

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5304381号  
(P5304381)

(45) 発行日 平成25年10月2日(2013.10.2)

(24) 登録日 平成25年7月5日(2013.7.5)

(51) Int.Cl.

F I

G06F 9/44 (2006.01)

G06F 9/06 G20K

請求項の数 4 (全 19 頁)

<p>(21) 出願番号 特願2009-78113 (P2009-78113)                  (22) 出願日 平成21年3月27日 (2009.3.27)                  (65) 公開番号 特開2010-231479 (P2010-231479A)                  (43) 公開日 平成22年10月14日 (2010.10.14)                  審査請求日 平成23年12月5日 (2011.12.5)</p>	<p>(73) 特許権者 000005223                  富士通株式会社                  神奈川県川崎市中原区上小田中4丁目1番1号                  (74) 代理人 100074099                  弁理士 大菅 義之                  (74) 代理人 100133570                  弁理士 ▲徳▼永 民雄                  (72) 発明者 中山 大義                  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 前田 浩</p>
--	--

最終頁に続く

(54) 【発明の名称】 実行形式プログラム、およびその生成装置、生成方法

(57) 【特許請求の範囲】

【請求項1】

実行形式プログラムの生成に必要な開発資源と、前記生成された実行形式プログラムと、を含むリソース結合型実行形式プログラムにおいて、  
 情報処理装置に

実行開始指示を受けると、

前記実行開始指示が前記実行形式プログラムの実行指示である場合には、該実行形式プログラムを起動し、前記実行開始指示が前記開発資源の出力指示である場合には、前記開発資源を展開して出力するステップと、

を実行させることを特徴とするリソース結合型実行形式プログラム。

10

【請求項2】

前記実行形式プログラムは、更に前記開発資源および前記実行形式プログラムが該実行形式プログラムを構成するファイル中のいずれの位置に格納されているかを示す情報、および、前記開発資源および前記実行形式プログラムのデータサイズ量を示す情報を格納するヘッダ域を備え、

前記ヘッダ域に格納されている情報に基づいて、前記実行形式プログラムの起動や、前記開発資源の展開を行うことを特徴とする請求項1記載のリソース結合型実行形式プログラム。

【請求項3】

前記開発資源は、ソースファイル、ヘッダファイル、オブジェクトファイル、ライブラ

20

リを含むことを特徴とする請求項 1 または請求項 2 記載のリソース結合型実行形式プログラム。

【請求項 4】

実行形式プログラムの生成に必要な開発資源と、前記生成された実行形式プログラムと、を含むリソース結合型実行形式プログラムを記録した記録媒体において、

情報処理装置に、

実行開始指示を受けるステップと、

前記実行開始指示が前記実行形式プログラムの実行指示である場合には、該実行形式プログラムを起動し、前記実行開始指示が前記開発資源の出力指示である場合には、前記開発資源を展開して出力するステップと、

を実行させることを特徴とするリソース結合型実行形式プログラムを記録した記録媒体

。

【発明の詳細な説明】

【技術分野】

【0001】

コンピュータのソフトウェア開発における実行形式プログラムの生成技術に関する。

【背景技術】

【0002】

従来よりソフトウェア開発においては、ソースファイル（原始プログラム）やヘッダファイルをコンパイラでコンパイルしてオブジェクトファイルを生成し、生成されたオブジェクトファイルやライブラリをリンカでリンクして実行形式プログラムが生成されている。

【0003】

図 15 は、従来のソフトウェア開発の処理の流れを説明する図である。

ソフトウェア開発者は、ソースファイル（図 15 の「ソースファイル 1」および「ソースファイル 2」）やヘッダファイル（図 15 の「ヘッダファイル 1」および「ヘッダファイル 2」）を作成する。これをコンパイラでコンパイルし、オブジェクトファイル（図 15 の「オブジェクトファイル 1」および「オブジェクトファイル 2」）を生成する。そして、オブジェクトファイルと既に用意されているライブラリをリンカによりリンクすることによって、実行形式プログラムが出力される。

【0004】

ソフトウェア開発者はこれらの一連の作業を行い、またソースファイル、ヘッダファイル、オブジェクトファイル、ライブラリ、リンカ、実行形式プログラム、コンパイラ、などの開発資源（リソース。以下本明細書では「開発資源」と「リソース」という言葉を同意で用いる。）を保管する。一方、開発されたソフトウェアとしての実行形式プログラムはユーザ（お客様）に配布され、使用および保管される。

【0005】

このように、今日までのソフトウェア開発では、実行形式プログラムの生成に用いられたソースファイル、ヘッダファイル、ライブラリ、及びコンパイラ、リンカなどの開発資源を、ソフトウェア開発者が保管・管理しておかなければならなかった。

【0006】

一方、製品開発やシステム保守のための要員が減少し、一人のソフトウェア開発者が複数の製品開発を担当することが当たり前となっている。このような状況で、各製品の開発の際に使用したソースファイルやライブラリ、およびコンパイラやリンカなどの開発資源の管理をすることは非常に煩雑である。しかし、製品の機能追加や不具合修正に対応するためには開発資源の管理は重要である。もし管理がきちんとなされていないと、実行形式プログラムとソースファイルの対応関係が一致せず、版数の古いソースファイルを元にして機能追加してしまうリスク等が存在する。また、多くの製品開発を行うなかで、ソフトウェア開発者がソースファイル自体を紛失してしまい、実行形式プログラムの修正さえ不可能になってしまう可能性も存在する。

10

20

30

40

50

## 【先行技術文献】

## 【特許文献】

## 【0007】

【特許文献1】特開平5 - 100830号公報

【特許文献2】特開2007 - 172514号公報

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0008】

本発明の課題は実行形式プログラムを生成した際に用いた開発資源の保管・管理を容易にする技術を提供することにある。

10

## 【課題を解決するための手段】

## 【0009】

この技術の特徴は、実行形式プログラム生成の際に、ソースファイルやヘッダファイル、ライブラリ等の開発資源を含んだリソース結合型実行形式プログラムとして生成し、該リソース結合型実行形式プログラムをユーザ等に配布するところにある。そして、リソース結合型実行形式プログラムは、従来の実行形式プログラムと同様にそれ自体が実行することが可能であり、また、リソース結合型実行形式プログラムから必要に応じてソースファイルやヘッダファイル、ライブラリ等の開発資源を展開して取得することが可能である。

## 【0010】

20

これにより、実行形式プログラムを生成するために必要な開発資源は、該実行形式プログラムと共に存在し、必要となったときに該リソース結合型実行形式プログラムから展開して取得することが可能であるため、ソフトウェア開発者は開発資源の保管・管理を行う必要がなくなる。

## 【発明の効果】

## 【0011】

開示の技術は、実行形式プログラム生成の際に、開発資源を含ませたリソース結合型実行形式プログラムとして生成し、該リソース結合型実行形式プログラムを配布する。該リソース結合型実行形式プログラムは、従来の実行形式プログラムと同様にそれ自体を実行することが可能であり、また、必要に応じてソースファイル、ヘッダファイルなどの開発資源を該リソース結合型実行形式プログラムから展開して取得することが可能である。

