

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4354514号
(P4354514)

(45) 発行日 平成21年10月28日 (2009. 10. 28)

(24) 登録日 平成21年8月7日 (2009. 8. 7)

(51) Int. Cl.	F I
G 0 6 F 9/45 (2006. 01)	G 0 6 F 9/44 3 2 0 F
G 0 6 F 9/44 (2006. 01)	G 0 6 F 9/06 6 2 0 K

請求項の数 6 (全 18 頁)

(21) 出願番号	特願2008-241970 (P2008-241970)	(73) 特許権者	500303571
(22) 出願日	平成20年9月22日 (2008. 9. 22)		株式会社システムズ
(62) 分割の表示	特願2003-396663 (P2003-396663)		東京都品川区西五反田7丁目24番地5号
原出願日	平成15年11月27日 (2003. 11. 27)	(74) 代理人	100115749
(65) 公開番号	特開2009-54168 (P2009-54168A)		弁理士 谷川 英和
(43) 公開日	平成21年3月12日 (2009. 3. 12)	(72) 発明者	長島 廣仲
審査請求日	平成20年9月22日 (2008. 9. 22)		埼玉県さいたま市南区大谷口1429-1
早期審査対象出願		(72) 発明者	浅尾 茂
			東京都品川区荏原4-2-2
		(72) 発明者	鶴巻 正伸
			東京都練馬区練馬1-42-11
		審査官	坂庭 剛史
最終頁に続く			

(54) 【発明の名称】 プログラム自動変換方法、プログラム自動変換装置、およびプログラム

(57) 【特許請求の範囲】

【請求項1】

一の環境で動作していたプログラムである旧ソースプログラムを、他の環境で動作させるための新ソースプログラムに変換するプログラム自動変換方法であって、
 旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を1以上、記憶媒体に格納しており、
 旧ソースプログラムを記憶媒体に格納しており、
 前記旧ソースプログラムにおいて、前記1以上の第一命令パターン情報が何回出現するかを、解析手段を用いて解析する解析ステップと、
 前記解析ステップにおいて第一命令パターン情報の出現回数が、所定回数（所定回数は2以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所をコメントアウトした情報を含むように、当該旧ソースプログラムの記述箇所を当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する命令パターン変換ステップと、
 前記命令パターン変換ステップにおいて前記変換する処理が行われた後、前記命令パターン変換ステップでコメントアウトされた情報を含む手作業前の新ソースプログラムであり、
 前記命令パターン変換ステップで前記旧ソースプログラムを変換した結果である手作業前の新ソースプログラムを、出力手段を用いて出力する新プログラム出力ステップと、
 前記新プログラム出力ステップにおいて、前記手作業前の新ソースプログラムが出力され

た後、前記解析ステップにおいて第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する、前記手作業前の新ソースプログラムの記述箇所に対して、入力手段を用いた、ユーザの手作業による変更の入力を受け付けるステップを具備するプログラム自動変換方法。

【請求項 2】

前記新プログラム出力ステップは、新ソースプログラムの中で、前記命令パターン変換ステップで変換した箇所と変換していない箇所を視覚的に区別して出力する請求項 1 記載のプログラム自動変換方法。

【請求項 3】

一の環境で動作していたプログラムである旧ソースプログラムを、他の環境で動作させるための新ソースプログラムに変換するプログラム自動変換装置であって、

旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を 1 以上、格納している命令パターン対応情報格納部と、

旧ソースプログラムを格納しているプログラム格納部と、

前記旧ソースプログラムにおいて、前記 1 以上の第一命令パターン情報が何回出現するかを解析する解析部と、

前記解析部において第一命令パターン情報の出現回数が、所定回数（所定回数は 2 以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所をコメントアウトした情報を含むように、当該旧ソースプログラムの記述箇所を当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する命令パターン変換部と、

前記命令パターン変換部において前記変換する処理が行われた後、前記命令パターン変換部でコメントアウトされた情報を含む手作業前の新ソースプログラムであり、前記命令パターン変換部で前記旧ソースプログラムを変換した結果である手作業前の新ソースプログラムを出力する新プログラム出力部と、

前記新プログラム出力部において、前記手作業前の新ソースプログラムが出力された後、前記解析部において第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する、前記手作業前の新ソースプログラムの記述箇所に対して、ユーザの手作業による変更の入力を行う入力手段とを具備するプログラム自動変換装置。

【請求項 4】

前記新プログラム出力部は、新ソースプログラムの中で、前記命令パターン変換部が変換した箇所と変換していない箇所を視覚的に区別して出力する請求項 3 記載のプログラム自動変換装置。

【請求項 5】

旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を 1 以上、記憶媒体に格納しており、

旧ソースプログラムを記憶媒体に格納しており、

コンピュータに、

前記旧ソースプログラムにおいて、前記 1 以上の第一命令パターン情報が何回出現するかを解析する解析ステップと、

前記解析ステップにおいて第一命令パターン情報の出現回数が、所定回数（所定回数は 2 以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所をコメントアウトした情報を含むように、当該旧ソースプログラムの記述箇所を当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する命令パターン変換ステップと、

前記命令パターン変換ステップにおいて前記変換する処理が行われた後、前記命令パターン変換部でコメントアウトされた情報を含む手作業前の新ソースプログラムであり、前記

命令パターン変換部で前記旧ソースプログラムを変換した結果である手作業前の新ソースプログラムを、出力手段を用いて出力する新プログラム出力ステップと、
前記新プログラム出力ステップにおいて、前記手作業前の新ソースプログラムが出力された後、前記解析ステップにおいて第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する、前記手作業前の新ソースプログラムの記述箇所に対して、入力手段を用いた、ユーザの手作業による変更の入力を受け付けるステップを実行させるためのプログラム。

【請求項 6】

前記新プログラム出力ステップにおいて、新ソースプログラムの中で、前記命令パターン変換部で変換した箇所と変換していない箇所を視覚的に区別して出力する請求項 5 記載のプログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ある環境で動作していた旧ソースプログラムから他の環境で動作させるための新ソースプログラムを生産するプログラムの生産方法等に関する。

【背景技術】

【0002】

従来、ソースプログラムやスクリプトを自動変換する技術として、コンパイラ差分により発生する記述の違いを自動検出し、自動修正もしくはリストアップして開発作業を向上させる技術があった（特許文献 1 参照）。本技術は、C 言語で記述された原始プログラムを構文解析し、1 トークン毎のデータベースと中間ファイルを作成し、データベース上の対象になる文字列を削除・変更し、データベースや中間ファイルの情報を C 言語ファイルに生成し、目的プログラムを生成するものである。

