

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4592367号  
(P4592367)

(45) 発行日 平成22年12月1日(2010.12.1)

(24) 登録日 平成22年9月24日(2010.9.24)

(51) Int. Cl.

F I

<b>G06F</b>	<b>9/45</b>	<b>(2006.01)</b>	G06F	9/44	322H
<b>G06F</b>	<b>12/08</b>	<b>(2006.01)</b>	G06F	12/08	563
			G06F	12/08	565
			G06F	12/08	581

請求項の数 9 (全 17 頁)

(21) 出願番号 特願2004-262490 (P2004-262490)  
 (22) 出願日 平成16年9月9日(2004.9.9)  
 (65) 公開番号 特開2006-79338 (P2006-79338A)  
 (43) 公開日 平成18年3月23日(2006.3.23)  
 審査請求日 平成18年2月22日(2006.2.22)

(73) 特許権者 308014341  
 富士通セミコンダクター株式会社  
 神奈川県横浜市港北区新横浜二丁目10番  
 23  
 (74) 代理人 100092152  
 弁理士 服部 毅巖  
 (72) 発明者 渡辺 学  
 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 稲垣 良一

最終頁に続く

(54) 【発明の名称】 プログラムセクションレイアウト方法およびレイアウト処理プログラム

(57) 【特許請求の範囲】

【請求項1】

プログラム実行時に一部のプログラムセクションをキャッシュメモリに格納して動作するコンピュータのプログラムセクションレイアウト方法において、

前記コンピュータが、

任意の処理機能の実行単位である複数のプログラムセクションを所定のアルゴリズムに基づく条件を満たす複数の第1プログラムセクションと前記所定のアルゴリズムに基づく条件を満たさない第2プログラムセクションとに分けるとともに前記所定のアルゴリズムに従って前記複数の第1プログラムセクションを順序付けし、

プログラム格納領域をキャッシュメモリサイズに分割した複数のセクショングループ格納領域の一のセクショングループ格納領域に一の第1プログラムセクションを配置できる場合には前記一のセクショングループ格納領域に前記一の第1プログラムセクションを配置し、前記一の第1プログラムセクションを配置できない場合には前記一のセクショングループ格納領域に空き領域を残して次のセクショングループ格納領域に前記一の第1プログラムセクションを配置して、前記複数の第1プログラムセクションを前記順序どおりに前記複数のセクショングループ格納領域に配置し、

前記空き領域に前記第2プログラムセクションを配置できる場合には前記第2プログラムセクションを前記空き領域に配置し、前記第2プログラムセクションを配置できない場合には次の配置可能な空き領域に前記第2プログラムセクションを配置し、

前記第1のプログラムセクションが配置された前記セクショングループ格納領域と前記

10

20

空き領域に前記第2プログラムセクションを配置する前の前記第2プログラムセクションとがメモリ上に占める第1メモリサイズから前記空き領域に配置された前記第2プログラムセクションがメモリ上に占める第2メモリサイズを減算したメモリサイズが所定メモリサイズ以下である場合には前記第2プログラムセクションの配置を終了する、

ことを特徴とするプログラムセクションレイアウト方法。

【請求項2】

プログラム実行時に一部のプログラムセクションをキャッシュメモリに格納して動作するコンピュータのプログラムセクションレイアウト方法において、

前記コンピュータが、

任意の処理機能の実行単位である複数のプログラムセクションを所定のアルゴリズムに基づく条件を満たす複数の第1プログラムセクションと前記所定のアルゴリズムに基づく条件を満たさない第2プログラムセクションとに分けるとともに前記所定のアルゴリズムに従って前記複数の第1プログラムセクションを順序付けし、

プログラム格納領域をキャッシュメモリサイズに分割した複数のセクショングループ格納領域の一のセクショングループ格納領域に一の第1プログラムセクションを配置できる場合には前記一のセクショングループ格納領域に前記一の第1プログラムセクションを配置し、前記一の第1プログラムセクションを配置できない場合には前記一のセクショングループ格納領域に空き領域を残して次のセクショングループ格納領域に前記一の第1プログラムセクションを配置して、前記複数の第1プログラムセクションを前記順序どおりに前記複数のセクショングループ格納領域に配置し、

前記空き領域に前記第2プログラムセクションを配置できる場合には前記第2プログラムセクションを前記空き領域に配置し、前記第2プログラムセクションを配置できない場合には次の配置可能な空き領域に前記第2プログラムセクションを配置し、

前記第1のプログラムセクションが配置された前記セクショングループ格納領域と前記空き領域に前記第2プログラムセクションを配置する前の前記第2プログラムセクションとがメモリ上に占める第1メモリサイズから前記空き領域に配置された前記第2プログラムセクションがメモリ上に占める第2メモリサイズを減算した第1サイズと、前記複数の第1プログラムセクションのサイズと前記複数の第2プログラムセクションのサイズを加えた第2サイズとの割合を示す膨張率が所定膨張率以下である場合には前記第2プログラムセクションの配置を終了する、

ことを特徴とするプログラムセクションレイアウト方法。

【請求項3】

最後尾に配置されている前記第2プログラムセクションから順に配置される、

ことを特徴とする請求項1または2記載のプログラムセクションレイアウト方法。

【請求項4】

プログラム実行時に一部のプログラムセクションをキャッシュメモリに格納して動作するコンピュータのプログラムセクションレイアウト方法において、

前記コンピュータが、

任意の処理機能の実行単位であるプログラムセクションのうち所定のアルゴリズムに基づく条件を満たす前記プログラムセクションを順序付けし、前記所定のアルゴリズムによって順序付けがされた前記プログラムセクションを、プログラム格納領域をキャッシュメモリサイズ単位に分割したセクショングループ格納領域の空き領域に前記順序に従って配置し、前記プログラムセクションをセクショングループにグループ分けするグループ分け手順と、

前記セクショングループを構成する前記プログラムセクションの配置関係を保持したまま、前記所定のアルゴリズムによって順序付けがされなかったプログラムセクションを前記セクショングループ格納領域の空き領域に配置し、前記セクショングループを配列順序に従って奇数番号グループと偶数番号グループとに分け、前記奇数番号グループもしくは前記偶数番号グループのいずれかの前記セクショングループを構成する前記プログラムセクションの配列順を逆順にして配置し、キャッシュコンフリクトを緩和するプログラムセ

クシヨンのレイアウトを生成することでレイアウトを最適化するレイアウト最適化手順と

、  
 を実行することを特徴とするプログラムセクションレイアウト方法。

【請求項5】

前記所定のアルゴリズムは、実行される処理全体に対して前記プログラムセクションが実行される割合を示す占有率に基づいて前記順序を決定することを含み、

前記所定のアルゴリズムに基づく条件は、前記プログラムセクションの占有率が所定占有率よりも大きいことを含むこと、

を特徴とする請求項1、2、3または4に記載のプログラムセクションレイアウト方法

。

10

【請求項6】

前記所定のアルゴリズムは、所定時間内に前記プログラムセクションが他のプログラムセクションから参照される回数を示す参照頻度に基づいて前記順序を決定することを含み、

前記所定のアルゴリズムに基づく条件は、前記プログラムセクションの参照頻度が所定参照頻度よりも大きいことを含むこと、

を特徴とする請求項1、2、3または4に記載のプログラムセクションレイアウト方法

。

【請求項7】

プログラム実行時に一部のプログラムセクションをキャッシュメモリに格納して動作するコンピュータのプログラムセクションのレイアウト処理を実行させるレイアウト処理プログラムにおいて、

