

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2009-522677
(P2009-522677A)

(43) 公表日 平成21年6月11日(2009.6.11)

(51) Int.Cl. F I テーマコード (参考)
G06F 12/00 (2006.01) G06F 12/00 520P 5B082
 G06F 12/00 531D

審査請求 未請求 予備審査請求 未請求 (全 23 頁)

(21) 出願番号 特願2008-548967 (P2008-548967)
 (86) (22) 出願日 平成18年12月20日 (2006.12.20)
 (85) 翻訳文提出日 平成20年7月1日 (2008.7.1)
 (86) 国際出願番号 PCT/EP2006/070022
 (87) 国際公開番号 W02007/077132
 (87) 国際公開日 平成19年7月12日 (2007.7.12)
 (31) 優先権主張番号 11/325,750
 (32) 優先日 平成18年1月5日 (2006.1.5)
 (33) 優先権主張国 米国 (US)

(71) 出願人 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
 (74) 代理人 100108501
 弁理士 上野 剛史
 (74) 代理人 100112690
 弁理士 太佐 種一
 (74) 代理人 100091568
 弁理士 市位 嘉宏

最終頁に続く

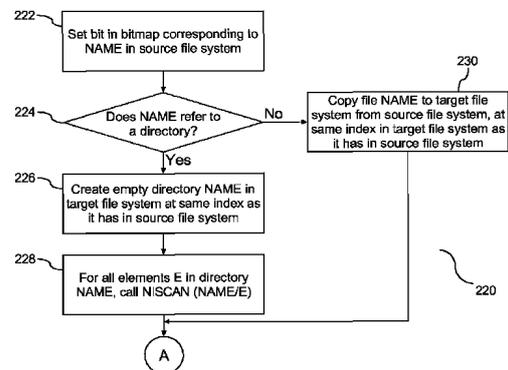
(54) 【発明の名称】 ノードの番号付けによるファイル・システムのダンプ/復元のための方法、システム、およびデバイス

(57) 【要約】

【課題】データ構造として編成されたファイル・システムをソース・サーバからターゲット・サーバへ複製するための、方法およびシステムを提供すること。

【解決手段】この複製プロセスは、複製プロセスのすべての段階で、ターゲット・ファイル・システム内のiノード・テーブル・エントリの順序、ならびにターゲット・サーバ上のデータ構造の複製の内部整合性を保持する。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

データ構造として編成されたソース・ファイル・システムをツリー順にスキャンするステップと、

前記ソース・ファイル・システム内のディレクトリおよびディレクトリ・インデックスを識別するステップと、

ターゲット・ファイル・システム内でディレクトリ・インデックスを含む前記ソース・ファイル・システムからディレクトリを作成するステップと、

前記ソース・ファイル・システム内のディレクトリから、前記ターゲット・ファイル・システム内の対応するディレクトリへと、コンテンツを再帰的にコピーするステップと、
を含み、前記ターゲット・ファイル・システム内の前記ディレクトリ・インデックスは前記ソース・ファイル・システム内の前記ディレクトリ・インデックスと同じである、
ファイル・システムを複製するための方法。

10

【請求項 2】

ディレクトリ・コンテンツの複製に続いて、前記ソース・ファイル・システムのテーブル内にビットを設定するステップをさらに含む、請求項 1 に記載の方法。

【請求項 3】

ディレクトリ・エントリなしのオブジェクトに関する前記ソース・ファイル・システム内のノード・テーブルを検索するステップをさらに含む、請求項 2 に記載の方法。

20

【請求項 4】

前記ディレクトリ・エントリなしの前記オブジェクトを、前記ソース・ファイル・システムから、前記識別インデックスで、前記ターゲット・ファイル・システムにコピーするステップをさらに含む、請求項 3 に記載の方法。

【請求項 5】

前記ソース・ファイル・システム内のディレクトリからコンテンツを再帰的にコピーするステップが、前記ソース・ファイル・システム内の各オブジェクトに関する前記識別インデックスを、前記ターゲット・ファイル・システム内の前記識別インデックスでの各オブジェクトの属性として、前記ターゲット・ファイル・システムに伝送するステップを含む、請求項 1 に記載の方法。

【請求項 6】

プロセッサと、
メモリと、

を備えるコンピュータ・システムであって、

前記プロセッサは前記メモリに結合され、前記プロセッサは、

データ構造として編成されたファイル・システムをツリー順にスキャンする操作と、

前記スキャン済みファイル・システム内のディレクトリおよびディレクトリ・インデックスを識別する操作と、

ディレクトリ・インデックスを含む前記スキャン済みファイル・システムから、ディレクトリを作成する操作であって、前記作成されたディレクトリが前記スキャン済みディレクトリからリモートの異なるファイル・システム内にある、作成する操作と、

40

前記スキャン済みファイル・システム内のディレクトリから、前記異なるファイル・システム内の対応するリモート・ディレクトリへと、コンテンツを再帰的にコピーする操作と、を含む、操作を実行するように動作可能である、コンピュータ・システム。

【請求項 7】

コンテンツの複製完了後に、前記スキャン済みファイル・システム内の前記配置されたテーブル内に設定されるように適合されたビットをさらに備える、請求項 6 に記載のシステム。

【請求項 8】

ディレクトリ・エントリなしに前記スキャン済みファイル・システム内にオブジェクトを配置するように適合されたオブジェクト・レプリケータをさらに備える、請求項 7 に記

50

載のシステム。

【請求項 9】

前記再帰的コピー操作が、前記オブジェクトが再作成されている場合、前記スキャン済みファイル・システム内の各オブジェクトに関する識別番号を、その識別番号での各オブジェクトの属性として伝送する、請求項 7 に記載のシステム。

【請求項 10】

ファイル・システムを複製するための操作を実行するためにコンピュータによって実行可能な命令を確実に記録するコンピュータによって読み取り可能な、プログラム・ストレージ・デバイスであって、前記操作が、

データ構造として編成されたソース・ファイル・システムをツリー順にスキャンする操作と、

前記ソース・ファイル・システム内のディレクトリおよびディレクトリ・インデックスを識別する操作と、

ターゲット・ファイル・システム内でディレクトリ・インデックスを含む前記ソース・ファイル・システムからディレクトリを作成する操作と、

前記ソース・ファイル・システム内のディレクトリから、前記ターゲット・ファイル・システム内の対応するディレクトリへと、コンテンツを再帰的にコピーする操作と、を含み、前記ターゲット・ファイル・システム内の前記ディレクトリ・インデックスは前記ソース・ファイル・システム内の前記ディレクトリ・インデックスと同じである、プログラム・ストレージ・デバイス。

【請求項 11】

ディレクトリ・コンテンツの複製に続いて、前記ソース・ファイル・システムのテーブル内にビットを設定するための操作をさらに含む、請求項 10 に記載のプログラム・ストレージ・デバイス。

【請求項 12】

ディレクトリ・エントリなしのオブジェクトに関する前記ソース・ファイル・システム内のノード・テーブルを検索するための操作をさらに含む、請求項 11 に記載のプログラム・ストレージ・デバイス。

【請求項 13】

前記ディレクトリ・エントリなしの前記オブジェクトを、前記ソース・ファイル・システムから、前記識別インデックスで、前記ターゲット・ファイル・システムにコピーするための操作をさらに含む、請求項 12 に記載のプログラム・ストレージ・デバイス。

【請求項 14】

前記ソース・ファイル・システム内のディレクトリからコンテンツを再帰的にコピーするための操作が、前記ソース・ファイル・システム内の各オブジェクトに関する前記識別インデックスを、前記ターゲット・ファイル・システム内の前記識別インデックスでの各オブジェクトの属性として、前記ターゲット・ファイル・システムに伝送する操作を含む、請求項 10 に記載のプログラム・ストレージ・デバイス。

【請求項 15】

ファイル・システムを複製するための操作を実行するためにコンピュータによって実行可能な命令を確実に記録するコンピュータによって読み取り可能な、プログラム・ストレージ・デバイスであって、前記操作が、

データ構造として編成されたソース・ファイル・システムをノード順にスキャンする操作と、

前記スキャンで見つめられた各ディレクトリのコンテンツを銘記する操作と、

ディレクトリに関する数値識別子が、前記ディレクトリ内の各オブジェクトに関する識別子に先行するように、ターゲット・ファイル・システムへノードを報告する操作と、を含む、プログラム・ストレージ・デバイス。

