

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4663878号
(P4663878)

(45) 発行日 平成23年4月6日 (2011.4.6)

(24) 登録日 平成23年1月14日 (2011.1.14)

(51) Int. Cl.	F I
GO6F 9/44 (2006.01)	GO6F 9/06 62OK
GO6F 12/00 (2006.01)	GO6F 12/00 52OP
GO6F 17/30 (2006.01)	GO6F 17/30 24OA
	GO6F 17/30 419B

請求項の数 2 (全 11 頁)

(21) 出願番号	特願2000-559511 (P2000-559511)	(73) 特許権者	591003943
(86) (22) 出願日	平成11年6月28日 (1999.6.28)		インテル・コーポレーション
(65) 公表番号	特表2002-520714 (P2002-520714A)		アメリカ合衆国 95052 カリフォル
(43) 公表日	平成14年7月9日 (2002.7.9)		ニア州・サンタクララ・ミッション カレ
(86) 国際出願番号	PCT/US1999/014722		ッジ ブレーバード・2200
(87) 国際公開番号	W02000/003334	(74) 代理人	100064621
(87) 国際公開日	平成12年1月20日 (2000.1.20)		弁理士 山川 政樹
審査請求日	平成18年6月20日 (2006.6.20)	(74) 代理人	100098394
(31) 優先権主張番号	09/113,810		弁理士 山川 茂樹
(32) 優先日	平成10年7月9日 (1998.7.9)	(74) 代理人	100067138
(33) 優先権主張国	米国 (US)		弁理士 黒川 弘朗
前置審査		(74) 代理人	100081743
			弁理士 西山 修

最終頁に続く

(54) 【発明の名称】 記憶媒体内のファイルを管理する方法および装置

(57) 【特許請求の範囲】

【請求項 1】

(i) 記憶媒体上の新しい場所へ移動した第2ファイルへのレファレンスを有する第1ファイルを求めて記憶媒体をサーチすることであって、前記第1ファイルのレファレンスは前記第2ファイルの移動前の場所を示すパスであり、さらに、前記第1、第2ファイルはともに実行できるコードとデータの内の1つを含んでいる、サーチすること、および

(ii) 前記第1ファイルがアプリケーション特定ファイルであるか否か識別すること

を行うサーチ・ユニットと、

前記サーチユニットに接続し、前記第1ファイルがアプリケーション特定ファイルであつた場合に、第2ファイルを移動前の場所に戻すことを行うフラグging・ユニットと、

前記サーチ・ユニットに接続し、前記第1ファイルがアプリケーション特定ファイルでない場合に、前記第1ファイルに含まれた前記第2ファイルへのレファレンスを更新して更新後のレファレンスが前記移動後の第2ファイルが存在する記憶媒体上の新しい場所を示すパスであるように更新する更新ユニットと

を含むことを特徴とする記憶媒体管理装置。

【請求項 2】

バスと、

このバスに接続したプロセッサと、

このバスに接続した記憶媒体と

10

20

を含むコンピュータ・システムであって、前記コンピュータ・システムは、

記憶媒体上の新しい場所へ移動した第2ファイルへのレファレンスを有する第1ファイルを求めて記憶媒体をサーチすることであって、前記第1ファイルのレファレンスは前記第2ファイルの移動前の場所を示すパスであり、さらに、前記第1、第2ファイルはともに実行できるコードとデータの内の1つを含んでいる、サーチすること、

前記第1ファイルがアプリケーション特定ファイルであるか否か識別すること、

前記第1ファイルがアプリケーション特定ファイルであった場合は、第2ファイルを移動前の場所に戻すことを行い、

前記第1ファイルがアプリケーション特定ファイルでない場合は、前記第1ファイルに含まれた前記特定の第2ファイルへのレファレンスを更新して更新後のレファレンスが前記移動後の第2ファイルが存在する記憶媒体上の新しい場所を示すパスであるように更新すること

とを行うことを特徴とするコンピュータ・システム。

【発明の詳細な説明】

【0001】

(分野)

本発明は、コンピュータ・システムの記憶媒体内のファイル管理に関する。より詳細には、本発明は、記憶媒体内のファイルの依存性を分析して、リンクされたファイルの移動が依存性を有するファイルの機能を削減しないように依存性リンクを更新する方法および装置に関する。

【0002】

(背景)

ファイル依存性は、第1のファイルが第2のファイルからのデータを要求するときに存在する。第1のファイルは、第2のファイルを検索するレジストリ・ファイルを介して第2のファイルにリンクすることができる。別法として、第1のファイルは、第1のファイル内の第2のファイルへのリファレンスを介して、第2のファイルに直接的にリンクしてもよい。ファイルが記憶媒体の1つの場所から別の場所に移動される状況においては、依存ファイル間のリンクは、リファレンスが古くなったときに壊れる場合がある。第1のファイルが第2のファイルを見付けることができないと、第1のファイルの機能性は削減される場合がある。

