

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6491438号
(P6491438)

(45) 発行日 平成31年3月27日 (2019.3.27)

(24) 登録日 平成31年3月8日 (2019.3.8)

(51) Int. Cl.		F I	
G06F 8/51	(2018.01)	G06F 8/51	
G06F 8/76	(2018.01)	G06F 8/76	
G06F 17/22	(2006.01)	G06F 17/22	617

請求項の数 2 (全 13 頁)

(21) 出願番号	特願2014-174745 (P2014-174745)	(73) 特許権者	596127554
(22) 出願日	平成26年8月29日 (2014.8.29)		株式会社日立社会情報サービス
(65) 公開番号	特開2016-51235 (P2016-51235A)		東京都江東区東陽2丁目4番18号
(43) 公開日	平成28年4月11日 (2016.4.11)	(74) 代理人	110001807
審査請求日	平成29年8月18日 (2017.8.18)		特許業務法人磯野国際特許商標事務所
		(72) 発明者	坂井 孝介
			東京都江東区東陽2丁目4番18号 株式
			会社日立公共システム内
		(72) 発明者	城代 佳範
			東京都江東区東陽2丁目4番18号 株式
			会社日立公共システム内
		(72) 発明者	粟河 厚志
			東京都江東区東陽2丁目4番18号 株式
			会社日立公共システム内

最終頁に続く

(54) 【発明の名称】 マイグレーション支援装置

(57) 【特許請求の範囲】

【請求項 1】

第1のコンピュータから第2のコンピュータへのマイグレーションを支援するマイグレーション支援装置であって、

前記第1のコンピュータが有する第1の文書ファイル中の文字データに割り当てられた第1の文字コードを、記憶部が有する文字コード変換表を参照して、前記第2のコンピュータが有する第2の文書ファイル中の文字データに割り当てられた第2の文字コードに変換する文字コード変換部と、

前記第1のコンピュータが有する、前記第1の文書ファイル処理するための第1のプログラムを、前記第2のコンピュータが有する、前記第2の文書ファイル処理するための第2のプログラムに変換するプログラム変換部と、

前記第2の文字コードが割り当てられた文字データを前記第2のプログラムに読み込ませることで、前記読み込まれた文字データについて、前記第2のプログラムが指定するメモリ上のエリアの数を、前記文字データに割り当てられていた第1の文字コードを表現するバイト列のバイト数と同じに定める交換情報を生成する交換情報生成部と、

前記交換情報により定められた数からなる前記エリアに、前記読み込まれた文字データに割り当てられた1つの前記第2の文字コードを格納するエリア格納部と、を備える、

ことを特徴とするマイグレーション支援装置。

【請求項 2】

前記文字データが半角英数文字、半角記号、または半角カナ文字の文字データである場

10

20

合には、前記交換情報により定められた、前記エリアの数は1であり、

前記文字データが全角文字の文字データである場合には、前記交換情報により定められた、前記エリアの数は2である、

ことを特徴とする請求項1に記載のマイグレーション支援装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、いわゆるレガシーマイグレーション（以下、単に、「マイグレーション」と称する場合がある）の技術に関し、特に、文字コード体系の切り替えを伴うマイグレーションの技術に関する。

10

【背景技術】

【0002】

近年、これまで現行コンピュータで稼働してきた業務システム（レガシーシステム）を新規コンピュータに移行させるためのマイグレーションサービスを望む企業、自治体などが多い。マイグレーションの形態としては、例えば、汎用系のホストコンピュータ（または、オフコン）から、WINDOWS（登録商標）、UNIX（登録商標）、LINUX（登録商標）などのOS（Operating System）が稼働するオープン系のサーバコンピュータへのマイグレーション、という形態がある。なお、マイグレーションに関する技術は、数多く公開されており、例えば、特許文献1に公開されている。

【0003】

20

しかし、所定の文字コード体系（例：EBCDIC（Extended Binary Coded Decimal Interchange Kana Code）、KEIS（Kanji processing Extended Information System）、JIS8、SJIS（Shift JIS）。以下、「旧文字コード体系」と称する場合がある）でデータを取り扱っているホストコンピュータが、その文字コード体系にて標準では登録されていない外字を数多く登録していた場合（ホストコンピュータの外字エリアは9024文字分）がある。この場合、小さな外字エリアしか提供できないOS（WINDOWSが提供する外字エリアは1880文字分）が稼働するサーバコンピュータへのマイグレーションは実現できない。

【0004】

また、近年では、使用できる文字数が限られている現行コンピュータに対して、新規コンピュータでは、使用できる文字数を増やしてほしい、という要望が多くの企業、自治体などから出されている。具体的には、国際化に伴い、漢字だけでなく簡体字やハングル文字などの外国の文字も表現できるようにして欲しい、個人を正しく表記するために旧漢字も表現できるようにして欲しい、などの要望がある。

