

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-146601

(P2008-146601A)

(43) 公開日 平成20年6月26日(2008.6.26)

(51) Int.Cl.
G06F 21/24 (2006.01)F I
G06F 12/14 560Cテーマコード (参考)
5B017

審査請求 未請求 請求項の数 7 O L (全 14 頁)

(21) 出願番号 特願2006-336372 (P2006-336372)
(22) 出願日 平成18年12月13日 (2006.12.13)(71) 出願人 000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号
(74) 代理人 100076428
弁理士 大塚 康德
(74) 代理人 100112508
弁理士 高柳 司郎
(74) 代理人 100115071
弁理士 大塚 康弘
(74) 代理人 100116894
弁理士 木村 秀二
(72) 発明者 今井 聡
東京都大田区下丸子3丁目30番2号 キ
ヤノン株式会社内

最終頁に続く

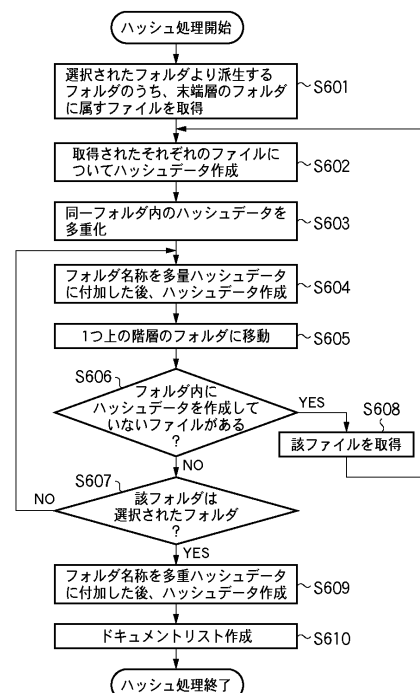
(54) 【発明の名称】 情報処理装置及び情報処理方法

(57) 【要約】

【課題】 フォルダに格納されたファイルの改竄等を容易に検証できるようにする。

【解決手段】 フォルダに格納されたファイルをハッシュ処理する情報処理方法であって、フォルダを指定する指定工程と、前記指定工程により指定されたフォルダに含まれるファイルそれぞれについて、ハッシュデータを作成する第1の作成工程（ステップS601、602）と、前記第1の作成工程において作成されたハッシュデータを順次結合するとともに、結合するごとに新たにハッシュデータを作成する処理を繰り返すことで、1つの多重ハッシュデータを作成する第2の作成工程（ステップS603～609）と、前記指定工程により指定されたフォルダに対応付けて、前記多重ハッシュデータを格納する工程（ステップS610）とを備える。

【選択図】 図6



【特許請求の範囲】**【請求項 1】**

フォルダに格納されたファイルをハッシュ処理する情報処理装置であって、

フォルダを指定する指定手段と、

前記指定手段により指定されたフォルダに含まれるファイルそれぞれについて、ハッシュデータを作成する第 1 の作成手段と、

前記第 1 の作成手段において作成されたハッシュデータを順次結合するとともに、結合するごとに新たにハッシュデータを作成する処理を繰り返すことで、1 つの多重ハッシュデータを作成する第 2 の作成手段と、

前記指定手段により指定されたフォルダに対応付けて、前記多重ハッシュデータを格納する格納手段と

10

を備えることを特徴とする情報処理装置。

【請求項 2】

前記指定手段により指定されたフォルダに含まれるファイルの改竄の有無を検証する検証手段を更に備え、

前記検証手段は、前記指定手段により指定されたフォルダに含まれるファイルに対して、前記第 1 および第 2 の作成手段を動作させることにより作成された多重ハッシュデータと、前記指定手段により指定されたフォルダに対応付けて、前記格納手段に既に格納されている多重ハッシュデータとを比較することにより、改竄の有無を検証することを特徴とする請求項 1 に記載の情報処理装置。

20

【請求項 3】

前記第 1 の作成手段は、

前記指定手段により指定されたフォルダに含まれるファイルのうち、該フォルダの指定を行ったユーザがアクセス権を有するファイルについて、前記ハッシュデータを作成することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 4】

前記第 2 の作成手段は、

前記ハッシュデータを結合するにあたり、該ハッシュデータの作成に用いられたファイルが属するフォルダの名称を付加することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 5】

30

フォルダに格納されたファイルをハッシュ処理する情報処理方法であって、

フォルダの指定を受け付ける受付工程と、

前記受付工程において指定を受け付けたフォルダに含まれるファイルそれぞれについて、ハッシュデータを作成する第 1 の作成工程と、

前記第 1 の作成工程において作成されたハッシュデータを順次結合するとともに、結合するごとに新たにハッシュデータを作成する処理を繰り返すことで、1 つの多重ハッシュデータを作成する第 2 の作成工程と、

