

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2008年10月23日 (23.10.2008)

PCT

(10) 国際公開番号
WO 2008/126299 A1(51) 国際特許分類:
G06F 9/45 (2006.01)

(21) 国際出願番号: PCT/JP2007/057233

(22) 国際出願日: 2007年3月30日 (30.03.2007)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(71) 出願人 (米国を除く全ての指定国について): 富士通株式会社 (FUJITSU LIMITED) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中4丁目1番1号 Kanagawa (JP).

(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 小澤 誠 (KOZAWA, Makoto) [JP/JP]; 〒2118588 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 Kanagawa (JP).

(74) 代理人: 酒井 宏明 (SAKAI, Hiroaki); 〒1006020 東京都千代田区霞が関三丁目2番5号 霞が関ビルディング 酒井国際特許事務所 Tokyo (JP).

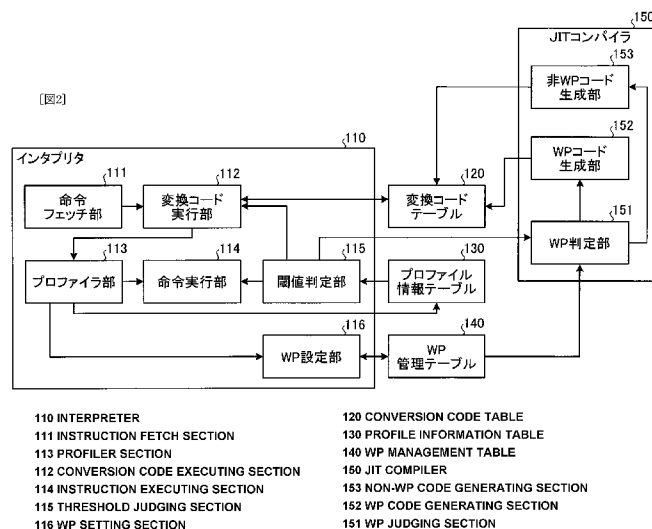
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, 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.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE,

[続葉有]

(54) Title: COMPILE APPARATUS, COMPILE METHOD, AND CODE GENERATION PROGRAM

(54) 発明の名称: コンパイル装置、コンパイル方法、およびコード生成プログラム



(57) **Abstract:** The efficiency of compilation by a dynamic compiler is improved to execute a program at high speed. An interpreter (110) takes out an instruction code from a program to be executed and executes the instruction code or a conversion code after the compilation. The interpreter (110) also sets/removes write protection against the instruction code and records the presence/absence of the occurrence of write protection violation in a WP management table (140). The WP management table (140) stores whether or not the write protection violation occurred in the past for each instruction code. A JIT compiler (150) generates a WP code assuming that the write protection is set against the instruction code in which the write protection violation did not occur in the past and generates a non-WP code not assuming that the write protection is set against the instruction code in which the write protection violation occurred in the past.

[続葉有]



IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

(57) 要約: 動的コンパイラによるコンパイルの効率化を図り、プログラムを高速に実行すること。この課題を解決するために、インタプリタ(110)は、実行対象のプログラムから命令コードを取り出して、命令コードまたはコンパイル後の変換コードを実行する。また、インタプリタ(110)は、命令コードに対するライトプロテクションの設定・解除をし、ライトプロテクション違反の発生の有無をWP管理テーブル(140)に記録する。WP管理テーブル(140)は、命令コードごとに過去にライトプロテクション違反が発生したか否かを記憶する。JITコンパイラ(150)は、過去にライトプロテクション違反が発生していない命令コードについてはライトプロテクションを設定することを前提としたWPコードを生成し、過去にライトプロテクション違反が発生した命令コードについてはライトプロテクションを設定することを前提としない非WPコードを生成する。

明 細 書

コンパイル装置、コンパイル方法、およびコード生成プログラム

技術分野

[0001] 本発明は、プログラムの実行時に当該プログラムを構成する命令コードをコンパイルして変換コードを生成するコンパイル装置、コンパイル方法、およびコード生成プログラムに関し、特に、動的コンパイラによるコンパイルの効率化を図り、プログラムを高速に実行することができるコンパイル装置、コンパイル方法、およびコード生成プログラムに関する。

背景技術

[0002] 従来、例えばコンピュータなどでソフトウェアを動作させる際には、プログラミング言語で記述された命令コードがインタプリタによって解釈されたり、コンパイラでコンパイルされたりするのが一般的である。インタプリタは、命令コードを1つ1つ解釈しながら実行するのに対し、コンパイラは、命令コードをあらかじめまとめてコンパイルし、変換コードを生成しておく。このため、コンパイラによれば、プログラムの実行時に変換コードを直接実行することができ、実行速度を速めることができる。

[0003] また、近年では、例えば特許文献1に記載されたように、動的コンパイラが注目されている。動的コンパイラは、JIT (Just In Time) コンパイラとも呼ばれ、プログラムの実行時に、実行回数が所定の閾値以上となった命令コードのまとまりをコンパイルして実行するようになっている。したがって、上述した通常のコンパイラと比較すると、OS (Operating System) やCPU (Central Processing Unit) に依存しない命令コードの形式でソフトウェアを配布することが可能となるとともに、インタプリタと比較すると、実行回数が多い命令コードをコンパイルする分だけ実行速度を速めることができる。

[0004] このようなJITコンパイラにおいては、頻繁に処理される命令コードをコンパイルしながら、コンパイルにより得られる変換コードを実行するため、変換コードのコンパイル元となる命令コードが変更された場合には、コンパイル元の命令コードとコンパイル先の変換コードとが指示する処理内容が異なってしまう。そして、元の命令コードと処理内容が異なってしまうと、変換コードをそのまま実行することはできない。そこで、

コンパイル元の命令コードが変更されたことを検出するために、ライトプロテクション方式と非ライトプロテクション方式の2通りのコンパイル方式が考えられている。

[0005] ライトプロテクション方式は、図1の上図に示すように、JITコンパイラが命令コードをコンパイルする際、コンパイル元の命令コードが存在するページにライトプロテクションを設定する方式である。ライトプロテクションによって、ハードウェア的に書き込み保護されることにより、一度コンパイルされた命令コードが存在するページに対する書き込みが行われる場合には、ライトプロテクション違反が発生し、確実に命令コードの変更を検出することができる。

[0006] 一方、非ライトプロテクション方式は、図1の下図に示すように、JITコンパイラが命令コードをコンパイルする際、特に書き込みを行うストア命令については、書き込み先に既にコンパイル済みの命令コードがあるか否かを判定する判定コードを埋め込む方式である。この判定コードにより、変換コード内のストア命令が実行される際には、一度コンパイルされた命令コードが変更されるか否かを判定することができ、結果として、命令コードの変更を検出することができる。

[0007] 特許文献1:特開2000-132408号公報

発明の開示

発明が解決しようとする課題

[0008] ところで、上述したライトプロテクション方式および非ライトプロテクション方式を比較すると、ライトプロテクション方式は、変換コードに判定コードが埋め込まれることがないため、非ライトプロテクション方式よりも変換コードを高速に動作させることができる。その反面、ライトプロテクション方式は、命令コードが存在するページ全体にライトプロテクションを設定するため、ライトプロテクション違反が発生した場合には、ページ全体について改めてコンパイルし直すなどの必要が生じる。

[0009] このため、一般には、命令コードの実行回数が所定の閾値以上となると、最初はライトプロテクション方式で命令コードのコンパイルを行い、ライトプロテクション違反が発生すると、改めて非ライトプロテクション方式で命令コードのコンパイルを行うことが考えられている。こうすることで、ライトプロテクション違反が発生しないページについては、ライトプロテクション方式の変換コードを実行して高速動作を実現しつつ、ライト