30

【0005】

そこで、これらの事情に対する対応策として、UTF（Unicode Transformation Format）-8、UTF-16など、といったより大規模な文字コード体系を、新文字コード体系として取り扱う新規コンピュータへのマイグレーションが考えられる。

【0006】

マイグレーションでは、主に、（1）業務システム上の既存のデータの移行、および、（2）そのようなデータにアクセスする、業務システム上で動作する既存のプログラムの移行、がなされる。よって、移行する既存の文字データは、新文字コード体系に対応するように文字コードを変換する必要がある。また、既存のプログラム（例えば、COBOL言語で記述されたプログラム）は、文字コードを変換した文字データを読み込むことができるように変換する必要がある。

40

【0007】

しかし、従来技術では、文字データに割り当てられた文字コードの変換と比較して、プログラムの変換は、非常に煩雑かつ困難である、という問題点があった。この問題点は、旧文字コード体系と新文字コード体系との組み合わせによっては、同じ文字であっても、その文字を表現するバイト列のバイト数が両文字コード体系間で相違すること、既存のプ

50

プログラムが文字のバイト列を格納するために指定するメモリ上のエリアの長さが固定長であること、に起因する。プログラムの変換の際は、これらの事情を考慮してプログラムの記述内容を適宜修正する必要がある（修正をしないと、文字データの溢れ、位置ずれなどが生じ、プログラムは、目的とする文字データとは異なる文字データを取得してしまう）。しかし、エリアに格納される文字のバイト列によって修正パターンが異なるため、修正は非常に煩雑かつ困難な作業となる。特許文献1の技術を含めた従来技術において、このような作業に対する改善策は何ら存在しない。

【先行技術文献】

【特許文献】

【0008】

10

【特許文献1】特許第4405571号公報

【発明の概要】

【発明が解決しようとする課題】

【0009】

そこで、本発明は、このような事情に鑑みてなされたものであり、異なる文字コード体系への切り替えが伴うマイグレーションにおいて、マイグレーションの対象となるプログラムの変換を容易にすることを目的とする。

【課題を解決するための手段】

【0010】

20

前記目的を達成するために、本発明は、

第1のコンピュータから第2のコンピュータへのマイグレーションを支援するマイグレーション支援装置であって、

前記第1のコンピュータが有する第1の文書ファイル中の文字データに割り当てられた第1の文字コードを、記憶部が有する文字コード変換表を参照して、前記第2のコンピュータが有する第2の文書ファイル中の文字データに割り当てられた第2の文字コードに変換する文字コード変換部と、

前記第1のコンピュータが有する、前記第1の文書ファイルを処理するための第1のプログラムを、前記第2のコンピュータが有する、前記第2の文書ファイルを処理するための第2のプログラムに変換するプログラム変換部と、

前記第2の文字コードが割り当てられた文字データを前記第2のプログラムに読み込ませることで、前記読み込まれた文字データについて、前記第2のプログラムが指定するメモリ上のエリアの数を、前記文字データに割り当てられていた第1の文字コードを表現するバイト列のバイト数と同じに定める交換情報を生成する交換情報生成部と、

30

前記交換情報により定められた数からなる前記エリアに、前記読み込まれた文字データに割り当てられた1つの前記第2の文字コードを格納するエリア格納部と、を備える、

ことを特徴とする。

その他の手段については後記する。

【0011】

レガシーとしての第1のプログラムは、文字データのサイズ（項目の長さ）をバイト列のバイト数として扱い、バイト数と同じ数のエリアをメモリ上に指定して文字データのバイト列を格納していた。つまり、従来のように、第1のプログラムは、メモリ上に指定するエリアを、1バイトのデータを格納するためのエリアとし、バイト数単位で文字データを処理していた。また、第1のプログラムのソースコードの記述内容はその処理に対応したものとなっていた。

40

これに対し、変換した第2のプログラムは、文字コードの変換によって、1文字を表現するバイト列のバイト数が異なった文字データを処理する際、交換情報を参照することで、第1のプログラムが使用したエリアの数と同じ数のエリアを使用することができる。つまり、第2のプログラムは、メモリ上に指定するエリアを、1文字のデータを格納するための1または複数のエリアとし、文字数単位で文字データを処理することができる。よって、第2のプログラムで組まれたロジックを第1のプログラムで組まれたロジックと同じ

