

(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2008-257584  
(P2008-257584A)

(43) 公開日 平成20年10月23日(2008.10.23)

(51) Int.Cl.		F I			テーマコード (参考)
<b>G06F 12/00</b>	<b>(2006.01)</b>	G06F 12/00	511A		5B065
<b>G06F 3/08</b>	<b>(2006.01)</b>	G06F 3/08	C		5B082

審査請求 未請求 請求項の数 12 O L (全 18 頁)

(21) 出願番号	特願2007-100823 (P2007-100823)	(71) 出願人	000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号
(22) 出願日	平成19年4月6日(2007.4.6)	(74) 代理人	110000350 ポレール特許業務法人
		(72) 発明者	伊藤 昭博 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
		(72) 発明者	中野 道樹 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
		Fターム(参考)	5B065 BA09 ZA16 5B082 GA01 JA07

(54) 【発明の名称】 RAMディスクの処理方法及びシステム

(57) 【要約】

【課題】

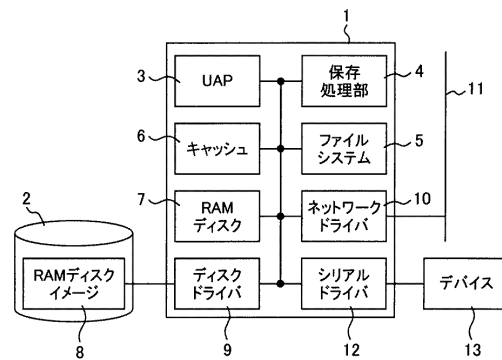
RAMディスクを既存のファイルシステムとのデータ形式と互換性を保ちつつ、RAMディスクイメージの圧縮率を向上させる。

【解決手段】

RAMディスクの利用を開始する時に、二次記憶装置に圧縮して保存されたRAMディスクイメージをから読み出してメモリに格納し、RAMディスクの利用を終了する時にRAMディスクイメージを二次記憶装置へ圧縮保存するRAMディスクの処理方法において、RAMディスクイメージを圧縮保存処理する前に、RAMディスクの無効領域をクリア処理する。クリア処理は、例えばRAMディスクイメージを二次記憶装置へ保存する直前に、無効領域に所定のデータ0'を連続的に書き込むことにより行う。

【選択図】 図1

図 1



**【特許請求の範囲】****【請求項 1】**

R A M ディスクの利用を開始する時に、二次記憶装置に圧縮して保存された R A M ディスクイメージをから読み出してメモリに格納し、該 R A M ディスクの利用を終了する時に該 R A M ディスクイメージを該二次記憶装置へ圧縮保存する R A M ディスクの処理方法において、該 R A M ディスクイメージを圧縮保存処理する前に、該 R A M ディスクの無効領域をクリア処理することを特徴とする R A M ディスクの処理方法。

**【請求項 2】**

前記クリア処理は、該 R A M ディスクイメージを該二次記憶装置へ保存する直前に、該無効領域に所定のデータを連続的に書き込むことにより行うことを特徴とする請求項 1 の R A M ディスクの処理方法。 10

**【請求項 3】**

前記クリア処理は、該 R A M ディスクを使用中に、該無効領域を任意の容量ずつ分けて、所定のデータを連続的に書き込むことにより行うことを特徴とする請求項 1 の R A M ディスクの処理方法。

**【請求項 4】**

前記クリア処理は、ファイルの削除又はファイルサイズを縮小する時に、該 R A M ディスクの無効となった領域に対して、所定のデータを連続的に書き込むことにより行うことを特徴とする請求項 1 の R A M ディスクの処理方法。

**【請求項 5】**

R A M ディスクを利用する第 1 の装置と、 R A M ディスクイメージを管理する第 2 の装置がネットワークを介して接続され、第 1 の装置は R A M ディスクの利用を開始する時に該 R A M ディスクイメージを第 2 の装置から読み出してメモリに格納し、該 R A M ディスクの利用を終了する時に該 R A M ディスクイメージを該メモリから読み出して該第 2 の装置に保存する R A M ディスクの処理方法において、  
該 R A M ディスクの利用率が第 1 の基準値未満の場合、第 1 の装置で、該 R A M ディスクの無効領域をクリアする処理と、該 R A M ディスクイメージを圧縮する処理を行い、  
該 R A M ディスクの利用率が第 2 の基準値未満の場合、該第 2 の装置で、該 R A M ディスクの無効領域をクリアする処理と、該 R A M ディスクイメージを圧縮する処理を行い、  
該 R A M ディスクの利用率が第 2 の基準値以上の場合、該 R A M ディスクの無効領域をク 30  
リアする処理を行わずに、該 R A M ディスクイメージを圧縮処理しないで該第 2 の装置に保存することを特徴とする R A M ディスクの処理方法。

**【請求項 6】**

R A M ディスクを利用する第 1 の装置と、 R A M ディスクイメージを管理する第 2 の装置がネットワークを介して接続され、第 1 の装置は R A M ディスクの利用を開始する時に該 R A M ディスクイメージを第 2 の装置から読み出してメモリに格納し、該 R A M ディスクの利用を終了する時に該 R A M ディスクイメージを該メモリから読み出して該第 2 の装置に保存する R A M ディスクの処理方法において、  
該第 2 の装置は、該第 1 の装置へ配信する該 R A M ディスクイメージを第 2 の装置内で更新することでメンテナンス処理を行い、  
メンテナンス処理が終了した後、該第 2 の装置は該メンテナンス処理によって生じた該 R A M ディスクの無効領域をクリア処理した後、該 R A M ディスクイメージを圧縮処理して保存することを特徴とする R A M ディスクの処理方法。 40

**【請求項 7】**

少なくともファイルを含む R A M ディスクイメージを記憶する二次記憶装置から該 R A M ディスクイメージをロードして使用する情報処理システムにおいて、  
ロードされた該 R A M ディスクイメージを格納するメモリと、プログラムを実行する C P U とを有し、該 C P U によるプログラムの実行によって、少なくとも該二次記憶装置に対する該 R A M ディスクイメージの保存処理を行う保存処理部と、該メモリに格納された該 R A M ディスクに含まれるファイルを管理するファイルシステムを実現し、 50

該保存処理部は、該RAMディスクの利用を終了する時に、該RAMディスクの無効領域をクリア処理し、該クリア処理された該RAMディスクイメージを圧縮処理して、該二次記憶装置に保存することを特徴とする情報処理システム。

【請求項8】

前記保存処理部は、該RAMディスクイメージを該二次記憶装置へ保存する前に、該無効領域に所定のデータを連続的に書き込むことを特徴とする請求項7の情報処理システム。

【請求項9】

前記ファイルシステムと協働して実行するディスクチェッカ部を有し、該ディスクチェッカ部は、該RAMディスクを使用中に、該無効領域を任意の容量ずつ分けて、所定のデータを連続的に書き込むことを特徴とする請求項7の情報処理システム。

10

【請求項10】

前記ファイルシステムは、ファイルの削除又はファイルサイズを縮小する時に、該RAMディスクの無効となった領域に対して、所定のデータを連続的に書き込むことを特徴とする請求項7の情報処理システム。

【請求項11】

該RAMディスクは、該ファイルシステムの管理情報を保存するファイルシステム管理領域と、各ファイルの属性情報を記録するファイル管理領域と、各ファイルの内容を保存するデータブロック領域を有し、

前記保存処理部は、該メモリにロードされた該RAMディスク内の、該ファイル管理領域又はデータブロック領域の少なくとも一方の領域に存する無効領域のクリア処理を行うことを特徴とする請求項7乃至10のいずれかの情報処理システム。

20

【請求項12】

少なくともファイルを含むRAMディスクイメージを記憶する二次記憶装置から該RAMディスクイメージをロードして使用する情報処理装置において実行されるプログラムであって、該二次記憶装置に該RAMディスクイメージをアンロードする時に、該RAMディスクの無効領域をクリア処理し、該クリア処理された該RAMディスクイメージを圧縮処理することを特徴とするRAMディスクの保存処理用のプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はRAMディスクの処理方法及びシステムに係り、特に組込機器など計算機資源に制限がある情報処置装置において、RAMディスクイメージを圧縮して二次記憶装置に保存するRAMディスクの運用処理に関する。

