

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5614348号
(P5614348)

(45) 発行日 平成26年10月29日(2014.10.29)

(24) 登録日 平成26年9月19日(2014.9.19)

(51) Int.Cl. F I
G06F 9/45 (2006.01) G O 6 F 9/44 3 2 O C
 G O 6 F 9/44 3 2 2 L

請求項の数 5 (全 22 頁)

(21) 出願番号	特願2011-61324 (P2011-61324)	(73) 特許権者	000005223 富士通株式会社
(22) 出願日	平成23年3月18日 (2011.3.18)		神奈川県川崎市中原区上小田中4丁目1番1号
(65) 公開番号	特開2012-198657 (P2012-198657A)	(74) 代理人	100070150 弁理士 伊東 忠彦
(43) 公開日	平成24年10月18日 (2012.10.18)		100146776 弁理士 山口 昭則
審査請求日	平成25年11月29日 (2013.11.29)	(72) 発明者	前田 宗則 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	小沢 年弘 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 命令処理方法、命令処理装置、及び命令処理プログラム

(57) 【特許請求の範囲】

【請求項1】

実行対象プログラムに含まれる命令のうち、逐次解釈による実行回数が閾値以上となった命令の翻訳コードを生成して第一の記憶部に記憶し、

所定のタイミングで前記翻訳コードの一部又は全部を前記第一の記憶部より破棄する処理をコンピュータが実行し、

前記生成する処理は、前記翻訳コードが破棄されてから所定期間において、前記閾値を前記実行回数に対して相対的に低下させることを特徴とする命令処理方法。

【請求項2】

前記破棄する処理は、破棄対象の前記翻訳コードの中で、当該翻訳コードの生成後に前記実行対象プログラムにおいて当該翻訳コードに対応する命令が更新されていない翻訳コードを示す情報を第二の記憶部に記憶し、

前記生成する処理は、前記第二の記憶部が記憶する情報が示す前記翻訳コードに関して、破棄されてからの所定期間において前記閾値を低下させる請求項1記載の命令処理方法。

【請求項3】

前記破棄する処理は、前記第一の記憶部が前記翻訳コードに関連付けて記憶する該翻訳コードに対応する命令の識別情報を退避し、

前記生成する処理は、退避されている前記識別情報に係る命令に関して、破棄されてか

らの所定期間において前記閾値を低下させる請求項 1 又は 2 記載の命令処理方法。

【請求項 4】

実行対象プログラムに含まれる命令のうち、逐次解釈による実行回数が閾値以上となった命令の翻訳コードを生成して第一の記憶部に記憶する生成部と、

所定のタイミングで前記翻訳コードの一部又は全部を前記第一の記憶部より破棄する破棄部とを有し、

前記生成部は、前記翻訳コードが破棄されてから所定期間において、前記閾値を前記実行回数に対して相対的に低下させる命令処理装置。

【請求項 5】

実行対象プログラムに含まれる命令のうち、逐次解釈による実行回数が閾値以上となった命令の翻訳コードを生成して第一の記憶部に記憶し、

所定のタイミングで前記翻訳コードの一部又は全部を前記第一の記憶部より破棄する処理をコンピュータに実行させ、

前記生成する処理は、前記翻訳コードが破棄されてから所定期間において、前記閾値を前記実行回数に対して相対的に低下させる命令処理プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、命令処理方法、命令処理装置、及び命令処理プログラムに関する。

【背景技術】

【0002】

或る CPU の動作を異なるアーキテクチャの CPU でエミュレーションするために、インタプリタとジャストインタイムコンパイラ (Just - In - Time コンパイラ: 以下、「JIT コンパイラ」という。) とが用いられる場合がある。インタプリタは、プログラムの命令を逐次解釈しながら、各命令の実行頻度を監視する。インタプリタは、実行頻度の高い命令について、機械語への翻訳要求を JIT コンパイラに入力する。JIT コンパイラは、翻訳要求に係る命令を機械語に翻訳 (コンパイル) し、翻訳されたコード (以下、「翻訳コード」という。) をコードキャッシュと呼ばれるメモリ領域に記録する。以後、当該命令の実行時には、コードキャッシュに記録された翻訳コードが実行される。その結果、プログラムの実行速度が高速化される。

【0003】

従来、インタプリタは、命令毎に実行回数を計測し、実行回数が既定の閾値を超えた場合に翻訳要求を行っていた。JIT コンパイラが多くの翻訳要求を受けている場合に、閾値を大きくすることも行われている。例えば、インタプリタからの翻訳要求がキューによって管理され、キューの長さに応じて、閾値の増減が図られている。

【0004】

コードキャッシュに対して所定量以上の翻訳コードが記録された場合、新たな翻訳コードの記録に備えて、既存の翻訳コードの一部又は全部について、コードキャッシュからの破棄が行われる。コードキャッシュにおける翻訳コードの破棄 (すなわち、コードキャッシュの空き領域の確保) に関する処理は、一般的に、ガーベージコレクション (Garbage Collection: GC) と呼ばれる。

【先行技術文献】

【特許文献】

【0005】

【特許文献 1】特表 2003 - 526135 号公報

【特許文献 2】特表 2002 - 519752 号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

GC 実行後における命令実行時には、実行対象の命令に対する翻訳コードが存在しない

10

20

30

40

50

可能性が高くなる。その結果、GC実行前には翻訳コードを用いて高速実行されていた命令に関して、逐次解釈が行われ、性能が低下してしまう可能性がある。

【0007】

GCによってコードキャッシュから破棄された命令が、GC実行前の程度まで性能が復帰するには、当該命令の実行回数が改めて上記閾値以上となるまでの時間が必要とされた。

【0008】

1側面では、翻訳コードの破棄による性能の低下期間を短縮することのできる命令処理方法、命令処理装置、及び命令処理プログラムの提供を目的とする。

【課題を解決するための手段】

【0009】

一案では、命令処理方法は、実行対象プログラムに含まれる命令のうち、逐次解釈による実行回数が閾値以上となった命令の翻訳コードを生成して第一の記憶部に記憶し、所定のタイミングで前記翻訳コードの一部又は全部を前記第一の記憶部より破棄する処理をコンピュータが実行し、前記生成する処理は、前記翻訳コードが破棄されてから所定期間において、前記閾値を前記実行回数に対して相対的に低下させる。

【発明の効果】

【0010】

一態様によれば、翻訳コードの破棄による性能の低下期間を短縮することができる。

【図面の簡単な説明】

【0011】

【図1】本発明の実施の形態における情報処理装置のハードウェア構成例を示す図である。

【図2】第一の実施の形態における情報処理装置の機能構成例を示す図である。

【図3】第一の実施の形態の情報処理装置が実行する処理手順の一例を説明するための図である。

【図4】第一の実施の形態の管理テーブルの構成例を示す図である。

【図5】第一の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。

【図6】第二の実施の形態の情報処理装置が実行する処理手順の一例を説明するための図である。

【図7】第二の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。

【図8】翻訳コードに不整合が発生する状況の一例を説明するための図である。

【図9】第三の実施の形態の管理テーブルの構成例を示す図である。

【図10】第三の実施の形態における情報処理装置の機能構成例を示す図である。

【図11】強制破棄フラグ記憶部の構成例を示す図である。

【図12】第三の実施の形態のGCの実行処理の処理手順の一例を説明するためのフローチャートである。

【図13】第三の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。

【図14】第四の実施の形態における情報処理装置の機能構成例を示す図である。

【図15】第四の実施の形態の翻訳コード破棄部の初期化処理の処理手順の一例を示す図である。

【図16】第四の実施の形態のGCの実行処理の処理手順の一例を説明するためのフローチャートである。

【図17】第四の実施の形態における翻訳部の処理手順の一例を説明するためのフローチャートである。

【図18】第四の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。

10

20

30

40

50

【図19】第五の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。

【発明を実施するための形態】

【0012】

以下、図面に基づいて本発明の実施の形態を説明する。図1は、本発明の実施の形態における情報処理装置のハードウェア構成例を示す図である。図1の情報処理装置10は、それぞれバスBで相互に接続されているドライブ装置100、補助記憶装置102、メモリ装置103、CPU104、及びインタフェース装置105等を有する。

【0013】

情報処理装置10での処理を実現するプログラムは、記録媒体101によって提供される。プログラムを記録した記録媒体101がドライブ装置100にセットされると、プログラムが記録媒体101からドライブ装置100を介して補助記憶装置102にインストールされる。但し、プログラムのインストールは必ずしも記録媒体101より行う必要はなく、ネットワークを介して他のコンピュータよりダウンロードするようにしてもよい。補助記憶装置102は、インストールされたプログラムを格納すると共に、必要なファイルやデータ等を格納する。

10

【0014】

メモリ装置103は、プログラムの起動指示があった場合に、補助記憶装置102からプログラムを読み出して格納する。CPU104は、メモリ装置103に格納されたプログラムに従って情報処理装置10に係る機能を実行する。インタフェース装置105は、

