

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4875792号
(P4875792)

(45) 発行日 平成24年2月15日(2012.2.15)

(24) 登録日 平成23年12月2日(2011.12.2)

(51) Int.Cl. F I
G06F 11/28 (2006.01) G O 6 F 11/28 P
G06F 11/36 (2006.01) G O 6 F 9/06 6 2 O M

請求項の数 9 (全 32 頁)

(21) 出願番号	特願2008-296185 (P2008-296185)	(73) 特許権者	708005105
(22) 出願日	平成20年11月20日(2008.11.20)		佐治 章生
(65) 公開番号	特開2009-271906 (P2009-271906A)		神奈川県川崎市麻生区白山5丁目1番4-108号
(43) 公開日	平成21年11月19日(2009.11.19)	(72) 発明者	佐治章生
審査請求日	平成20年12月7日(2008.12.7)		神奈川県川崎市麻生区白山5-1-4108号
審査番号	不服2009-18646 (P2009-18646/J1)		台議体
審査請求日	平成21年10月1日(2009.10.1)		審査長 赤川 誠一
(31) 優先権主張番号	特願2008-297320 (P2008-297320)		審査官 清木 泰
(32) 優先日	平成20年10月25日(2008.10.25)		審査官 田中 秀人
(33) 優先権主張国	日本国(JP)		
早期審査対象出願			

最終頁に続く

(54) 【発明の名称】 一元化デバッガ及び一元化統合開発環境

(57) 【特許請求の範囲】

【請求項1】

他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にコマンド入力無しにグラフィカルにデバッグを行えることを特徴とする一元化された一元化OS上の一元化デバッガであって、

起動時に若しくは、コマンド入力無しにグラフィカルに他のOSにログオンして他のOS上のデバッガを起動するデバッガ起動手段を有し、

コマンド入力無しにグラフィカルに他のOSにデバッグ実行(プログラムのブレークポイント設定実行、ラン実行、段階的実行)命令を通知するデバッグ実行命令通知手段と、

デバッグ実行命令結果を受信するデバッグ実行命令結果受信手段とを有し、

ソースファイルを表示するソースファイル表示手段

を有し、

デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する行番号抽出手段と、

行番号抽出手段により抽出された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段と

を有することを特徴とする一元化OS上の一元化デバッガ。

10

20

【請求項2】

他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にコマンド入力無しにグラフィカルにデバッグを行えることを特徴とする一元化された一元化OS上の一元化デバッガーであって、

請求項1に記載の一元化OS上の一元化デバッガーにおいて、

デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する変数値抽出手段と、

変数値抽出手段により抽出された変数値をグラフィカルに表示する変数値グラフィカル表示手段と

を有することを特徴とする一元化OS上の一元化デバッガー。

10

【請求項3】

他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグを行えることを特徴とする一元化された一元化OS上の一元化デバッガーであって、

請求項2に記載の一元化OS上の一元化デバッガーにおいて、

コマンド入力無しにグラフィカルにソースファイルをアップロードするアップロード実行命令手段

を有し、

コマンド入力無しにグラフィカルに他のOSにコンパイル実行命令を通知するコンパイル実行命令通知手段

を有することを特徴とする一元化OS上の一元化デバッガー。

20

【請求項4】

他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグを行えることを特徴とする一元化された一元化OS上の一元化デバッガーであって、

請求項3に記載の一元化OS上の一元化デバッガーにおいて、

デバッガー起動結果を受信するデバッガー起動結果受信手段と、

デバッガー起動結果をグラフィカルに表示するデバッガー起動結果表示手段と

を有し、

受信したデバッグ実行命令結果をグラフィカルに表示するデバッグ実行命令結果表示手段

を有し、

アップロード実行命令結果を受信するアップロード実行命令結果受信手段と、

アップロード実行命令結果をグラフィカルに表示するアップロード実行命令結果表示手段と

を有し、

コンパイル実行命令結果を受信するコンパイル実行命令結果受信手段と、

受信したコンパイル実行命令結果をグラフィカルに表示するコンパイル実行命令結果表示手段と

を有することを特徴とする一元化OS上の一元化デバッガー。

30

40

【請求項5】

他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグ及び統合開発を行えることを特徴とする一元化された一元化OS上の一元化デバッガーであって、

請求項4に記載の一元化OS上の一元化デバッガーにおいて、

前記ソースファイルを表示するソースファイル表示手段を、編集可能なソースファイル

50

を表示するソースファイル表示手段とし、

ディレクトリ群及びソースファイル群をグラフィカルに一覧表示するディレクトリ/ソースファイル群一覧表示手段と、

ディレクトリ/ソースファイル群一覧表示手段により一覧表示されたディレクトリ群及びソースファイル群からディレクトリ及びソースファイルを選択するディレクトリ/ソースファイル選択手段と、

ディレクトリ/ソースファイル選択手段により選択されたソースファイル名が示すソースファイルをソースファイル表示手段に表示する選択ソースファイル表示手段と
を有し、

コマンド入力無しにグラフィカルに編集ファイルをアップロードするアップロード実行命令手段と、

アップロード実行命令結果を受信するアップロード実行命令結果受信手段と、

アップロード実行命令結果をグラフィカルに表示するアップロード実行命令結果表示手段と

を有することを特徴とする一元化OS上の一元化デバッガ―又は一元化統合開発環境。

【請求項6】

他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグ及び統合開発を行えることを特徴とする一元化された一元化OS上の一元化デバッガ―又は一元化統合開発環境であって、

請求項5に記載の一元化OS上の一元化デバッガ―において、

コマンド入力無しにグラフィカルに他のOSに他のOSコマンド実行命令を通知する他のOSコマンド実行命令通知手段と、

他のOSコマンド実行命令結果を受信する他のOSコマンド実行命令結果受信手段と、

受信した他のOSコマンド実行命令結果をグラフィカルに表示する他のOSコマンド実行命令結果表示手段と

を有し、

他のOSのディレクトリ及びファイルをグラフィカルに一覧表示する他のOSディレクトリ/ファイル一覧表示手段と、

他のOSディレクトリ/ファイル一覧表示手段により表示されたディレクトリ及びファイル一覧から、ディレクトリ及びファイルを選択する他のOSディレクトリ/ファイル選択手段と

を有し、

他のOSディレクトリ/ファイル選択手段により選択されたファイルを表示する他のOSファイル表示手段

を有することを特徴とする一元化OS上の一元化デバッガ―又は一元化統合開発環境。

【請求項7】

請求項1に記載の一元化デバッガ―において、

デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する行番号抽出手段と、

行番号抽出手段により抽出された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段と

を有する代わりに、

他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、

他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する他のOS上行番号抽出手段と、

他のOS上に設置された、他のOS上行番号抽出手段により抽出された行番号を一元化OS上

10

20

30

40

50

に通知する一元化OS向行番号通知手段と、

一元化OS上に設置された、一元化OS向行番号通知手段により通知された行番号を受信する行番号受信手段と、

一元化OS上に設置された、行番号受信手段により受信された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段と
を有することを特徴とする請求項 1 に記載の一元化デバッガ。

【請求項 8】

請求項 2 乃至請求項 6 に記載の一元化デバッガ又は請求項 5 乃至請求項 6 に記載の一元化統合開発環境において、

デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する行番号抽出手段と、

行番号抽出手段により抽出された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段と

を有し、

デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する変数値抽出手段と、

変数値抽出手段により抽出された変数値をグラフィカルに表示する変数値グラフィカル表示手段と

を有する代わりに、

他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、

他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する他のOS上行番号抽出手段と、

他のOS上に設置された、他のOS上行番号抽出手段により抽出された行番号を一元化OS上に通知する一元化OS向行番号通知手段と、

一元化OS上に設置された、一元化OS向行番号通知手段により通知された行番号を受信する行番号受信手段と、

一元化OS上に設置された、行番号受信手段により受信された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段と

を有し、

他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、

他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する他のOS上変数値抽出手段と、

他のOS上に設置された、他のOS上変数値抽出手段により抽出された変数値を一元化OS上に通知する一元化OS向変数値通知手段と、

一元化OS上に設置された、一元化OS向変数値通知手段により通知された変数値を受信する変数値受信手段と、

一元化OS上に設置された、変数値受信手段により受信された変数値をグラフィカルに表示する変数値グラフィカル表示手段と

を有することを特徴とする請求項 2 乃至請求項 6 に記載の一元化デバッガ又は請求項 5 乃至請求項 6 に記載の一元化統合開発環境。

【請求項 9】

請求項 1 乃至請求項 6 に記載の一元化デバッガ又は請求項 5 乃至請求項 6 に記載の一元化統合開発環境において、

さらに、

他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、

10

20

30

40

50

他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果を前記デバッグ実行命令結果受信手段に通知する一元化OS向デバッグ実行命令結果通知手段と

を有し、

請求項 1 乃至請求項 6 に記載の一元化デバッガ―又は請求項 5 乃至請求項 6 に記載の一元化統合開発環境のデバッグ実行命令結果受信手段は、一元化OS向デバッグ実行命令結果通知手段により通知されたデバッグ実行命令結果を受信する

ことを特徴とする請求項 1 乃至請求項 6 に記載の一元化デバッガ―又は請求項 5 乃至請求項 6 に記載の一元化統合開発環境。

【発明の詳細な説明】

10

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

【0002】

本発明は、ソフトウェア開発におけるデバッガ―及び統合開発環境に関する。

【従来の技術】

【0003】

【発明の開示】

【発明が解決しようとする課題】

20

【0004】

ここで、コマンドを操作に使用又は主に使用するOSを、本明細書中、特許請求の範囲中、図面中、要約書中においては、コマンド操作系OSと記し、またCMD系OS又はC-OSと略称することもある。また、コマンドを操作に使用又は主に使用するデバッガ―を、本明細書中、特許請求の範囲中、図面中、要約書中においては、コマンド操作系デバッガ―と記し、CMD系DBG又はC-DBGと略称することもある。また、グラフィカルな操作を使用又は主に使用するOSを、本明細書中、特許請求の範囲中、図面中、要約書中においては、グラフィカル操作系OSと記し、GRF系OS又はG-OSと略称することもある。また、ソフトウェアのグラフィカルな操作を使用又は主に使用するデバッガ―を、本明細書中、特許請求の範囲中、図面中、要約書中においては、グラフィカル操作系デバッガ―と記し、GRF系DBG又はG-

30

DBGと略称することもある。

【0005】

従来、ソフトウェア開発において、コマンド操作系OS上のソフトウェアの開発中に使用するデバッガ―は、グラフィカル操作系OS上のソフトウェアのデバッガ―に比べて使用しづらいという問題があった。

【0006】

また、コマンド操作系OSはグラフィカル操作系OSと比較して使用しづらいという問題があった。

【0007】

また、ソフトウェア開発中のOSが変わるごとにそれぞれのOSにおいてのみ使用可能なデバッガ―のみしか使用できないという問題があった。また、そのソフトウェア開発中のOSを使用しなければならないという問題があった。

40

【0008】

その為、デバッグ及び開発を一元的に行いにくいという問題があった。また、一元的にソースファイルの管理がしづらいという問題があった。

【0009】

本発明は、上記問題を解決する為に、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にソースファイルを管理でき、一元的にグラフィカルにデバッグ及び統合開発を行える一元化された一元化デバッガ―及び一元化統合開発環境を

50

提供することを目的とする。

【課題を解決するための手段】

【0010】

上記課題を解決するために、請求項1に記載の一元化デバッガーは、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にコマンド入力無しにグラフィカルにデバッグを行える（デバッグ中のプログラムの行番号をグラフィカルに表示し、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知することを含む）ことを特徴とする一元化された一元化OS上の一元化デバッガーであって、起動時に若しくは、コマンド入力無しにグラフィカルに他のOSにログオンして他のOS上のデバッガーを起動するデバッガー起動手段を有し、コマンド入力無しにグラフィカルに他のOSにデバッグ実行（プログラムのブレークポイント設定実行、ラン実行、段階的実行）命令を通知するデバッグ実行命令通知手段と、デバッグ実行命令結果を受信するデバッグ実行命令結果受信手段とを有し、ソースファイルを表示するソースファイル表示手段を有し、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する行番号抽出手段と、行番号抽出手段により抽出された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段とを有することを特徴とする一元化OS上の一元化デバッガーとした。

10

【0011】

また、上記課題を解決するために、請求項7に記載の一元化デバッガーは、請求項1に記載の一元化デバッガーにおいて、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する行番号抽出手段と、行番号抽出手段により抽出された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段とを有する代わりに、他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する他のOS上行番号抽出手段と、他のOS上に設置された、他のOS上行番号抽出手段により抽出された行番号を一元化OS上に通知する一元化OS向行番号通知手段と、一元化OS上に設置された、一元化OS向行番号通知手段により通知された行番号を受信する行番号受信手段と、一元化OS上に設置された、行番号受信手段により受信された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段とを有することを特徴とする請求項1に記載の一元化デバッガーとした。