20

【0003】

また、第二の従来技術として、特定のコンピュータ上に開発されたプログラム・オブジェクトをもとに、他のコンピュータで実行可能なソフトウェアを自動生成する技術があった（特許文献 2 参照）。本技術は、与えられたコンピュータ資源のもとで実行可能なオブジェクトコードは、セグメント分割器により、命令コード領域 D 2 とデータコード領域 D 3 に分割される。続いて、命令コード領域 D 2 は、命令コード変換器により、命令セットや OS プリミティブを解析し、高級プログラム言語による命令文に変換される。また、データコード領域 D 3 は、参照している命令コードの種別に応じたデータ型を判別し、データ構文変換器により、高級プログラム言語によるデータ定義文に変換される。さらに、参照領域変換器が、データ定義文に対して、命令文が利用するための参照名を付加し、命令文が参照名を利用する形式に変換して高級言語プログラムを生成するものである。

30

【特許文献 1】特開 2002-41286（第 1 頁、図 1 等）

【特許文献 2】特開平 6-83630（第 1 頁、図 1 等）

【発明の開示】

【発明が解決しようとする課題】

【0004】

40

しかしながら、上記の特許文献 1 の技術は、変換対象のソースプログラムが記載されたプログラム言語の仕様が有するすべての命令を自動変換することを前提としている、と考えられる。また、上記の特許文献 2 の技術は、特定のコンピュータ上におけるすべてのプログラム・オブジェクトを自動変換することを前提としている、と考えられる。つまり、従来技術において、一のプログラムの動作環境を移行するために、一のプログラムを移行後の環境においても効率良く動作させることを考慮していない。従って、従来技術では、変換対象とするソースプログラムのどの命令パターンを自動変換し、どの命令パターンを手作業で行うかが判断できない。従来技術において、一のプログラムを自動変換または半自動変換する自動変換プログラムを開発する場合に、どの命令パターンに対応する自動変換プログラムを開発すべきかが判断できず、効率的に旧ソースプログラムから、高品質な

50

新ソースプログラムを生産できなかった。具体的には、従来技術では、変換対象のプログラムにおいて1度しか出現しない命令パターンも自動変換するプログラムを開発することが前提となっている。

【課題を解決するための手段】

【0005】

上記の課題を解決するために、旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を1以上、記憶部に格納しており、旧ソースプログラムにおいて、前記1以上の第一命令パターン情報が何回出現するかを、解析手段を用いて解析する解析ステップと、前記解析ステップにおける解析結果を、出力手段を用いて出力する解析結果出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が、所定回数（所定回数は2以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所を、当該第一命令パターン情報と対になる第二命令パターン情報に対応するように、変換手段を用いて変換する命令パターン変換ステップと、前記命令パターン変換ステップにおいて前記旧ソースプログラムを変換した結果である新ソースプログラムを、出力手段を用いて出力する新プログラム出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所に対して、ユーザの手作業による当該旧ソースプログラムの記述箇所を新ソースプログラムに変更する、入力手段を用いた入力を受け付けるステップを具備する、ある環境で動作していた旧ソースプログラムから、他の環境で動作させるための新ソースプログラムを生産するプログラムの生産方法により、効率的に不具合が少ないソースプログラムを生産できる。かかるソースプログラムの生産方法は、旧環境で動作していた旧ソースプログラムから、新しい環境で動作させるための高品質な新ソースプログラムを効率的に生産するプログラムの生産方法である。

【0006】

また、旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を1以上、格納している命令パターン対応情報格納部と、旧ソースプログラムにおいて、前記1以上の第一命令パターン情報が何回出現するかを解析する解析部と、前記解析部における解析結果を出力する解析結果出力部と、前記解析部において第一命令パターン情報の出現回数が、所定回数（所定回数は2以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所を、当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する命令パターン変換部と、前記命令パターン変換部において前記旧ソースプログラムを変換した結果である新ソースプログラムを出力する新プログラム出力部と、前記解析部において第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所に対して、ユーザの手作業による、当該旧ソースプログラムの記述箇所を新ソースプログラムに変更する入力を受け付ける入力受付部を具備する、ある環境で動作していた旧ソースプログラムから他の環境で動作させるための新ソースプログラムを生産するプログラムの生産装置であり、かかる生産装置により、旧環境で動作していた旧ソースプログラムから、新しい環境で動作させるための高品質な新ソースプログラムを効率的に生産できる。

さらに、上記新プログラム出力部は、新ソースプログラムの中で、命令パターン変換部が変換した箇所と変換していない箇所を視覚的に区別して出力することが好ましい。

【発明の効果】

【0007】

本発明によれば、旧環境で動作していた旧ソースプログラムから、新しい環境で動作させるための高品質な新ソースプログラムを効率的に生産できる。

【発明を実施するための最良の形態】

【0008】

以下、本発明のプログラム解析装置およびプログラム自動変換装置等の実施形態について図面を参照して説明する。なお、実施の形態において同じ符号を付した構成要素は同様の動作を行うので、再度の説明を省略する場合がある。

(実施の形態1)

【0009】

図1は、プログラム解析装置のブロック図を示す。プログラム解析装置は、命令パターン情報格納部101、プログラム格納部102、解析部103、解析結果出力部104を有する。プログラム解析装置は、例えば、MPUやメモリやハードディスクなどを備えたパーソナルコンピュータである。

【0010】

命令パターン情報格納部101は、プログラムの命令パターンを示す情報である命令パターン情報を1以上格納している。命令パターン情報格納部101は、不揮発性の記録媒体、または揮発性の記録媒体でも実現可能である。

【0011】

プログラム格納部102は、プログラムを格納している。プログラム格納部102は、不揮発性の記録媒体、または揮発性の記録媒体でも実現可能である。プログラムとは、ソースプログラムであり、その言語は問わない。プログラムは、いわゆるスクリプトと言われるHTML、XMLなども含む。また、プログラムは、手続き型のプログラム、宣言型のプログラム等問わない、また、プログラムは、プログラムが動作するためのデータ群（ヘッダーファイルなど）も含む。

【0012】

解析部103は、プログラム格納部102に格納されているプログラムにおいて、1以上の命令パターン情報が何回出現するかを解析する。解析部103は、通常、MPUやメモリ等から実現され得る。解析部103が解析するための処理手順は、通常、ソフトウェアで実現され、当該ソフトウェアはROM等の記録媒体に記録されている。但し、ハードウェア（専用回路）で実現しても良い。