50

にすることができ、第2のプログラムのソースコードの記述内容のうち、ロジックに関する部分（例えば、COBOL言語における桁数）を修正する必要はない。

したがって、異なる文字コード体系への切り替えが伴うマイグレーションにおいて、マイグレーションの対象となるプログラムの変換を容易にすることができる。

【発明の効果】

【0012】

本発明によれば、異なる文字コード体系への切り替えが伴うマイグレーションにおいて、マイグレーションの対象となるプログラムの変換を容易にすることができる。

【図面の簡単な説明】

【0013】

10

【図1】本実施形態のマイグレーション支援装置の機能構成を示す図である。

【図2】交換情報のデータ構造を示す図である。

【図3】本実施形態のマイグレーション支援装置の処理を示すフローチャートである。

【図4】比較例として、EBCDIK+KEISコードからUTF-8コードへの変換に合わせてCOBOL言語のプログラムを変換する際、ソースコードの記述内容の修正を必要とすることを説明するための図である。

【図5】本実施例として、EBCDIK+KEISコードからUTF-8コードへの変換に合わせてCOBOL言語のプログラムを変換する際、ソースコードの記述内容の修正を不要とすることを説明するための図である。

【発明を実施するための形態】

20

【0014】

図1に示すように、作業用PC1は、現行コンピュータ2から新規コンピュータ3へのマイグレーションを担当する作業員が操作するコンピュータであって、本実施形態のマイグレーション支援装置である。作業用PC1は、現行コンピュータ2から入力ファイル21および入力プログラム22を取得し、所定の変換（詳細は後記する）をした後、出力ファイル31および出力プログラム32として新規コンピュータ3に出力する。

【0015】

現行コンピュータ2（第1のコンピュータ）は、汎用系のホストコンピュータである。

新規コンピュータ3（第2のコンピュータ）は、オープン系のサーバコンピュータである。

30

【0016】

入力ファイル21（第1の文書ファイル）は、文字データを含む文書ファイルであって、現行コンピュータ2のレガシーである。入力ファイル21中の文字データは、現行コンピュータ2が取り扱っている文字コード体系に従う。現行コンピュータ2が取り扱っている文字コード体系は、半角英数文字、半角記号、および半角カナ文字の文字データについてはEBCDIKであり、全角文字の文字データについてはKEISである。本実施形態では、入力ファイル21中の文字データに割り当てられた文字コードを「EBCDIK+KEISコード」と称する場合がある。

【0017】

なお、EBCDIKは、半角英数文字、半角記号、および半角カナ文字については、1文字を1バイトで表現する（バイト数=1）。KEISは、全角文字については、1文字を2バイトで表現する（バイト数=2）。

40

【0018】

入力プログラム22（第1のプログラム）は、入力ファイル21を処理するためのプログラムであって、現行コンピュータ2のレガシーである。入力プログラム22は、COBOL言語で記述されており、その記述内容は、EBCDIK兼KEISからなる文字コード体系に即している。

【0019】

出力ファイル31（第2の文書ファイル）は、文字データを含む文書ファイルである。出力ファイル31中の文字データは、新規コンピュータ3が取り扱っている文字コード体

50

系に従う。新規コンピュータ3が取り扱っている文字コード体系は、半角英数文字、半角記号、半角カナ文字、および全角文字のいずれの文字の文字データについてもUTF-8である。本実施形態では、出力ファイル31中の文字データに割り当てられた文字コードを「UTF-8コード」と称する場合がある。

【0020】

なお、UTF-8は、半角英数文字および半角記号については、1文字を1バイトで表現し（バイト数＝1）、半角カナ文字および全角文字については、1文字を3バイトで表現する（バイト数＝3）。

【0021】

出力プログラム32（第2のプログラム）は、出力ファイル31を処理するためのプログラムである。本実施形態では、出力プログラム32は、COBOL言語で記述されているとする。しかし、周知の形式的な記述を施すことで、出力プログラム32を、JAVA（登録商標）言語で記述することができる。

10

【0022】

なお、作業用PC1は、入力部、出力部、制御部、および記憶部といったハードウェアを含む。例えば、制御部がCPU（Central Processing Unit）から構成される場合、その制御部を含むコンピュータによる情報処理は、CPUによるプログラム実行処理で実現する。また、そのコンピュータが含む記憶部は、CPUが指令し、そのコンピュータの機能を実現するためのプログラムを記憶する。これによりソフトウェアとハードウェアの協働が実現される。前記プログラムは、記録媒体に記録したり、ネットワークを経由したり

20

【0023】

図1に示すように、作業用PC1は、文字コード変換部11と、プログラム変換部12と、交換情報生成部13と、エリア格納部14といった機能部を有し、文字コード変換表Tと、交換情報Eとを記憶部に記憶している。

【0024】

文字コード変換部11は、入力ファイル21中の文字データに割り当てられたEBCDIK+KEISコード（第1の文字コード）を、文字コード変換表Tを参照して、出力ファイル31中の文字データに割り当てられたUTF-8コード（第2の文字コード）に変換する。

30

【0025】

プログラム変換部12は、文字コード変換部11による文字コードの変換に対応するように、入力プログラム22を出力プログラム32に変換する。プログラム変換部12は、出力プログラム32の記述言語を、入力プログラム22の記述言語と同じにするように変換することもできるし（例：COBOL→COBOL）、異なるように変換することもできる（例：COBOL→JAVA）。