30

## 【0012】

これにより、実行形式プログラムを生成するために必要な開発資源は、該実行形式プログラムと共に存在することになるため、ソフトウェア開発者は開発資源の保管・管理業務から解放されることになる。結果的にソフトウェア開発者の作業効率を上げることが可能である。

## 【図面の簡単な説明】

## 【0013】

【図1】実施例1の構成を示す図である。

【図2】リソース結合型実行形式プログラム作成部の動作を説明するフローチャートである。

40

【図3】リソース結合制御部の動作を説明するフローチャートである。

【図4】リソース結合型実行形式プログラムの動作を説明するフローチャートである

【図5】リソース結合型実行形式プログラムの実行の際のコマンド（パラメタ無）を示す図である。

【図6】（a）リソース結合型実行形式プログラムの実行の際のコマンド（パラメタ有）を示す図である。（b）パラメタの説明を示す図である。

【図7】リソース結合型実行形式プログラムの実行（パラメタ無）の動作を説明する図である。

【図8】リソース結合型実行形式プログラムの実行（パラメタ有）の動作を説明する図で

50

ある。

【図 9】本発明の実施例 2 の構成を示す図である。

【図 10】実施例 2 におけるリソース結合型実行形式プログラムの構成を示す図である。

【図 11A】実施例 2 におけるリソース結合制御部の動作を説明するフローチャートである。(その 1)

【図 11B】実施例 2 におけるリソース結合制御部の動作を説明するフローチャートである。(その 2)

【図 12】実施例 2 におけるリソース結合型実行形式プログラムの動作を説明するフローチャートである。

【図 13】リソース結合型実行形式プログラム作成部を含む情報処理装置のハードウェア構成を示す図である。 10

【図 14】リソース結合型実行形式プログラム作成部を実現するプログラムの情報処理装置(コンピュータ)へのローディングを説明する図である。

【図 15】従来のソフトウェア開発の処理の流れを説明する図である

【発明を実施するための形態】

【0014】

以下、図面を参照して、本発明の実施の形態について説明する。

【実施例 1】

【0015】

図 1 に、開示のリソース結合型実行形式プログラム、およびその生成装置、生成方法についての実施例 1 の構成を示す。 20

図 1 は、ソフトウェアを開発する情報処理装置上で実現される構成である。そして、図 1 は、実行形式プログラムに開発資源であるリソースを含ませる形で生成されるリソース結合型実行形式プログラム 15 がリソース結合型実行形式プログラム作成部 11 によって生成されることを示した図である。

【0016】

リソース結合型実行形式プログラム 15 は、図 1 に示すように、実行エンジン 25 が配置される実行エンジン配置域 16、実行形式プログラム 24 が配置される実行形式プログラム配置域 17、ソースファイル 20、ヘッダファイル 21、オブジェクトファイル 22、ライブラリ 22 などの開発資源が配置される開発資源配置域 18 を含むように生成される。尚、実行エンジン 25 とはリソース結合型実行形式プログラムが実行された場合に、そのプログラムの実行を制御する機能を備えるものであり、動作説明は後述する。 30

【0017】

また、リソース結合型実行形式プログラム作成部 11 は、コンパイラ 12、リンカ 13、リソース結合制御部 14 を備える。コンパイラ 12、リンカ 13 は従来のコンパイラ、リンカと同等の処理を行うものである。

【0018】

図 2 に、リソース結合型実行形式プログラム作成部 11 の動作のフローチャートを示したので、これと図 1 を参照してリソース結合型実行形式プログラム 15 が出力されるまでの処理について説明する。 40

【0019】

まず、リソース結合型実行形式プログラム作成部 11 のコンパイラ 12 が、ソフトウェア開発者が作成したソースファイル 20 やヘッダファイル 21 を入力として、オブジェクトファイル 22 を出力する(図 2 の S21)。

【0020】

次に、リソース結合型実行形式プログラム作成部 12 のリンカ 13 がオブジェクトファイル 22 とライブラリ 23 を入力として実行形式プログラム 24 を出力する(図 2 の S22)。

【0021】

そして、リソース結合型実行形式プログラム作成部 13 のリソース結合制御部 14 が、 50

ソースファイル 20、ヘッダファイル 21、オブジェクトファイル 22、ライブラリ 23、実行形式プログラム 24、実行エンジン 25 を入力としてリソース結合型実行形式プログラム 15 を出力する(図 2 の S 23)。

【0022】

更に、図 2 の S 23 の処理であるリソース結合制御部 14 の処理について、図 3 のフローチャートを参照してより詳しく説明する。

まず、リソース結合制御部 14 では、ソースファイル 20 やヘッダファイル 21、オブジェクトファイル 22、ライブラリ 23、実行形式プログラム 24、実行エンジン 25 といった入力ファイルのサイズの合計を算出する(図 3 の S 31)。

【0023】

次に、S 31 で求めた入力ファイルのサイズの合計サイズに基づいて、出力ファイルであるリソース結合型実行形式プログラムのサイズを算出する(図 3 の S 32)。

そして、S 32 で求めた出力ファイルのサイズに基づいて出力ファイルを作成し(図 3 の S 33)、出力ファイルに入力ファイルの内容を書き込んで、リソース結合型実行形式プログラムを出力する(図 3 の S 34)。

【0024】

以上、図 2 および図 3 に示したように、図 1 に示した構成により、実行形式プログラムに開発資源が結合したリソース結合型実行形式プログラムを生成することができる。

【0025】

次に、生成されたリソース結合型実行形式プログラム 15 の実行について説明する。

図 4 に、リソース結合型実行形式プログラム 15 を実行した際のフローチャートを示す。

【0026】

まず、利用者がリソース結合型実行形式プログラム 15 を実行する(図 4 の S 41)。リソース結合型実行形式プログラム 15 が実行されると、実行エンジン 25 が動作を開始する(図 4 の S 42)。そして、実行エンジン 25 はリソース結合型実行形式プログラム 15 が実行された際のコマンドにパラメタが指定されたか否かを判定する(図 4 の S 43)。

【0027】

パラメタとは、リソース結合型実行形式プログラム 15 の実行を指示するコマンドに含まれるものである。本実施例では、実行形式プログラム 24 の部分を実行する場合、パラメタ無のコマンドを情報処理装置に入力して実行を指示し、リソース結合型実行形式プログラム 15 からソースファイル 20 やヘッダファイル 21 などの一つ以上の開発資源を復元して取得する場合にパラメタを指定してコマンドを情報処理装置に入力して実行を指示することとする。

【0028】

つまり、リソース結合型実行形式プログラムの名前が“arcobj”であるとき、実行形式プログラムの部分を実行するためには、図 5 に示すようなコマンドを入力して実行する。すると、リソース結合型実行形式プログラム 15 の実行形式プログラム配置域 17 に配置されている実行形式プログラム 24 が実行される。一方、図 6 (a) に示すようなパラメタを含むコマンドを入力して実行すると、リソース結合型実行形式プログラム 15 の開発資源配置域 18 に配置されている開発資源のうち、指定された第一パラメタに対応する開発資源を第二パラメタに対応する場所に展開する。図 6 (b) に本実施例におけるパラメタの種類を示す。すなわち、以下の通り。