前記指定工程により指定されたフォルダに対応付けて、前記多重ハッシュデータを格納する格納工程と

40

を備えることを特徴とする情報処理方法。

【請求項 6】

請求項 5 に記載の情報処理方法をコンピュータによって実現させるための制御プログラムを格納した記憶媒体。

【請求項 7】

請求項 5 に記載の情報処理方法をコンピュータによって実現させるための制御プログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、フォルダに格納されたファイル进行处理する情報処理技術に関するものである

50

。

【背景技術】

【0002】

従来より、ファイル管理システムにおいて省令に基づくファイルを管理する場合、フォルダは、“2005年10月フォルダ”、“2005年11月フォルダ”等といったように、年月日に基づいて構成されてきた。

【0003】

このように年月日に基づいてフォルダを構成することで、ユーザは、いつ作成され保管されたファイルであるのかを一目で判別することができ、ファイルを探す手間を省くことができるというメリットがある。また、保管期限に達したファイルは廃棄する必要があるが、その場合でも、上記のように年月日に基づくフォルダ構成にしておくことで、フォルダ単位でまとめて削除することができる。

【特許文献1】特開2002-229835号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

一方で、セキュリティ向上のためのe文書法が2005年春に制定されたことに伴い、その後、各省庁から関連省令が出されてきているが、かかる省令の中には、ファイルの改竄等（以下、改竄等）を検知するための「検証」手段を必須としているものもある。例えば、国税庁、金融庁、総務省、厚生労働省より新たに出された省令、庁令等がこれにあたる。

【0005】

ファイルの改竄等を検知するためには、予めファイル管理システムにおいて管理されているファイルに対して、改竄等を検知できるようにするための処理（例えば、署名やタイムスタンプを付加し、ハッシュデータを作成する処理等）を施しておく必要がある。

【0006】

しかしながら、格納された大量のファイルを1つ1つ読み出して、このような処理を施すことはユーザにとって作業負荷が大きい。

【0007】

また、このような処理が施されたファイルを1つ1つ読み出して、定期的に、改竄等の有無を検証するのは、ユーザにとって利便性が悪い。

【0008】

このため、複数のファイルに対して一括して処理を施したり、検証したりできるファイル管理システムの開発が望まれている。特に、省令に基づくファイルの場合、上述のようにフォルダ単位で管理されていることが多いことから、フォルダ単位で処理ができることが望ましい。

【0009】

本発明は、上記課題に鑑みてなされたものであり、ファイル管理システムのフォルダに格納されたファイルの改竄等の検証を容易に行うことができるようにする。

【課題を解決するための手段】

【0010】

上記の目的を達成するために本発明に係る情報処理装置は以下のような構成を備える。即ち、

フォルダに格納されたファイルをハッシュ処理する情報処理装置であって、
フォルダを指定する指定手段と、

前記指定手段により指定されたフォルダに含まれるファイルそれぞれについて、ハッシュデータを作成する第1の作成手段と、

前記第1の作成手段において作成されたハッシュデータを順次結合するとともに、結合するごとに新たにハッシュデータを作成する処理を繰り返すことで、1つの多重ハッシュデータを作成する第2の作成手段と、

10

20

30

40

50

前記指定手段により指定されたフォルダに対応付けて、前記多重ハッシュデータを格納する格納手段とを備える。

【発明の効果】

【0011】

本発明によれば、ファイル管理システムのフォルダに格納されたファイルの改竄等の検証を容易に行うことができるようになる。

【発明を実施するための最良の形態】

【0012】

以下、必要に応じて添付図面を参照しながら本発明の各実施形態を詳細に説明する。

【0013】

[第1の実施形態]

1. ファイル管理システムの構成

図1は、本発明の第1の実施形態にかかるサーバ装置（情報処理装置）を備えるファイル管理システム100の構成を示す図である。

【0014】

同図に示すように、ファイル管理システム100は、サーバ装置101と複数のクライアント装置103がネットワーク102を介して通信可能に接続されている。

【0015】

2. サーバ装置の構成

図2は、サーバ装置101の構成を示す図である。図2において、201は制御メモリ（ROM）、202は中央演算処理装置（CPU）、203はメモリ（RAM）、204は外部記憶装置である。また、205は入力装置、206は表示装置、207はネットワーク102と接続するためのI/F装置、208はバスである。本実施形態にかかる情報処理機能を実現するための制御プログラムやその制御プログラムで用いるデータは、外部記憶装置204に記憶される。

【0016】

なお、本実施形態にかかる情報処理機能を実現するための制御プログラムとは、例えば、ユーザインタフェース処理部211、処理対象特定部212、ハッシュ処理部213、検証処理部214等を指す。また、その制御プログラムで用いるデータとは、ファイル管理システム100にて管理されるファイルや、後述するハッシュ処理により作成されるハッシュデータ、ドキュメントリスト等を指す。

【0017】

これらの制御プログラムやデータは、中央演算処理装置202の制御のもと、バス208を通じて適宜メモリ203に取り込まれ、中央演算処理装置202によって実行される。

【0018】

ユーザインタフェース処理部211は、ファイル管理システム100にて管理されているファイルに対して後述するハッシュ処理や検証処理を施す際のユーザインタフェースをユーザに提供する。ファイル管理システム100にて管理されているファイルは、該ユーザインタフェース処理部211を介して表示装置206に表示される。そしてユーザは表示装置206の該表示内容に基づいて、後述するハッシュ処理や検証処理を施す対象および処理内容を入力装置205を介して入力する。入力装置205を介して入力されたこれらの指示内容はユーザインタフェース処理部211にて認識される。