20

【0015】

なお、記録媒体101の一例としては、CD-ROM、DVDディスク、又はUSBメモリ等の可搬型の記録媒体が挙げられる。また、補助記憶装置102の一例としては、HDD(Hard Disk Drive)又はフラッシュメモリ等が挙げられる。記録媒体101及び補助記憶装置102のいずれについても、コンピュータ読み取り可能な記録媒体に相当する。

【0016】

図2は、第一の実施の形態における情報処理装置の機能構成例を示す図である。同図において、情報処理装置10は、逐次解釈部11、翻訳要求キュー12、翻訳部13、コードキャッシュ部14、及び翻訳コード破棄部15等を有する。逐次解釈部11、翻訳部13、及び翻訳コード破棄部15は、情報処理装置10にインストールされたプログラムがCPU104に実行させる処理によって実現される。翻訳要求キュー12、コードキャッシュ部14は、例えば、メモリ装置103を用いて実現可能である。

30

【0017】

逐次解釈部11は、実行対象プログラムTpのソースコード又は中間的な表現(中間コード)の各命令を逐次的に解釈して実行する。実行対象プログラムTpは、本実施の形態において実行対象とされるプログラムが、メモリ装置103にロードされた状態のデータである。逐次解釈部11は、実行対象プログラムTpが含む命令ごとに、逐次解釈による当該命令の実行回数(すなわち、逐次解釈の実行回数)を計測する。逐次解釈部11は、

40

実行回数が閾値以上となった命令が発生した場合、当該命令のアドレス(すなわち、実行対象プログラムTp内の位置情報)を指定して、翻訳要求を翻訳要求キュー12に入力する。翻訳要求キュー12は、翻訳要求をFIFO(First-In First-Out)のリスト構造で保持する。

【0018】

なお、厳密には、実行回数は、実行対象プログラムTp内の各分岐命令の分岐先ごとに計測される。分岐命令間に含まれる命令群の実行回数は一致するため、命令ごとに実行回数を計測する必要はないからである。したがって、翻訳要求には、分岐先のアドレスが指定される。

50

【0019】

翻訳部 13 は、翻訳要求キュー 12 より翻訳要求を取り出し、取り出された翻訳要求に指定されたアドレスに係る命令群を機械語によるネイティブコードに翻訳する。例えば、翻訳要求に指定されたアドレスに係る命令から次の分岐命令までの命令群（命令が一つの場合も含む。）が翻訳される。以下、翻訳部 13 によって翻訳された（又は生成された）コードを「翻訳コード」という。翻訳部 13 は、翻訳コードをコードキャッシュ部 14 に記憶する。

【 0020】

コードキャッシュ部 14 は、翻訳コードや、翻訳コードの管理情報を記憶するための管理テーブル等を記憶する。

【 0021】

翻訳コード破棄部 15 は、コードキャッシュ部 14 に関して、ガーベージコレクション（GC）を実行する。すなわち、翻訳コード破棄部 15 は、コードキャッシュ部 14 の空き領域を確保するため、コードキャッシュ部 14 が記憶する一部又は全部の翻訳コード及び一部又は全部の管理情報等を所定のタイミングで破棄（削除）する。

【 0022】

なお、逐次解釈部 11 は、「インタプリタ」と呼ばれるプログラムを用いて実現されてもよい。また、翻訳部 13 は、「ジャストインタイムコンパイラ（Just-In-Time コンパイラ）」と呼ばれるプログラムを用いて実現されてもよい。更に、翻訳コード破棄部 15 は、「ガーベージコレクタ」と呼ばれるプログラムを用いて実現されてもよい。

【 0023】

以下、情報処理装置 10 が実行する処理手順について説明する。図 3 は、第一の実施の形態の情報処理装置が実行する処理手順の一例を説明するための図である。

【 0024】

実行対象プログラム Tp がメモリ装置 103 にロードされ、実行対象プログラム Tp の実行が開始されると、実行対象プログラム Tp の終了まで、以下の処理手順が繰り返し実行される。

【 0025】

まず、逐次解釈部 11 は、実行対象プログラム Tp より命令を読み出す（S1）。逐次解釈部 11 は、読み出された命令が分岐命令でなければ、当該命令を逐次的に解釈し、解釈結果を CPU 104 に実行させる（S2）。逐次解釈部 11 は、また、読み出された命令が分岐命令であり、分岐先のアドレスに関してコードキャッシュ部 14 に翻訳コードが生成されていない場合、分岐先のアドレスに係る命令を逐次的に解釈し、解釈結果を CPU 104 に実行させる（S2）。

【 0026】

一方、読み出された命令が分岐命令であり、かつ、分岐先のアドレスに関してコードキャッシュ部 14 に翻訳コードが生成されている場合、逐次解釈部 11 は、当該翻訳コードを CPU 104 に実行させる。（S3）。翻訳コードの実行が終了すると、逐次解釈部 11 に処理制御が復帰する（S4）。

【 0027】

なお、分岐先のアドレスに関して翻訳コードが生成されているか否かや、コードキャッシュ部 14 内における翻訳コードの位置（アドレス）は、例えば、コードキャッシュ部 14 内の管理テーブルを参照して判定される。

【 0028】

図 4 は、第一の実施の形態の管理テーブル 14 Ta の構成例を示す図である。同図において、管理テーブル 14 Ta は、アドレス、実行回数、翻訳済フラグ、翻訳先アドレス等を記憶する。

【 0029】

アドレスは、分岐先の命令のアドレスである。なお、アドレスは、管理テーブル 14 Ta に含まれていなくてもよい。アドレスと管理テーブル 14 Ta の各レコードとの対応付

10

20

30

40

50

けが別途管理されていてもよい。実行回数は、当該アドレスに係る命令群の逐次的な実行回数である。翻訳済フラグは、当該アドレスに係る命令群に関して翻訳コードが生成されているか否かを示す情報である。例えば、「true」は、翻訳コードは生成されている（翻訳済である）ことを示し、「false」は、翻訳コードは生成されていない（翻訳済でない）ことを示す。翻訳先アドレスは、コードキャッシュ部14において、翻訳コードが記憶された位置（アドレス）を示す。また、実行回数は、コードキャッシュ部14外において（すなわち、管理テーブル14Ta外において）管理されてもよい。

【0030】

したがって、逐次解釈部11は、管理テーブル14Taにおいて、分岐先のアドレスに対応する翻訳済みフラグがtrueであれば、分岐先のアドレスに関して翻訳コードは生成されていると判定する。この場合、逐次解釈部11は、当該アドレスに対応する翻訳先アドレスに基づいて、翻訳コードの記憶位置を特定する。

10

【0031】

但し、翻訳済フラグの代わりに、翻訳先アドレスに基づいて、分岐先のアドレスに関して翻訳コードが生成されているか否かが判定されてもよい。すなわち、翻訳先アドレスに値が記録されている場合、分岐先のアドレスに関して翻訳コードは生成されていると判定されてもよい。この場合、管理テーブル14Taに翻訳済フラグは含まれていなくてもよい。

【0032】

なお、逐次解釈部11は、ステップS2の実行時において、読み出された命令が分岐命令である場合、管理テーブル14Taにおいて、分岐先のアドレスに対応する実行回数に1を加算する。但し、分岐先のアドレスに対応するレコードが無い場合、逐次解釈部11は、当該アドレスに対応するレコードを管理テーブル14Taに追加し、当該レコードの実行回数に1を記録する。

20

【0033】

1の加算の結果、実行回数が閾値以上となった場合、逐次解釈部11は、翻訳要求を翻訳要求キュー12に入力する（S5）。翻訳要求には、分岐先の命令のアドレスが指定される。なお、図3の翻訳キュー12内における各矩形は、翻訳要求を表現する。

【0034】

翻訳部13は、翻訳要求キュー12に入力された翻訳要求を古い順に取り出す（S6）。翻訳部13は、取り出された翻訳要求に指定されたアドレスに係る命令群について翻訳処理を実行する（S7）。翻訳処理によって、当該命令群に関する翻訳コードが生成される。翻訳部13は、生成された翻訳コードをコードキャッシュ部14に記憶する（S8）。翻訳部13は、また、管理テーブル14Taにおいて、当該アドレスに対応する翻訳済フラグを「true」に更新し、当該アドレスに対応する翻訳先アドレスに、翻訳コードが記憶されたアドレスを記録する。

30

【0035】

例えば、管理テーブル14Taに新たなレコードが追加され、又は新たな翻訳コードがコードキャッシュ部14に記憶され、コードキャッシュ部14の空き領域が所定量以下になると、翻訳コード破棄部15に対して、GCの実行が要求される（S11又はS12）。

40

【0036】

翻訳コード破棄部15は、要求に応じ、GCを実行する（S13）。すなわち、翻訳コード破棄部15は、一部又は全部の翻訳コード、及び管理テーブル14Taの一部又は全部のレコードをコードキャッシュ部14より破棄（削除）する（S13）。その結果、破棄された翻訳コード又はレコードに使用されていたメモリ領域は解放される。