プロテクション違反が発生したページについては、ページの部分的な削除などが可能な非ライトプロテクション方式の変換コードが得られることになる。

[0010] しかしながら、ライトプロテクション方式が適用される命令コードが存在するページがコンパイル後に変更される可能性が高い場合には、ライトプロテクション方式の変換コードを生成する処理が無駄になることがあるという問題がある。すなわち、ライトプロテクション方式の変換コードを生成しても、一度ライトプロテクション違反が発生すれば、このライトプロテクション方式の変換コードは破棄されて、改めて非ライトプロテクション方式の変換コードが生成される。このため、ライトプロテクション方式の変換コード生成後、すぐにライトプロテクション違反が発生すれば、ライトプロテクション方式の変換コードを生成する処理が無駄になり、効率が悪い。

[0011] 特に、ライトプロテクション方式が適用される命令コードが存在するページに、命令コードとデータが混在している場合には、データは頻繁に書き換えられると考えられるため、ライトプロテクション方式の変換コードがまったく実行されない可能性もある。そして、実行されないライトプロテクション方式の変換コードが生成された後に、非ライトプロテクション方式の変換コードが改めて生成されるため、コンパイルによる実行速度の高速化にも支障が生じる。

[0012] 本発明はかかる点に鑑みてなされたものであり、動的コンパイラによるコンパイルの効率化を図り、プログラムを高速に実行することができるコンパイル装置、コンパイル方法、およびコード生成プログラムを提供することを目的とする。

課題を解決するための手段

[0013] 上記課題を解決するために、本発明は、プログラムの実行時に当該プログラムを構成する命令コードをコンパイルして変換コードを生成するコンパイル装置であって、命令コードごとの実行回数が所定の閾値以上であるか否かを判定する判定手段と、前記判定手段による判定の結果、実行回数が所定の閾値未満である命令コードを実行する実行手段と、前記実行手段によって実行される命令コードに対する書き込み保護違反が発生するか否かを監視する監視手段と、前記判定手段による判定の結果、実行回数が所定の閾値以上である命令コードから前記監視手段の監視結果に応じて異なる変換コードを生成する生成手段とを有することを特徴とする。

- [0014] また、本発明は、上記発明において、前記生成手段は、書き込み保護違反が発生した命令コードから、データの書き込み先が既にコンパイル済みであるか否かを判定する判定コードが埋め込まれた非ライトプロテクション方式の変換コードを生成することを特徴とする。
- [0015] また、本発明は、上記発明において、前記生成手段は、書き込み保護違反が発生しなかった命令コードを書き込み保護しつつ、この命令コードをコンパイルしてライトプロテクション方式の変換コードを生成することを特徴とする。
- [0016] また、本発明は、上記発明において、前記監視手段は、前記実行手段によって初めて実行される命令コードに書き込み保護を設定する設定手段と、前記設定手段によって設定された書き込み保護の違反が発生したか否かを記憶する記憶手段とを含むことを特徴とする。
- [0017] また、本発明は、上記発明において、前記判定手段による判定の結果、命令コードの実行回数が所定の閾値以上である場合に、この命令コードから前記生成手段によって生成された変換コードを実行する変換コード実行手段をさらに有することを特徴とする。
- [0018] また、本発明は、プログラムの実行時に当該プログラムを構成する命令コードをコンパイルして変換コードを生成するコンパイル方法であって、命令コードごとの実行回数が所定の閾値以上であるか否かを判定する判定工程と、前記判定工程における判定の結果、実行回数が所定の閾値未満である命令コードを実行する実行工程と、前記実行工程にて実行される命令コードに対する書き込み保護違反が発生するか否かを監視する監視工程と、前記判定手段における判定の結果、実行回数が所定の閾値以上である命令コードから前記監視工程の監視結果に応じて異なる変換コードを生成する生成工程とを有することを特徴とする。
- [0019] また、本発明は、上記発明において、前記生成工程は、書き込み保護違反が発生した命令コードから、データの書き込み先が既にコンパイル済みであるか否かを判定する判定コードが埋め込まれた非ライトプロテクション方式の変換コードを生成することを特徴とする。
- [0020] また、本発明は、上記発明において、前記生成工程は、書き込み保護違反が発生

しなかった命令コードを書き込み保護しつつ、この命令コードをコンパイルしてライトプロテクション方式の変換コードを生成することを特徴とする。

[0021] また、本発明は、上記発明において、前記監視工程は、前記実行工程にて初めて実行される命令コードに書き込み保護を設定する設定工程と、前記設定工程にて設定された書き込み保護の違反が発生したか否かを命令コードごとのフラグとして記録する記録工程とを含むことを特徴とする。

[0022] また、本発明は、上記発明において、前記判定工程における判定の結果、命令コードの実行回数が所定の閾値以上である場合に、この命令コードから前記生成工程にて生成された変換コードを実行する変換コード実行工程をさらに有することを特徴とする。

[0023] また、本発明は、プログラムの実行時に当該プログラムを構成する命令コードをコンパイルして変換コードを生成するコンピュータによって実行されるコード生成プログラムであって、前記コンピュータに、命令コードごとの実行回数が所定の閾値以上であるか否かを判定する判定工程と、前記判定工程における判定の結果、実行回数が所定の閾値未満である命令コードを実行する実行工程と、前記実行工程にて実行される命令コードに対する書き込み保護違反が発生するか否かを監視する監視工程と、前記判定手段における判定の結果、実行回数が所定の閾値以上である命令コードから前記監視工程の監視結果に応じて異なる変換コードを生成する生成工程とを実行させることを特徴とする。

[0024] また、本発明は、上記発明において、前記生成工程は、書き込み保護違反が発生した命令コードから、データの書き込み先が既にコンパイル済みであるか否かを判定する判定コードが埋め込まれた非ライトプロテクション方式の変換コードを生成することを特徴とする。

[0025] また、本発明は、上記発明において、前記生成工程は、書き込み保護違反が発生しなかった命令コードを書き込み保護しつつ、この命令コードをコンパイルしてライトプロテクション方式の変換コードを生成することを特徴とする。

[0026] また、本発明は、上記発明において、前記監視工程は、前記実行工程にて初めて実行される命令コードに書き込み保護を設定する設定工程と、前記設定工程にて設

定された書き込み保護の違反が発生したか否かを命令コードごとのフラグとして記録する記録工程とを含むことを特徴とする。

- [0027] また、本発明は、上記発明において、前記判定工程における判定の結果、命令コードの実行回数が所定の閾値以上である場合に、この命令コードから前記生成工程にて生成された変換コードを実行する変換コード実行工程をさらに前記コンピュータに実行させることを特徴とする。

発明の効果

- [0028] 本発明によれば、命令コードごとの実行回数が所定の閾値以上であるか否かを判定し、判定の結果、実行回数が所定の閾値未満である命令コードを実行し、実行される命令コードに対する書き込み保護違反が発生するか否かを監視し、実行回数が所定の閾値以上である命令コードから監視結果に応じて異なる変換コードを生成する。このため、実行回数が所定の閾値未満である間に書き込み保護違反が発生する可能性が高い命令コードであるか否かを判断することができ、書き込み保護違反が発生する可能性が高い命令コードの実行回数が所定の閾値以上となった際には、最初から書き込み保護違反が発生することを前提としたコンパイルを実行することができる。結果として、命令コードのコンパイル時に、すぐに破棄される可能性がある変換コードの生成を省くことができ、動的コンパイラによるコンパイルの効率化を図り、プログラムを高速に実行することができる。
- [0029] また、本発明によれば、書き込み保護違反が発生した命令コードから、データの書き込み先が既にコンパイル済みであるか否かを判定する判定コードが埋め込まれた非ライトプロテクション方式の変換コードを生成する。このため、実行回数が所定の閾値に到達した後も書き込み保護違反が発生する可能性が高い命令コードに関しては、最初からライトプロテクション以外の方法で命令コードの変更が検知されるようにコンパイルすることができ、無駄なコンパイルを防止することができる。
- [0030] また、本発明によれば、書き込み保護違反が発生しなかった命令コードを書き込み保護しつつ、この命令コードをコンパイルしてライトプロテクション方式の変換コードを生成する。このため、実行回数が所定の閾値に到達した後に書き込み保護違反が発生する可能性が低い命令コードに関しては、ライトプロテクションを前提とした方法で