【0019】

212は処理対象特定部であり、ユーザインタフェース処理部211にて認識された指示内容に基づいて、ハッシュ処理や検証処理を施す対象となるファイルを特定する。

【0020】

213はハッシュ処理部であり、ユーザインタフェース処理部211において認識された指示内容がハッシュ処理であった場合に、処理対象特定部212にて特定されたファイルに対してハッシュ処理を実行する。

10

20

30

40

50

【 0 0 2 1 】

2 1 4 は検証処理部であり、ユーザインタフェース処理部 2 1 1 において認識された指示内容が検証処理であった場合に、処理対象特定部 2 1 2 にて特定されたファイルに対して検証処理を実行する。

【 0 0 2 2 】

3 . ユーザインタフェース

図 3 は、ユーザインタフェース処理部 2 1 1 により提供されるユーザインタフェース 3 0 0 の一例を示す図である。

【 0 0 2 3 】

ユーザインタフェース 3 0 0 において、3 0 1 はファイルボタンであり、表示領域 3 0 6 に表示されたファイルやフォルダに対して各種操作（例えば、新規作成、削除、名前の変更、コピー等）を行うためのボタンである。3 0 2 はハッシュボタンであり、ユーザはハッシュボタン 3 0 2 を押下することにより、ハッシュ処理を指示する。3 0 3 は検証ボタンであり、ユーザは検証ボタン 3 0 3 を押下することにより、検証処理を指示する。

【 0 0 2 4 】

3 0 6 は表示領域であり、ファイル管理システム 1 0 0 にて管理されているファイルをフォルダにわけて表示する。なお、本実施形態では、3 0 4 に示される記号はフォルダを表し、3 0 5 に示す記号はファイルを表すものとする。

【 0 0 2 5 】

なお、ハッシュ処理や検証処理を施す対象は、ポインタ 3 0 7 により、表示領域 3 0 6 に表示されているファイルまたはフォルダをユーザが直接選択することにより指定するものとする。

【 0 0 2 6 】

図 4 は、表示領域 3 0 6 上でユーザがフォルダ 4 0 1 を選択することにより、ハッシュ処理を施す対象を指定した様子を示している。

【 0 0 2 7 】

図 4 に示すように、フォルダ 4 0 1 には、1 つ下の階層に 2 つのフォルダ 4 0 2 、4 0 3 があり、フォルダ 4 0 2 には、ファイル 4 0 4 と 4 0 5 とが格納されている。また、フォルダ 4 0 3 には、ファイル 4 0 6 と 4 0 5 とが格納されている。したがって、ユーザがハッシュ処理を施す対象としてフォルダ 4 0 1 を選択した場合、処理対象特定部 2 1 2 では、ハッシュ処理を施すファイルとして、ファイル 4 0 4 、4 0 5 、4 0 6 、4 0 7 を特定する。

【 0 0 2 8 】

このように、本実施形態にかかるサーバ装置 1 0 1 では、処理を施す対象をファイル単位で指定することができるだけでなく、フォルダ単位で選択し該フォルダに含まれる全てのファイルを指定することもできる。

【 0 0 2 9 】

4 . ハッシュ処理

次に、図 5、図 6 を用いてハッシュ処理部 2 1 3 におけるハッシュ処理の詳細を説明する。図 5 は、フォルダ 4 0 1 を選択しハッシュ処理を指示した場合の、ハッシュ処理の流れを示す概念図である。また、図 6 はハッシュ処理部 2 1 3 におけるハッシュ処理の流れを示すフローチャートである。

【 0 0 3 0 】

ステップ S 6 0 1 では、処理対象特定部 2 1 2 において特定された処理対象となるファイルのうち、最初にハッシュ処理を施すファイルを取得する。具体的には、選択されたフォルダより派生するフォルダのうち、末端層のフォルダに含まれるファイルを取得する。

【 0 0 3 1 】

図 5 の例では、選択されたフォルダ 4 0 1 より派生するフォルダとしては、フォルダ 4 0 2 とフォルダ 4 0 3 とがあり、両者はいずれも末端層のフォルダ（それよりも下位層にフォルダがないフォルダ）である。このため、ステップ S 6 0 1 では、ファイル 4 0 4 、

10

20

30

40

50

405、406、407が取得される。

【0032】

ステップS602では、ステップS601で取得されたそれぞれのファイル(404～407)についてハッシュデータを作成する。具体的には、ファイル404についてはハッシュデータ502が、ファイル405についてはハッシュデータ501が作成される。また、ファイル406についてはハッシュデータ506が、ファイル407についてはハッシュデータ505がそれぞれ作成される。

