

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7348752号
(P7348752)

(45)発行日 令和5年9月21日(2023.9.21)

(24)登録日 令和5年9月12日(2023.9.12)

(51)国際特許分類	F I
G 0 6 F 16/18 (2019.01)	G 0 6 F 16/18 1 0 0
G 0 6 F 16/13 (2019.01)	G 0 6 F 16/13 1 1 0
G 0 6 F 3/08 (2006.01)	G 0 6 F 3/08 H
G 0 6 F 3/06 (2006.01)	G 0 6 F 3/06 3 0 1 K

請求項の数 8 (全12頁)

(21)出願番号	特願2019-102121(P2019-102121)	(73)特許権者	310021766
(22)出願日	令和1年5月31日(2019.5.31)		株式会社ソニー・インタラクティブエン
(65)公開番号	特開2020-197787(P2020-197787		タテインメント
	A)		東京都港区港南1丁目7番1号
(43)公開日	令和2年12月10日(2020.12.10)	(74)代理人	100105924
審査請求日	令和3年7月26日(2021.7.26)		弁理士 森下 賢樹
		(74)代理人	100109047
			弁理士 村田 雄祐
		(74)代理人	100109081
			弁理士 三木 友由
		(74)代理人	100134256
			弁理士 青木 武司
		(72)発明者	沖野 直人
			東京都港区港南1丁目7番1号 株式会
			社ソニー・インタラクティブエンタテイ
			最終頁に続く

(54)【発明の名称】 情報処理装置

(57)【特許請求の範囲】

【請求項1】

情報処理装置であって、
アプリケーションが生成したファイルを複数のデータブロックに分割して記憶装置に記憶するブロック処理部と、
前記複数のデータブロックのそれぞれの前記記憶装置における記憶場所を特定するための情報および前記複数のデータブロックのそれぞれのハッシュ値を含む第1メタデータと、
前記第1メタデータの前記記憶装置における記憶場所を特定するための情報を含む第1スーパーブロックと、前記第1スーパーブロックにアクセスするための第1参照ポイントとを前記記憶装置に記憶するメタデータ処理部と、
前記第1参照ポイントを選択することで、ファイルアクセスに、前記第1スーパーブロックおよび前記第1メタデータを使用することを設定するメタデータ切替部と、を備え、
ファイルが更新された際に、前記ブロック処理部は、更新があったデータブロックを前記記憶装置における別の記憶場所に記憶し、
前記メタデータ処理部は、更新があったデータブロックを含む前記複数のデータブロックのそれぞれの前記記憶装置における記憶場所を特定するための情報および更新があったデータブロックを含む前記複数のデータブロックのそれぞれのハッシュ値を含む第2メタデータと、前記第2メタデータの前記記憶装置における記憶場所を特定するための情報を含む第2スーパーブロックと、前記第2スーパーブロックにアクセスするための第2参照ポイントとを、前記記憶装置における前記第1メタデータ、前記第1スーパーブロック、

および前記第 1 参照ポインタの記憶場所とは別の記憶場所に記憶し、

前記メタデータ切替部は、前記第 2 参照ポインタを選択することで、ファイルアクセスに、前記第 2 スーパーブロックおよび前記第 2 メタデータを使用することを設定する、
ことを特徴とする情報処理装置。

【請求項 2】

前記メタデータ処理部は、更新があったデータブロックと、既に記憶されており且つ更新がなかったデータブロックのそれぞれの前記記憶装置における記憶場所を特定するための情報および更新があったデータブロックと、既に記憶されており且つ更新がなかったデータブロックのそれぞれのハッシュ値を含む前記第 2 メタデータを前記記憶装置に記憶する、

10

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 3】

前記メタデータ切替部が、前記第 2 参照ポインタを選択して、ファイルアクセスに、前記第 2 スーパーブロックおよび前記第 2 メタデータを使用することを設定した後、前記ブロック処理部は、更新前のデータブロックを前記記憶装置から削除し、前記メタデータ処理部は、前記第 1 メタデータを前記記憶装置から削除する、

ことを特徴とする請求項 1 または 2 に記載の情報処理装置。

【請求項 4】

前記アプリケーションがファイルの更新指示と、更新の反映を許可する許可指示をこの順に出力すると、許可指示の前の更新指示が実施されたことを条件として、前記メタデータ切替部が、メタデータの切り替え処理を実行する、