【0003】

ファイルを移動させながら、依存ファイルのリンクを壊すことを防止するために過去に用いられた1つの手法は、ファイルをそれらの元の場所からアンインストールして、それらの新しい場所で再インストールすることを含んでいた。ファイルを再インストールすることにより、移動されたファイルと移動されたファイルに依存するファイルとの間のリンクは、それらの現在の場所を反映するように更新された。移動されたファイルへのリンクは、移動されたファイルの現在の場所を示す新しいリファレンスをレジストリに書き込むかまたは依存ファイルに直接的に書き込むことにより更新された。

【0004】

この手法の1つの欠点は、ファイルをアンインストールして再インストールするプロセスが、かなりの時間を必要とすることであった。また、それに続いてそれらのファイルによって作成され、それらのファイルと共に保存された重要なデータ・ファイルが、アンインストールのプロセス中に誤って削除されてしまう場合がある。

【0005】

(概要)

記憶媒体を管理する方法を開示する。記憶媒体は、記憶媒体上で新しい場所に移動された第2のファイルへのリファレンスを有する第1のファイルを求めてサーチされる。リファレンスは、第2のファイルが新しい場所に移動したことをそれが示すように更新される。

【0006】

本発明は、添付の図面の図において、例示によって示すものであって、限定によって示す

10

20

30

40

50

ものではなく、図面においては、同じ符号は同じ要素を示す。

【 0 0 0 7 】

(詳細な説明)

図 1 を参照すると、本発明の実施形態を実施できるコンピュータ・システムが 1 0 0 として示されている。コンピュータ・システム 1 0 0 は、データ信号を処理するプロセッサ 1 0 1 を含む。プロセッサ 1 0 1 は、複雑命令セット・コンピュータ (C I S C) マイクロプロセッサ、縮小命令セット・コンピューティング (R I S C) マイクロプロセッサ、非常に長い命令語 (V L I W) マイクロプロセッサ、命令のセットの組合せを実施するプロセッサ、または他のプロセッサ装置とすることができる。図 1 は、単一プロセッサのコンピュータ・システム 1 0 0 で実施される本発明の例を示している。しかし、本発明は多数のプロセッサを有するコンピュータ・システムにおいて実施できることが理解されるであろう。プロセッサ 1 0 1 は、コンピュータ・システム 1 0 0 において、プロセッサ 1 0 1 と他の構成要素との間でデータ信号を送信する C P U バス 1 1 0 に接続されている。

10

【 0 0 0 8 】

コンピュータ・システム 1 0 0 はメモリ 1 1 3 を含む。メモリ 1 1 3 は、ダイナミック・ランダム・アクセス・メモリ (D R A M) デバイス、スタティック・ランダム・アクセス・メモリ (S R A M) デバイス、または他のメモリ・デバイスとすることができる。メモリ 1 1 3 は、プロセッサ 1 0 1 によって実行でき、データ信号によって表される命令および符号を記憶することができる。プロセッサ 1 0 1 の内部には、メモリ 1 1 3 に記憶されたデータ信号を記憶するキャッシュ・メモリ 1 0 2 が常駐する。キャッシュ 1 0 2 は、そのアクセスの局所性を利用することにより、プロセッサ 1 0 1 によるアクセスの速度を上げる。コンピュータ・システム 1 0 0 の代替実施形態においては、キャッシュ 1 0 2 はプロセッサ 1 0 1 の外部に常駐している。

20

【 0 0 0 9 】

C P U バス 1 1 0 とメモリ 1 1 3 とには、ブリッジ・メモリ・コントローラ 1 1 1 が接続されている。ブリッジ・メモリ・コントローラ 1 1 1 は、プロセッサ 1 0 1 と、メモリ 1 1 3 と、コンピュータ・システム 1 0 0 の他の構成要素との間においてデータ信号を方向付けて、C P U バス 1 1 0 と、メモリ 1 1 3 と、第 1 の I / O バス 1 2 0 との間においてデータ信号をブリッジする。