【0033】

ステップS603では、同一フォルダ内のハッシュデータを多重化する処理を行う。ハッシュデータの多重化とは、ハッシュデータとハッシュデータとをあわせて再度固定長のハッシュデータを作成する処理をいう。なお、同一フォルダ内にハッシュデータが3つ以上ある場合には、ファイルの更新順序の古い順から順次多重化を行うものとする。

【0034】

図5の例では、ハッシュデータ501と502とが同一フォルダ402内のハッシュデータであるため、これらを多重化し、ハッシュデータ503を作成する。同様に、ハッシュデータ505と506とが同一フォルダ403内のハッシュデータであるため、これらを多重化し、ハッシュデータ507を作成する。

【0035】

ステップS604では、多重化されたハッシュデータに、該ハッシュデータの作成に用いられたファイルが属するフォルダのフォルダ名称を付加した後、再度ハッシュ処理を行う。図5の例では、フォルダ402のフォルダ名称をハッシュデータ503に付加した後、ハッシュ処理を行い、ハッシュデータ504を作成する。同様に、フォルダ403のフォルダ名称をハッシュデータ507に付加した後、ハッシュ処理を行い、ハッシュデータ508を作成する。

【0036】

ステップS605では、1つ上の階層のフォルダに移動する。本実施形態の場合、末端層のフォルダであるフォルダ402、403から、1つ上の階層のフォルダであるフォルダ401に移動する。

【0037】

ステップS606では、移動後のフォルダ内にハッシュデータを作成していないファイルがあるか否かを判定する。移動後のフォルダ内にハッシュデータを作成していないファイルがあると判定された場合には、ステップS608に進む。ステップS608では、該ファイルを取得した後、ステップS602に戻る。一方、移動後のフォルダ内にハッシュデータを作成していないファイルがないと判定された場合には、ステップS607に進む。ステップS607では、該移動後のフォルダが選択されたフォルダであるか否かを判定し、選択されたフォルダでないと判定された場合には、ステップS604に戻る。一方、選択されたフォルダであると判定された場合にはステップS609に進む。

【0038】

図5の例では、移動後のフォルダ401内にハッシュデータを作成していないファイルはないため、ステップS607に進むこととなる。更に、移動後のフォルダ401は、選択されたフォルダであるため、ステップS609に進むこととなる。

【0039】

ステップS609では、フォルダ内のハッシュデータを多重化し、該フォルダのフォルダ名称を付加した後、再度ハッシュ処理を行う。図5の例では、フォルダ401内のハッシュデータ504と508とを多重化し、ハッシュデータ509を作成した後、フォルダ401のフォルダ名称を付加し、再度ハッシュ処理を行う。これにより、ハッシュデータ510が作成される。

【0040】

ステップS610では、ステップS601～ステップS609の処理に基づいて、ドキュメントリストを作成する。ドキュメントリストとは、選択されたフォルダと、選択され

10

20

30

40

50

たフォルダについて得られたハッシュデータ、ならびに、該選択されたフォルダに含まれるファイルとそのハッシュデータとの対応関係を示したものである。

【 0 0 4 1 】

図 7 はステップ S 6 1 0 にて作成されたドキュメントリスト 7 0 0 の一例を示す図である。7 0 1 には選択されたフォルダとして、フォルダ 4 0 1 が格納され、7 0 2 には該選択されたフォルダ 4 0 1 についてハッシュ処理された結果得られた多重ハッシュデータとしてハッシュデータ 5 1 0 が格納される。

【 0 0 4 2 】

7 0 3 にはフォルダ 4 0 1 に含まれるファイル 4 0 4 ~ 4 0 7 がフォルダ 4 0 1 と対応付けて格納され、7 0 4 には該ファイル 4 0 4 ~ 4 0 7 のハッシュデータ 5 0 1、5 0 2、5 0 5、5 0 6 が格納される。

【 0 0 4 3 】

ステップ S 6 1 0 においてドキュメントデータの作成が完了すると、ハッシュ作成処理を終了する。

【 0 0 4 4 】

なお、上記ハッシュ処理では、各ファイルについてハッシュデータを作成し、多重化した後に、該ファイルの属するフォルダのフォルダ名称を付加した上で、再度ハッシュデータを作成する構成としているが、本発明は特にこれに限られない。単に各ファイルについてハッシュデータを作成し、これらを所定の順序で多重化して多重ハッシュデータを作成するようにしてもよい。

【 0 0 4 5 】

なお、本実施形態のような方法でハッシュデータを作成するのは、改竄等を検知する際に、ファイルそのものの改竄等を検知するのみならず、ファイルを別のフォルダに移動する改竄をも検知できるようにするためである。

【 0 0 4 6 】

つまり、単に各ファイルについてハッシュデータを作成し、これらを所定の順序で多重化していくだけの場合、特定のファイルを別のフォルダに移動する改竄を行った場合にこれを検知することができないことがあるからである。移動前に作成された多重ハッシュデータと、移動後に作成された多重ハッシュデータとが同じになる場合がこれにあたり、本実施形態の方法によればこのような事態を回避することができる。