20

ことを特徴とする請求項 1 から 3 のいずれかに記載の情報処理装置。

【請求項 5】

前記アプリケーションが許可指示を出力するまで、前記ブロック処理部は、更新指示されたファイルをバッファメモリに記憶し、

前記アプリケーションが許可指示を出力すると、前記ブロック処理部は、前記バッファメモリに記憶したファイルの前記記憶装置への書込を開始する、

ことを特徴とする請求項 4 に記載の情報処理装置。

【請求項 6】

前記記憶装置は、SSDである、

30

ことを特徴とする請求項 1 から 5 のいずれかに記載の情報処理装置。

【請求項 7】

前記ファイルは、ゲームのセーブデータである、

ことを特徴とする請求項 1 から 6 のいずれかに記載の情報処理装置。

【請求項 8】

コンピュータに、

アプリケーションが生成したファイルを複数のデータブロックに分割して記憶装置に記憶する機能と、

前記複数のデータブロックのそれぞれの前記記憶装置における記憶場所を特定するための情報および前記複数のデータブロックのそれぞれのハッシュ値を含む第 1 メタデータと、

40

前記第 1 メタデータの前記記憶装置における記憶場所を特定するための情報を含む第 1 スーパーブロックと、前記第 1 スーパーブロックにアクセスするための第 1 参照ポインタとを前記記憶装置に記憶する機能と、

前記第 1 参照ポインタを選択することで、ファイルアクセスに、前記第 1 スーパーブロックおよび前記第 1 メタデータを使用することを設定する機能と、

ファイルが更新された際に、更新があったデータブロックを前記記憶装置における別の記憶場所に記憶する機能と、

更新があったデータブロックを含む前記複数のデータブロックのそれぞれの前記記憶装置における記憶場所を特定するための情報および更新があったデータブロックを含む前記複数のデータブロックのそれぞれのハッシュ値を含む第 2 メタデータと、前記第 2 メタデ

50

ータの前記記憶装置における記憶場所を特定するための情報を含む第2スーパーブロックと、前記第2スーパーブロックにアクセスするための第2参照ポイントとを、前記記憶装置における前記第1メタデータ、前記第1スーパーブロック、および前記第1参照ポイントの記憶場所とは別の記憶場所に記憶する機能と、

前記第2参照ポイントを選択することで、ファイルアクセスに、前記第2スーパーブロックおよび前記第2メタデータを使用することを設定する機能と、

を実現させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ファイルを管理する技術に関する。

【背景技術】

【0002】

ゲームなどのアプリケーションは、現在の状態を示す各種データを記憶装置に保存するセーブ機能をもつ。ファイルシステムは、セーブデータを記憶装置に記憶する際、同時にファイルの記憶場所や改竄チェック用のハッシュ値など様々なメタデータも記憶装置に記憶する。

【発明の概要】

【発明が解決しようとする課題】

【0003】

メタデータの種別が多くなり、データサイズが大きくなると、メタデータの更新に、ある程度の時間が必要となる。メタデータの更新中に電源断などのトラブルが発生すると、メタデータ同士の関係が不整合となってセーブデータを消失する可能性がある。そこでファイルシステムにおいて、メタデータを不整合を生じさせることなく更新する技術の開発が望まれている。

【課題を解決するための手段】

【0004】

上記課題を解決するために、本発明のある態様の情報処理装置は、アプリケーションが生成したファイルを複数のデータブロックに分割して記憶装置に記憶するブロック処理部と、複数のデータブロックのそれぞれの記憶場所を特定するための情報およびハッシュ値を含む第1メタデータを記憶装置に記憶するメタデータ処理部と、ファイルアクセスに第1メタデータを使用することを設定するメタデータ切替部とを備える。ファイルが更新された際に、ブロック処理部は、更新があったデータブロックを別の記憶場所に記憶し、メタデータ処理部は、更新があったデータブロックを含む複数のデータブロックのそれぞれの記憶場所を特定するための情報およびハッシュ値を含む第2メタデータを記憶装置に記憶し、メタデータ切替部は、ファイルアクセスに第2メタデータを使用することを設定する。

【0005】

なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、記録媒体、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【0006】

本発明の情報処理技術によると、メタデータを不整合を生じさせることなく更新することが可能となる。

【図面の簡単な説明】

【0007】

【図1】本発明の実施例にかかる情報処理システムを示す図である。

【図2】情報処理装置のハードウェア構成を示す図である。

【図3】メタデータ更新機能を実現するための機能ブロックを示す図である。

10

20

30

40

50

【図４】セーブデータ領域の記憶状態を模式的に示す図である。

【図５】メタデータを変更した状態を示す図である。

【図６】更新前のデータブロックおよびメタデータを削除した状態を示す図である。

【発明を実施するための形態】

【０００８】

図１は、本発明の実施例にかかる情報処理システム１を示す。

情報処理装置１０は、ユーザが操作する入力装置６と無線または有線で接続し、入力装置６はユーザの操作結果を示す操作情報を情報処理装置１０に出力する。情報処理装置１０は入力装置６から操作情報を受け付けるとＯＳ（システムソフトウェア）やアプリケーションの処理に反映し、出力装置４から処理結果を出力させる。情報処理システム１において情報処理装置１０はゲームプログラムを実行するゲーム装置であり、入力装置６はゲームコントローラなど情報処理装置１０に対してユーザの操作情報を供給する機器であってよい。入力装置６は複数のプッシュ式の操作ボタンや、アナログ量を入力できるアナログスティック、回動式ボタンなどの複数の入力部を有して構成される。

