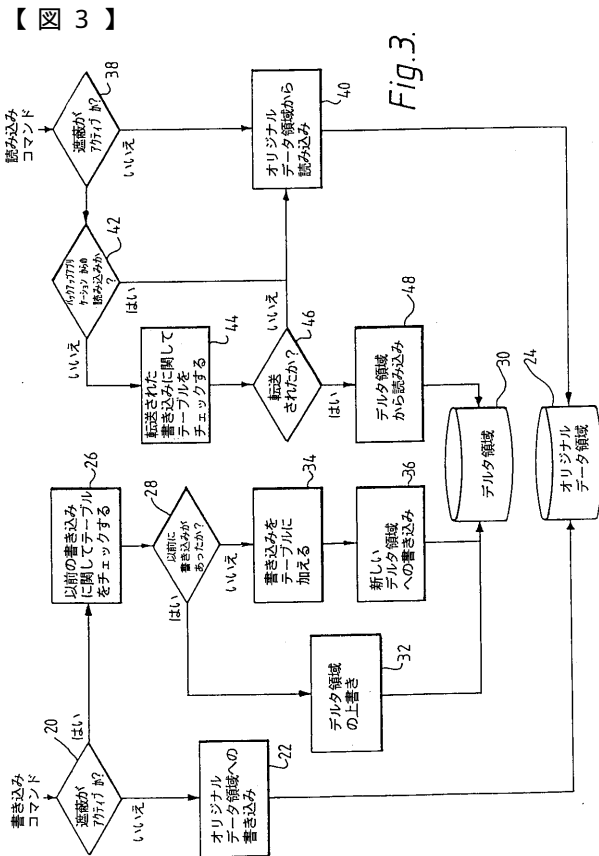


Fig.3.

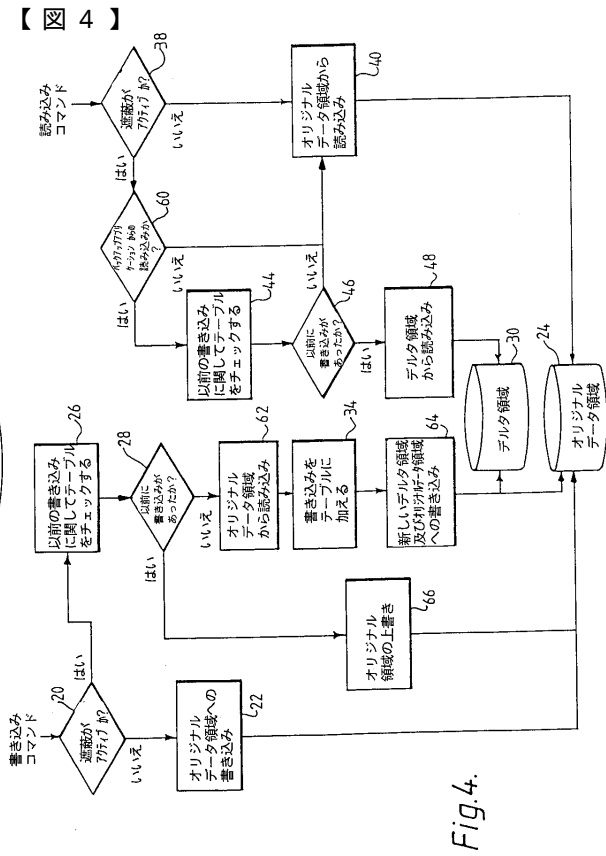


Fig.4.

【 図 5 】

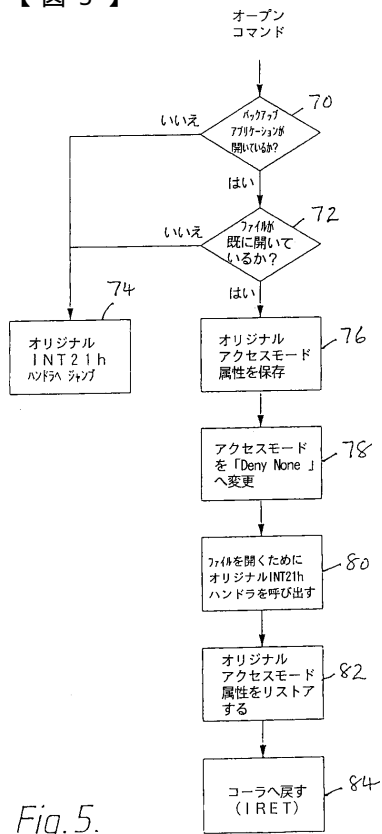


Fig. 5.

フロントページの続き

(74)代理人

弁理士 今城 俊夫

(74)代理人

弁理士 小川 信夫

(74)代理人

弁理士 村社 厚夫

(72)発明者 マルカム ピーター ブライアン

イギリス デーヴォン イーエックス20 4ピーユー リュードン ハイアー ロング パーク
(番地なし)

審査官 丹治 彰

(56)参考文献 特開平07-013829(JP,A)

特開平04-065733(JP,A)

特表平04-507015(JP,A)

特開平06-044010(JP,A)

特開平06-168171(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/16 310

G06F 12/00 531