第一パラメタ

/A: ソースファイル、ヘッダファイル、オブジェクトファイル、ライブラリ、実行エンジン、実行形式プログラムすべてを展開する。

/S: ソースファイルを展開する。

/H: ヘッダファイルを展開する。

/O: オブジェクトファイルを展開する。

10

20

30

40

50

/L：ライブラリを展開する。

/EE：実行エンジンを展開する。

/EP：実行形式プログラムを展開する。

## 第二パラメタ

第一オペランドで指定された対象物を展開するディレクトリを指定する。

### 【0029】

図4のフローチャートの説明に戻る。

図4のS43で、実行コマンドにおいてパラメタが指定されなかった場合(N O) S44に進み、パラメタが指定された場合(Y E S) S47に進む。

S44では、実行エンジン25がリソース結合型実行形式プログラム15の実行形式プログラム24を展開する。そして、S45で実行エンジン25が展開して得られた実行形式プログラム24を起動する。実行形式プログラム24の実行がおわると、S46で実行形式プログラム24から実行エンジン25に復帰し、S48に進む。

### 【0030】

S47では、実行エンジン25が指定されたパラメタに基づいて開発資源を展開し、S48に進む。

そして、S48で実行エンジン25が動作を終了し、S49で利用者が実行したリソース結合型実行形式プログラム15の動作が終了する。

### 【0031】

以上のように図4を参照して、リソース結合型実行形式プログラムの動作について説明をしたが、パラメタ無でリソース結合型実行形式プログラムを実行する場合と、パラメタ有でリソース結合型実行形式プログラムを実行する場合について、それぞれ図7および図8を参照して、より詳しく説明する。

### 【0032】

まず、パラメタ無のコマンドでリソース結合型実行形式プログラムを実行する場合について図7を参照して説明する。

まず、図5に示したようなコマンドで、リソース結合型実行形式プログラムが起動される(図7の(1))と、最初にリソース結合型実行形式プログラムの実行エンジンが動作する(図7の(2))。そして、実行エンジンはリソース結合型実行形式プログラムから実行形式プログラムを展開(図7の(3))するとともに、実行形式プログラムを呼び出し(図7の(4))、実行形式プログラムが動作する。実行形式プログラムが終了すると実行エンジンに復帰し(図7の(5))、実行エンジンが動作を終了する(図7の(6))。そして、リソース結合型実行形式プログラムが終了する(図7の(7))。

### 【0033】

次に、パラメタ有のコマンドでリソース結合型実行形式プログラムを実行する場合について図8を参照して説明する。

まず、図6に示したようなコマンドでリソース結合型実行形式プログラムが起動される(図8の(1))と、最初にリソース結合型実行形式プログラムの実行エンジンが動作する(図8の(2))。そして、実行エンジンはコマンドで指定されたパラメタに対応してソースファイル、ヘッダファイル、オブジェクトファイルなどの開発資源をディスクに展開する(図8の(3))。開発資源の展開が終了すると実行エンジンが動作を終了し(図8の(4))、リソース結合型実行形式プログラムが終了する(図8の(5))。

### 【0034】

以上のように、リソース結合型実行形式プログラムが実行エンジンを含むことで、実行形式プログラムを実行するのか、開発資源を復元するのかを区別して実行することが可能である。

### 【実施例2】

### 【0035】

次に実施例2について説明する。

図9に実施例2の構成を示す。実施例2は実施例1と構成は同様であるが、ソースファ

10

20

30

40

50

イルに「printf(テスト);」という記述を含む構成であることを示している。

【 0 0 3 6 】

すなわち実施例 2 は、情報処理装置などの表示装置などに「テスト」と表示する実行形式プログラムを含み、実施例 1 で説明したリソース結合型実行形式プログラムのデータ構造やその動作についてより詳しく説明するものである。

【 0 0 3 7 】

まず、図 1 0 にリソース結合型実行形式プログラムのデータ構造を示す。

リソース結合型実行形式プログラムは、一つの出力ファイルとして生成される。

実施例 1 では、実行エンジン配置域、実行形式プログラム配置域、開発資源配置域を含む構成を示したが、詳細には、ファイルの先頭にヘッダ域を備え、また、実行エンジン配置域、実行形式プログラム配置域、開発資源配置域にそれぞれの域毎にヘッダ部分を備える。

10

【 0 0 3 8 】

まず、ファイル先頭のヘッダ域には、実行エン트리先頭相対位置、実行エンジン配置域先頭相対位置、実行エンジン配置域サイズ、実行形式プログラム配置域先頭相対位置、実行形式プログラム配置域サイズ、開発資源配置域先頭相対位置、開発資源配置遺棄サイズ、の情報が含まれる。

【 0 0 3 9 】

そしてヘッダ域に、実行エンジン配置域が続く。実行エンジン配置域のヘッダ部分には実行エンジン先頭相対位置、実行エンジンサイズの情報が含まれ、ヘッダ部分の次に実行エンジン本体が配置される。

20

【 0 0 4 0 】

実行エンジン配置域の次に、実行形式プログラム配置域が続く。実行形式プログラム配置域のヘッダ部分には、実行形式プログラム先頭相対位置、実行形式プログラムサイズの情報が含まれ、ヘッダ部分の次に実行形式プログラム本体が配置される。本実施例では、“printf(テスト)”を実行するバイナリコードが実行形式プログラムとして配置されている。

【 0 0 4 1 】

実行形式プログラム配置域の次に、開発資源配置域が続く。開発資源配置域のヘッダ部分には、開発資源配置域ヘッダサイズ、ソースファイル 1 先頭相対位置、ソースファイル 1 サイズ、ソースファイル 2 先頭相対位置、ソースファイル 2 サイズ、ヘッダファイル 1 先頭相対位置、ヘッダファイル 1 サイズ、ヘッダファイル 2 先頭相対位置、ヘッダファイル 2 サイズ、オブジェクトファイル 1 先頭相対位置、オブジェクトファイル 1 サイズ、オブジェクトファイル 2 先頭相対位置、オブジェクトファイル 2 サイズ、ライブラリ先頭相対位置、ライブラリサイズなど、リソース結合型実行形式プログラムに含まれる開発資源であるファイルの配置位置、該ファイルのサイズなどの情報が含まれる。そして、ヘッダ部分に引き続いて、ソースファイルやヘッダファイル、オブジェクトファイルなど、順に開発資源が配置される。

30

【 0 0 4 2 】

図 1 0 に示すように、リソース結合型実行形式プログラムは、実行エンジン、実行形式プログラム、開発資源がマージされた一つのファイルで構成されている。

40

図 1 0 に示したリソース結合型実行形式プログラムを出力する際のリソース結合制御部(図 9 の 9 4)の処理を、図 1 1 A, B に示したフローを参照して説明する。

【 0 0 4 3 】

まず、ソースファイル、ヘッダファイル、オブジェクトファイル、ライブラリ、実行形式プログラム、実行エンジンの各サイズを算出する(図 1 1 A の S 1 1 1)。

次に、S 1 1 2 で、実行エンジン配置域のサイズ(A)を求める。(A = 実行エンジン先頭相対位置格納域サイズ + 実行エンジンサイズ格納域サイズ + 実行エンジンサイズ)

次に、S 1 1 3 で、実行形式プログラム配置域のサイズ(B)を求める。(B = 実行形式プログラム先頭相対位置格納域サイズ + 実行形式プログラムサイズ格納域サイズ + 実行