【0013】

解析結果出力部104は、解析部103における解析結果を出力する。出力とは、通常、ディスプレイへの表示をいうが、プリンタへの印字、他の装置への送信なども含む。また、出力とは、ハードディスクなどの記録媒体への蓄積も含む概念である。また、解析結果出力部104は、出力ディスプレイや記録媒体等を含むと考えても含まないと考えても良い。解析結果出力部104は、例えば、出力デバイスのドライバーソフト等で実現され得る。

以下、本プログラム解析装置の動作について図2のフローチャートを用いて説明する。

【0014】

(ステップS201) 解析部103は、プログラム格納部102のプログラムを読み込む。このプログラムは、解析対象のプログラムである。プログラム格納部102に複数のプログラムが格納されている場合、例えば、解析部103は、ユーザが指示したプログラムを読み込む。

(ステップS202) 解析部103は、命令パターン情報格納部101に格納されている1以上の命令パターン情報を読み込む。

【0015】

(ステップS203) 解析部103は、命令パターン情報の出現回数である命令パターン出現回数をすべて0にする。なお、すべての命令パターン情報に対応する命令パターン出現回数は、例えば、解析部103が保持しているメモリに記憶されている。

【0016】

(ステップS204) 解析部103は、カウンタ*i*に1を代入する。このカウンタは、ステップS201で読み込んだプログラムのプログラムカウンタである。つまり、解析部103は、ステップS201で読み込んだプログラムの*i*番目の行を以下の処理により解析する。なお、ここでは、プログラムは、1行1命令パターンのプログラムである、とす

10

20

30

40

50

る。

【0017】

(ステップS205) 解析部103は、ステップS201で読み込んだプログラムにおいて、i番目のプログラムソース行が存在するか否かを判断する。i番目のプログラムソース行が存在すればステップS206に行き、i番目のプログラムソース行が存在しなければステップS213に飛ぶ。

【0018】

(ステップS206) 解析部103は、カウンタjに1を代入する。カウンタjは、ステップS202で読み込んだ1以上の命令パターン情報のうち、解析に利用する命令パターン情報を取得するためのカウンタである。

10

【0019】

(ステップS207) 解析部103は、i番目のプログラムソース行は、j番目の命令パターン情報に対応する命令パターンの行であるか否かを判断する。i番目のプログラムソース行がj番目の命令パターンであればステップS208に行き、j番目の命令パターンでなければステップS210に飛ぶ。i番目のプログラムソース行がj番目の命令パターン情報に対応するか否かの判断は、構文解析、字句解析等で可能であり、既存技術であるので、ここでの詳細な説明は省略する。

(ステップS208) 解析部103は、j番目の命令パターン情報に対応する命令パターン出現回数を1インクリメントする。

(ステップS209) 解析部103は、iをインクリメントする。ステップS205に戻る。

20

(ステップS210) 解析部103は、jをインクリメントする。

【0020】

(ステップS211) 解析部103は、j番目の命令パターンが、命令パターン情報格納部101に存在するか否かを判断する。j番目の命令パターンが存在すればステップS207に戻り、j番目の命令パターンが存在しなければステップS212に行く。

(ステップS212) 解析部103は、「どの命令パターンにも該当しない旨」のエラー出力をする。但し、このエラー出力処理はなくても良い。

(ステップS213) 解析結果出力部104は、解析結果を出力する。解析結果とは、各命令パターン情報の命令パターン出現回数である。

30

【0021】

以下、本実施の形態におけるプログラム解析装置等の具体的な動作について説明する。図3は、命令パターン情報格納部101が保持している命令パターン情報管理表である。命令パターン情報管理表は、「ID」「命令パターン情報」を有するレコードを複数格納している。「ID」はレコードを識別する情報であり、表管理上利用される情報である。「命令パターン情報」は、命令パターン情報を格納する属性である。

【0022】

また、図4は、解析部103が保持している命令パターン出現回数管理表である。命令パターン出現回数管理表は、「ID」「出現回数」を有するレコードを1以上有する。「ID」は、図3の命令パターン情報管理表の「ID」に対応している。「出現回数」は、命令パターンの出現回数を示す。「出現回数」の初期値は、すべて「0」である。

40

【0023】

図5は、プログラム格納部102に格納されているプログラムの例である。プログラム解析装置は、ユーザの指示により、図5のプログラムの解析を開始する。図5のプログラムにおいて「*」で始まる行はコメント行である。なお、プログラム解析装置は、コメント行であることを判断した後の解析は行わない。

【0024】

解析部103は、まず、図5のプログラムの1行目を読み込む。1行目はコメント行であるので、次の行に移る(プログラムカウンタを進める)。解析部103は、2行目から8行目までもコメント行であるので、解析は行わない。

50

【0025】

次に、解析部103は、プログラムの9行目「IDENTIFICATION DIVISION.」を読み込む。解析部103は、プログラムの9行目は、図3の命令パターン情報管理表の「ID=1」の命令パターン情報に対応する、と判断する。そして、解析部103は、図4の命令パターン出現回数管理表の「ID」=「1」のレコードの「出現回数」の値を1インクリメントする。その結果が、図6の命令パターン出現回数管理表である。

【0026】

次に、解析部103は、プログラムの10行目「PROGRAM-ID. PSD712.」を読み込む。解析部103は、プログラムの10行目は、図3の命令パターン情報管理表の「ID=2」の命令パターン情報に対応する、と判断する。なお、図3の命令パターン情報管理表の「変数01」「変数02」などは、変数を意味する。プログラムにおいて、変数は、英数字の文字列で構成される。ここでは、「PSD712」は変数である。そして、解析部103は、図6の命令パターン出現回数管理表の「ID」=「2」の「出現回数」の値を1インクリメントする。

以上のような処理を、図5のプログラムのすべての行に対して行う。そして、図7に示す命令パターン出現回数管理表を得る。

【0027】

次に、解析結果出力部104は、図8に示すような解析結果を出力する。解析結果は、「ID」「回数」「命令パターン」を有する。「回数」は出現回数である。また、図8の解析結果は、図3の命令パターン情報管理表と、図7の命令パターン出現回数管理表から構成される。具体的には、図8の解析結果は、図3の命令パターン情報管理表の「ID」と、図7の命令パターン出現回数管理表「ID」をキーとして、結合演算され、出力された結果である。

【0028】

以上、本実施の形態によれば、変換対象とするソースプログラムが含む命令パターンの解析が可能となり、その結果、効率的にソースプログラムを変換する自動変換プログラムを開発できる。なお、本実施の形態における解析結果を受けて、プログラムを自動変換する処理については、実施の形態2以降で述べる。