20

30

【0012】

また、上記課題を解決するために、請求項9に記載の一元化デバッガーは、請求項1に記載の一元化デバッガーにおいて、さらに、他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果を前記デバッグ実行命令結果受信手段に通知する一元化OS向デバッグ実行命令結果通知手段とを有し、請求項1に記載の一元化デバッガーのデバッグ実行命令結果受信手段は、一元化OS向デバッグ実行命令結果通知手段により通知されたデバッグ実行命令結果を受信することを特徴とする請求項1に記載の一元化デバッガーとした。

40

【0013】

請求項1に記載の一元化デバッガーは、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、デバッガー起動手段により、起動時に若しくは、コマンド入力無しにグラフィカルに他のOSにログオンして他のOS上のデバッガーが起動され、デバッグ実行命令通知手段により、コマンド入力無しにグラフィカルに他のOSにデバッグ実行（プログラムのブレークポイント設定実行、ラン実行、段階的実行）命令が通知され、デバッグ実行命令結果受信手段により、デバッグ実行命令結果が受信され、ソースファイルを表示するソースファイル表示手段を有し、行番号抽出手段により、

50

デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号が抽出され、行番号グラフィカル表示手段により、行番号抽出手段により抽出された行番号がソースファイル表示手段上にグラフィカルに表示され、各手段がグラフィカルな手段であるので、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、デバッグ作業が容易となり、デバッグ作業の効率が向上する。また、デバッグ中のプログラムの行番号がソースファイル表示手段上にグラフィカルに表示されるので、一元化OS上の一元化デバッガーで、デバッグ中のプログラムの行番号が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

10

【 0 0 1 4 】

請求項7に記載の一元化デバッガーは、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、デバッガー起動手段により、起動時に若しくは、コマンド入力無しにグラフィカルに他のOSにログオンして他のOS上のデバッガーが起動され、デバッグ実行命令通知手段により、コマンド入力無しにグラフィカルに他のOSにデバッグ実行(プログラムのブレークポイント設定実行、ラン実行、段階的実行)命令が通知され、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により、デバッグ実行命令結果が受信され、ソースファイルを表示するソースファイル表示手段を有し、他のOS上に設置された、他のOS上行番号抽出手段により、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号が抽出され、他のOS上に設置された、一元化OS向行番号通知手段により、他のOS上行番号抽出手段により抽出された行番号が一元化OS上に通知され、一元化OS上に設置された、行番号受信手段により、一元化OS向行番号通知手段により通知された行番号が受信され、一元化OS上に設置された、行番号グラフィカル表示手段により、行番号受信手段により受信された行番号がソースファイル表示手段上にグラフィカルに表示され、各手段がグラフィカルな手段であるので、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、デバッグ作業が容易となり、デバッグ作業の効率が向上する。また、デバッグ中のプログラムの行番号がソースファイル表示手段上にグラフィカルに表示されるので、一元化OS上の一元化デバッガーで、デバッグ中のプログラムの行番号が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

20

30

【 0 0 1 5 】

請求項9に記載の一元化デバッガーは、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、デバッガー起動手段により、起動時に若しくは、コマンド入力無しにグラフィカルに他のOSにログオンして他のOS上のデバッガーが起動され、デバッグ実行命令通知手段により、コマンド入力無しにグラフィカルに他のOSにデバッグ実行(プログラムのブレークポイント設定実行、ラン実行、段階的実行)命令が通知され、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により、デバッグ実行命令結果が受信され、他のOS上に設置された、一元化OS向デバッグ実行命令結果通知手段により、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果が前記デバッグ実行命令結果受信手段に通知され、請求項1に記載の一元化デバッガーのデバッグ実行命令結果受信手段により、一元化OS向デバッグ実行命令結果通知手段により通知されたデバッグ実行命令結果が受信され、ソースファイルを表示するソースファイル表示手段を有し、行番号抽出手段により、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号が抽出され、行番号グラフィカル表示手段により、行番号抽出手段により抽出された行番号がソースファイル表示手段上にグラフィカルに表示され、各手段がグラフィカルな手段であるので、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にコマンド入力無しにグラフィカルに、視覚的

40

50

に一目でデバッグを行え、デバッグ作業が容易となり、デバッグ作業の効率が向上する。
また、デバッグ中のプログラムの行番号がソースファイル表示手段上にグラフィカルに表示されるので、一元化OS上の一元化デバッガーで、デバッグ中のプログラムの行番号が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

【 0 0 1 6 】

従って、請求項 1、請求項 7、請求項 9 に記載の一元化デバッガーによれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、デバッグが容易となり、デバッグ作業の効率が向上する。

10

【 0 0 1 7 】

また、上記課題を解決するために、請求項 2 に記載の一元化デバッガーは、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にコマンド入力無しにグラフィカルにデバッグを行える（デバッグ中のプログラムの行番号及び変数値をグラフィカルに表示し、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知することを含む）ことを特徴とする一元化された一元化OS上の一元化デバッガーであって、請求項 1 に記載の一元化OS上の一元化デバッガーにおいて、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する変数値抽出手段と、変数値抽出手段により抽出された変数値をグラフィカルに表示する変数値グラフィカル表示手段とを有することを特徴とする一元化OS上の一元化デバッガーとした。

20

【 0 0 1 8 】

また、上記課題を解決するために、請求項 8 に記載の一元化デバッガーは、請求項 2 に記載の一元化デバッガーにおいて、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する行番号抽出手段と、行番号抽出手段により抽出された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段とを有し、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する変数値抽出手段と、変数値抽出手段により抽出された変数値をグラフィカルに表示する変数値グラフィカル表示手段とを有する代わりに、他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する他のOS上行番号抽出手段と、他のOS上に設置された、他のOS上行番号抽出手段により抽出された行番号を一元化OS上に通知する一元化OS向行番号通知手段と、一元化OS上に設置された、一元化OS向行番号通知手段により通知された行番号を受信する行番号受信手段と、一元化OS上に設置された、行番号受信手段により受信された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段とを有し、他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する他のOS上変数値抽出手段と、他のOS上に設置された、他のOS上変数値抽出手段により抽出された変数値を一元化OS上に通知する一元化OS向変数値通知手段と、一元化OS上に設置された、一元化OS向変数値通知手段により通知された変数値を受信する変数値受信手段と、一元化OS上に設置された、変数値受信手段により受信された変数値をグラフィカルに表示する変数値グラフィカル表示手段とを有することを特徴とする請求項 2 に記載の一元化デバッガーとした。

30

40

【 0 0 1 9 】

50

また、上記課題を解決するために、請求項 9 に記載の一元化デバッガーは、請求項 2 に記載の一元化デバッガーにおいて、さらに、他の OS 上に設置された、デバッグ実行命令結果を受信する他の OS 上デバッグ実行命令結果受信手段と、他の OS 上に設置された、他の OS 上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果を前記デバッグ実行命令結果受信手段に通知する一元化 OS 向デバッグ実行命令結果通知手段とを有し、請求項 1 に記載の一元化デバッガーのデバッグ実行命令結果受信手段は、一元化 OS 向デバッグ実行命令結果通知手段により通知されたデバッグ実行命令結果を受信することを特徴とする請求項 1 に記載の一元化デバッガーとした。

【 0 0 2 0 】

請求項 2 に記載の一元化デバッガーは、変数値抽出手段により、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値が抽出され、変数値グラフィカル表示手段により、変数値抽出手段により抽出された変数値がグラフィカルに表示されるので、デバッグ中のプログラムの変数値が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

10

【 0 0 2 1 】

請求項 8 に記載の一元化デバッガーは、他の OS 上に設置された、他の OS 上デバッグ実行命令結果受信手段により、デバッグ実行命令結果が受信され、他の OS 上に設置された、他の OS 上変数値抽出手段により、他の OS 上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値が抽出され、他の OS 上に設置された、一元化 OS 向変数値通知手段により、他の OS 上変数値抽出手段により抽出された変数値が一元化 OS 上に通知され、一元化 OS 上に設置された、変数値受信手段により、一元化 OS 向変数値通知手段により通知された変数値が受信され、一元化 OS 上に設置された、変数値グラフィカル表示手段により、変数値受信手段により受信された変数値がグラフィカルに表示されるので、デバッグ中のプログラムの変数値が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

20

【 0 0 2 2 】

請求項 9 に記載の一元化デバッガーは、他の OS 上に設置された、他の OS 上デバッグ実行命令結果受信手段により、デバッグ実行命令結果が受信され、他の OS 上に設置された、一元化 OS 向デバッグ実行命令結果通知手段により、他の OS 上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果が前記デバッグ実行命令結果受信手段に通知され、請求項 2 に記載の一元化デバッガーのデバッグ実行命令結果受信手段により、一元化 OS 向デバッグ実行命令結果通知手段により通知されたデバッグ実行命令結果が受信され、変数値抽出手段により、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値が抽出され、変数値グラフィカル表示手段により、変数値抽出手段により抽出された変数値がグラフィカルに表示されるので、デバッグ中のプログラムの変数値が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

30

【 0 0 2 3 】

従って、請求項 2、請求項 8、請求項 9 に記載の一元化デバッガーによれば、他の OS 上で動作し、他の OS 上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他の OS とは異なる OS である一元化 OS 上で、他の OS がいかなる OS であっても、OS の違いを吸収して、OS の違いを意識せずに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、デバッグが容易となり、デバッグ作業の効率が向上する。

40

【 0 0 2 4 】

また、上記課題を解決するために、請求項 3 に記載の一元化デバッガーは、他の OS 上で動作し、他の OS 上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他の OS とは異なる OS である一元化 OS 上で、OS の違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグを行える（デバッグ中

50

のプログラムの行番号及び変数値をグラフィカルに表示し、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知することを含む)ことを特徴とする一元化された一元化OS上の一元化デバッガーであって、請求項2に記載の一元化OS上の一元化デバッガーにおいて、コマンド入力無しにグラフィカルにソースファイルをアップロードするアップロード実行命令手段を有し、コマンド入力無しにグラフィカルに他のOSにコンパイル実行命令を通知するコンパイル実行命令通知手段を有することを特徴とする一元化OS上の一元化デバッガーとした。

【0025】

請求項3に記載の一元化デバッガーは、アップロード実行命令手段により、コマンド入力無しにグラフィカルにソースファイルがアップロードされると共に、コンパイル実行命令通知手段により、コマンド入力無しにグラフィカルに他のOSにコンパイル実行命令が通知されるので、一元化OS上の一元化デバッガーで、コンパイルが行えるようになり、一元化OS上の一元化デバッガーで、ソースファイルの一元管理が可能となり、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、さらに、一元化でき、さらに、デバッグが容易となり、デバッグ作業の効率が向上する。

【0026】

従って、請求項3に記載の一元化デバッガーによれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にソースファイルを管理でき、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え(デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができることを含む)、さらに、デバッグが容易となり、デバッグ作業の効率が向上する。

【0027】

また、上記課題を解決するために、請求項4に記載の一元化デバッガーは、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、OSの違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグを行える(デバッグ中のプログラムの行番号及び変数値をグラフィカルに表示し、各実行命令結果をグラフィカルに表示し、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知することを含む)ことを特徴とする一元化された一元化OS上の一元化デバッガーであって、請求項3に記載の一元化OS上の一元化デバッガーにおいて、デバッガー起動結果を受信するデバッガー起動結果受信手段と、デバッガー起動結果をグラフィカルに表示するデバッガー起動結果表示手段とを有し、受信したデバッグ実行命令結果をグラフィカルに表示するデバッグ実行命令結果表示手段を有し、アップロード実行命令結果を受信するアップロード実行命令結果受信手段と、アップロード実行命令結果をグラフィカルに表示するアップロード実行命令結果表示手段とを有し、コンパイル実行命令結果を受信するコンパイル実行命令結果受信手段と、受信したコンパイル実行命令結果をグラフィカルに表示するコンパイル実行命令結果表示手段とを有することを特徴とする一元化OS上の一元化デバッガーとした。