20

前記コンピュータに、

任意の処理機能の実行単位である複数のプログラムセクションを所定のアルゴリズムに基づく条件を満たす複数の第1プログラムセクションと前記所定のアルゴリズムに基づく条件を満たさない第2プログラムセクションとに分けるとともに前記所定のアルゴリズムに従って前記複数の第1プログラムセクションを順序付けし、

プログラム格納領域をキャッシュメモリサイズに分割した複数のセクショングループ格納領域の一のセクショングループ格納領域に一の第1プログラムセクションを配置できる場合には前記一のセクショングループ格納領域に前記一の第1プログラムセクションを配置し、前記一の第1プログラムセクションを配置できない場合には前記一のセクショングループ格納領域に空き領域を残して次のセクショングループ格納領域に前記一の第1プログラムセクションを配置して、前記複数の第1プログラムセクションを前記順序どおりに前記複数のセクショングループ格納領域に配置し、

30

前記空き領域に前記第2プログラムセクションを配置できる場合には前記第2プログラムセクションを前記空き領域に配置し、前記第2プログラムセクションを配置できない場合には次の配置可能な空き領域に前記第2プログラムセクションを配置し、

前記第1のプログラムセクションが配置された前記セクショングループ格納領域と前記空き領域に前記第2プログラムセクションを配置する前の前記第2プログラムセクションとがメモリ上に占める第1メモリサイズから前記空き領域に配置された前記第2プログラムセクションがメモリ上に占める第2メモリサイズを減算したメモリサイズが所定メモリサイズ以下である場合には前記第2プログラムセクションの配置を終了する、

40

手順を実行させることを特徴とするレイアウト処理プログラム。

【請求項8】

プログラム実行時に一部のプログラムセクションをキャッシュメモリに格納して動作するコンピュータのプログラムセクションのレイアウト処理を実行させるレイアウト処理プログラムにおいて、

前記コンピュータに、

任意の処理機能の実行単位であるプログラムセクションのうち所定のアルゴリズムに基づく条件を満たす前記プログラムセクションを順序付けし、前記所定のアルゴリズムによ

50

って順序付けがされた前記プログラムセクションを、プログラム格納領域をキャッシュメモリサイズ単位に分割したセクショングループ格納領域の空き領域に前記順序に従って配置し、前記プログラムセクションをセクショングループにグループ分けするグループ分け手順と、

前記セクショングループを構成する前記プログラムセクションの配置関係を保持したまま、前記所定のアルゴリズムによって順序付けがされなかったプログラムセクションを前記セクショングループ格納領域の空き領域に配置し、前記セクショングループを配列順序に従って奇数番号グループと偶数番号グループとに分け、前記奇数番号グループもしくは前記偶数番号グループのいずれかの前記セクショングループを構成する前記プログラムセクションの配列順を逆順にして配置し、キャッシュコンフリクトを緩和するプログラムセクションのレイアウトを生成することでレイアウトを最適化するレイアウト最適化手順と

10

を実行させることを特徴とするレイアウト処理プログラム。

【請求項 9】

プログラム実行時に一部のプログラムセクションをキャッシュメモリに格納して動作するコンピュータのプログラムセクションのレイアウト処理を実行させるレイアウト処理プログラムを記録したコンピュータ読み取り可能な記録媒体において、

任意の処理機能の実行単位である複数のプログラムセクションを所定のアルゴリズムに基づく条件を満たす複数の第 1 プログラムセクションと前記所定のアルゴリズムに基づく条件を満たさない第 2 プログラムセクションとに分けるとともに前記所定のアルゴリズムに従って前記複数の第 1 プログラムセクションを順序付けし、

20

プログラム格納領域をキャッシュメモリサイズに分割した複数のセクショングループ格納領域の一のセクショングループ格納領域に一の第 1 プログラムセクションを配置できる場合には前記一のセクショングループ格納領域に前記一の第 1 プログラムセクションを配置し、前記一の第 1 プログラムセクションを配置できない場合には前記一のセクショングループ格納領域に空き領域を残して次のセクショングループ格納領域に前記一の第 1 プログラムセクションを配置して、前記複数の第 1 プログラムセクションを前記順序どおりに前記複数のセクショングループ格納領域に配置し、

前記空き領域に前記第 2 プログラムセクションを配置できる場合には前記第 2 プログラムセクションを前記空き領域に配置し、前記第 2 プログラムセクションを配置できない場合には次の配置可能な空き領域に前記第 2 プログラムセクションを配置し、

30

前記第 1 のプログラムセクションが配置された前記セクショングループ格納領域と前記空き領域に前記第 2 プログラムセクションを配置する前の前記第 2 プログラムセクションとがメモリ上に占める第 1 メモリサイズから前記空き領域に配置された前記第 2 プログラムセクションがメモリ上に占める第 2 メモリサイズを減算したメモリサイズが所定メモリサイズ以下である場合には前記第 2 プログラムセクションの配置を終了する、

手順をコンピュータに実行させるレイアウト処理プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

40

【技術分野】

【0001】

本発明はプログラムセクションレイアウト方法およびレイアウト処理プログラムに関し、特にプログラム実行時に一部のプログラムセクションをキャッシュメモリに格納して動作するコンピュータのプログラムセクションレイアウト方法およびレイアウト処理プログラムに関する。

【背景技術】

【0002】

従来からキャッシュメモリが用意されたプログラム実行環境において、キャッシュメモリの使用効率を上げるプログラムセクションの配置方法が求められている。

50

一般的にキャッシュを実装するにはコストがかかるため、プログラム全体が格納されるメモリと比較してキャッシュサイズは小さい。そこで、プログラム格納メモリをキャッシュサイズ単位で分割し、キャッシュメモリとプログラム格納メモリの分割領域とを対応付け、分割領域ごとにプログラムセクションを任意のアルゴリズムに従って配置しておく。あるアドレスに対する最初アクセスで、そのアドレスに対応するプログラムセクションをキャッシュメモリの割当て領域にコピーし、次に同じアドレスでアクセスがあった場合、キャッシュメモリを直接アクセスする。このようにすることで、高速なプログラム処理を実現している。アクセスの際に該当するプログラムセクションがキャッシュメモリに存在しないキャッシュミスが発生したときには、再びプログラム格納メモリからコピーを行う。アルゴリズムは、キャッシュミスを削減するように設定される。

10

**【0003】**

また、プログラム格納メモリからキャッシュメモリへのコピーは、所定のサイズ単位で行われ、このサイズでキャッシュを分割した領域をキャッシュラインと呼ぶ。したがって、同一のキャッシュラインに割当てられたプログラムセクションは、いずれか一つだけがキャッシュ上にコピーされている。このため、プログラムが切替えられるごとにキャッシュにプログラムをコピーし直す必要が生じる。これをキャッシュ・コンフリクトという。このようなキャッシュ・コンフリクトやキャッシュミスが頻繁に発生すると、結果としてプログラムの実行速度が遅くなってしまいう問題がある。