【0037】

なお、破棄対象が一部の翻訳コードに限定される場合、管理テーブル14Taにおいて破棄されるレコードは、破棄対象の翻訳コードに対応するレコードである。すなわち、破棄対象の翻訳コードのアドレスを翻訳先アドレスとして含むレコードが破棄される。

50

【 0 0 3 8 】

続いて、翻訳コード破棄部 1 5 は、翻訳コード等を破棄した時刻（以下、「G C 実行時刻」という。）を、例えば、メモリ装置 1 0 3 に記録する（S 1 4）。

【 0 0 3 9 】

G C 実行時刻は、逐次解釈部 1 1 が、実行回数に対する閾値を動的に変化させるために用いられる。すなわち、上述したステップ S 2 において、逐次解釈部 1 1 は、G C 実行時刻から所定期間においては、実質的又は相対的に当該閾値を低下させる。閾値が低下することにより、G C 実行時刻から所定期間においては、通常より少ない実行回数でも、翻訳コードが生成されるようになる。したがって、実行頻度の高い命令群に関して、翻訳コードが再生成される時期を早めることができる。その結果、G C の実行による性能の低下期間の短縮化を期待することができる。

10

【 0 0 4 0 】

図 3 における逐次解釈部 1 1 の処理手順について、更に詳細に説明する。図 5 は、第一の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。なお、同図は、読み出された命令が分岐命令である場合の処理手順を示す。

【 0 0 4 1 】

ステップ S 1 1 0 において、逐次解釈部 1 1 は、管理テーブル 1 4 T a を参照して、分岐先のアドレスに対する翻訳コードの有無を判定する。分岐先のアドレスに対応するレコードが管理テーブル 1 4 T a に有り、かつ、当該レコードの翻訳済フラグの値が「true」である場合、逐次解釈部 1 1 は、当該翻訳コードは有ると判定する。一方、分岐先のアドレスに対応するレコードが管理テーブル 1 4 T a に無い場合、又は当該レコードが管理テーブル 1 4 T a に有り、かつ、当該レコードの翻訳済フラグの値が「false」である場合、逐次解釈部 1 1 は、当該翻訳コードは無いと判定する。

20

【 0 0 4 2 】

当該翻訳コードが無い場合（S 1 1 0 で No）、逐次解釈部 1 1 は、分岐先のアドレスの命令を逐次的に解釈し、解釈結果を CPU 1 0 4 に実行させる（S 1 2 0）。続いて、逐次解釈部 1 1 は、管理テーブル 1 4 T a において、分岐先のアドレスに対応する実行回数に 1 を加算する（S 1 3 0）。但し、分岐先のアドレスに対応するレコードが無い場合、逐次解釈部 1 1 は、当該アドレスに対応するレコードを管理テーブル 1 4 T a に追加し、当該レコードの実行回数に 1 を記録する。なお、後述のステップ S 1 7 0 における実行回数は、ステップ S 1 3 0 において更新又は記録された実行回数である。

30

【 0 0 4 3 】

続いて、逐次解釈部 1 1 は、G C 実行時刻がメモリ装置 1 0 3 に記録されているか否かを判定する（S 1 4 0）。すなわち、実行対象プログラム T p の実行開始後において、G C が実行されたか否かが判定される。G C 実行時刻がメモリ装置 1 0 3 に記録されている場合（すなわち、G C が実行されている場合）（S 1 4 0 で Yes）、逐次解釈部 1 1 は、現在時刻と G C 実行時刻の差分（すなわち、G C 実行時刻からの経過時間）を算出する（S 1 5 0）。算出結果は、変数 `after_gc_time` に代入される。一方、G C 実行時刻がメモリ装置 1 0 3 に記録されていない場合（すなわち、G C が実行されていない場合）（S 1 4 0 で No）、逐次解釈部 1 1 は、変数 `after_gc_time` に MAX__GC__TIME を代入する（S 1 6 0）。MAX__GC__TIME は、G C 実行後、実行回数に対する閾値を低下させる期間である。当該期間（MAX__GC__TIME）は、予め決定され、例えば、補助記憶装置 1 0 2 に記録されている。

40

【 0 0 4 4 】

ステップ S 1 5 0 又は S 1 6 0 に続いて、逐次解釈部 1 1 は、以下の式（1）が閾値以上であるか否かを判定する（S 1 7 0）。

【 0 0 4 5 】

$$\text{MAX_GC_TIME} \times \text{実行回数} / \min(\text{after_gc_time}, \text{MAX_GC_TIME}) \cdots (1)$$

ここで、 $\min(x, y)$ は、 x と y との最小値を示す。したがって、`after_gc`

50

$c_time < MAX_GC_TIME$ の場合(すなわち、GC実行時刻からの経過時間が MAX_GC_TIME 未満である場合)、 $\min(after_gc_time, MAX_GC_TIME)$ の値は、 $after_gc_time$ となる。一方、 $after_gc_time \geq MAX_GC_TIME$ の場合(すなわち、GC実行時刻からの経過時間が MAX_GC_TIME 以上である場合)、 $\min(after_gc_time, MAX_GC_TIME)$ の値は、 MAX_GC_TIME となる。

【0046】

上記より、ステップS170では、GC実行時刻からの経過時間が MAX_GC_TIME 未満である場合、式(1)の値は、実際の実行回数より大きな値となる。一方、GC実行時刻からの経過時間が MAX_GC_TIME 以上である場合、式(1)の値は、実際の実行回数と一致する。したがって、GC実行時刻からの経過時間が MAX_GC_TIME 未満である場合、当該経過時間が小さい程、閾値は実質的に小さくなる。一方、GC実行時刻からの経過時間が MAX_GC_TIME 以上である場合、閾値は、本来の値となる。

10

【0047】

式(1)の値が閾値以上である場合(S170でYes)、逐次解釈部11は、分岐先のアドレスが指定された翻訳要求を生成し、当該翻訳要求を翻訳要求キュー12に入力する(S180)。一方、式(1)の値が閾値未満である場合(S170でNo)、ステップS180は実行されない。

【0048】

なお、分岐先のアドレスに対する翻訳コードが有る場合(S110でYes)、逐次解釈部11は、管理テーブル14Taにおいて、当該アドレスに対する翻訳先アドレスに記憶されている翻訳コードをCPU104に実行させる(S190)。

20

【0049】

なお、上記では、GC実行時刻が記録される例を示したが、時刻を計測するタイマーが用いられてもよい。この場合、図3のステップS14において、翻訳コード破棄部15は、当該タイマーの値を0に初期化する。図5のステップS150において、逐次解釈部11は、その時のタイマーの値を $after_gc_time$ に代入する。

【0050】

次に、第二の実施の形態について説明する。第二の実施の形態では、第一の実施の形態と異なる点について説明する。したがって、特に言及しない点については、第一の実施の形態と同様でよい。

30

【0051】

図6は、第二の実施の形態の情報処理装置が実行する処理手順の一例を説明するための図である。図6中、図3と同一部分には同一符号を付し、その説明は省略する。

【0052】

図6では、図3のステップS14がステップS14aに置き換わっている。ステップS14aにおいて、翻訳コード破棄部15は、GC実行時(GC実行直後)における、コードキャッシュ部14の翻訳コードの残量(以下、「GC実行時翻訳コード量」という。)を、例えば、メモリ装置103に記録する。

40

【0053】

GC実行時翻訳コード量は、逐次解釈部11が、実行回数に対する閾値を動的に変化させるために用いられる。すなわち、図6のステップS2において、逐次解釈部11は、GC実行時翻訳コード量から所定量の翻訳コードが生成されるまで(すなわち、GC実行後の翻訳量が所定量になるまで)は、実質的に当該閾値を低下させる。その結果、第一の実施の形態と同様の効果を得ることができる。このように、GC実行後の所定期間は、時刻ではなく、翻訳コード量等、時刻の経過に応じて変化する他の指標を用いて計測されてもよい。

【0054】

第一の実施の形態における図5は、図7に示されるように置き換えられればよい。図7

50

は、第二の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。図7中、図5と同一ステップには同一符号を付し、その説明は省略する。図7では、図5のステップS140、S150、S160、及びS170のそれぞれが、ステップS140a、S150a、S160a、又はS170aに置き換わっている。

【0055】

ステップS140aにおいて、逐次解釈部11は、GC実行時翻訳コード量がメモリ装置103に記録されているか否かを判定する。GC実行時翻訳コード量がメモリ装置103に記録されている場合（すなわち、GCが実行されている場合）（S140aでYes）、逐次解釈部11は、現在のコードキャッシュ部14における翻訳コードの残量と、GC実行時翻訳コード量との差分（すなわち、GC実行時からの翻訳コード量）を算出する（S150a）。算出結果は、変数after_gc_volに代入される。