30

【背景技術】

【0002】

一般に情報処理システムでは、データの保存にハードディスク(HDD)やコンパクトフラッシュ(登録商標)等の記録媒体が利用されている。しかし、これらの記録媒体は、(1)アクセス速度が遅い、(2)アクセス中にシステム障害が発生すると媒体自体が壊れてしまうことがある、等の欠点がある。この欠点を補う技術として、ランダムアクセスメモリ(RAM)を記録媒体として利用するRAMディスクが実用化している。

40

【0003】

RAMディスクはシステム障害が発生しても記録媒体であるRAMが壊れることはないが、システムを停止する度にデータが消えるという欠点を持つ。そこで、RAMディスクを利用する場合、システム停止時にRAMディスクの内容をHDDなどの他の記録媒体に保存しておき、システム起動時にその記録媒体からRAMディスクを復元する運用が行われている。

【0004】

例えば特開2003-122647公報には、情報処理装置内で故障が検出されると、RAMディスクの内容を二次記憶装置に保存する技術が開示されている。とりわけ、その保存の際に、RAMディスクイメージを圧縮し、かつRAMディスクの有効領域のみ保存

50

することで、二次記憶装置の記録量を削減する工夫が図られている。

【0005】

【特許文献1】特開2003-122647公報

【発明の開示】

【発明が解決しようとする課題】

【0006】

一般的なファイルシステムにおいてはファイルを削除する時に管理情報のみを削除するため、ディスクイメージ内の無効領域には以前に記録されたデータが残される。従って、特許文献1に記載されているRAMディスクイメージを圧縮する方法を運用し続けると、無効領域に残されたデータも含めたRAMディスクイメージを圧縮するために、圧縮率が次第に低下するという課題がある。圧縮率が低下すると、データ圧縮処理（データのモデル化や符号化処理）が複雑になるため、圧縮処理時間が増加するという副次的な問題も発生する。

10

【0007】

また特許文献1には、別の保存方式としてRAMディスクの有効領域のみを保存する方式が記載されている。この方式では上記のような問題は発生しないが、作成されるRAMディスクイメージが元のRAMディスクのファイルシステムイメージとは異なる形式で保存されるため、他のシステムとの互換性がなくなるという課題がある。

【0008】

RAMディスクを利用する組込機器のシステムメンテナンスでは、組込機器からRAMディスクイメージを読み出し、他の管理端末上でRAMディスクイメージをマウントして内容を更新した後、更新後のRAMディスクイメージを組込機器へ書き戻すという運用が行われている。RAMディスクイメージが他のシステムと互換性が無くなると、管理端末上でRAMディスクイメージをマウントすることができなくなり、運用に支障をきたす。

20

【0009】

また、停電やバッテリー不足等で情報処理装置の緊急停止が必要になった場合、速やかにRAMディスクを二次記憶装置へ保存する必要があるが、RAMディスクの有効領域のみを保存するためにはRAMディスクの管理領域をチェックする必要があり、プロセッサ性能が低い組込機器では、処理に時間がかかるという課題がある。

【0010】

本発明の目的は、RAMディスクを既存のファイルシステムのデータ形式と互換性を保ちつつ、RAMディスクイメージの圧縮率を向上させることができるRAMディスクの処理方法及びそのシステムを提供する。

30

【課題を解決するための手段】

【0011】

本発明に係るRAMディスクの処理（第1発明）は、好ましくは、RAMディスクの利用を開始する時に、二次記憶装置に圧縮して保存されたRAMディスクイメージをから読み出してメモリに格納し、RAMディスクの利用を終了する時にRAMディスクイメージを二次記憶装置へ圧縮保存するRAMディスクの処理方法において、RAMディスクイメージを圧縮保存処理する前に、RAMディスクの無効領域をクリア処理することにより実現される。

40

好ましくは、前記クリア処理は、RAMディスクイメージを二次記憶装置へ保存する直前に、無効領域に所定のデータ、例えばゼロ0'を連続的に書き込むことにより行う。

また、好ましくは、前記クリア処理は、RAMディスクを使用中に、無効領域を任意の容量ずつ分けて、所定のデータを連続的に書き込むことにより行う。

また、好ましくは、前記クリア処理は、ファイルの削除又はファイルサイズを縮小する時に、RAMディスクの無効となった領域に対して、所定のデータを連続的に書き込むことにより行う。

【0012】

本発明に係るRAMディスクの処理方法（第2発明）は、好ましくは、RAMディスク

50

を利用する第1の装置（例えば情報処理装置）と、RAMディスクイメージを管理する第2の装置（例えば管理端末）がネットワークを介して接続され、第1の装置はRAMディスクの利用を開始する時にRAMディスクイメージを第2の装置から読み出してメモリに格納し、RAMディスクの利用を終了する時にRAMディスクイメージをメモリから読み出して第2の装置に保存するRAMディスクの処理方法において、RAMディスクの利用率が第1の基準値未満の場合、第1の装置で、RAMディスクの無効領域をクリアする処理と、RAMディスクイメージを圧縮する処理を行い、RAMディスクの利用率が第2の基準値未満の場合、第2の装置で、RAMディスクの無効領域をクリアする処理と、RAMディスクイメージを圧縮する処理を行い、RAMディスクの利用率が第2の基準値以上の場合、RAMディスクの無効領域をクリアする処理を行わずに、RAMディスクイメージを圧縮処理しないで第2の装置に保存するRAMディスクの処理方法として構成される。

10

**【0013】**

また、本発明に係るRAMディスクの処理方法（第3発明）は、好ましくは、RAMディスクを利用する第1の装置と、RAMディスクイメージを管理する第2の装置がネットワークを介して接続され、第1の装置はRAMディスクの利用を開始する時にRAMディスクイメージを第2の装置から読み出してメモリに格納し、RAMディスクの利用を終了する時にRAMディスクイメージをメモリから読み出して第2の装置に保存するRAMディスクの処理方法において、第2の装置は、第1の装置へ配信するRAMディスクイメージを第2の装置内で更新することでメンテナンス処理を行い、メンテナンス処理が終了した後、第2の装置はメンテナンス処理によって生じたRAMディスクの無効領域をクリア処理した後、RAMディスクイメージを圧縮処理して保存するRAMディスクの処理方法として構成される。

20

また、本発明は、上記第1乃至第3の処理を実行する情報処理システム、及びプログラムとしても把握され得る。

**【発明の効果】****【0014】**

本発明によれば、RAMディスクイメージを二次記憶装置に保存するときにRAMディスクの無効領域をクリア処理することで、既存システムとの互換性を保ちつつ、RAMディスクイメージの圧縮率を向上することが可能である。

30

また、RAMディスク利用中に無効領域のクリア処理を任意の容量ずつ分けて実行することで、RAMディスクイメージを保存する時にクリア処理を省略することが可能となり、RAMディスクイメージの保存処理を高速化することができる。

また、ファイル削除及びファイルサイズ縮小時にRAMディスクの無効領域をクリア処理し、RAMディスクイメージを保存する時に全ての無効領域がクリアされている状態とすることで、RAMディスクの圧縮率を向上させることができる。

**【0015】**

また、第2発明によれば、RAMディスクの利用率に応じて、第1の装置か又は第2の装置で無効領域のクリア処理とRAMディスクイメージの圧縮処理を行うか、或いは、RAMディスクの無効領域のクリア処理を行わずに、RAMディスクイメージを圧縮しないで保存することで、ネットワークの負荷と圧縮処理時間のバランスを考慮してRAMディスクを運用することが可能となる。

40

また、第3発明によれば、第2の装置は第1の装置へ配信するRAMディスクイメージを第2の装置内で更新することでメンテナンスし、メンテナンス処理を終了した後、第2の装置は上記メンテナンス処理で発生したRAMディスクの無効領域をクリア処理してからRAMディスクイメージを圧縮保存することで、RAMディスクの圧縮率を向上させることができる。

**【発明を実施するための最良の形態】****【0016】**

[実施例1]