命令コードの変更が検知されるようにコンパイルすることができ、コンパイル後の変換コードを短くしてプログラムの実行速度を向上することができる。

[0031] また、本発明によれば、初めて実行される命令コードに書き込み保護を設定し、設定された書き込み保護の違反が発生したか否かを記憶するため、命令コードの実行回数が所定の閾値に到達するまでの間、書き込み保護違反が発生するか否かを確実に監視することができる。

[0032] また、本発明によれば、命令コードの実行回数が所定の閾値以上である場合に、この命令コードから生成された変換コードを実行するため、頻繁に実行される処理については、コンパイル後の変換コードを実行することができ、プログラムの実行速度を向上することができる。

図面の簡単な説明

[0033] [図1]図1は、2通りのコンパイル方式を説明する図である。

[図2]図2は、本発明の一実施の形態に係るコンパイル装置の要部構成を示すブロック図である。

[図3]図3は、一実施の形態に係る変換コードテーブルの例を示す図である。

[図4]図4は、一実施の形態に係るプロファイル情報テーブルの例を示す図である。

[図5]図5は、一実施の形態に係るWP管理テーブルの例を示す図である。

[図6]図6は、一実施の形態に係るインタプリタの動作を示すフロー図である。

[図7]図7は、一実施の形態に係るプロファイル処理を示すフロー図である。

[図8]図8は、一実施の形態に係るJITコンパイラの動作を示すフロー図である。

符号の説明

- [0034] 110 インタプリタ
- 111 命令フェッチ部
- 112 変換コード実行部
- 113 プロファイラ部
- 114 命令実行部
- 115 閾値判定部
- 116 WP設定部

120 変換コードテーブル

130 プロファイル情報テーブル

140 WP管理テーブル

150 JITコンパイラ

151 WP判定部

152 WPコード生成部

153 非WPコード生成部

発明を実施するための最良の形態

[0035] 本発明の骨子は、命令コードの実行回数が所定の閾値に達するまでの間に、この命令コードに対するライトプロテクション違反が発生するか否かを監視しておき、命令コードの実行回数が所定の閾値に達した際には、監視中におけるライトプロテクション違反の発生の有無に応じて生成する変換コードを決定することである。以下、本発明の一実施の形態について、図面を参照して詳細に説明する。

[0036] 図2は、本発明の一実施の形態に係るコンパイル装置の要部構成を示すブロック図である。同図に示すコンパイル装置は、インタプリタ110、変換コードテーブル120、プロファイル情報テーブル130、ライトプロテクション(Write Protection:以下「WP」と略記することもある)管理テーブル140、およびJITコンパイラ150を有している。

[0037] インタプリタ110は、実行対象のプログラムから命令コードを取り出して、そのまま実行したり、コンパイル後の変換コードを実行したりする。また、インタプリタ110は、取り出した命令コードに対するライトプロテクション(WP)の設定・解除をするとともに、ライトプロテクション違反の発生の有無をWP管理テーブル140に記録する。

[0038] 具体的には、インタプリタ110は、命令フェッチ部111、変換コード実行部112、プロファイラ部113、命令実行部114、閾値判定部115、およびWP設定部116を有している。

[0039] 命令フェッチ部111は、実行対象のプログラムから命令コードを取り出して(フェッチして)、変換コード実行部112へ出力する。

[0040] 変換コード実行部112は、命令コードに対応する変換コードを変換コードテーブル120から検索し、変換コードが変換コードテーブル120に記憶されていれば、この変

換コードを実行する。また、変換コード実行部112は、変換コードが変換コードテーブル120に記憶されていないならば、命令コードをプロファイラ部113へ出力する。さらに、変換コード実行部112は、閾値判定部115から変換コードの実行を指示された場合に、JITコンパイラ150によって変換コードが生成されるのを待機した後、変換コードテーブル120から変換コードを読み出して実行する。

- [0041] プロファイラ部113は、命令コードがプロファイル対象のものであるか否かを判断し、プロファイル対象の命令コードであれば、この命令コードの実行回数をプロファイル情報テーブル130に記録する。すなわち、プロファイラ部113は、変換コード実行部112から命令コードが出力されるたびにこの命令コードが実行されるものとして、プロファイル情報テーブル130に記憶された実行回数を1インクリメントする。そして、プロファイラ部113は、命令コードを命令実行部114およびWP設定部116へ出力する。
- [0042] 命令実行部114は、閾値判定部115から命令コードの実行を指示された場合に、プロファイラ部113から出力された命令コードを解釈して実行する。
- [0043] 閾値判定部115は、プロファイル情報テーブル130において命令コードの実行回数が更新されると、この命令コードの実行回数が所定の閾値以上となったか否かを判定する。そして、閾値判定部115は、実行回数が所定の閾値以上となった命令コードについて、変換コードを生成するようにJITコンパイラ150へ指示する。また、閾値判定部115は、更新された命令コードの実行回数が所定の閾値未満であれば、命令実行部114に対して命令コードの実行を指示し、更新された命令コードの実行回数が所定の閾値以上であれば、変換コード実行部112に対して変換コードの実行を指示する。
- [0044] WP設定部116は、プロファイラ部113から出力された命令コードが存在するプログラムのページに対するライトプロテクションの有無とWP管理テーブル140におけるWPフラグとから、命令コードが存在するページにライトプロテクションを設定する。すなわち、WP設定部116は、命令コードが存在するページにライトプロテクションが未設定であり、かつ、後述するWPフラグが「偽」である場合に、この命令コードが存在するページにライトプロテクションを設定する。WPフラグは、過去にライトプロテクション違反が発生していれば「真」とされ、過去にライトプロテクション違反が発生していなければ「偽」とされる。

ば「偽」とされている。

- [0045] また、WP設定部116は、実行対象のプログラムに対するライトプロテクション違反を検知し、ライトプロテクション違反が発生したページに対するライトプロテクションを解除するとともに、WP管理テーブル140において、このページの命令コードに対応するWPフラグを「真」に設定する。
- [0046] このように、インタプリタ110は、命令コードおよび変換コードの実行やライトプロテクションの設定・解除を行う。
- [0047] 変換コードテーブル120は、命令コードのアドレス(命令アドレス)に対応付けてJITコンパイラ150によって生成された変換コードを記憶する。なお、本実施の形態においては、命令コードが存在するページにライトプロテクションを設定することを前提としたライトプロテクション方式の変換コード(以下「WPコード」と略記する)、または変換コード中に所定の判定コードを埋め込んで命令コードの書き換えを検知する非ライトプロテクション方式の変換コード(以下「非WPコード」と略記する)が生成される。
- [0048] このため、変換コードテーブル120は、例えば図3に示すように、命令アドレス「aaa a」および「bbbb」に対応付けてそれぞれ「非WPコード」および「WPコード」を記憶している。また、図3において、命令アドレス「cccc」および「dddd」に対応する変換コードが記憶されていないが、これらの命令コードについては、実行回数が所定の閾値以上となっていないため、変換コードが生成されていないことを示す。
- [0049] プロファイル情報テーブル130は、命令コードごとの実行回数を記憶する。すなわち、プロファイル情報テーブル130は、例えば図4に示すように、命令アドレス「aaaa」から「dddd」に対応付けて、それぞれの命令コードの実行回数を記憶している。
- [0050] WP管理テーブル140は、命令コードごとに過去に一度でもライトプロテクション違反が発生したか否かを記憶する。すなわち、WP管理テーブル140は、例えば図5に示すように、命令アドレス「aaaa」から「dddd」に対応付けて、それぞれの命令コードが存在するページのライトプロテクション違反が発生したか否かを示すWPフラグを記憶している。上述したように、WPフラグが「真」であれば、この命令コードが存在するページに対するライトプロテクション違反が過去に発生したことを示し、WPフラグが「偽」であれば、この命令コードが存在するページに対するライトプロテクション違反が