50

形式プログラムサイズ)

【0044】

次に、S 1 1 4 で、開発資源配置域のサイズ ( C ) を求める。( C = 開発資源配置域ヘッダサイズ格納域 + ソースファイル 1 先頭相対位置格納域サイズ + ソースファイル 1 サイズ格納域サイズ + ソースファイル 2 先頭相対位置格納域サイズ + ソースファイル 2 サイズ格納域サイズ + オブジェクトファイル 1 先頭相対位置格納域サイズ + オブジェクトファイル 1 サイズ格納域サイズ + オブジェクトファイル 2 先頭相対位置格納域サイズ + オブジェクトファイル 2 サイズ格納域サイズ + ライブラリ先頭相対位置格納域サイズ + ライブラリサイズ格納域サイズ + ソースファイル 1 サイズ + ソースファイル 2 サイズ + ヘッダファイル 1 サイズ + ヘッダファイル 2 サイズ + オブジェクトファイル 1 サイズ + オブジェクトサイズ 2 サイズ + ライブラリサイズ)

10

【0045】

次に、S 1 1 5 でリソース結合型実行形式プログラムのサイズ ( X ) を求める。( X = ヘッダ域サイズ + 実行エンジン配置域サイズ ( A ) + 実行形式プログラム配置域サイズ ( B ) + 開発資源配置域サイズ ( C ) )

【0046】

次に、S 1 1 6 で、サイズが X の出力ファイルを作成する。そして、S 1 1 7 で作成した出力ファイルに図 1 0 に示したリソース結合型実行形式プログラムの各項目の情報を書き込む。

以上のように、図 1 0 に示したリソース結合型実行形式プログラムが生成される。

20

【0047】

次に、リソース結合型実行形式プログラムが実行された場合の動作を図 1 2 に示すフローチャートを参照して説明する。尚、本実施例におけるコマンド実行の際のパラメタについては実施例 1 と同一であるとする。

【0048】

まず、S 1 2 0 で、利用者がリソース結合型実行形式プログラムを図 5 または図 6 に示したようなコマンドにより実行する。

次に、S 1 2 1 で動作 O S がリソース結合型実行形式プログラムのヘッダ域の実行エントリ先頭相対位置に基づいて、実行エンジンに制御を渡し、実行エンジンが動作を開始する。

30

【0049】

そして、S 1 2 2 で実行エンジンが、パラメタが指定されたか否かを判断し、パラメタが指定された場合 ( Y E S ) には S 1 2 7 に、パラメタが指定されなかった場合 ( N O ) には S 1 2 3 に進む。

【0050】

パラメタが指定された場合には、S 1 2 7 でリソース結合型実行形式プログラムのヘッダ域から開発資源配置域を求め、開発資源配置域ヘッダの各種資源の先頭相対位置とサイズを参照し、指定されたパラメタに対応する資源を情報処理装置のディスク上に展開する。そして、S 1 2 8 に進む。

【0051】

パラメタが指定されなかった場合には、S 1 2 3 で、実行エンジンがリソース結合型実行形式プログラムのヘッダ域から実行形式プログラム配置域を求め、実行形式プログラム配置域の実行形式プログラム先頭相対位置とサイズを参照し、実行形式プログラムを情報処理装置のディスク上に展開する。そして、S 1 2 4 で、ディスクに展開した実行形式プログラムを実行エンジンが起動する。S 1 2 5 で、実行形式プログラムが動作し、printf 文により、情報処理装置の表示装置に「テスト」と表示される。そして、S 1 2 6 で実行形式プログラムの動作が終了し、処理が実行エンジンに復帰する。そして S 1 2 8 に進む。

40

【0052】

S 1 2 8 では、実行エンジンが動作を終了し、S 1 2 9 で利用者が実行したリソース結

50

合型実行形式プログラムが終了する。

図 1 2 に示したフローにより、実行エンジンは図 7 や図 8 に示したような実行の制御を行うことが可能である。

【 0 0 5 3 】

以上のように、実施例 1 および実施例 2 について図面を参照して詳細に説明した。上述した実施例によれば、実行形式プログラム生成の際に、ソースファイルやヘッダファイル、ライブラリ等の開発資源を含んだリソース結合型実行形式プログラムとして生成し、該リソース結合型実行形式プログラムをユーザ等に配布する。リソース結合型実行形式プログラムは、従来の実行形式プログラムと同様にそれ自体が実行することが可能である。更に、リソース結合型実行形式プログラムから必要に応じてソースファイルやヘッダファイル、ライブラリ等の開発資源を展開して取得することが可能である。

10

【 0 0 5 4 】

これにより、ユーザは従来と同様に実行形式プログラムを実行することが可能で、また必要に応じてソースファイル、ヘッダファイル等の開発資源を展開して取得することが可能である。このように実行形式プログラムの生成に必要な開発資源は、該実行形式プログラムと共に存在することになるため、ソフトウェア開発者は開発資源の保管・管理業務から解放されることになり、結果的にソフトウェア開発者の作業効率を上げることになる。また、ソフトウェア開発者が、ユーザの要求に応じて実行形式プログラムの機能をカスタマイズしても、その機能追加などを記述したソースプログラムファイル等はユーザが保管する実行形式プログラムと共に存在することとなるため、その後の保守管理作業や更なるカスタマイズ等を容易にすることが可能である。

20

【 0 0 5 5 】

以上、開示のリソース結合型実行形式プログラム、およびその生成装置、生成方法についての実施例について詳細に説明したが、上記実施例に記載したことに限定されないことは言うまでもない。上記実施例では、開発資源としてソースファイル、ヘッダファイル、ライブラリ等を対象に説明したが、開発資源はこれに限定されることはなく、ソフトウェアの開発に必要な、いずれの情報資源であってもかまわない。また、リソース結合型実行形式プログラムを実行するコマンドのパラメタとして図 6 にその種類を示したが、これに限定されることもない。このように、開示のリソース結合型実行形式プログラム、およびその生成装置、生成方法についての趣旨を逸脱しない範囲において様々な変更が可能である。

30

【 0 0 5 6 】

ところで、図 1 3 に上記実施例を実現する情報処理装置のハードウェア構成を示す。

情報処理装置は、CPU 1 3 1、メモリ 1 3 2、入力装置 1 3 3、出力装置 1 3 4、外部記録装置 1 3 5、媒体駆動装置 1 3 6、可搬記録媒体 1 3 9、ネットワーク接続装置 1 3 7 がバス 1 3 8 で接続される構成である。

【 0 0 5 7 】

メモリ 1 3 2 は例えば、ROM (Read Only Memory)、RAM (Random Access Memory) 等を含み、実行形式プログラムの実行を処理するためのデータ等の格納、またはリソース結合型実行形式プログラムを生成する際の処理等に用いられる。

40

【 0 0 5 8 】

CPU 1 3 1 は、メモリ 1 3 2 を利用して上述したフローを実行するのに用いられる。

入力装置 1 3 3 は、たとえばキーボード、ポインティングデバイス、タッチパネル等である。出力装置 1 3 4 は、たとえばディスプレイやプリンタ等である。

【 0 0 5 9 】

外部記憶装置 1 3 5 は、たとえば磁気ディスク装置、光ディスク装置、光磁気ディスク装置等である。この外部記憶装置 1 3 5 はリソース結合型実行形式プログラムの格納に用いられ、展開された開発資源が格納されたりする。また、リソース結合型実行形式プログラム作成部を実現するソフトウェアプログラムを保存しておき、必要に応じてそれら