【0026】

交換情報生成部13は、UTF-8コードが割り当てられた文字データを出力プログラム32に読み込ませることで、読み込まれた文字データについて、出力プログラム32が指定するメモリ上のエリアの数を、文字データに割り当てられていたEBCDIK+KEISコードを表現するバイト列のバイト数と同じに定める交換情報Eを生成する。

40

出力プログラム32が読み込む、UTF-8コードが割り当てられた文字データは、例えば、出力ファイル31から抽出した文字データである。

【0027】

エリア格納部14は、交換情報Eにより定められた数からなる前記エリアに、出力プログラム32に読み込まれた文字データに割り当てられた1つのUTF-8コードを格納する。

【0028】

文字コード変換表Tは、所定の文字集合（例えば、現行コンピュータ2が取り扱うEBCDIK兼KEISからなる文字コード体系にて規定されている文字の文字集合）に含

50

れる文字について、当該文字に割り当てられている、EBCDIK+KEISコードとUTF-8コードとを対応付けている。対応付けの詳細は周知であり、説明は省略する。

【0029】

交換情報生成部13が生成する交換情報Eは、UTF-8コードが割り当てられた文字データごとに、当該文字データのサイズ（項目の長さ）であるバイト数と、出力プログラム32が指定するメモリ上のエリアの数とを対応付けている。

図2に示すように、さまざまな文字データに割り当てられるUTF-8コードは、半角英数記号の文字（半角英数文字+半角記号）を表す文字コード、半角カナの文字を表す文字コード、全角文字を表す文字コードに分類することができる。分類された文字コードに対して、上記した「バイト数」および「エリアの数」が決定される。

10

【0030】

半角英数記号の文字を表す文字コードに対しては、先述の通り、UTF-8は対応する1文字を1バイトで表現するので、「バイト数」は「1」となる。また、先述の通り、EBCDIKは、半角英数文字および半角記号については、1文字を1バイトで表現するので、交換情報生成部13の機能により、「エリアの数」は「1」となる。

【0031】

半角カナの文字を表す文字コードに対しては、先述の通り、UTF-8は対応する1文字を3バイトで表現するので、「バイト数」は「3」となる。また、先述の通り、EBCDIKは、半角カナについては、1文字を1バイトで表現するので、交換情報生成部13の機能により、「エリアの数」は「1」となる。

20

【0032】

全角文字を表す文字コードに対しては、先述の通り、UTF-8は対応する1文字を3バイトで表現するので、「バイト数」は「3」となる。また、先述の通り、KEISは、全角文字については、1文字を2バイトで表現するので、交換情報生成部13の機能により、「エリアの数」は「2」となる。

【0033】

交換情報Eの内容は、現行コンピュータ2で取り扱う文字コード体系と、新規コンピュータ3で取り扱う文字コード体系との組み合わせによって決まる。

【0034】

《処理》

30

本実施形態の処理について説明する。この処理の主体は、作業用PC1の制御部であるが、説明の便宜上、「制御部」という語は省略する。

図3に示すように、作業用PC1は、現行コンピュータ2から新規コンピュータ3へのマイグレーションを行うにあたり、ステップS1から処理を開始する。

【0035】

ステップS1において、作業用PC1は、現行コンピュータ2から入力ファイル21および入力プログラム22を取得する。ステップS1の後、ステップS2に進む。

【0036】

ステップS2において、作業用PC1は、文字コード変換部11によって、取得した入力ファイル21中の文字データに対して、文字コードを、EBCDIK+KEISコードからUTF-8コードに変換し、出力ファイル31を生成する。ステップS2の後、ステップS3に進む。

40

【0037】

ステップS3において、作業用PC1は、プログラム変換部12によって、取得した入力プログラム22を出力プログラム32に変換する。ステップS3の後、ステップS4に進む。

【0038】

ステップS4において、作業用PC1は、UTF-8コードが割り当てられた文字データを出力プログラム32で読み込む。ステップS4の後、ステップS5に進む。

【0039】

50

ステップ S 5 において、作業用 P C 1 は、交換情報生成部 1 3 によって、ステップ S 4 にて読み込まれた文字データについて、交換情報 E を生成する。ステップ S 5 の後、ステップ S 6 に進む。

【0040】

ステップ S 6 において、作業用 P C 1 は、エリア格納部 1 4 によって、交換情報 E が定めた数からなるエリア（出力プログラム 3 2 が指定するメモリ上のエリア）に、対応する U T F - 8 コード、つまり、ステップ S 4 にて読み込まれた文字データに割り当てられた U T F - 8 コードを格納する。ステップ S 6 の後、図 3 の処理を終了する。