【0029】

なお、本実施の形態において、ある一定以上の回数が出現する命令パターンを他の命令パターンと視覚的に区別して表示しても良い。かかる場合、例えば、図9に示すような解析結果が表示される。図9によれば、3回以上出現する命令パターンを他の命令パターンと視覚的に区別して表示している。なお、一定以上の回数が出現する命令パターンを他の命令パターンと視覚的に区別して表示する態様は、図9における態様に限らず、何でも良い。

【0030】

また、本実施の形態において説明した動作は、コンピュータ読み取り可能なプログラムで実現しても良い。当該プログラムは、CD-ROMなどの記録媒体に記録されて流布しても良いし、ネットワーク配信により流布しても良いし、放送で流布しても良い。かかることも、他の実施の形態においても同様である。なお、本実施の形態において説明した動作を実現するプログラムは、コンピュータに、プログラムの命令パターンを示す情報である命令パターン情報を1以上格納しており、プログラムにおいて、1以上の命令パターン情報が何回出現するかを解析する解析ステップと、解析ステップにおける解析結果を出力する解析結果出力ステップを実行させるためのプログラム、である。

【0031】

さらに、本実施の形態において、一のプログラム解析装置でプログラムの解析を行ったが、複数の装置で処理を分担しても良い。つまり、プログラムの命令パターンを示す情報である命令パターン情報を1以上格納しており、プログラムにおいて、1以上の命令パターン情報が何回出現するかを解析する解析ステップと、解析ステップにおける解析結果を

10

20

30

40

50

出力する解析結果出力ステップを具備するプログラム解析方法により、1以上の装置を用いてプログラムが解析できれば、上述した効果を奏することとなる。

(実施の形態2)

【0032】

図10は、プログラム自動変換装置のブロック図を示す。プログラム自動変換装置は、命令パターン対応情報格納部1001、プログラム格納部102、解析部103、条件格納部1002、命令パターン変換部1003、新プログラム出力部1004を具備する。プログラム自動変換装置は、例えば、MPUやメモリやハードディスクなどを備えたパーソナルコンピュータである。

【0033】

命令パターン対応情報格納部1001は、旧プログラム(変換対象のプログラム)の命令パターンを示す情報である第一命令パターン情報と新プログラム(変換後のプログラム)の命令パターンを示す情報である第二命令パターン情報の対を1以上格納している。命令パターン対応情報格納部1001は、不揮発性の記録媒体、または揮発性の記録媒体でも実現可能である。

【0034】

条件格納部1002は、第一命令パターン情報に対応する旧プログラムの一部を第二命令パターン情報に対応するように自動変換する条件を格納している。条件格納部1002は、不揮発性の記録媒体、または揮発性の記録媒体でも実現可能である。条件は、命令パターン変換部1003が行う処理を実現するプログラム中に埋め込まれている(ヘッダファイルなどに定義されている場合も含む)場合が多い。なお、条件は、解析部103の解析結果に基づく条件である。

【0035】

命令パターン変換部1003は、プログラム格納部102に格納されているプログラム(旧プログラム)に対して、条件格納部1002に格納されている条件に合致する第一命令パターン情報に対応する旧プログラムの一部を、当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する処理を行う。命令パターン変換部1003は、通常、MPUやメモリ等から実現され得る。命令パターン変換部1003が情報を変換するための処理手順は、通常、ソフトウェアで実現され、当該ソフトウェアはROM等の記録媒体に記録されている。但し、ハードウェア(専用回路)で実現しても良い。

【0036】

新プログラム出力部1004は、命令パターン変換部1003がプログラムを変換した結果である新プログラムを出力する。出力とは、通常、ディスプレイへの表示をいうが、プリンタへの印字、他の装置への送信、記録媒体(ハードディスク、デジタル多目的ディスクなど)への蓄積なども含む。また、新プログラム出力部1004は、出力ディスプレイや記録媒体を含むと考えると含まないとも考えても良い。新プログラム出力部1004は、例えば、出力デバイスのドライバソフト等で実現され得る。

以下、本プログラム自動変換装置の動作について図11のフローチャートを用いて説明する。

【0037】

(ステップS1101) プログラム解析ルーチンを実行する。本プログラム解析ルーチンは、図2のフローチャートを用いて説明したプログラムの解析処理であり、変換対象のプログラムである旧プログラム中における、命令パターンごとの出現回数を出力する。

【0038】

(ステップS1102) 命令パターン変換部1003は、プログラム格納部102のプログラムを読み込む。プログラム格納部102に2以上のプログラムが存在する場合は、例えば、命令パターン変換部1003は、ユーザが指示したプログラムを読み込む。

(ステップS1103) 命令パターン変換部1003は、命令パターン対応情報格納部1001が保持している1以上の第一命令パターン情報を読み込む。

【0039】

(ステップS1104) 命令パターン変換部1003は、カウンタ*i*に1を代入する。このカウンタは、ステップS1102で読み込んだプログラムのプログラムカウンタである。つまり、命令パターン変換部1003は、ステップS1102で読み込んだプログラムの*i*番目の行を以下の処理により変換する。なお、ここでは、プログラムは、1行1命令パターンのプログラムである、とする。

【0040】

(ステップS1105) 命令パターン変換部1003は、ステップS1102で読み込んだプログラムにおいて、*i*行目のプログラムソースが存在するか否かを判断する。*i*行目のプログラムソースが存在すればステップS1106に行き、*i*行目のプログラムソースが存在しなければステップS1116に飛ぶ。

10

【0041】

(ステップS1106) 命令パターン変換部1003は、カウンタ*j*に1を代入する。カウンタ*j*は、ステップS1103で読み込んだ1以上の第一命令パターン情報のうち、変換に利用する第一命令パターン情報を取得するためのカウンタである。

【0042】

(ステップS1107) 命令パターン変換部1003は、*i*行目のプログラムソースは、*j*番目の第一命令パターン情報に対応するか否かを判断する。*j*番目の第一命令パターン情報に対応すればステップS1108に行き、*j*番目の第一命令パターン情報に対応しなければステップS1113に飛ぶ。なお、*i*行目のプログラムソースが*j*番目の第一命令パターン情報に対応するか否かは、*i*行目のプログラムソースの構文解析、字句解析により可能である。

20

【0043】

(ステップS1108) 命令パターン変換部1003は、*j*番目の第一命令パターン情報が条件格納部1002の条件に合致するか否かを、ステップS1101のプログラム解析ルーチンの解析結果に基づいて決定する。*j*番目の第一命令パターン情報が条件に合致すればステップS1109に行き、*j*番目の第一命令パターン情報が条件に合致しなければステップS1115に飛ぶ。