【0028】

請求項4に記載の一元化デバッガーは、デバッガー起動結果受信手段により、デバッガー起動結果が受信され、デバッガー起動結果表示手段により、デバッガー起動結果がグラフィカルに表示されると共に、デバッグ実行命令結果表示手段により、受信したデバッグ実行命令結果がグラフィカルに表示されると共に、アップロード実行命令結果受信手段により、アップロード結果が受信され、アップロード実行命令結果表示手段により、アップロード結果がグラフィカルに表示されると共に、コンパイル実行命令結果受信手段により、コンパイル実行命令結果が受信され、コンパイル実行命令結果表示手段により、受信したコンパイル実行命令結果がグラフィカルに表示されるので、一元化OS上の一元化デバッガーで、実行命令結果がグラフィカルに表示される為、一元化OS上の一元化デバッガーで

10

20

30

40

50

、さらに、一元的にソースファイルを管理でき、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、さらに、一元化でき、さらに、デバッグが容易となり、デバッグ作業の効率が向上する。

【 0 0 2 9 】

従って、請求項 4 に記載の一元化デバッガーによれば、他の OS 上で動作し、他の OS 上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他の OS とは異なる OS である一元化 OS 上で、他の OS がいかなる OS であっても、OS の違いを吸収して、OS の違いを意識せずに、さらに、一元的にソースファイルを管理でき、実行命令結果がグラフィカルに表示でき、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え（デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができることを含む）、さらに、デバッグが容易となり、デバッグ作業の効率が向上する。

10

【 0 0 3 0 】

また、上記課題を解決するために、請求項 5 に記載の一元化デバッガー又は一元化統合開発環境は、他の OS 上で動作し、他の OS 上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他の OS とは異なる OS である一元化 OS 上で、OS の違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグ及び統合開発を行える（デバッグ中のプログラムの行番号及び変数値をグラフィカルに表示し、各実行命令結果をグラフィカルに表示し、コマンド入力無しにグラフィカルに他の OS に各実行命令を通知することを含む）ことを特徴とする一元化された一元化 OS 上の一元化デバッガーであって、請求項 4 に記載の一元化 OS 上の一元化デバッガーにおいて、前記ソースファイルを表示するソースファイル表示手段を、編集可能なソースファイルを表示するソースファイル表示手段とし、ディレクトリ群及びソースファイル群をグラフィカルに一覧表示するディレクトリ/ソースファイル群一覧表示手段と、ディレクトリ/ソースファイル群一覧表示手段により一覧表示されたディレクトリ群及びソースファイル群からディレクトリ及びソースファイルを選択するディレクトリ/ソースファイル選択手段と、ディレクトリ/ソースファイル選択手段により選択されたソースファイル名が示すソースファイルをソースファイル表示手段に表示する選択ソースファイル表示手段とを有し、コマンド入力無しにグラフィカルに編集ファイルをアップロードするアップロード実行命令手段と、アップロード実行命令結果を受信するアップロード実行命令結果受信手段と、アップロード実行命令結果をグラフィカルに表示するアップロード実行命令結果表示手段とを有することを特徴とする一元化 OS 上の一元化デバッガー又は一元化統合開発環境とした。

20

30

【 0 0 3 1 】

請求項 5 に記載の一元化デバッガー又は一元化統合開発環境は、前記ソースファイルを表示するソースファイル表示手段は、編集可能なソースファイルを表示するソースファイル表示手段であって、ディレクトリ/ソースファイル群一覧表示手段により、ディレクトリ群及びソースファイル群がグラフィカルに一覧表示され、ディレクトリ/ソースファイル選択手段により、ディレクトリ/ソースファイル群一覧表示手段により一覧表示されたディレクトリ群及びソースファイル群からディレクトリ及びソースファイルが選択され、選択ソースファイル表示手段により、ディレクトリ/ソースファイル選択手段により選択されたソースファイル名が示すソースファイルがソースファイル表示手段に表示され、アップロード実行命令手段により、コマンド入力無しにグラフィカルに編集ファイルがアップロードされ、アップロード実行命令結果受信手段によりアップロード結果が受信され、アップロード実行命令結果表示手段により、アップロード結果がグラフィカルに表示されるので、一元化 OS 上の一元化デバッガー又は一元化統合開発環境で、統合開発が可能となり、一元化 OS 上の一元化デバッガー又は一元化統合開発環境で、さらに、ソースファイルを一元管理でき、一元的にコマンド入力無しにグラフィカルに、デバッグ及び統合開発を行え、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する

40

50

。

【 0 0 3 2 】

従って、請求項 5 に記載の一元化デバッガ―又は一元化統合開発環境によれば、他の OS 上で動作し、他の OS 上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他の OS とは異なる OS である一元化 OS 上で、他の OS がいかなる OS であっても、OS の違いを吸収して、OS の違いを意識せずに、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え（デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示できることを含む）、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

10

【 0 0 3 3 】

また、上記課題を解決するために、請求項 6 に記載の一元化デバッガ―又は一元化統合開発環境は、他の OS 上で動作し、他の OS 上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他の OS とは異なる OS である一元化 OS 上で、OS の違いを吸収して、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルにデバッグ及び統合開発を行える（デバッグ中のプログラムの行番号及び変数値をグラフィカルに表示し、各実行命令結果をグラフィカルに表示し、コマンド入力無しにグラフィカルに他の OS に各実行命令を通知し、コマンド入力無しにグラフィカルに他の OS に他の OS コマンド実行命令を通知し、他の OS のディレクトリ及びファイルの一覧をグラフィカルに表示することを含む）ことを特徴とする一元化された一元化 OS 上の一元化デバッガ―であって、請求項 5 に記載の一元化 OS 上の一元化デバッガ―において、コマンド入力無しにグラフィカルに他の OS に他の OS コマンド実行命令を通知する他の OS コマンド実行命令通知手段と、他の OS コマンド実行命令結果を受信する他の OS コマンド実行命令結果受信手段と、受信した他の OS コマンド実行命令結果をグラフィカルに表示する他の OS コマンド実行命令結果表示手段とを有し、他の OS のディレクトリ及びファイルをグラフィカルに一覧表示する他の OS ディレクトリ/ファイル一覧表示手段と、他の OS ディレクトリ/ファイル一覧表示手段により表示されたディレクトリ及びファイル一覧から、ディレクトリ及びファイルを選択する他の OS ディレクトリ/ファイル選択手段とを有し、他の OS ディレクトリ/ファイル選択手段により選択されたファイルを表示する他の OS ファイル表示手段を有することを特徴とする一元化 OS 上の一元化デバッガ―又は一元化統合開発環境とした。

20

30

【 0 0 3 4 】

請求項 6 に記載の一元化デバッガ―又は一元化統合開発環境は、他の OS コマンド実行命令通知手段により、コマンド入力無しにグラフィカルに他の OS に他の OS コマンド実行命令が通知され、他の OS コマンド実行命令結果受信手段により、他の OS コマンド実行命令結果が受信され、他の OS コマンド実行命令結果表示手段により、受信した他の OS コマンド実行命令結果がグラフィカルに表示されると共に、他の OS ディレクトリ/ファイル一覧表示手段により、他の OS のディレクトリ及びファイルがグラフィカルに一覧表示され、他の OS ディレクトリ/ファイル選択手段により、他の OS ディレクトリ/ファイル一覧表示手段により表示されたディレクトリ及びファイル一覧から、ディレクトリ及びファイルが選択されると共に、他の OS ファイル表示手段により、他の OS ディレクトリ/ファイル選択手段により選択されたファイルが表示されるので、他の OS コマンドの実行及び他の OS ディレクトリ/ファイルを表示できる為、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え、さらに、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

40

【 0 0 3 5 】

従って、請求項 6 に記載の一元化デバッガ―又は一元化統合開発環境によれば、他の OS 上で動作し、他の OS 上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他の OS とは異なる OS である一元化 OS 上で、他の OS がいかなる OS であっても、OS の違いを吸収して、OS の違いを意識せずに、一元的にソースファイルを管理でき、さらに、一元

50

的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え（デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示できることを含む）、さらに、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

【0036】

また、上記課題を解決するために、請求項8に記載の一元化デバッガー又は一元化統合開発環境は、請求項3乃至請求項6に記載の一元化デバッガー又は請求項5乃至請求項6に記載の一元化統合開発環境において、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する行番号抽出手段と、行番号抽出手段により抽出された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段とを有し、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する変数値抽出手段と、変数値抽出手段により抽出された変数値をグラフィカルに表示する変数値グラフィカル表示手段とを有する代わりに、他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの行番号を抽出する他のOS上行番号抽出手段と、他のOS上に設置された、他のOS上行番号抽出手段により抽出された行番号を一元化OS上に通知する一元化OS向行番号通知手段と、一元化OS上に設置された、一元化OS向行番号通知手段により通知された行番号を受信する行番号受信手段と、一元化OS上に設置された、行番号受信手段により受信された行番号をソースファイル表示手段上にグラフィカルに表示する行番号グラフィカル表示手段とを有し、他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値を抽出する他のOS上変数値抽出手段と、他のOS上に設置された、他のOS上変数値抽出手段により抽出された変数値を一元化OS上に通知する一元化OS向変数値通知手段と、一元化OS上に設置された、一元化OS向変数値通知手段により通知された変数値を受信する変数値受信手段と、一元化OS上に設置された、変数値受信手段により受信された変数値をグラフィカルに表示する変数値グラフィカル表示手段とを有することを特徴とする請求項3乃至請求項6に記載の一元化デバッガー又は請求項5乃至請求項6に記載の一元化統合開発環境とした。

【0037】

また、上記課題を解決するために、請求項9に記載の一元化デバッガー又は一元化統合開発環境は、請求項3乃至請求項6に記載の一元化デバッガー又は請求項5乃至請求項6に記載の一元化統合開発環境において、さらに、他のOS上に設置された、デバッグ実行命令結果を受信する他のOS上デバッグ実行命令結果受信手段と、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果を前記デバッグ実行命令結果受信手段に通知する一元化OS向デバッグ実行命令結果通知手段とを有し、請求項3乃至請求項6に記載の一元化デバッガー又は請求項5乃至請求項6に記載の一元化統合開発環境のデバッグ実行命令結果受信手段は、一元化OS向デバッグ実行命令結果通知手段により通知されたデバッグ実行命令結果を受信することを特徴とする請求項3乃至請求項6に記載の一元化デバッガー又は請求項5乃至請求項6に記載の一元化統合開発環境とした。

【0038】

請求項8に記載の一元化デバッガー又は一元化統合開発環境は、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により、デバッグ実行命令結果が受信され、他のOS上に設置された、他のOS上変数値抽出手段により、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値が抽出され、他のOS上に設置された、一元化OS向変数値通知手段により、他のOS上変数値抽出手段により抽出された変数値が一元化OS上に通知され、一元化OS上に設置された、変数値

10

20

30

40

50

受信手段により、一元化OS向変数値通知手段により通知された変数値が受信され、一元化OS上に設置された、変数値グラフィカル表示手段により、変数値受信手段により受信された変数値がグラフィカルに表示されるので、デバッグ中のプログラムの変数値が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

【0039】

請求項9に記載の一元化デバッガー又は一元化統合開発環境は、他のOS上に設置された、他のOS上デバッグ実行命令結果受信手段により、デバッグ実行命令結果が受信され、他のOS上に設置された、一元化OS向デバッグ実行命令結果通知手段により、他のOS上デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果が前記デバッグ実行命令結果受信手段に通知され、請求項3乃至請求項6に記載の一元化デバッガー又は請求項5乃至請求項6に記載の一元化統合開発環境のデバッグ実行命令結果受信手段により、一元化OS向デバッグ実行命令結果通知手段により通知されたデバッグ実行命令結果が受信され、変数値抽出手段により、デバッグ実行命令結果受信手段により受信されたデバッグ実行命令結果からデバッグ中のプログラムの変数値が抽出され、変数値グラフィカル表示手段により、変数値抽出手段により抽出された変数値がグラフィカルに表示されるので、デバッグ中のプログラムの変数値が視覚的に一目でわかる為、デバッグが容易となり、デバッグ作業の効率が向上する。

【0040】

従って、以上に示したように、本発明の一元化デバッガー又は一元化統合開発環境によれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にソースファイルを管理でき、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え（デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示でき、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知でき、かつ、コマンド入力無しにグラフィカルに他のOSに他のOSコマンド実行命令を通知でき、他のOSのディレクトリ及びファイルの一覧をグラフィカルに表示できることを含む）、さらに、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

【0041】

また、上記した各請求項において、他のOSをコマンド操作系OSとし、一元化OSをグラフィカル操作系OSとすると、コマンド操作系OS上で動作し、コマンド操作系OS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、コマンド操作系OSとは異なるOSであるグラフィカル操作系OS上で、OSの違いを吸収して、OSの違いを意識せずに、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え（デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示でき、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知でき、かつ、コマンド入力無しにグラフィカルに他のOSに他のOSコマンド実行命令を通知でき、他のOSのディレクトリ及びファイルの一覧をグラフィカルに表示できることを含む）、さらに、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