【 0 0 4 7 】

5 . 検証処理

次に、図 8 を用いて検証処理について説明する。図 8 は検証処理部 2 1 4 における処理の流れを示すフローチャートである。

【 0 0 4 8 】

ユーザインタフェース 3 0 0 上で所定のフォルダが選択された後、検証ボタン 3 0 3 が押下されると、検証処理部 2 1 4 では検証処理を開始する。

【 0 0 4 9 】

ステップ S 8 0 1 では、ユーザインタフェース 3 0 0 上で選択されたフォルダについてハッシュ作成処理（図 6）を実行する。なお、ここでもフォルダ 4 0 1 が選択されたものとして説明を進める。

【 0 0 5 0 】

ステップ S 8 0 2 では、ステップ S 8 0 1 のハッシュ処理の結果得られたハッシュデータを取得する。

【 0 0 5 1 】

ステップ S 8 0 3 では、選択されたフォルダに対応するハッシュデータをドキュメントリスト 7 0 0 に基づいて取得する。図 7 の例では、選択されたフォルダ 4 0 1 に対応するハッシュデータとしてハッシュデータ 5 1 0 が取得される。

【 0 0 5 2 】

ステップ S 8 0 4 では、ステップ S 8 0 2 において取得されたハッシュデータとステッ

10

20

30

40

50

ブ S 8 0 3 において取得されたハッシュデータとを比較する。比較の結果、同じであればステップ S 8 0 5 に進む。ステップ S 8 0 2 において取得されたハッシュデータとステップ S 8 0 3 において取得されたハッシュデータが同じであるということは、選択されたフォルダ 4 0 1 に含まれるファイル 4 0 4 ~ 4 0 7 について改竄等がなされていないことを意味する。また、選択されたフォルダ 4 0 1 に含まれるファイル 4 0 4 ~ 4 0 7 の中に、フォルダをまたいで移動されたファイルがないことを意味する。したがって、ステップ S 8 0 5 では“ 検証 O K ”を報知する（ユーザインタフェース 3 0 0 上に表示する）。

【 0 0 5 3 】

一方、比較の結果、異なっていた場合には、ステップ S 8 0 6 に進む。ステップ S 8 0 2 において取得されたハッシュデータとステップ S 8 0 3 において取得されたハッシュデータが異なっているということは、選択されたフォルダ 4 0 1 に含まれるファイル 4 0 4 ~ 4 0 7 のいずれかについて改竄等がなされたことを意味する。あるいは、選択されたフォルダ 4 0 1 に含まれるファイル 4 0 4 ~ 4 0 7 の中に、フォルダをまたいで移動されたファイルがあったことを意味する。したがって、ステップ S 8 0 6 では、“ 検証 N G ”を報知する（ユーザインタフェース 3 0 0 上に表示する）。

【 0 0 5 4 】

ステップ S 8 0 5 またはステップ S 8 0 6 において、いずれかの報知が完了すると、検証処理を終了する。

【 0 0 5 5 】

以上の説明から明らかなように、本実施形態によれば、ファイル管理システムにおいて管理されているファイルに対して、改竄等を検知できるようにするための処理を施すにあたり、フォルダ単位で処理を実行させることが可能となる。この結果、従来のような、ファイルごとに処理を実行させる場合と比べて、ユーザの負荷を大幅に軽減させることが可能となる。

【 0 0 5 6 】

また、本実施形態によれば、ファイル管理システムにおいて管理されているファイルに対して、改竄等の有無を検証するにあたり、フォルダ単位で処理を実行させることが可能となる。この結果、従来のような、ファイルごとに検証処理を実行させる場合と比べて、ユーザの負荷を大幅に軽減させることが可能となる。

【 0 0 5 7 】

[第 2 の実施形態]

上記第 1 の実施形態では、ユーザが所定のフォルダを選択した場合、該フォルダに含まれるすべてのファイルを処理対象としたが、本発明は特にこれに限定されない。ファイル管理システムによっては、ログイン時にユーザ ID を入力させ、ユーザ ID ごとにアクセス権を設定することで、アクセスできるファイルに制限を加えているものもある。このようなシステムにおいて、アクセス権をもたないファイルについても上述のような処理が行われることは望ましくない。

【 0 0 5 8 】

これを考慮し、本実施形態では、ユーザが所定のフォルダを選択した場合、該フォルダに含まれるファイルのうち、該ユーザがアクセス権を有するファイルのみを処理対象とすることとした。以下、本実施形態にかかるサーバ装置について説明する。なお、ファイル管理システムの構成、サーバ装置の構成等は上記第 1 の実施形態と同じであるため、ここでは説明を割愛する。

【 0 0 5 9 】

1 . ユーザインタフェース

図 9 は、本実施形態にかかるサーバ装置のユーザインタフェース 3 0 0 を示す図である。同図において、例えば、実線で図示された記号 3 0 5 はアクセス可能なファイルを示している。一方、点線で図示された記号 9 0 1、9 0 2 で表されるファイルは、それぞれファイル管理システムにおいて管理されているファイルであるが、現在ログインしているユーザがアクセス権を有していないファイルである。なお、図 9 の例では、便宜上、点線で