【０００９】

補助記憶装置２はＳＳＤ（ソリッドステートドライブ）などの大容量記憶装置であり、内蔵型記憶装置であってよく、またＵＳＢ（Universal Serial Bus）などによって情報処理装置１０と接続する外部記憶装置であってもよい。出力装置４は画像を出力するディスプレイおよび音声出力するスピーカを有するテレビであってよく、またヘッドマウントディスプレイであってもよい。

【００１０】

アクセスポイント（以下、「ＡＰ」とよぶ）８は、無線アクセスポイントおよびルータの機能を有し、情報処理装置１０は、無線または有線経由でＡＰ８に接続して、インターネットなどの外部ネットワーク上のサーバと通信可能に接続できる。

【００１１】

図２は、情報処理装置１０のハードウェア構成を示す。情報処理装置１０は、メイン電源ボタン２０、電源ＯＮ用ＬＥＤ２１、スタンバイ用ＬＥＤ２２、システムコントローラ２４、クロック２６、デバイスコントローラ３０、メディアドライブ３２、ＵＳＢモジュール３４、フラッシュメモリ３６、無線通信モジュール３８、有線通信モジュール４０、サブシステム５０およびメインシステム６０を有して構成される。

【００１２】

メインシステム６０は、メインＣＰＵ（Central Processing Unit）、主記憶装置であるメモリおよびメモリコントローラ、ＧＰＵ（Graphics Processing Unit）などを備える。ＧＰＵはゲームプログラムの演算処理に主として利用される。これらの機能はシステムオンチップとして構成されて、１つのチップ上に形成されてよい。メインＣＰＵは補助記憶装置２に記録されたゲームプログラムを実行する機能をもつ。

【００１３】

サブシステム５０は、サブＣＰＵ、主記憶装置であるメモリおよびメモリコントローラなどを備え、ＧＰＵを備えず、ゲームプログラムを実行する機能をもたない。サブＣＰＵの回路ゲート数は、メインＣＰＵの回路ゲート数よりも少なく、サブＣＰＵの動作消費電力は、メインＣＰＵの動作消費電力よりも少ない。サブＣＰＵは、メインＣＰＵがスタンバイ状態にある間においても動作し、消費電力を低く抑えるべく、その処理機能を制限されている。

【００１４】

メイン電源ボタン２０は、ユーザからの操作入力が行われる入力部であって、情報処理装置１０の筐体の前面に設けられ、情報処理装置１０のメインシステム６０への電源供給をオンまたはオフするために操作される。電源ＯＮ用ＬＥＤ２１は、メイン電源ボタン２０がオンされたときに点灯し、スタンバイ用ＬＥＤ２２は、メイン電源ボタン２０がオフされたときに点灯する。

【００１５】

10

20

30

40

50

システムコントローラ 24 は、ユーザによるメイン電源ボタン 20 の押下を検出する。メイン電源がオフ状態にあるときにメイン電源ボタン 20 が押下されると、システムコントローラ 24 は、その押下操作を「オン指示」として取得し、一方で、メイン電源がオン状態にあるときにメイン電源ボタン 20 が押下されると、システムコントローラ 24 は、その押下操作を「オフ指示」として取得する。

【0016】

クロック 26 はリアルタイムクロックであって、現在の日時情報を生成し、システムコントローラ 24 やサブシステム 50 およびメインシステム 60 に供給する。デバイスコントローラ 30 は、サウスブリッジのようにデバイス間の情報の受け渡しを実行する LSI (Large-Scale Integrated Circuit) として構成される。図示のように、デバイスコントローラ 30 には、システムコントローラ 24、メディアドライブ 32、USB モジュール 34、フラッシュメモリ 36、無線通信モジュール 38、有線通信モジュール 40、サブシステム 50 およびメインシステム 60 などのデバイスが接続される。デバイスコントローラ 30 は、それぞれのデバイスの電気特性の違いやデータ転送速度の差を吸収し、データ転送のタイミングを制御する。

【0017】

メディアドライブ 32 は、ゲームなどのソフトウェア、およびライセンス情報を記録した ROM 媒体 44 を装着して駆動し、ROM 媒体 44 からプログラムやデータなどを読み出すドライブ装置である。ROM 媒体 44 は、光ディスクや光磁気ディスク、ブルーレイディスクなどの読出専用の記録メディアである。