【発明の効果】

【0042】

請求項1、請求項7、又は請求項9に記載の一元化デバッガーによれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、デバッグ中のプログラムの行番号をソースファイル上にグラフィカ

10

20

30

40

50

ルに表示することができ、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知でき、デバッグが容易となり、デバッグ作業の効率が向上する。

【 0 0 4 3 】

また、請求項 2、請求項 8、又は請求項 9 に記載の一元化デバッガーによれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、デバッグ中のプログラムの行番号をソースファイル上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知でき、デバッグが容易となり、デバッグ作業の効率が向上する。

10

【 0 0 4 4 】

また、請求項 3、請求項 8、又は請求項 9 に記載の一元化デバッガーによれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にソースファイルを管理でき、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え（デバッグ中のプログラムの行番号をソースファイル上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知できることを含む）、さらに、デバッグが容易となり、デバッグ作業の効率が向上する。

20

【 0 0 4 5 】

また、請求項 4、請求項 8、又は請求項 9 に記載の一元化デバッガーによれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、さらに、一元的にソースファイルを管理でき、実行命令結果がグラフィカルに表示でき、さらに、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグを行え、さらに、デバッグが容易となり、デバッグ作業の効率が向上する。

【 0 0 4 6 】

また、請求項 5、請求項 8、又は請求項 9 に記載の一元化デバッガー若しくは一元化統合開発環境によれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え（デバッグ中のプログラムの行番号をソースファイル上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示でき、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知できることを含む）、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

30

40

【 0 0 4 7 】

また、請求項 6、請求項 8、又は請求項 9 に記載の一元化デバッガー若しくは一元化統合開発環境によれば、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、さらに、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え（デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示でき、かつ、コマンド

50

入力無しにグラフィカルに他のOSに各実行命令を通知でき、かつ、コマンド入力無しにグラフィカルに他のOSに他のOSコマンド実行命令を通知でき、他のOSのディレクトリ及びファイルの一覧をグラフィカルに表示できることを含む)、さらに、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

【0048】

従って、本発明により、他のOS上で動作し、他のOS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、他のOSとは異なるOSである一元化OS上で、他のOSがいかなるOSであっても、OSの違いを吸収して、OSの違いを意識せずに、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行える(デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示でき、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知でき、かつ、コマンド入力無しにグラフィカルに他のOSに他のOSコマンド実行命令を通知でき、他のOSのディレクトリ及びファイルの一覧をグラフィカルに表示できることを含む)一元化デバッガ―又は一元化統合開発環境を提供することができる。その為、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

【0049】

また、上記した本発明の各請求項において、他のOSをコマンド操作系OSとし、一元化OSをグラフィカル操作系OSとすると、コマンド操作系OS上で動作し、コマンド操作系OS上で単独でデバッグ及び開発可能なソフトウェアのデバッグ及び開発を、コマンド操作系OSとは異なるOSであるグラフィカル操作系OS上で、OSの違いを吸収して、OSの違いを意識せずに、一元的にソースファイルを管理でき、一元的にコマンド入力無しにグラフィカルに、視覚的に一目でデバッグ及び統合開発を行え(デバッグ中のプログラムの行番号をソースファイル表示手段上にグラフィカルに表示することができ、かつ、デバッグ中のプログラムの変数値をグラフィカルに表示することができ、かつ、実行命令結果がグラフィカルに表示でき、かつ、コマンド入力無しにグラフィカルに他のOSに各実行命令を通知でき、かつ、コマンド入力無しにグラフィカルに他のOSに他のOSコマンド実行命令を通知でき、他のOSのディレクトリ及びファイルの一覧をグラフィカルに表示できることを含む)、さらに、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上する。

【発明の実施の形態】

【0050】

<本発明の一元化デバッガ―の1実施の形態としての第1の実施の形態>

【0051】

本発明の一元化デバッガ―の1実施の形態としての第1の実施の形態を図1を用いて説明する。

【0052】

本発明の一元化デバッガ―の1実施の形態としての第1の実施の形態の構成について説明する。本発明の一元化デバッガ―の1実施の形態としての第1の実施の形態は、C-OS又はC-OS上のC-DBGからのステップ実行命令結果(610)をソケット通信で受信するステップ実行命令結果受信部(21)と、ステップ実行命令結果受信部(21)で受信したステップ実行命令結果(610)からステップ実行中のプログラムの行番号を抽出する行番号抽出部(31)と、ソースファイル(10)を表示するソースファイル表示部(5)と、ソースファイル(10)の内容をソースファイル表示部(5)に表示実行するソースファイル表示実行部(9)と、抽出した行番号をソースファイル(10)上にグラフィカルに表示する行番号表示部(6)と、抽出した行番号をソースファイル(10)上行番号表示部(6)をグラフィカルに表示する行番号グラフィカル表示部(41)とから構成されている。尚、ここで、ステップ実行とは、プログラムの段階的実行のことを示し、C-DBGでは、step,next,...等、プログラムを段階的に実行することを示す。また、ステップ実行命令結果受信部(21)は、後述するデバッグ実行命令受信部(20)に設けられており、デバッグ実行命令受信部(20)は、コマンド実行命令結果受信

部(2)に設けられており、ステップ実行命令結果受信部(21)(コマンド実行命令結果受信部(2))は、ステップ実行命令結果(610)(各種コマンド実行命令結果(60))をソケット通信で受信する。また、ステップ実行命令結果(610)は、後述するデバッグ実行命令(600)に含まれ、デバッグ実行命令(600)は、後述するコマンド実行命令(60)に含まれている。また、行番号抽出部(31)は、各種情報抽出部(3)に設けられており、行番号抽出部(31)(各種情報抽出部(3))は、受信したステップ実行命令結果(610)(各種コマンド実行命令結果(60))からステップ(デバッグ)実行中のプログラムの行番号(各種情報)を抽出する。また、行番号グラフィカル表示部(41)は、グラフィカル表示部(4)に設けられており、行番号グラフィカル表示部(41)(グラフィカル表示部(4))は、抽出した行番号(各種情報)をグラフィカルに表示する。

10

【0053】

次に、本発明の一元化デバッガの1実施の形態としての第1の実施の形態の動作について説明する。ステップ実行命令結果受信部(21)は、C-OS又はC-OS上のC-DBGでのステップ実行命令結果(610)をソケット通信で受信する。受信したステップ実行命令結果(610)により、行番号抽出部(31)は、ステップ実行中のプログラムの行番号を抽出する。行番号グラフィカル表示部(41)は、抽出された行番号をソースファイル(10)上にグラフィカルに表示する。例えば、一つ前に実行中の行番号から現在実行中の行番号への移行を矢印で示されるステップ実行中行表示部(6)により表示する(ステップ実行中行表示部(6))。尚、行番号表示は、矢印での表示に限ることはなく、実行中の行の色を反転表示させるなど、視覚的に一目でわかる方法であれば何でもよい。これにより、G-OS上の一元化デバッガで、現在ステップ(デバッグ)実行中のプログラムの行番号がソースファイル(10)上で視覚的に一目でわかり、デバッグが容易となり、デバッグ作業の効率が向上する。

20

【0054】

尚、上記第1の実施の形態において、図1-1に示すように、C-OS又はC-OS上のC-DBGからのステップ実行命令結果(610)を受信するステップ実行命令結果受信部(21)を含むコマンド実行命令結果受信部(2)と、ステップ実行命令結果受信部(21)で受信したステップ実行命令結果(610)からステップ実行中のプログラムの行番号を抽出する行番号抽出部(31)を含んだ各種情報抽出部(3)とをC-OS上に設置し、さらにC-OS上で抽出した行番号をG-OS上にソケット通信により通知する行番号通知部(111)を含んだ各種情報通知部(101)をC-OS上に設け、さらにG-OS上に通知された行番号をソケット通信により受信する行番号受信部(121)を含んだ各種情報受信部(102)を設けるようにしてもよい。これにより、G-OS上の実行処理負担が減少し、G-OS上でのデバッグ作業の効率が向上する。また、図1-2に示すように、C-OS又はC-OS上のC-DBGからのステップ実行命令結果(610)を受信するステップ実行命令結果受信部(21)を含んだコマンド実行命令結果受信部(2)と、ステップ実行命令結果受信部(21)で受信したステップ実行命令結果(610)からステップ実行中のプログラムの行番号を抽出する行番号抽出部(31)を含んだ各種情報通知部(3)とを削除し、C-DBG内に行番号を抽出する行番号抽出部(131)を含んだ各種抽出情報抽出部(103)と、抽出したステップ実行中のプログラムの行番号をG-OS上にソケット通信により通知する行番号通知部(111)を含んだ各種情報通知部(101)とをC-OS上に設け、G-OS上にソケット通信により通知された行番号を受信する行番号受信部(121)を含んだ各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS、C-OS上の実行処理負担が減少し、デバッグ作業の効率がさらに向上する。また、上記第1の実施の形態において、G-OS上に設置された又はG-OS上とC-OS上にまたがって設置された、行番号をコマンド(デバッグ(ステップ))実行命令結果を受信して抽出する、又は行番号を抽出して受信する手段は、上記構成に限られることはなく、受信して抽出する、又は抽出して受信する手段であればどのような構成であってもよく(例えば、C-OS上に、コマンド(デバッグ(ステップ))実行命令結果を受信するコマンド(デバッグ(ステップ))実行命令結果受信手段と、コマンド(デバッグ(ステップ))実行命令結果受信手段により受信されたコマンド(デバッグ(ステップ))実行命令結果をG-OSに通知するコマンド(デバッグ(ステップ))実行命令結果通知手段とを設け、コマンド(デバッグ(ステップ))実行命令結果通知手段により通知されたコマンド(デバッグ(ステップ))

30

40

50

)実行命令結果をコマンド(デバッグ(ステップ))実行命令結果受信部(2(20(21)))で受信するようによい。)、抽出元もC-OS又はC-DBG内部に保持されているデバッグに関する全てのデバッグ情報から抽出するような構成としてよく、さらに抽出情報であるステップ実行中のプログラムの行番号が取得できる手段であればどのような構成であってもよい。また、上記第1の実施の形態において、ソースファイル(10)を表示する手段として、ソースファイル表示部(5)とソースファイル表示実行部(9)とを設けた構成としたが、ソースファイル(10)を表示する手段であればどのような構成でもよく、また、行番号をグラフィカルに表示する手段として、行番号表示部(6)と行番号グラフィカル表示部(41)とを設けた構成としたが、行番号をグラフィカルに表示する手段であればどのような構成でもよい。また、通信手段は、ソケット通信に限ることはなく、シリアル通信としてもよく、通信ができる手段であれば何でもよい。また、C-OS上のC-DBGでのデバッグ時のステップ実行は、C-OS上でのC-DBGのコマンドによるステップ実行命令(510)であってもよいし、G-OS上の一元化デバッガからC-OS上のデバッガに通知するステップ実行命令(510)による実行でもよい。すなわち、ステップ実行命令(510)をC-OS又はC-OS上のC-DBGに通知するステップ実行命令通知部(81)をコマンド実行命令通知部(8)に設け、ステップ実行命令通知部(81)からC-OS又はC-OS上のC-DBGにステップ実行命令(510)をソケット通信により通知する。通知を受けたC-OS又はC-OS上のC-DBGは、ステップ実行命令(510)を実行する。これにより、デバッグがさらに容易となり、デバッグ作業の効率がさらに向上する。尚、ステップ実行命令(510)は、後述するデバッグ実行命令(500)に設けられており、デバッグ実行命令(500)は、コマンド実行命令(50)に含まれている。また、ステップ実行命令通知部(81)は、後述するデバッグ実行命令通知部(80)に含まれており、デバッグ実行命令通知部(80)は、コマンド実行命令通知部(8)に設けられている。

【0055】

<本発明の一元化デバッガの1実施の形態としての第2の実施の形態>

【0056】

本発明の一元化デバッガの1実施の形態としての第2の実施の形態を図1を用いて説明する。

【0057】

本発明の一元化デバッガの1実施の形態としての第2の実施の形態の構成について説明する。本発明の一元化デバッガの1実施の形態としての第2の実施の形態は、上記第1の実施の形態に以下の構成を追加した実施の形態である。すなわち、ステップ実行命令結果受信部(21)でソケット通信により受信したステップ実行命令結果(610)からステップ実行中の行の変数値又はステップ実行中のプログラムの変数値(以下においては、纏めてステップ実行中のプログラムの変数値と記す。)を抽出する変数値抽出部(32)を各種情報抽出部(3)にさらに設け、抽出された変数値をグラフィカルに表示する変数値表示部(7)をさらに設け、抽出された変数値を変数値表示部(7)にグラフィカルに表示する変数値グラフィカル表示部(42)をグラフィカル表示部(4)にさらに設けた構成とした。