**【0004】**

そこで、命令キャッシュに関し、仮想命令処理単位のメモ上の配置を動的に入れ替え、インタプリタ実行時における命令キャッシュのヒット率の向上を図る方法がある（たとえば、特許文献1参照）。また、直接的、間接的にキャッシュ・コンフリクトを発生する可能性のある関数呼び出しパターンを抽出し、このパターンを避けながら実行頻度の高い関数順にプログラムを配置するレイアウト方法がある（たとえば、特許文献2参照）。

20

【特許文献1】特開2000-267861号公報（段落番号〔0023〕～〔0030〕、図1）

【特許文献2】特開2001-216140号公報（段落番号〔0033〕～〔0038〕、図1）

**【発明の開示】****【発明が解決しようとする課題】**

30

**【0005】**

従来のプログラムセクションレイアウト方法では、キャッシュメモリの使用効率を上げるための様々なアルゴリズムがとられているが、キャッシュミスを削減するためにプログラム構造の解析や関数の呼び出し関係の解析を行うなど、アルゴリズムのプログラム依存性が高くなっているという問題点がある。この結果、アルゴリズムの種類も多くなり、レイアウト処理の自動化を難しくしている。

**【0006】**

また、上記アルゴリズムに従ってプログラムセクションの順序付けが行われると、順番に従ってキャッシュサイズ単位の分割領域に配置する。同じ分割領域内に配置されるプログラムセクションの一群をセクショングループと呼ぶ。そして、1つのセクショングループに配置されたプログラムセクションサイズの合計が分割領域のサイズ、すなわち、キャッシュサイズ単位を超えた場合、新規セクショングループを作成してプログラムセクションを配置していく。こうして、プログラムセクションはいずれかのセクショングループに属してプログラム格納メモリ上に配置される。

40

**【0007】**

しかしながら、プログラムセクションはそれぞれにサイズが異なり、順序付けられたプログラムセクションを順に分割領域に割当てていくと、使用されない無駄な領域ができてしまうという問題点がある。この結果、全体のプログラムセクションを格納するために必要なメモリサイズが大きくなってしまいう問題点がある。特に、プログラムをROM（Read Only Memory）に格納する場合など、プログラムセクション領域が有限な

50

プログラム実行環境においては、キャッシュメモリの使用効率の向上とともに、プログラムセクションサイズを抑えることが求められている。

【0008】

本発明はこのような点に鑑みてなされたものであり、プログラムの性質によらずキャッシュの使用効率を向上させるプログラムセクションレイアウト方法およびレイアウト処理プログラムを提供することを目的とする。また、本発明の他の目的は、レイアウト後のプログラムセクションサイズを抑えることにある。

【課題を解決するための手段】

【0009】

本発明では上記課題を解決するために、コンピュータに図1に示すような処理手順を実行させるプログラムセクションレイアウト方法が提供される。

このようなプログラムセクションレイアウト方法によれば、キャッシュメモリに対応させ、キャッシュメモリサイズ単位に分割されたメモリ上の領域（セクショングループ格納領域）に格納するプログラムセクションをグループ分けし、セクショングループを構成するプログラムセクションの配置順を最適化する。プログラムセクションは、任意の処理機能の実行単位であり、たとえば、C言語の場合の関数などに相当する。このとき、グループ分け手段1では、プログラムセクションを所定のアルゴリズムに基づく順序に従って、セクショングループにグループ分けする。同一のセクショングループにグループ分けされたプログラムセクションの一群は、同一のセクショングループ格納領域に配置される。グループ分け手段1では、グループ分けされたプログラムセクションの合計サイズがキャッシュメモリサイズを超えないようにグループ分けを行う。続いて、レイアウト最適化手段2は、グループ分けによって決められたセクショングループを構成するプログラムセクションの配置関係を保持したまま、レイアウトの最適化を行う。このとき、配置関係、すなわち、プログラムセクションの配置順序は、そのままもしくは逆順にして保持される。レイアウト最適化では、配置関係を保持したまま、セクショングループとグループ分けされなかったプログラムセクション、もしくはセクショングループ同士を組み合わせ、再レイアウトを行う。

【発明の効果】

【0010】

本発明のプログラムセクションレイアウト方法では、所定のアルゴリズムで順序付けされるプログラムセクションをグループ分けして配置した後、セクショングループ内のプログラムセクションの配置関係を保持したまま、セクショングループとセクショングループに属さないプログラムセクション、もしくはセクショングループ同士を組み合わせ、再レイアウトする。これによりアルゴリズム結果を保持しながら、セクショングループ同士やセクショングループと単独のプログラムセクションを組み合わせることによってセクショングループ格納領域のレイアウトを最適化することができる。すなわち、アルゴリズム結果をそのまま利用して最適化を行うので、プログラムの性質などによらず最適化される。この結果、プログラムの性質によらずキャッシュの使用効率を向上させることが可能となる。また、レイアウト後のメモリ領域のサイズを抑えることができる。

【発明を実施するための最良の形態】

【0011】

以下、本発明の実施の形態を図面を参照して説明する。まず、実施の形態に適用される発明の概念について説明し、その後、実施の形態の具体的な内容を説明する。

本発明は、キャッシュメモリを搭載したコンピュータ用のアプリケーションプログラムについて、キャッシュミス削減し、メモリ使用効率を向上させるプログラムセクションレイアウト方法を実現する。図1は、実施の形態に適用される発明の概念図である。

【0012】

本発明に係るプログラムセクションレイアウト方法は、このプログラムセクションレイアウト方法の処理手順をコンピュータに実行させるレイアウト処理プログラムによって実

10

20

30

40

50

現される。レイアウト処理プログラムをコンピュータに実行させることにより、コンピュータはプログラムセクションレイアウト方法による処理手順を実行し、レイアウト処理装置として機能する。

【0013】

レイアウト処理装置は、プログラムセクションをグループ分けするグループ分け手段1とレイアウト最適化手段2を具備する。なお、プログラムセクションは、任意の処理機能の実行単位、たとえば、C言語の場合の関数などに相当する。グループ分け手段1には、ソースプログラムをコンパイル/アセンブルしたオブジェクトの形式で入力する。また、プログラムセクションには、コード(命令)セクションとデータセクションとがあるが、本発明はコードセクションに関するものである。

10

【0014】

グループ分け手段1は、プログラムセクションを所定のアルゴリズムによって、グループ分けする。このとき生成されるセクショングループは、キャッシュメモリサイズ単位で分割されたプログラム格納領域のセクショングループ格納領域の1つに配置されるプログラムセクション群を指す。図の例では、グループ1メモリ領域101、グループ2メモリ領域102、グループ3メモリ領域103が、それぞれのセクショングループ格納領域に相当する。グループ分けにおいては、所定のアルゴリズムに基づく順序に従って、順番が最上位(もしくは最下位)のプログラムセクションから順にグループ1へグループ分けしていく。グループ1へグループ分けされたプログラムセクションの合計サイズがセクショングループ格納領域のメモリサイズ(キャッシュサイズに相当する)を超えない範囲で順次プログラムセクションをグループ分けしていく。そのプログラムセクションでサイズが超えてしまう場合、新たなセクショングループが設定され、同様の処理を繰り返す。このようにして、所定のアルゴリズムに基づく順序に従って、グループ1から順にプログラムセクションがグループ分けされる。