過去に一度も発生していないことを示す。なお、WP管理テーブル140におけるWPフラグは、初期状態ではすべて「偽」となっている。

- [0051] JITコンパイラ150は、実行回数が所定の閾値以上となった命令コードに対応する変換コードを生成する。このとき、JITコンパイラ150は、過去にライトプロテクション違反が一度も発生していない命令コードについてはライトプロテクションを設定することを前提としたWPコードを生成し、過去にライトプロテクション違反が発生した命令コードについてはライトプロテクションを設定することを前提としない非WPコードを生成する。
- [0052] 具体的には、JITコンパイラ150は、WP判定部151、WPコード生成部152、および非WPコード生成部153を有している。
- [0053] WP判定部151は、インタプリタ110の閾値判定部115から命令コードに対応する変換コードの生成が指示されると、この命令コードが存在するページにライトプロテクションが設定されているか否かを確認する。そして、WP判定部151は、ライトプロテクションが設定されている場合には、WPコード生成部152へWPコードを生成するように指示し、ライトプロテクションが設定されていない場合には、非WPコード生成部153へ非WPコードを生成するように指示する。
- [0054] ここで、命令コードが存在するページにライトプロテクションが設定されているか否かは、この命令コードの実行回数が所定の閾値に到達するまでにライトプロテクション違反が発生したか否かに対応している。すなわち、実行回数が所定の閾値に到達するまでにライトプロテクション違反が一度も発生しなければ、命令コードにはライトプロテクションが設定されており、実行回数が所定の閾値に到達するまでにライトプロテクション違反が一度でも発生すれば、命令コードにはライトプロテクションが設定されていない。そして、実行回数が所定の閾値に到達するまでのライトプロテクション違反の有無は、WP管理テーブル140におけるWPフラグでも確認可能であるため、WP判定部151は、WP管理テーブル140のWPフラグを参照して、命令コードにライトプロテクションが設定されているか否かを判断しても良い。
- [0055] また、WP判定部151は、実行対象のプログラムに対するライトプロテクション違反を検知し、ライトプロテクション違反が発生したページに対応するWPコードの破棄をW

Pコード生成部152へ指示するとともに、このページの非WPコードの生成を非WPコード生成部153へ指示する。

- [0056] WPコード生成部152は、WP判定部151の指示に従って、実行回数が所定の閾値以上となった命令コードが存在するページをライトプロテクション方式でコンパイルしてWPコードを生成し、変換コードテーブル120に登録する。WPコード生成部152が生成するWPコードに対応する命令コードにはライトプロテクションが設定されており、命令コードの実行回数が所定の閾値に到達するまでにライトプロテクション違反が発生していない。
- [0057] また、WPコード生成部152は、ライトプロテクション違反発生時にWP判定部151からWPコードの破棄を指示されると、指示された命令アドレスに対応する変換コードを変換コードテーブル120において確認し、この変換コードがWPコードであれば破棄する。
- [0058] 非WPコード生成部153は、WP判定部151の指示に従って、実行回数が所定の閾値以上となった命令コードが存在するページを非ライトプロテクション方式でコンパイルして非WPコードを生成し、変換コードテーブル120に登録する。非WPコード生成部153が生成する非WPコードに対応する命令コードにはライトプロテクションが設定されておらず、命令コードの実行回数が所定の閾値に到達するまでにライトプロテクション違反が発生している。
- [0059] また、非WPコード生成部152は、ライトプロテクション違反発生時にWP判定部151から非WPコードの生成を指示されると、指示された命令アドレスの命令コードから非WPコードを生成し、変換コードテーブル120に登録する。
- [0060] このように、JITコンパイラ150は、命令コードの実行回数が所定の閾値に到達すると、実行回数が所定の閾値に到達するまでのライトプロテクション違反の有無に応じてWPコードおよび非WPコードのどちらを生成するか決定し、決定された変換コードを生成して変換コードテーブル120に登録する。
- [0061] 次いで、上記のように構成されたインタプリタ110の動作について、図6に示すフロー図を参照しながら説明する。
- [0062] 実行対象のプログラムを実行する際、まず、インタプリタ110の命令フェッチ部111

によって、プログラム中の命令コードがフェッチされる(ステップS101)。この命令コードは、変換コード実行部112へ出力され、変換コード実行部112によって変換コードテーブル120が検索されることにより、命令コードに対応する変換コードが既に記憶されているか否かが確認される(ステップS102)。この結果、既に変換コードが記憶されていれば(ステップS102Yes)、変換コード実行部112によって、変換コードテーブル120から変換コードが読み出され、実行される(ステップS103)。

[0063] 一方、変換コードが変換コードテーブル120に記憶されていなければ(ステップS102No)、命令コードは、プロファイラ部113へ出力され、プロファイル対象の命令コードであるか否かが判定される(ステップS104)。この結果、命令コードがプロファイル対象でなければ(ステップS104No)、命令コードは命令実行部114へ出力され、実行される(ステップS107)。

[0064] また、命令コードがプロファイル対象であれば(ステップS104Yes)、プロファイラ部113によってプロファイル情報テーブル130に記憶された命令コードの実行回数が1インクリメントされるなどのプロファイル処理が実行される(ステップS105)。また、プロファイル処理においては、WP設定部116によって、命令コードが存在するページのライトプロテクションの設定が実行される。このプロファイル時の処理については、後に詳述する。

[0065] そして、プロファイラ部113によってプロファイル情報テーブル130に記憶された実行回数が1インクリメントされると、閾値判定部115によって、更新された実行回数が所定の閾値以上であるか否かが判定される(ステップS106)。この結果、実行回数が所定の閾値未満であれば(ステップS106No)、まだ変換コードが生成されることはなため、閾値判定部115から命令実行部114へ命令を実行するように指示され、命令実行部114によってプロファイラ部113から出力される命令コードが実行される(ステップS107)。

[0066] 一方、命令コードの実行回数が所定の閾値以上であれば(ステップS106Yes)、命令コードが頻繁に実行されていることから、閾値判定部115によって、JITコンパイラ150のWP判定部151へ変換コードの生成が指示されるとともに(ステップS108)、変換コード実行部112へ変換コードの実行が指示される。そして、変換コード実行部11

2によって、JITコンパイラ150による変換コードの生成が待機され、変換コード生成後、変換コードテーブル120から変換コードが読み出されて実行される(ステップS109)。このとき、JITコンパイラ150によって生成される変換コードは、命令コードに対する過去のライトプロテクション違反の有無によって、WPコードおよび非WPコードのいずれかとなっている。