(ステップS1109) 命令パターン変換部1003は、*j*番目の第一命令パターン情報と対になる第二命令パターン情報を、命令パターン対応情報格納部1001から取得する。

30

(ステップS1110) 命令パターン変換部1003は、*i*行目のプログラムソースを、ステップS1109で取得した第二命令パターン情報が示す命令パターンに変換する。

【0044】

(ステップS1111) 命令パターン変換部1003は、ステップS1110で変換した*i*行目のソースを新プログラムに追加する。なお、新プログラムは、例えば、あるファイルに格納され、初期化段階（図示しない）でファイル生成とファイルオープンがされている、とする。

(ステップS1112) カウンタ*i*をインクリメントする。ステップS1105に戻る。

(ステップS1113) カウンタ*j*をインクリメントする。

40

【0045】

(ステップS1114) 命令パターン変換部1003は、*j*番目の第一命令パターン情報が存在するか否かを判断する。*j*番目の第一命令パターン情報が存在すればステップS1107に戻り、*j*番目の第一命令パターン情報が存在しなければステップS1115に行く。

(ステップS1115) 命令パターン変換部1003は、*i*行目のソースを新プログラムに追加する。ここでの*i*行目のソースは、例えば、コメント行である。

(ステップS1116) 新プログラム出力部1004は、変換された後のプログラム（新プログラム）を出力する。ここでの出力は、記録媒体（ハードディスク等）に蓄積されるだけの処理も含む。

50

【0046】

なお、第一命令パターン情報が条件格納部1002の条件に合致せず、第一命令パターンを第二命令パターンに自動変換しなかった第一命令パターンに関して、プログラマーが、キーボードなどの入力手段を用いて、手作業で第二命令パターンに書き換える。つまり、プログラム自動変換装置は、図示しないが、キーボードやマウスなどの入力手段を具備する。

【0047】

以下、本実施の形態におけるプログラム変換装置の具体的な動作について説明する。図12は、命令パターン対応情報格納部1001が保持している命令パターン情報管理表である。命令パターン情報管理表は、「ID」「第一命令パターン情報」「第二命令パターン情報」を有するレコードを複数格納している。「ID」はレコードを識別する情報である。命令パターン情報管理表は、変換対象のプログラム中に出現する第一命令パターン情報のソース行を対になる第二命令パターン情報に対応するように変換する場合に利用する対応表である。具体的には、「第一命令パターン情報」が「SOURCE-COMPUTER. XXXXXX.」である場合には、「第二命令パターン情報」「#DEL# * SOURCE-COMPUTER. XXXXXX. #CHG# SOURCE-COMPUTER. IBM-AS400.」に変換することを示す。なお「XXXXXX」は、任意の変数名を示す。また、「#DEL#」は削除したことを示す。「#CHG#」は変換した結果であることを示す。また、本ソースにおいて「*」はコメント行であり、「#」から「#」はコメントであることを示す。

10

【0048】

また、図5は、プログラム格納部102に格納されている旧プログラムの例である。図8は、解析部103が解析した結果を示す命令パターン出現回数管理表である。さらに、条件格納部1002は、出現回数が2回以上の第一命令パターン情報のプログラム行を第二命令パターン情報に変換する、という条件を保持している、とする。

20

【0049】

そして、命令パターン変換部1003が、図5の旧プログラムを変換して生成し、新プログラム出力部1004が出力した新プログラムの例を図13に示す。図13において、新プログラムにおいて新たに生成された行が下線で示されている。つまり、新プログラムにおいて、変換された行と変換されていない行が視覚的に区別して表れている。

図13の新プログラムにおいて、(1)は、命令パターン情報管理表の「ID=1」の行により変換された箇所である。(2)は、命令パターン情報管理表の「ID=2」の行により変換された箇所である。

30

【0050】

なお、図12の命令パターン情報管理表において、説明の便宜上、旧プログラム1行に対して、新プログラムn行(nは自然数)が対応する場合のみを記載したが、旧プログラムm行(mは2以上の自然数)に対して、新プログラムn行(nは自然数)が対応しても良い。

【0051】

以上、本実施の形態によれば、変換対象とするソースプログラムが含む命令パターンの解析が可能となり、その結果、効率的にソースプログラムを自動変換または半自動変換できる。具体的には、ソースプログラムを変換するためのプログラムを開発するまでもない一定未満の出現回数の命令パターンは、手作業で修正し、一定以上の出現回数(たとえば、2回以上の出現回数)の命令パターンのみ、ソースプログラムを変換するためのプログラムを開発しておき、当該変換プログラムを用いて自動変換することにより、効率的にプログラムを変換できる。かかるプログラム自動変換装置により、例えば、ある環境で動作していたプログラムを新しい環境で動作させる作業が非常に効率的に、かつ安全に行える。

40

【0052】

なお、本実施の形態において、変換プログラムを開発するまでもない一定未満の出現回数の命令パターンは、手作業で旧ソースプログラムを修正するので、プログラム自動変換

50

装置は、手作業で修正するための入力手段を具備する。入力手段とは、例えば、キーボードやマウス等である。

【0053】

なお、本実施の形態において、図14に示すように、変換後の新プログラムにおいて、手作業によって変換した箇所と、自動変換した箇所と、変換していない箇所を視覚的に区別する態様で、新プログラムを蓄積や表示等しても良い。図14の新プログラムにおいて、「@」から「@」が手作業による作業を示すコメントである。「@DEL@」は手作業で削除した行であることを示し、「@ADD@」は手作業で追加した行であることを示す。

【0054】

また、本実施の形態において、自動変換するか手作業で変換するかを判断するための条件は、プログラムにおける第一命令パターン情報の出現回数をパラメータとする条件（上記例では、2回以上の出現回数の場合に自動変換するという条件）であったが、他の条件により決定しても良い。また、ユーザの入力により自動変換する第一命令パターンを指示する構成でも良い。

【0055】

また、本実施の形態において説明した動作は、コンピュータ読み取り可能なプログラムで実現しても良い。当該プログラムは、CD-ROMなどの記録媒体に記録されて流布しても良いし、ネットワーク配信により流布しても良いし、放送で流布しても良い。かかることも、他の実施の形態においても同様である。なお、本実施の形態において説明した動作を実現するプログラムは、コンピュータに、旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を1以上格納しており、旧ソースプログラムにおいて、前記1以上の第一命令パターン情報が何回出現するかを解析する解析ステップと、前記解析ステップにおける解析結果を出力する解析結果出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が、所定回数（所定回数は2以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所を、当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する命令パターン変換ステップと、前記命令パターン変換ステップで前記旧ソースプログラムを変換した結果である新プログラムを出力する新プログラム出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所に対して、ユーザの手作業による当該旧ソースプログラムの記述箇所を新ソースプログラムに変更する入力を受け付けるステップを実行させるためのプログラム、である。