50

以下、図面を参照して一実施例について説明する。

図 1 は、一実施例における情報処理システムの構成を示す。図示される情報処理装置 1 の構成は、プログラムの実行によって実現される各種機能を示す。情報処理装置 1 のハードウェアの構成は、図 15 に示される。まず、図 15 を参照するに、情報処理装置 1 は、関係するプログラムを実行して所定の処理機能を実現する CPU 1501、実行する予定のプログラムや本発明に係る RAM ディスクイメージを記憶するメモリ 1502、RAM ディスクイメージ及び種々のプログラム等を記憶する HDD 等の二次記憶装置 1503、デバイスを接続するデバイスアダプタ 1504、及び外部ネットワークと接続するための通信インタフェース部 1505 を有して構成される。なお、情報処理装置 1 のハードウェア構成は、後述する実施例 1 ~ 4 においても同様である。

10

#### 【0017】

次に図 1 を参照するに、ユーザアプリケーション (UAP) 3 は CPU 1501 で実行され、デバイス 13 の制御及びそれから取得されたデータの処理を行う。UAP 3 は、例えば外部ネットワーク 11 に接続された上位装置 (図示せず) からの指示に従ってデバイス 13 を駆動し、その処理結果を上位装置に返す。デバイス 13 はシリアルドライバ 12 を介して駆動される各種デバイスであり、具体的には RFI D リーダ/ライタ、温度センサ、赤外線センサ等である。情報処理装置 1 は、ネットワークドライバ 10 を介して外部ネットワーク 11 と通信可能である。

#### 【0018】

二次記憶装置 2 には RAM ディスクイメージ 8 が保存され、CPU 1501 は、起動時に二次記憶装置 2 に格納された RAM ディスクイメージ 8 を、ディスクドライバ 9 を経由してメモリ 1502 内に読み込み、RAM ディスク 7 を構成する。RAM ディスクイメージ 8 としては、OS (オペレーティングシステム)、USP 3、ファイルシステム 5 で使用される各種ファイルが含まれる。

20

保存処理部 4 は、情報処理装置 1 の起動時に、二次記憶装置 2 から RAM ディスクイメージ 8 をメモリ 1502 に読み込むマウント処理、及び情報処理装置 1 を停止する時に、RAM ディスク 7 をアンマウントし、ディスクドライバ 9 を経由して RAM ディスクイメージを、二次記憶装置 2 に保存する処理を行う。本実施例では、RAM ディスクのアンマウントに際して特徴的な処理を行う。

#### 【0019】

通常、CPU 1501 がデータ処理を行う場合、UAP 3 はファイルシステム 5 を経由して RAM ディスク 7 へアクセスする。ファイルシステム 5 は、RAM ディスク 7 に格納されたデータを解釈し、UAP 3 が RAM ディスク 7 をツリー構造のファイルシステムとして利用できるようにする。またファイルシステム 5 は、RAM ディスク 7 へアクセスする際に利用頻度が高いデータをキャッシュ 6 に蓄えることで、RAM ディスク 7 とのアクセス回数を減らす。

30

なお、情報処理装置 1 はデバイス 13 を制御するので、デバイス制御端末と呼んでもよく、その呼び名は特定されない。

#### 【0020】

図 2 は RAM ディスク 7 及びキャッシュ 6 のメモリの構成を示す。

40

RAM ディスク 7 は、ファイルシステム管理領域 71、ファイル管理領域 72、データブロック領域 73 から構成される。ファイルシステム管理領域 71 は、ディスクサイズ、空きデータブロック数、ボリューム名等の、ファイルシステム自体の管理情報を保存する領域である。ファイル管理領域 72 は、ファイルサイズ、ファイル更新日時、ファイルアクセス権等の、各ファイルの属性情報を記録する領域である。データブロック領域 73 は、各ファイルの内容を保存する領域である。

#### 【0021】

キャッシュ 6 のメモリ構造は RAM ディスク 7 と同様であり、ファイルシステム管理領域 61、ファイル管理領域 62、データブロック領域 63 から構成される。キャッシュ 6 には、実際にアクセスしたファイルに関するデータのみが蓄えられるため、ファイル管理

50

領域 6 2 とデータブロック領域 6 3 のサイズは、対応する R A M ディスク 7 の領域以下の大きさになる。ファイルシステム管理領域 6 1 は、R A M ディスク 7 のファイルシステム管理領域 7 1 をそのままコピーするため、領域サイズは等しくなる。

ここで、ファイル管理領域 7 2 及びデータブロック領域 7 3 はファイルが削除されたり更新されると無効領域が生じるので、後述するようなクリア処理が行われる。しかし、ファイルシステム管理領域 7 1 に保持されるシステム管理情報は基本的に全て有効であるので、この領域に対してはクリア処理が行われない。

#### 【 0 0 2 2 】

図 3 は R A M ディスク 7 内のファイル管理領域 7 2 のメモリ構成を示す。

ファイル管理領域 7 2 は、ビットマップ 7 4 の次に、複数のファイル管理情報 7 5 が続く構造になっている。ファイル管理情報 7 5 はファイルと 1 対 1 に対応し、属性情報 7 6 とブロックリスト 7 7 から構成される。属性情報 7 6 にはファイルサイズ、ファイル更新日時、ファイルアクセス権など各ファイルの属性情報を記録する。ブロックリスト 7 7 には、ファイルの内容が記録されているデータブロック 7 9 のアドレスが保持される。ファイルサイズが増加すると、1 つのファイルは通常複数のデータブロック 7 9 を使用することになるので、ブロックリスト 7 7 は複数のデータブロックの位置を記録可能な構成となっている。

10

#### 【 0 0 2 3 】

ビットマップ 7 4 は、ファイル管理領域 7 2 内のファイル管理情報 7 5 の各エントリが有効であるか無効であるかを示すビットマップである。すなわち、ビットマップ 7 4 の N 番目のビットが 1 (有効) であれば、ファイル管理情報 7 5 内の N 番目のエントリが有効である。例えばファイルを削除すれば、削除したファイルに対応するファイル管理情報 7 5 が無効になるので、ファイルシステム 5 はビットマップ 7 4 の対応するビットは 1 (有効) から 0 (無効) に変更される。

20

#### 【 0 0 2 4 】

図 4 は R A M ディスク 7 内のデータブロック領域 7 3 のメモリ構成を示す。

データブロック領域 7 3 は、ビットマップ 7 8 の次に、複数のデータブロック 7 9 が続く構成になっている。データブロック 7 9 にはファイルの内容が記録され、ファイル管理情報 7 5 内のブロックリスト 7 7 から参照される。

ビットマップ 7 8 は、データブロック 7 9 の各エントリが有効であるか無効であるかを示すビットマップである。すなわちビットマップ 7 8 の N 番目のビットが 1 (有効) であれば、データブロック領域 7 3 内の N 番目のエントリが有効である。例えばファイルが削除されたり、ファイルサイズが縮小されると、ファイルが利用していたデータブロックが無効になるので、ファイルシステム 5 はビットマップ 7 8 の対応するビットは 1 (有効) から 0 (無効) に変更される。

30

#### 【 0 0 2 5 】

以下、情報処理装置 1 が停止する時に、保存処理部 4 が R A M ディスク 7 の R A M ディスクイメージを二次記憶装置 2 へ保存する手順について説明する。ファイルシステム 5 はファイルサイズ縮小やファイル削除を行う際、ファイル管理領域 7 2 内のビットマップ 7 4 や、データブロック領域 7 3 内のビットマップ 7 8 のビットを 0 (無効) にするだけであり、ファイル管理情報 7 5 やデータブロック 7 9 のクリアは行わない。従って、R A M ディスク 7 の無効領域 (無効となったファイル管理情報 7 5 やデータブロック 7 9) には、以前使用していたデータがそのまま残されており、R A M ディスク 7 をそのまま圧縮すると無効領域に残されたデータも圧縮するため、圧縮率が低下する。そこで、保存処理部 4 はシステム停止時に R A M ディスクの内容をチェックし、無効領域を 0 クリア処理する。0 クリアしたデータの圧縮率は非常によいため、これによって R A M ディスク 7 全体の圧縮率が向上し、圧縮処理時間が短縮される。