10

【0056】

一方、一方、GC実行時翻訳コード量がメモリ装置103に記録されていない場合（すなわち、GCが実行されていない場合）（S140aでNo）、逐次解釈部11は、変数after_gc_volにMAX_GC_VOLを代入する（S160a）。上記において、第二の実施の形態では、GC実行後の翻訳量が所定量になるまで閾値が実質的に低下すると述べた。MAX_GC_VOLは、当該所定量を示す。MAX_GC_VOLは、予め決定され、例えば、補助記憶装置102に記録されている。

【0057】

ステップS150a又はS160bに続いて、逐次解釈部11は、以下の式(2)が閾値以上であるか否かを判定する（S170a）。

20

【0058】

$MAX_GC_VOL \times \text{実行回数} / \min(after_gc_vol, MAX_GC_VOL) \dots (2)$

式(2)は、第一の実施の形態の式(1)におけるafter_gc_time及びMAX_GC_TIMEのそれぞれが、after_gc_vol又はMAX_GC_VOLに置換されたものである。したがって、GC実行時からの翻訳コード量がMAX_GC_VOL未満である場合、当該翻訳コード量が小さい程、閾値は実質的に小さくなる。一方、GC実行時刻からの経過時間がMAX_GC_VOL以上である場合、閾値は、本来の値となる。

30

【0059】

なお、上記第一及び第二の実施の形態において、GC実行時刻からの経過時間がMAX_GC_TIME未満である場合、又はGC実行時からの翻訳コード量がMAX_GC_VOL未満である場合において、式(1)又は式(2)の値は一定となるようにしてもよい。この場合、例えば、式(1)におけるafter_gc_time又は式(2)におけるafter_gc_volの値を、MAX_GC_TIME/n、又はMAX_GC_VOL/nとしてもよい。ここで、 $n > 1$ である。その結果、GC実行時刻からの経過時間がMAX_GC_TIME未満である場合、又はGC実行時からの翻訳コード量がMAX_GC_VOL未満である場合において、閾値の実質的な低下の程度は一定となる。

40

【0060】

また、第一及び第二の実施の形態では、所定期間において命令の実行回数を過大評価されることにより、閾値が実質的に低下する例について説明したが、閾値それ自体が動的に変化するようにしてもよい。例えば、翻訳要求キュー12に蓄積されている翻訳要求の数に応じて閾値が変化してもよい。この場合、ステップS170又はS170aにおいて用いられる閾値は、例えば、以下の式(3)に基づいて算出される。

【0061】

閾値 = (翻訳要求キュー12に蓄積されている翻訳要求の数 × 定数A) + 定数B …… (3)

例えば、定数A = 1024、定数B = 1が式(3)に当てはめられて、閾値が算出され

50

てもよい。

【 0 0 6 2 】

次に、第三の実施の形態について説明する。第三の実施の形態では、第一の実施の形態と異なる点について説明する。したがって、特に言及しない点については、第一の実施の形態と同様でよい。

【 0 0 6 3 】

第三の実施の形態では、翻訳コードと実行対象プログラム T p との間に不整合が発生する場合について考慮する。斯かる不整合の発生は、実行対象プログラム T p が、CPU 104 と異なるアーキテクチャの CPU のエミュレーションを実現するためのプログラムである場合に、特に顕在化する。すなわち、第三の実施の形態は、実行対象プログラム T p 、逐次解釈部 11、コードキャッシュ部 14、及び翻訳コード破棄部 15 が、CPU 104 と異なるアーキテクチャの CPU をエミュレータとして機能する場合に好適である。但し、他の場合に第三の実施の形態が適用されてもよい。

10

【 0 0 6 4 】

翻訳コードの不整合は、例えば、図 8 に示されるような状況において発生する。図 8 は、翻訳コードに不整合が発生する状況の一例を説明するための図である。

【 0 0 6 5 】

同図では、自己書き換えの例が示されている。すなわち、実行対象プログラム T p の命令の実行（逐次的な実行又は当該命令を含む翻訳コードによる実行）により、実行対象プログラム T p の一部の命令 P が動的に書き換えられることがある。このように、プログラムが実行時に自分自身の命令を書き換えることを「自己書き換え」という。この場合、図 8 において、書き換えられた命令 P を含む翻訳コード C p は、実行対象プログラム T p の現在の状態と異なることになる。すなわち、翻訳コード C p に不整合が発生した状態となる。

20

【 0 0 6 6 】

不整合な翻訳コード C p の実行は、回避されなければならない。そこで、逐次解釈部 11 は、不整合な翻訳コード C p に対して無効であることを示す情報を関連付ける。当該情報を関連付けるため、第三の実施の形態の管理テーブル 14 T は、例えば、図 9 に示されるような構成を有する。

【 0 0 6 7 】

図 9 は、第三の実施の形態の管理テーブルの構成例を示す図である。同図において、管理テーブル 14 T b は、アドレスに関連付けて無効化フラグを記憶する。無効化フラグは、当該アドレスに係る翻訳コードが無効であるか否かを示す情報である。無効化フラグの初期値は「false」である。翻訳コードに不整合が発生した又は発生した可能性が有る場合、無効化フラグの値は「true」に更新される。翻訳コードに関して無効化フラグが「true」に更新されることを、本実施の形態において、翻訳コードの無効化という。

30

【 0 0 6 8 】

逐次解釈部 11 は、分岐先のアドレスに対する無効化フラグの値が「true」である場合、翻訳コードは実行対象とせず、改めて、当該分岐先の命令を逐次的に実行する。

40

【 0 0 6 9 】

本実施の形態において、無効化フラグの更新は、逐次解釈部 11 によって行われる。或る翻訳コードに不整合が発生したことを検知するために、逐次解釈部 11 又は翻訳部 13 は、翻訳コードが生成された命令が属する実行対象プログラム T p のページに関して、ページテーブルのアクセス権として「読み込み可 (Read Only)」を設定しておく。すなわち、当該ページに対する書き込みは禁止される。その後、実行対象プログラム T p の命令の実行により、当該ページに対して自己書き換えを実行しようとする、アクセス違反に関する例外が発生する。逐次解釈部 11 は、当該例外を補足（キャッチ）するためのハンドラを介して、当該例外の発生を検知する。逐次解釈部 11 は、当該例外の発生を検知すると、当該例外によって通知されたアドレスを含むページに属する翻訳コードに係

50

るアドレスに関して、管理テーブル 14 T b の無効化フラグを「true」に更新する。

【0070】

なお、翻訳コードの不整合は、新たなプロセスが生成された場合やスワッピングが発生した場合等において、仮想アドレスと物理アドレスとのマッピングの変更が必要となったときにも発生する。この場合に関しても、逐次解釈部 11 は、影響を受ける翻訳コードに関して無効化フラグを「true」にする。

【0071】

ここで、無効化された翻訳コード及び無効化されていない（有効な）翻訳コードのそれぞれと、GC との関係について説明する。無効化された翻訳コードは、不要な翻訳コードであり、今後使用することができない翻訳コードである。したがって、当該翻訳コードに関して GC が実行されることによる性能上の不利益は、相対的に小さいものと考えられる。一方、有効な翻訳コードは、今後も使用することができ、また、使用される可能性の有る翻訳コードである。したがって、当該翻訳コードに関して GC が実行されることによる性能上の不利益は、相対的に大きいものと考えられる。

10

【0072】

以上に鑑みて、第三の実施の形態では、無効化された翻訳コードと有効な翻訳コードとの間で、GC の実行後の実行回数に対する閾値を変えることができるようにする。すなわち、有効な翻訳コードは、早期に復帰されるように、閾値が実質的に下げられる。

【0073】

このような制御を実現するため、第三の実施の形態の情報処理装置 10 は、例えば、図 10 に示されるような機能構成を有する。

20

【0074】

図 10 は、第三の実施の形態における情報処理装置の機能構成例を示す図である。同図において、情報処理装置 10 は、強制破棄フラグ記憶部 16 を更に有する。強制破棄フラグ記憶部 16 は、例えば、メモリ装置 103 又は補助記憶装置 102 を用いて実現可能である。

【0075】

強制破棄フラグ記憶部 16 は、有効な翻訳コードであったが、GC の実行により強制的に破棄された翻訳コードを識別するための情報（以下、「強制破棄フラグ」という。）を記憶する。但し、翻訳コードごとに強制破棄フラグを管理するのは、管理情報の増加を招く。そこで、本実施の形態において、強制破棄フラグは、実行対象プログラム Tp のページ単位に管理される。

30

【0076】

図 11 は、強制破棄フラグ記憶部の構成例を示す図である。同図に示されるように、強制破棄フラグ記憶部 16 は、ページアドレスごとに強制破棄フラグを記憶する。ページアドレスは、ページの開始アドレスである。ページのサイズは決まっているため、開始アドレスに基づいて、各ページの範囲が特定される。なお、ページのサイズは、OS（Operating System）によって管理されるページのサイズと必ずしも一致していなくてもよい。