20

【0015】

レイアウト最適化手段2は、このようにしてグループ分けされ、順序に従ってプログラムセクションが配置されたセクショングループについて、セクショングループ内の配置関係を保持しながら、セクショングループとグループ分けされなかったプログラムセクション、もしくはプログラムセクション同士を組み合わせ、レイアウトの最適化処理を行い、最適化されたリンク情報を出力する。セクショングループ内の配置関係を保持するとは、所定のアルゴリズムに従って順序付けて配置されたプログラムセクションの配列を保持することを意味し、配列をそのままの順序で保持する場合と、逆順に並べ替える場合とがある。たとえば、参照頻度によって参照頻度の高いものから順番に配置されたプログラムセクションを逆順に並び替えると、参照頻度の低いものから順番に配置されることになり、アルゴリズムの主題は保持されている。最適化処理について説明する。たとえば、セクショングループとグループ分けされていないプログラムセクションの場合、グループ分け終了後、グループ1メモリ領域101内には、プログラムセクションが格納された領域101aと、次の順番のプログラムセクションが格納できなかったために生じた隙間領域101bがある。最適化処理では、この隙間領域101bに、割当てなし領域104に格納されているグループ割当てされなかったプログラムセクションが配置可能であるかどうかを判定し、可能であれば、隙間領域101bにこれを配置する。ここでは、割当てなし領域104の最後尾のプログラムセクション104aがグループ1メモリ領域101の隙間領域101bに配置されている。また、最適化処理として、各セクショングループのプログラムセクションの配列を逆順にすることもできる。

30

40

【0016】

以上の処理手順が実行されることにより、アルゴリズムに従った配置結果を保持しながら、セクショングループ格納領域のレイアウトを最適化することができる。

以下、実施の形態を図面を参照して詳細に説明する。

【0017】

図2は、プログラムセクションレイアウト方法が適用されるシステム構成の一例を示し

50

た概略図である。

コンピュータのプログラム開発を支援する支援装置では、コーディングされたソースプログラムが作成されると、コンパイラ/アセンブラ部 10 でコンパイル処理とアSEMBル処理が実行され、オブジェクトファイルが生成される。オブジェクトファイルは、処理機能単位の複数のオブジェクトセクションから構成される。作成されたオブジェクトセクションは、リンカ部 20 によってリンクされ、コンピュータが実行可能なロードオブジェクトが作成される。ここで、リンカ部 20 は、オブジェクトセクションをある基準で並べ替えることができる。リンカ部 20 によって作成されたロードオブジェクトは、プロファイル部 30 でシミュレーション実行され、性能評価が行われる。性能評価として、たとえば、セクションごとのキャッシュ情報、呼び出し回数、占有率などが測定される。プロファイル部 30 で得られた性能評価に基づいて、リンカ部 20 でオブジェクトセクションが再配置される。なお、支援装置は、コーディング、コンパイラ/アセンブラ、リンカ、プロファイルの各処理を同一の装置で行うこともできるし、各処理を分担する複数の装置からなるシステム構成としてもよい。実施の形態のプログラムセクションレイアウト方法は、リンカ部 20 に適用される。

10

#### 【0018】

リンカ部 20 の処理機能を実現するレイアウト処理装置について説明する。図 3 は、本実施の形態のレイアウト処理装置のハードウェア構成例を示すブロック図である。

レイアウト処理装置 100 は、CPU (Central Processing Unit) 111 によって装置全体が制御されている。CPU 111 には、バス 117 を介して RAM (Random Access Memory) 112、ハードディスクドライブ (HDD: Hard Disk Drive) 113、グラフィック処理装置 114、入力インタフェース 115、通信インタフェース 116 が接続されている。

20

#### 【0019】

RAM 112 には、CPU 111 に実行させる OS (Operating System) のプログラムやアプリケーションプログラムの少なくとも一部が一時的に格納される。また、RAM 112 には、CPU 111 による処理に必要な各種データが格納される。HDD 113 には、OS やアプリケーションのプログラムが格納される。グラフィック処理装置 114 には、モニタ 118 が接続されており、CPU 111 からの命令に従って画像をモニタ 118 の画面に表示させる。入力インタフェース 115 には、キーボード 119a やマウス 119b が接続されており、キーボード 119a やマウス 119b から送られてくる信号を、バス 117 を介して CPU 111 に送信する。通信インタフェース 116 は、ネットワーク 120 に接続されており、ネットワーク 120 を介して外部装置との間でデータの送受信を行う。

30

#### 【0020】

このようなハードウェア構成によって、本実施の形態の処理機能を実現することができる。なお、図 2 に示した処理機能を実現する支援装置も同様のハードウェア構成で実現することができる。

#### 【0021】

以下、レイアウト処理手順について詳細に説明する。まず、第 1 の実施の形態として、セクショングループとグループ分けされなかったプログラムセクションを組み合わせる場合について説明し、次に、第 2 の実施の形態として、セクショングループ同士を組み合わせる場合について説明する。以下、プログラムセクションはセクション、セクショングループは単にグループと表記する。

40

#### 【0022】

##### [第 1 の実施の形態]

第 1 の実施の形態では、プロファイル部 30 によって作成された性能評価に基づくアルゴリズムによって、プログラムセクションのグループ分けを行う。

#### 【0023】

図 4 は、第 1 の実施の形態のグループ分け処理手順を示したフローチャートである。プ

50



ロファイル部 30 による性能評価取得後、処理が起動される。

【ステップ S 1 1】 新規グループを作成する。グループは、キャッシュサイズ単位に分割されるグループ格納領域に対応して設けられる。

【0024】

【ステップ S 1 2】 性能評価結果に基づくアルゴリズムに従って、次の順番で配置されるセクションを検索する。

【ステップ S 1 3】 該当するセクションがあるかどうかを判定され、セクションがない場合は処理をステップ S 1 6 へ進める。

【0025】

【ステップ S 1 4】 該当するセクションがあった場合、このグループの残りの領域のサイズと検索されたセクションのサイズを比較し、セクションをグループに設定した場合にグループ格納領域サイズ（キャッシュサイズ）内に収まるかどうかを判定する。サイズに収まらない場合は、ステップ S 1 1 に戻って、新規グループの作成からの処理を行う。

【0026】

【ステップ S 1 5】 次の順番に該当するセクションがグループ格納領域サイズに収まる場合、そのセクションをグループに追加する。そして、処理をステップ S 1 2 へ進め、次のセクション検索からの処理を行う。