【 0 0 1 0 】

第 1 の I / O バス 1 2 0 は、単一のバスまたは多数のバスの組合せとすることができる。例として、第 1 の I / O バス 1 2 0 は、周辺構成要素相互接続 (P C I) バス、パーソナル・コンピュータ・メモリ・カード・インターナショナル・アソシエーション (P C M C I A) バス、N u B u s、または他のバスを含むことができる。第 1 の I / O バス 1 2 0 は、コンピュータ・システム 1 0 0 において構成要素間の通信リンクを構成している。第 1 の I / O バス 1 2 0 にはネットワーク・コントローラ 1 2 1 が接続されている。ネットワーク・コントローラ 1 2 1 は、コンピュータ・システム 1 0 0 をコンピュータのネットワーク (図 1 には図示せず) にリンクし、これらのマシン間の通信を支援する。第 1 の I / O バス 1 2 0 には表示装置コントローラ 1 2 2 が接続されている。表示装置コントローラ 1 2 2 は、表示装置をコンピュータ・システム 1 0 0 へ接続させ、表示装置とコンピュータ・システム 1 0 0 との間のインターフェイスとして機能する。表示装置コントローラは、単色表示アダプタ (M D A) カード、カラー・グラフィクス・アダプタ (C G A) カード、エンハンスド・グラフィクス・アダプタ (E G A) カード、拡張グラフィクス・アレイ (X G A) カード、または他の表示装置コントローラとすることができる。表示装置は、テレビ受像器、コンピュータ・モニタ、フラット・パネル・ディスプレイ、または他の表示装置とすることができる。表示装置は、表示装置コントローラ 1 2 2 を介してプロセッサ 1 0 1 からデータ信号を受け取り、その情報およびデータ信号を、コンピュータ・システム 1 0 0 のユーザに表示する。第 1 の I / O バス 1 2 0 にはビデオ・カメラ 1 2 3 が光学的に接続されている。ビデオ・カメラ 1 2 3 は、物体の画像を取り込むように動作する。ビデオ・カメラ 1 2 3 は、取り込んだ画像をデジタル・グラフィカル・データに変

30

40

50

換する、内部デジタル・ビデオ・キャプチャ・ハードウェアを有するデジタル・ビデオ・カメラとすることができる。別法として、ビデオ・カメラ１２３は、取り込んだ画像をデジタル化するビデオ・カメラ１２３の外部のデジタル・ビデオ・キャプチャ・ハードウェアを有するアナログ・ビデオ・カメラであってもよい。

【００１１】

第２のＩ／Ｏバス１３０は、単一のバスまたは多数のバスの組合せとすることができる。例として、第２のＩ／Ｏバス１３０は、ＰＣＩバス、ＰＣＭＣＩＡバス、NuBus、業界標準アーキテクチャ（ＩＳＡ）バス、または他のバスを含むことができる。第２のＩ／Ｏバス１３０は、コンピュータ・システム１００において構成要素間の通信リンクとなっている。Ｉ／Ｏバス１３０には記憶媒体１３１が接続されている。記憶媒体１３１は、データを記憶するためにデータ記憶デバイスを含むことができる。データ記憶デバイスは、ハード・ディスク・ドライブ、フロッピー・ディスク・ドライブ、ＣＤ－ＲＯＭデバイス、フラッシュ・メモリ・デバイス、または他の記憶デバイスとすることができる。記憶媒体１３１は、１つまたは複数の説明したデータ記憶デバイスを含んでもよい。

10

【００１２】

第２のＩ／Ｏバス１３０にはキーボード・インターフェイス１３２が接続されている。キーボード・インターフェイス１３２は、キーボード・コントローラまたは他のキーボード・インターフェイスとすることができる。キーボード・インターフェイス１３２は専用デバイスであってもよく、またはバス・コントローラまたは他のコントローラなどの別の装置に常駐することができる。キーボード・インターフェイス１３２は、キーボードをコンピュータ・システム１００に接続させ、データ信号をキーボードからコンピュータ・システム１００に送信する。第２のＩ／Ｏバス１３０にはオーディオ・コントローラ１３３が光学的に接続されている。オーディオ・コントローラ１３３は、音の記録および再生を調整するように動作し、またＩ／Ｏバス１３０に接続される。

20

【００１３】

バス・ブリッジ１２４は、第１のＩ／Ｏバス１２０を第２のＩ／Ｏバス１３０に接続する。バス・ブリッジ１２４は、第１のＩ／Ｏバス１２０と第２のＩ／Ｏバス１３０との間でデータ信号を緩衝しブリッジするように動作する。

【００１４】

本発明は、記憶媒体内のファイルを管理するためのコンピュータ・システム１００の使用に関する。一実施形態によれば、記憶媒体内のファイルを管理することは、プロセッサ１０１が主メモリ１１３において命令のシーケンスを実行することに反応して、コンピュータ・システム１００によって行われる。かかる命令は、データ記憶デバイス１３１などの別のコンピュータ可読媒体から、またはネットワーク・コントローラ１２１を介して別のソースから、メモリ１１３に読み込むことができる。命令のシーケンスの実行は、後で説明するように、プロセッサ１０１に記憶媒体内のファイルを管理させる。代替実施形態においては、本発明を実施するために、ソフトウェア命令の代わりにまたはそれと組み合わせでハードワイヤド回路を使用してもよい。このように、本発明は、ハードウェア回路およびソフトウェアのいかなる特定の組合せにも限定されない。