【0077】

GC 実行時に既に無効化されていた翻訳コードを含むページに関しては、強制破棄フラグに「false」が記録され、有効であった翻訳コードのみを含むページに関しては、強制破棄フラグに「true」が記録される。なお、強制破棄フラグは、当該翻訳コードの生成後に、実行対象プログラム Tp において当該翻訳コードに対応する命令が更新されていない翻訳コードを示す情報の一例である。

40

【0078】

以下、第三の実施の形態の処理手順について説明する。図 12 は、第三の実施の形態の GC の実行処理の処理手順の一例を説明するためのフローチャートである。同図は、一回の GC において（すなわち、図 3 の S13 の一回の実行において）実行される処理である。

【0079】

50

ステップS 2 0 1において、翻訳コード破棄部 1 5 は、強制破棄フラグ記憶部 1 6 における、全てのページアドレスに対する強制破棄フラグを「f a l s e」に初期化する。続いて、翻訳コード破棄部 1 5 は、破棄対象の翻訳コード群を決定する（S 2 0 2）。破棄対象の決定方法は、公知の方法が用いられればよい。一部の翻訳コードが破棄対象とされてもよいし、全部の翻訳コードが破棄対象とされてもよい。

【 0 0 8 0 】

破棄対象の翻訳コード群が有る場合（S 2 0 3でY e s）、翻訳コード破棄部 1 5 は、当該翻訳コード群のうち一つの翻訳コードを処理対象（以下、「対象翻訳コード」という。）とする（S 2 0 4）。続いて、翻訳コード破棄部 1 5 は、管理テーブル 1 4 T b において、対象翻訳コードに対応するレコードの無効化フラグを参照して、対象翻訳コードは無効化されているか否かを判定する（S 2 0 5）。すなわち、無効化フラグが「t u r e」であるか「f a l s e」であるかが確認される。

10

【 0 0 8 1 】

対象翻訳コードが無効化されていない場合（S 2 0 5でN o）、翻訳コード破棄部 1 5 は、実行対象プログラムT pのページの中で、対象翻訳コードが属するページのアドレスを求める（S 2 0 6）。続いて、翻訳コード破棄部 1 5 は、強制破棄フラグ記憶部 1 6 において、当該ページのアドレスに対する強制破棄フラグの値を「t r u e」に更新する（S 2 0 7）。すなわち、対象翻訳コードに関して、対象翻訳コードの生成後、実行対象プログラムT pにおいて対象翻訳コードに対応する命令は更新されていないことを示す情報が強制破棄フラグ記憶部 1 6 に記録される。ここでいう更新とは、例えば、自己書き換え等による更新である。

20

【 0 0 8 2 】

続いて、翻訳コード破棄部 1 5 は、対象翻訳コード、及び管理テーブル 1 4 T b において対象翻訳コードに対応するレコード等をコードキャッシュ部 1 4 より破棄する（S 2 0 8）。なお、対象翻訳コードが無効化されている場合（S 2 0 5でY e s）、強制破棄フラグの更新は行われずに、対象翻訳コード等に関して破棄が実行される（S 2 0 9）。

【 0 0 8 3 】

続いて、翻訳コード破棄部 1 5 は、破棄対象の翻訳コード群に属する全ての翻訳コードについてステップS 2 0 3以降を繰り返す。

【 0 0 8 4 】

図 1 2 の処理の結果、強制破棄フラグ記憶部 1 6 は、有効であった翻訳コードのみを含むページに関して「t r u e」を記憶した状態となる。

30

【 0 0 8 5 】

続いて、第三の実施の形態の逐次解釈部 1 1 の処理手順について説明する。図 1 3 は、第三の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。図 1 3 中、図 5 と同一ステップには同一符号を付し、その説明は省略する。図 1 3 では、ステップS 1 3 2 及びS 1 3 3 が追加されている。

【 0 0 8 6 】

ステップS 1 3 2 において、逐次解釈部 1 1 は、分岐先アドレスを含むページに対する強制破棄フラグの値を強制破棄フラグ記憶部 1 6 より取得する。取得された強制破棄フラグの値が「t r u e」である場合（S 1 3 3でY e s）、ステップS 1 4 0 以降が実行される。すなわち、G C 実行後の一定期間において、閾値が実質的に下げられる。一方、取得された強制破棄フラグの値が「f a l s e」である場合（S 1 3 3でN o）、ステップS 1 6 0 以降が実行される。すなわち、閾値の実質的な低下は行われない。

40

【 0 0 8 7 】

なお、図 1 3 のステップS 1 4 0、S 1 5 0、S 1 6 0、及びS 1 7 0 のそれぞれは、図 7 のステップS 1 4 0 a、S 1 5 0 a、S 1 6 0 a、又はS 1 7 0 a に置き換えられてもよい。すなわち、第二の実施の形態と同様に、G C 実行後の所定期間が、翻訳コード量によって計測されてもよい。

【 0 0 8 8 】

50

また、上記では、ページごとに強制破棄フラグが管理される例を示したが、翻訳コードごと（分岐先アドレスごと）に強制破棄フラグが管理されてもよい。この場合、より厳密に、無効化されていた翻訳コードと、有効であった翻訳コードとの閾値を実質的に変えることができる。

【 0 0 8 9 】

上述したように、第三の実施の形態によれば、使用される可能性があった翻訳コードについて、使用される可能性の無い翻訳コードに比較して、GC実行後の復帰時期を早めることができる。その結果、翻訳コードの破棄による性能の低下期間を短縮することができる。

【 0 0 9 0 】

次に、第四の実施の形態について説明する。第四の実施の形態では、第一の実施の形態と異なる点について説明する。したがって、特に言及しない点については、第一の実施の形態と同様でよい。

【 0 0 9 1 】

図14は、第四の実施の形態における情報処理装置の機能構成例を示す図である。同図において情報処理装置10は、コードキャッシュ部14a及びコードキャッシュ部14bの二つのコードキャッシュ部14を有する。

【 0 0 9 2 】

コードキャッシュ部14a及びコードキャッシュ部14bは、第一の実施の形態のコードキャッシュ部14が二分割されたものでもよいし、第一の実施の形態のコードキャッシュ部14の二つに相当するものであってもよい。

【 0 0 9 3 】

以下、第四の実施の形態の処理手順について説明する。図15は、第四の実施の形態の翻訳コード破棄部の初期化処理の処理手順の一例を示す図である。同図の処理手順は、実行対象プログラムTpの一回の実行に関して、翻訳コード破棄部15が最初に起動される際に実行される。

【 0 0 9 4 】

ステップS301において、翻訳コード破棄部15は、コードキャッシュ部14aがカレントコードキャッシュであることをメモリ装置103に記録する。例えば、カレントコードキャッシュを特定するための変数に、コードキャッシュ部14aの先頭アドレスが代入される。なお、カレントコードキャッシュとは、使用対象とされるコードキャッシュ部14をいう。

【 0 0 9 5 】

図16は、第四の実施の形態のGCの実行処理の処理手順の一例を説明するためのフローチャートである。同図は、一回のGCにおいて（すなわち、図3のS13の一回の実行において）実行される処理である。

【 0 0 9 6 】

ステップS311において、翻訳コード破棄部15は、カレントコードキャッシュは、ラストコードキャッシュであることをメモリ装置103に記録する。例えば、カレントコードキャッシュを特定するための変数の値が、ラストコードキャッシュを特定するための変数に代入される。なお、ラストコードキャッシュとは、最後に使用されたコードキャッシュ部14をいう。

【 0 0 9 7 】

続いて、翻訳コード破棄部15は、コードキャッシュ部14a及びbのうち、ラストコードキャッシュでない方が、カレントコードキャッシュであることをメモリ装置103に記録する（S312）。すなわち、ステップS311及びS312では、カレントコードキャッシュとラストコードキャッシュとの入れ替えが行われる。具体的には、コードキャッシュ部14a及びbのうち、前回のGC実行時から現時点までに使用対象でなかった方が、使用対象（カレントコードキャッシュ）とされる。

【 0 0 9 8 】

10

20

30

40

50

続いて、翻訳コード破棄部 15 は、カレントコードキャッシュについて G C を実行する (S 3 1 3)。

【 0 0 9 9 】

例えば、図 16 の開始時点において、コードキャッシュ部 14 a がカレントコードキャッシュであったとする。コードキャッシュ部 14 a は、G C が実行されるまで使用されていたため、翻訳コードが記録され、管理テーブル 14 T に対する記録等も行われている。

【 0 1 0 0 】

ステップ S 3 1 1 及び S 3 1 2 の実行により、コードキャッシュ部 14 a はラストコードキャッシュとなり、コードキャッシュ部 14 b がカレントコードキャッシュとなる。したがって、ステップ S 3 1 3 では、コードキャッシュ部 14 b が、G C の対象とされる。その結果、コードキャッシュ部 14 a に対する記録内容は、退避 (又は保存) された状態となる。