【0041】

作業用 P C 1 にて生成された出力ファイル 3 1、出力プログラム 3 2、および交換情報 E は、新規コンピュータ 3 に出力される。ここで、新規コンピュータ 3 にて、所定の業務処理を実行するために、出力プログラム 3 2 が出力ファイル 3 1 を開く場合を考える。この場合、出力プログラム 3 2 は、交換情報 E を参照して、出力プログラム 3 2 が指定するメモリ上のエリアに格納されている U T F - 8 コードに、出力プログラム 3 2 が定める順番でアクセスする。

【0042】

入力プログラム 2 2 は、入力ファイル 2 1 中の文字データのサイズ（項目の長さ）をバイト列のバイト数として扱い、バイト数と同じ数のエリアをメモリ上に指定して文字データのバイト列を格納していた。つまり、従来のように、現行コンピュータ 2 にて、入力プログラム 2 2 は、メモリ上に指定するエリアを、1 バイトのデータを格納するためのエリアとし、バイト数単位で入力ファイル 2 1 中の文字データを処理することで、実質的に文字データを 1 文字ずつ順番に処理していた。

【0043】

E B C D I K + K E I S コードから U T F - 8 コードに文字コードが変換されたことでバイト列のバイト数が変更した文字データに対して、交換情報 E は、出力プログラム 3 2 がメモリ上に指定するエリアの数を、入力プログラム 2 2 がメモリ上に指定していたエリアの数と同じにすることを可能にする。例えば、E B C D I K + K E I S コードから U T F - 8 コードに変換されると、バイト列のバイト数が「2」から「3」に変更される全角文字の文字データに対して、出力プログラム 3 2 は、交換情報 E を参照することで、メモリ上に指定するエリアの数を、従来技術のように「3」ではなく、「2」にすることができる。エリア格納部 1 4 は、（連続する）2 つ分のエリアに当該全角文字に割り当てられた 1 つの U T F - 8 コードを格納する。

【0044】

よって、出力プログラム 3 2 は、メモリ上に指定するエリアを、1 バイトのデータを格納するためのエリアではなく、1 文字のデータを格納するためのエリアとすることができ、文字数単位で出力ファイル 3 1 中の文字データを処理することができる。その結果、入力プログラム 2 2 が入力ファイル 2 1 中の文字データを 1 文字ずつ順番に処理するのと同様に、新規コンピュータ 3 にて、出力プログラム 3 2 は出力ファイル 3 1 中の文字データを 1 文字ずつ順番に処理することができる。つまり、文字データのサイズが異なる文字コードの変換を伴うマイグレーションを行ったとしても、出力プログラム 3 2 で組まれたロジックを入力ファイル 2 1 で組まれたロジックと同じままにすることができる。マイグレーションを行う作業者は、出力プログラム 3 2 のソースコードの記述内容のうち、ロジックに関する部分を修正する必要はない。

【0045】

なお、作業用 P C 1 は、U T F - 8 コードが割り当てられた文字データのバイト列を 1 バイトずつ格納する規定個数分（例えば、全角文字であれば 3 個分）のエリア（1 バイトのデータを格納するためのエリア）を、出力プログラム 3 2 がメモリ上に別途指定するように制御することができる。そして、作業用 P C 1 は、エリア格納部 1 4 が 1 つの U T F - 8 コードを格納する 1 つまたは 2 つ分のエリアと、前記規定個数分のエリアとを紐づけるように制御する。よって、新規コンピュータ 2 にて、出力プログラム 3 2 が、エリア格

10

20

30

40

50

納部 1 4 が格納した U T F - 8 コードにアクセスするとき、前記紐づけられたエリアに格納されているバイト列にアクセスすることで、対象となる文字データを処理することができる。

【0046】

≪具体例≫

図 4、図 5 を参照して、文字コード体系の切り替えを伴うマイグレーションによってプログラムを変換することの具体例を説明する。本具体例では、変換前プログラム（入力プログラム 2 2 に相当）も変換後プログラム（出力プログラム 3 2 に相当）も C O B O L 言語で記述されている。変換前プログラムが扱う文字コードは E B C D I K + K E I S コードであり、変換後プログラムが扱う文字コードは U T F - 8 コードである。

10

【0047】

図 4 には、従来技術としての比較例を示す。図 4（a）の上部には、変換前プログラムのソースコードのうちデータ部ワーキング節の記述例が示されている。集団項目 DATA - A のなかに、DATA - A1 および DATA - A2 という変数（項目）がこの順番で宣言されている。

DATA - A1 において、「PIC X」は、1 文字 1 バイトのデータ（E B C D I K）格納エリアをメモリ上に確保することを表しており、「(03)」は、このエリアが 3 つあることを表している（桁数は 3）。よって、DATA - A1 に（半角文字）3 文字分のデータを入力できる。