30

【００１５】

図２は、記憶媒体１３１内において適切にリンクされたファイルを示したブロック図である。記憶媒体１３１は、複数のファイル２１０、２２０、２３０、２４０および２５０を記憶している。記憶媒体１３１は、第２のファイル２２０に関する依存性を有する第１のファイル２１０を含む。第１のファイル２１０は、第１のファイルが第２のファイル２２０からデータを取得できるように、第１のファイルを第２のファイル２２０に直接リンクする第２のファイル２２０へのリファレンス２１１を含む。リファレンス２１１は、第２のファイル２２０の場所を識別するパス名を含むことができる。

40

【００１６】

第１のファイル２１０は実行可能ファイルとすることができる。実行可能ファイルは、プロセッサが直接実行できる形式でデータを含む。第１のファイル２１０は、第２の実行可

50

能ファイル、ダイナミック・リンク・ライブラリ・ファイル、データ・ファイル、または他のファイルへのリファレンスを含むことができる。第1のファイル210は、ダイナミック・リンク・ライブラリ・ファイルであってもよい。ダイナミック・リンク・ライブラリは、他のファイルによって使用できる実行可能な関数またはデータのライブラリである。通常、ダイナミック・リンク・ライブラリ・ファイルは、1つまたは複数の特定の関数を提供し、ファイルは、ダイナミック・リンク・ライブラリ・ファイルへの静的または動的なリンクを生成することにより関数にアクセスする。静的なリンクは、プログラムの実行中は一定のままであり、一方、動的なリンクは必要に応じてプログラムによって生成される。ダイナミック・リンク・ライブラリ・ファイルはデータを含んでもよい。ダイナミック・リンク・ライブラリ・ファイルは通常、DLLファイル名拡張子を有する。ダイナミック・リンク・ライブラリ・ファイルは、第2のダイナミック・リンク・ライブラリ・ファイルまたは他のファイルへのリファレンスを含むことができる。第1のファイル210は初期化ファイルであってもよい。初期化ファイルは通常、構成情報を含むプレーン・テキスト・ファイルである。これらのファイルは、優先順位および動作環境に関する情報を保存するために他のファイルによって使用される。初期化ファイルは通常、INIファイル名拡張子を有する。初期化ファイルは、実行可能ファイル、データ・ファイル、または他のファイルへのリファレンスを含むことができる。第1のファイル210はショート・カット・ファイルであってもよい。ショート・カット・ファイルは、ユーザのデスクトップにアイコンを作成するファイルである。選択されたときに、ショート・カットはそのショート・カットがリンクされているファイル内のプログラムを実行する。ショート・カット・ファイルは実行可能ファイルへのリンクを含むことができる。ショート・カット・ファイルは通常、プログラム情報ファイル(PIF)拡張子またはリンク(LNK)ファイル名拡張子を有する。

【0017】

第2のファイル220は、ダイナミック・リンク・ライブラリ・ファイル、実行可能ファイル、または他のファイルである第1のファイル210にリンクされた、ダイナミック・リンク・ライブラリ・ファイルであってもよい。第2のファイル220は、実行可能ファイル、ダイナミック・リンク・ライブラリ・ファイル、初期化ファイル、ショート・カット・ファイル、または他のファイルである第1のファイル210にリンクされた、実行可能ファイルであってもよい。第2のファイル210は、実行可能ファイル、初期化ファイル、または他のファイルにリンクされる場合があるデータ・ファイルであってもよい。

【0018】

記憶媒体131は、第4のファイル240に関する依存性を有する第3のファイル230を含む。第3のファイル230は、レジストリ・ファイル250を介して第4のファイル240にリンクされている。レジストリ・ファイル250は、構成情報、ファイル・コンテンツ、およびファイル場所情報を記憶するためにオペレーティング・システムによって使用されるデータベースである。レジストリ・ファイル250は、ファイル場所情報を記憶するパス名ファイル251を含む。第3のファイル230は、第3のファイル230によって要求されたデータを識別するグローバル・ユニーク識別子(GUID)231へのリファレンスを含む。データは、ダイナミック・リンク・ライブラリ・ファイル、実行可能ファイル、データ・ファイル、または他のデータの関数であってもよい。GUIDは、ファイルを要求されたデータに関連付けるために、オペレーティング・システムによって使用される。オペレーティング・システムはレジストリ・ファイル250をサーチし、データを有するファイルを第3のファイル230にリンクする。レジストリ・ファイル250は、ファイルのパス名などデータを備えたファイルのファイル場所情報を含むので、オペレーティング・システムはデータを備えたファイルを第3のファイル230とリンクすることができる。図2によれば、第4のファイル240はデータを有し、レジストリ250を介して第3のファイル230にリンクされている。