【 0 1 0 1 】

このように、第四の実施の形態では、G C までにカレントコードキャッシュに対して行われた記録内容が退避される。

【 0 1 0 2 】

続いて、図 17 は、第四の実施の形態における翻訳部の処理手順の一例を説明するためのフローチャートである。図 17 中、図 3 と同一ステップ (S 6、S 7) には同一ステップ番号を付し、その説明は省略する。

【 0 1 0 3 】

図 17 では、ステップ S 8 の代わりにステップ S 8 a が実行される。ステップ S 8 a において、翻訳部 13 は、生成された翻訳コードをカレントコードキャッシュに記憶する。また、翻訳部 13 は、カレントコードキャッシュの管理テーブル 14 T において、翻訳要求に指定されたアドレスに対応する翻訳済フラグを「 true 」に更新し、当該アドレスに対応する翻訳先アドレスに、翻訳コードが記憶されたアドレスを記録する。

【 0 1 0 4 】

続いて、図 18 は、第四の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。図 18 中、図 5 と同一ステップには同一符号を付し、その説明は省略する。図 13 では、ステップ S 110 がステップ S 110 a に置き換えられ、ステップ S 131 が追加されている。

【 0 1 0 5 】

ステップ S 110 a において、逐次解釈部 11 は、カレントコードキャッシュの管理テーブル 14 T a を参照して、分岐先のアドレスに対する翻訳コードの有無を判定する。

【 0 1 0 6 】

また、ステップ S 131 において、逐次解釈部 11 は、ラストコードキャッシュの管理テーブル 14 T a を参照して、分岐先のアドレスに対する翻訳コードの有無を判定する。ラストコードキャッシュに当該翻訳コードが有る場合 (S 131 で Yes)、ステップ S 140 以降が実行される。すなわち、G C 実行後の一定期間において、閾値が実質的に下げられる。一方、ラストコードキャッシュに当該翻訳コード無い場合 (S 131 で No)、ステップ S 160 以降が実行される。すなわち、閾値の実質的な低下は行われない。

【 0 1 0 7 】

なお、図 18 のステップ S 140、S 150、S 160、及び S 170 のそれぞれは、図 7 のステップ S 140 a、S 150 a、S 160 a、又は S 170 a に置き換えられてもよい。すなわち、第二の実施の形態と同様に、G C 実行後の所定期間が、翻訳コード量によって計測されてもよい。

【 0 1 0 8 】

なお、上記では、カレントコードキャッシュ及びラストコードキャッシュに関して、コードキャッシュ部 14 a 及びコードキャッシュ部 14 b を入れ替えることにより、カレントコードキャッシュの記憶内容を退避する例について説明した。但し、コードキャッシュ部 14 a が常に使用対象 (カレントコードキャッシュ) とされてもよい。この場合、図 1

10

20

30

40

50

6のステップS311及びS312の代わりに、コードキャッシュ部14aの内容が、コードキャッシュ部14bにコピーされる処理が実行されればよい。その結果、GC前のコードキャッシュ部14aの内容を退避することができる。なお、退避対象は、カレントコードキャッシュの記憶内容の全部でなくてよい。翻訳コードが生成されていた命令群を識別可能な情報が退避されればよい。本実施の形態では、管理テーブル14Taにおけるアドレスの値が退避されればよい。

【0109】

上述したように、第四の実施の形態によれば、過去に翻訳コードが生成されている命令群に関して、実質的に閾値を低下させることができる。したがって、斯かる命令群の翻訳を優先させることができる。過去に翻訳コードが生成されている命令群は、以降の実行頻度も高い傾向にある。したがって、斯かる命令群の翻訳コードの再生成が早期化されることにより、翻訳コードの破棄による性能の低下期間を短縮することができる。

10

【0110】

また、全ての命令群に関して一律に閾値が低下される場合に比べて、翻訳コード量の増大を抑制することができる。

【0111】

なお、ステップS131において、分岐先のアドレスに対する翻訳コードの有無のみではなく、分岐先アドレスに対する実行回数が閾値以上であるか否かが判定されてもよい。当実行回数が閾値以上であれば、ステップS140以降が実行され、閾値未満であれば、ステップS160以降が実行されるようにしてもよい。そうすることで、より実行頻度の高い命令群に翻訳対象を限定することができる。この場合、退避対象には、管理テーブル14Taにおけるアドレスの値に加えて、実行回数の値が含まれるようにすればよい。また、閾値は、ステップS170において利用される閾値と同じであってもよいし、異なってもよい。

20

【0112】

次に、第五の実施の形態について説明する。第五の実施の形態は、第三の実施の形態と第四の実施の形態とを組み合わせたものである。したがって、逐次解釈部11は、図19に示される処理を実行する。

【0113】

図19は、第五の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャートである。図19は、図18のステップS131の後に、図16のステップS132及びS133が追加されたものである。同図の処理手順は、図16及び図18等より自明であるため省略する。

30

【0114】

第五の実施の形態によれば、第三及び第四の実施の形態の双方の効果を得ることができる。

【0115】

なお、上記各実施の形態において、翻訳部13は、生成部の一例である。翻訳コード破棄部15は、破棄部の一例である。コードキャッシュ部14は、第一の記憶部の一例である。強制破棄フラグ記憶部16は、第二の記憶部の一例である。

40

【0116】

以上、本発明の実施例について詳述したが、本発明は斯かる特定の実施形態に限定されるものではなく、特許請求の範囲に記載された本発明の要旨の範囲内において、種々の変形・変更が可能である。

【0117】

以上の説明に関し、更に以下の項を開示する。

(付記1)

実行対象プログラムに含まれる命令のうち、逐次解釈による実行回数が閾値以上となった命令の翻訳コードを生成して第一の記憶部に記憶し、

所定のタイミングで前記翻訳コードの一部又は全部を前記第一の記憶部より破棄する処

50

理をコンピュータが実行し、

前記生成する処理は、前記翻訳コードが破棄されてから所定期間において、前記閾値を前記実行回数に対して相対的に低下させることを特徴とする命令処理方法。

(付記 2)

前記破棄する処理は、破棄対象の前記翻訳コードの中で、当該翻訳コードの生成後に前記実行対象プログラムにおいて当該翻訳コードに対応する命令が更新されていない翻訳コードを示す情報を第二の記憶部に記憶し、

前記生成する処理は、前記第二の記憶部が記憶する情報が示す前記翻訳コードに関して、破棄されてからの所定期間において前記閾値を低下させる付記 1 記載の命令処理方法。

10

(付記 3)

前記破棄する処理は、前記第一の記憶部が前記翻訳コードに関連付けて記憶する該翻訳コードに対応する命令の識別情報を退避し、

前記生成する処理は、退避されている前記識別情報に係る命令に関して、破棄されてからの所定期間において前記閾値を低下させる付記 1 又は 2 記載の命令処理方法。

(付記 4)

実行対象プログラムに含まれる命令のうち、逐次解釈による実行回数が閾値以上となった命令の翻訳コードを生成して第一の記憶部に記憶する生成部と、

所定のタイミングで前記翻訳コードの一部又は全部を前記第一の記憶部より破棄する破棄部とを有し、

20

前記生成部は、前記翻訳コードが破棄されてから所定期間において、前記閾値を前記実行回数に対して相対的に低下させる命令処理装置。

(付記 5)

前記破棄部は、破棄対象の前記翻訳コードの中で、当該翻訳コードの生成後に前記実行対象プログラムにおいて当該翻訳コードに対応する命令が更新されていない翻訳コードを示す情報を第二の記憶部に記憶し、

前記生成部は、前記第二の記憶部が記憶する情報が示す前記翻訳コードに関して、破棄されてからの所定期間において前記閾値を低下させる付記 4 記載の命令処理装置。

(付記 6)

前記破棄部は、前記第一の記憶部が前記翻訳コードに関連付けて記憶する該翻訳コードに対応する命令の識別情報を退避し、

30

前記生成部は、退避されている前記識別情報に係る命令に関して、破棄されてからの所定期間において前記閾値を低下させる付記 4 又は 5 記載の命令処理装置。

(付記 7)

実行対象プログラムに含まれる命令のうち、逐次解釈による実行回数が閾値以上となった命令の翻訳コードを生成して第一の記憶部に記憶し、

所定のタイミングで前記翻訳コードの一部又は全部を前記第一の記憶部より破棄する処理をコンピュータに実行させ、

前記生成する処理は、前記翻訳コードが破棄されてから所定期間において、前記閾値を前記実行回数に対して相対的に低下させる命令処理プログラム。

40

(付記 8)

前記破棄する処理は、破棄対象の前記翻訳コードの中で、当該翻訳コードの生成後に前記実行対象プログラムにおいて当該翻訳コードに対応する命令が更新されていない翻訳コードを示す情報を第二の記憶部に記憶し、

前記生成する処理は、前記第二の記憶部が記憶する情報が示す前記翻訳コードに関して、破棄されてからの所定期間において前記閾値を低下させる付記 7 記載の命令処理プログラム。

(付記 9)