【0027】

【ステップ S 1 6】 グループに配置されるセクションがない場合、グループごとに生じたグループ格納領域の隙間を埋める処理を行う。隙間埋め処理の詳細は後述する。

【ステップ S 1 7】 オプション作成の要求があれば、オプション作成処理を実行する。

【0028】

隙間埋め処理について説明する。図 5 は、第 1 の実施の形態の隙間埋め処理の手順を示したフローチャートである。

セクションがグループ分けされた後、処理が開始される。グループ分けにより、セクションが配置された複数のグループと、グループ分けされないセクションとがアルゴリズムに基づく順序に従って並べられている。グループ分けされないセクションとは、アルゴリズムによってグループ分けする必要がないと判定されるセクションで、たとえば、呼び出し回数の多い順に並べた場合、呼び出し回数が 0 のものをグループ分けしないと設定したような場合に生じる。

【0029】

また、処理が開始される前に、あらかじめ、膨張率の制限値や最大サイズなどが設定される（ステップ S 6 1）。ここで、膨張率は、全セクションのサイズと、グループ分けされてグループ格納領域に格納された後の全セクションがメモリ上に占めるサイズの割合に応じた値である。グループ分けする際、グループ格納領域に隙間が生じるなどによって、グループ分けされた後の全セクションがメモリ上に占めるサイズは、もとの全セクションのサイズよりも大きくなっている。膨張率は、最適化によってグループ化された全セクションのサイズをどの程度許容するかを示す値として設定される。また、直接制限される最大サイズを設定することもできる。

【0030】

【ステップ S 6 2】 先頭グループから順に、グループ格納領域に隙間があるかどうかを探す。

【ステップ S 6 3】 隙間のあるグループ格納領域があるかどうかを判定する。ない場合、処理を終了する。

【0031】

【ステップ S 6 4】 隙間のサイズに見合ったセクションを末端（最後尾）のセクションから順に検索する。

【ステップ S 6 5】 隙間のサイズに対応するセクションがあるかどうかを判定する。対応するセクションがない場合は、保留とし、隙間をあけたままステップ S 6 2 に戻って

10

20

30

40

50

、次のグループの隙間の検索からの処理を行う。

【0032】

[ステップS66] 対応するセクションがあった場合、このセクションをグループ格納領域の隙間に埋める。これにより、このセクションのサイズ分のメモリ領域が、全セクションが占めるメモリサイズから減算される。

【0033】

[ステップS67] ステップS66による隙間埋め込み処理によって、減算された全セクションが占めるメモリサイズと最大サイズが比較される。また、比較は、減算された全セクションが占めるメモリサイズの膨張率と設定された膨張率で行ってもよい。そして、今回の隙間埋め処理によって、最大サイズもしくは膨張率が設定を満足したかどうかを判定する。設定が満足された場合は、処理を終了する。設定が満足されなかった場合は、ステップS62に戻って、次のグループからの処理を行う。

【0034】

以上の処理手順は、最終的に指定の膨張率または最大サイズが満たされるまで、すべてのグループの隙間について処理が行われる。このように、グループ格納領域の隙間にグループ分けされないセクションを埋め込むことによって、アルゴリズムによって配列された順序を保持しながら、全セクションを格納するメモリサイズを最適化することができる。特に、ROMなどのようにプログラムセクション領域が有限なプログラム実行環境において有効である。

【0035】

具体例を用いて第1の実施の形態の動作を説明する。図6は、プロファイル部による性能評価結果の一例を示した図である。

図の例では、セクション名「a」、「b」、「c」、「d」、「e」、「f」の6つのセクションについて、それぞれの占有率、参照頻度を算出している。占有率は、実行された処理全体のうち、このセクションが実行されていた割合を示している。たとえば、全体の実行時間のうち、このセクションが実行されていた時間などで計測する。参照頻度は、所定の時間内に他のセクションから参照される回数を計測した値である。図の例では、セクション「a」は、占有率0.1%で参照頻度は1回である。また、セクション「b」は占有率2%で参照回数は15回、セクション「c」は占有率80%で参照回数は1000回、セクション「d」は占有率10%で参照回数は300回、セクション「e」は占有率8%で参照回数は10回およびセクション「f」は占有率0%で参照回数は0回である。

【0036】

このような性能評価に基づくグループ分けのアルゴリズムは、任意に設定される。たとえば、占有率の多い順にセクションを順序付けるアルゴリズムであれば、セクションは、「c」、「d」、「e」、「b」、「a」、「f」の順に順序付けられる。また、参照頻度の大きい順にセクションを順序付けるアルゴリズムであれば、「c」、「d」、「b」、「e」、「a」、「f」の順に順序付けられる。ここでは、占有率の多い順に順序付けるとともに、占有率が1%未満のセクションについてはグループ分けしないというアルゴリズムを設定するとする。

【0037】

このようなアルゴリズムに従って、セクションが順序付けされ、各グループにグループ分けされる。グループ分けの後には、セクションをグループの隙間に埋め込む処理が行われる。図7は、第1の実施の形態におけるグループ分け後とレイアウト最適化後のグループ構成の一例を示した図である。

【0038】

グループ分け後の構成は、グループ1(210)、グループ2(220)の2つのグループと、グループ分けされていないセクション「a」230とセクション「f」240が順に格納されている。グループ1(210)には、最も占有率の高いセクション「c」211と、次に占有率の高いセクション「d」212が順に配置され、隙間領域213がで

10

20

30

40

50

きている。グループ2(220)には、3番目に占有率の高いセクション「e」221、4番目に占有率の高いセクション「b」222が順に配置され、隙間領域223が生じている。グループ分けが終了した時点では、セクションが格納される全メモリのサイズは、グループ格納領域サイズ(キャッシュサイズ)×2にセクション「a」230とセクション「f」240のサイズを加えたサイズである。

【0039】

レイアウト最適化処理では、グループ1(210)の隙間領域213と末端のセクション「f」240のサイズを比較し、セクション「f」240が隙間領域213に収まればここへ埋め込む。ここでは、サイズ内に収まるとし、セクション「f」240がグループ1(210)のセクション「d」212の次に配置される。この処理で、セクションが格納される全メモリのサイズは、セクション「f」240のサイズ分が減算される。また、末端のセクションは、セクション「a」230になる。

10

【0040】

続いて、隙間が検索され、グループ2(220)の隙間領域223が検索される。そして、グループ2(220)の隙間領域223と現時の末端のセクション「a」230のサイズを比較し、セクション「a」230が隙間領域223に収まればここへ埋め込む。ここでは、サイズ内に収まるとし、セクション「a」230がグループ2(220)のセクション「b」222の次に配置される。この処理で、セクションが格納される全メモリのサイズは、さらに、セクション「a」230のサイズ分が減算され、図の例では、グループ格納領域サイズ(キャッシュサイズ)×2になる。

20

【0041】

このように、第1の実施の形態では、アルゴリズムによってグループ化され、グループ内で順序付けされた順番を保ちながら、占有率の低いセクションをグループ格納領域の隙間に埋めることで、キャッシュミスの発生確率を抑えたまま、メモリ領域を節約することができる。また、この手法は、所定のアルゴリズムによって設定された順序付けに基づいて最適化を行うため、どのようなアルゴリズムによってグループ分けと順序付けがされたセクションであっても、また、プログラムの性質に依存することなく最適化を行うことができる。

【0042】

[第2の実施の形態]