【0018】

USB モジュール 34 は、外部機器と USB ケーブルで接続するモジュールである。USB モジュール 34 は補助記憶装置 2 と USB ケーブルで接続してもよい。フラッシュメモリ 36 は内部ストレージを構成する補助記憶装置である。無線通信モジュール 38 は、Bluetooth (登録商標) プロトコルや IEEE802.11 プロトコルなどの通信プロトコルで、たとえば入力装置 6 と無線通信する。なお無線通信モジュール 38 は、デジタル携帯電話方式に対応してもよい。有線通信モジュール 40 は、外部機器と有線通信し、たとえば AP 8 を介して外部のネットワークに接続する。

【0019】

図 3 は、情報処理装置 10 のメタデータ更新機能を実現するための機能ブロックを示す。処理部 100 はメインシステム 60 に構成され、実行部 102 およびファイルシステム 110 を備える。ファイルシステム 110 は、ブロック処理部 112、メタデータ処理部 114、メタデータ切替部 116 およびファイル読出部 118 を備える。

【0020】

処理部 100 の構成は、ハードウェアコンポーネントでいえば、任意のコンピュータの CPU、メモリ、メモリにロードされたプログラム、ストレージなどによって実現されるが、ここではそれらの連携によって実現される機能ブロックを描いている。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは、当業者には理解されるところである。

【0021】

実行部 102 はゲームプログラムを実行して、出力装置 4 からゲーム画像およびゲーム音声を出力する。ゲームプログラムの実行中、セーブデータの保存指示が、ゲームプログラムからファイルシステム 110 に供給されると、ファイルシステム 110 は、セーブデータを補助記憶装置 2 に保存する。

【0022】

ブロック処理部 112 は、ゲームプログラムが生成したセーブデータファイルを複数のデータブロックに分割して補助記憶装置 2 に記憶する。実施例のゲームプログラムは、複数のセーブデータファイル A ~ Z を生成するが、以下、ファイルシステム 110 が、ゲームプログラムから供給されるファイル A を記憶し、更新する処理について説明する。

【0023】

ブロック処理部 112 は、ファイル A を複数のデータブロック A1、A2、A3、A4 に分割して、補助記憶装置 2 に記憶する。補助記憶装置 2 には、予め、セーブデータを格納するためのセーブデータ領域が確保されており、ブロック処理部 112 は、空いている記憶場所を各データブロックに割り当て、各データブロックを記憶する。

【0024】

メタデータ処理部 114 は、データブロックのハッシュ値を算出する機能を有し、各データブロックのハッシュ値を算出する。メタデータ処理部 114 は、複数のデータブロックのそれぞれの記憶場所を特定するための情報（ロケーション情報）およびハッシュ値を含むメタデータをセーブデータ領域に記憶する。メタデータ処理部 114 は、各データブロックのロケーション情報およびハッシュ値をテーブル（ファイル A）の形式で管理してよい。ロケーション情報は、データブロックの記憶場所を特定できる情報であればよく、代表的には SSD におけるページの相対位置を指定するページ番号であり、ページ番号を算出するためのデータサイズやブロックオフセットなどの情報であってもよい。

10

【0025】

メタデータ処理部 114 は、ファイル A 用のテーブルのロケーション情報およびハッシュ値を含むメタデータをセーブデータ領域に記憶する。このメタデータは、スーパーブロックの形式で管理されてよい。なお図 3 には、ファイル A 用のテーブルおよびファイル A のデータブロック A1 ~ A4 を示しているが、他のファイル B ~ Z 用のテーブルおよびデータブロックも同様にセーブデータ領域に記憶されている。したがってスーパーブロックは、全てのファイル A ~ Z にアクセスするための情報を含んで構成されている。

20

【0026】

メタデータ処理部 114 は、スーパーブロックのロケーション情報およびハッシュ値を含むメタデータをセーブデータ領域に記憶する。このメタデータは、スーパーブロックにアクセスするための参照ポインタであり、ファイルシステム 110 が参照ポインタにアクセスすることで、スーパーブロックを介して、全てのファイル A ~ Z にアクセスすることが可能となる。

【0027】

メタデータ切替部 116 は、ファイルアクセスに使用するメタデータを設定する。メタデータ切替部 116 は、参照ポインタを選択することで、スーパーブロックおよびテーブル（ファイル A）を含むメタデータを、ファイルアクセスに使用するメタデータに設定する。この状態でファイルシステム 110 が、ゲームプログラムから、セーブデータの読出指示を受けると、ファイル読出部 118 が、メタデータ切替部 116 により選択されている参照ポインタを参照して、指示されたセーブデータを読み出す。

30

【0028】

次に、ファイル A がゲームプログラムによりファイル A' に更新されて、更新されたファイル A' をセーブデータ領域に記憶する処理について説明する。ブロック処理部 112 は、ファイル A' を複数のデータブロック A1、A2'、A3、A4 に分割する。この場合、更新されたファイル A' と更新前のファイル A とを比較すると、データブロック A2 がデータブロック A2' に変更されており、それ以外のデータブロック A1、A3、A4 に更新はない。そこでブロック処理部 112 は、更新があったデータブロック A2' を、空いている別の記憶場所に記憶する。