前記破棄する処理は、前記第一の記憶部が前記翻訳コードに関連付けて記憶する該翻訳コードに対応する命令の識別情報を退避し、

50

前記生成する処理は、退避されている前記識別情報に係る命令に関して、破棄されてからの所定期間において前記閾値を低下させる付記7又は8記載の命令処理プログラム。

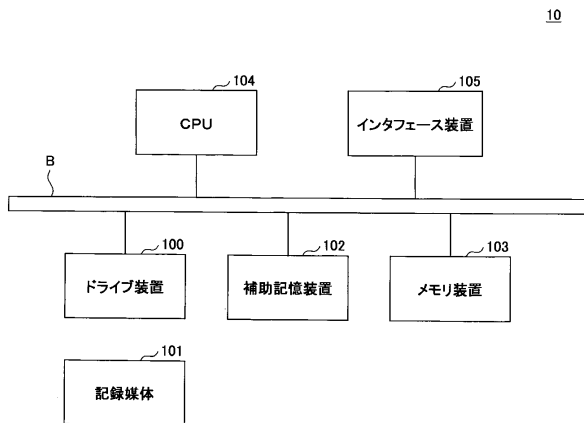
【符号の説明】

【0118】

- 10 情報処理装置
- 11 逐次解釈部
- 12 翻訳要求キュー
- 13 翻訳部
- 14 コードキャッシュ部
- 15 翻訳コード破棄部
- 16 強制破棄フラグ記憶部
- 100 ドライブ装置
- 101 記録媒体
- 102 補助記憶装置
- 103 メモリ装置
- 104 CPU
- 105 インタフェース装置
- 10 インタフェース装置
- B バス

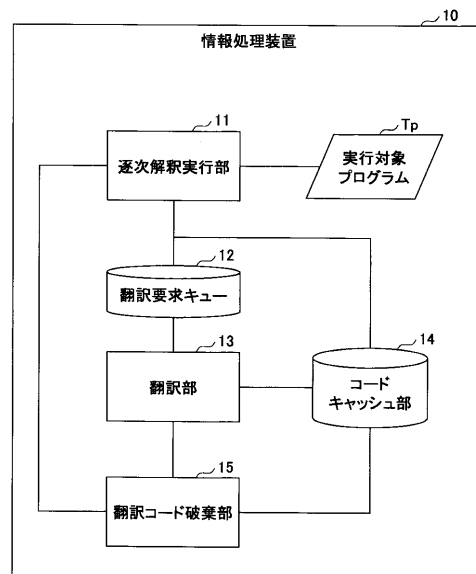
【図1】

本発明の実施の形態における情報処理装置のハードウェア構成例を示す図



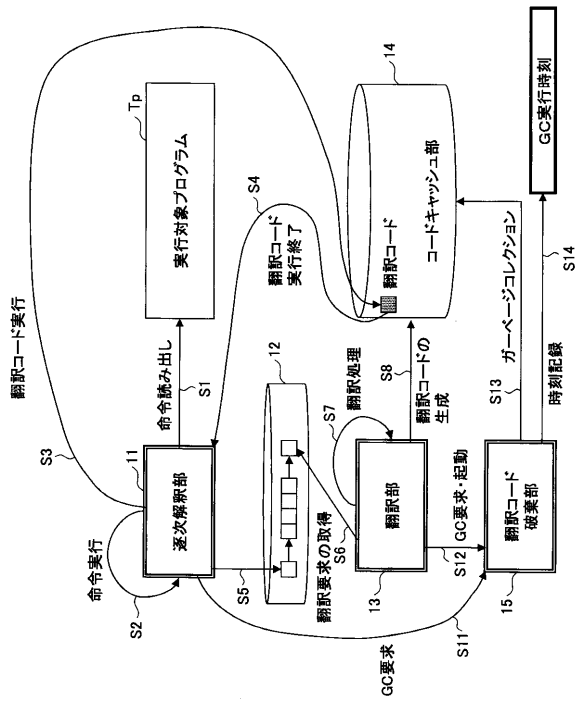
【図2】

第一の実施の形態における情報処理装置の機能構成例を示す図



【図3】

第一の実施の形態の情報処理装置が実行する処理手順の一例を説明するための図



【図4】

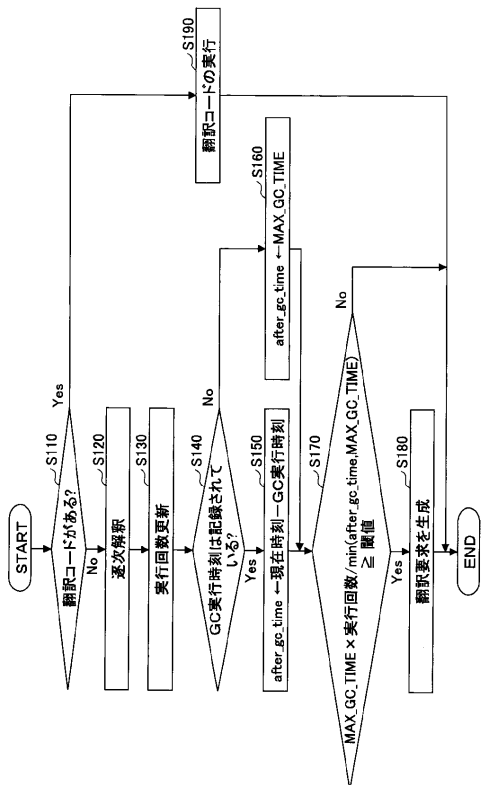
第一の実施の形態の管理テーブルの構成例を示す図

14Ta

アドレス	実行回数	翻訳済フラグ	翻訳先アドレス	...