50

をメモリ132上にロードして使用するのにも用いられる。

【0060】

媒体駆動装置136は、可搬型記録媒体139を駆動し、その記録内容にアクセスする。可搬型記録媒体139としては、メモリカード、メモリスティック、フレキシブルディスク、CD-ROM (Compact Disc Read Only Memory)、光ディスク、光磁気ディスク、DVD (Digital Versatile Disk) 等、任意のコンピュータで読み取り可能な記録媒体が用いられる。この可搬型記録媒体139にリソース結合型実行形式プログラムや、リソース結合型実行形式プログラム作成部を実現するソフトウェアプログラムを格納しておき、必要に応じてそれを外部記録装置135やメモリ132にロードして使用することもできる。

10

【0061】

ネットワーク接続装置137は、LAN, WAN等の任意のネットワーク(回線)を介して外部の装置と通信し、通信に伴うデータ変換を行う。また、必要に応じて、リソース結合型実行形式プログラムや、リソース結合型実行形式プログラム作成部を実現するソフトウェアプログラムを、ネットワークを介して外部装置から受け取り、それらを外部記憶装置135やメモリ132にロードして使用することもできる。

【0062】

また、リソース結合型実行形式プログラムの情報処理装置へのローディング方法および、リソース結合型実行形式プログラム作成部を実現するソフトウェアプログラムの情報処理装置へのローディング方法について図14に示しておく。

20

【0063】

図14(a)は、情報処理装置のハードディスクなどの外部記憶装置に格納されたプログラムやデータを情報処理装置がローディングする方法を示したものである。

図14(b)は、CD-ROMやDVDなどの可搬型の記憶媒体に記録されたプログラムやデータを情報処理装置の媒体駆動装置を介してローディングする方法を示したものである。

図14(c)は、ネットワークなどの回線を介して、情報提供者が提供するプログラムやデータを、情報処理装置の通信装置を介してローディングする方法を示したものである。

【0064】

上述した実施例は、上述したリソース結合型実行形式プログラムを生成する処理、すなわちリソース結合型実行形式プログラム作成部の処理を情報処理装置に行わせるためのプログラムとして構成してもよい。また、上述した実施例は、リソース結合型実行形式プログラム、またはリソース結合型実行形式プログラム作成部の処理を情報処理装置に行わせるためのプログラム、を記録したコンピュータ読み出し可能な記録媒体として構成することもできる。また、上述した実施例は、搬送波に具現化された上述したリソース結合型実行形式プログラム、または、リソース結合型実行形式プログラム作成部の処理を情報処理装置に行わせるためのプログラム、を表現するコンピュータ・データ・シグナルとして構成することもできる。

30

【0065】

以上のように、開示のリソース結合型実行形式プログラム、およびその生成装置、生成方法は上述したことに限定されるものではなく、本実施の形態の要旨を逸脱しない範囲内で種々の構成または形状を取ることができることは言うまでもない。

40

【0066】

以上の実施例1, 2を含む実施形態に関し、更に以下の付記を開示する。

(付記1)

情報処理装置に処理を実行させる実行形式プログラムであって、

前記実行形式プログラムは、

実行形式プログラムの生成に必要な開発資源と、

前記生成された実行形式プログラムと、

50

を含み、

実行開始指示を受けるステップと、

前記実行開始指示が前記実行形式プログラムの実行指示である場合には、該実行形式プログラムを起動し、前記実行開始指示が前記開発資源の出力指示である場合には、前記開発資源を展開して出力するステップと、

を含むことを特徴とする実行形式プログラム。

(付記 2)

前記実行形式プログラムは、更に前記開発資源および前記実行形式プログラムが該実行形式プログラムを構成するファイル中のいずれの位置に格納されているかを示す情報、および、前記開発資源および前記実行形式プログラムのデータサイズ量を示す情報を格納するヘッダ域を備え、

前記ヘッダ域に格納されている情報に基づいて、前記実行形式プログラムの起動や、前記開発資源の展開を行うことを特徴とする付記 1 記載の実行形式プログラム。

(付記 3)

前記開発資源は、ソースファイル、ヘッダファイル、オブジェクトファイル、ライブラリを含むことを特徴とする付記 1 または付記 2 記載の実行形式プログラム。

(付記 4)

情報処理装置に処理を実行させる実行形式プログラムを記録した記憶媒体であって、

該実行形式プログラムは、

実行形式プログラムの生成に必要な開発資源と、

前記生成された実行形式プログラムと、

を含み、

実行開始指示を受けるステップと、

前記実行開始指示が前記実行形式プログラムの実行指示である場合には、該実行形式プログラムを起動し、前記実行開始指示が前記開発資源の出力指示である場合には、前記開発資源を展開して出力するステップと、

を含むことを特徴とする実行形式プログラムを記録した記録媒体。

(付記 5)

前記実行形式プログラムは、更に前記開発資源および前記実行形式プログラムが該実行形式プログラムを構成するファイル中のいずれの位置に格納されているかを示す情報、および、前記開発資源および前記実行形式プログラムのデータサイズ量を示す情報を格納するヘッダ域を備え、該ヘッダ域に格納されている情報に基づいて、前記実行形式プログラムの起動や、前記開発資源の展開を行うことを特徴とする付記 4 記載の記憶媒体。

(付記 6)

前記開発資源は、ソースファイル、ヘッダファイル、オブジェクトファイル、ライブラリを含むことを特徴とする付記 4 または付記 5 記載の記憶媒体。

(付記 7)

開発資源を含む実行形式プログラムである実行形式プログラムを生成する実行形式プログラム生成装置であって、

前記開発資源から前記実行形式プログラムを生成する実行形式プログラム生成部と、

前記開発資源と、前記実行形式プログラムと、前記実行形式プログラムの実行を制御する実行エンジンとを結合して実行形式プログラムを生成するリソース結合制御部と、

を含むことを特徴とする実行形式プログラム生成装置。

(付記 8)

前記リソース結合制御部は、

前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータサイズの合計を算出して、生成する前記実行形式プログラムのサイズを算出し、該サイズに基づいて出力ファイルを作成し、該出力ファイルに前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータを書き出して該実行形式プログラムを生成する、ことを特徴とする付記 7 記載の実行形式プログラム生成装置。

10

20

30

40

50

( 付記 9 )

前記実行形式プログラム生成部は、  
ソースファイルおよびヘッダファイルをコンパイルしオブジェクトファイルを生成するコンパイラと、  
前記オブジェクトファイルとライブラリを結合し実行形式プログラムを生成するリンカと、  
を含むことを特徴とする付記 7 または付記 8 記載の実行形式プログラム生成装置。

( 付記 10 )

前記実行エンジンは、前記実行形式プログラムへの実行開始指示を受け、該実行開始指示が前記実行形式プログラムの実行指示である場合には該実行形式プログラムを起動し、  
該実行開始指示が前記開発資源の出力指示である場合には前記開発資源を展開して出力することを特徴とする付記 7 乃至付記 9 いずれか一項に記載の実行形式プログラム生成装置