第2の実施の形態は、第1の実施の形態のグループ分け処理の「隙間埋め処理」をグループの配列を交互に逆順にする「反転アルゴリズム処理」にしたものである。グループ分け処理自体は、第1の実施の形態の場合と同じであるので、説明は省略する。

30

【0043】

図8は、第2の実施の形態の反転アルゴリズム処理の手順を示したフローチャートである。グループ分けが終了した時点で、各グループは、アルゴリズムに基づく順序付けに従ってセクションが配置されている。たとえば、参照頻度の高い順、あるいは、占有率の高い順に並べられている。

【0044】

[ステップS81] グループを先頭から検索する。すなわち、グループ1から処理を開始させる。

40

[ステップS82] グループ番号が偶数であるかどうかを判定する。偶数でない場合は、処理をステップS84へ進める。

【0045】

[ステップS83] グループ番号が偶数であった場合、グループ内のセクションの先頭/終了アドレスをシンメトリックに反転する。すなわち、アルゴリズムに従って最も上位(先頭アドレス)に配置されたセクションを最も下位(終了アドレス)に移す。順番に従ってセクションを移動させていき、最終的には、アルゴリズムの順序付けとは逆順に並び替える。たとえば、アルゴリズムが参照頻度の高い順であれば、参照頻度の低い順に上位から配置される。すなわち、アルゴリズムが反転されている。しかしながら、アルゴリ

50

ズム本来の主題である参照頻度に従った順序を付けるということは満たされている。

【 0 0 4 6 】

[ ステップ S 8 4 ] 次のグループ番号を検索する。

[ ステップ S 8 5 ] 次のグループがあるかどうか判定され、なければ処理を終了する。次のグループがあれば、ステップ S 8 2 に戻って処理を繰り返す。

【 0 0 4 7 】

以上の処理手順が実行されることにより、偶数番号のグループ番号を持つグループのセクションが逆順に並び替えられる。たとえば、参照頻度が高い順に順序付けられているとすると、どのグループも上部に存在するセクションがアクセスされる可能性が高くなる。このため、上部では、キャッシュ・コンフリクトが発生しやすくなる。そこで、奇数番のグループを占有率が高い順に並べ、偶数番のグループをその反対の参照頻度が低い順に並べれば、キャッシュ・コンフリクトを緩和させることができる。

10

【 0 0 4 8 】

なお、上記の説明では、偶数番号のグループのセクション配置を反転させるとしたが、奇数番号のグループのセクションの配置を反転させるとしても同様の効果が得られる。

具体例を用いて説明する。図 9 は、第 2 の実施の形態におけるグループ分け後とレイアウト最適化後のグループ構成の一例を示した図である。

【 0 0 4 9 】

グループ分け後の状態では、グループ 1 ( 3 1 0 )、グループ 2 ( 3 2 0 )、グループ 3 ( 3 3 0 )、グループ 4 ( 3 4 0 ) の 4 つのグループが設けられている。それぞれのグループは、たとえば、参照頻度の高い順に 1、2、3、4、5 と配置されている。もちろん、占有率の高い順であってもよい。図の例では、グループ 1 ( 3 1 0 ) は、参照頻度の高い順に a 1 ( 3 1 1 )、a 2 ( 3 1 2 )、a 3 ( 3 1 3 )、a 4 ( 3 1 4 ) が並べられている。同様に、参照頻度の高い順に、グループ 2 ( 3 2 0 ) は、b 1 ( 3 2 1 )、b 2 ( 3 2 2 )、b 3 ( 3 2 3 )、b 4 ( 3 2 4 )、b 5 ( 3 2 5 ) と配置され、グループ 3 ( 3 3 0 ) は、c 1 ( 3 3 1 )、c 2 ( 3 3 2 )、c 3 ( 3 3 3 ) と配置され、グループ 4 ( 3 4 0 ) は、d 1 ( 3 4 1 )、d 2 ( 3 4 2 )、d 3 ( 3 4 3 )、d 4 ( 3 4 4 ) と配置される。すなわち、それぞれ「1」が占める上部は参照頻度が高く、下部は参照頻度が低い。このため、この状態では、上部ほど参照される回数が多く、キャッシュ・コンフリクトが発生しやすくなる。

20

30

【 0 0 5 0 】

そこで、レイアウト最適化処理では、偶数番のグループ 2 ( 3 2 0 ) とグループ 4 ( 3 4 0 ) のセクションを逆順に再配置する。図の例では、グループ 2 ( 3 2 0 ) は、b 1 ( 3 2 1 )、b 2 ( 3 2 2 )、b 3 ( 3 2 3 )、b 4 ( 3 2 4 )、b 5 ( 3 2 5 ) の順番が、b 5 ( 3 2 5 )、b 4 ( 3 2 4 )、b 3 ( 3 2 3 )、b 2 ( 3 2 2 )、b 1 ( 3 2 1 ) の順に再配置される。同様に、グループ 4 ( 3 4 0 ) は、d 1 ( 3 4 1 )、d 2 ( 3 4 2 )、d 3 ( 3 4 3 )、d 4 ( 3 4 4 ) の順番が、d 4 ( 3 4 4 )、d 3 ( 3 4 3 )、d 2 ( 3 4 2 )、d 1 ( 3 4 1 ) の順に再配置される。

【 0 0 5 1 】

図 10 は、第 2 の実施の形態のレイアウト最適化処理後のグループの参照回数頻度の一例を示した図である。図 9 と同じものには同じ番号を付す。図の参照頻度は、値が右へ行くほど参照頻度が高くなる。

40

【 0 0 5 2 】

図から明らかなように、グループ 1 ( 3 1 0 )、グループ 3 ( 3 3 0 ) は、参照頻度の高い順に配置されており、下部になるほど参照頻度が低くなる。一方、グループ 2 ( 3 2 )、グループ 4 ( 3 4 0 ) は、逆順、すなわち、参照頻度の低い順に配置されており、下部になるほど参照頻度が高くなる。このように、参照頻度のパターンを反転させることで、キャッシュ・コンフリクトを緩和することが可能となる。

【 0 0 5 3 】

ところで、第 1 の実施の形態と第 2 の実施の形態を別々に説明したが、どちらもアルゴ

50

リズムに基づく順序付けは保持されているので、両方のレイアウト最適化処理を実行することもできる。たとえば、第1の実施の形態によってメモリサイズを抑えた後、さらに、偶数もしくは奇数のグループの配置順を逆順にする。

#### 【0054】

なお、上記の処理機能は、コンピュータによって実現することができる。その場合、レイアウト処理装置が有すべき機能の処理内容を記述したプログラムが提供される。そのプログラムをコンピュータで実行することにより、上記処理機能がコンピュータ上で実現される。処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリなどがある。磁気記録装置には、ハードディスク装置(HDD)、フレキシブルディスク(FD)、磁気テープなどがある。光ディスクには、DVD(Digital Versatile Disc)、DVD-RAM(Random Access Memory)、CD-ROM(Compact Disc Read Only Memory)、CD-R(Recordable)/RW(ReWritable)などがある。光磁気記録媒体には、MO(Magneto-Optical disk)などがある。

10

#### 【0055】

プログラムを流通させる場合には、たとえば、そのプログラムが記録されたDVD、CD-ROMなどの可搬型記録媒体が販売される。また、プログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することもできる。