[0067] ここまでの処理により、命令フェッチ部111によってフェッチされた命令コードは、実行回数が所定の閾値未満であれば命令コードそのものとして実行され、実行回数が所定の閾値以上であれば対応する変換コードが実行されたことになる。そして、このとき、WP設定部116によって、ライトプロテクション違反が発生したか否かが監視される(ステップS110)。すなわち、WP設定部116によって、プロファイル処理時にライトプロテクションが設定されたページに対する書き込みが発生したか否かが判断され、ライトプロテクション違反が発生した場合には(ステップS110Yes)、WP設定部116によって、該当するページのライトプロテクションが解除されるとともに(ステップS111)、WP管理テーブル140において、ページ内の命令コードに対応するWPフラグが「真」に設定される(ステップS112)。

[0068] ここで、ライトプロテクション違反が発生するページに関しては、例えば命令コードとデータが混在しているなどのために、書き込みが頻繁に発生するページであると考えられ、このようなページについては、ライトプロテクション方式のWPコードを生成してもすぐに破棄する必要が生じる。

[0069] そこで、本実施の形態においては、プロファイル処理時に設定したライトプロテクションの違反が発生した場合には、このページ内の命令コードに対応するWPフラグを「真」に設定しておき、命令コードの実行回数が所定の閾値以上となって変換コードを生成する際に、WPコードを生成することなく最初から非WPコードを生成する。このため、ライトプロテクション方式のコンパイルが適していない命令コードのページについては、非WPコードが生成され、コンパイルの効率化を図り、プログラムを高速に実行することができる。なお、変換コード生成の処理については、後に詳述する。

[0070] 次に、命令フェッチ部111によってフェッチされた命令コードがプロファイル対象である場合のプロファイル時の処理について、図7に示すフロー図を参照しながら説明

する。

- [0071] プロファイラ部113において、命令コードがプロファイル対象であると判断されると、プロファイラ部113によって、この命令コードの実行回数がプロファイル情報テーブル130に記録される(ステップS201)。具体的には、プロファイラ部113によって、命令コードに対応付けて記憶された実行回数が1インクリメントされる。
- [0072] また、WP設定部116によって、命令コードが存在するページに対してライトプロテクションが既に設定されているか否かが判断され(ステップS202)、既に設定されていれば(ステップS202Yes)、ライトプロテクションを維持したままプロファイル処理が終了する。
- [0073] 一方、ライトプロテクションが設定されていなければ(ステップS202No)、WP管理テーブル140が参照され、命令コードに対応するWPフラグが「真」であるか否かが確認される(ステップS203)。ここで、WPフラグが「真」であれば(ステップS203Yes)、過去にライトプロテクション違反が発生したことを意味しており、この命令コードが存在するページに対する書き込みが比較的頻繁に実行されると考えられることから、WP設定部116によって再度ライトプロテクションが設定されることなく、プロファイル処理が終了する。また、WPフラグが「偽」であれば(ステップS203No)、WP設定部116によって、この命令コードが存在するページにライトプロテクションが設定される(ステップS204)。
- [0074] このようなプロファイル処理により、プロファイル対象の命令コードの初回実行時には、必ずライトプロテクションが設定され、その後、一度でもライトプロテクション違反が発生した命令コードについては、ライトプロテクションは解除されたままで、再びライトプロテクションが設定されることはない。同時に、一度でもライトプロテクション違反が発生した命令コードについては、WP管理テーブル140におけるWPフラグが「真」とされたままとなる。
- [0075] このため、命令コードの実行回数が所定の閾値に到達した際、ライトプロテクションの設定の有無またはWP管理テーブル140のWPフラグを参照することにより、それぞれの命令コードが比較的頻繁に書き込みが実行されるページ内の命令コードであるか否かを判定することが可能となる。したがって、次に説明するように、ライトプロテ

クションが設定されていないか、またはWPフラグが「真」である命令コードについては、再びページ内に書き込みが発生することを想定して最初から非WPコードを生成し、コンパイルの効率化を図ることが可能となる。

[0076] 図8は、本実施の形態に係るJITコンパイラ150の動作を示すフロー図である。JITコンパイラ150は、命令コードの実行回数が所定の閾値以上となった時に、この命令コードに対応する変換コードを生成する。

[0077] すなわち、インタプリタ110の閾値判定部115によって命令コードの実行回数が所定の閾値以上であると判定されると、変換コード生成の指示が出され、この指示がJITコンパイラ150のWP判定部151によって取得される(ステップS301)。この指示を受け、WP判定部151によって、コンパイル対象の命令コードが存在するページにライトプロテクションが設定されているか否かが判定される(ステップS302)。上述したように、WP判定部151による判定は、WP管理テーブル140においてコンパイル対象の命令コードに対応するWPフラグが「真」であるか否かの判定でも良い。

[0078] この判定の結果、ライトプロテクションが設定されている、もしくはWPフラグが「偽」である場合(ステップS302Yes)、この命令コードが存在するページについては、過去に一度もライトプロテクション違反が発生していないことになる。このため、以後も命令コードが存在するページに対する書き込みが比較的発生しにくいと考えられ、WP判定部151によって、WPコード生成部152へ変換コードの生成が指示される。これにより、WPコード生成部152によって、命令コードが存在するページに対するライトプロテクション方式のコンパイルが行われ、WPコードが生成される(ステップS303)。このWPコードは、元の命令コードのアドレスに対応付けられ、変換コードテーブル120に記憶される。

[0079] 一方、WP判定部151における判定の結果、ライトプロテクションが設定されていない、もしくはWPフラグが「真」である場合(ステップS302No)、この命令コードが存在するページについては、過去にライトプロテクション違反が発生していることになる。このため、以後も命令コードが存在するページに対する書き込みが比較的発生しやすいと考えられ、WP判定部151によって、非WPコード生成部153へ変換コードの生成が指示される。これにより、非WPコード生成部153によって、命令コードが存在す

るページに対する非ライトプロテクション方式のコンパイルが行われ、非WPコードが生成される(ステップS304)。この非WPコードは、元の命令コードの命令アドレスに対応付けられ、変換コードテーブル120に記憶される。

[0080] このようにして生成された変換コードは、変換コードテーブル120に登録後、既に説明したように、インタプリタ110の変換コード実行部112によって実行される。本実施の形態においては、特に非WPコードが生成される場合に、一度WPコードが生成されてから改めて非WPコードが生成されるということがなく、変換コードが実行されるまでの時間の短縮を図ることができる。結果として、プログラムを高速に実行することができる。

[0081] WPコードまたは非WPコードの生成後、JITコンパイラ150においては、WP判定部151によって、常にライトプロテクション違反の発生が監視される(ステップS305)。そして、ライトプロテクション違反が発生した場合には(ステップS305Yes)、WP判定部151によって、ライトプロテクション違反が発生したページのWPコードを破棄する旨の指示がWPコード生成部152へ出される。この指示を受け、WPコード生成部152によって、ライトプロテクション違反が発生したページ内の命令コードに対応付けて変換コードテーブル120に記憶された変換コードが確認され、この変換コードがWPコードであれば破棄される(ステップS306)。

[0082] 同時に、WP判定部151によって、ライトプロテクション違反が発生したページの非WPコードを生成する旨の指示が非WPコード生成部153へ出される。この指示を受け、非WPコード生成部153によって、ライトプロテクション違反が発生したページの非WPコードが生成され(ステップS307)、変換コードテーブル120に登録される。

[0083] これにより、実行回数が所定の閾値に到達するまでにライトプロテクション違反が発生しなかったことからWPコードが生成された命令コードについても、WPコードの生成後、ライトプロテクション違反が発生した場合には、改めて非WPコードが生成される。したがって、命令コードと変換コードの処理内容が異なってしまうことがなく、確実に命令コードと変換コードの処理内容を一致させることができる。

