(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5072599号
(P5072599)
(45)発行日　平成24年11月14日(2012.11.14)　(24)登録日　平成24年8月31日(2012.8.31)

(51)Int.Cl.　F I
G06F 9/44 (2006.01)　G06F 9/06 620A

請求項の数 18 (全 8 頁)

(21)出願番号	特願2007-538925 (P2007-538925)	(73)特許権者	310021766
(86)(22)出願日	平成17年9月23日 (2005.9.23)		株式会社ソニー・コンピュータエンタテインメント
(65)公表番号	特表2008-518355 (P2008-518355A)		東京都港区港南１丁目７番１号
(43)公表日	平成20年5月29日 (2008.5.29)	(74)代理人	100105924
(86)国際出願番号	PCT/US2005/034460		弁理士　森下　賢樹
(87)国際公開番号	WO2006/049740	(74)代理人	100109047
(87)国際公開日	平成18年5月11日 (2006.5.11)		弁理士　村田　雄祐
審査請求日	平成20年9月22日 (2008.9.22)	(74)代理人	100109081
(31)優先権主張番号	10/976,264		弁理士　三木　友由
(32)優先日	平成16年10月28日 (2004.10.28)	(74)代理人	100134256
(33)優先権主張国	米国(US)		弁理士　青木　武司

最終頁に続く

(54)【発明の名称】異種混合アーキテクチャからの複数のオブジェクトファイルを一組のファイルに統合する方法

(57)【特許請求の範囲】

【請求項１】

異なったアーキテクチャをもつ複数のプロセッサを含む異種混合アーキテクチャを有し、前記異種混合アーキテクチャからの複数のオブジェクトファイルを統合するマルチプロセッサシステムであって、

第１プロセッサと互換性をもつコンパイラまたはアセンブラで作成される第１のタイプのオブジェクトコードが動作する第１プロセッサと、

第２プロセッサと互換性をもつコンパイラまたはアセンブラで作成される第２のタイプのオブジェクトコードが動作する第２プロセッサと、

第２プロセッサ上で動作する前記第２のタイプのオブジェクトコードであるＯＣ２コードに対応させて第１プロセッサ上で動作する前記第１のタイプのオブジェクトコードであるＯＣ１コードを生成するように構成されたシステムツールを含み、

前記システムツールは、前記ＯＣ１コードに前記ＯＣ２コードをラッパーの形で同封し、かつ、前記ＯＣ２コードへのポインタであるシンボルの定義を含めることにより前記ＯＣ２コードに対応する前記ＯＣ１コードを生成することを特徴とするマルチプロセッサシステム。

【請求項２】

当該マルチプロセッサシステムは、前記第２プロセッサ上での前記ＯＣ２コードの実行を生じさせるために、前記ＯＣ２コードに対応する前記ＯＣ１コードを前記第１プロセッサ上で実行するように構成されることを特徴とする請求項１に記載のマルチプロセッサシ