40

【0029】

図 4 は、ファイル A がファイル A' に更新されたときのセーブデータ領域の記憶状態を模式的に示す。図 3 を参照して、図 4 における「スーパーブロック 1」、「参照ポインタ 1」は、図 3 に示す「スーパーブロック」、「参照ポインタ」を表現している。

【0030】

まずブロック処理部 112 が、更新があったデータブロック A2' を、空いている記憶場所に記憶する。なおデータブロック A1、A3、A4 については更新がなかったため、ブロック処理部 112 は、データブロック A1、A3、A4 を空いている記憶場所に書き込まない。

50

【 0 0 3 1 】

メタデータ処理部 1 1 4 は、更新があったデータブロック A 2 ' を含む複数のデータブロックのそれぞれの記憶場所を特定するための情報（ロケーション情報）およびハッシュ値を含むメタデータを記憶装置のセーブデータ領域に記憶する。図 4 において、メタデータ処理部 1 1 4 は、各データブロックのロケーション情報およびハッシュ値を、テーブル（ファイル A ' ）としてセーブデータ領域に記憶する。図示されるように、テーブル（ファイル A ' ）は、更新があったデータブロック A 2 ' と、既にセーブデータ領域に記憶されており且つ更新がなかったデータブロック A 1、A 3、A 4 のそれぞれの記憶場所を特定するための情報およびハッシュ値を含むメタデータを含む。

【 0 0 3 2 】

メタデータ処理部 1 1 4 は、ファイル A ' 用のテーブルのロケーション情報およびハッシュ値を含むメタデータをセーブデータ領域に記憶する。このメタデータは、スーパーブロックの形式で管理されてよい。メタデータ処理部 1 1 4 は、ファイル A '、ファイル B ~ Z 用のテーブルのロケーション情報およびハッシュ値を含むメタデータを、「スーパーブロック 2」としてセーブデータ領域に記憶する。

【 0 0 3 3 】

メタデータ処理部 1 1 4 は、スーパーブロック 2 のロケーション情報およびハッシュ値を含むメタデータをセーブデータ領域に記憶する。このメタデータは、スーパーブロック 2 にアクセスするための参照ポインタ 2 であり、ファイルシステム 1 1 0 が参照ポインタ 2 にアクセスすることで、スーパーブロック 2 を介して、ファイル A '、B ~ Z にアクセスすることが可能となる。

【 0 0 3 4 】

従来のファイル更新処理では、データブロック A 2 ' を書き込む際に、同時にデータブロック A 2 を削除し、またファイル A ' に関するメタデータを書き込む際に、同時にファイル A に関するメタデータを削除するファイル管理が行われている。そのため複数種類のメタデータを書き込んでいる途中で電源断などによりシステムダウンすると、更新されたメタデータと更新されなかったメタデータが混在して、システム復帰後に、メタデータの整合性がとれなくなる事態が発生しうる。この場合ファイルシステムは、新たに書き込んだセーブデータにも、その前に書き込んでいたセーブデータにもアクセスできず、結果としてセーブデータを消失する。

【 0 0 3 5 】

実施例のファイルシステム 1 1 0 によれば、更新があったデータブロック、および対応するメタデータを書き込むときに、既に記憶されているデータブロックおよびメタデータを削除しない。そのため更新があったデータブロックおよびメタデータの書込中に、電源断などのトラブルが発生した場合であっても、少なくとも整合性のとれる記憶済みのデータブロックおよびメタデータがセーブデータ領域に残っているため、完全にセーブデータが消失する事態を回避できる。

【 0 0 3 6 】

メタデータ切替部 1 1 6 は、更新があったデータブロックおよび対応するメタデータの書込が終了すると、ファイルアクセスに使用するメタデータを、スーパーブロック 2 およびテーブル（ファイル A ' ）を含むメタデータに設定する。

【 0 0 3 7 】

図 5 は、ファイルアクセスに使用するメタデータが、スーパーブロック 2 およびテーブル（ファイル A ' ）を含むメタデータに変更された状態を示す。メタデータ切替部 1 1 6 は、参照ポインタ 2 を選択することで、スーパーブロック 2 およびテーブル（ファイル A ' ）を含むメタデータを、ファイルアクセスに使用するメタデータとして設定する。この状態でファイルシステム 1 1 0 が、ゲームプログラムから、セーブデータの読出指示を受けると、ファイル読出部 1 1 8 が、メタデータ切替部 1 1 6 により選択されている参照ポインタ 2 を参照して、指示されたセーブデータを読み出す。