[0084] 以上のように、本実施の形態によれば、命令コードの初回実行時にこの命令コードにライトプロテクションを設定し、命令コードの実行回数が所定の閾値に到達するま

でライトプロテクション違反が発生しなければライトプロテクション方式で命令コードをコンパイルし、ライトプロテクション違反が発生すれば非ライトプロテクション方式で命令コードをコンパイルする。このため、命令コードの実行回数が所定の閾値に到達して変換コードを生成する際、最初から適した方式で命令コードをコンパイルすることができ、動的コンパイラによるコンパイルの効率化を図り、プログラムを高速に実行することができる。

産業上の利用可能性

[0085] 本発明は、動的コンパイラによるコンパイルの効率化を図り、プログラムを高速に実行する場合に適用することができる。

請求の範囲

- [1] プログラムの実行時に当該プログラムを構成する命令コードをコンパイルして変換コードを生成するコンパイル装置であって、
命令コードごとの実行回数が所定の閾値以上であるか否かを判定する判定手段と、
、
前記判定手段による判定の結果、実行回数が所定の閾値未満である命令コードを実行する実行手段と、
前記実行手段によって実行される命令コードに対する書き込み保護違反が発生するか否かを監視する監視手段と、
前記判定手段による判定の結果、実行回数が所定の閾値以上である命令コードから前記監視手段の監視結果に応じて異なる変換コードを生成する生成手段と
を有することを特徴とするコンパイル装置。
- [2] 前記生成手段は、
書き込み保護違反が発生した命令コードから、データの書き込み先が既にコンパイル済みであるか否かを判定する判定コードが埋め込まれた非ライトプロテクション方式の変換コードを生成することを特徴とする請求項1記載のコンパイル装置。
- [3] 前記生成手段は、
書き込み保護違反が発生しなかった命令コードを書き込み保護しつつ、この命令コードをコンパイルしてライトプロテクション方式の変換コードを生成することを特徴とする請求項1記載のコンパイル装置。
- [4] 前記監視手段は、
前記実行手段によって初めて実行される命令コードに書き込み保護を設定する設定手段と、
前記設定手段によって設定された書き込み保護の違反が発生したか否かを記憶する記憶手段と
を含むことを特徴とする請求項1記載のコンパイル装置。
- [5] 前記判定手段による判定の結果、命令コードの実行回数が所定の閾値以上である場合に、この命令コードから前記生成手段によって生成された変換コードを実行する

変換コード実行手段をさらに有することを特徴とする請求項1記載のコンパイル装置
。

- [6] プログラムの実行時に当該プログラムを構成する命令コードをコンパイルして変換コードを生成するコンパイル方法であって、
命令コードごとの実行回数が所定の閾値以上であるか否かを判定する判定工程と、
、
前記判定工程における判定の結果、実行回数が所定の閾値未満である命令コードを実行する実行工程と、
前記実行工程にて実行される命令コードに対する書き込み保護違反が発生するか否かを監視する監視工程と、
前記判定手段における判定の結果、実行回数が所定の閾値以上である命令コードから前記監視工程の監視結果に応じて異なる変換コードを生成する生成工程と
を有することを特徴とするコンパイル方法。
- [7] 前記生成工程は、
書き込み保護違反が発生した命令コードから、データの書き込み先が既にコンパイル済みであるか否かを判定する判定コードが埋め込まれた非ライトプロテクション方式の変換コードを生成することを特徴とする請求項6記載のコンパイル方法。
- [8] 前記生成工程は、
書き込み保護違反が発生しなかった命令コードを書き込み保護しつつ、この命令コードをコンパイルしてライトプロテクション方式の変換コードを生成することを特徴とする請求項6記載のコンパイル方法。
- [9] 前記監視工程は、
前記実行工程にて初めて実行される命令コードに書き込み保護を設定する設定工程と、
前記設定工程にて設定された書き込み保護の違反が発生したか否かを命令コードごとのフラグとして記録する記録工程と
を含むことを特徴とする請求項6記載のコンパイル方法。
- [10] 前記判定工程における判定の結果、命令コードの実行回数が所定の閾値以上で

ある場合に、この命令コードから前記生成工程にて生成された変換コードを実行する変換コード実行工程をさらに有することを特徴とする請求項6記載のコンパイル方法。

[11] プログラムの実行時に当該プログラムを構成する命令コードをコンパイルして変換コードを生成するコンピュータによって実行されるコード生成プログラムであって、前記コンピュータに、

命令コードごとの実行回数が所定の閾値以上であるか否かを判定する判定工程と、

前記判定工程における判定の結果、実行回数が所定の閾値未満である命令コードを実行する実行工程と、

前記実行工程にて実行される命令コードに対する書き込み保護違反が発生するか否かを監視する監視工程と、

前記判定手段における判定の結果、実行回数が所定の閾値以上である命令コードから前記監視工程の監視結果に応じて異なる変換コードを生成する生成工程と
を実行させることを特徴とするコード生成プログラム。

[12] 前記生成工程は、

書き込み保護違反が発生した命令コードから、データの書き込み先が既にコンパイル済みであるか否かを判定する判定コードが埋め込まれた非ライトプロテクション方式の変換コードを生成することを特徴とする請求項11記載のコード生成プログラム。

[13] 前記生成工程は、

書き込み保護違反が発生しなかった命令コードを書き込み保護しつつ、この命令コードをコンパイルしてライトプロテクション方式の変換コードを生成することを特徴とする請求項11記載のコード生成プログラム。

[14] 前記監視工程は、

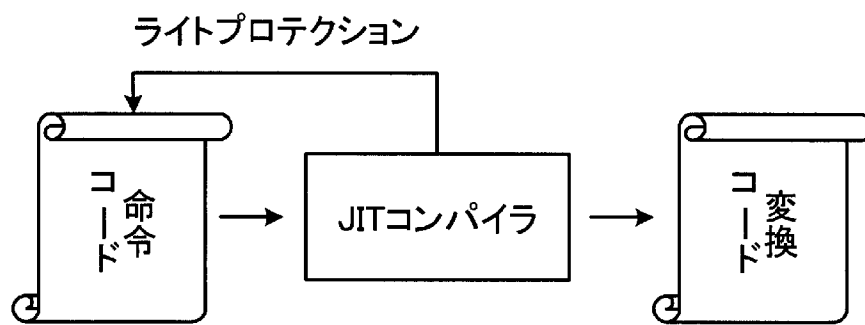
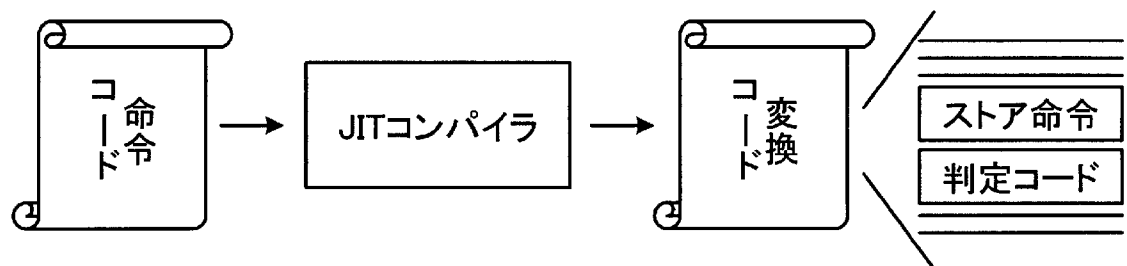
前記実行工程にて初めて実行される命令コードに書き込み保護を設定する設定工程と、