DATA - A2 において、「PIC N」は、1 文字 2 バイトのデータ（K E I S）格納エリアをメモリ上に確保することを表しており、「(03)」は、このエリアが 3 つあることを表している（桁数は 3）。よって、DATA - A2 に（全角文字）3 文字分のデータを入力できる。

20

なお、C O B O L 言語は、変数を固定長で宣言する。

【0048】

図 4（a）の下部には、上記記述例を具現化したエリアの模式図が示されている。1 つのエリアを 1 つのボックスで表わすと、このボックスは、1 バイトのデータ格納エリアを表している。この模式図によれば、変換前プログラムは、DATA - A1 に対して 3 バイト分のエリアをメモリ上に指定することで、DATA - A1 に 3 文字分のデータを入力できる。また、DATA - A2 に対して 6 バイト（2 バイト × 3）分のエリアをメモリ上に指定することで、DATA - A2 に 3 文字分のデータを入力できる。このように、変換前プログラムは、従来のように、文字データのバイト列が格納されるエリアを 1 バイトごとに指定しており、バイト数単位で文字データを処理する（左から順番にボックス内のバイト列に 1 つずつアクセスする）。

30

【0049】

ここで、マイグレーションにて文字コードを変換し、プログラムも変換する場合、1 文字を表現するバイト列のバイト数が異なった文字データを間違いなく処理するために（目的とした文字データを確実に読み出すために）、従来技術では、変換後プログラムのロジックを手作業で修正する必要があった。

【0050】

図 4（b）の上部には、変換後プログラムのソースコードのうちデータ部ワーキング節の記述例が示されている。プログラムの変換前後でロジックを同じにするためには、図 4（a）の記述例に対して図中の下線部で示したような記述を追加する修正が必要である。

40

前記修正として、DATA - A1 については、桁数を 3 から 9 に変更している。このように桁数を変更させる理由は、E B C D I K が半角カナ 1 文字を 1 バイトで表現するのに対し、U T F - 8 は半角カナ 1 文字を 3 バイトで表現するため、DATA - A1 に半角カナ 3 文字分のバイト列が入力された場合に対応できるように（データの溢れを防ぐように）、DATA - A1 に 9 バイト分のエリア（3 バイト × 3 文字）を持たせるためである。

また、前記修正として、DATA - A2 については、桁数を 3 から 5 に変更している。このように桁数を変更させる理由は、K E I S が全角文字 1 文字を 2 バイトで表現するのに対し、U T F - 8 は全角文字 1 文字を 3 バイトで表現するため、DATA - A2 に全角文字 3 文字のバイト列が入力された場合に対応できるように、DATA - A2 に少なくとも 9 バイト分のエリ

50

ア（3バイト×3文字）を持たせるためである。図4（b）の例では、DATA - A2の桁数を5にすることで、DATA - A2に10バイト分のエリアを持たせている。

【0051】

図4（b）の下部には、上記修正がなされた記述例を具現化したエリアの模式図が示されている。図4（b）に示すボックスは、図4（a）に示すボックス同様、1バイトのデータ格納エリアを表している。前記修正の結果、ボックスの数を増やすことで、DATA - A1に3文字分のデータを入力できること、および、DATA - A2に3文字分のデータを入力できること、という変換前プログラムの特性が変換後プログラムにおいても保持される。ただ、このようなボックスを増やすように、プログラムに組まれたロジックを修正することは、プログラム中のすべての変数に対して行う必要があるので、多大な作業量を必要とする。

10

【0052】

図5には、本実施例を示す。図5（a）は、図4（a）と同じである。つまり、変数DATA - A1には3文字分のデータを入力でき、変数DATA - A2には3文字分のデータを入力できる。

図5（b）の上部には、変換後プログラムのソースコードのうちデータ部ワーキング節の記述例が示されている。本実施例にてプログラムを変換する場合、すでに説明した交換情報Eが用いられる。

【0053】

すでに説明したように、交換情報Eによって、変換後プログラムがメモリ上に指定するエリアは、1バイトのデータを格納するためのエリアではなく、1文字のデータを格納するためのエリアとして機能する。このことは、図5（b）の下部に示すように、1つのボックスが、1つのエリアを半角英数記号カナ文字1文字のデータ格納エリアとして表すことと同義である。ここで、「半角英数記号カナ文字」という語は、半角英数文字、半角記号、および半角カナ文字をまとめた語である。半角英数記号カナ文字1文字のデータ格納エリアは、2つ並べると全角文字1文字のデータ格納エリアを表すことができる。

20

【0054】

したがって、図5（b）の記述例において、DATA - A1の「PIC X(03)」は、半角英数記号カナ文字1文字のデータ（UTF-8）格納エリアをメモリ上に3つ確保することを表すことができる。このことは、図4（b）のように桁数を増やさなくても（ロジックを修正しなくても）、変数DATA - A1には3文字分のデータ（UTF-8コードが割り当てられた文字データ）を入力できることを意味する。