【0056】

また、本実施の形態において、一のプログラム自動変換装置でプログラムの変換を行ったが、複数の装置で処理を分担しても良い。つまり、以下のプログラム自動変換方法を複数の装置で実現しても良い。旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を1以上、記憶部に格納しており、旧ソースプログラムにおいて、前記1以上の第一命令パターン情報が何回出現するかを、解析手段を用いて解析する解析ステップと、前記解析ステップにおける解析結果を、出力手段を用いて出力する解析結果出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が、所定回数（所定回数は2以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所を、当該第一命令パターン情報と対になる第二命令パターン情報に対応するように、変換手段を用いて変換する命令パターン変換ステップと、前記命令パターン変換ステップにおいて前記旧ソースプログラムを変換した結果である新ソースプログラムを、出力手段を用いて出力する新プログラム出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所に対して、ユーザの手作業による当該旧ソースプログラムの記述箇所を新ソースプログラムに変更する、入力手段を用いた入力を

10

20

30

40

50

受け付けるステップを具備する、ある環境で動作していたプログラムを他の環境で動作させるためのプログラム自動変換方法により、効率的に、かつ、不具合が少なくプログラムが移植できる。

【0057】

また、本実施の形態において、上述したプログラム自動変換装置は、新しい環境（つまり、移植先の環境）で動作するプログラムを生産する方法を実施する装置である、といえる。つまり、プログラム自動変換装置は、以下のプログラム生産方法を実現する。それは、旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を1以上、記憶部に格納しており、旧ソースプログラムにおいて、前記1以上の第一命令パターン情報が何回出現するかを解析する解析ステップと、前記解析ステップにおける解析結果を出力する解析結果出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が、所定回数（所定回数は2以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所を、当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する命令パターン変換ステップと、前記命令パターン変換ステップにおいて前記旧ソースプログラムを変換した結果である新ソースプログラムを出力する新プログラム出力ステップと、前記解析ステップにおいて第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所に対して、ユーザの手作業による当該旧ソースプログラムの記述箇所を新ソースプログラムに変更する入力を受け付けるステップを具備する、ある環境で動作していた旧ソースプログラムから他の環境で動作させるための新ソースプログラムを生産するプログラムの生産方法、である。また、上記のプログラムの生産方法における新プログラム出力ステップは、新ソースプログラムの中で、前記命令パターン変換ステップで変換した箇所と変換していない箇所を視覚的に区別して出力するプログラムの生産方法、でも良い。

【0058】

さらに、本実施の形態において、上述したプログラム自動変換装置は、新しい環境（つまり、移植先の環境）で動作するプログラムを生産する装置である、といえる。かかるプログラムの生産装置は、旧ソースプログラムの命令パターンを示す情報である第一命令パターン情報と新ソースプログラムの命令パターンを示す情報である第二命令パターン情報の対を1以上、格納している命令パターン対応情報格納部と、旧ソースプログラムにおいて、前記1以上の第一命令パターン情報が何回出現するかを解析する解析部と、前記解析部における解析結果を出力する解析結果出力部と、前記解析部において第一命令パターン情報の出現回数が、所定回数（所定回数は2以上である）以上であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所を、当該第一命令パターン情報と対になる第二命令パターン情報に対応するように変換する命令パターン変換部と、前記命令パターン変換部において前記旧ソースプログラムを変換した結果である新ソースプログラムを出力する新プログラム出力部と、前記解析部において第一命令パターン情報の出現回数が前記所定回数未満であると解析された第一命令パターン情報に対応する旧ソースプログラムの記述箇所に対して、ユーザの手作業による、当該旧ソースプログラムの記述箇所を新ソースプログラムに変更する入力を受け付ける入力受付部を具備する、ある環境で動作していた旧ソースプログラムから他の環境で動作させるための新ソースプログラムを生産するプログラムの生産装置、である。

【産業上の利用可能性】

【0059】

以上のように、本発明にかかるプログラムの生産方法および生産装置は、ある環境で動作していた旧ソースプログラムから他の環境で動作させるための新ソースプログラムを生産するプログラムの生産方法、生産装置等として利用できる。