【 0 0 3 8 】

10

20

30

40

50

メタデータ切替部 116 が、ファイルアクセスに、スーパーブロック 2 およびテーブル（ファイル A'）を含むメタデータを使用することを設定した後、ブロック処理部 112 は、更新前のデータブロック A2 をセーブデータ領域から削除し、メタデータ処理部 114 は、更新前のメタデータをセーブデータ領域から削除する。

【0039】

図 6 は、更新前のデータブロックおよびメタデータを削除した状態を示す。メタデータ切替部 116 が参照ポインタを切り換えた後に、更新前のデータブロックおよびメタデータを削除することで、予期せぬシステムダウン時にセーブデータが消失する可能性を低減できる。なお補助記憶装置 2 が SSD である場合、データの消去は、複数のページが集まったブロックを単位として実施されるため、データブロックやテーブルなどは、SSD におけるブロックを単位として設定されてよい。

10

【0040】

以上、ファイル A の更新処理について説明したが、セーブデータでは、複数のファイルが連携していることがある。たとえば、ファイル B が、セーブ時のゲームステージを示すデータであり、ファイル C が、セーブ時にプレイヤーキャラクタが保有している武器の情報を示すデータであるとする。ステージごとに特別な武器を入手できるシナリオをもつゲームでは、セーブデータにおいて、ゲームステージの情報と、保有武器の情報とに矛盾が生じてはいけな。そこでセーブデータの更新処理において、ファイル B とファイル C は、同時に確実に更新されることが必要であり、ファイル B は更新できたが、ファイル C は更新できなかったという事態は回避しなければならない。

20

【0041】

そこでメタデータ切替部 116 は、ゲームプログラムから更新の反映を許可する許可指示を受け取することを契機として、メタデータの切り替え処理を実行することが好ましい。以下、ファイル B、C が連携している場合について説明する。

【0042】

ゲームプログラムは、ファイル B、C の更新指示と、更新の反映を許可する許可指示とを、この順にファイルシステム 110 に供給する。ゲームプログラムからの指示は、ファイルシステム 110 において指示の順番にしたがって実施され、ファイルシステム 110 は、ファイル B の更新指示にしたがってファイル B を更新し、ファイル C の更新指示にしたがってファイル C を更新し、その後、許可指示にしたがってコミットして、更新されたファイル B、C をアクセス可能な状態にする。

30

【0043】

具体的に、ファイルシステム 110 はこの一連の指示を受けると、ブロック処理部 112 が、ファイル B において更新があったデータブロックを空いている記憶場所に記憶し、メタデータ処理部 114 が、更新されたファイル B に対応するメタデータを空いている記憶場所に記憶する。続いてブロック処理部 112 は、ファイル C において更新があったデータブロックを空いている記憶場所に記憶し、メタデータ処理部 114 は、更新されたファイル C に対応するメタデータを空いている記憶場所に記憶する。それからメタデータ切替部 116 が、新たな参照ポインタを選択することで、更新されたメタデータをファイルアクセスに使用することを設定する。

40

【0044】

つまりメタデータ切替部 116 は、許可指示の供給前に供給されたファイル B、C の更新指示が実施されたことを条件として、メタデータの切り替え処理を実行する。この処理は、ファイル更新にトランザクション管理の概念を導入したものであり、ファイル B と C とが矛盾なく更新されることを実現する。

【0045】

なおゲームプログラムは、一連の指示をファイルシステム 110 に出力するが、ブロック処理部 112 は、ゲームプログラムが許可指示を出力するまで、更新指示されたファイルを補助記憶装置 2 に書き込むのではなく、バッファメモリ（図示せず）に記憶してもよい。ブロック処理部 112 は、アプリケーションが許可指示を出力すると、バッファメモ

50

りに記憶したファイルの補助記憶装置 2 への書込を開始する。補助記憶装置 2 が S S D である場合、S S D はデータの並列書込を可能とするため、バッファメモリにキャッシュした複数のファイルを並列に S S D に書き込むことで、書き込み時間を短縮できる。このとき、複数のファイルを 1 つずつ S S D に書き込む場合と比較すると、メタデータの書込回数を減らすことができるため、S S D の動作寿命にも好適である。なおキャッシュするデータサイズがバッファメモリのサイズを超える場合には、その前に、ブロック処理部 1 1 2 が、古いファイルから S S D に書き込む必要がある。

【 0 0 4 6 】

以上、本発明を実施例をもとに説明した。この実施例は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。たとえばファイルシステム 1 1 0 は、許可指示を受けたときに、古いメタデータを削除せずに残しておくことで、複数世代のセーブデータを残しておくことができる。また実施例では、セーブデータファイルの例について説明したが、ファイルの種類はそれに限るものではない。

【符号の説明】

【 0 0 4 7 】

1・・・情報処理システム、2・・・補助記憶装置、10・・・情報処理装置、100・・・処理部、102・・・実行部、110・・・ファイルシステム、112・・・ブロック処理部、114・・・メタデータ処理部、116・・・メタデータ切替部、118・・・ファイル読出部。