【請求項 16】

ディレクトリおよびファイルに関する前記識別子を、オブジェクトが複製されている間

10

20

30

40

50

に各オブジェクトの属性として、前記ソース・ファイル・システムから前記ターゲット・ファイル・システムへと伝送するための操作をさらに含む、請求項 15 に記載のプログラム・ストレージ・デバイス。

【請求項 17】

ターゲット・ファイル・システムに関するディレクトリを、それらディレクトリのコンテンツを作成する前に作成するための操作をさらに含む、請求項 15 に記載のプログラム・ストレージ・デバイス。

【請求項 18】

ソース・ファイル・システムをノード順にスキャンするための、非再帰的ルーチンである操作をさらに含む、請求項 15 に記載のプログラム・ストレージ・デバイス。

【請求項 19】

前記ソース・ファイル・システム内の複製されたデータを、前記ターゲット・ファイル・システムに報告されている間に再順序付けするための操作をさらに含む、請求項 15 に記載のプログラム・ストレージ・デバイス。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ファイル・システムの複製 (replication) に関する。とりわけ本発明は、ソース・ファイル・システムからターゲット・ファイル・システムへデータを複製しながら、テーブル・エントリを保持するための方法およびツールに関する。

【背景技術】

【0002】

ファイルとは、ユーザにとっては、ストレージ・メディア内に保持されたデータの単一の連続ブロックとして見える、関連情報の名前付き集合である。データ・ブロックとは、ファイル用の実際のデータを格納するために使用される構造体である。ファイル・システムとは、ストレージ・メディア上のデータおよびメタデータの構造であり、それらメディア上でのデータの読み取り/書き込みを可能にする。一実施形態では、ファイル・システムは、コンピュータ上でファイルを編成するために使用されるディレクトリの階層、すなわちディレクトリ・ツリーである。i ノードとは、メタデータなどのファイルに関する情報を格納するために使用されるファイル・システム上のデータ構造体である。i ノードに含まれる情報は、ファイルの所有権、ファイルへのアクセス許可、ファイルのサイズ、ファイル・タイプ、および、ファイルに関するデータ・ブロックのディスク上の場所への参照を含むことができる。i ノードは、しばしばディスク・メディア上の連続テーブル内に格納され、ファイルの i ノード番号は整数であって、これはこのテーブル内のその i ノードのインデックスである。ファイルが作成されると、名前および i ノード番号の両方が割り当てられる。ファイルは、i ノード内にルートを持っていることから i ノード番号を有し、何らかのディレクトリ内にファイル用に作成されたエントリを持っていることから名前を有する。ディレクトリ内のデータは、最小限、ファイル名のペアならびにそれらの対応する i ノード番号のリストであり、ディレクトリそれ自体は親ディレクトリ内にエントリを有することになり、すなわち、ほとんどのディレクトリは何らかの他のディレクトリのサブディレクトリであることに留意されたい。ファイル・システム内に明示的な親ディレクトリを持たないのは、ファイル・システムのルート・ディレクトリのみである。ユーザまたはプログラムが名前によってファイルを参照する場合は必ず、システムはその名前を使用してファイル・システム内のディレクトリを検索する。検索はルートから始まり、ファイルの完全な名前が使用され、検索によってそのファイルに関する i ノードが見つかるまで、引き続いてサブディレクトリを読み取りおよび検索し、これによってシステムは、さらなる動作を実行するためにファイルに関して必要な情報、すなわちメタデータを取得することができる。

【0003】

ファイル・システムは、カプセル化された i ノード・テーブル・インデックスへの参照

をファイル・システムのクライアントに渡すことによって、ソース位置からターゲット位置へとネットワークを横切ってエクスポートすることができる。本明細書で使用される場合、「ソース」という用語は、本発明の対象データの移動元である場所として広義に定義することが可能であり、「ターゲット」という用語は、データの移動先である場所として定義することが可能である。図1は、ソース・ファイル・サーバ(20)およびターゲット・ファイル・サーバ(40)を含む、相互接続されたファイル・サーバのペアを含む、ストレージ・システム環境を示すブロック図(10)である。ソースおよびターゲットのファイル・サーバ(20)および(40)は、それぞれ、ローカルまたはワイドのエリア・ネットワークを備えることが可能なネットワーク(30)を介してリンクされる。それぞれがネットワーク接続(22)、(42)を介してネットワーク(30)と通信する各ファイル・サーバ(20)および(40)と通信しながらその内部に常駐する、適切なネットワーク・アダプタ(32)、(52)が、ネットワーク(30)を介した通信を容易にする。ソースおよびターゲットのファイル・サーバ(20)および(40)は、それぞれ、プロセッサ(26)、(46)、メモリ(28)、(48)、ならびにネットワーク・アダプタ(32)、(52)を含む。各ファイル・サーバ(20)、(40)は、ストレージ・メディア上のディレクトリおよびファイルの階層構造として情報を論理的に編成するためにファイル・システムを実装する、ストレージ・オペレーティング・システム(図示せず)も含む。

10

20

30

40

50

【0004】

ネットワーク全体にわたってファイル・システムを複製するための方法は、いくつか知られている。ファイル・システムを複製するための従来技術の方法の1つは、ソース・ファイル・システムのディレクトリ・ツリーを複製する。ソース・ファイル・システム側のアプリケーションは、提示されたディレクトリ・ツリーをトラバースし、各ファイルおよびディレクトリを宛先ファイル・システムにコピーする。図2は、ソース・ファイル・システムのサンプルiノード・テーブルを示す従来技術のブロック図(100)である。図に示されるように、iノード・テーブル内には、ルート・ディレクトリ(102)に加えて3つのエントリがあり、そのエントリはそれぞれ、4(104)、6(106)、および7(108)のインデックスにある。この例では、インデックス1(102)がルート・ディレクトリであり、インデックス4(104)は編集プログラム用の実行可能イメージに関するエントリを有し、インデックス6(106)はサブディレクトリであり、インデックス7(108)は読み取り可能なテキスト・ファイルを有する。図に示されるように、インデックス2(110)、3(112)、および5(114)にはエントリがない。この空インデックスの理由は様々である。たとえば、あるインデックスは、かつてはインデックス内に常駐したファイルが削除された可能性があるため、空である場合がある。

【0005】

図3は、前述の従来技術のiノード・テーブル複製プロセスを使用して、図2のサンプル・ソースiノード・テーブルから解釈された、ターゲット・ファイル・システムでのサンプルiノード・テーブルを示すブロック図(150)である。図に示されるように、ソース・ファイル・システムからのノード・テーブル・インデックスの順序は複製プロセス時には保持されず、空インデックスは保持されなかった。ノード・テーブル・インデックスの順序は特有ではなく、アルファベット順、番号順などで表すことができる。したがって、たとえターゲット・ファイル・システムでの所与のファイルが、ソース・ファイル・システムでの同じファイルのコピーであっても、それらのネットワーク・ファイル参照は交換可能ではない。

【0006】

ファイル・システムを複製するための他の従来技術の方法は、iノード・テーブル内の各ファイルおよびディレクトリを番号順に複製することによって、ファイル・システムをコピーする。たとえば、第1ノードに対応するファイルは、ソース・ファイル・システムからターゲット・ファイル・システムへとコピーされる。この後に、第2ノードに対応するファイルが続き、それぞれのファイルは、ソース・ファイル・システムのノード・テー