【図面の簡単な説明】

【0060】

10

20

30

40

50

【図 1】実施の形態 1 におけるプログラム解析装置のブロック図

【図 2】実施の形態 1 におけるプログラム解析装置の動作について説明するフローチャート

【図 3】実施の形態 1 における命令パターン情報管理表

【図 4】実施の形態 1 における命令パターン出現回数管理表

【図 5】実施の形態 1 におけるプログラムの例を示す図

【図 6】実施の形態 1 における命令パターン出現回数管理表を示す図

【図 7】実施の形態 1 における命令パターン出現回数管理表を示す図

【図 8】実施の形態 1 における解析結果の例を示す図

【図 9】実施の形態 1 における解析結果の表示例を示す図

10

【図 10】実施の形態 2 におけるプログラム自動変換装置のブロック図

【図 11】実施の形態 2 におけるプログラム自動変換装置の動作について説明するフローチャート

【図 12】実施の形態 2 における命令パターン情報管理表を示す図

【図 13】実施の形態 2 における変換後の新プログラムの例を示す図

【図 14】実施の形態 2 における新プログラムの例を示す図

【符号の説明】

【0061】

101 命令パターン情報格納部

102 プログラム格納部

103 解析部

104 解析結果出力部

1001 命令パターン対応情報格納部

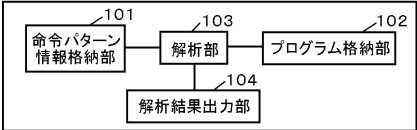
1002 条件格納部

1003 命令パターン変換部

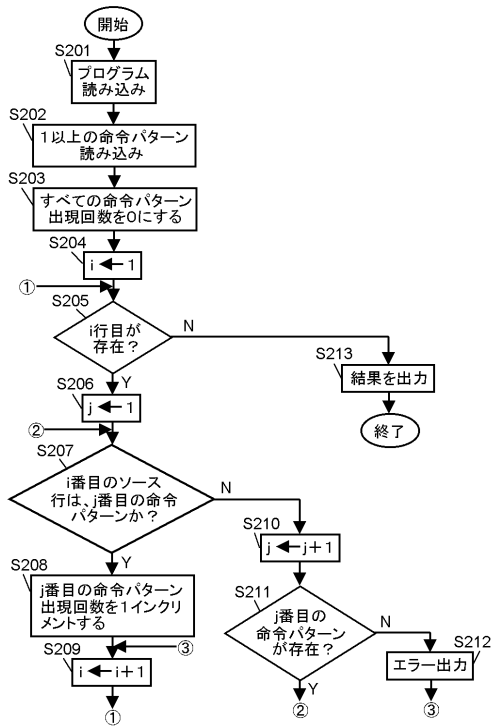
1004 新プログラム出力部

20

【図 1】



【図 2】



【図 3】

ID	命令パターン情報
1	IDENTIFICATION DIVISION
2	PROGRAM-ID 変数01
3	ENVIRONMENT DIVISION
4	CONFIGURATION SECTION
5	INPUT-OUTPUT SECTION
6	FILE-CONTROL
7	OBJECT-COMPUTER 変換01
8	SELECT 変数01 ASSIGN TO 変数02
9	SELECT 変数01 ASSIGN TO 変数02 ORGANIZATION INDEXED ACCESS MODE ... DYNAMIC RECORD KEY 変数03 FILE STATUS IS 変数04
10	SELECT 変数01 ASSIGN TO 変数02 ORGANIZATION INDEXED ACCESS MODE ... RANDOM RECORD KEY 変数03 FILE STATUS IS 変数04
11	SOURCE-COMPUTER 変数01
12	DATA DIVISION
13	FILE SECTION
14	WORKING-STORAGE SECTION
15	番号 変数01
16	番号 変数01 OCCURS 数字01 DEPENDING ON 変数02
17	番号 変数01 PIC X(数字01)
18	番号 変数01 PIC X(数字01)OCCURS 数字02 TIMES
...	...

【図 4】

ID	出現回数
1	0
2	0
3	0
4	0
...	...

【図 5】

000010*	*****
000020*	顧客名 : 電材パッケージ *
000030*	プログラム名 : 得意先売上順位表 *
000040*	プログラムID : PSD712 *
000050*	作成者名 : (LC) *
000060*	作成日 : 99.11.18 *
000070*	修正日 : 00.00.00 *
000080*	*****
000090	IDENTIFICATION DIVISION.
000100	PROGRAM-ID. PSD712.
000110*	*****
000120	ENVIRONMENT DIVISION.
000130	CONFIGURATION SECTION.
000140	SOURCE-COMPUTER. V7070.
000150	OBJECT-COMPUTER. V7070.
000160	INPUT-OUTPUT SECTION.
000170	FILE-CONTROL.
000180*	*****
000190*	ソート済得意先マスタ
000200*	*****
000210	SELECT SDK30 ASSIGN TO K3-DK00-SDK30
000220	ORGANIZATION SEQUENTIAL
000230	ACCESS SEQUENTIAL
000240	SHARE NO
000250	FILE STATUS STATUS-1 STATUS-2.
000260*	*****
000270*	コントロールファイル
000280*	*****
000290	SELECT SDC00 ASSIGN TO C0-DK00-SDC00
000300	ORGANIZATION RELATIVE
000310	ACCESS DYNAMIC
000320	RELATIVE C0-KEY
000330	SHARE LO
000340	FILE STATUS STATUS-1 STATUS-2.
	

【図 6】

ID	出現回数
1	1
2	0
3	0
4	0
⋮	⋮

【図 7】

ID	出現回数
1	2
2	2
3	2
4	2
⋮	⋮
17	42
⋮	⋮

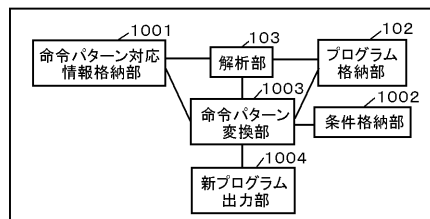
【図 8】

ID	回数	命令パターン情報
1	2	IDENTIFICATION DIVISION
2	2	PROGRAM-ID 変数01
3	2	ENVIRONMENT DIVISION
4	2	CONFIGURATION SECTION
5	2	INPUT-OUTPUT SECTION
6	2	FILE-CONTROL
7	2	OBJECT-COMPUTER 変換01
8	2	SELECT 変数01 ASSIGN TO 変数02
9	1	SELECT 変数01 ASSIGN TO 変数02 ORGANIZATION INDEXED ACCESS MODE ... DYNAMIC RECORD KEY 変数03 FILE STATUS IS 変数04
10	1	SELECT 変数01 ASSIGN TO 変数02 ORGANIZATION INDEXED ACCESS MODE ... RANDOM RECORD KEY 変数03 FILE STATUS IS 変数04
11	2	SOURCE-COMPUTER 変数01
12	2	DATA DIVISION
13	2	FILE SECTION
14	2	WORKING-STORAGE SECTION
15	12	番号 変数01
16	5	番号 変数01 OCCURS 数字01 DEPENDING ON 変数02
17	42	番号 変数01 PIC X(数字01)
18	2	番号 変数01 PIC X(数字01)OCCURS 数字02 TIMES
⋮		⋮

【図 9】

ID	回数	命令パターン情報
1	2	IDENTIFICATION DIVISION
2	2	PROGRAM-ID 変数01
3	2	ENVIRONMENT DIVISION
4	2	CONFIGURATION SECTION
5	2	INPUT-OUTPUT SECTION
6	2	FILE-CONTROL
7	2	OBJECT-COMPUTER 変換01
8	2	SELECT 変数01 ASSIGN TO 変数02 ORGANIZATION INDEXED ACCESS MODE ... DYNAMIC RECORD KEY 変数03 FILE STATUS IS 変数04
9	1	SELECT 変数01 ASSIGN TO 変数02 ORGANIZATION INDEXED ACCESS MODE ... RANDOM RECORD KEY 変数03 FILE STATUS IS 変数04
10	1	SELECT 変数01 ASSIGN TO 変数02 ORGANIZATION INDEXED ACCESS MODE ... RANDOM RECORD KEY 変数03 FILE STATUS IS 変数04
11	2	SOURCE-COMPUTER 変数01
12	2	DATA DIVISION
13	2	FILE SECTION
14	2	WORKING-STORAGE SECTION
15	12	番号 変数01
16	5	番号 変数01 OCCURS 数字01 DEPENDING ON 変数02
17	42	番号 変数01 PIC X(数字01)
18	2	番号 変数01 PIC X(数字01)OCCURS 数字02 TIMES
.....		