【0058】

次に、本発明の一元化デバッガの1実施の形態としての第2の実施の形態の動作について説明する。ステップ実行命令結果受信部(21)は、C-OS上のC-DBGでのステップ実行命令結果(610)をソケット通信で受信する。変数値抽出部(32)は、受信したステップ実行命令結果(610)からステップ実行中のプログラムの変数値を抽出する。変数値グラフィカル表示部(42)は、抽出された変数値を変数値表示部(7)にグラフィカルに表示する。例えば、表示中のソースファイル(10)上に用意された又は表示時に用意する表示領域である変数値表示部(7)にステップ実行中のプログラムの変数値をグラフィカルに表示する。尚、変数値を表示する手段として、変数名にカーソル又はマウス(72)をあわせるとその変数値が表示されるようにしてもよい。これにより、G-OS上の一元化デバッガで、ステップ(デバッグ)実行中の変数値を視覚的に一目で理解することができ、デバッグが容易となり、デバッグ作業の効率が向上する。

【0059】

尚、上記第2の実施の形態において、図1-1に示すように、C-OS又はC-OS上のC-DBGからのステップ実行命令結果(610)をソケット通信で受信するステップ実行命令結果受信部(21)を含んだコマンド実行命令結果受信部(2)と、ステップ実行命令結果受信部(21)で受信したステップ実行命令結果(610)からステップ実行中のプログラムの変数値を抽出する変数値抽出部(32)を含んだ各種情報抽出部(3)とをC-OS上に設置し、さらにC-OS上で抽出した変数値をG-OS上にソケット通信により通知する変数値通知部(112)を含んだ各種情報通知部(101)をC-OS上に設け、さらにG-OS上にソケット通信により通知された変数値を受信する変数値受信部(122)を含んだ各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS上の実行処理負担が減少し、G-OS上でのデバッグ作業の効率が向上する。また、図1-2に示すように、C-OS又はC-OS上のC-DBGからのステップ実行命令結果(610)をソケット通信により受信するステップ実行命令結果受信部(21)を含んだコマンド実行命令結果受信部(2)と、ステップ実行命令結果受信部(21)で受信したステップ実行命令結果(610)からステップ実行中のプログラムの変数値を抽出する変数値抽出部(32)を含んだ各種情報抽出部(3)とを削除し、C-DBG内にステップ実行中のプログラムの変数値を抽出する変数値抽出部(132)を含んだ各種抽出情報抽出部(103)と、抽出したステップ実行中のプログラムの変数値をG-OS上に通知する変数値通知部(112)を含んだ各種情報通知部(101)とをC-OS上に設け、さらに、G-OS上に通知されたステップ実行中のプログラムの変数値を受信する変数値受信部(122)を含んだ各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS、C-OS上の実行処理負担が減少し、デバッグ作業の効率がさらに向上する。また、上記第2の実施の形態において、G-OS上に設置された又はG-OS上とC-OS上にまたがって設置された、ステップ実行中のプログラムの変数値をコマンド(デバッグ(ステップ))実行命令結果を受信して抽出する、又はステップ実行中のプログラムの変数値を抽出して受信する手段は、上記構成に限られることはなく、受信して抽出する、又は抽出して受信する手段であればどのような構成であってもよく(例えば、C-OS上に、コマンド(デバッグ(ステップ))実行命令結果を受信するコマンド(デバッグ(ステップ))実行命令結果受信手段と、コマンド(デバッグ(ステップ))実行命令結果受信手段により受信されたコマンド(デバッグ(ステップ))実行命令結果をG-OSに通知するコマンド(デバッグ(ステップ))実行命令結果通知手段とを設け、コマンド(デバッグ(ステップ))実行命令結果通知手段により通知されたコマンド(デバッグ(ステップ))実行命令結果をコマンド(デバッグ(ステップ))実行命令結果受信部(2(20(21)))で受信するようにしてもよい。)

、抽出元もC-OS又はC-DBG内部に保持されているデバッグに関係する全てのデバッグ情報から抽出するような構成としてよく、さらに抽出情報であるステップ実行中のプログラムの変数値が取得できる手段であればどのような構成であってもよい。また、上記第2の実施の形態において、ソースファイル(10)を表示する手段として、ソースファイル表示部(5)とソースファイル表示実行部(9)とを設けた構成としたが、ソースファイル(10)を表示する手段であればどのような構成でもよく、また、行番号をグラフィカルに表示する手段として、行番号表示部(6)と行番号グラフィカル表示部(41)とを設けた構成としたが、行番号をグラフィカルに表示する手段であればどのような構成でもよく、また、変数値をグラフィカルに表示する手段として、変数値表示部(7)と変数値グラフィカル表示部(42)とを設けた構成としたが、変数値をグラフィカルに表示する手段であればどのような構成でもよい。また、通信手段は、ソケット通信に限ることはなく、シリアル通信としてもよく、通信ができる手段であれば何でもよい。また、C-OS上のC-DBGでのデバッグ時のステップ実行は、C-OS上でのC-DBGのコマンドによるステップ実行命令(510)であってもよいし、G-OS上の一元化デバッガからC-OS上のデバッガに通知するステップ実行命令(510)による実行でもよい。すなわち、ステップ実行命令(510)をC-OS又はC-OS上のC-DBGに通知するステップ実行命令通知部(81)をコマンド実行命令通知部(8)に設け、ステップ実行命令通知部(81)からC-OS又はC-OS上のC-DBGにステップ実行命令をソケット通信により通知する。通知を受けたC-OS又はC-OS上のC-DBGは、ステップ実行命令(510)を実行する。これにより、デバッグがさらに容易となり、デバッグ作業の効率がさらに向上する。尚、ステップ実行命令(510)は、後述するデバッグ実行命令(500)に設けられており、デバッグ実行命令

10

20

30

40

50

(500)は、コマンド実行命令(50)に含まれている。また、ステップ実行命令通知部(81)は、後述するデバッグ実行命令通知部(80)に含まれており、デバッグ実行命令通知部(80)は、コマンド実行命令通知部(8)に設けられている。

【0060】

尚、上記第2の実施の形態は、上記第1の実施の形態に該当する構成を削除した構成とする実施の形態としてもよい。また、上記第2の実施の形態は、上記第1の実施の形態に該当する構成を削除したものを構成する構成要素及び/又は上記第1を構成する構成要素とを自由に組み合わせた構成を実施の形態としてもよい。

【0061】

<本発明の一元化デバッガの1実施の形態としての第3の実施の形態>

10

【0062】

本発明の一元化デバッガの1実施の形態としての第3の実施の形態を図2を用いて説明する。

【0063】

本発明の一元化デバッガの1実施の形態としての第3の実施の形態は、上記第1、第2の実施の形態に以下の機能を追加する実施の形態となっている。すなわち、G-OS上の一元化デバッガ上に設けられたデバッグ実行命令(500)に対応する釦(770)の押下によりコマンド入力無しにグラフィカルにC-DBGのデバッグ実行命令(500)をデバッグ実行命令通知部(80)によりC-OS又はC-OS上のC-DBGにソケット通信により通知し、C-OS又はC-OS上のC-DBGに実行をさせる。デバッグ実行命令結果(600)は、デバッグ実行命令結果受信部(20)により受信され、行番号抽出部(31)及び変数値抽出部(32)によりデバッグ中のプログラムの行番号及び変数値が抽出される。行番号グラフィカル表示部(41)及び変数値グラフィカル表示部(42)は、抽出したデバッグ中のプログラムの行番号及び変数値を行表示部(61)及び変数値表示部(71)に表示し、コマンド実行命令結果グラフィカル表示部(44)は、受信したデバッグ実行命令結果(600)をコマンド実行命令結果表示部(351)に表示する。尚、以下において、C-DBGのデバッグ実行命令(500)とは、break, run, step, next, cont等の実行命令のことを示す。

20

【0064】

本発明の一元化デバッガの1実施の形態としての第3の実施の形態の構成について説明する。本発明の一元化デバッガの1実施の形態としての第3の実施の形態は、G-OS上に格納されている開発ディレクトリ群及び開発ソースファイル群を一覧表示するディレクトリ/ソースファイル群一覧表示部(301)と、ディレクトリ/ソースファイル群一覧表示部(301)にディレクトリ群及びソースファイル群を表示実行するディレクトリ/ソースファイル群一覧表示実行部(302)と、ディレクトリを選択するディレクトリ選択部と、選択されたディレクトリ中の選択されたソースファイル名(311)が示すソースファイル(10)をソースファイル表示部(5)に表示する選択ディレクトリ/ソースファイル表示部(321)と、デバッグ実行命令(500)をC-OS又はC-OS上のC-DBGにソケット通信により通知するデバッグ実行命令通知部(80)と、デバッグ実行命令(500)を命令する、デバッグ実行命令(500)に対応した釦(770)と、C-OS又はC-OS上のC-DBGからデバッグ実行命令結果(600)をソケット通信で受信するデバッグ実行命令結果受信部(20)と、受信したデバッグ実行命令結果(600)からデバッグ実行中のプログラムの行番号を抽出する行番号抽出部(31)と、抽出された行番号をグラフィカルに表示する行番号表示部(61)と、抽出された行番号を行番号表示部(61)にグラフィカルに表示実行する行番号グラフィカル表示部(41)と、受信したデバッグ実行命令結果(600)からデバッグ実行中のプログラムの変数値を抽出する変数値抽出部(32)と、抽出された変数値をグラフィカルに表示実行する変数値表示部(71)と、抽出された変数値を変数値表示部(71)にグラフィカルに表示する変数値グラフィカル表示部(42)とから構成されている。また、デバッグ実行命令通知部(80)は、break, run, step, next, cont等の実行命令に対応する実行命令通知部((821), (822), (841), (842), (843), (844))から構成されており、コマンド実行命令通知部(8)に含まれている。デバッグ実行命令結果受信部(20)は、break, run, step, next, cont等の実行命令に対応する実行命令

30

40

50

結果受信部((221),(222),(241),(242),(243),(244))から構成されており、コマンド実行命令結果受信部(60)に含まれている。各種情報抽出部(3)は、行番号抽出部(31)と変数値抽出部(32)とbreakポイント抽出部(33)とから構成されている。グラフィカル表示部(4)は、行番号グラフィカル表示部(41)と変数値グラフィカル表示部(42)とbreakポイントグラフィカル表示部(43)とから構成されている。

【0065】

次に、本発明の一元化デバッガの1実施の形態としての第3の実施の形態のうちの1つの動作について説明する。G-OS上に格納されている開発ディレクトリ群及び開発ソースファイル群がディレクトリ/ソースファイル群一覧表示実行部(302)によりG-OS上の一元化デバッガ上に用意されたディレクトリ/ソースファイル群一覧表示部(301)に一覧表示 10
されている。この一覧表示されているディレクトリ群から選択したディレクトリ中のソースファイル名(311)をクリック(331)することにより、選択ソースファイル表示部(321)により、ソースファイル(10)の内容が、用意されたソースファイル表示部(5)に表示される(332)。
そのソースファイル(10)上のbreakポイントを設定したい行の所望の位置をクリック(421)することにより、
コマンド入力無しに、グラフィカルに、break実行命令通知部(821)は、break実行命令(521)をC-OS又はC-OS上のC-DBGにソケット通信により通知する。
通知を受けたC-OS又はC-OS上のC-DBGは、break実行命令(521)を実行する。break実行命令結果受信部(221)は、break実行命令結果(621)をソケット通信で受信する。breakポイント抽出部(33)は、受信したbreak実行命令結果(621)からステップ実行中のプログラムの行番号 20
を抽出する。breakポイントグラフィカル表示部(43)は、抽出された行番号によりbreak
ポイントをグラフィカルに設定する(401)。例えば、印として表示する(401)。尚、表示方法は
印に限ることはなく、強調された表示であれば何でも良い。これにより、デバッグが、一元化OS上の一元化デバッガで一元的に実行でき、一元化できると共に、G-OS上の一元化デバッガで、視覚的に一目でbreak実行命令(521)が実行ができ、ソースファイル(10)上にbreakポイント(401)が設定できるので、設定されたbreakポイント(401)を視覚的に一目で理解でき、デバッグが容易となり、デバッグ作業の効率が上がる。