ステム。
【請求項３】
ＤＭＡコントローラをさらに含み、前記第１プロセッサはシステムメモリを含み、前記第２プロセッサはローカルストアを含み、当該マルチプロセッサシステムは、データが前記第１プロセッサのシステムメモリにロードされ、ＤＭＡコントローラによって前記第２プロセッサの前記ローカルストアに転送されることによって、そのデータが前記第２プロセッサに渡されるように構成されることを特徴とする請求項１または２に記載のマルチプロセッサシステム。
【請求項４】
前記ＯＣ２コードに対応して生成された前記ＯＣ１コードは、
シンボルの定義をもつヘッダ部と、前記ＯＣ２コードとを含み、
前記シンボルは前記ＯＣ２コードへのポインタであることを特徴とする請求項１から３のいずれかに記載のマルチプロセッサシステム。
【請求項５】
異なったアーキテクチャをもつ複数のプロセッサを含む異種混合アーキテクチャを有するマルチプロセッサシステムにおいて、前記異種混合アーキテクチャからの複数のオブジェクトファイルを統合するための方法であって、
前記マルチプロセッサシステムは、
第１プロセッサと互換性をもつコンパイラまたはアセンブラで作成される第１のタイプのオブジェクトコードが動作する第１プロセッサと、
第２プロセッサと互換性をもつコンパイラまたはアセンブラで作成される第２のタイプのオブジェクトコードが動作する第２プロセッサと、
第２プロセッサ上で動作する前記第２のタイプのオブジェクトコードであるＯＣ２コードへのポインタをもたせて、第１プロセッサ上で動作する前記第１のタイプのオブジェクトコードであるＯＣ１コードを生成するように構成されたシステムツールを含み、
前記システムツールは、前記ＯＣ１コードに前記ＯＣ２コードをラッパーの形で同封し、かつ、前記ＯＣ２コードへのポインタであるシンボルの定義を含めることにより前記ＯＣ２コードに対応する前記ＯＣ１コードを生成し、
前記第１プロセッサ上で前記生成されたＯＣ１コードを実行することにより、前記第２プロセッサ上での前記ＯＣ２コードの実行を生じさせることを特徴とする方法。
【請求項６】
前記生成されたＯＣ１コードは、
シンボルの定義をもつヘッダ部と、前記ＯＣ２コードとを含み、
前記シンボルは前記生成されたＯＣ１コードに含まれた前記ＯＣ２コードへのポインタであることを特徴とする請求項５に記載の方法。
【請求項７】
前記第１プロセッサから前記第２プロセッサへラッパーの形で同封された前記第２のタイプのオブジェクトコードである前記ＯＣ２コードを実行のために渡すステップをさらに含むことを特徴とする請求項５または６に記載の方法。
【請求項８】
前記ＯＣ１コードによる前記ＯＣ２コードへの参照を前記生成されたＯＣ１コードに対する参照として解決するステップをさらに含むことを特徴とする請求項５から７のいずれかに記載の方法。
【請求項９】
前記生成されたＯＣ１コードを実行可能なプログラムに変換するステップをさらに含むことを特徴とする請求項５から８のいずれかに記載の方法。
【請求項１０】
前記生成されたＯＣ１コードをアーカイブに変換するステップをさらに含むことを特徴とする請求項５から９のいずれかに記載の方法。
【請求項１１】

前記生成されたＯＣ１コードをダイナミック共有ライブラリに変換するステップをさらに含むことを特徴とする請求項５から１０のいずれかに記載の方法。
【請求項１２】
異なったアーキテクチャをもつ複数のプロセッサを含む異種混合アーキテクチャを有するマルチプロセッサシステムに、前記異種混合アーキテクチャからの複数のオブジェクトファイルを統合するための方法を実行させるコンピュータプログラムであって、
前記マルチプロセッサシステムは、
第１プロセッサと互換性をもつコンパイラまたはアセンブラで作成される第１のタイプのオブジェクトコードが動作する第１プロセッサと、
第２プロセッサと互換性をもつコンパイラまたはアセンブラで作成される第２のタイプのオブジェクトコードが動作する第２プロセッサと、
第１プロセッサ上で動作する前記第１のタイプのオブジェクトコードであるＯＣ１コードであって、第２プロセッサ上で動作する前記第２のタイプのオブジェクトコードであるＯＣ２コードへのポインタを含むＯＣ１コードを生成するように構成されたシステムツールを含み、
前記システムツールに、前記ＯＣ１コードに前記ＯＣ２コードをラッパーの形で同封し、かつ、前記ＯＣ２コードへのポインタであるシンボルの定義を含めることにより前記ＯＣ２コードに対応する前記ＯＣ１コードを生成させるためのコンピュータコードと、
前記第１プロセッサ上で前記生成されたＯＣ１コードを実行することにより、前記第２プロセッサ上での前記ＯＣ２コードの実行を生じさせるためのコンピュータコードとを含むことを特徴とするコンピュータプログラム。
【請求項１３】
前記生成されたＯＣ１コードは、
シンボルの定義をもつヘッダ部と、
前記ＯＣ２コードとを含み、
前記シンボルは前記生成されたＯＣ１コードに含まれた前記ＯＣ２コードへのポインタであることを特徴とする請求項１２に記載のコンピュータプログラム。
【請求項１４】
前記第１プロセッサから前記第２プロセッサへラッパーの形で同封された前記第２のタイプのオブジェクトコードである前記ＯＣ２コードを実行のために渡すためのコンピュータコードをさらに含むことを特徴とする請求項１２または１３に記載のコンピュータプログラム。
【請求項１５】
前記ＯＣ１コードによる前記ＯＣ２コードへの参照を前記生成されたＯＣ１コードに対する参照として解決するためのコンピュータコードをさらに含むことを特徴とする請求項１２から１４のいずれかに記載のコンピュータプログラム。
【請求項１６】
前記生成されたＯＣ１コードを実行可能なプログラムに変換するためのコンピュータコードをさらに含むことを特徴とする請求項１２から１５のいずれかに記載のコンピュータプログラム。
【請求項１７】
前記生成されたＯＣ１コードをアーカイブに変換するためのコンピュータコードをさらに含むことを特徴とする請求項１２から１６のいずれかに記載のコンピュータプログラム。
【請求項１８】
前記生成されたＯＣ１コードをダイナミック共有ライブラリに変換するためのコンピュータコードをさらに含むことを特徴とする請求項１２から１７のいずれかに記載のコンピュータプログラム。
【発明の詳細な説明】
【技術分野】