ブル内にあるそれぞれの連続ノードに対応する。しかしながら、この方法に関連付けられる欠点の1つは、ファイルが親ディレクトリによって参照されることに関連した任意の順序でコピーされるため、ターゲット・ファイル・システムの間状態がそれ自体と整合しなくなることである。たとえば、ファイルへの参照としての働きをするディレクトリ・エントリが、ファイル自体がコピーされる前にターゲット・ファイル・システムにコピーされた場合、ファイルに関するディレクトリ・エントリはまだ存在しないことになり、非整合状態 (inconsistent state) が生成される。これとは逆に、あるファイルに関するディレクトリ・エントリがソース・ファイル・システムからターゲット・ファイル・システムへとコピーされる前に、そのファイルがソース・ファイル・システムからターゲット・ファイル・システムへとコピーされた場合、別の非整合状態が生成される可能性がある。この中間状態が、存在し、スペースを消費するが、いかなるディレクトリ・エントリによっても指示されない、すなわち名前を持たない、ファイルを生成する。コピーが割り込みなしに完了した場合、ソースおよびターゲットのファイル・システム内のファイルが同じiノード番号を有することになり、すべてのファイルがディレクトリ・エントリを有することになり、ソース・ファイル・システムの場合と同様に、すべてのディレクトリ・エントリがファイルに対応することになるため、これらの中間状態はどちらも実際的には意味のないものである (moot)。しかしながら、2つの識別された中間状態に関連付けられた重大な欠点があり、これは、ターゲット・ファイル・システム内に第1のエントリが作成された後に、複製プロセスで割り込みが発生した場合に生じる。従来技術の複製プロセスにおける割り込みは、結果として、自己非整合型 (self-inconsistent) であり復活不可能 (not salvageable) な部分的ターゲット・ファイル・システムを生じさせる。

【発明の開示】

【発明が解決しようとする課題】

【0007】

したがって、すべての中間段階を含む複製プロセスのすべての段階でiノード・テーブル・エントリの順序を保持する、ファイル・システムを複製するための方法が求められている。このソリューションは、複製プロセス中に割り込みが発生した場合、ターゲット・ファイル・システムの残余部が復活可能である。

【課題を解決するための手段】

【0008】

本発明は、ネットワークを横切ってファイル・システムを複製するため、および、複製プロセス全体の間、ファイル・システム内で関連付けられたテーブルの順序を保持するための、方法およびシステムを有する。

【0009】

本発明の一態様では、ファイル・システムを複製するための方法が提供される。データ・構造として編成されたソース・ファイル・システムは、ツリー順にスキャンされる。ソース・ファイル・システムの各ディレクトリおよびディレクトリ・インデックスが識別される。同様に、ソース・ファイル・システムからのディレクトリおよびディレクトリ・インデックスは、ターゲット・ファイル・システム内でも作成される。ターゲット・ファイル・システム内でのインデックスおよびディレクトリの作成に続き、ソース・ファイル・システム内のディレクトリからのコンテンツが、ソース・ファイル・システム内で保持されているものと同じ識別インデックスと共に、ターゲット・ファイル・システム内の対応するディレクトリにコピーされる。ディレクトリ・コンテンツのコピー・プロセスは、再帰的に実行される。

【0010】

本発明の他の態様では、プロセッサがメモリに結合され、操作を実行するように動作可能な、プロセッサおよびメモリを備えたコンピュータ・システムが提供される。こうしたプロセッサ操作は、データ構造として編成されたファイル・システムをツリー順にスキャンすること、スキャン済みファイル・システム内のディレクトリおよびディレクトリ・インデックスを識別すること、異なるファイル・システム内で、ディレクトリ・インデック

スを含むスキャン済みファイル・システムからディレクトリを作成すること、ならびに、スキャン済みファイル・システム内のディレクトリから、異なるファイル・システム内の対応するリモート・ディレクトリへと、コンテンツを再帰的にコピーすること、を含む。作成されたディレクトリは、当該スキャン済みディレクトリからリモートの異なるファイル・システム内にある。

【 0 0 1 1 】

本発明の他の態様では、コンピュータによって読み取り可能なプログラム・ストレージ・デバイスが提供される。このコンピュータは、ファイル・システムを複製するための操作を実行するために、コンピュータによって実行可能な命令を確実に記録する。この操作は、データ構造として編成されたソース・ファイル・システムをツリー順にスキャンすることを含む。ソース・ファイル・システム内のディレクトリおよびディレクトリ・インデックスが識別され、ソース・ファイル・システムからのディレクトリおよびディレクトリ・インデックスがターゲット・ファイル・システム内で作成される。ソース・ファイル・システム内のディレクトリからのコンテンツが、ターゲット・ファイル・システム内の対応するディレクトリに再帰的にコピーされる。ターゲット・ファイル・システム内のディレクトリ・インデックスは、ソース・ファイル・システム内のディレクトリ・インデックスと同じである。

10

【 0 0 1 2 】

本発明の他の態様では、コンピュータによって読み取り可能なプログラム・ストレージ・デバイスが提供される。このコンピュータは、ファイル・システムを複製するための操作を実行するためにコンピュータによって実行可能な命令を確実に記録する。この命令は、データ構造として編成されたソース・ファイル・システムをノード順にスキャンすることを含む。スキャン中、見つけられた各ディレクトリのコンテンツが銘記 (remember) され、ターゲット・ファイル・システムへ報告されるため、結果としてディレクトリに関する数値識別子が、ディレクトリ内の各オブジェクトに関する識別子に先行する。

20

【 0 0 1 3 】

本発明の他の特徴および利点は、添付の図面に関連した、現時点で好ましい本発明の実施形態についての以下の詳細な説明から明らかとなる。

【 0 0 1 4 】

次に、本発明について、添付の図面を参照しながら単なる例として説明する。

30

【 発明を実施するための最良の形態 】**【 0 0 1 5 】**

2つのサーバがネットワークを横切って相互接続される。第1のサーバは1つのファイル・システムと通信し、第2のサーバは第2のファイル・システムと通信する。各ファイル・システムは、階層様式でデータ構造に編成される。ファイル・システムは、データ構造内のテーブルの順序を保持するように、一方のサーバから他方のサーバへとネットワークを横切って複製することができる。

【 0 0 1 6 】

本発明の以下の説明は、本発明のシステムおよび方法を実施するための現時点で好ましい実施形態の構造または処理のいずれかを説明するために、流れ図を使用することによって提示される。本発明を提示するためにこのようにして図を使用することは、本発明の範囲を限定するものと解釈されるべきではない。

40

【 0 0 1 7 】

本明細書では、ネットワークを横切ってファイル・システムを複製するための方法が開示される。この方法には、2つのセグメントを有する第1の部分と第2の部分がある。図4は、ディレクトリによって指示されたオブジェクトを報告する前にツリー内のディレクトリを報告する、ツリー順に編成されたソース・ファイル・システムをスキャンする、第1の複製方法の第1の部分の第1のセグメントを示す、流れ図 (200) である。ルート・ディレクトリが変数 NAME に割り当てられ (202)、複製プロセスに使用されるビットマップは消去される (204)。名前インデックスをコピーするための機能

50

N I S C A N が呼び出される (2 0 6) 。この機能は、1つのパラメータ、N A M E を有する。図 5 は、第 1 の複製方法の第 1 の部分の第 2 のセグメントを示す流れ図 (2 2 0) である。このセグメントでは、N I S C A N 機能が詳細に示される。ビットマップで使用するためのビットが、ソース・ファイル・システム内の変数 N A M E に設定される (2 2 2) 。その後、第 1 のテストで、コピー機能に対する引数 N A M E がソース・ファイル・システム内のディレクトリであるかどうかを判別する (2 2 4) 。ステップ (2 2 4) で、このテストに対する応答が肯定であれば、結果として、ソース・ファイル・システム内にあるのと同じインデックスで、ターゲット・ファイル・システム内に空のディレクトリ N A M E が作成される (2 2 6) 。ディレクトリをコピーするための機能 N I S C A N は、ディレクトリ N A M E 内のすべての要素 E をコピーするために再帰的に呼び出され、すなわち、N I S C A N (N A M E / E) である (2 2 8) 。N I S C A N の各呼び出しは、各ディレクトリのサブファイルを含む 1 つのファイルについてビットを設定する。ビットマップ内のビットの設定は、特定の i ノード、すなわち特定のファイルあるいはディレクトリまたはその両方がターゲット・ファイル・システム内で複製されたことの、インジケータとしての働きをする。ステップ (2 2 4) で、テストに対する応答が否定であれば、ファイル、すなわち N A M E は、ソース・ファイル・システム内にあるのと同じ、ターゲット・ファイル・システム内のインデックスで、ソース・ファイル・システムからターゲット・ファイル・システムへとコピーされる。ソース・ファイル・システムからのディレクトリのすべてのコンテンツがターゲット・ファイル・システムにコピーされると、名前インデックス・コピー機能は完了する。万一複製が完了する前に割り込みが発生しても、複製された部分のあらゆるファイルおよびディレクトリはある名前で作成されているため、複製されたターゲット・ファイル・システム・ディレクトリの整合性は損なわれないままである。