【0066】

尚、上記第3の実施の形態において、図1-1に示すように、C-OS又はC-OS上のC-DBGからのbreak実行命令結果(621)をソケット通信により受信するbreak実行命令受信部(221)を含めたデバッグ実行命令結果受信部(20)を含んだコマンド実行命令結果受信部(2)と、 30
受信したbreak実行命令結果(621)からbreakポイントを抽出するbreakポイント抽出部(33)を含めた各種情報抽出部(3)とをC-OS上に設置し、さらにC-OS上で抽出したbreakポイントをG-OS上にソケット通信により通知するbreakポイント通知部(不図示)を含めた各種抽出情報通知部(101)とをC-OS上に設置し、さらにG-OS上にソケット通信により通知されたbreakポイントを受信するbreakポイント受信部(不図示)を含めた抽出情報を受信する各種情報受信部(102)を設けるようにしてもよい。これにより、G-OS上の実行処理負担が減少し、G-OS上でのデバッグ作業の効率が向上する。また、図1-2に示すように、C-OS又はC-OS上のC-DBGからのbreak実行命令結果(621)をソケット通信により受信するbreak実行命令結果受信部(221)を含んだ、デバッグ実行命令結果(600)を受信するデバッグ実行命令結果受信部(20)を含んだコマンド実行命令結果受信部(2)と、break実行命令結果受信部(221) 40
で受信したbreak実行命令結果(621)からbreakポイントを抽出するbreakポイント抽出部(33)を含んだ、各種情報抽出部(3)とを削除し、C-DBG内にbreakポイントを抽出するbreakポイント抽出部(不図示)を含んだ各種抽出情報抽出部(103)と、抽出したbreakポイントをG-OS上にソケット通信により通知するbreakポイント通知部(不図示)を含んだ各種情報通知部(101)とをC-OS上に設け、さらにG-OS上にソケット通信により通知されたbreakポイントをソケット通信により受信するbreakポイント受信部(不図示)を含んだ各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS, C-OS上の実行処理負担が減少し、デバッグ作業の効率がさらに向上する。また、上記第3の実施の形態において、G-OS上に設置された又はG-OS上とC-OS上にまたがって設置された、各種情報をコマンド(デバッグ(ステップ))実行命令結果を受信して抽出する、又は各種情報を抽出して受信する手 50

段は、上記構成に限られることはなく、受信して抽出する、又は抽出して受信する手段であればどのような構成であってもよく（例えば、C-OS上に、コマンド(デバッグ(ステップ))実行命令結果を受信するコマンド(デバッグ(ステップ))実行命令結果受信手段と、コマンド(デバッグ(ステップ))実行命令結果受信手段により受信されたコマンド(デバッグ(ステップ))実行命令結果をG-OSに通知するコマンド(デバッグ(ステップ))実行命令結果通知手段とを設け、デバッグ情報通知手段により通知されたデバッグ情報をコマンド(デバッグ(ステップ))実行命令結果受信部(2(20(21)))で受信するようにしてもよい。）、抽出元もC-OS又はC-DBG内部に保持されているデバッグに関係する全てのデバッグ情報から抽出するような構成としてよく、さらに抽出情報が取得できる手段であればどのような構成であってもよい。また、上記第3の実施の形態において、ソースファイル群一覧を表示する手段として、ソースファイル群一覧表示部(301)とソースファイル群一覧表示実行部(302)とを設けた構成としたが、ソースファイル群一覧を表示する手段であればどのような構成でもよく、breakポイントを表示する手段として、breakポイント(401)とbreakポイントグラフィカル表示部(43)とを設けた構成としたが、breakポイントを表示する手段であればどのような構成でもよく、また、行番号をグラフィカルに表示する手段として、行番号表示部(61)と行番号グラフィカル表示部(41)とを設けた構成としたが、行番号をグラフィカルに表示する手段であればどのような構成でもよく、また、変数値をグラフィカルに表示する手段として、変数値表示部(71)と変数値グラフィカル表示部(42)とを設けた構成としたが、変数値をグラフィカルに表示する手段であればどのような構成でもよい。また、通信手段は、ソケット通信に限ることはなく、シリアル通信としてもよく、通信ができる手段であれば何でもよい。

【0067】

また、次に、本発明の一元化デバッガの1実施の形態としての第3の実施の形態のうちの1つの動作について説明する。G-OS上の一元化デバッガ上に用意されているrun釦(722)を押下することにより、コマンド入力無しに、グラフィカルに、run実行命令通知部(822)により、run実行命令(522)をソケット通信によりC-OS又はC-OS上のC-DBGに通知する。通知を受けたC-OS又はC-OS上のC-DBGは、受け取ったrun実行命令(522)を実行する。run実行命令結果受信部(222)は、run実行命令結果(622)をソケット通信で受信する。行番号抽出部(31)は、受信したrun実行命令結果(622)からデバッグ実行中のプログラムの行番号を抽出する。抽出された行番号は、行番号グラフィカル表示部(41)によりグラフィカルに表示される(61)。また、変数値抽出部(32)は、受信したrun実行命令結果(622)からデバッグ実行中のプログラムの変数値を抽出する。抽出された変数値は、変数値グラフィカル表示部(42)によりグラフィカルに表示される(61)。例えば、行番号は、上記第1、第2の実施の形態で示したように行表示部(61)により表示され、変数値は、変数値表示部(71)に表示される。これにより、さらにデバッグが、一元化OS上の一元化デバッガで一元的に実行でき、一元化できると共に、G-OS上の一元化デバッガで、視覚的に一目でrun実行命令が実行でき、停止したデバッグ実行中のプログラムの行番号及び変数値を視覚的に一目で理解でき、デバッグが容易となり、デバッグ作業の効率が向上する。

【0068】

尚、上記第3の実施の形態において、図1-1に示すように、C-OS又はC-OS上のC-DBGからのrun実行命令結果(622)をソケット通信により受信するrun実行命令結果受信部(222)を含んだデバッグ実行命令結果受信部(20)を含んだコマンド実行命令結果受信部(2)と、run実行命令結果受信部(222)でソケット通信により受信したrun実行命令結果(622)からデバッグ実行中のプログラムの行番号を抽出する行番号抽出部(31)とrun実行命令結果受信部(222)で受信したrun実行命令結果(622)からデバッグ実行中のプログラムの変数値を抽出する変数値抽出部(32)とを含んだ各種情報抽出部(3)をC-OS上に設置し、さらにC-OS上で抽出したデバッグ中のプログラムの行番号及び変数値をG-OS上にソケット通信により通知する各種情報通知部(101)をC-OS上に設置し、さらにG-OS上に通知されたデバッグ実行中のプログラムの行番号及び変数値をソケット通信により受信する各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS上の実行処理負担が減少し、G-OS

10

20

30

40

50

上でのデバッグ作業の効率が向上する。また、図1-2に示すように、C-OS又はC-OS上のC-DBGからのrun実行命令結果(622)をソケット通信により受信するrun実行命令結果受信部(222)と、run実行命令結果受信部(222)で受信したrun実行命令結果(622)からデバッグ中のプログラムの行番号を抽出する行番号抽出部(31)とrun実行命令結果受信部(222)でソケット通信により受信したrun実行命令結果(622)からデバッグ実行中のプログラムの変数値を抽出する変数値抽出部(32)とを含んだ各種情報抽出部(3)を削除し、C-DBG内にデバッグ実行中のプログラムの行番号を抽出する行番号抽出部(131)と、デバッグ実行中のプログラムの変数値を抽出する変数値抽出部(132)を含んだ各種抽出情報抽出部(103)と、抽出したデバッグ実行中のプログラムの行番号及び変数値をG-OS上にソケット通信により通知する行番号通知部(111)及び変数値通知部(112)とを含んだ各種情報通知部(101)とをC-OS上に設け、さらにG-OS上に通知されたデバッグ実行中のプログラムの行番号及び変数値をソケット通信で受信する行番号受信部(121)及び変数値受信部(122)とを含んだ各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS、C-OS上の実行処理負担が減少し、デバッグ作業の効率が向上する。また、上記第3の実施の形態において、G-OS上に設置された又はG-OS上とC-OS上にまたがって設置された、抽出情報をコマンド(デバッグ)実行命令結果を受信して抽出する、又は抽出情報を抽出して受信する手段は、上記構成に限られることはなく、受信して抽出する、又は抽出して受信する手段であればどのような構成であってもよく(例えば、C-OS上に、コマンド(デバッグ)実行命令結果を受信するコマンド(デバッグ)実行命令結果受信手段と、コマンド(デバッグ)実行命令結果受信手段により受信されたコマンド(デバッグ)実行命令結果をG-OSに通知するコマンド(デバッグ)実行命令結果通知手段とを設け、コマンド(デバッグ)実行命令結果通知手段により通知されたコマンド(デバッグ)実行命令結果をコマンド(デバッグ(ステップ))実行命令結果受信部(2(20))で受信するようにしてもよい。)、抽出元もC-OS又はC-DBG内部に保持されているデバッグに関する全てのデバッグ情報から抽出するような構成としてよく、さらに抽出情報が取得できる手段であればどのような構成であってもよい。また、上記第3の実施の形態において、行番号をグラフィカルに表示する手段として、行番号表示部(61)と行番号グラフィカル表示部(41)とを設けた構成としたが、行番号をグラフィカルに表示する手段であればどのような構成でもよく、また、変数値をグラフィカルに表示する手段として、変数値表示部(71)と変数値グラフィカル表示部(42)とを設けた構成としたが、変数値をグラフィカルに表示する手段であればどのような構成でもよい。また、通信手段は、ソケット通信に限ることはなく、シリアル通信としてもよく、通信ができる手段であれば何でもよい。

【0069】

また、次に、本発明の一元化デバッガの1実施の形態としての第3の実施の形態のうちの1つの動作について説明する。G-OS上の一元化デバッガ上に用意されているstep(next, cont, . . . 左記以外のC-DBGのコマンドに対応する<C-DBGの全てのコマンドに対応する>)釦(741)((742), (743), (744))(尚、以下において、「step(next, cont, . . . 左記以外のC-DBGのコマンドに対応する<C-DBGの全てのコマンドに対応する>)」を「step等のC-DBGの全てのコマンド」と記すこととする。)を押下することにより、コマンド入力無しに、グラフィカルに、デバッグ実行命令通知部(80)は、step等のC-DBGの全てのコマンド実行命令(541)((542), (543), (544))をソケット通信によりC-OS又はC-OS上のC-DBGに通知する。通知を受けたC-OS又はC-OS上のC-DBGは、受け取ったstep等のC-DBGの全てのコマンド実行命令(541)((542), (543), (544))を実行する。デバッグ実行命令結果受信部(20)は、デバッグ実行命令結果(600)をソケット通信により受信する。行番号及び変数値抽出部(31), (32)は、受信したデバッグ実行命令結果(600)からデバッグ実行中のプログラムの行番号及び変数値を抽出する。抽出されたデバッグ実行中のプログラムの行番号及び変数値により、行番号及び変数値グラフィカル表示部((41), (42))がグラフィカルにデバッグ実行中のプログラムの行番号及び変数値を表示する(行表示部(61), 変数値表示部(71))。例えば、デバッグ実行中のプログラムの行番号は、上記第1、第2の実施の形態で示したように行表示部(61)により表示され、デバッグ実行中のプログラムの変数値は、変数値表

10

20

30

40

50

示部(71)に表示される。これにより、さらにデバッグが、一元化OS上の一元化デバッガーで一元的に実行でき、一元化できると共に、G-OS上の一元化デバッガーで、視覚的に一目でstep等のC-DBGの全てのコマンド実行命令(541)((542),(543),(544))が実行でき、デバッグ実行中のプログラムの行番号及び変数値を視覚的に一目で理解でき、デバッグが容易となり、デバッグ作業の効率が向上する。尚、step実行命令等により、他のソースファイルを呼び出す場合は、呼び出されたソースファイル(10)がソースファイル表示部(5)に表示され、停止した行で上記第1、第2の実施の形態で示したように、停止した行に矢印が表示され(行表示部(61))、実行中の行の変数値が変数値表示部(71)に表示される。