#### 【0056】

プログラムを実行するコンピュータは、たとえば、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、自己の記憶装置に格納する。そして、コンピュータは、自己の記憶装置からプログラムを読み取り、プログラムに従った処理を実行する。なお、コンピュータは、可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することもできる。また、コンピュータは、サーバコンピュータからプログラムが転送される毎に、逐次、受け取ったプログラムに従った処理を実行することもできる。

20

#### 【図面の簡単な説明】

#### 【0057】

【図1】実施の形態に適用される発明の概念図である。

30

【図2】プログラムセクションレイアウト方法が適用されるシステム構成の一例を示した概略図である。

【図3】本実施の形態のレイアウト処理装置のハードウェア構成例を示すブロック図である。

【図4】第1の実施の形態のグループ分け処理手順を示したフローチャートである。

【図5】第1の実施の形態の隙間埋め処理の手順を示したフローチャートである。

【図6】プロファイル部による性能評価結果の一例を示した図である。

【図7】第1の実施の形態におけるグループ分け後とレイアウト最適化後のグループ構成の一例を示した図である。

【図8】第2の実施の形態の反転アルゴリズム処理の手順を示したフローチャートである

40

【図9】第2の実施の形態におけるグループ分け後とレイアウト最適化後のグループ構成の一例を示した図である。

【図10】第2の実施の形態のレイアウト最適化処理後のグループの参照回数頻度の一例を示した図である。

#### 【符号の説明】

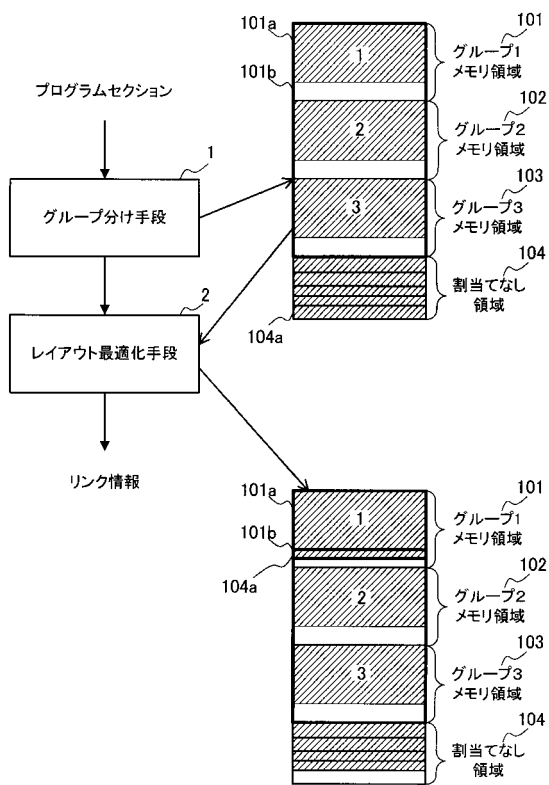
#### 【0058】

- 1 グループ分け手段
- 2 レイアウト最適化手段
- 10 コンパイラ/アセンブラ部

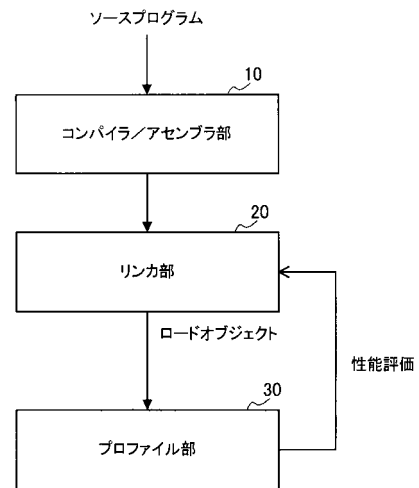