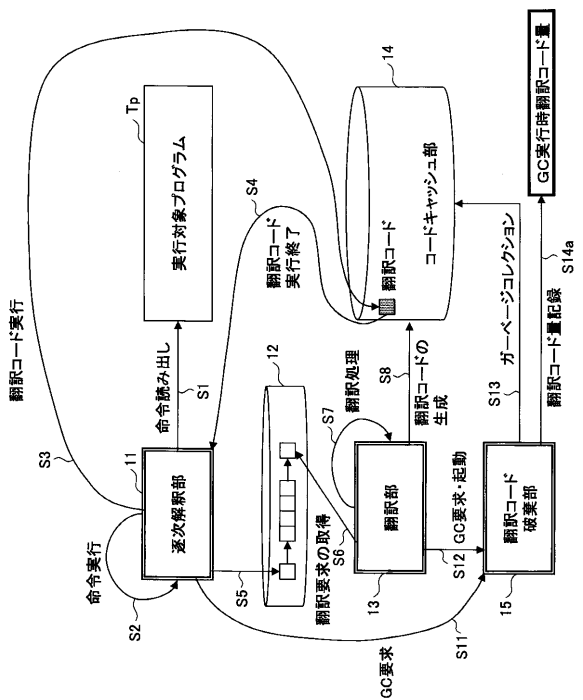
【図5】

第一の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャート



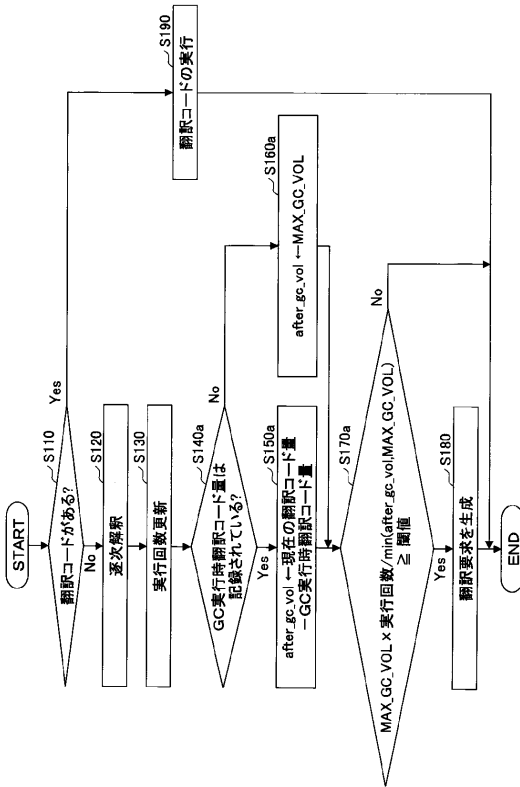
【図6】

第二の実施の形態の情報処理装置が実行する処理手順の一例を説明するための図



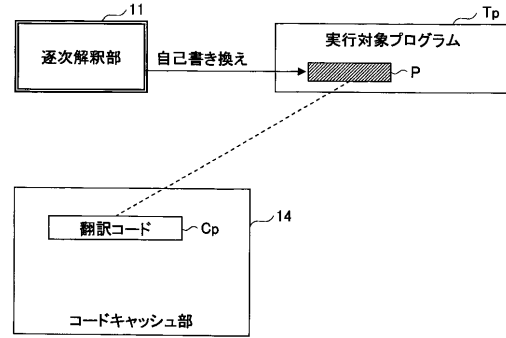
【図7】

第二の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャート



【図8】

翻訳コードに不整合が発生する状況の一例を説明するための図



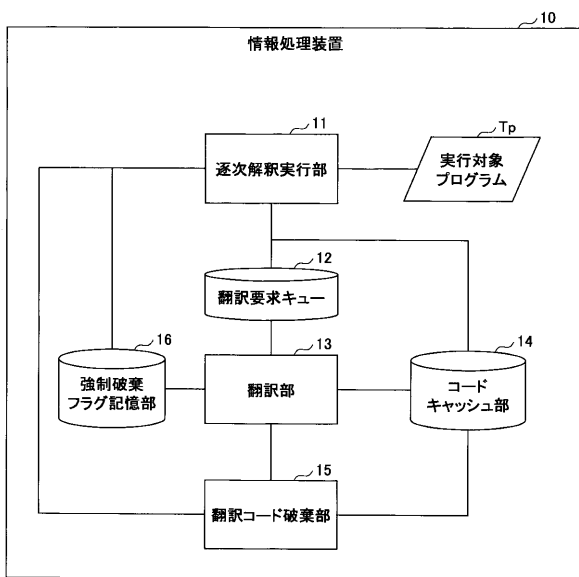
【図9】

第三の実施の形態の管理テーブルの構成例を示す図

アドレス	実行回数	翻訳済フラグ	翻訳先アドレス	無効化フラグ	...
...
...
...

【図10】

第三の実施の形態における情報処理装置の機能構成例を示す図



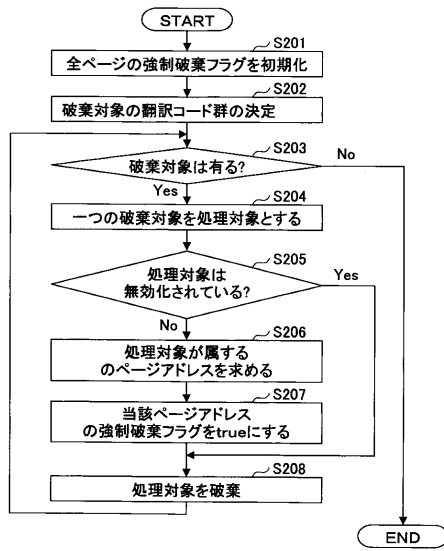
【図11】

強制破棄フラグ記憶部の構成例を示す図

ページアドレス	強制破棄フラグ

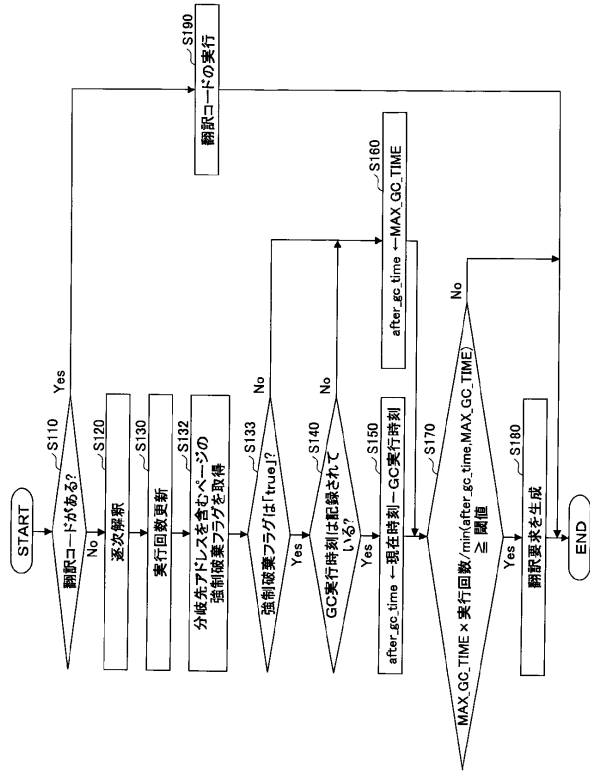
【図12】

第三の実施の形態のGCの実行処理の処理手順の一例を説明するためのフローチャート



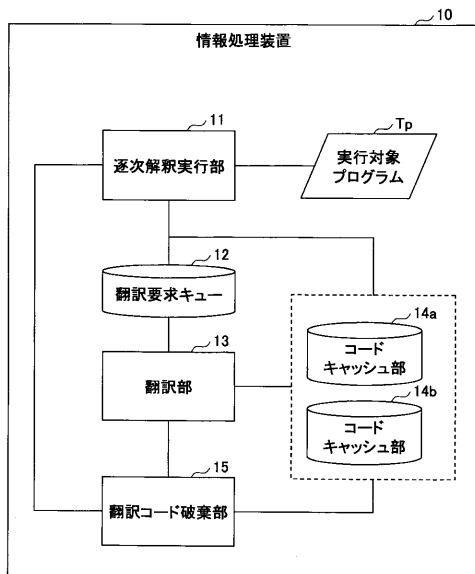
【図13】

第三の実施の形態の逐次解釈部の処理手順の一例を説明するためのフローチャート



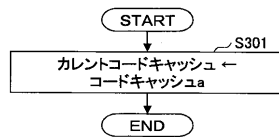
【図14】

第四の実施の形態における情報処理装置の機能構成例を示す図



【図15】

第四の実施の形態の翻訳コード破棄部の初期化処理の処理手順の一例を示す図



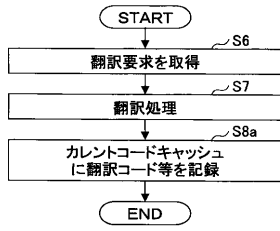
【図16】

第四の実施の形態のGCの実行処理の処理手順の一例を説明するためのフローチャート



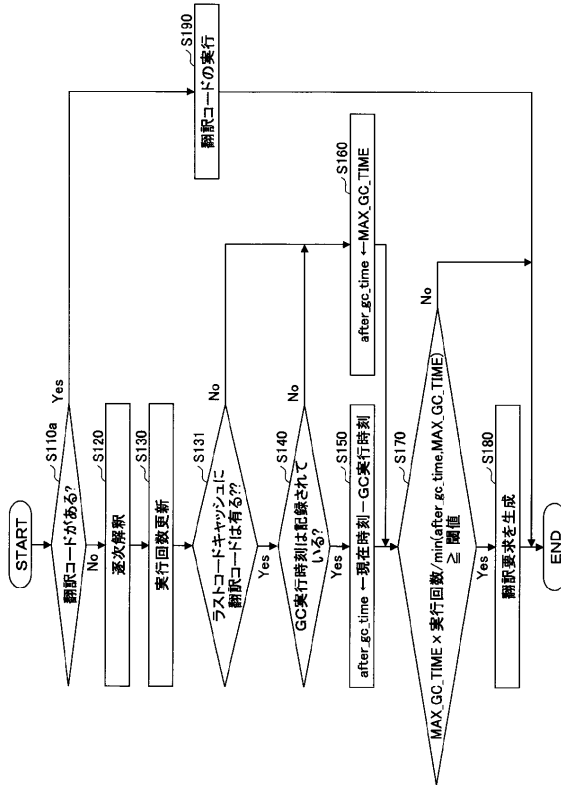
【図17】

第四の実施の形態における
翻訳部の処理手順の一例を説明するためのフローチャート



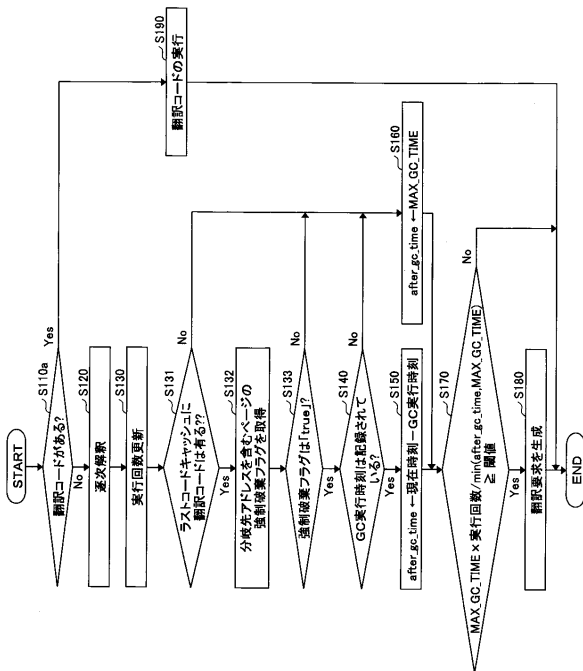
【図18】

第四の実施の形態の
逐次解釈部の処理手順の一例を説明するためのフローチャート



【図19】

第五の実施の形態の
逐次解釈部の処理手順の一例を説明するためのフローチャート



フロントページの続き

(72)発明者 竹内 剛
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

審査官 坂庭 剛史

(56)参考文献 特表2006-521597(JP,A)
特開2000-035890(JP,A)
特開平11-272476(JP,A)
特開2004-005650(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 9/45