30

【0055】

また、DATA - A2の「PIC N(03)」は、全角文字1文字のデータ（UTF-8）格納エリアをメモリ上に3つ確保することを表すことができる。このことは、図4（b）のように桁数を増やさなくても（ロジックを修正しなくても）、変数DATA - A2には3文字分のデータ（UTF-8コードが割り当てられた文字データ）を入力できることを意味する。

【0056】

すでに説明したように、1つまたは2つの半角英数記号カナ文字1文字のデータ格納エリアには、1つのUTF-8コードが格納される。よって、所定の業務処理の実行の際、変換後プログラムは、エリアに格納されたUTF-8コードに所定の順番でアクセスすれば、文字数単位で文字データを処理することができる。

40

【0057】

このように、交換情報Eを用いることで、変換後プログラムがメモリ上に指定するエリアの取り扱いを変えることで、プログラムに組まれたロジックを修正する、といった多大な作業量を無くすることができる。

【0058】

《まとめ》

本実施形態によれば、変換した出力プログラム32は、文字コードの変換によって、1文字を表現するバイト列のバイト数が異なった文字データを処理する際、交換情報Eを参

50

照することで、入力プログラム 3 2 が使用したエリアの数と同じ数のエリアを使用することができる。つまり、出力プログラム 3 2 は、メモリ上に指定するエリアを、1 文字のデータを格納するための 1 または複数のエリアとし、文字数単位で文字データを処理することができる。よって、出力プログラム 3 2 で組まれたロジックを入力プログラム 3 2 で組まれたロジックと同じにすることができ、出力プログラム 3 2 のソースコードの記述内容のうち、ロジックに関する部分を修正する必要はない。

したがって、異なる文字コード体系への切り替えが伴うマイグレーションにおいて、マイグレーションの対象となるプログラムの変換を容易にすることができる。

【0059】

《その他》

10

本実施形態では、EBCDIK および KEIS を用いた文字コード体系から、UTF-8 を用いた文字コード体系への切り替えが伴うマイグレーションについて説明した。しかし、JIS8 および SJIS を用いた文字コード体系から、UTF-8 を用いた文字コード体系への切り替えが伴うマイグレーションについても本発明を適用できる。

【0060】

なお、JIS8 は、半角英数文字、半角記号、および半角カナ文字については、1 文字を 1 バイトで表現する（バイト数＝1）。SJIS は、全角文字については、1 文字を 2 バイトで表現する（バイト数＝2）。

【0061】

また、本実施形態では、エリア格納部 1 4 が、出力プログラム 3 2 が指定するメモリ上のエリアに UTF-8 コードを格納していた。しかし、UTF-8 コードではなく、該当文字データを識別できる任意の形式のデータを格納することも可能である。

20

【0062】

また、本実施形態では、交換情報生成部 1 3 が交換情報 E を生成する際、出力プログラム 3 2 が読み込む、UTF-8 コードが割り当てられた文字データは、例えば、出力ファイル 3 1 から抽出した文字データとした。しかし、例えば、作業用 PC 1 が、所定の文字集合（例えば、UTF-8 を取り扱うオープン系サーバコンピュータへのマイグレーションの場合、現存するすべての文字からなる文字集合）に含まれるすべての文字について、交換情報 E を生成するために、UTF-8 コードが割り当てられた文字データを外部から事前に取得しておき、取得した文字データを出力プログラム 3 2 に読み込ませてもよい。

30

【0063】

また、本実施形態で説明した種々の技術を適宜組み合わせた技術を実現することもできる。

本実施形態で説明したソフトウェアをハードウェアとして実現することもでき、ハードウェアをソフトウェアとして実現することもできる。

その他、ハードウェア、ソフトウェア、フローチャートなどについて、本発明の趣旨を逸脱しない範囲で適宜変更が可能である。

【符号の説明】

【0064】

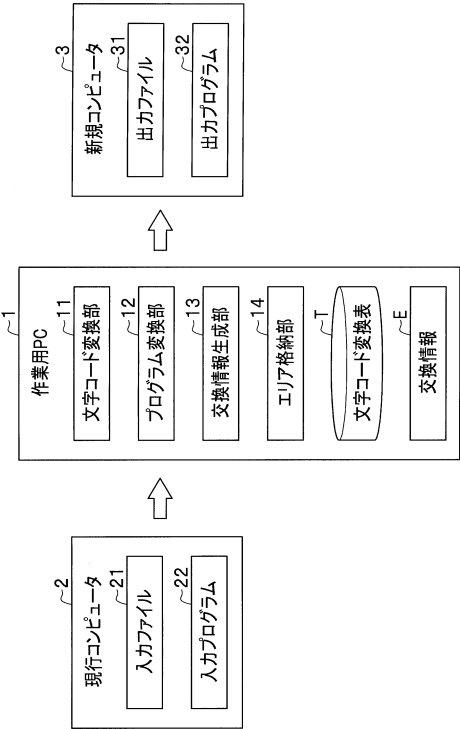
- 1 作業用 PC（マイグレーション支援装置）
- 1 1 文字コード変換部
- 1 2 プログラム変換部
- 1 3 交換情報生成部
- 1 4 エリア格納部
- 2 現行コンピュータ（第 1 のコンピュータ）
- 2 1 入力ファイル（第 1 の文書ファイル）
- 2 2 入力プログラム（第 1 のプログラム）
- 3 新規コンピュータ（第 2 のコンピュータ）
- 3 1 出力ファイル（第 2 の文書ファイル）
- 3 2 出力プログラム（第 2 のプログラム）