【０００１】
本発明は、一般に、オブジェクトファイルの処理に関し、特に、異種混合アーキテクチャからの複数のオブジェクトファイルを統合することに関する。
【背景技術】
【０００２】
異なるプロセッサに対応して別々の名前空間が設けられたマルチプロセッサにおいて、一つの名前空間で定められたプログラムは、もう一つの名前空間の上で定められたプログラムにリファレンスをつけることができる。これらのプロセッサは、異なるマシンタイプを構成しており、異なるアーキテクチャ、異なるインストラクションセット、および異なるオブジェクトファイルの形式をもつことがある。
【発明の開示】
【発明が解決しようとする課題】
【０００３】
リファレンスを解決するための従来の方法には、いくつかの問題がある。たとえば、リンカーは別のプロセッサで生成されたオブジェクトコードを誤って解釈することがあり、不正確にコードを処理することがあった。プログラマは、あるプロセッサで動いているプログラムから別のプロセッサの名前空間にあるプログラムへの呼び出しをハードコーディング（決め打ち）することもできるが、その手順はわずらわしいものになる。仮にハードコーディングを使ったとしても、オブジェクトコードをランタイム参照することはできないし、動的リンクやオブジェクト共有もできず、あるいは、マルチプロセッサの結合された名前空間からのオブジェクトを実行時に取り扱うこともできない。
【０００４】
したがって、異種混合アーキテクチャからの複数のオブジェクトファイルを統合する方法が必要であり、それによって、リンカーやローダーのようなシステムツールがオブジェクトファイルを適切に処理することができるようになり、そして、オブジェクトファイルへのランタイム参照、動的リンクとオブジェクト共有、マルチプロセッサの結合された名前空間からのオブジェクトの実行時の取り扱いも可能になる。
【課題を解決するための手段】
【０００５】
本発明は異種混合アーキテクチャからの複数のオブジェクトコードを統合するための方法である。第１プロセッサが第２プロセッサの名前空間からプログラムを参照するために、第２プロセッサのプログラム用のオブジェクトコードをラッパー（wrapper；包み込み）の形で同封させて、第１プロセッサの名前空間におけるオブジェクトコードを生成する。ラップされた（wrapped；包まれた）オブジェクトのヘッダは、第１プロセッサの名前空間において新しいシンボルを定義する。そのシンボルはラップされたオブジェクトコードに含まれる第２プロセッサのオブジェクトコードへのポインタである。第２プロセッサのオブジェクトコードを直接参照する代わりに、第１プロセッサ上の参照する側のプログラムがラップされたオブジェクトコードを参照する。
【発明を実施するための最良の形態】
【０００６】
本発明とその有利な点をより完全に理解するために、添付の図面と合わせて以下の説明を参照する。
【０００７】
以下の議論において、本発明を完全に理解するために、多数の具体的で詳細なことを述べるが、本発明はそのような具体的で詳細な事項がなくても実施できることは当業者にとって明らかである。他の例では、本発明を不必要なほど詳細にすることでかえってわかりにくくならないように、周知の構成要素については、概略図またはブロックダイヤグラムで説明した。
【０００８】
さらに、ここで述べられたすべての機能は、特に断らない限り、ハードウエアまたはソ

フトウエア、あるいはそれらの組み合わせで実行できることに留意する。しかしながら、より好ましい実施の形態では、特に断らない限り、これらの機能はコンピュータや電子データプロセッサのようなプロセッサによって、そのような機能を実行するためにコード化されたコンピュータプログラムコード、ソフトウエア、および／または集積回路にしたがって実行される。