40

#### 【 0 0 2 6 】

次に、図 5 のフローを参照して、この情報処理システムを停止する時の保存処理部 4 による動作について説明する。

50

情報処理装置 1 の停止処理が開始されると、保存処理部 4 はファイルシステム 5 に、RAM ディスク 7 のアンマウントを要求する。ファイルシステム 5 はキャッシュ 6 内に格納されたデータのうち、RAM ディスク 7 に書き戻す必要があるものを書き戻した後、RAM ディスクをアンマウントする (S 1 0)。

以下の処理動作において、保存処理部 4 は RAM ディスク 7 へアクセスするが、このとき保存処理部 4 はファイルシステム 5 を経由せずに直接 RAM ディスク 7 へアクセスする。例えば UNIX 系システムの場合、RAM ディスクに対応するデバイスファイルへアクセスすることで RAM ディスク 7 への直接アクセスする。

#### 【 0 0 2 7 】

保存処理部 4 は、RAM ディスク 7 のファイル管理領域 7 2 内のビットマップ 7 4 に記録されたビットを順にチェックする (S 1 1)。そしてビットが 0 であれば対応するファイル管理情報 7 5 が無効であるので、そのファイル管理情報 7 5 のエントリを 0 クリア処理する。すなわち、ビットマップ 7 4 の N 番目のビットが 0 (無効) であれば、N 番目のファイル管理情報 7 5 のエントリが使用するメモリ領域に 0 ' を書き込む。ビットが 1 であれば、対応するファイル管理情報 7 5 は有効であるので、0 クリア処理を行わない (S 1 2、S 1 3)。保存処理部 4 は、ビットマップ 7 4 の最終ビットの処理を終えるまで S 1 1 ~ S 1 3 の処理を繰り返す (S 1 4)。

#### 【 0 0 2 8 】

続いて、保存処理部 4 は、RAM ディスク 7 のデータブロック領域 7 3 内のビットマップ 7 8 に記録されたビットを順にチェックし (S 1 5)、ビットが 0 であれば、対応するデータブロック 7 9 が無効であるので、そのデータブロック 7 9 のエントリを 0 クリア処理する。すなわちビットマップ 7 8 の N 番目のビットが 0 (無効) であれば、N 番目のデータブロック 7 9 が使用するメモリ領域を 0 ' を書き込む。ビットが 1 であれば、対応するデータブロック 7 9 は有効であるので、0 クリア処理を行わない (S 1 6、S 1 7)。保存処理部 4 は、ビットマップ 7 8 の最終ビットの処理を終えるまで S 1 5 ~ S 1 7 の処理を繰り返す。(S 1 8)

以上の処理で RAM ディスク 7 内の無効領域は全て 0 クリアされる。最後に保存処理部 4 は、RAM ディスク 7 のデータを圧縮処理し、RAM ディスクイメージ 8 として二次記憶装置 2 に保存する (S 1 9)。以上で、保存処理部 4 による RAM ディスク 7 の保存処理が完了する。

#### 【 0 0 2 9 】

なお、本実施例では、無効領域に所定のデータとして 0 ' を書き込むことによりクリア処理を行っているが、所定のデータは単純なパターンであれば 0 ' に限らず何でも良く、例えば - 1 ' を書き込んでよい。以下の実施例でも同様である。

また、上記例では、ファイル管理領域 7 2 とデータブロック領域 7 3 の両方の無効領域に対して 0 クリア処理を行っているが、処理時間削減のためにいずれか一方のみを実施してもよい。また、各無効領域に対する 0 クリア処理を全部行わないで、部分的に実施してもよい。

また、上記例では、通常システム停止時に行う処理としているが、バッテリー残量不足、システム障害等の原因により、システムの緊急停止が必要な場合に行ってもよい。

#### 【 0 0 3 0 】

また、上記例において、ステップ S 1 0 と S 1 1 の間で、RAM ディスク 7 をデフラグすることで、S 1 9 における RAM ディスク 7 の圧縮率が更に向上する。デフラグ処理は RAM ディスク 7 内のフラグメンテーションを解消する処理であり、これによってディスク領域の無効領域が連結されるため、RAM ディスク 7 の圧縮率が向上する。但し、デフラグ処理は一般に時間がかかるので、好適には情報処理装置 1 の CPU 1 5 0 1 のプロセッサ性能がよいか、もしくはシステム停止にある程度時間をかけてもよい場合に限って行う。ここで、上記デフラグ処理は既存のツールを利用することができる。

なお、本実施例において、キャッシュ機能に対応した既存のファイルシステムを使用すると、RAM ディスクの場合もキャッシュが作成されることがあるが、RAM を記録媒体と

10

20

30

40

50

して利用する R A M ディスクではキャッシュは不要としてもよい。

【 0 0 3 1 】

〔実施例 2〕

実施例 1 では、R A M ディスクの無効領域 0 クリア処理をシステム停止時に行っているの  
で、システム停止処理に時間がかかる可能性がある。実施例 2 では、システム動作時に R  
A M ディスクの無効領域を分割して 0 クリア処理することで、システム停止時における R  
A M ディスクの無効領域 0 クリア処理が不要又は減少となる。これによって、システム停  
止処理を高速化することができる。

【 0 0 3 2 】

図 6 は本実施例における情報処理装置 1 の構成を示す。

実施例 1 と相違する点は、ディスクチェッカ 1 4 が追加されたことである。本実施例にお  
ける情報処理装置 1 の動作は実施例 1 と実質的に同じであるので、以下、相違点について  
説明する。

ディスクチェッカ 1 4 は、情報処理装置 1 のシステム動作時に定期的に起動し、キャッシ  
ュ 6 を参照しながら、R A M ディスクの無効領域を数ブロックずつ 0 クリアする。従って  
、システム停止時には R A M ディスク 7 の殆どもしくは全ての無効領域が、既に 0 クリア  
されていることになる。システム停止時、保存処理部 4 は R A M ディスク 7 をアンマウン  
トした後、直ちに R A M ディスク 7 をデータ圧縮して二次記憶装置 2 に保存する。すなわ  
ち図 5 における S 1 0 の後、S 1 9 を行うことになり、S 1 1 ~ S 1 8 の処理は省略する  
。

【 0 0 3 3 】

図 7 にキャッシュ 6 のメモリ構造を示す。

キャッシュ 6 は R A M ディスク 7 の一部がコピーされるので、実質的には図 2 ~ 図 4 に示  
した R A M ディスク 7 のメモリの構成と同様である。キャッシュ 6 のファイル管理領域 6  
2、ビットマップ 6 4、ファイル管理情報 6 5 は、R A M ディスク 7 のファイル管理領域  
7 2、ビットマップ 7 4、ファイル管理情報 7 5 に対応する。同様に、キャッシュ 6 のデ  
ータブロック領域 6 3、ビットマップ 6 8、データブロック 6 9 は、R A M ディスク 7 のデ  
ータブロック領域 7 3、ビットマップ 7 8、データブロック 7 9 に対応する。

【 0 0 3 4 】

但し、R A M ディスク内のどの部分のコピーであることを示すために、ファイル管理情報  
6 5 とデータブロック 6 9 には、管理番号 6 5 a 及び管理番号 6 9 b が追加されている。フ  
ァイル管理情報 6 5 に追加された管理番号 6 5 a は、ファイル管理情報 6 5 が R A M ディ  
スク 7 内のファイル管理領域 7 2 に記録された何番目のファイル管理情報 7 5 のエントリ  
に対応するかを示す。また、データブロック 6 9 に追加された管理番号 6 9 b は、デー  
タブロック 6 9 が R A M ディスク 7 内のデータブロック領域 7 3 に記録された何番目のデ  
ータブロック 7 9 のエントリに対応するかを示す。