10

20

30

40

50

【 0 0 1 8 】

ターゲット・ファイル・システム内のディレクトリおよびそのエントリの複製に続いて、ディレクトリ・エントリを持たないソース・ファイル・システム内の任意のオブジェクトを報告するために、第 2 の機能が呼び出される。図 5 は、この第 2 の機能を示す流れ図 (2 5 0) である。インデックス変数が 1 に設定される (2 5 2) 。その後、インデックス変数がソース i ノード・テーブルよりも大きいかどうかを判別するためのテストが実行される (2 5 4) 。ステップ (2 5 4) で、テストに対する応答が肯定の場合、インデックスがソース i ノード・テーブルよりも大きく、ソース・ファイル・システム内のすべての i ノードがスキャンされたことを示しているため、ソース i ノード・テーブルに関するコピー・プロセスは完了する (2 5 6) 。しかしながら、ステップ (2 5 4) で、テストに対する応答が否定の場合、ソース・ファイル・システムの i ノード・テーブルのインデックスにファイルがあるかどうかを判別するためのテストが実行される (2 5 8) 。ステップ (2 5 8) で、テストに対する応答が肯定の場合は、インデックス内のファイルをターゲット・ファイル・システム内で複製する必要がある可能性があることを示している。しかしながら、ステップ (2 5 8) で、テストに対する応答が否定の場合は、インデックス変数が増分される (2 6 0) 。その後、プロセスはステップ (2 5 4) に戻る。ステップ (2 5 8) で、テストに対する応答が肯定の場合、i ノード・テーブル内の設定インデックスのファイルが、ターゲット・ファイル・システムで複製されたかどうかを判別するために次のテストが実行される (2 6 2) 。一実施形態では、ビットマップ内にビットが設定されることになり、このビットマップは、複製されていればこのファイルに対応するソース・ファイル・システム内で最初からずっと一時構造である。ステップ (2 6 2) で、テストに対する応答が肯定の場合は、ステップ (2 6 0) に戻り、インデックス変数が増分される。同様に、ステップ (2 6 2) で、テストに対する応答が否定の場合は、そのインデックスで、ソース・ファイル・システムからターゲット・ファイル・システムへ i ノード・テーブル内のインデックスのファイルが複製されることになり、その後、インデックス変数の増分のためにステップ (2 6 0) に戻る。スイープ (sweep) ・プロセスは、テーブルの終わりに達したことを示す、ステップ (2 5 4) でのテストの肯定応答が戻

されるまで続行される。したがって本明細書に示されるように、オブジェクトは、ソース・ファイル・システムと同じノード・インデックスで、ソース・ファイル・システムからターゲット・ファイル・システムへと複製される。

【0019】

図4、図5、および図6で上述されたプロセスは、コピーされる要素を保持しながら、ツリー順でソース・ファイル・システムをスキャンすることによって、ファイル・システムを複製するための再帰的方法である。他の実施形態では、再帰的関数を呼び出さずに、コピーされる要素のiノード・インデックスを保持しながら、ソース・ファイル・システムからのテーブルをターゲット・ファイル・システムに複製することができる。この複製プロセスに含まれる3つの主なルーチンがある。図7は、ノード順にファイル・システムを複製するための、単一パスの非再帰的方法における第1のルーチンを示す流れ図(300)である。第1のルーチンの開始に先立ち、ターゲット・ファイル・システム内にマッピングの空テーブルが作成される(302)。この空テーブルは一時構造である。ステップ(302)に続いて、インデックス変数が1に設定される(304)。その後、インデックス変数が、スキャンされるソース・テーブルの終わりを超えているかどうかを判断するために、テストが実行される(306)。一実施形態では、ターゲット・ファイル・システム内で複製されるテーブルは、ソース・ファイル・システムのiノード・テーブルである。ステップ(306)で、テストに対する応答が肯定の場合、第1のルーチンは完了し、ステップ(402)で第2のルーチンが開始される。しかしながら、ステップ(306)で、テストに対する応答が否定の場合、複製されるソース・テーブル内の設定インデックスにオブジェクトがあるかどうかを判断するために、テストが実行される(308)。ステップ(308)で、テストに対する応答が肯定の場合、続いて、ソース・ファイル・システム内のオブジェクトがディレクトリであるかどうかを判断するためのテストが実行されることになる(310)。ステップ(310)で、テストに対する応答が否定の場合、ターゲット・テーブルに空白のエントリが追加され、これによって、その後の時点で使用されることになるスペース用にテーブル・エントリが作成される(312)。同様に、ステップ(310)で、テストに対する応答が肯定の場合、ソース・ファイル・システム・テーブル・ディレクトリのディレクトリ内の各エントリに関するデリミッタおよびノード番号と共に、一時ターゲット・テーブルにエントリが追加されることになる(314)。ステップ(312)または(314)のいずれかに続いて、あるいは、ステップ(308)でのテストに対する否定応答に続いて、インデックス変数が増分され(316)、その後ステップ(306)に戻る。ステップ(306)から(316)に示されたルーチンは、ソース・ファイル・システム・テーブルをスキャンするため、および一時複製済みテーブルを作成するための、プロセスを示す。

【0020】

図8は、図2に示されたソース・ファイル・システムのサンプルiノード・テーブルに基づく、ターゲット・ファイル・システム内の一時テーブルを示すブロック図(350)である。図に示されるように、4つのエントリがある。各エントリが、ソースiノード・テーブル内のインデックス1、4、6、および7を表し、各エントリは、それぞれが「;」文字で表される2つのデリミッタを有する。図に示されるように、インデックス2、3、および5は空のインデックスであるため、これらを表す場所にはエントリがない。

【0021】

一時テーブルの各エントリは、第1のデリミッタの前のオプション整数X、2つのデリミッタの間のオプション文字列Y、および第2のデリミッタの後のオプションのpair listという、オプションの値を備えた、いくつかのフィールドを含む。pair listとは、tixと呼ばれるそれぞれの整数の後に名前が続く、ペアのシーケンスである。

【0022】

ソース・ファイル・システム・テーブルのスキャンおよび複製プロセスの完了に続き、図7で作成された一時テーブルをソートするための第2のルーチンが開始される。一実施

10

20

30

40

50

形態では、ソート・ルーチンには2つの別々のアルゴリズムが含まれる。図9は、図7で作成された一時テーブルをソートするための第1のアルゴリズムを示す流れ図(400)である。初期ステップで、一時テーブルへのインデックスPに、整数1が割り当てられる(402)。その後、一時テーブルへのインデックスが一時ターゲット・テーブルのサイズよりも大きいかどうかを判別するために、テストが実行される(404)。ステップ(404)で、テストに対する応答が肯定である場合、一時テーブルの第1のアルゴリズムは完了し、図11の第2のアルゴリズムの開始へと進むことになる。しかしながら、ステップ(404)で、テストに対する応答が否定である場合、たとえば($i \times a$; ; $pairlist$)のように、変数Pに関する一時テーブル・エントリからフィールドを引き出し(406)、このインデックスを一時テーブルに割り当て(408)、フィールドから抽出されたオブジェクトのリストに $pairlist$ を割り当てる(410)。リスト $pairlist$ 内の各要素について、要素 $i \times$ に関するテーブル・エントリ内の $i \times$ に割り当てられた値が、その要素の一時テーブル・フィールドに関する一時テーブル・エントリに追加され、 tix となる(412)。すなわち、 tix フィールド値は更新するエントリを選択するために使用され、 $i \times$ 値はエントリのXフィールドに挿入される。同様に、 $pairlist$ 内のその tix 値とペアになっている名前が、修正されるエントリのYフィールドに挿入される。ステップ(412)に続いて、変数P、すなわち一時テーブルへのインデックスが増分され(414)、ルーチンはステップ(404)に戻る。したがって、ステップ(402)~(414)に示された第1のソート・ルーチンは、ステップ(404)でのテストに関して肯定応答が受け取られるまで続行される。