【0070】

尚、図1-1に示すように、C-OS又はC-OS上のC-DBGからのstep等のC-DBGの全てのコマンド実行命令結果(641)((642),(643),(644))をソケット通信により受信するstep等のC-DBGの全てのコマンド実行命令結果受信部(241)((242),(243),(244))と、step等のC-DBGの全てのコマンド実行命令結果受信部(241)((242),(243),(244))でソケット通信により受信したstep等のC-DBGの全てのコマンド実行命令結果(641)((642),(643),(644))からデバッグ実行中のプログラムの行番号を抽出する行番号抽出部(31)とstep等のC-DBGの全てのコマンド実行命令結果受信部(241)((242),(243),(244))でソケット通信により受信したstep等のC-DBGの全てのコマンド実行命令結果(641)((642),(643),(644))からデバッグ実行中のプログラムの変数値を抽出する変数値抽出部(32)とを含んだ各種情報抽出部(3)をC-OS上に設置し、さらにC-OS上で抽出したデバッグ実行中のプログラムの行番号及び変数値をソケット通信によりG-OS上に通知する各種情報通知部(101)をC-OS上に設置し、さらにG-OS上に通知されたデバッグ実行中のプログラムの行番号及び変数値をソケット通信により受信する各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS上の実行処理負担が減少し、G-OS上でのデバッグ作業の効率が向上する。また、図1-2に示すように、C-OS又はC-OS上のC-DBGからのstep等のC-DBGの全てのコマンド実行命令結果(641)((642),(643),(644))をソケット通信により受信するstep等のC-DBGの全てのコマンド実行命令結果受信部(241)((242),(243),(244))を含んだデバッグ実行命令結果受信部(20)を含んだコマンド実行命令結果受信部(2)と、step等のC-DBGの全てのコマンド実行命令結果受信部(241)((242),(243),(244))でソケット通信により受信したstep等のC-DBGの全てのコマンド実行命令結果(641)((642),(643),(644))からデバッグ実行中のプログラムの行番号を抽出する行番号抽出部(31)とstep等のC-DBGの全てのコマンド実行命令結果受信部(241)((242),(243),(244))でソケット通信により受信したstep等のC-DBGの全てのコマンド実行命令結果(641)((642),(643),(644))からデバッグ実行中のプログラムの変数値を抽出する変数値抽出部(32)とを含んだ各種情報抽出部(3)を削除し、C-DBG内にデバッグ実行中のプログラムの行番号を抽出する行番号抽出部(131)と、デバッグ実行中のプログラムの変数値を抽出する変数値抽出部(132)とを含んだ各種抽出情報抽出部(103)と、抽出したデバッグ実行中のプログラムの行番号及び変数値をG-OS上に通知する各種情報通知部(101)とをC-OS上に設け、さらにG-OS上に通知されたデバッグ実行中のプログラムの行番号及び変数値をソケット通信により受信する各種情報受信部(102)をG-OS上に設けるようにしてもよい。これにより、G-OS、C-OS上の実行処理負担が減少し、デバッグ作業の効率が向上する。また、上記第3の実施の形態において、G-OS上に設置された又はG-OS上とC-OS上にまたがって設置された、抽出情報をコマンド(デバッグ)実行命令を受信して抽出する、又は抽出情報を抽出して受信する手段は、上記構成に限られることはなく、受信して抽出する、又は抽出して受信する手段であればどのような構成であってもよく(例えば、C-OS上に、コマンド(デバッグ)実行命令結果を受信するコマンド(デバッグ)実行命令結果受信手段と、コマンド(デバッグ)実行命令結果受信手段により受信されたコマンド(デバッグ)実行命令結果をG-OSに通知するコマンド(デバッグ)実行命令結果通知手段とを設け、コマンド(デバッグ)実行命令結果通知手段により通知されたコマンド(デバッグ)実行命令結果をコマンド(デバッグ(ステップ))実行命令結果受信部(2(20))で受信するようにしてもよい。)、抽出元もC-OS又はC-DBG内部に保持されているデバッグに関係する全てのデバッグ情報から抽出するような構成としてよく、さらに抽出情報が取得できる手段であればどの

10

20

30

40

50

ような構成であってもよい。また、上記第3の実施の形態において、行番号をグラフィカルに表示する手段として、行番号表示部(61)と行番号グラフィカル表示部(41)とを設けた構成としたが、行番号をグラフィカルに表示する手段であればどのような構成でもよく、また、変数値をグラフィカルに表示する手段として、変数値表示部(71)と変数値グラフィカル表示部(42)とを設けた構成としたが、変数値をグラフィカルに表示する手段であればどのような構成でもよい。また、通信手段は、ソケット通信に限ることはなく、シリアル通信としてもよく、通信ができる手段であれば何でもよい。

【0071】

尚、上記の本発明の一元化デバッガーの1実施の形態としての第3の実施の形態において、上記のデバッグ実行命令(500)に対応する釦(770)を押下する代わりに、それぞれに対応するキーを割り当てて、そのキーを押下することにより、コマンド入力無しに、グラフィカルに、デバッグ実行命令(500)を通知するようにしてもよい。これにより、さらにデバッグが容易となり、さらにデバッグ作業の効率が向上する。

10

【0072】

尚、上記の本発明の一元化デバッガーの1実施の形態としての第3の実施の形態において、デバッグ実行命令結果表示部を含むコマンド実行命令結果表示部(351)をさらに設け、グラフィカル表示部(4)にデバッグ実行命令結果受信部(20)で受信したデバッグ実行命令結果(600)をコマンド実行命令結果表示部(351)に表示するコマンド実行命令結果グラフィカル表示部(44)をさらに設け、デバッグ実行命令結果受信部(20)で受信したデバッグ実行命令結果(600)をコマンド実行命令結果グラフィカル表示部(44)によりコマンド実行命令結果表示部(351)に表示するようにしてもよい。これにより、デバッグ作業の効率が向上する。尚、コマンド実行命令結果グラフィカル表示部(44)は、全てのコマンド実行命令結果(60)をグラフィカルにコマンド実行命令結果表示部(351)に表示する。また、上記第3の実施の形態において、デバッグ実行命令結果(600)をグラフィカルに表示する手段として、コマンド実行命令結果表示部(351)とコマンド実行命令結果グラフィカル表示部(44)とを設けた構成としたが、デバッグ実行命令結果(600)をグラフィカルに表示する手段であればどのような構成でもよい。

20

【0073】

尚、上記第3の実施の形態は、上記第1、及び/又は上記第2の実施の形態に該当する構成を削除した構成とした実施の形態としてもよい。また、上記第3の実施の形態は、上記第1、及び/又は上記第2の実施の形態に該当する構成を削除したものを構成する構成要素及び/又は上記第1を構成する構成要素、及び/又は上記第2を構成する構成要素とを自由に組み合わせた構成を実施の形態としてもよい。

30

【0074】

<本発明の一元化デバッガーの1実施の形態としての第4の実施の形態>

【0075】

本発明の一元化デバッガーの1実施の形態としての第4の実施の形態を図2を用いて説明する。

【0076】

本発明の一元化デバッガーの1実施の形態としての第4の実施の形態の構成について説明する。本発明の一元化デバッガーの1実施の形態としての第4の実施の形態は、上記第1、第2、第3の実施の形態に以下の構成を追加した実施の形態となっている。すなわち、さらにC-DBGコマンド実行命令(581)を命令する為のC-DBG釦(781)と、C-DBGコマンド実行命令(581)であるログオン及びC-DBG起動命令をC-OS又はC-OS上のC-DBGに通知する為のC-DBGコマンド実行命令通知部(881)を含むコマンド実行命令通知部(8)と、C-DBGコマンド実行命令結果(681)をソケット通信により受信するC-DBGコマンド実行命令結果受信部(281)を含むコマンド実行命令結果受信部(2)とをさらに設けた構成となっており、C-DBGコマンド実行命令結果受信部(281)によりソケット通信で受信されたC-DBGコマンド実行命令結果(681)は、上記コマンド実行命令結果グラフィカル表示部(44)(第3の実施の形態の尚書きに記載)により、コマンド実行結果表示部(351)にグラフィカルに表示される構成と

40

50

なっている。ここで、C-DBGコマンド実行命令(581)は、コマンド実行命令(50)のうちの一つであり、コマンド実行命令(50)に含まれるものである。

【0077】

次に、本発明の一元化デバッガの1実施の形態としての第4の実施の形態の動作について説明する。G-OS上の一元化デバッガの起動時に、又はG-OS上の一元化デバッガの表示領域に用意されたC-DBG釦(781)を押下することにより、コマンド入力無しに、グラフィカルに、C-DBGコマンド実行命令部(881)により、C-DBGコマンド実行命令(581)が開発中のOS(C-OS)に通知され、開発中のOS(C-OS)にログオンし、C-DBGを起動する。尚、この起動時に又はC-DBG釦(781)押下時に、C-DBGコマンド実行命令通知部(881)により、G-OS上に保持してある開発ソースファイルをftp通信によりC-OS上にアップロードしてもよい。C-D 10
BGコマンド実行命令結果受信部(281)によりソケット通信で受信されたC-DBGコマンド実行命令結果(681)は、コマンド実行命令結果グラフィカル表示部(44)によりグラフィカルにコマンド実行命令結果表示部(351)に表示される。これにより、開発ソースファイルのアップロード及びC-DBG起動の手間が省け、さらにデバッグが、一元化OS上の一元化デバッガで一元的に実行でき、一元化できると共に、デバッグが容易となり、デバッグ作業の効率が向上する。ここで、起動時に開発中のOS(C-OS)にログオンし、C-DBGを起動する手段は、C-DBGコマンド実行命令釦(781)を押下した際と同様の処理を行う。また、起動時に開発中のOS(C-OS)にログオンし、C-DBGを起動する手段もコマンド実行命令(50)のうちの一つであり、コマンド実行命令(50)に含まれるものである。

【0078】

尚、上記の本発明の一元化デバッガの1実施の形態としての第4の実施の形態において、C-DBG実行命令結果受信部(281)を含んだコマンド実行命令結果受信部(2)及び各種情報抽出部(3)をC-OS上に移動して設け、受信したC-DBG実行命令結果(681)を各種情報抽出部(3)で受信し、各種情報抽出部(3)で抽出した各種情報をG-OS上にソケット通信により通知する通知部を含む各種情報通知部(101)をさらにC-OS上に設け、G-OS上でC-DBG実行命令結果(681)をソケット通信により受信する受信部を含む各種情報受信部(102)をG-OS上に設け、各種情報受信部(102)で受信した各種情報をグラフィカル表示部(4)に通知するようにしてもよい。また、C-DBG実行命令結果受信部(281)を含んだコマンド実行命令結果受信部(2)をさらにC-OS上に設け、受信したC-DBG実行命令結果(681)をG-OS上にソケット通信により通知する通知部を含む各種情報通知部をさらにC-OS上に設け、C-DBG実行命令結果受 30
信部(281)を含んだコマンド実行命令結果受信部(2)で各種情報通知部から通知されたC-DBG実行命令結果(681)を受信するようにしてもよい。また、上記第4の実施の形態において、C-DBG実行命令結果(681)を含むコマンド実行命令結果(60)をグラフィカルに表示する手段として、コマンド実行命令結果表示部(351)とコマンド実行命令結果グラフィカル表示部(44)とを設けた構成としたが、コマンド実行命令結果(60)をグラフィカルに表示する手段であればどのような構成でもよい。また、通信手段は、ソケット通信に限ることはなく、シリアル通信としてもよく、通信ができる手段であれば何でもよい。

【0079】

尚、上記の本発明の一元化デバッガの1実施の形態としての第4の実施の形態において、C-DBG釦(781)を押下する代わりに、対応するキーを割り当てて、そのキーを押下することにより、コマンド入力無しに、グラフィカルに、上記のC-DBGコマンド実行命令(50)を通知するようにしてもよい。これにより、さらにデバッグが容易となり、さらにデバッグ作業の効率が向上する。

【0080】

尚、上記第4の実施の形態は、上記第1、及び/又は上記第2、及び/又は上記第3の実施の形態に該当する構成を削除した構成とした実施の形態としてもよい。また、上記第4の実施の形態は、上記第1、及び/又は上記第2、及び/又は上記第3の実施の形態に該当する構成を削除したものを構成する構成要素、及び/又は上記第1を構成する構成要素、及び/又は上記第2を構成する構成要素、及び/又は上記第3を構成する構成要素とを自由に組み合わせた構成を実施の形態としてもよい。

10

20

30

40

50

【 0 0 8 1 】

<本発明の一元化デバッガーの1実施の形態としての第5の実施の形態，本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態>