前記設定工程にて設定された書き込み保護の違反が発生したか否かを命令コードごとのフラグとして記録する記録工程と

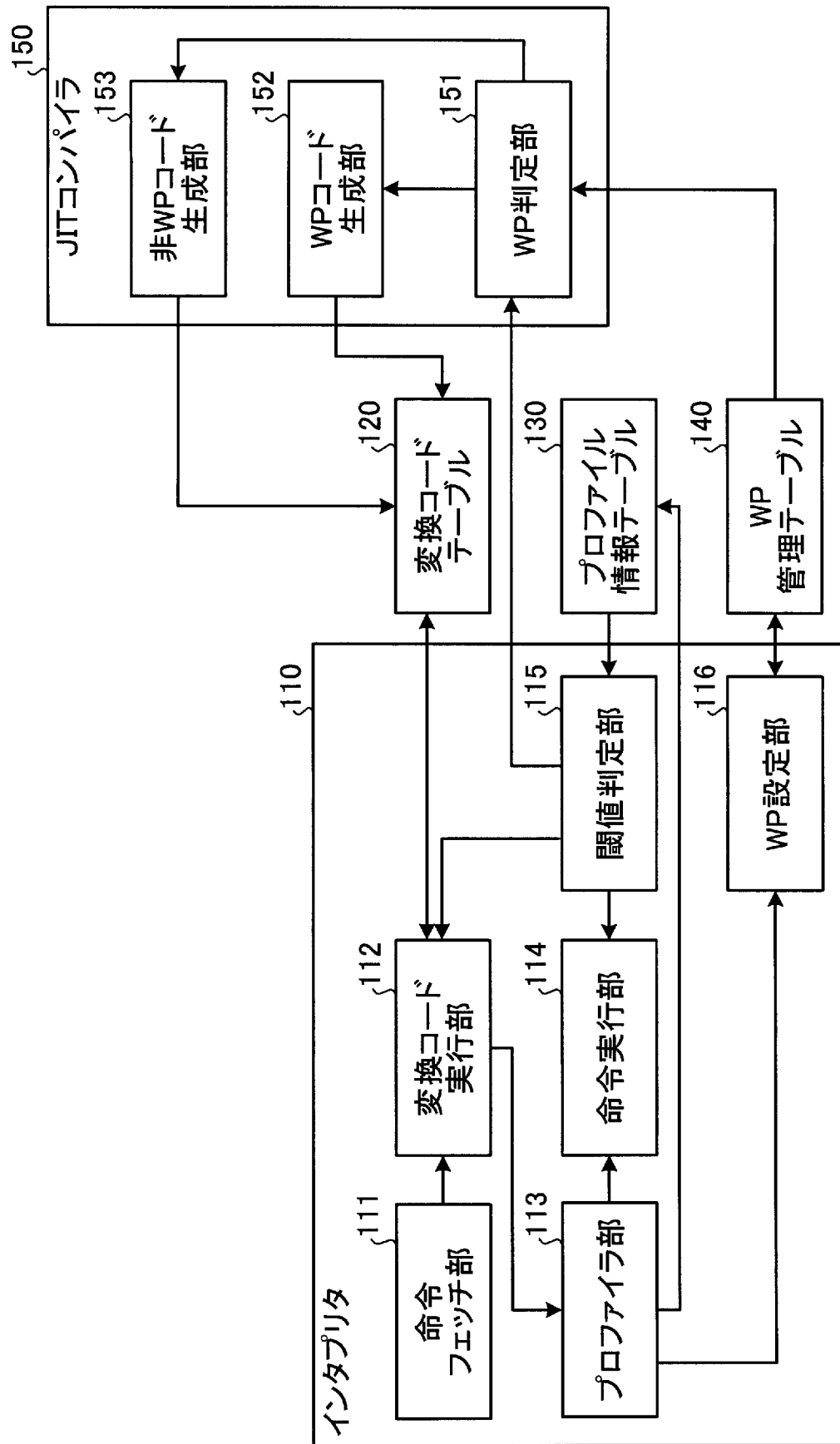
を含むことを特徴とする請求項11記載のコード生成プログラム。

- [15] 前記判定工程における判定の結果、命令コードの実行回数が所定の閾値以上である場合に、この命令コードから前記生成工程にて生成された変換コードを実行する変換コード実行工程をさらに前記コンピュータに実行させることを特徴とする請求項11記載のコード生成プログラム。

[図1]

ライトプロテクション方式非ライトプロテクション方式

[図2]



[図3]

命令アドレス	変換コード
aaaa	非WPコード
bbbb	WPコード
cccc	—
dddd	—
⋮	

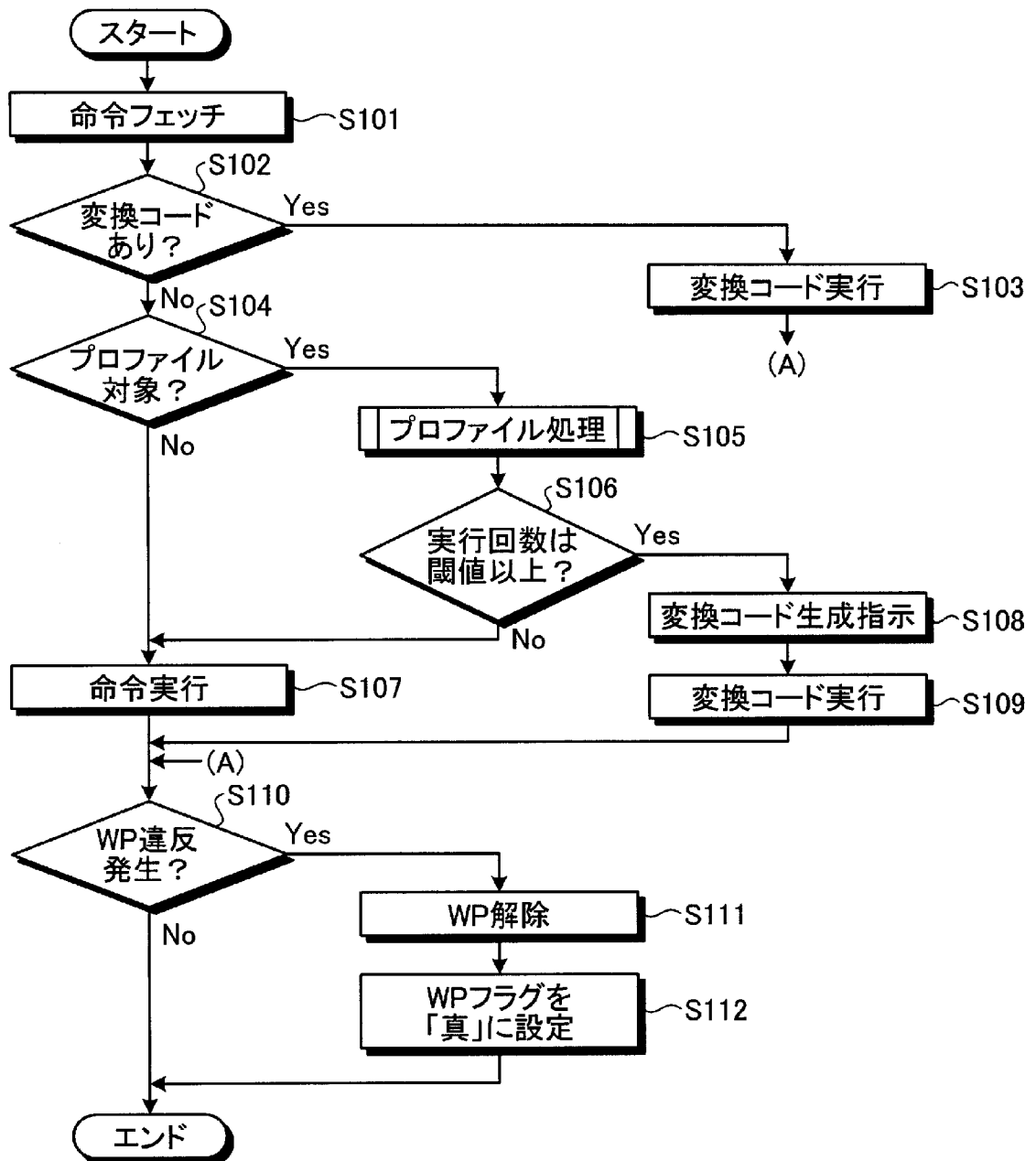
[図4]