【0019】

第3のファイル230は、ダイナミック・リンク・ファイル、実行可能ファイル、または

他のファイルであってもよい。第４のファイルは、ダイナミック・リンク・ファイル、実行可能ファイル、または他のファイルである第３のファイルにリンクされたダイナミック・リンク・ライブラリ・ファイルであってもよい。第４のファイル２４０は、ダイナミック・リンク・ライブラリ・ファイル、実行可能ファイル、または他のファイルにリンクされた実行可能ファイルであってもよい。第４のファイル２４０は、ダイナミック・リンク・ライブラリ・ファイル、実行可能ファイル、または他のファイルにリンクされたデータ・ファイルであってもよい。

【００２０】

図３は、壊れたリンクを有するファイルを示したブロック図である。データ・ファイルが記憶媒体内で移動されたとき、そのファイルには新しいパス名が与えられる。移動されたファイルのパス名が、移動されたファイルに依存しているファイル内で、または移動されたファイルをリンクする責任を有するレジストリ内で更新されない限り、移動されたファイルへのリンクは壊れる。図３は図２に示したファイルを図示しており、ここで、第２のファイル２２０は記憶媒体１３１内の新しい場所に移動している。記憶媒体１３１内の新しい場所は、たとえば、第２のファイル２２０が新しいパス名を与えられている新しいドライブの場所または新しいディレクトリであろう。第１のファイル２１０は、第２のファイル２２０の、ブロック３２０で示された古い場所へのリファレンス２１１を含む。したがって、第１のファイル２１０が第２のファイル２２０のファイルにアクセスしようとすると、第１のファイル２１０は第２のファイル２２０とインターフェイスできなくなる。

【００２１】

図３は、記憶媒体１３１内の新しい場所に移動された第４のファイル２４０を示している。記憶媒体１３１内の新しい場所は、第４のファイル２４０が新しいパス名を与えられている新しいドライブ場所または新しいディレクトリであろう。レジストリ・ファイル２５０は、第４のファイルの、ブロック３４０で示された古い場所へのリファレンスを含むパス名ファイル２５１を含む。したがって、オペレーティング・システムが、第３のファイル２３０に第４のファイル２４０のデータにアクセスさせるために、第３のファイル２３０を第４のファイル２４０とリンクさせようとするとき、第３のファイル２３０は第４のファイル２４０とインターフェイスできないであろう。

【００２２】

図４は、本発明の実施形態による記憶媒体マネージャ４００を実現したモジュールを示したブロック図である。本発明の実施形態において、このモジュールはソフトウェアによって実現され、記憶媒体１１３（図１に示した）内に命令のシーケンスとして常駐している。このモジュールは、バス１２０（図１に示した）に接続された構成要素としてハードウェアによって、またはハードウェアとソフトウェア両方の組合せによって実現されてもよいことを理解されたい。

【００２３】

ブロック４１０はマネージャ・インターフェイスを表している。マネージャ・インターフェイス４１０は、記憶媒体１３１（図１に示した）内の１つの場所から別の場所に移動されたファイルのアイデンティティを取得する。本発明の実施形態によれば、マネージャ・インターフェイス４１０は、コンピュータ・システム１００のオペレーティング・システムから直接この情報を取得できる。本発明の代替実施形態によれば、マネージャ・インターフェイス４１０は、ユーザが記憶媒体上の１つの場所から別の場所に移動させているファイルをユーザに識別させるユーザ・インターフェイスを含んでもよい。ファイルのアイデンティティは、ドライブおよび、ファイルが記憶されたあらゆるディレクトリおよびサブ・ディレクトリ、ファイル名、ならびにファイルの拡張子を含むことができるファイルのパス名とすることができる。

【００２４】

マネージャ・インターフェイス４１０には依存性サーチ・ユニット４２０が接続されている。依存性サーチ・ユニット４２０は、マネージャ・インターフェイス４１０から移動したファイルのアイデンティティを受け取る。依存性サーチ・ユニット４２０は、移動して