10

( 付記 11 )

開発資源を含む実行形式プログラムである実行形式プログラムを生成する処理を情報処理装置に実行させるプログラムであって、

前記開発資源から前記実行形式プログラムを生成する実行形式プログラム生成ステップと、

前記開発資源と、前記実行形式プログラムと、前記実行形式プログラムの実行を制御する実行エンジンとを結合してリソース結合実行形式プログラムを生成するリソース結合ステップと、

20

を含むことを特徴とするプログラム。

( 付記 12 )

前記リソース結合ステップは、

前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータサイズの合計を算出して、生成する前記実行形式プログラムのサイズを算出するステップと、

該サイズに基づいて出力ファイルを作成し、該出力ファイルに前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータを書き出して該実行形式プログラムを生成するステップと、

を含むことを特徴とする付記 11 記載のプログラム。

30

( 付記 13 )

前記実行形式プログラム生成ステップは、

ソースファイルおよびヘッダファイルをコンパイルしオブジェクトファイルを生成するステップと、

前記オブジェクトファイルとライブラリを結合し実行形式プログラムを生成するステップと、

を含むことを特徴とする付記 11 または付記 12 記載のプログラム。

( 付記 14 )

前記実行エンジンは、前記実行形式プログラムへの実行開始指示を受け、該実行開始指示が前記実行形式プログラムの実行指示である場合には該実行形式プログラムを起動し、  
該実行開始指示が前記開発資源の出力指示である場合には前記開発資源を展開して出力することを特徴とする付記 11 乃至付記 13 いずれか一項に記載のプログラム。

40

( 付記 15 )

開発資源を含む実行形式プログラムである実行形式プログラムを生成する処理を情報処理装置に実行させるプログラムを記憶した記録媒体であって、

該プログラムは、

前記開発資源から前記実行形式プログラムを生成する実行形式プログラム生成ステップと、

前記開発資源と、前記実行形式プログラムと、前記実行形式プログラムの実行を制御する実行エンジンとを結合してリソース結合実行形式プログラムを生成するリソース結合ス

50

テップと、

を含むことを特徴とする、該プログラムを記憶した記憶媒体。

(付記 16)

前記リソース結合ステップは、

前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータサイズの合計を算出して、生成する前記実行形式プログラムのサイズを算出するステップと、

該サイズに基づいて出力ファイルを作成し、該出力ファイルに前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータを書き出して該実行形式プログラムを生成するステップと、

を含むことを特徴とする付記 15 記載の記憶媒体。

10

(付記 17)

前記実行形式プログラム生成ステップは、

ソースファイルおよびヘッダファイルをコンパイルしオブジェクトファイルを生成するステップと、

前記オブジェクトファイルとライブラリを結合し実行形式プログラムを生成するステップと、

を含むことを特徴とする付記 15 または付記 16 記載の記憶媒体。

(付記 18)

前記実行エンジンは、前記実行形式プログラムへの実行開始指示を受け、該実行開始指示が前記実行形式プログラムの実行指示である場合には該実行形式プログラムを起動し、該実行開始指示が前記開発資源の出力指示である場合には前記開発資源を展開して出力することを特徴とする付記 15 乃至付記 17 いずれか一項に記載の記憶媒体。

20

(付記 19)

開発資源を含む実行形式プログラムである実行形式プログラムを生成する方法であって、

前記開発資源から前記実行形式プログラムを生成し、

前記開発資源と、前記実行形式プログラムと、前記実行形式プログラムの実行を制御する実行エンジンとを結合して実行形式プログラムを生成する、

ことを特徴とする方法。

(付記 20)

30

前記実行形式プログラムを生成する際に、

前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータサイズの合計を算出して、生成する前記実行形式プログラムのサイズを算出し、

該サイズに基づいて出力ファイルを作成し、該出力ファイルに前記開発資源および前記実行形式プログラムおよび前記実行エンジンのデータを書き出して該実行形式プログラムを生成する、

ことを特徴とする付記 19 記載の方法。

(付記 21)

前記開発資源から前記実行形式プログラムを生成する際に、

ソースファイルおよびヘッダファイルをコンパイルしオブジェクトファイルを生成し、

前記オブジェクトファイルとライブラリを結合し実行形式プログラムを生成する、

ことを特徴とする付記 19 または付記 20 記載の方法。

40

(付記 22)

前記実行エンジンは、前記実行形式プログラムへの実行開始指示を受け、該実行開始指示が前記実行形式プログラムの実行指示である場合には該実行形式プログラムを起動し、該実行開始指示が前記開発資源の出力指示である場合には前記開発資源を展開して出力することを特徴とする付記 19 乃至付記 21 いずれか一項に記載の方法。

【符号の説明】

【0067】

11、91 リソース結合型実行形式プログラム作成部

50

- 1 2、9 2      コンパイラ
- 1 3、9 3      リンカ
- 1 4、9 4      リソース結合制御部
- 1 5、9 5      リソース結合型実行形式プログラム
- 1 6、9 6      実行エンジン配置域
- 1 7、9 7      実行形式プログラム配置遺棄
- 1 8、9 8      開発資源配置域
- 2 0      ソースファイル
- 2 1      ヘッダファイル
- 2 2      オブジェクトファイル
- 2 3      ライブラリ
- 2 4      実行形式プログラム
- 2 5      実行エンジン
- 1 3 1      C P U
- 1 3 2      メモリ
- 1 3 3      入力装置
- 1 3 4      出力装置
- 1 3 5      外部記録装置
- 1 3 6      媒体駆動装置
- 1 3 7      ネットワーク接続装置
- 1 3 8      バス
- 1 3 9      可搬型記録媒体