【０００９】

図１は、異なったアーキテクチャをもつプロセッサを含むマルチプロセッサのブロック図を示す。マルチプロセッサ１００は２つのプロセッサ、ＰＵ１０２とＳＰＵ１１０を含む異種混合アーキテクチャである。あるプロセッサで動作するオブジェクトファイルは、他のプロセッサ上では動作しない。そこで、ＰＵ１０２上で動作しているコードが、ＳＰＵ１１０上で動作するように設計されたコードを参照できるようにした。２つのプロセッサＰＵ１０２とＳＰＵ１１０は、データのアクセスの仕方が異なる。ＰＵ１０２は、第１ＤＭＡコントローラ１０６の管理下でシステムメモリ１０８とキャッシュ１０４にアクセスする。ＤＭＡコントローラ１０６は、システムメモリ１０８、キャッシュ１０４、およびＰＵ１０２との間でデータを転送するためのロード命令とストア命令を処理する。システムメモリ１０８との間で移動するデータは、システムバス１１６上で伝送される。

【００１０】

ＳＰＵ１１０は、ロード命令とストア命令によってシステムメモリ１０８をアクセスすることはない。第２ＤＭＡコントローラ１１４は、データをシステムメモリ１０８からローカルストア１１２へ転送し、ＳＰＵ１１０はそこからロードしたりストアすることができる。このＤＭＡコントローラ１１４は、システムバス１１６を介してシステムメモリ１０８に接続されている。

【００１１】

本発明の他の実施の形態では、マルチプロセッサ１００のアーキテクチャが異なる。ある実施の形態では、マルチプロセッサ１００は複数のＰＵ１０２を含み、全てが一つのシステムメモリを共有する。ある実施の形態では、複数のＰＵ１０２は一つのキャッシュを共有する。別の実施の形態では、一つ以上のＰＵのグループがキャッシュを共有する一方、他のＰＵはキャッシュにアクセスしない。ある実施の形態では、複数のＳＰＵが存在する。ある実施の形態では、ＳＰＵ１００はそれ自身で独立したメモリを有する。

【００１２】

図２は、ＥＬＦ形式のオブジェクトコードをラッパーに同封する様子を例示する。ＳＰＵ１１０ルーチン用のＥＬＦ形式のオブジェクトコード２００は、ＥＬＦヘッダ部２０２と、残りの部分である当該ルーチン用のオブジェクトコード２０４とを含む。その残りの部分は、プログラムとデータを含む。オブジェクトコード２００は、ラッパー（包み込み）２１０を加えることによって、ＰＵ１０２のオブジェクトであるオブジェクトコード２０８に変換される。ラッパー２１０は、ＳＰＵ１１０ルーチンと同じ名前をもつＰＵ１０２オブジェクトのシンボル定義を含んでいる。たとえば、ＳＰＵ１１０ルーチンがＢＡＲ－ＳＰＵであるなら、ラッパー２１０はＰＵ１０２オブジェクトであるシンボルＢＡＲ－ＳＰＵを定義している。オブジェクトコード２０８もまた、オブジェクトコード２００をもち、ＥＬＦヘッダ２１２とその残りの部分であるオブジェクトコード２１４を含む。シンボルＢＡＲ－ＳＰＵは、オブジェクトコード２０８内のオブジェクトコード２００へのポインタもしくは参照である。ＳＰＵオブジェクトコード２００は、一つのＳＰＵオブジェクトＢＡＲ－ＳＰＵ．ｏであり、ラップされた（包まれた）コード２０８は、一つのＰＵオブジェクトＢＡＲ－ＳＰＵ－ＰＵ．ｏである。

【００１３】

ラッピング（包み込み）のプロセスによって、異種混合アーキテクチャからの複数のオブジェクトファイルを統合することが可能になる。ＳＰＵ１１０オブジェクトをラッピングすることにより、他の任意のＰＵ１０２オブジェクトとしてリンクしたり、ロードする目的で扱うことができる一つのＰＵ１０２オブジェクトが作られる。実行の間、ラップされたＳＰＵ１１０オブジェクトは、正しく処理される。結果的に、ラッピング処理により