おり、移動したファイルの以前の場所にリンクされたファイルに依存するあらゆるファイルのアイデンティティを求めて記憶媒体 1 3 1 をサーチする。依存性サーチ・ユニット 4 2 0 は、移動したファイルを検索するあらゆるファイルを求めて記憶媒体 1 3 1 をサーチする。リファレンスは移動されたファイルへの依存性リンクを提供する。依存性サーチ・ユニット 4 2 0 はレジストリ・ファイル・インターフェイス 4 2 1 を含む。レジストリ・ファイル・インターフェイス 4 2 1 は、移動されたファイルへのリファレンスを求めて記憶媒体 1 3 1 内のレジストリ・ファイルをサーチする。レジストリ・ファイル・インターフェイス 4 2 1 は、レジストリ・ファイルによって検索されたファイルのパス名などのファイル場所情報を記憶するレジストリ・ファイル内のパス名ファイルにアクセスする。レジストリ・ファイルは、たとえば実行可能ファイル、ダイナミック・リンク・ライブラリ・ファイル、データ・ファイル、アプリケーション特定データ・ファイル、または他のファイルを含むことができる移動されたファイルを検索してもよい。依存性サーチ・ユニット 4 2 0 はファイル・インターフェイス 4 2 2 を含む。ファイル・インターフェイス 4 2 2 は、移動されたファイルへのリファレンスを求めて記憶媒体 1 3 1 内の非レジストリ・ファイルをサーチする。リファレンスは、移動されたファイルへの依存性リンクを提供する。ファイル・インターフェイス 4 2 2 は、記憶媒体 1 3 1 上の全ての非レジストリ・ファイルにアクセスし、移動されたファイルのパス名などのリファレンスを求めて非レジストリ・ファイルをサーチする。アクセスできる非レジストリ・ファイルは、ダイナミック・リンク・ライブラリ・ファイル、実行可能ファイル、初期化ファイル、ショート・カット・ファイル、およびデータ・ファイルを含む。非レジストリ・ファイルは、たとえば実行可能ファイル、ダイナミック・リンク・ライブラリ・ファイル、データ・ファイル、アプリケーション特定データ・ファイル、または他のファイルを含むことができる移動されたファイルを検索してもよい。レジストリ・インターフェイス 4 2 1 またはファイル・インターフェイス 4 2 2 が、アプリケーション特定データ・ファイル内の移動されたファイルへのリファレンスを発見すると、ファイル・インターフェイス 4 2 2 は、これが更新不可能なファイルであることを認識する。

【 0 0 2 5 】

依存性サーチ・ユニット 4 2 0 には更新ユニット 4 3 0 が接続されている。更新ユニット 4 3 0 は、移動されたファイルに依存する更新可能なファイルのアイデンティティを受け取る。更新ユニット 4 3 0 は、移動されたファイルに依存するファイル内の移動されたファイルのリファレンスを更新する。更新ユニット 4 3 0 は、パス書込みユニット 4 3 1 を含む。パス書込みユニット 4 3 1 は、移動されたファイルの新しい場所を反映する新しいパス名で、移動されたファイルの古いパス名を上書きする。本発明の実施形態によれば、パス書込みユニット 4 3 1 は、レジストリ・ファイル、初期化ファイル、および他のファイルにおいて検索された移動したファイルの古いパス名を上書きする。更新ユニット 4 3 0 はパス更新ユニット 4 3 2 を含む。パス更新ユニット 4 3 2 は、移動されたファイルに依存するファイルのサーチ・パスを、移動されたファイルの新しいパス名を含むように更新する。本発明の実施形態によれば、パス更新ユニット 4 3 2 は、ダイナミック・リンク・ライブラリ・ファイル、ショート・カット・ファイル、および他のファイルにおいて参照された移動されたファイルのサーチ・パスを更新する。

【 0 0 2 6 】

依存性サーチ・ユニット 4 2 0 にはフラグging・ユニット 4 4 0 が接続されている。フラグging・ユニット 4 4 0 は、移動されたファイルを検索する更新不可能なファイルのアイデンティティを受け取る。アプリケーション特定ファイル、およびアプリケーション特定ファイルを検索するファイルは、通常、ファイルの更新で問題が生じる専用データ形式でかかっている。ファイルがアプリケーション特定データ・ファイルの更新されたパス名を検索することを防止するために、フラグging・ユニット 4 4 0 は、移動されたファイルをその元の場所に戻して、ユーザに通知する。本発明の代替実施形態によれば、フラグging・ユニット 4 4 0 は、移動されたファイルをその元の場所に戻すことなく、ユーザに壊れたリンクについて通知する。

10

20

30

40

50

【 0 0 2 7 】

マネージャ・インターフェイス 4 1 0、依存性サーチ・ユニット 4 3 0、レジストリ・インターフェイス 4 2 1、ファイル・インターフェイス 4 2 2、更新ユニット 4 3 0、パス書込みユニット 4 3 1、パス更新ユニット 4 3 2、およびフラグging・ユニット 4 4 0 は、あらゆる既知の回路またはあらゆる既知の技術を用いて実施することができる。記憶媒体マネージャ 4 0 0 がハードウェアで実施されている本発明の実施形態においては、依存性サーチ・ユニット 4 2 0、レジストリ・インターフェイス 4 2 1、ファイル・インターフェイス 4 2 2、更新ユニット 4 3 0、パス書込みユニット 4 3 1、パス更新ユニット 4 3 2、およびフラグging・ユニット 4 4 0 は全て単一のシリコン基板に常駐する。