【 0 0 3 5 】

また、各ファイル管理情報 6 5 やデータブロック 6 9 を R A M ディスク 7 に書き戻す必  
要があるかどうかを示すために、ファイル管理情報 6 5 とデータブロック 6 9 には、Dirt  
y ビット 6 5 b 及び Dirty ビット 6 9 c が追加されている。書き戻す必要があれば、Dirty ビ  
ット 6 5 b 及び Dirty ビット 6 9 c には 1 がセットされ、書き戻す必要がなければ 0 がセッ  
トされる。尚、ファイル削除やファイルサイズ縮小等によって、ファイル管理情報 6 5 や  
データブロック 6 9 が無効となった場合、ファイルシステム 5 はこれらエントリが Dirty  
であっても書き戻さずにこれらエントリが占有するメモリを廃棄する。

【 0 0 3 6 】

ビットマップ 6 4 及びビットマップ 6 8 については、全情報が R A M ディスク 7 からキャ  
ッシュ 6 にコピーされ、R A M ディスク内のビットマップ 7 4 及びビットマップ 7 8 の  
完全な複製となっている。従って、キャッシュ 6 内のビットマップ 6 4 及びビットマップ  
6 8 を参照すれば、ファイル管理情報 6 5 やデータブロック 6 9 の有効/無効を判定する  
ことができる。なお、ファイルシステム 5 はビットマップ 6 4 及びビットマップ 6 8 に変

10

20

30

40

50

更がある度にこれらをRAMディスク7に書き戻すので、ビットマップ64とビットマップ68にはDirtyフラグが追加されていない。もちろん、書き戻し処理を効率化するために、これらにDirtyフラグを追加してもよい。

#### 【0037】

ディスクチェッカ14は定期的起動され、ビットマップ64及びビットマップ68を参照して、RAMディスク7内の無効となったファイル管理情報75及びデータブロック79を0クリアする。無効となった領域はキャッシュ6からRAMディスク7へ書き戻されることはないので、ディスクチェッカ14は、キャッシュ6内のファイル管理情報65及びデータブロック69の0クリアは行わない。

#### 【0038】

図8を参照して、RAMディスクの無効領域の0クリア処理動作について説明する。この処理は、ディスクチェッカ14が定期的に行う処理である。ディスクチェッカ14は、ファイル管理領域インデックス番号、及びデータブロック領域インデックス番号という2つの変数を内部に保持する。ファイル管理領域インデックス番号は、ファイル管理領域62のビットマップ64のチェックすべきビット番号である。データブロック領域インデックス番号は、データブロック領域63のビットマップ68のチェックすべきビット番号である。

#### 【0039】

ディスクチェッカ14が起動すると、RAMディスク7及びキャッシュ6へのアクセスをロックするようファイルシステム5に要求する(S20)。これによりUAP3からRAMディスク7へのアクセスが一時的にブロックされる。

次に、ディスクチェッカ14はキャッシュ6のファイル管理領域62内のビットマップ64についてファイル管理領域インデックス番号のビットをチェックする(S21)。もしビットが0(無効)であれば、RAMディスク7のファイル領域72内の対応するファイル管理情報75を0クリアする(S22、S23)。ディスクチェッカ14はファイル管理領域インデックス番号をインクリメントし、一定数チェックするまでS21～S23の処理を繰り返す。なお、ファイル管理領域インデックス番号がビットマップ64の最終ビット番号を超えたら0に戻す(S24、S25)。

#### 【0040】

次に、ディスクチェッカ14は、キャッシュ6のデータブロック領域63内のビットマップ68についてデータブロック領域インデックス番号のビットをチェックする(S26)。もしビットが0(無効)であれば、RAMディスク7のデータブロック領域73内の対応するデータブロック79を0クリアする(S27、S28)。ディスクチェッカ14は、データブロック領域インデックス番号をインクリメントし、一定数チェックするまでS26～S28の処理を繰り返す。なお、データブロック領域インデックス番号がビットマップ68の最終ビット番号を超えたら0に戻す(S29、S30)。

#### 【0041】

次に、ディスクチェッカは、RAMディスク7及びキャッシュ6のアクセスをロック解除するようファイルシステム5に要求する。これにより、UAP3からのRAMディスク7へのアクセスが再開される。以上で、ディスクチェッカ14によるRAMディスクの無効領域0クリア処理が終了する。

なお、上記例では、ディスクチェッカ14の処理を定期的に行っているが、ディスクチェッカ14がOSのスケジューラと連携して、システムがアイドル状態の時に、上記した処理を実行するようにしてもよい。

#### 【0042】

##### [実施例3]

実施例2では、RAMディスクの無効領域0クリア処理を定期的に行うので、システムの停止時に全ての無効領域の0クリア処理が完了していない場合がある。本実施例では、ファイル削除やファイルサイズの縮小などによってRAMディスク7内に無効領域が生成された時にファイルシステム5が無効領域を0クリアする。これにより、システム停止時に

10

20

30

40

50

R A Mディスクの全無効領域が0クリアされていることが保障され、R A Mディスクの圧縮率が向上する。

【0043】

本実施例における情報処理装置1の構成は、実施例1における場合と同様である。すなわち、システム停止時、保存処理部4はR A Mディスク7をアンマウントした後、直ちにR A Mディスク7をデータ圧縮して二次記憶装置2に保存する。

本実施例におけるファイルシステム5の動作は実施例1、2と異なる。以下、相違点を中心にファイルシステム5の処理動作を説明する。ファイルシステム5はR A Mディスク5に保存されたファイルの削除やファイルサイズを縮小する時に、R A Mディスク内に発生した無効領域を0クリアする。

10

【0044】

図9を参照して、ファイルシステム5によるR A Mディスクの無効領域の0クリア処理動作について説明する。

U A P 3がファイルシステム5に対してファイル削除もしくはファイルサイズ縮小を指示すると、ファイルシステム5は、上記ファイルに対応するファイル管理情報75内のブロックリスト77を確認し、ブロックリスト77に記録されているデータブロックのうち必要なものをブロックリスト77から削除する。すなわち、ファイル削除の場合はブロックリスト77に記載されている全てのデータブロックのアドレスを削除する。また、ファイルサイズ縮小の場合は縮小した部分に対応するデータブロックのアドレスを削除する。そしてファイルシステム5は、データブロック領域73のビットマップ78について、上記の削除したデータブロックに対応するビットを0(無効)にする。0'にするビットのビット番号は、

20

$$([\text{削除したデータブロックのアドレス}] - [\text{データブロックの開始アドレス}]) / [\text{データブロック79のサイズ}]$$
として算出する。なお、このステップの処理は一般的なファイルシステムの動作と同様である(S40)。

【0045】

次に、ファイルシステム5はS40で削除したデータブロックのリストを作成し(S41)、上記リストに記載されたデータブロック79を全て0クリアする。このステップの処理は、本実施例に特徴的な処理動作である(S42、S43)。

ファイルサイズ縮小の場合、ファイルシステム5は処理を完了する。ファイル削除の場合、削除するファイルに対応するファイル管理情報75を無効化する必要があるので、次のステップに進む(S44)。

30

【0046】

ファイルシステム5は、ファイル管理領域72のビットマップ74について、削除するファイルに対応するファイル管理情報75を示すビットを0(無効)にする。上記ビットのビット番号は、
$$([\text{ファイル管理情報75のアドレス}] - [\text{ファイル管理情報の開始アドレス}]) / [\text{ファイル管理情報75のサイズ}]$$
から算出する。なお、このステップの処理は一般的なファイルシステムの動作と同様である(S45)。

ファイルシステム5は、削除するファイルに対応するファイル管理情報75を0クリアする。このステップの処理は、本実施例に特徴的な処理動作である(S46)。以上で、ファイル削除又はファイルサイズ縮小処理が終了する。

40

【0047】

なお、上記例では、ファイルシステム5はR A Mディスク7に対して処理を行っているが、処理対象のファイル管理領域72やデータブロック領域73の一部がキャッシュ6に存在する場合は、キャッシュ6とR A Mディスク7の両方に対して処理を行う必要がある。この場合、ビットマップのビットを0(無効)にする処理(S40、S45)は、キャッシュ6とR A Mディスク7両方のビットマップに対して行うが、無効領域の0クリア処理(S41~S43、S46)は、R A Mディスク7内のデータブロック79とファイル管理情報75に対してのみ行う。キャッシュ6内のデータブロック69とファイル管理情報65については、これらのエントリが無効となった時点でファイルシステム5によって