10

20

30

40

50

図示しているが、実際のユーザインタフェース 300 上では、ファイル 901、902 は表示されていないものとする。

【0060】

2. ハッシュ処理

図10は、本実施形態にかかるサーバ装置におけるハッシュ処理の流れを示すフローチャートである。なお、図6に示したフローチャートと同じ処理を行うステップについては、同じ参照番号を付すこととし、説明は省略する。

【0061】

ステップ S1001 では、処理対象特定部 212 において特定された処理対象となるファイルのうち、最初にハッシュ処理を施すファイルを取得する。具体的には、選択されたフォルダより派生するフォルダのうち、末端層のフォルダに含まれるファイルであって、かつユーザがアクセス権を有するファイルを取得する。

10

【0062】

図9の例では、選択されたフォルダ 401 より派生するフォルダとして、フォルダ 402 とフォルダ 403 とがあり、両者はいずれも末端層のフォルダである。また、フォルダ 402 とフォルダ 403 に含まれるファイルとして、ファイル 404、902、406、407 がある。しかし、このうちファイル 902 については、現在ログインしているユーザにはアクセス権がないことから、ステップ S1001 では、ファイル 902 を除くファイル 404、406、407 が取得される。

【0063】

20

ステップ S602 からステップ S605 については図6のステップ S602 からステップ S605 の処理と同様であるため、説明は省略する。

【0064】

ステップ S1002 では、移動後のフォルダ内に、ハッシュデータを作成していないファイルであって、現在ログインしているユーザがアクセス権を有するファイルがあるか否かを判定する。移動後のフォルダ内に、ハッシュデータを作成していないファイルであって、ユーザがアクセス権を有するファイルがあると判定された場合には、ステップ S607 に進む。一方、移動後のフォルダ内に、ハッシュデータを作成していないファイルであって、ユーザがアクセス権を有するファイルがあると判定された場合には、ステップ S1003 に進む。ステップ S1003 では、当該ファイルを取得し、ステップ S602 に戻る。

30

【0065】

一方、ステップ S607、ステップ S609、ステップ S610 については図6のステップ S607、ステップ S609、ステップ S610 の処理と同様であるため、説明は省略する。

【0066】

以上の説明から明らかなように、本実施形態によれば、ファイル管理システムにおいて管理されているファイルを、フォルダ単位で処理を実行させることが可能となるうえ、ユーザのアクセス権の有無に応じて選択的に処理を実行させることが可能となる。

【0067】

40

この結果、アクセス権をもたないファイルに対しても処理が実行されてしまうといった事態を回避させることが可能となる。

【0068】

また、検証を行う際も同様に、ユーザのアクセス権のあるファイルのみを対象として、フォルダ単位で処理を実行させることができる。このように構成することにより、フォルダ内に含まれる当該ユーザと関係のないファイル（当該ユーザの興味がないファイル）が更新されたとしても、改竄されたと報知しないようにできる。言い換えれば、フォルダ単位で検証を行う際に、ユーザのアクセス権のあるファイル（ユーザの興味があるファイル）のみを対象として検証を行うようにできる。

【0069】

50

〔第3の実施形態〕

上記第1および第2の実施形態では、ファイルの改竄等の検知を、ファイルのハッシュデータの一致／不一致に基づいて行うこととしたが、本発明は特にこれに限定されない。

【0070】

例えば、認証局が発行する秘密鍵で作成した署名情報をファイルに付加したうえで、ハッシュデータを作成するようにしてもよい。

【0071】

〔他の実施形態〕

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0072】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給するよう構成することによっても達成されることはいうまでもない。この場合、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することにより、上記機能が実現されることとなる。なお、この場合、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0073】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0074】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現される場合に限られない。例えば、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることはいうまでもない。

【0075】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、前述した実施形態の機能が実現される場合も含まれる。つまり、プログラムコードがメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって実現される場合も含まれる。