【 0 0 2 8 】

図 5 は、本発明の実施形態による記憶媒体を管理する方法を示したフローチャートである。ステップ 5 0 1 で、記憶媒体上で第 1 の場所から第 2 の場所に移動したファイルが識別される。本発明の実施形態によれば、移動されたファイルは、オペレーティング・システムまたはコンピュータ・システムのユーザによって識別できる。

【 0 0 2 9 】

ステップ 5 0 2 で、記憶媒体は、移動された第 2 のファイルへのリファレンスを有する第 1 のファイルを求めてサーチされる。本発明の実施形態によれば、レジストリ・ファイルおよび非レジストリ・ファイルは、第 2 のファイルへのリファレンスを求めてサーチされる。第 2 のファイルへのリファレンスは、第 2 のファイルのパス名であってもよい。サーチされる非レジストリ・ファイルは、ダイナミック・リンク・ライブラリ・ファイル、実行可能ファイル、データ・ファイル、初期化ファイル、ショート・カット・ファイル、または他のファイルを含んでもよい。

【 0 0 3 0 】

ステップ 5 0 3 で、移動された第 2 のファイルを検索する第 1 のファイルが更新可能なファイルであるかどうか判断される。第 1 のファイルが更新可能なファイルであれば、制御はステップ 5 0 5 に進む。第 1 のファイルが更新不可能なファイルであれば、制御はステップ 5 0 4 に進む。本発明の実施形態によれば、アプリケーション特定データ・ファイルを検索するファイルは、更新不可能なファイルである。

【 0 0 3 1 】

ステップ 5 0 4 で、第 1 のファイルと移動された第 2 のファイルとの間に壊れたリンクが存在することを示す通知が送られる。本発明の実施形態によれば、移動された第 2 のファイルはその最初の場所に戻される。

【 0 0 3 2 】

ステップ 5 0 5 で、第 1 のファイル内のリファレンスは、第 2 のファイルが第 2 の場所に移動したことを示すように更新される。本発明の実施形態によれば、リファレンスを更新することは、第 2 のファイルに関するサーチパスを更新するか、第 2 のファイルへのリファレンスをその新しいパス名で上書きすることにより達成できる。

【 0 0 3 3 】

上記の説明においては、本発明をその特定の例示的实施形態を検索して説明している。しかし、特許請求の範囲に記載した本発明のより広い主旨および範囲から逸脱することなく、これらの実施形態に対して様々な修正や変更を行えることが明らかになるであろう。明細書および図面は、限定的な意味ではなく例示的な意味で捉えられるものとする。