50

廃棄されるので、0クリアする必要がない。

【実施例4】

デバイス13としての組込機器は内蔵する二次記憶装置の容量が少なく、RAMディスクイメージを格納することができない場合がある。そこで本実施例においては、情報処理装置自体はRAMディスクイメージを二次記憶装置に保持せず、複数の情報処理装置はRAMディスク利用開始時に管理サーバからRAMディスクイメージをダウンロードして使用し、その使用が終了した時に管理サーバにRAMディスクイメージをアップロードする。

【0048】

RAMディスクイメージを圧縮することで、RAMディスクイメージのアップロード/ダウンロードに伴うネットワーク負荷を低減することができる。しかしRAMディスク内の無効領域が少なくなってくると、RAMディスクイメージをデータ圧縮する効果が減少し、特に性能が低い組込機器では圧縮処理時間の増加が顕著になる。そこで本実施例では、RAMディスクの利用率に応じてRAMディスクイメージを圧縮処理する装置を切り替え、またRAMディスクの利用率が基準値以上の場合はRAMディスクイメージを圧縮しないで保存する。これによって、ネットワーク負荷と圧縮処理時間のバランスを考慮した運用が可能となる。

【0049】

図10は本実施例における情報処理システムの構成を示す。

このシステムは、管理サーバ16にネットワーク11を介して複数の情報処理装置101～10n（全てを総称して情報処理装置1ということがある）が接続して構成される。管理サーバ16は、システム内に存在する複数の情報処理装置101～10nのRAMディスクイメージ801～80n（全てを総称してRAMディスクイメージ80ということがある）を二次記憶装置20に記憶する。管理DB22は、各RAMディスクイメージ80と、情報処理装置1の識別子の対応関係を保持しており、管理プログラム17は管理DB22と連携してRAMディスクイメージ80を管理する。情報処理装置1の保存処理部4は、情報処理装置1の起動時に管理サーバ16から外部ネットワーク11を介してRAMディスクイメージをダウンロードし、情報処理装置の停止時に管理サーバ16へRAMディスクイメージをアップロードする。情報処理装置1の識別子は、情報処理装置1を識別することができる情報であれば何でもよく、例えば情報処理装置1に付与されたネットワークインターフェイスのMACアドレスを利用してもよい。なお、図10では情報処理装置に接続されるデバイスの図示が省略されている。

【0050】

次に、図11を参照して、管理サーバ16から情報処理装置1へRAMディスクイメージをダウンロードする処理動作について説明する。

例えば情報処理装置101が起動されると、保存処理部4は管理サーバ16へ情報処理装置1の識別子を含む要求メッセージを送信して、管理サーバ16が保持するRAMディスクイメージを要求する。この要求メッセージを受けた管理プログラム17は、要求メッセージに含まれる識別子に対応するRAMディスクイメージを管理DB22から検索して、そのRAMディスクイメージ801を情報処理装置101へ送信する（S50）。

【0051】

管理プログラム17は、RAMディスクイメージT3の送信開始時にRAMディスクイメージT3が圧縮されているかどうかを判別し、圧縮されているかどうかを示す圧縮情報を管理端末1に送信する。続いて管理プログラム17は、RAMディスクイメージT3を管理端末1に送信する。保存処理部4は圧縮情報を受信することで、受信するRAMディスクイメージが圧縮されているかどうかを判定する（S51）。

【0052】

保存処理部4はRAMディスクイメージをダウンロードする処理と、RAMディスク7の作成をパイプライン的に行う。すなわち、RAMディスクイメージが圧縮されていなければ、保存処理部4はRAMディスクイメージのダウンロードとRAMディスク7へのコ

10

20

30

40

50

ピーを数バイトずつ繰り返して行う（S52）。一方、RAMディスクイメージが圧縮されていれば、保存処理部4はRAMディスクイメージのダウンロード、圧縮の展開、RAMディスク7へのコピーを数バイトずつ繰り返して行う（S53）。もちろん、情報処理装置1に十分な空きメモリが存在する場合や、情報処理装置1が十分な容量を持つ二次記憶装置1503を備える場合は、RAMディスクイメージ全体をいったんメモリ又は二次記憶装置1503にダウンロードした後、RAMディスク7を作成してもよい。

#### 【0053】

RAMディスクイメージのダウンロードが完了すると、保存処理部4はRAMディスク7をマウントする（S54）。この例では、圧縮情報とRAMディスクイメージを送信しているが、上記データに加えて管理サーバ16内で算出したRAMディスクイメージのハッシュ値を管理サーバ16から情報処理装置1へ送信し、情報処理装置1は受信したディスクイメージのハッシュ値を別途算出し、これら2つのハッシュ値を比較し、両者が一致しない場合にダウンロード処理をやり直す構成にしてもよい。これによって通信障害等によるデータ改変を防止することができる。

10

#### 【0054】

次に図12を参照して、情報処理装置1から管理サーバ16へRAMディスクイメージをアップロードする処理について説明する。

情報処理装置1が停止処理を開始すると、保存処理部4はRAMディスク7をアンマウントし、RAMディスク7の利用率を求める。保存処理部4はファイル管理領域72内のビットマップ74をチェックし、0であるビットの個数を「無効であるファイル管理情報75の個数」とみなす。また、データブロック領域73内のビットマップ78をチェックし、0であるビットの個数を「無効であるデータブロック79の個数」とみなす。そして、無効領域の容量を、

20

$[\text{無効であるファイル管理情報75の個数}] \times [\text{ファイル管理情報75のサイズ}] + [\text{無効であるデータブロック79の個数}] \times [\text{データブロック79のサイズ}]$ 、として算出する。

RAMディスク7の利用率は、 $1 - [\text{無効領域の容量}] / [\text{RAMディスクの総容量}]$ で算出できる。

なお、この例では、無効であるファイル管理情報75の個数と、無効であるデータブロック79の個数をビットマップ74及びビットマップ78をチェックすることで算出しているが、ファイルシステム管理領域71にこれらの情報を保持しておき、その情報を用いてもよい（S60）。

30

#### 【0055】

RAMディスク7の利用率が算出されると、次に、RAMディスク7の利用率に応じて、RAMディスクイメージの圧縮処理を切替える処理を行う。すなわち、求めたRAMディスク7の利用率と所定の基準値（例えば50%）と比較し（S60）、その結果、利用率が基準値よりも小さければ、保存処理部4はS11～S18で説明した手順で無効領域を0クリアし（S62）、その0クリアされたRAMディスクイメージを圧縮処理した後、管理サーバ16へ送信する。この時、保存処理部4はRAMディスクイメージが圧縮されていることを示す圧縮情報と情報処理装置1の識別子を送信した後、圧縮したRAMディスクイメージを送信する。なお、好ましくはRAMディスクイメージの圧縮と送信処理はパイプライン的に行うのがよい。すなわち、RAMディスクイメージの圧縮処理と送信処理を数バイトずつ行う。これによって、情報処理装置1の空きメモリが少ない場合でも容量が大きなRAMディスクイメージを管理サーバ16に送信することができる（S63）。

40

#### 【0056】

一方、RAMディスク7の利用率が基準値（50%）以上であれば（S60）、保存処理部4はRAMディスクイメージが圧縮されていないことを示す圧縮情報と情報処理装置1の識別子を管理サーバ16へ送信した後、RAMディスクイメージを圧縮処理しないでそのまま送信する（ステップ64）。なお、S63、S64の処理では、RAMディスクイメージのダウンロード処理において説明した処理と同様に、情報処理装置1内で算出し

50

た R A M ディスクイメージのハッシュ値を管理サーバ 1 6 に送信し、上記ハッシュ値と、管理サーバ 1 6 内で別途算出したハッシュ値を比較し、両者が一致しない場合にアップロードをやり直すように処理してもよい。