【0023】

図10は、図8に示された一時テーブルに基づく複製プロセスの第2の段階での一時テーブルを示す、図7で作成された一時テーブルをソートするための第1のアルゴリズムの完了に続くブロック図(450)である。図に示されるように、4つのエントリがある。それぞれのエントリは、ソースiノード・テーブル内の1、4、6、および7のインデックスを表し、4、6、および7に関連付けられたエントリは、図9に示されたアルゴリズムの完了に従って修正されている。

【0024】

図10は、図9のステップ(404)でのテストに対する肯定応答に続いて開始される、一時テーブルをソートするための第2のアルゴリズムを示す流れ図(500)である一時テーブルへのインデックスを表す変数Pに、整数1が割り当てられる(502)。その後、Pが一時テーブルのサイズよりも大きいかどうかを判別するためにテストが実行される(504)。ステップ(504)で、テストに対する応答が肯定の場合、一時テーブルをソートするための第2のアルゴリズムは完了することになる。しかしながら、ステップ(504)で、テストに対する応答が否定の場合、 $i \times$ および $pairlist$ を変数Pのテーブル・エントリに割り当てることによって、一時テーブルが行ごとに更新されることになる(506)。ステップ(506)での割り当てに続き、ステップ(506)から割り当てられたリストである、 $pairlist$ が空であるかどうかを判別するためにテストが実行される(508)。ステップ(508)で、テストに対する応答が肯定である場合、ステップ(510)で変数Pが増分され、ステップ(504)に戻る。しかしながら、ステップ(508)で、テストに対する応答が否定である場合、抽出されたリスト、 $pairlist$ 内のすべての要素の一時テーブル・インデックス・フィールド(tix)内の最小値が計算され、この最小値が変数Bに割り当てられる(512)。その後、変数Bが一時ターゲット・テーブルへのインデックス $i \times$ よりも小さいかどうかを判別するためのテストが実行される(514)。ステップ(514)で、テストに対する応答が肯定である場合、変数Pが増分され、ステップ(504)に戻る。同様に、ステップ(514)で、テストに対する応答が否定である場合、続いて、一時テーブルから $i \times$ に関するテーブル・エントリが除去され、抽出されたリスト、 $pairlist$ のエントリ内のそれぞれの tix 変数に関連付けられたすべてのエントリの前に、 $i \times$ に関するテーブル・エントリが挿入される(518)。このステップ(516)および(518)

での除去および挿入は、それぞれ、一時テーブル内での順序付けの変更、すなわち、データ構造の更新をサポートする。ステップ(518)での挿入に続いて、プロセスは変数Pの増分のためにステップ(510)に戻る。したがって、一時テーブルをソートするための第2のアルゴリズムは、一時テーブルの再順序付けを実行する。

【0025】

図12は、図10に示されたソース・ファイル・システムのサンプル一時テーブルに基づく一時テーブルを示し、図7で作成された一時テーブルをソートするための第2のアルゴリズムの完了に基づくテーブルの再順序付けに従った、ブロック図(550)である。図に示されるように、4つのエントリがある。それぞれのエントリは、ソースiノード・テーブル内の1、4、6、および7のインデックスを表す。図に示されるように、一時テーブル内のインデックスの順序付けは、図11に示されたアルゴリズムの実行完了に従って修正されている。

10

【0026】

図11に示されたような第2ソート・ルーチンの完了に続いて、ソース・ファイル・システムからテーブルを複製するための最終ルーチンが開始される。図13および図14は、ソース・ファイル・システムからターゲット・ファイル・システムへテーブルを複製するための最終ルーチンを示す流れ図(600)である。一時テーブルへのインデックスPに整数1が割り当てられる(602)。その後、一時テーブルへのインデックスPが、図11に示された第2のソート・アルゴリズムで再編成された一時テーブルのサイズよりも大きいかどうかを判別するためのテストが実行される(604)。ステップ(604)で、テストに対する応答が肯定である場合、ファイル・システムの複製が完了することになる(606)。しかしながら、ステップ(604)で、テストに対する応答が否定である場合、たとえば($i x$ a X ; Y ; pair list)のように、変数Pに関する一時テーブル・エントリからフィールドを引き出し(608)、変数*ix*と共に一時テーブルにインデックスを割り当て(610)、第1のデリミッタの前の整数を変数Xに割り当て(612)、デリミッタ間の文字列を変数Yに割り当て(614)、フィールドから抽出されたオブジェクトのリストにpair listを割り当てる(616)。ステップ(616)での割り当てに続いて、変数*ix*に対応するソース・インデックス内のオブジェクトがディレクトリであるかどうかを判別するためのテストが実行される(618)。ステップ(618)で、テストに対する応答が否定である場合、一時テーブル内のインデックス*ix*でオブジェクトが作成され(620)、インデックス*ix*のディレクトリがあれば、そこにオブジェクトが作成され、これに名前Yが与えられ(622)、ソース・インデックス*ix*からターゲット・インデックス*ix*へとファイル・コンテンツがコピーされることになる(624)。同様に、ステップ(618)で、テストに対する応答が肯定である場合、ターゲット・ファイル・システム内のインデックス*ix*で、またはインデックスXに配置されたディレクトリのサブディレクトリとして、空のディレクトリが作成されるか、一時ターゲット・テーブルがインデックスXを有する場合、名前Yでサブディレクトリが作成されることになる(626)。一時テーブルが変数Xに関するエントリを有する場合、インデックスXのディレクトリのサブディレクトリとして、名前Yで空のディレクトリが作成される。ステップ(624)および(626)の完了に続いて、変数Pが増分され(628)、プロセスはステップ(604)に戻る。したがって、コピー・ルーチンが完了すると、ターゲット・ファイル・システム内のソース・テーブルが複写される。

20

30

40

【0027】

本発明は、階層構造データの複製に関する方法およびシステムを企図する。本発明は、完全なハードウェア実施形態、完全なソフトウェア実施形態、またはハードウェア要素およびソフトウェア要素の両方を含む実施形態の形を取ることができる。図15は、ソース・ファイル・システム内のディレクトリからターゲット・ファイル・システム内の対応するディレクトリへ、すべてのコンテンツをコピーするための、再帰的プロセスを呼び出すように構成された、ハードウェア環境におけるマネージャを示すブロック図(700)で

50

ある。図に示されるように、ソース・ファイル・サーバ(720)は、プロセッサ(726)、メモリ(728)、およびネットワーク・アダプタ(732)を含む。同様に、ターゲット・ファイル・サーバ(740)は、プロセッサ(746)、メモリ(748)、およびネットワーク・アダプタ(752)を含む。ソースおよびターゲットのファイル・サーバ(720)および(740)は、それぞれ、ローカルまたはワイドのエリア・ネットワークを備えることが可能なネットワーク接続(722)および(742)を通じたネットワーク(730)を介してリンクされる。ソース・ファイル・サーバは、マネージャ(760)も含む。テーブル・ロケータ(762)、配置されたソース・ファイル・システムのテーブルの各ディレクトリをターゲット・ファイル・システム内に作成するためのツール(764)、および、ソース・テーブルのディレクトリ内のコンテンツをターゲット・ファイル・システム内の対応するディレクトリに再帰的にコピーするためのディレクタ(766)も含む。一実施形態では、そのテーブル・ロケータ(762)、ツール(764)、およびディレクタ(766)を備えたマネージャ(760)を、ターゲット・ファイル・サーバ(740)内、またはプロセッサ、メモリ、およびネットワーク・アダプタを備えた代替処理ユニット内で、具体化することができる。

10

20

30

40

50

【0028】

好ましい実施形態では、本発明は、ファームウェア、常駐ソフトウェア、マイクロコードなどを含むがこれらに限定されることのない、ソフトウェアに実装される。ソフトウェア要素に関して、ソース・ファイル・システムおよびターゲット・ファイル・システムの両方が、それぞれ、それぞれのファイル・システム内のファイル・システム・サーバのメモリ内に常駐するマネージャを有することができる。ソース・ファイル・システム・マネージャは、上記で概説および考察したアルゴリズムを呼び出すための、命令あるいはプログラム・コードまたはその両方を含むことができる。同様に、ハードウェア環境では、ソースおよびターゲットのファイル・システム・マネージャは、それぞれのファイル・システム内のファイル・システム・サーバのメモリ外部に常駐することができる。