40

50

T 文字コード変換表
E 交換情報

【図 1】

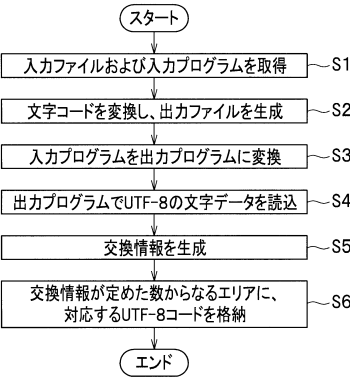


【図 2】

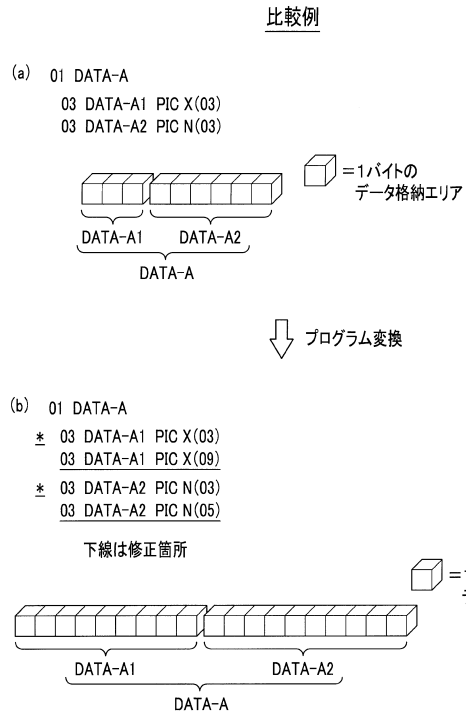
E 交換情報

UTF-8コード	バイト数	エリアの数
半角英数記号の文字を表す文字コード	1	1
半角カナの文字を表す文字コード	3	1
全角文字を表す文字コード	3	2

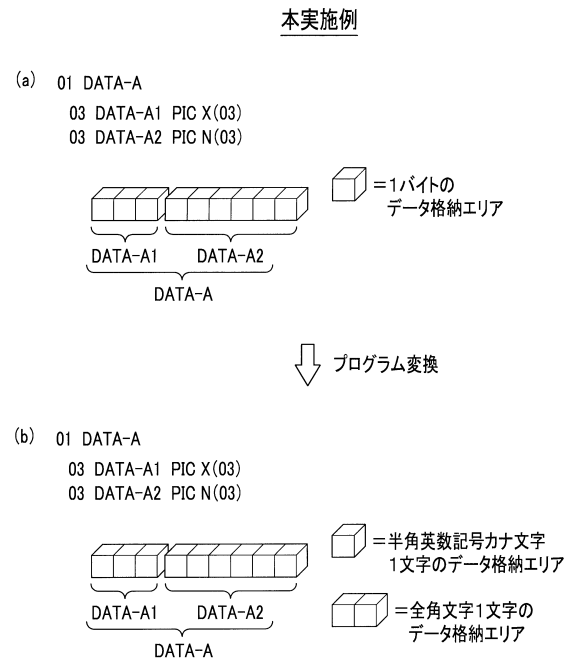
【図 3】



【図 4】



【図 5】



フロントページの続き

審査官 坂庭 剛史

- (56)参考文献 特開平06-348453 (JP, A)
特開2008-226010 (JP, A)
特開2010-224656 (JP, A)
安室浩和, APIから知るWindowsの仕組み: 第16回 Unicodeを利用して複数の言語に対応する, 日経ソフトウェア, 日本, 日経BP社, 2003年 9月24日, 第6巻, 第10号 (通巻65号), p. 160-168, ISSN 1347-4685
COBOL2002ユーザズガイド 解説・手引書, 日本, 株式会社日立製作所, 2004年 9月, 第3版 (3020-3-D42-10), p. 686-687

- (58)調査した分野(Int.Cl., DB名)
G06F 8/51
G06F 8/76
G06F 17/22