【図面の簡単な説明】

【0076】

【図1】本発明の第1の実施形態にかかるサーバ装置（情報処理装置）を備えるファイル管理システム100の構成を示す図である。

【図2】サーバ装置101の構成を示す図である。

【図3】ユーザインタフェース処理部211により提供されるユーザインタフェース300の一例を示す図である。

【図4】表示領域306上でユーザがフォルダ401を選択することにより、ハッシュ処理を施す対象を指定した様子を示す図である。

【図5】フォルダ401を選択した場合の、ハッシュ処理の流れを示す概念図である。

【図6】ハッシュ処理部213における処理の流れを示すフローチャートである。

【図7】ステップS609にて作成されたドキュメントデータ700の一例を示す図である。

【図8】検証処理部214における処理の流れを示すフローチャートである。

【図9】本発明の第2の実施形態にかかるサーバ装置のユーザインタフェース300を示

10

20

30

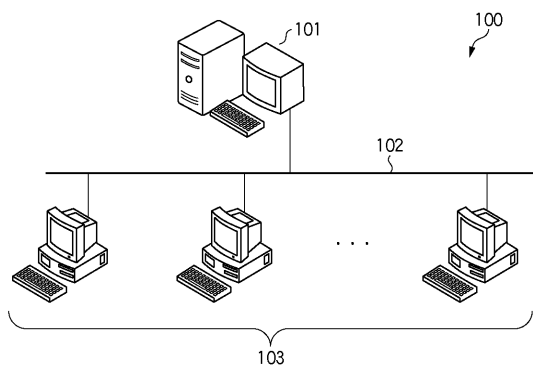
40

50

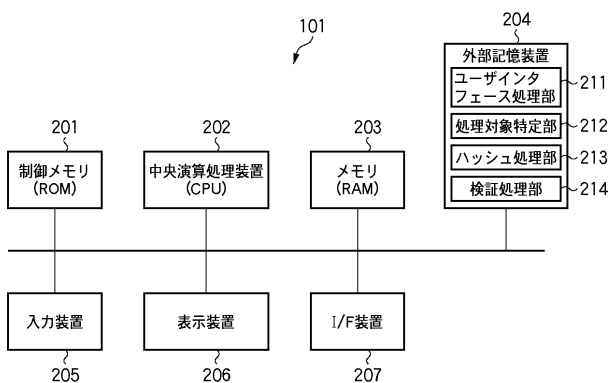
す図である。

【図10】本発明の第2の実施形態にかかるサーバ装置におけるハッシュ作成処理の流れを示すフローチャートである。

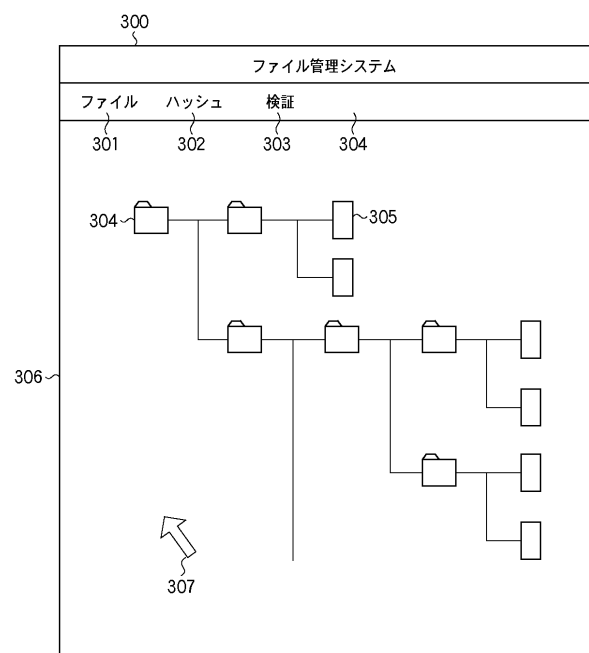