【0029】

さらに本発明は、コンピュータまたは任意の命令実行システムによって、あるいはそれらに関連して、使用するためのプログラム・コードを提供する、コンピュータ使用可能またはコンピュータ読み取り可能な媒体からアクセスできる、コンピュータ・プログラム製品の形を取ることができる。この説明の目的で、コンピュータ使用可能またはコンピュータ読み取り可能な媒体は、命令実行のシステム、装置、またはデバイスによって、あるいはそれらに関連して、使用するためのプログラムを含む、格納する、通信する、伝播する、または移送することができる、任意の装置とすることができる。

【0030】

本発明の範囲内の諸実施形態は、内部に符号化されたプログラム・コードを有するプログラム・ストレージ手段を備える製品も含む。こうしたプログラム・ストレージ手段は、汎用または特定用途向けのコンピュータによってアクセス可能な、任意の使用可能媒体とすることができる。例を挙げると、こうしたプログラム・ストレージ手段は、RAM、ROM、EEPROM、CD-ROM、または他の光ディスク・ストレージ、磁気ディスク・ストレージ、または他の磁気ストレージ・デバイス、あるいは、所望なプログラム・コード手段を格納するために使用可能であり、汎用または特定用途向けのコンピュータによってアクセス可能な、任意の他の媒体を含むことができるが、これらに限定されるものではない。前述の組み合わせも、プログラム・ストレージ手段の範囲に含まれるものとする。

【0031】

媒体は、電子、磁気、光、電磁、赤外線、または半導体のシステム(あるいは装置またはデバイス)、あるいは伝播媒体とすることができる。コンピュータ読み取り可能媒体の例には、半導体またはソリッド・ステート・メモリ、磁気テープ、取り外し可能コンピュータ・ディスク、ランダム・アクセス・メモリ(RAM)、読み取り専用メモリ(ROM)、剛性磁気ディスク、および光ディスクが含まれる。光ディスクの現行の例には、

コンパクト・ディスクB読み取り専用(CD-ROM)、コンパクト・ディスクB読み取り/書き込み(CD-R/W)、およびDVDが含まれる。

【0032】

プログラム・コードの格納あるいは実行またはその両方に好適なデータ処理システムには、システム・バスを介してメモリ要素に直接または間接的に結合された、少なくとも1つのプロセッサが含まれることになる。メモリ要素は、プログラム・コードの実際の実行中に使用されるローカル・メモリ、大容量ストレージ、および、実行中に大容量ストレージからコードを取り出さなければならない回数を減らすために少なくともいくつかのプログラム・コードの一時ストレージを提供するキャッシュ・メモリを、含むことができる。

【0033】

入力/出力またはI/Oデバイス(キーボード、ディスプレイ、ポインティング・デバイスなどを含むが、これらに限定されない)を、直接、またはI/Oコントローラの介入を通じて、システムに結合することができる。

【0034】

ネットワーク・アダプタをシステムに結合して、専用または公衆のネットワークの介入を通じて、他のデータ処理システム、あるいはリモートのプリンタまたはストレージ・デバイスに、データ処理システムを結合できるようにすることもできる。モデム、無線、およびイーサネット(登録商標)のアダプタは、現在使用可能なネットワーク・アダプタ・タイプの一部に過ぎない。

【0035】

以上、本明細書では、本発明の特定の実施形態について例示の目的で説明してきたが、本発明の範囲を逸脱することなく様々な修正が実行可能であることを理解されよう。とりわけ、本発明は、ソース・ファイル・システムからターゲット・ファイル・システムへのiノード・テーブルの複製に限定されるべきではない。むしろ、本明細書に示されたアルゴリズムは、任意のデータ構造および関連するテーブルの複製を含むように拡張することができる。

【図面の簡単な説明】

【0036】

【図1】従来技術のコンピュータ・システムを示すブロック図である。

【図2】ソース・ファイル・システムからの従来技術のiノード・テーブルを示すブロック図である。

【図3】図2のソース・ファイル・システムから複製された場合の、ターゲット・ファイル・システムでの従来技術のiノード・テーブルを示すブロック図である。

【図4】本発明の好ましい実施形態に従った、ネットワークを横切ってファイル・システムを複製するための第1の方法の第1の部分を示す流れ図である。

【図5】本発明の好ましい実施形態に従った、ネットワークを横切ってファイル・システムを複製するための第1の方法の第1の部分を示す流れ図である。

【図6】ネットワークを横切ってファイル・システムを複製するための第1の方法の第2の部分を示す流れ図である。

【図7】ネットワークを横切ってファイル・システムを複製するための第2の方法の第1の部分を示す流れ図である。

【図8】複製プロセスにおける第1の段階での一時iノード・テーブルを示すブロック図である。

【図9】ネットワークを横切ってファイル・システムを複製するための第2の方法の第2の部分を示す流れ図である。

【図10】複製プロセスにおける第2の段階での一時iノード・テーブルを示すブロック図である。

【図11】ネットワークを横切ってファイル・システムを複製するための第2の方法の第3の部分を示す流れ図である。

【図12】複製プロセスにおける第3の段階での一時iノード・テーブルを示すブロック

10

20

30

40

50

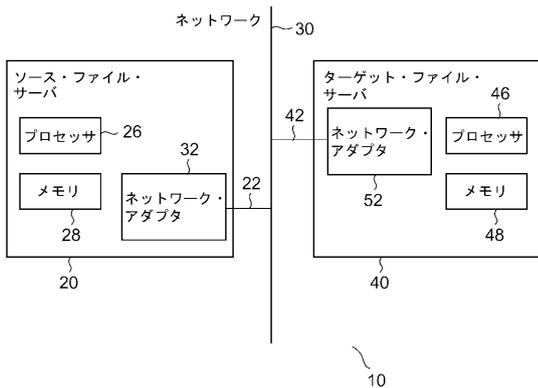
図である。

【図13】ネットワークを横切ってファイル・システムを複製するための第2の方法の第4の部分を示す流れ図である。

【図14】ネットワークを横切ってファイル・システムを複製するための第2の方法の第4の部分を示す流れ図である。

【図15】本発明の一実施形態に従ったマネージャの配置を示すコンピュータ・システムを示すブロック図である。

【図1】



【図3】

インデックス	構造内の名前
1	(ルート・ディレクトリ) /
2	/ Readme
3	/ bin
4	/ bin / edit

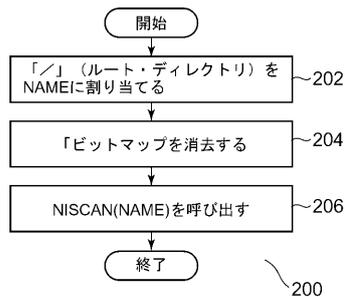
150

【図2】

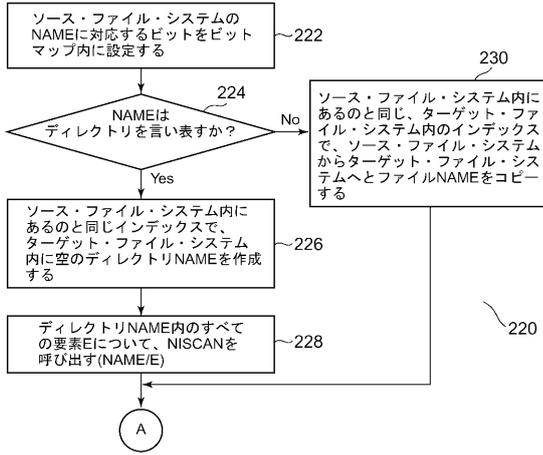
1	ディレクトリ: コンテンツ (("bin",6),("README",7))	102
2	空	110
3	空	112
4	ファイル: 「編集」プログラム用の 実行可能イメージ	104
5	空	114
6	ディレクトリ: コンテンツ (("edit",4)	106
7	ファイル: 読み取り可能テキスト・ ファイル	108