【 0 0 5 7 】

次に、図 1 3 を参照して、管理サーバ 1 6 が情報処理装置 1 から R A M ディスクイメージを受信する処理動作について説明する。

図 1 0 に示すシステムを引用すると、管理サーバ 1 6 の管理プログラム 1 7 は、情報処理装置 1 から受信した R A M ディスクイメージが圧縮されていない場合、R A M ディスクの利用率が基準値（例えば 8 0 %）よりも小さければ R A M ディスクイメージを圧縮処理して二次記憶装置 1 5 0 3 に保存する。好ましくは、管理サーバ 1 6 が圧縮を行う基準値を、情報処理装置 1 が圧縮を行う基準値よりも大きくすることで、圧縮率が高く圧縮処理時間が短い場合は情報処理装置 1 内で圧縮処理し、逆に圧縮率が低く圧縮処理時間が長い場合は管理サーバ 1 6 内で圧縮処理するようにする。これによって、ネットワーク負荷と圧縮処理時間のバランスを考慮した運用が可能となる。

【 0 0 5 8 】

管理サーバ 1 6 の管理プログラム 1 7 は、情報処理装置 1 から圧縮情報と情報処理装置 1 の識別子を受信する。管理プログラム 1 7 は、圧縮情報から R A M ディスクイメージが圧縮されているかどうかを判定する。また、管理プログラム 1 7 は管理 D B 2 2 にアクセスして情報処理装置 1 の識別子から、保存すべき R A M ディスクイメージ 8 を求める。続いて、管理プログラム 1 7 は R A M ディスクイメージを受信する（ S 7 0 ）。

【 0 0 5 9 】

もし R A M ディスクイメージが圧縮されていれば、受信した R A M ディスクイメージを S 7 0 で求めた R A M ディスクイメージ 8 0 i に保存して処理を終了する（ S 7 1、 S 7 6 ）。もし R A M ディスクイメージが圧縮されていなければ、管理プログラム 1 7 は S 6 0 と同様の手順で、R A M ディスクイメージの利用率を求める（ S 7 2 ）。利用率が基準値（例えば 8 0 %）よりも小さければ、 S 6 2 と同様の手順で、R A M ディスクの無効領域を 0 クリアし（ S 7 4 ）、R A M ディスクをデータ圧縮した後、 S 7 0 で求めた R A M ディスクイメージ T 3 に保存して処理を終了する（ S 7 5、 S 7 6 ）。

一方、利用率が基準値（ 8 0 %）以上であれば、R A M ディスクイメージをデータ圧縮せずに、 S 7 0 で求めた R A M ディスクイメージ T 3 に保存して処理を終了する（ S 7 3、 S 7 6 ）。

【 0 0 6 0 】

次に、図 1 4 を参照して、R A M ディスクイメージのメンテナンス処理動作について説明する。

図 1 0 のシステムでは、例えば、管理サーバ 1 6 内の R A M ディスクイメージ 8 0 1 ~ 8 0 n を一括更新することでシステムのメンテナンスを行う。メンテナンス時に、管理プログラム 1 7 は更新が必要な R A M ディスクイメージ T 3 を順に自身のファイルシステムにマウントし、R A M ディスクイメージ T 3 内のファイルを更新していく。この処理では、R A M ディスクイメージ T 3 内のファイル作成 / 削除 / 更新を行うので、R A M ディスク内に新たに無効領域が生成される。そこで、管理プログラム 1 7 は、更新処理が終了した後、R A M ディスクイメージ 8 0 1 ~ 8 0 n をチェックして、無効領域の 0 クリアを行ってから圧縮処理して二次記憶装置に保存する。これにより R A M ディスクイメージの圧縮率低下を抑止する。

【 0 0 6 1 】

図 1 4 を参照するに、システムメンテナンスが開始されると、管理プログラム 1 7 は管理者の指示に従い、更新が必要な R A M ディスクイメージ 8 0 i を選択する（ S 8 0 ）。上記指示は、管理者が管理サーバ 1 6 のコンソール画面から逐一行ってもよいし、あらかじめ作成したスクリプト等に従って処理を自動的に行ってもよい。

【 0 0 6 2 】

S 8 0 で選択された R A M ディスクイメージ 8 0 i が圧縮されていれば、管理プログラ

10

20

30

40

50

ム 17 は R A M ディスクイメージ 8 0 i を展開し ( S 8 1、S 8 2 )、展開した後の R A M ディスクイメージ 8 0 i を自身のファイルシステムにマウントする ( S 8 3 )。R A M ディスクイメージ 8 0 i が圧縮されていなければ、管理プログラム 17 は R A M ディスクイメージ 8 0 i をそのまま自身のファイルシステムにマウントする ( S 8 1、S 8 3 )。

【 0 0 6 3 】

次に、管理プログラム 17 は R A M ディスクイメージ 8 0 i 内のファイルシステムにアクセスし、R A M ディスクイメージ 8 0 i のメンテナンスを行う。この処理ステップでは、アプリケーションのインストール/アンインストール/アップデート、設定ファイルやレジストリ情報の整合性チェック、コンピュータウイルス等ユーザが意図しないプログラムの有無のチェック等を行う ( S 8 4 )。

10

【 0 0 6 4 】

次に、管理プログラム 17 は R A M ディスクイメージ 8 0 i をアンマウントし ( S 8 5 )、S 6 0 と同様の手順で、R A M ディスクイメージの利用率を求める ( S 8 6 )。利用率が基準値 ( 8 0 % ) よりも小さければ、S 6 2 と同様の手順で、R A M ディスクの無効領域を 0 クリアし ( S 8 8 )、R A M ディスクイメージを圧縮処理した後、S 8 0 で選択した R A M ディスクイメージ 8 0 i に保存して処理を終了する ( S 8 9、S 9 0 )。一方、利用率が基準値 ( 8 0 % ) 以上であれば、R A M ディスクイメージを圧縮処理しないで、S 8 0 で求めた R A M ディスクイメージ 8 0 i に保存して処理を終了する ( S 8 7、S 9 0 )。

20

【 図面の簡単な説明 】

【 0 0 6 5 】

【 図 1 】 実施例 1 における情報処理システムの構成図。

【 図 2 】 実施例 1 における R A M ディスク及びキャッシュの構成を示す図。

【 図 3 】 実施例 1 における R A M ディスク内のファイル管理領域 7 2 の構成を示す図。

【 図 4 】 実施例 1 における R A M ディスク内のデータブロック領域 7 3 の構成を示す図。

【 図 5 】 実施例 1 における R A M ディスクの圧縮保存処理動作を示すフロー図。

【 図 6 】 実施例 2 における情報処理システムの構成図。

【 図 7 】 キャッシュ内のファイル管理領域及びデータブロックの構造を示す図。

【 図 8 】 実施例 2 における R A M ディスクの無効領域の 0 クリア処理動作を示すフロー図。

30

【 図 9 】 実施例 3 における R A M ディスクの無効領域の 0 クリア処理動作を示すフロー図。

【 図 1 0 】 実施例 4 における情報処理システムの構成図。

【 図 1 1 】 実施例 4 における R A M ディスクイメージのダウンロード処理動作を示すフロー図。

【 図 1 2 】 実施例 4 における R A M ディスクイメージのアップロード処理動作を示すフロー図。

【 図 1 3 】 実施例 4 による管理サーバ 1 6 における R A M ディスクイメージの受信処理動作を示すフロー図。

【 図 1 4 】 実施例 4 による管理サーバ 1 6 における R A M ディスクイメージのメンテナンス処理動作を示すフロー図。

40

【 図 1 5 】 実施例 1 における情報処理装置 1 のハードウェア構成図。

【 符号の説明 】

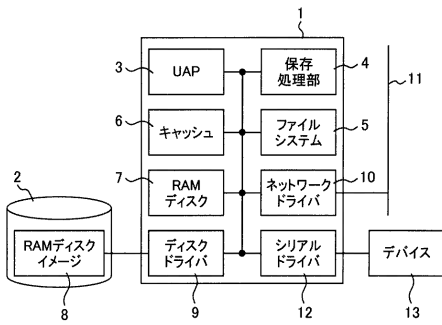
【 0 0 6 6 】

1 : 情報処理装置 2、2 0、1 5 0 3 : 二次記憶装置 3 : ユーザアプリケーション ( U A P ) 4 : 保存処理部 5 : ファイルシステム 6 : キャッシュ 7 : R A M ディスク 8、8 0 1 ~ 8 0 n : R A M ディスクイメージ 1 3 : デバイス 1 4 : ディスクチェック