【図1】



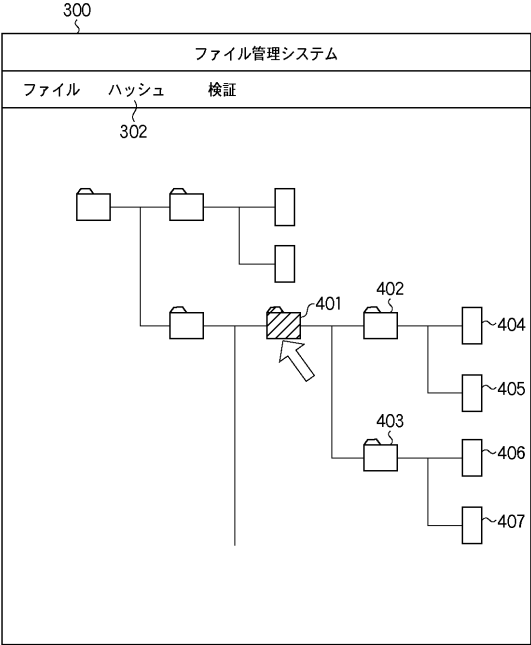
【図2】



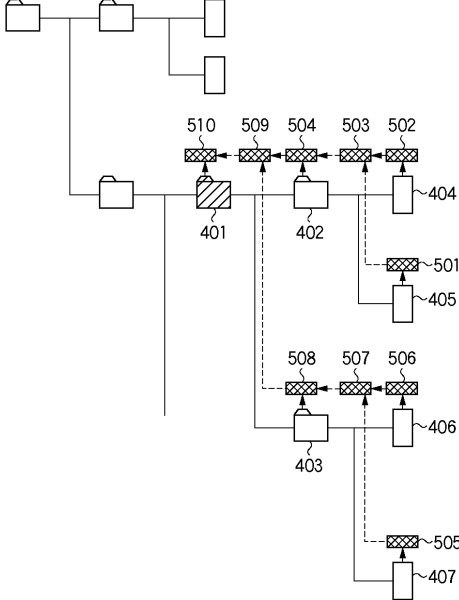
【図3】



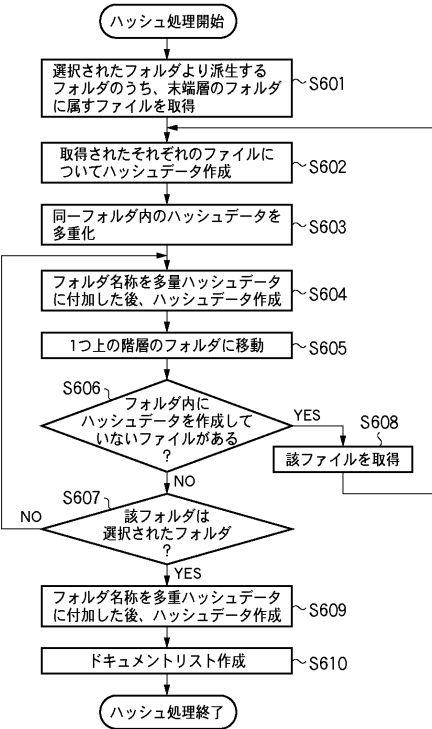
【 図 4 】



【 図 5 】



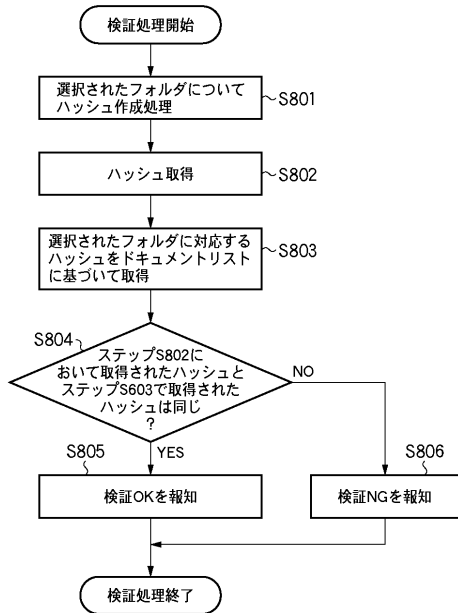
【 図 6 】



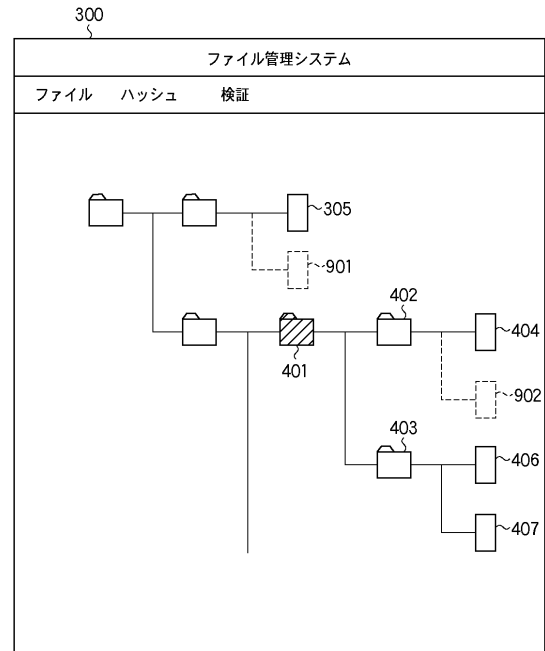
【 図 7 】

701 フォルダ	702 多重ハッシュ データ	703 ファイル	704 ハッシュ
フォルダ 401	ハッシュ データ 510	ファイル405	ハッシュデータ 501
		ファイル404	ハッシュデータ 502
		ファイル407	ハッシュデータ 505
		ファイル406	ハッシュデータ 506

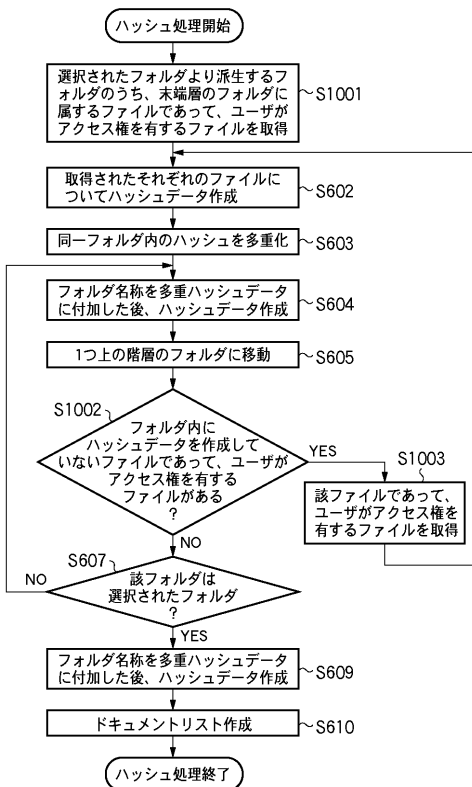
【 図 8 】



【 図 9 】



【 図 1 0 】



フロントページの続き

(72)発明者 黒田 健

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

Fターム(参考) 5B017 AA08 CA16