、ＰＵ１０２オブジェクトとＳＰＵ１１０オブジェクトの統合が可能になる。

【００１４】

たとえば、ＳＰＵ１１０オブジェクトＢＡＲ－ＳＰＵに対するリファレンスを解決するために、リンカーはＰＵ１０２オブジェクトＢＡＲ－ＳＰＵ－ＰＵ．ｏにリンクする。この方法は、静的および動的リンクと、ＳＰＵ１１０オブジェクトのオブジェクト共有をサポートする。同様に、ラッピングにより任意のＳＰＵ１１０のファイルフォーマットをロードすることが可能になる。ラップされたＰＵオブジェクト２０８がロードされる。さらに、ＰＵ１０２のランタイム参照がＳＰＵ１１０オブジェクトに対してなされる。このＰＵ１０２上のランタイム参照は、ＰＵ１０２オブジェクトＢＡＲ－ＳＰＵ－ＰＵに対するものである。

【００１５】

また、ラッピングはＰＵ１０２オブジェクトの名前空間とＳＰＵ１１０オブジェクトの名前空間の明確な分離を可能にする。ＰＵ１０２上で実行中のコードは、直接的にはＳＰＵ１１０オブジェクトを参照する必要はない。その代わり、ＳＰＵ１１０オブジェクトをラップしてＰＵ１０２オブジェクトを生成し、ＰＵ１０２コードは、ラップされたオブジェクトであるＰＵ１０２オブジェクトを参照する。その結果はＰＵ１０２のプログラム参照のための単純なシンボル関連付け（association）でもある。ＰＵ１０２コードが、ＳＰＵ１１０オブジェクトへのポインタであるＰＵ１０２シンボルを参照する。結果的に、ＰＵ１０２オブジェクトとＳＰＵ１１０オブジェクトをあらかじめリンク（pre-linking）して混合する能力が与えられる。最後に、ラッピングプロセスは、静的および動的両方のニーズのためにライブラリをパッケージングするのに好都合である。

【００１６】

図３は、別のプロセッサからの呼び出しの後、あるプロセッサ上でオブジェクトコードを実行する手順を示すフローチャート３００である。ＰＵ１０２上で実行中のプログラムＦＯＯがＳＰＵ１１０上で動くルーチンＢＡＲを呼び出すと、ＢＡＲへの呼び出しは、ＰＵ１０２オブジェクトＢＡＲ－ＳＰＵ－ＰＵ．ｏへの呼び出しであると解釈される。ステップ３０２において、ラップされたコードＢＡＲ－ＳＰＵ－ＰＵ．ｏがＰＵ１０２上で実行される。その後、ステップ３０４において、ラップされたコードＢＡＲ－ＳＰＵ－ＰＵ．ｏに含まれるＢＡＲ用のＳＰＵオブジェクトコードは、ＳＰＵ１１０のローカルストア１１２にＤＭＡ転送される。ステップ３０６において、ＳＰＵ１１０がそのコードを実行し始める。実行が終了すると、ステップ３０８において、その結果がＰＵ１０２にＤＭＡ転送で戻される。

【００１７】

図４は、ＳＰＵ１１０オブジェクトコードを含むラップされたオブジェクトの生成手順を示すフローチャート４００である。一例として、ＳＰＵ１１０ルーチンは、ＢＡＲという名前をつけられている。ステップ４０２において、ＳＰＵ１１０オブジェクトファイルＢＡＲ－ＳＰＵ．ｏがＥＬＦ形式でＢＡＲ用に作成される。このオブジェクトファイルは、プロセッサＳＰＵ１１０と互換性を持つコンパイラまたはアセンブラで作成される。ステップ４０４において、ＰＵ１０２オブジェクトコードＢＡＲ－ＳＰＵ－ＰＵ．ｏを作成するためにこのコード上にラッパーが配置される。ある実施の形態では、ラッパーを作成するためのシステムツールがマルチプロセッサ１００で利用可能である。ステップ４０６において、システムツールは、ラッパー内でＰＵ１０２シンボルＢＡＲ－ＳＰＵを、ＰＵ１０２オブジェクトＢＡＲ－ＳＰＵ－ＰＵ．ｏ内に含まれるＳＰＵ１１０オブジェクトＢＡＲ－ＳＰＵ．ｏへのポインタとして定義する。いったんＳＰＵ１１０ファイルがＰＵ１０２オブジェクトファイル内に埋め込まれると、それは通常のＰＵ１０２ファイルとして扱うことができ、ステップ４０８において、ユーザーはそれを任意のファイルフォーマット、例えば実行可能なダイナミック共用ライブラリやアーカイブ形式などに変換することができる。