1 6 : 管理サーバ 2 0 : 二次記憶装置 2 2 : 管理 D B 1 7 : 管理プログラム

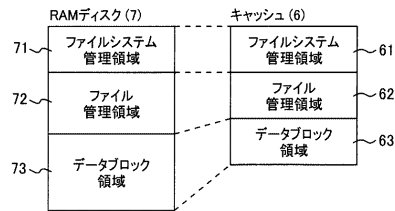
【 図 1 】

図 1



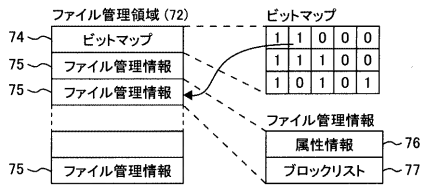
【 図 2 】

図 2



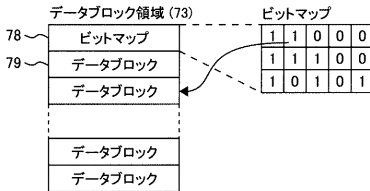
【 図 3 】

図 3



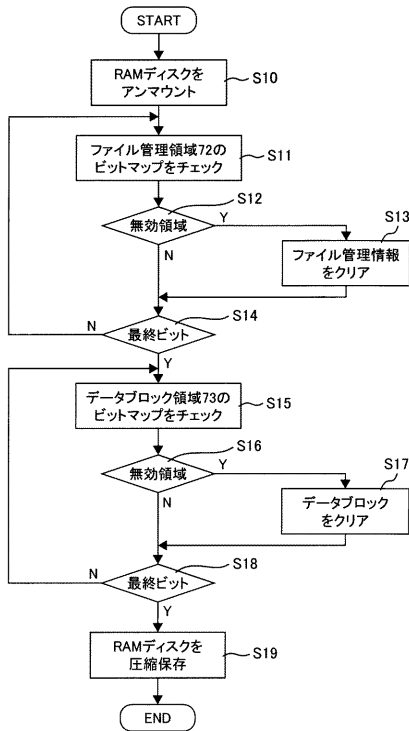
【 図 4 】

図 4



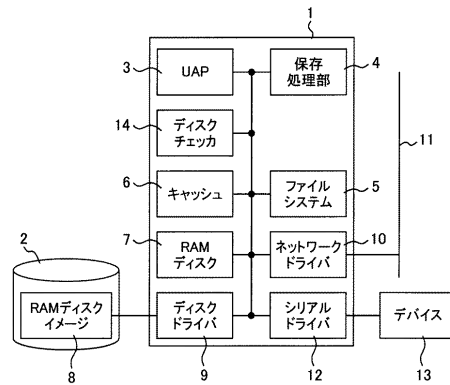
【 図 5 】

図 5



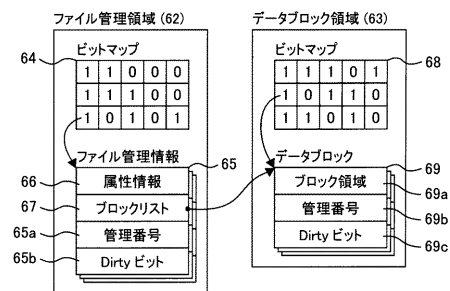
【 図 6 】

図 6

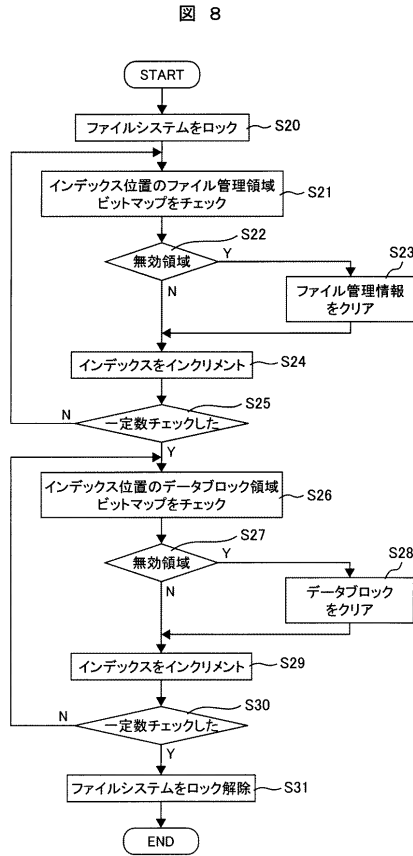


【 図 7 】

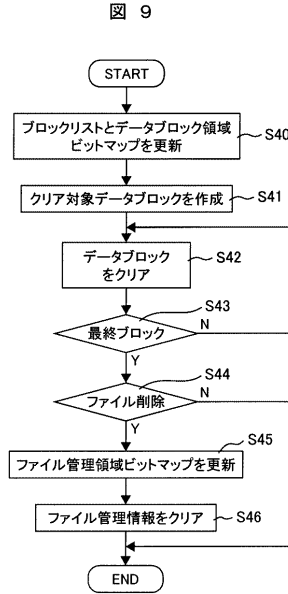
図 7



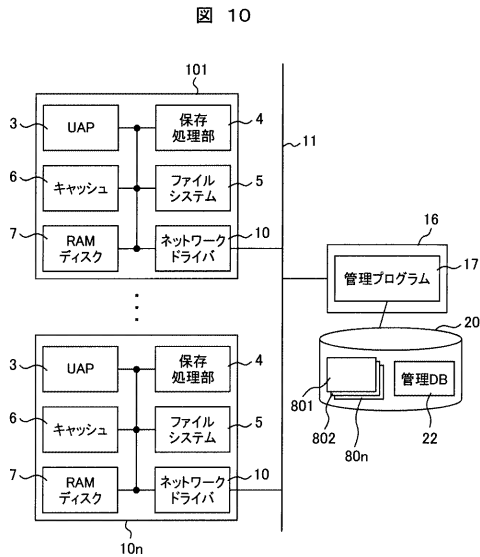
【 図 8 】



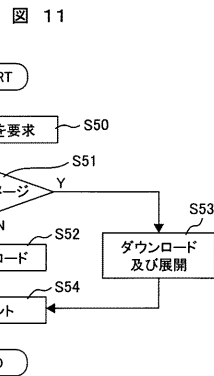
【 図 9 】



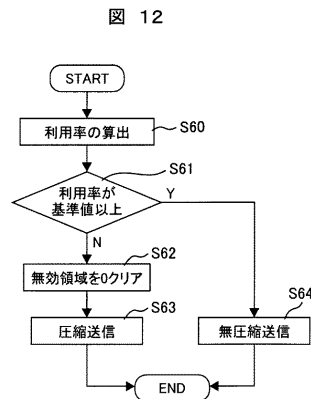
【 図 10 】



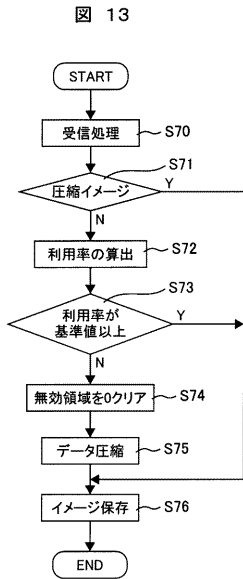
【 図 11 】



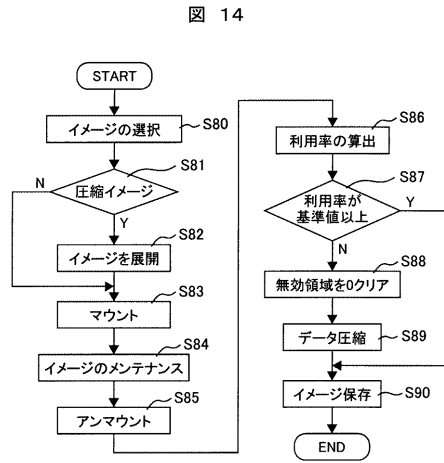
【 図 12 】



【 図 1 3 】



【 図 1 4 】



【 図 1 5 】