【 図面の簡単な説明 】

【 図 1 】 本発明の実施形態を実施するコンピュータ・システムのブロック図である。

【 図 2 】 記憶媒体内で適切にリンクされたファイルを示すブロック図である。

【 図 3 】 壊れたリンクを有する記憶媒体内のファイルを示すブロック図である。

【 図 4 】 本発明の実施形態によるリンク更新ユニットのブロック図である。

【 図 5 】 本発明の実施形態による記憶媒体上でファイルを管理する方法を示したフローチャートである。

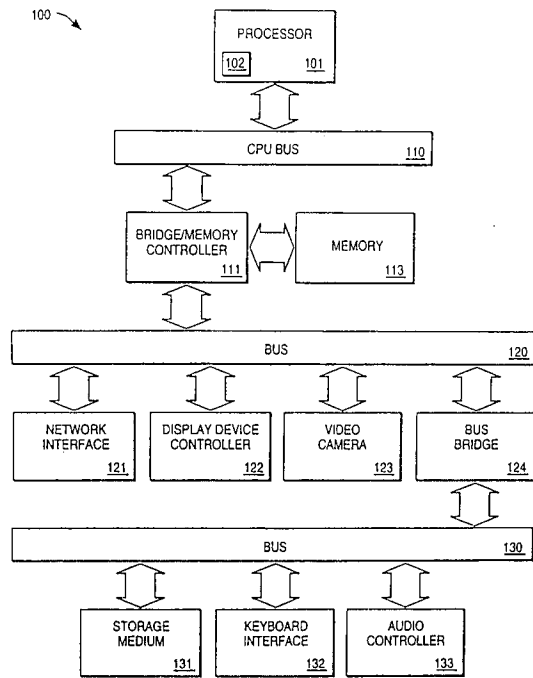
10

20

30

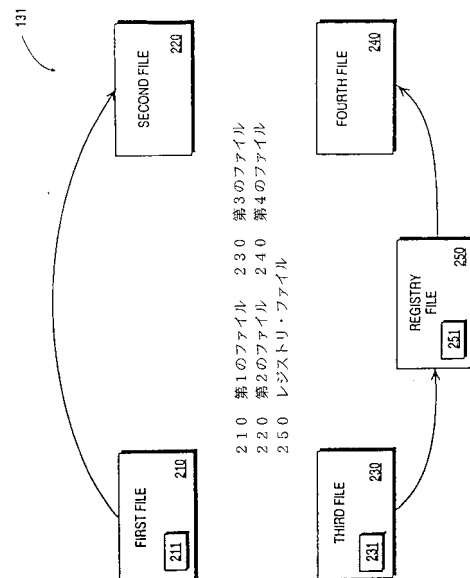
40

【図 1】

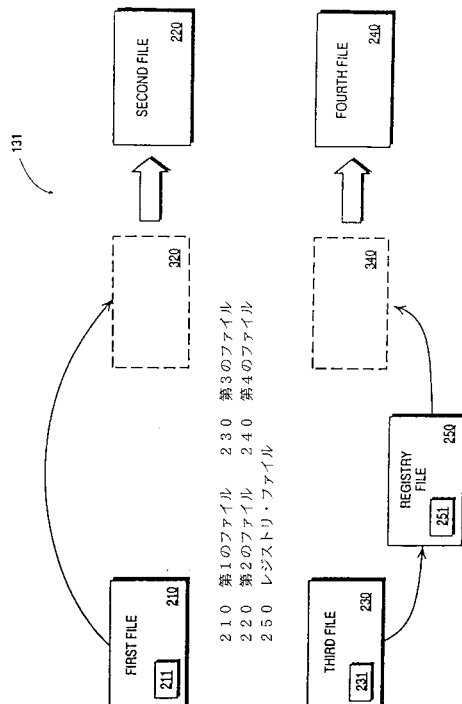


101 プロセッサ
 102 CPUバス
 110 CPUバス
 111 ブリッジ/メモリ・コントローラ
 113 メモリ
 120 バス
 121 ネットワーク・インターフェイス
 122 ディスプレイ・デバイス・コントローラ
 123 ビデオ・カメラ
 124 バス・ブリッジ
 130 バス
 131 記憶媒体
 132 キーボード・インターフェイス
 133 オーディオ・コントローラ

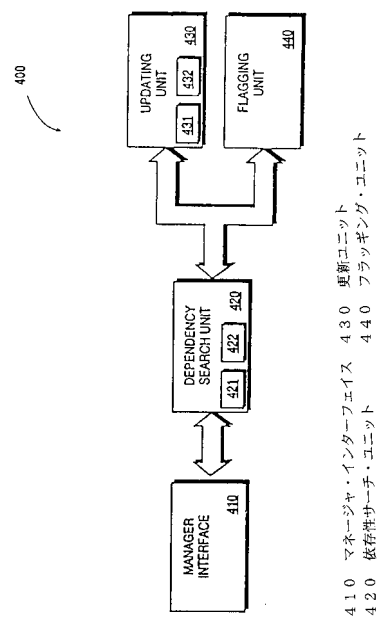
【図 2】



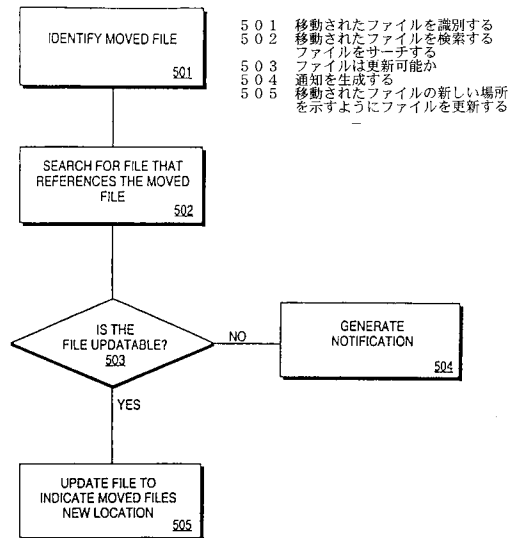
【図 3】



【図 4】



【図 5】



フロントページの続き

(72)発明者 グロス, マーク・ティ

アメリカ合衆国・97229・オレゴン州・ポートランド・ノースウエスト ブラックコム ドラ
イブ・3455

審査官 坂庭 剛史

(56)参考文献 特開平09-288606(JP, A)

特開平09-185633(JP, A)

特開平09-293010(JP, A)

特開平09-034755(JP, A)

特開平08-161215(JP, A)

特開平08-161210(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/44

G06F 12/00

G06F 17/30