命令アドレス	実行回数
aaaa	1000
bbbb	2000
cccc	10
dddd	20
⋮	

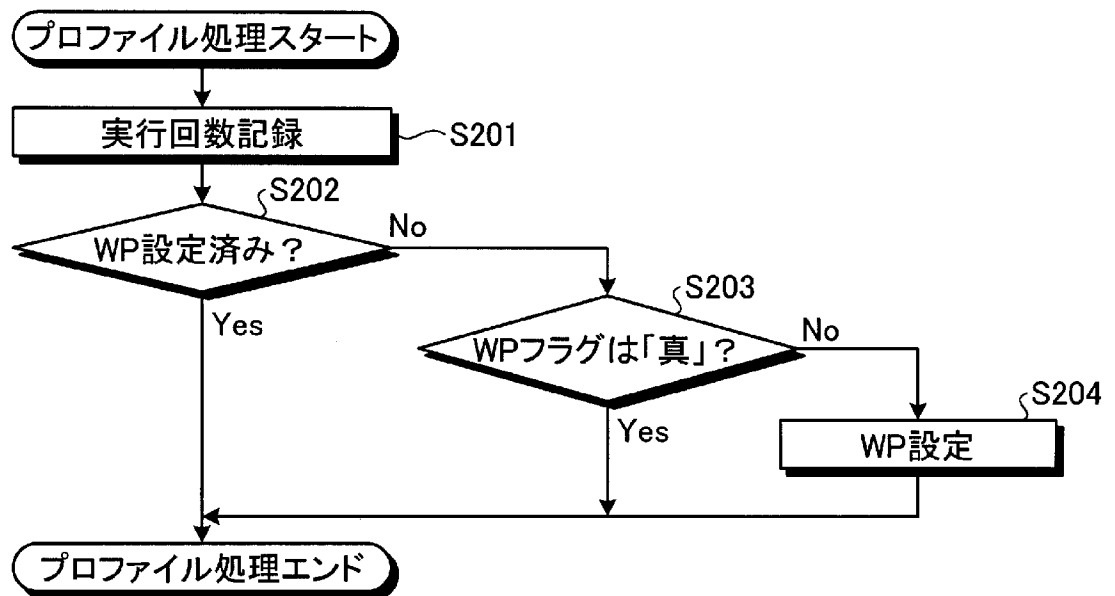
[図5]

命令アドレス	WPフラグ
aaaa	真
bbbb	偽
cccc	偽
dddd	真
⋮	

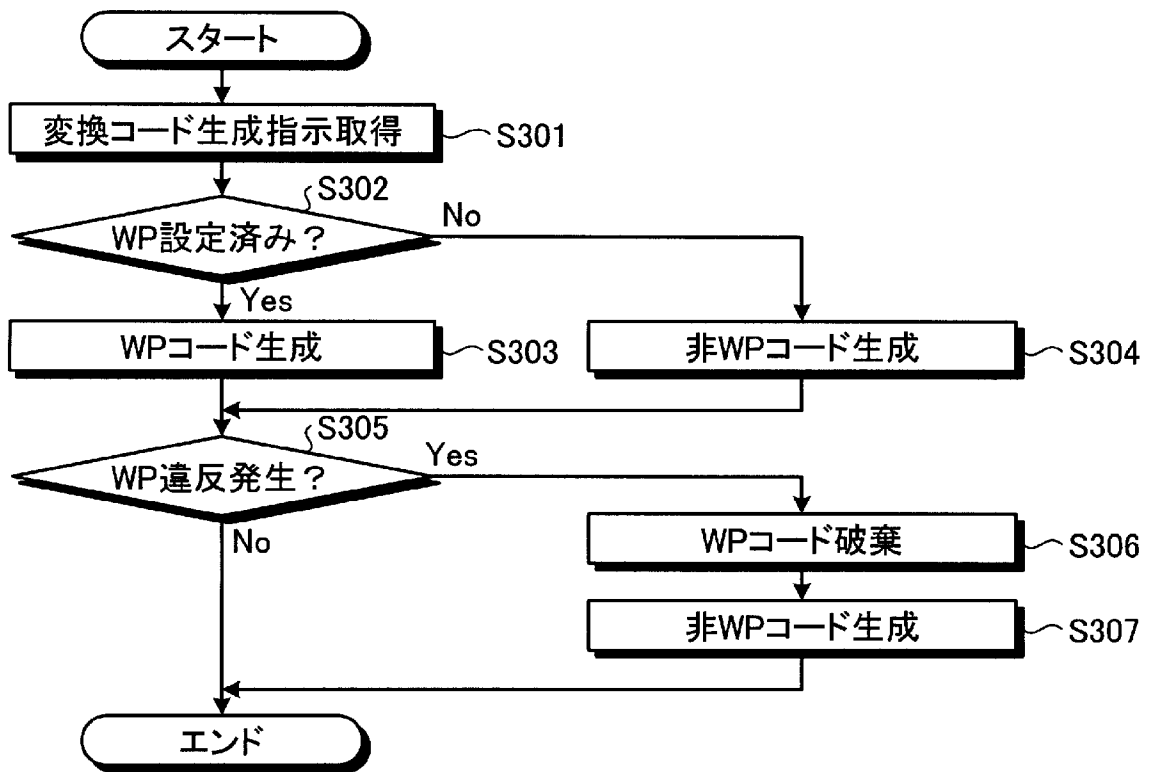
[図6]



[図7]



[図8]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2007/057233

A. CLASSIFICATION OF SUBJECT MATTER

G06F9/45 (2006.01) i

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)

G06F9/45

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2007

Kokai Jitsuyo Shinan Koho 1971-2007 Toroku Jitsuyo Shinan Koho 1994-2007

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CSDB (Nipponkoku Tokkyocho)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Motoaki SATOYAMA et al., 'Kumikomi System-yo JavaOS', Computer System Symposium Ronbunshu, 19 November, 1998 (19.11.98), Vol.98, No.15, pages 47 to 54, ISSN:1344-0640, page 52, right column, lines 39 to 40, page 53, left column, lines 26 to 28	1-15

☐ Further documents are listed in the continuation of Box C.☐ 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 application or patent 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

"&" document member of the same patent family

Date of the actual completion of the international search

17 April, 2007 (17.04.07)

Date of mailing of the international search report

01 May, 2007 (01.05.07)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F9/45(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06F9/45

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2007年
日本国実用新案登録公報	1996-2007年
日本国登録実用新案公報	1994-2007年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C S D B（日本国特許庁）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	里山元章・他,「組込みシステム用 J a v a O S」, コンピュータシステム・シンポジウム論文集, 1998. 11. 19, V o l. 98, N o. 15, p. 47-54, I S S N : 1344-0640, 第52頁右コラム第39-40行, 第53頁左コラム第26-28行	1-15

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

17.04.2007

国際調査報告の発送日

01.05.2007

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）
郵便番号100-8915
東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

久保 光宏

5B

9189

電話番号 03-3581-1101 内線 3545