100

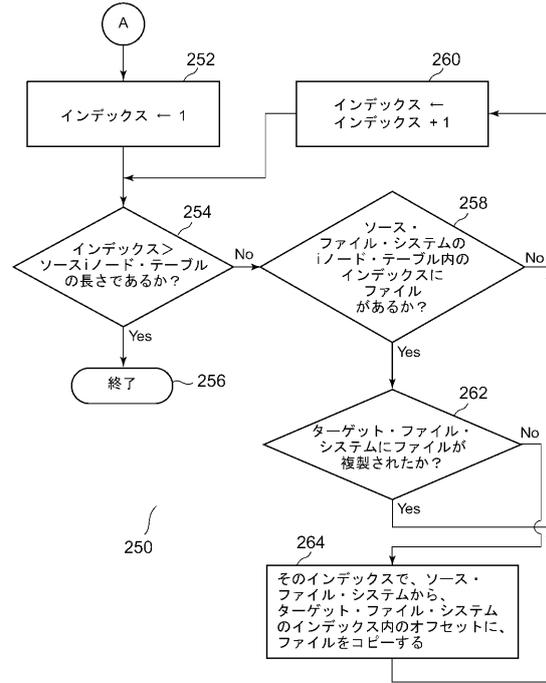
【図4】



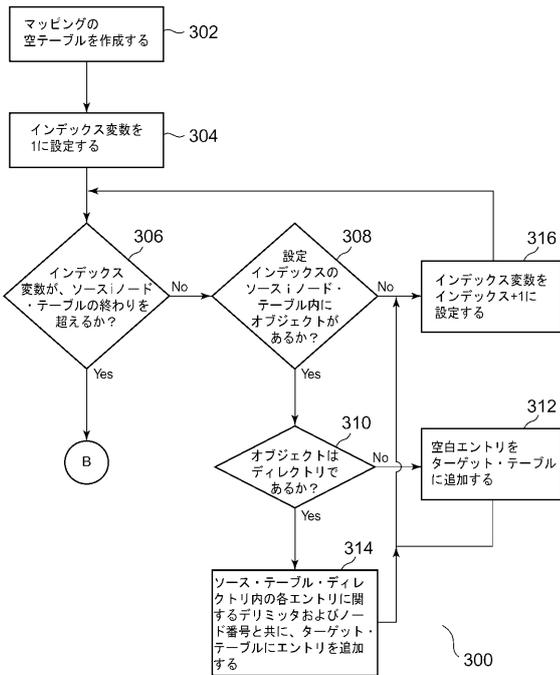
【 図 5 】



【 図 6 】



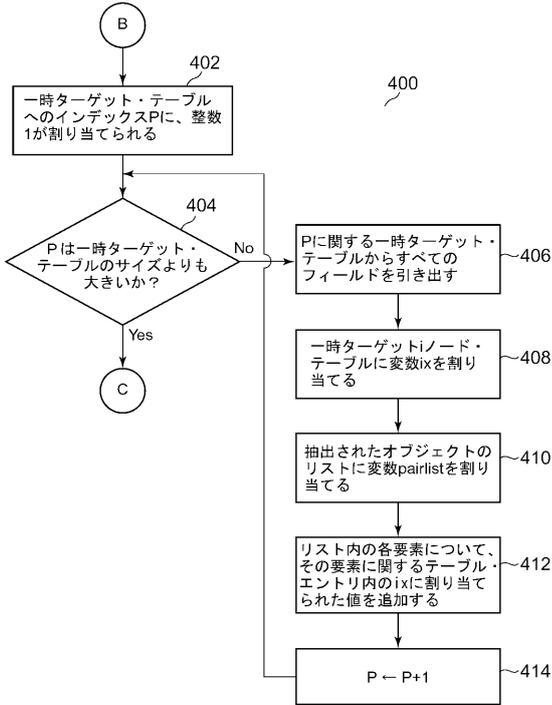
【 図 7 】



【 図 8 】

1	:: (6,"bin"),(7,"README")
4	::
6	:: (4,"edit")
7	::

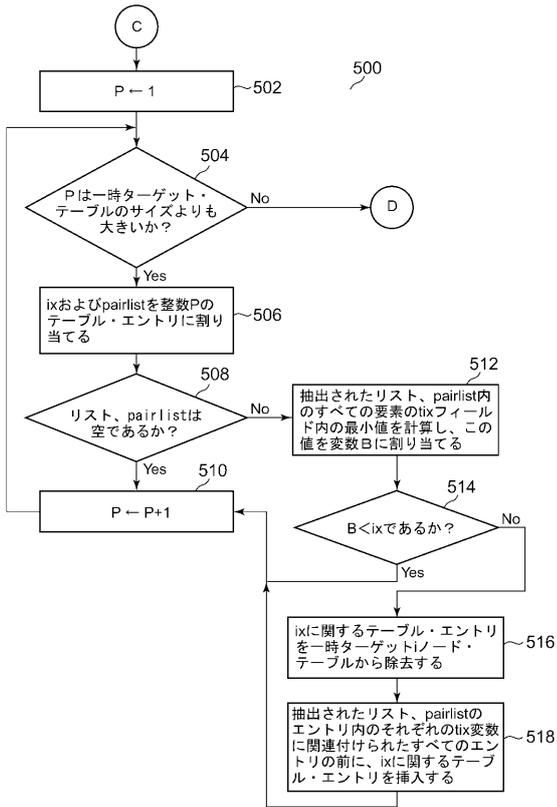
【 図 9 】



【 図 1 0 】

1	:: (6,"bin"),(7,"README")
4	6;"edit";
6	1;"bin";(4;"edit")
7	1;"README";

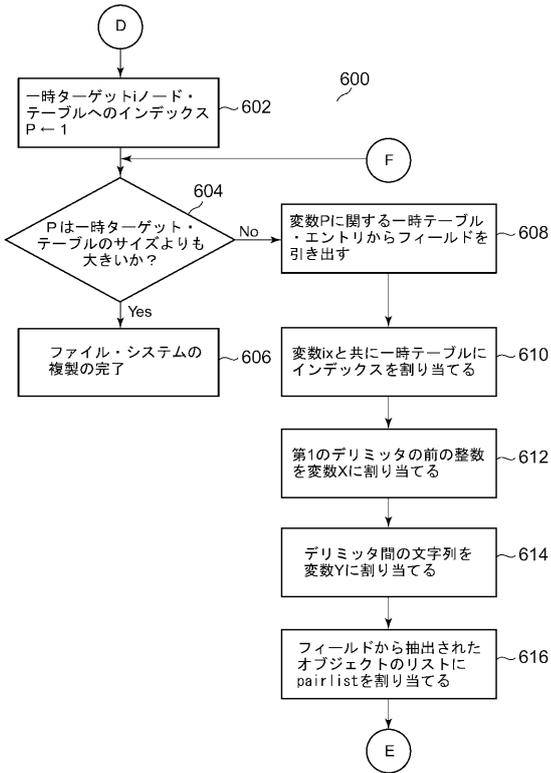
【 図 1 1 】



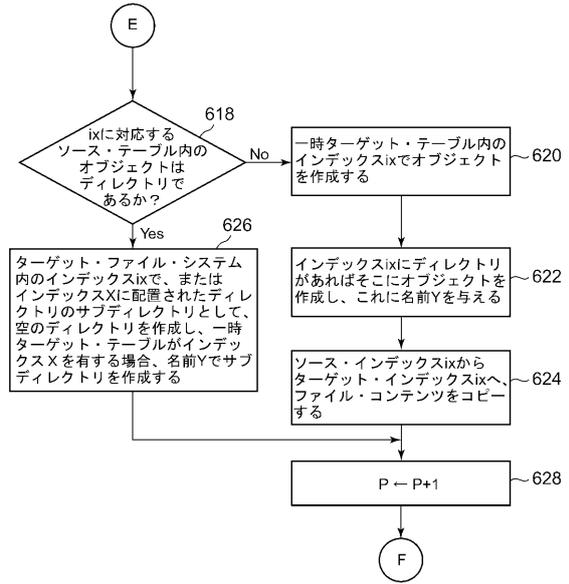
【 図 1 2 】

1	:: (6,"bin"),(7,"README")
6	1;"bin";(4;"edit");
4	6;"edit";
7	1;"README";