50

- 20 リンカ部
- 30 プロファイル部

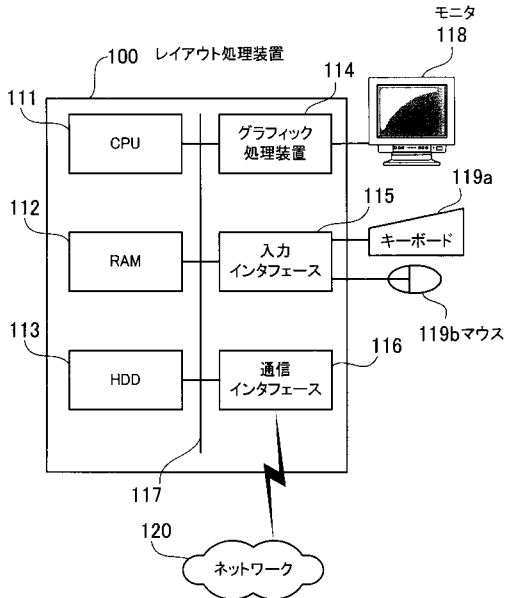
【図1】



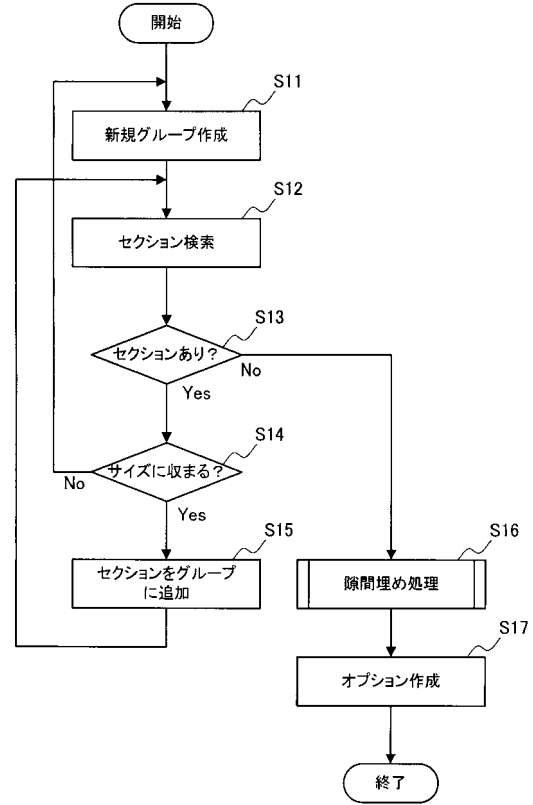
【図2】



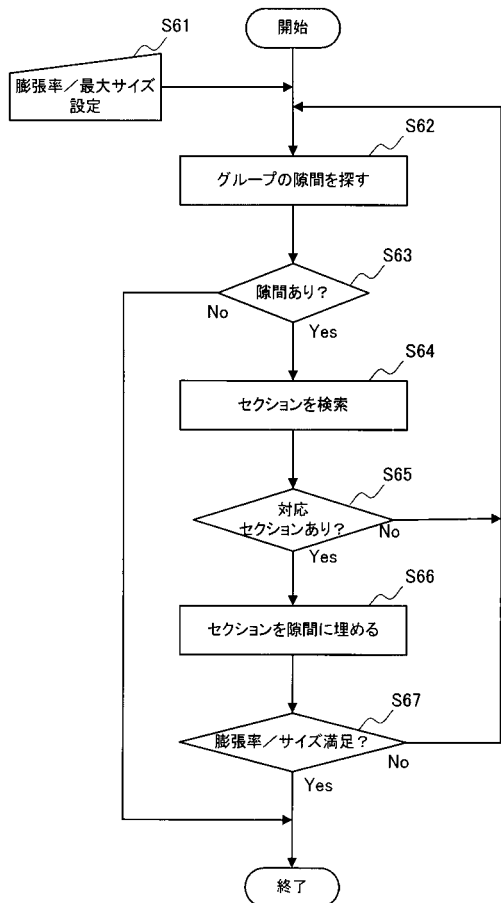
【図3】



【図4】



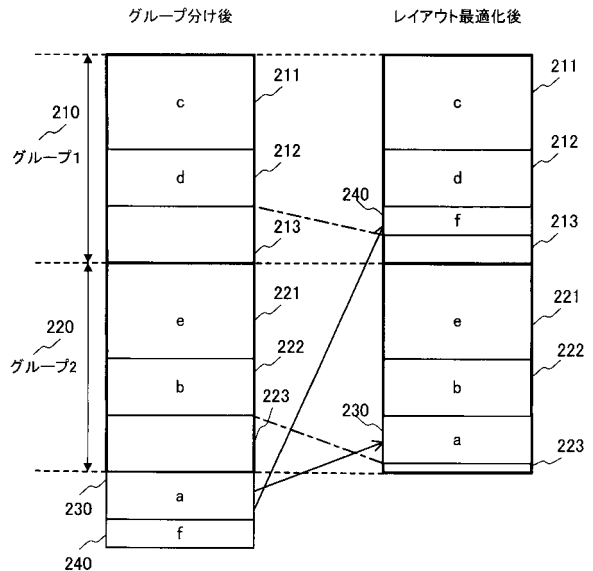
【図5】



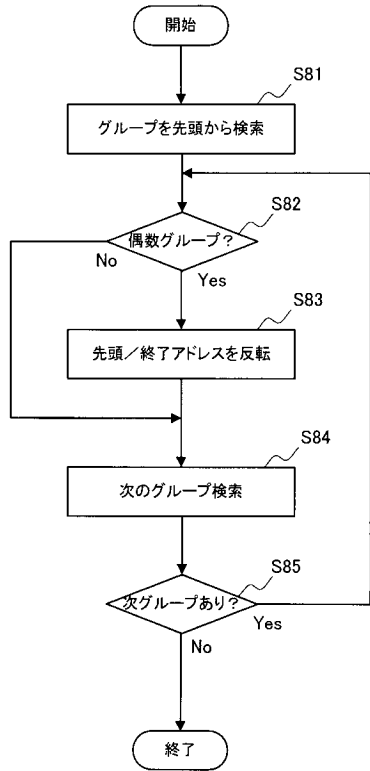
【図6】

セクション名	占有率	参照頻度
a	0.1%	1回
b	2%	15回
c	80%	10000回
d	10%	300回
e	8%	10回
f	0%	0回

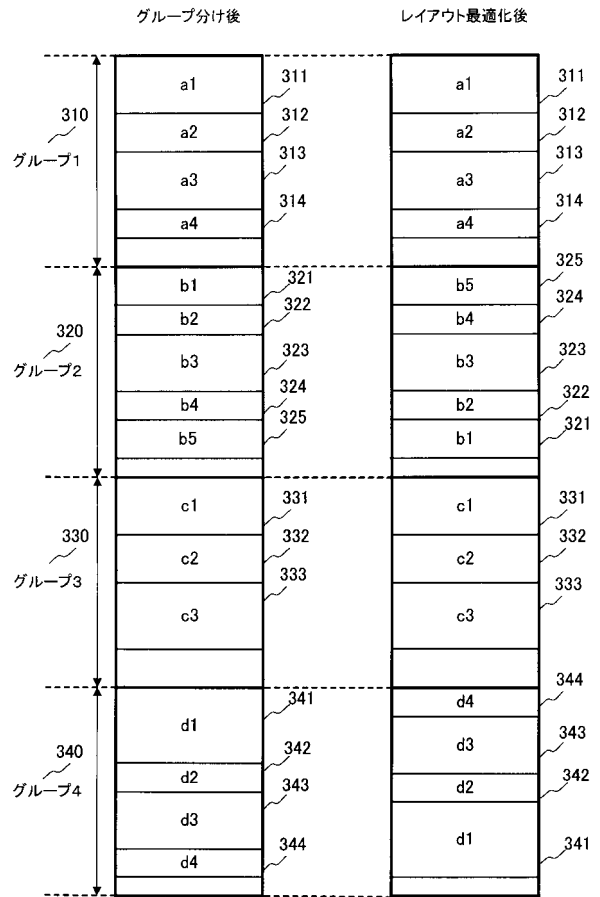
【図7】



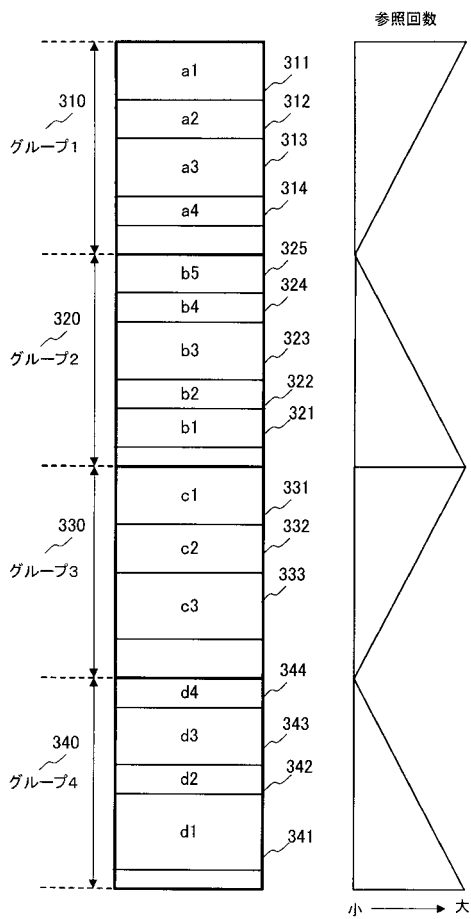
【図8】



【図9】



【図10】





---

フロントページの続き

- (56)参考文献 特開平10-124327(JP,A)  
特開2000-66899(JP,A)  
特開2003-271394(JP,A)  
特開平10-187460(JP,A)  
特開2000-3268(JP,A)  
特開平6-214803(JP,A)  
橋本 敬介 他, 命令キャッシュミス削減のための基本ブロック単位でのコード再配置手法, 並列処理シンポジウム JSPP'99, 社団法人情報処理学会, 1999年 6月 9日, Vol.99, No.6, pp.31-38  
富山 宏之 他, 命令キャッシュのヒット率向上を目的としたコード配置手法, 情報処理学会研究報告, 社団法人情報処理学会, 1995年12月15日, Vol.95, No.119, pp.133-138

(58)調査した分野(Int.Cl., DB名)

G06F 9/45  
G06F 12/08  
JSTPlus(JDreamII)