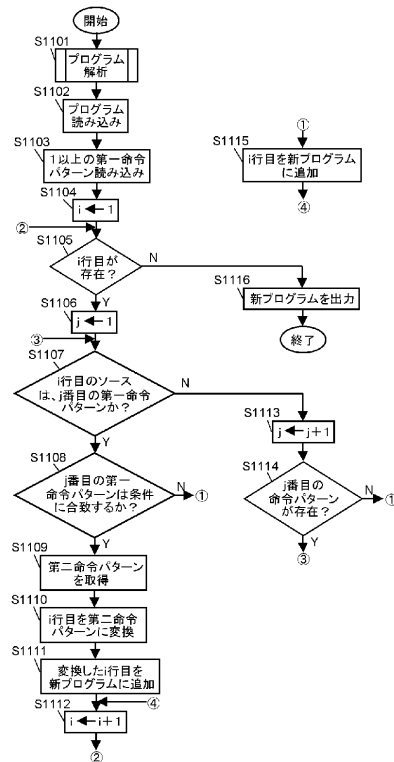
【図 10】



【図 12】

ID	第一命令パターン情報	第二命令パターン情報
1	SOURCE-COMPUTER XXXXXX	#DEL# * SOURCE-COMPUTER XXXXXX. #CHG# SOURCE-COMPUTER IBM-AS400
2	OBJECT-COMPUTER XXXXXX	#DEL# * OBJECT-COMPUTER XXXXXX. #CHG# OBJECT-COMPUTER IBM-AS400
3	SELECT filexxx ASSIGN xx-DK00-~	#DEL# *SELECT filexxx ASSIGN xx-DK00-~ #CHG# SELECT filexxx ASSIGN DATABASE-FC0x
4	SELECT filexxx ASSIGN PR-S100	#DEL# *SELECT filexxx ASSIGN PR-S100 #CHG# SELECT filexxx ASSIGN PRINTER-PR
5	SELECT filexxx ASSIGN SORTWORK	#DEL# *SELECT filexxx ASSIGN SORTWORK. #CHG# SELECT filexxx ASSIGN DATABASE-SORTWORK
6	SELECT SCREEN ASSIGN GA-DK00 ~.	#DEL# *FORMAT-CONTROL. #DEL# * SELECT SCREEN ASSIGN GA-DK00 #DEL# * ~.
7	FORMAT SECTION	#DEL# *FORMAT SECTION.
8	MD 画面名	#DEL# *MD 画面名
9	WORKING-STORAGE SECTION	WORKING-STORAGE SECTION. #ADD# COPY CONVWORK.
.....		

【図 11】



【図 13】

```
#ADD# PROCESS GRAPHIC CVTPICGRAPHIC QUOTE
#ADD# IDENTIFICATION DIVISION.
000010 *****
000020 顧客名      : 電材パッケージ          *
000030 プログラム名 : 得意先売上順位表          *
000040 プログラムID : PSD712                  *
000050 作成者名    : (LC)                    *
000060 作成日      : 99.11.18                *
000070 修正日      : 00.00.00                *
000080 *****

#DEL# *IDENTIFICATION DIVISION.
000100 PROGRAM-ID.      PSD712.
000110 *
000120 ENVIRONMENT      DIVISION.
000130 CONFIGURATION    SECTION.
#DEL# *SOURCE-COMPUTER  V7070.
#CHG# *SOURCE-COMPUTER  IBM-AS400.  (1)
#DEL# *OBJECT-COMPUTER  V7070.
#CHG# *OBJECT-COMPUTER  IBM-AS400.  (2)
#ADD# SPECIAL-NAMES.
#ADD# COPY CONVSPCN.
000160 INPUT-OUTPUT    SECTION.
000170 FILE-CONTROL.
000180 *****
000190 ソート済得意先マスタ
000200 *****

#DEL# * SELECT SDK30 ASSIGN TO K3-DK00-SDK30
#CHG# SELECT SDK30 ASSIGN TO DATABASE-FC3
000220 ORGANIZATION      SEQUENTIAL.
000230 ACCESS              RELATIVE
#DEL# * SHARE          NO
#DEL# * FILE STATUS   STATUS-1 STATUS-2.
#ADD# FILE STATUS     STATUS-2.
000260 *****
000270 コントロールファイル
000280 *****

#DEL# * SELECT SDC00 ASSIGN TO C0-DK00-SDC00
#CHG# SELECT SDC00 ASSIGN TO DATABASE-FC00
000300 ORGANIZATION      RELATIVE
000310 ACCESS              DYNAMIC
000320 RELATIVE            C0-KEY
#DEL# * SHARE          LO
#DEL# * FILE STATUS   STATUS-1 STATUS-2.
#ADD# FILE STATUS     STATUS-2.
.....
```


【図 14】

```
000810*****
000820* DATA DIVISION
000830*****
000840 DATA DIVISION.
000850 FILE SECTION.
@DEL@ * COPY C-SDK30.
@ADD@ FD SDK30.
000860 COPY SDK30.
000870/
@DEL@ * COPY C-SDC00.
@ADD@ FD SDC00.
000880 COPY SDC00.
000890/
@DEL@ * COPY C-SDC20.
@ADD@ FD SDC20.
000900 COPY SDC20.
000910/
@DEL@ * COPY C-SDC40.
@ADD@ FD SDC40.
000920 COPY SDC40.
000930/
@DEL@ * COPY C-SDCQ0.
@ADD@ FD SDCQ0.
000940 COPY SDCQ0.
000950/
@DEL@ * COPY C-SJR00.
@ADD@ FD SJR00.
000960 COPY SJR00.
000970*
@DEL@ * COPY C-PRINT.
@ADD@ FD PRINTER IS EXTERNAL.
000980 COPY PRINT.
000990*****
001000* SORT WORK
001010*****
#DEL# *SD SRTWORK1.
#CHG# SD SRTWORK1.
001030 01 SW-REC.
* 03 SW-000 PIC X(01).
001050 03 SW-001-BUMON PIC 9(04).
001060 03 SW-001-CHIKU PIC 9(02).
001070 03 SW-002-TOKUI PIC S9(07) COMP-3.
001080 03 SW-003.
001090 05 SW-003-IDX PIC 9(04).
001100 05 SW-004-URIAGE PIC S9(13) COMP-3.
.....
```

フロントページの続き

- (56)参考文献 特開平04-248624 (JP, A)
特開平02-010434 (JP, A)
特開2001-282523 (JP, A)
特開平04-191933 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/45
G06F 9/44