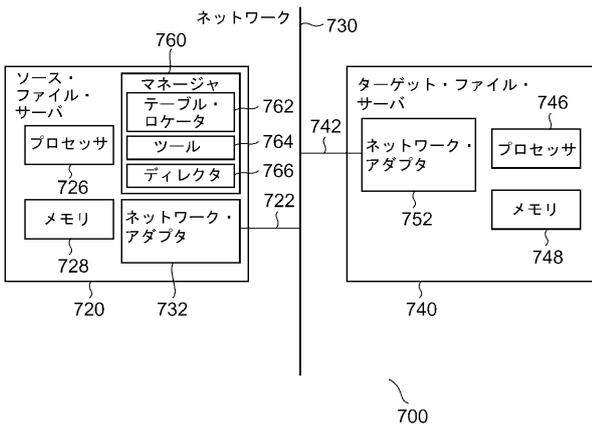
【 図 1 3 】



【 図 1 4 】



【 図 1 5 】



【手続補正書】

【提出日】平成21年3月18日(2009.3.18)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

データ構造として編成されたソース・ファイル・システムをツリー順にスキャンするステップと、

前記ソース・ファイル・システム内のディレクトリおよびディレクトリ・インデックスを識別するステップと、

ターゲット・ファイル・システム内でディレクトリ・インデックスを含む前記ソース・ファイル・システムからディレクトリを作成するステップと、

前記ソース・ファイル・システム内のディレクトリから、前記ターゲット・ファイル・システム内の対応するディレクトリへと、コンテンツを再帰的にコピーするステップと、を含み、前記ターゲット・ファイル・システム内の前記ディレクトリ・インデックスは前記ソース・ファイル・システム内の前記ディレクトリ・インデックスと同じである、ファイル・システムを複製するための方法。

【請求項2】

ディレクトリ・コンテンツの複製に続いて、前記ソース・ファイル・システムのテーブル内にビットを設定するステップをさらに含む、請求項1に記載の方法。

【請求項3】

ディレクトリ・エントリなしのオブジェクトに関する前記ソース・ファイル・システム内のノード・テーブルを検索するステップをさらに含む、請求項2に記載の方法。

【請求項4】

前記ディレクトリ・エントリなしの前記オブジェクトを、前記ソース・ファイル・システムから、前記識別インデックスで、前記ターゲット・ファイル・システムにコピーするステップをさらに含む、請求項3に記載の方法。

【請求項5】

前記ソース・ファイル・システム内のディレクトリからコンテンツを再帰的にコピーするステップが、前記ソース・ファイル・システム内の各オブジェクトに関する前記識別インデックスを、前記ターゲット・ファイル・システム内の前記識別インデックスでの各オブジェクトの属性として、前記ターゲット・ファイル・システムに伝送するステップを含む、請求項1に記載の方法。

【請求項6】

プロセッサと、
メモリと、

を備えるコンピュータ・システムであって、

前記プロセッサは前記メモリに結合され、前記プロセッサは、

データ構造として編成されたファイル・システムをツリー順にスキャンする操作と、

前記スキャン済みファイル・システム内のディレクトリおよびディレクトリ・インデックスを識別する操作と、

ディレクトリ・インデックスを含む前記スキャン済みファイル・システムから、ディレクトリを作成する操作であって、前記作成されたディレクトリが前記スキャン済みディレクトリからリモートの異なるファイル・システム内にある、作成する操作と、

前記スキャン済みファイル・システム内のディレクトリから、前記異なるファイル・システム内の対応するリモート・ディレクトリへと、コンテンツを再帰的にコピーする操作と、を含む、操作を実行するように動作可能である、コンピュータ・システム。

【請求項 7】

コンテンツの複製完了後に、前記スキャン済みファイル・システム内の前記配置されたテーブル内に設定されるように適合されたビットをさらに備える、請求項 6 に記載のシステム。

【請求項 8】

ディレクトリ・エントリなしに前記スキャン済みファイル・システム内にオブジェクトを配置するように適合されたオブジェクト・レプリケータをさらに備える、請求項 7 に記載のシステム。

【請求項 9】

前記再帰的コピー操作が、前記オブジェクトが再作成されている場合、前記スキャン済みファイル・システム内の各オブジェクトに関する識別番号を、その識別番号での各オブジェクトの属性として伝送する、請求項 7 に記載のシステム。

【請求項 10】

ファイル・システムを複製するための操作を実行するためにコンピュータに、データ構造として編成されたソース・ファイル・システムをツリー順にスキャンする操作と、

前記ソース・ファイル・システム内のディレクトリおよびディレクトリ・インデックスを識別する操作と、

ターゲット・ファイル・システム内でディレクトリ・インデックスを含む前記ソース・ファイル・システムからディレクトリを作成する操作と、

前記ソース・ファイル・システム内のディレクトリから、前記ターゲット・ファイル・システム内の対応するディレクトリへと、コンテンツを再帰的にコピーする操作であって、

前記ターゲット・ファイル・システム内の前記ディレクトリ・インデックスは前記ソース・ファイル・システム内の前記ディレクトリ・インデックスと同じである、前記コンテンツを再帰的にコピーする操作と、

を実行させるプログラム。

【請求項 11】

ファイル・システムを複製するための操作を実行するためにコンピュータに、データ構造として編成されたソース・ファイル・システムをノード順にスキャンする操作と、

前記スキャンで見つめられた各ディレクトリのコンテンツを銘記する操作と、

ディレクトリに関する数値識別子が、前記ディレクトリ内の各オブジェクトに関する識別子に先行するように、ターゲット・ファイル・システムへノードを報告する操作と、

を実行させるプログラム。

【請求項 12】

ディレクトリおよびファイルに関する前記識別子を、オブジェクトが複製されている間に各オブジェクトの属性として、前記ソース・ファイル・システムから前記ターゲット・ファイル・システムへと伝送するための操作をさらに含む、請求項 11 に記載のプログラム。

【請求項 13】

前記ターゲット・ファイル・システムに関するディレクトリを、それらディレクトリのコンテンツを作成する前に作成するための操作をさらに含む、請求項 11 に記載のプログラム。

【請求項 14】

前記ソース・ファイル・システムをノード順にスキャンするための、非再帰的ルーチンである操作をさらに含む、請求項 11 に記載のプログラム。

【請求項 15】

前記ソース・ファイル・システム内の複製されたデータを、前記ターゲット・ファイル・システムに報告されている間に再順序付けするための操作をさらに含む、請求項 11 に

記載のプログラム。

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2006/070022

A. CLASSIFICATION OF SUBJECT MATTER INV. G06F17/30		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 513 314 A (KANDASAMY DAVID R [US] ET AL) 30 Apr11 1996 (1996-04-30) column 16, line 45 - column 17, line 33	1-19
X	US 6 460 055 B1 (MIDGLEY CHRISTOPHER [US] ET AL) 1 October 2002 (2002-10-01) column 9, line 39 - column 11, line 20	1-19
A	EP 0 278 472 A2 (IBM [US]) 17 August 1988 (1988-08-17) page 5, line 31 - page 6, line 14	1-19
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents :		
A document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed		*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *Z* document member of the same patent family
Date of the actual completion of the international search 25 September 2007		Date of mailing of the international search report 05/10/2007
Name and mailing address of the ISA/ European Patent Office, P.B. 5618 Patentlaan 2 NL - 2280 HV Rijswijk Tel: (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer Zubrzycki, Wojciech

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2006/070022

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5513314	A	30-04-1996	AU 4767796 A 14-08-1996
			CA 2211654 A1 01-08-1996
			EP 0806010 A1 12-11-1997
			JP 11502644 T 02-03-1999
			WO 9623259 A1 01-08-1996
US 6460055	B1	01-10-2002	US 2002174139 A1 21-11-2002
EP 0278472	A2	17-08-1988	BR 8800330 A 13-09-1988
			DE 3853727 D1 14-06-1995
			DE 3853727 T2 25-01-1996
			JP 1860471 C 27-07-1994
			JP 5064811 B 16-09-1993
			JP 63201743 A 19-08-1988
			US 5151989 A 29-09-1992

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

(74)代理人 100086243

弁理士 坂口 博

(72)発明者 エヴァーハート、クレイグ、フルマー

アメリカ合衆国 2 7 5 1 4 ノース・カロライナ州チャペル・ヒル ハンティントン・ドライブ 2
2 5

Fターム(参考) 5B082 DE04 EA01 HA01