10

20

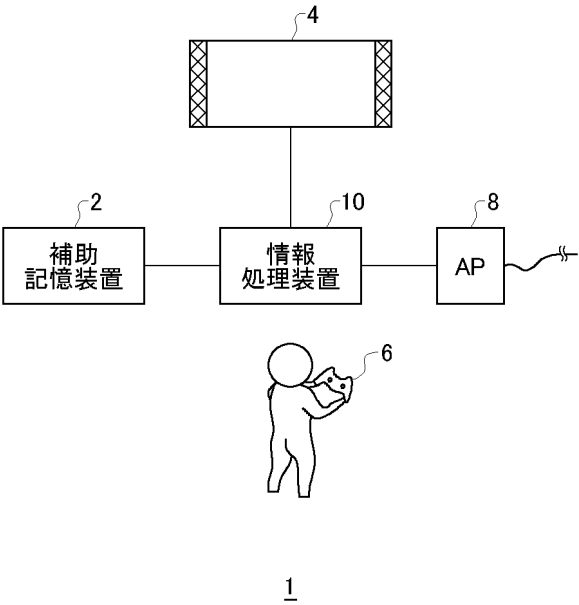
30

40

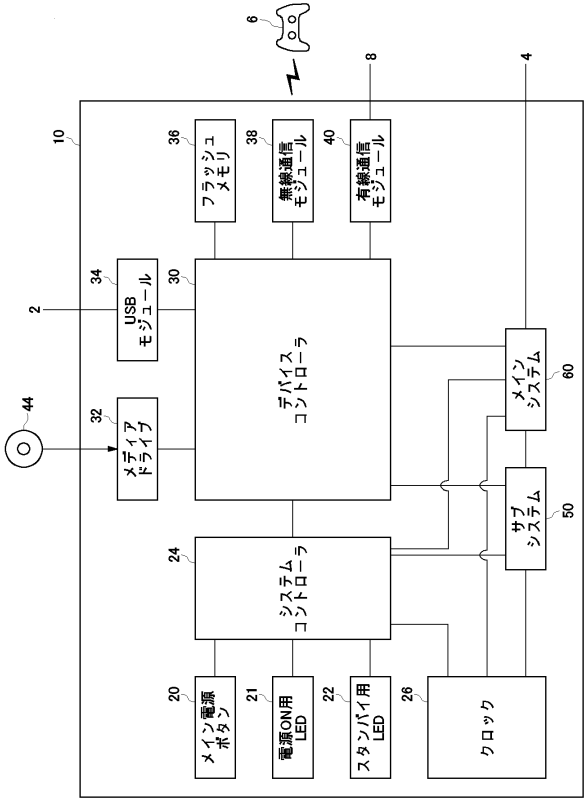
50

【図面】

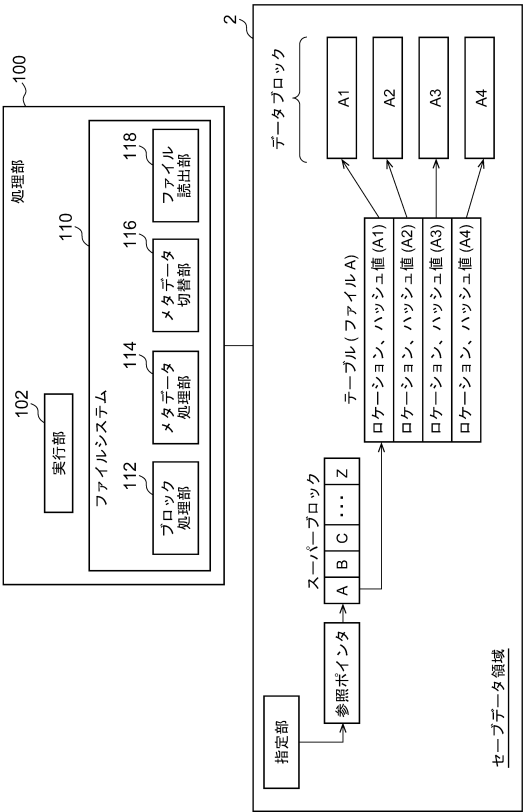
【図 1】



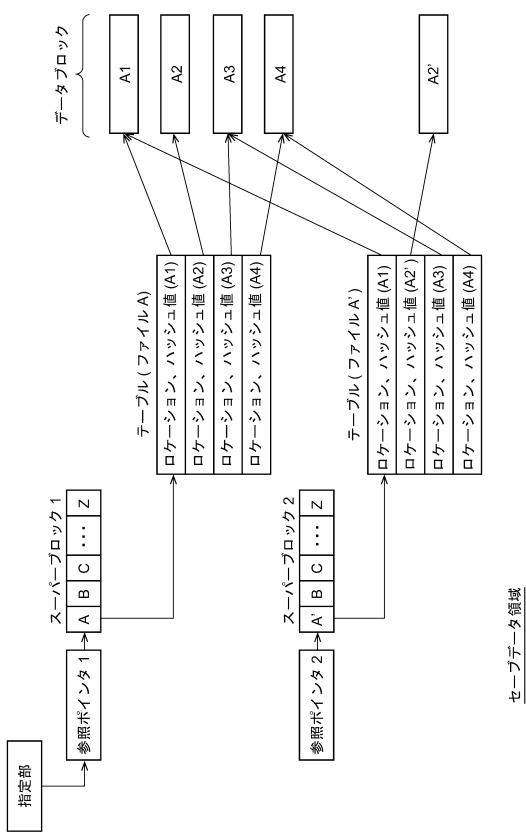
【図 2】



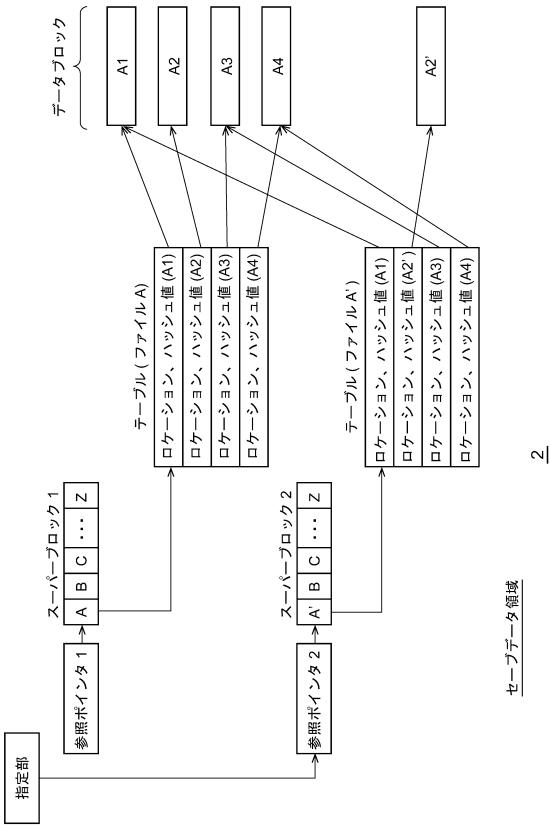
【図 3】



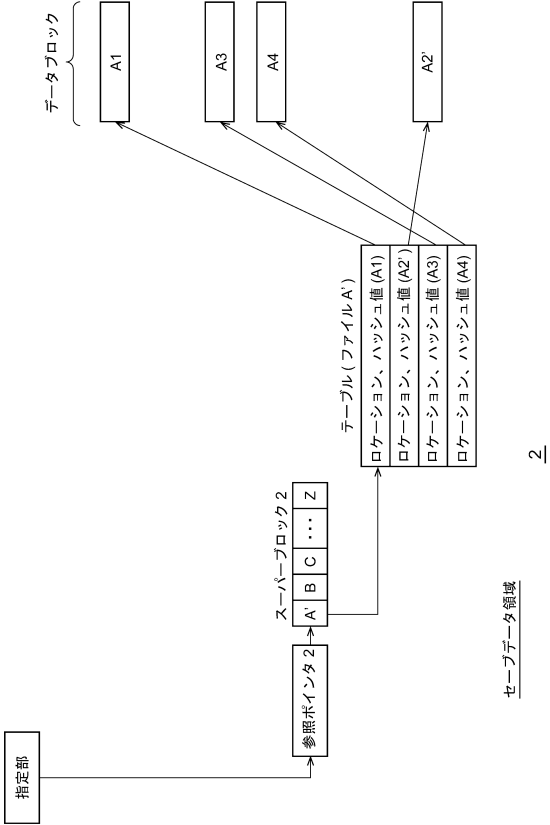
【図 4】



【図 5】



【図 6】



10

20

30

40

50

フロントページの続き

ンメント内

審査官 酒井 恭信

- (56)参考文献 特開 2 0 1 5 - 2 0 7 1 4 5 (J P , A)
米国特許出願公開第 2 0 1 5 / 0 3 0 1 8 2 3 (U S , A 1)
特開 2 0 1 7 - 0 2 7 2 0 3 (J P , A)
特開 2 0 1 5 - 0 8 8 1 4 3 (J P , A)
米国特許出願公開第 2 0 1 5 / 0 1 2 6 2 8 4 (U S , A 1)
特開 2 0 1 3 - 2 2 2 3 7 3 (J P , A)
- (58)調査した分野 (Int.Cl. , D B 名)
G 0 6 F 1 6 / 1 0 - 1 6 / 1 8 8
G 0 6 F 3 / 0 6
G 0 6 F 3 / 0 8