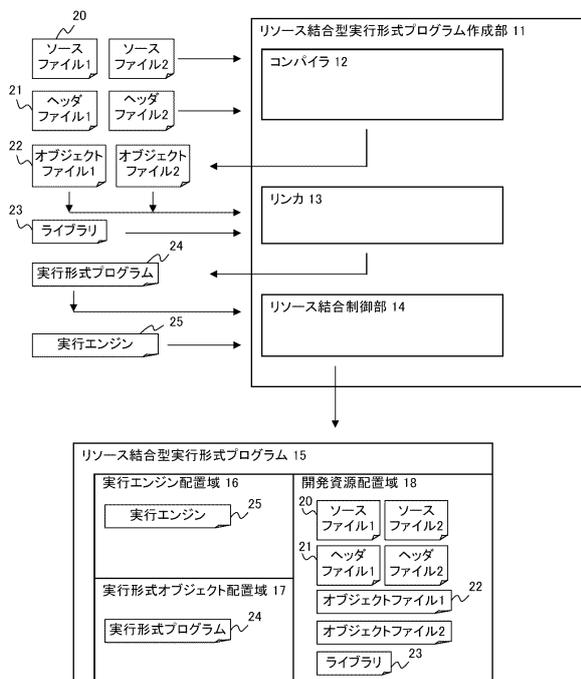
10

20

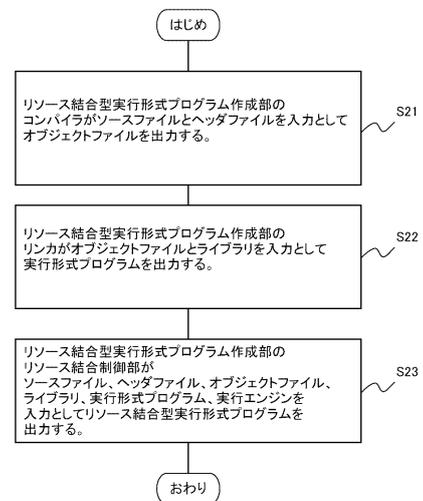
【 図 1 】

【 図 2 】

実施例1の構成を示す図

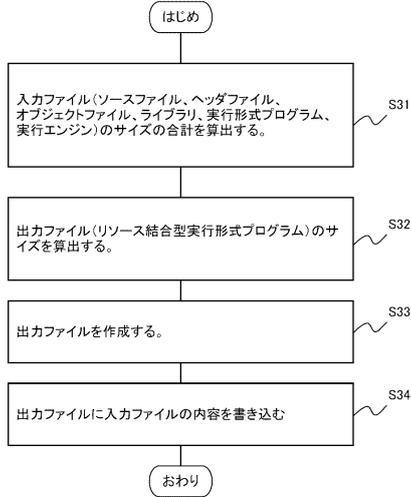


リソース結合型実行形式プログラム作成部の動作を説明するフローチャート



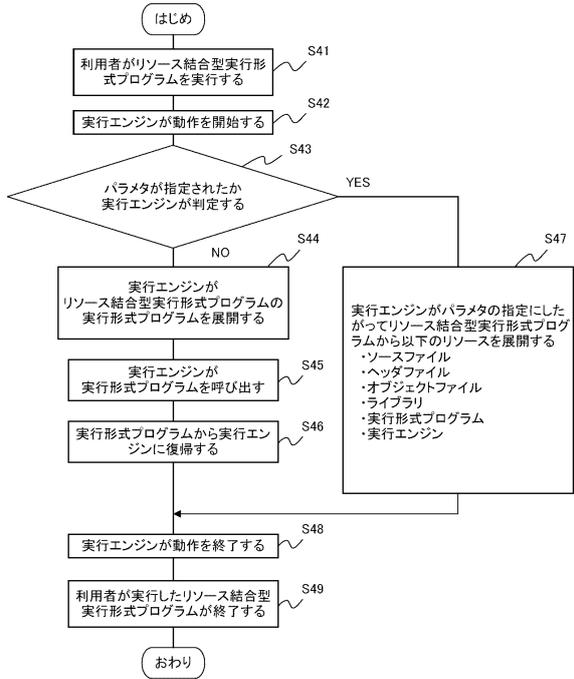
【 図 3 】

リソース結合制御部の動作を説明するフローチャート



【 図 4 】

リソース結合型実行形式プログラムの動作を説明するフローチャート



【 図 5 】

リソース結合型実行形式プログラムの実行の際のコマンド(パラメタ無)を示す図



【 図 6 】

(a)リソース結合型実行形式プログラムの実行の際のコマンド(パラメタ有)を示す図  
(b)パラメタの説明を示す図

(a)パラメタ指定ありの場合のコマンドの例



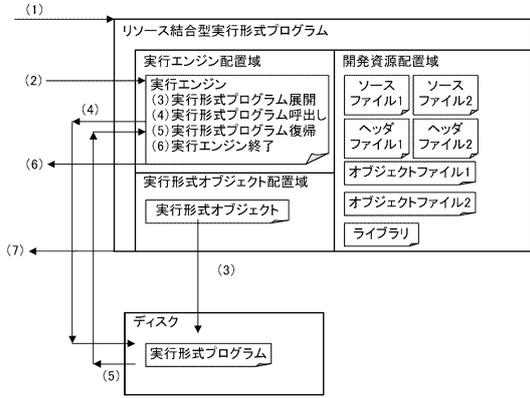
(b)パラメタの説明

第1パラメタ	
/A	ソースファイル、ヘッダファイル、オブジェクトファイル、ライブラリ、実行エンジン、実行形式プログラムすべてを展開する。
/S	ソースファイルを展開する。
/H	ヘッダファイルを展開する。
/O	オブジェクトファイルを展開する。
/L	ライブラリを展開する。
/EE	実行エンジンを展開する。
/EP	実行形式プログラムを展開する。