【００１８】

以上、好ましい実施の形態を参照して本発明を説明したが、開示された実施形態は例示

であり、何ら限定するものではなく、上記の開示内容には幅広いバリエーション、変形、変更、および置換が考えられること、そして、事例によっては、本発明のいくつかの特徴は、他の特徴を使うことなく利用することもできることに留意する。そのような多くのバリエーションや変形は、当業者が好ましい実施の形態についての上記の説明を再考することにより、望ましいものであるとみなされることもある。したがって、添付の請求項は広く、かつ、本発明の範囲から逸脱することなく解釈されることが適当である。

【図面の簡単な説明】

【００１９】

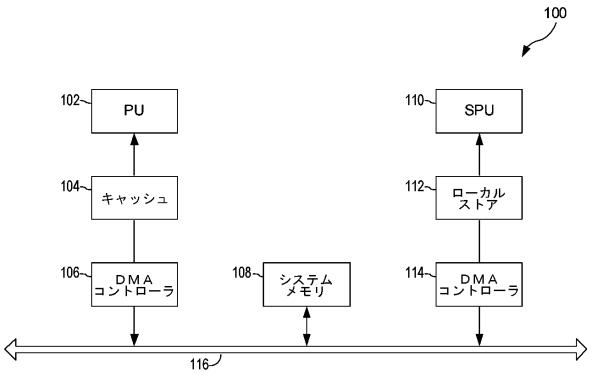
【図１】異なったアーキテクチャをもつプロセッサを含むマルチプロセッサのブロック図である。

【図２】オブジェクトコードをＥＬＦフォーマットでラッパーに同封する様子を例示する図である。

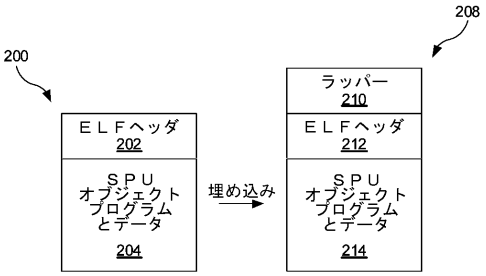
【図３】別のプロセッサからの呼び出しの後、あるプロセッサ上でオブジェクトコードを実行する手順を示すフローチャートである。

【図４】オブジェクトコードを含むラップされたオブジェクトの生成手順を示すフローチャートである。

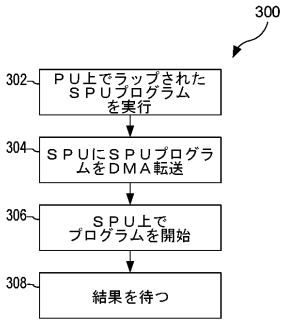
【図１】



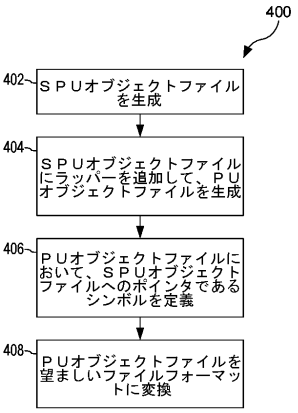
【図２】



【図３】



【図４】



フロントページの続き

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(73)特許権者　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100105924
弁理士　森下　賢樹
(72)発明者　アレックス・チュンヘン・チョウ
アメリカ合衆国７８７５８、テキサス州オースティン、バーネットロード１１４００　インターナショナル・ビジネス・マシーンズ・コーポレイション内
(72)発明者　マイケル・ノルマン・デイ
アメリカ合衆国７８７５８、テキサス州オースティン、バーネットロード１１４００　インターナショナル・ビジネス・マシーンズ・コーポレイション内
(72)発明者　マイケル・スタン・ゴーウェン
アメリカ合衆国７８７５８、テキサス州オースティン、バーネットロード１１４００　インターナショナル・ビジネス・マシーンズ・コーポレイション内
(72)発明者　井上　敬介
東京都港区南青山２丁目６番２１号　株式会社ソニー・コンピュータエンタテインメント内
(72)発明者　ジェームズ・ヘニディス
アメリカ合衆国７８７５８、テキサス州オースティン、バーネットロード１１４００　インターナショナル・ビジネス・マシーンズ・コーポレイション内
(72)発明者　内川　貴幸
アメリカ合衆国、カリフォルニア州、サンノゼ、リンコン、サークル、１０６０

審査官　金子　秀彦

(56)参考文献　特開平０３－２０８１８７（ＪＰ，Ａ）
米国特許第０５２４７６７８（ＵＳ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　9/44