第2パラメタ 第1パラメタで指定された対象物を展開するディレクトリを指定する。

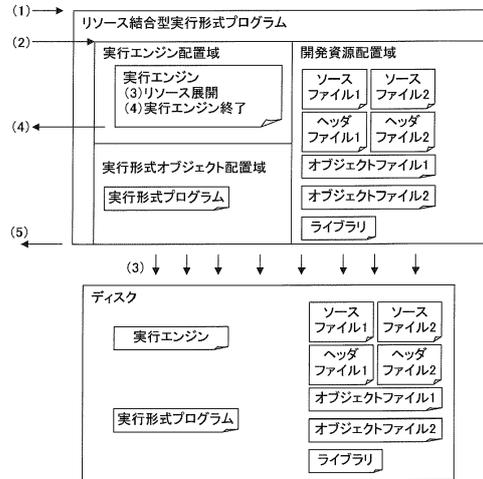
【 図 7 】

リソース結合型実行形式プログラムの実行(パラメタ無)の動作を説明する図



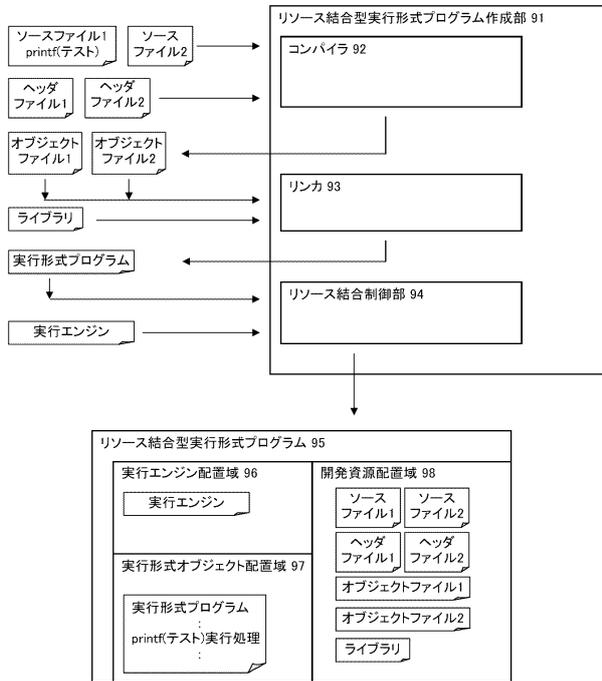
【 図 8 】

リソース結合型実行形式プログラムの実行(パラメタ有)の動作を説明する図



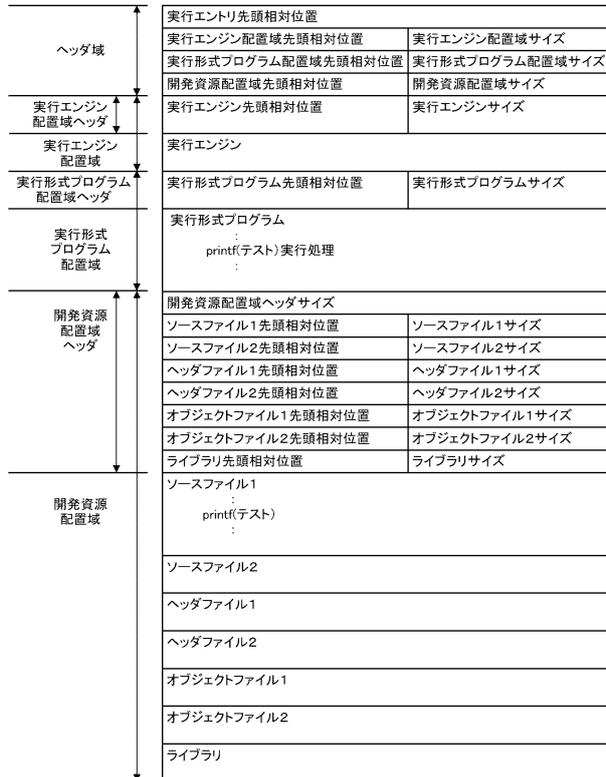
【 図 9 】

本発明の実施例2の構成を示す図



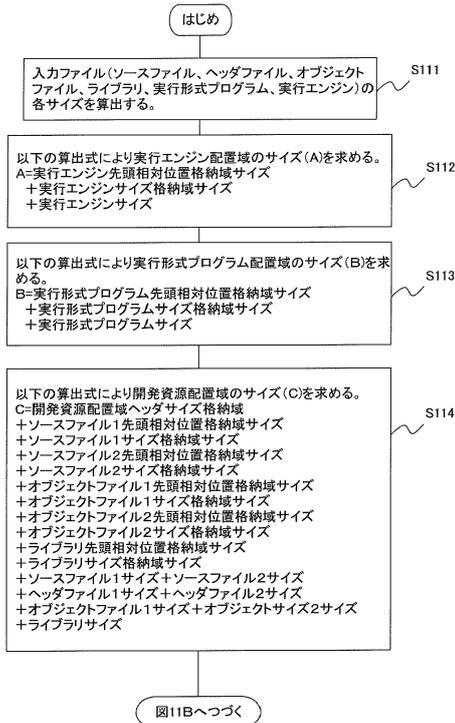
【 図 10 】

実施例2におけるリソース結合型実行形式プログラムの構成を示す図



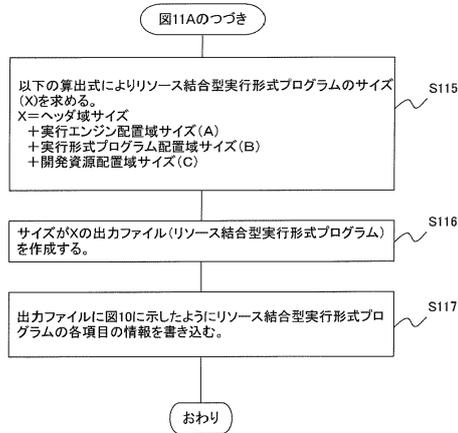
【図11A】

実施例2におけるリソース結合制御部の動作を説明するフローチャート(その1)



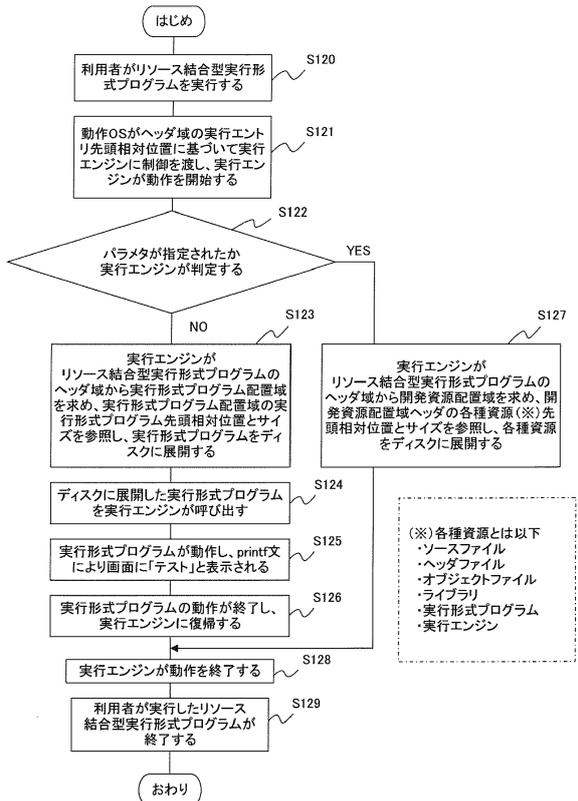
【図11B】

実施例2におけるリソース結合制御部の動作を説明するフローチャート(その2)



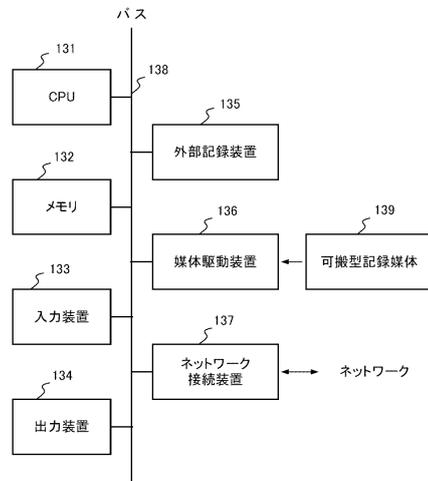
【図12】

実施例2におけるリソース結合型実行形式プログラムの動作を説明するフローチャート



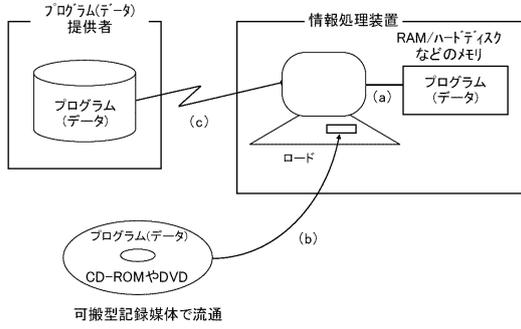
【図13】

リソース結合型実行形式プログラム作成部を含む情報処理装置のハードウェア構成を示す図



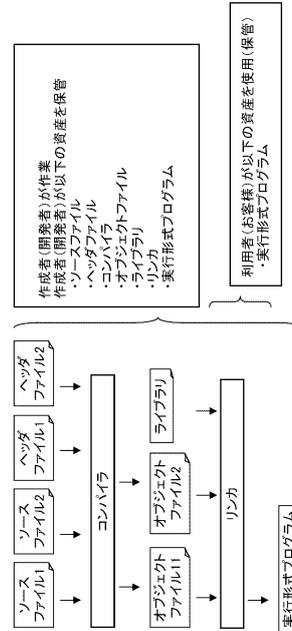
【図14】

リソース結合型実行形式プログラム作成部を実現するプログラムの情報処理装置(コンピュータ)へのローディングを説明する図



【図15】

従来のソフトウェア開発の処理の流れを説明する図



---

フロントページの続き

(56)参考文献 特開平08 - 095759 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/44