【 0 0 8 2 】

本発明の一元化デバッガーの1実施の形態としての第5の実施の形態を図2を用いて説明する。

【 0 0 8 3 】

本発明の一元化デバッガーの1実施の形態としての第5の実施の形態の構成について説明する。本発明の一元化デバッガーの1実施の形態としての第5の実施の形態は、上記第1、第2、第3、第4の実施の形態に以下の構成を追加した実施の形態となった、本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態である。すなわち、さらに上記の第1、第2、第3、第4の実施の形態のG-OS上の一元化デバッガーに、アップロード実行命令を命令する更新釦(782)と、表示されているソースファイル(10)を編集可能(11)としたソースファイル表示部(5)と、編集ソースファイル(10)をftp通信によりC-OS上にアップロード実行を命令するアップロード実行命令部(882)をコマンド実行命令通知部(8)に設け、さらに、アップロード実行命令結果(682)をソケット通信により受信するアップロード実行命令結果受信部(282)をコマンド実行命令結果受信部(20)に設けた構成とし、アップロード実行命令結果受信部(282)がソケット通信により受信したアップロード実行命令結果(682)は、コマンド実行命令結果グラフィカル表示部(44)によりコマンド実行命令結果表示部(351)にグラフィカルに表示される構成とした。ここで、アップロード実行命令(582)は、コマンド実行命令(50)のうちの一つであり、コマンド実行命令(50)に含まれるものである。

【 0 0 8 4 】

次に、本発明の一元化デバッガーの1実施の形態としての第5の実施の形態の動作について説明する。ソースファイル(10)の編集後に、この更新釦(782)を押下することにより、コマンド入力無しに、グラフィカルに、アップロード実行命令部(882)により、編集ソースファイル(10)をftp通信によりC-OS上にアップロードする。アップロード実行命令結果受信部(282)は、アップロード実行命令結果(682)をソケット通信により受信する。受信したアップロード実行命令結果(682)は、コマンド実行命令結果グラフィカル表示部(44)により、コマンド実行命令結果表示部(351)に表示される。編集ソースファイル(10)のアップロード実行時、G-OS上に格納されているソースファイル(10)も更新される。これにより、さらにデバッグが、一元化OS上の一元化デバッガーで一元的に実行でき、一元化できると共に、G-OS上の一元化デバッガー又は一元化統合開発環境で、ソースファイルを一元管理でき、デバッグ又は統合開発作業がさらに容易となり、デバッグ又は統合開発作業の効率がさらに向上する。

【 0 0 8 5 】

また、上記構成にさらに、make実行命令を命令するmake釦(783)と、make実行命令(583)をC-OS又はC-OS上のC-DBGにソケット通信により通知するmake実行命令通知部(883)を設けたコマンド実行命令通知部(8)と、make実行命令結果(283)をソケット通信により受信するmake実行命令結果受信部(283)をコマンド実行命令結果受信部(2)に設け、make実行命令結果受信部(283)によりソケット通信で受信したmake実行命令結果(683)をコンパイル結果表示部(341)にグラフィカルに表示するコンパイル結果グラフィカル表示部(45)をグラフィカル表示部(4)に設け、make実行命令結果受信部(283)により受信したmake実行命令結果(683)をコンパイル結果表示部(341)に表示する構成としてもよい。ここで、make実行命令(583)は、コマンド実行命令(50)のうちの一つであり、コマンド実行命令(50)に含まれる。

【 0 0 8 6 】

次に、上記構成の動作について説明する。make釦(783)を押下することにより、コマンド入力無しに、グラフィカルに、make実行命令通知部(883)はmake実行命令(583)をソケット通信によりC-OS又はC-OS上のC-DBGに通知する。通知を受けたC-OS又はC-OS上のC-DBGはmake実行命令(583)を実行する。make実行命令結果受信部(283)は、コンパイル結果(make

実行命令結果(683))をソケット通信により受信する。受信したコンパイル結果(make実行命令結果(683))は、コンパイル結果グラフィカル表示部(45)によりグラフィカルにコンパイル結果表示部(341)に表示される。これにより、デバッグ及び統合開発がG-OS上においてコマンド入力無しにグラフィカルに行え、さらにデバッグが、一元化OS上の一元化デバッガーで一元的に実行でき、一元化できると共に、デバッグ及び統合開発が容易となり、デバッグ及び統合開発作業の効率が向上すると共にソースファイルのG-OS上での一元管理が可能となる。

【0087】

尚、上記の本発明の一元化デバッガーの1実施の形態としての第5の実施の形態、又は本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態において、更新釦(782)、make釦(783)を押下する代わりに、それぞれに対応するキーを割り当てて、そのキーを押下することにより、コマンド入力無しに、グラフィカルに、上記の実行命令((582), (583))を通知するようにしてもよい。これにより、さらにデバッグが容易となり、さらにデバッグ作業の効率が向上する。

【0088】

また、尚、上記の本発明の一元化デバッガーの1実施の形態としての第5の実施の形態、又は本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態において、更新(アップロード)、make実行命令結果受信部((282), (283))を含んだコマンド実行命令結果受信部(2)及び各種情報抽出部(3)をC-OS上に移動して設け、受信した更新、make実行命令結果((682), (683))をG-OS上にソケット通信により通知する通知部を含んだ各種情報受信部(101)をさらにC-OS上に設け、G-OS上で更新、make実行命令結果((682), (683))をソケット通信により受信する受信部を含んだ各種情報受信部(102)をG-OS上に設け、各種情報受信部(102)で受信した各種情報をグラフィカル表示部(4)に通知するようにしてもよい。また、更新(アップロード)、make実行命令結果受信部((282), (283))を含んだコマンド実行命令結果受信部(2)をさらにC-OS上に設け、受信した更新、make実行命令結果((682), (683))をG-OS上にソケット通信により通知する通知部を含む各種情報通知部をさらにC-OS上に設け、更新(アップロード)、make実行命令結果受信部((282), (283))を含んだコマンド実行命令結果受信部(2)で各種情報通知部から通知された更新、make実行命令結果((682), (683))を受信するようにしてもよい。また、上記第5の実施の形態において、更新実行命令結果(682)を含むコマンド実行命令結果(60)をグラフィカルに表示する手段として、コマンド実行命令結果表示部(351)とコマンド実行命令結果グラフィカル表示部(44)とを設けた構成としたが、コマンド実行命令結果(60)をグラフィカルに表示する手段であればどのような構成でもよく、また、make実行命令結果(683)をグラフィカルに表示する手段として、コンパイル結果表示部(341)とコンパイル結果グラフィカル表示部(45)とを設けた構成としたが、make実行命令結果(683)をグラフィカルに表示する手段であればどのような構成でもよい。また、通信手段は、ソケット通信に限ることはなく、シリアル通信としてもよく、通信ができる手段であれば何でもよい。

【0089】

また、尚、上記の本発明の一元化デバッガーの1実施の形態としての第5の実施の形態、又は本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態において、ソースファイル(10)編集をより便利に行える機能、例えば、指定行へのジャンプ、指定文字の検索、置換等の機能を備えるようにしてもよい。これにより、デバッグ及び統合開発がさらに容易となり、デバッグ及び統合開発作業の効率がさらに向上する。

【0090】

また、尚、上記の本発明の一元化デバッガーの1実施の形態としての第5の実施の形態、又は本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態において、コマンド実行命令(50)に対応する釦(760)は、C-DBGの全てのコマンドに対応する釦を備えており、また、さらに、全てのC-OSコマンド実行命令に対応できるような釦を備えるようにしてもよい。すなわち、釦(760)を押下することにより、コマンド入力無しに、グラフィカルに、全てのC-DBGコマンド実行命令及び全てのC-OSコマンド実行命令に対応する実

10

20

30

40

50

行命令がコマンド実行命令通知部(8)によりソケット通信で通知される。C-OS又はC-OS上のC-DBGは、通知されたコマンド実行命令(50)を実行する。実行された実行命令結果(60)は、コマンド実行命令結果受信部(2)によりソケット通信で受信される。受信されたコマンド実行命令結果(60)は、抽出情報グラフィカル表示部(4)により、対応する実行命令結果表示部(コマンド実行命令結果表示部(351)、ステップ実行中行表示部(61)、変数値表示部(71)、コンパイル結果表示部(341))にグラフィカルに表示され、又はで表示される。これにより、デバッグ及び統合開発がさらに容易となり、デバッグ及び統合開発作業の効率がさらに向上する。また、上記の本発明の一元化デバッガの1実施の形態としての第5の実施の形態、又は本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態において、コマンド実行命令結果(60)をグラフィカルに表示する手段として、対応する実行命令結果表示部(コマンド実行命令結果表示部(351)、ステップ実行中行表示部(61)、変数値表示部(71)、コンパイル結果表示部(341))と、抽出情報グラフィカル表示部(4)とを設けた構成としたが、コマンド実行命令結果(60)をグラフィカルに表示する手段であればどのような構成でもよい。

10

【0091】

また、尚、上記の本発明の一元化デバッガの1実施の形態としての第5の実施の形態、又は本発明の一元化統合開発環境の1実施の形態としての第1の実施の形態において、C-OS上のディレクトリ構成を一覧表示する一覧表示部(不図示)を設け、現在の、又は実行命令(全てのC-DBGコマンド実行命令及び全てのC-OSコマンド実行命令及び上記のコマンド実行命令を含む全ての実行命令)結果による、ディレクトリ、ファイルの状況をグラフィカルに表示するようにしてもよい。また、C-OS上のファイルの内容を表示するC-OSファイル表示部(不図示)を設け、一覧表示部(不図示)にツリー表示されたディレクトリ、ファイル一覧からファイルをクリックすることにより、ファイルの内容をC-OSファイル表示部(不図示)に表示するようにしてもよい。これにより、デバッグ及び統合開発がさらに容易となり、デバッグ及び統合開発作業の効率がさらに向上する。

20

【0092】

尚、以上において、コマンド実行命令とは、上記実施の形態に示す、ステップ実行命令、デバッグ実行命令、C-OSコマンド実行命令、・・・を含み、かつそれ以外の全ての各種コマンド命令を示す。

【0093】

尚、上記の本発明の一元化デバッガの第1の実施の形態から第5の実施の形態、及び一元化統合開発環境の第1の実施形態において、C-OS上のデバッガはC-DBGに限ることはなく、他のデバッガを使用してもよい。また、他のOSはC-OSに限ることはなく、全てのOSに適用可能であり、また一元化OSはG-OSに限ることはなく、全てのOSに適用可能である。また、上記の本発明の一元化デバッガの第1の実施の形態から第5の実施の形態、及び一元化統合開発環境の第1の実施形態において、他のOSとしてコマンド操作系OSを使用し、コマンド操作系OSとしてC-OSを使用した^が、C-OSに限ることはなく、コマンド操作系OSにも限ることはなく、全てのOSに適用可能であり、また、一元化OSとしてグラフィカル操作系OSを使用し、グラフィカル操作系OSとしてG-OSを使用した^が、G-OSに限ることはなく、グラフィカル操作系OSにも限ることはなく、全てのOSに適用可能である、また、他のOS上のデバッガとしてコマンド操作系デバッガを使用し、コマンド操作系デバッガとしてC-DBGを使用した^が、C-DBGに限ることはなく、またコマンド操作系デバッガにも限ることはなく、全てのデバッガに適用可能である。また、上記において、一元化デバッガと記したものは、機能的に一元化統合開発環境と読み替えることができると共に、一元化統合開発環境と記したものは、一元化デバッガと機能的に読み替えることができる。また、上記の本発明の一元化デバッガの第1の実施の形態から第5の実施の形態、及び一元化統合開発環境の第1の実施形態において、其々の実施の形態は、其々、他の実施の形態を省いた独立した実施の形態とすることができ、其々の独立した実施の形態は、其々「及び/又は」の関係で組み合わせた実施の形態とすることができ、また、其々独立した実施の形態における各構成要素も其々独立した構成とすることができ、其々の独

30

40

50

立した構成要素が其々「及び/又は」の関係で自由に組み合わせた実施の形態とすることができる。また、本発明は上記の実施の形態に限定されるものではない。

【図面の簡単な説明】

【0094】

【図1】本発明の1実施の形態を示す概念図である。

【図1-1】本発明の1実施の形態の他の形態を示す概念図である。

【図1-2】本発明の1実施の形態の他の形態を示す概念図である。

【図2】本発明の1実施の形態を示す概念図である。

【符号の説明】

【0095】

1 一元化デバッガ―又は一元化統合開発環境, 2 コマンド実行結果受信部, 3 各種情報抽出部, 4 抽出情報グラフィカル表示部, 5 ソースファイル表示部, 6 ステップ実行中行表示部, 7 ステップ実行中変数値表示部, 8 コマンド実行命令通知部, 9 ソースファイル表示実行部, 10 ソースファイル, 11 編集可能, 20 デバッグ実行命令受信部, 21 ステップ実行結果受信部, 31 行番号抽出部, 32 変数値抽出部, 33 breakポイント抽出部, 41 行番号グラフィカル表示部, 42 変数値グラフィカル表示部, 43 breakポイントグラフィカル表示部, 44 コマンド実行結果グラフィカル表示部, 45 コンパイル結果グラフィカル表示部, 50 コマンド実行命令, 60 コマンド実行命令結果, 61 デバッグ中行表示部, 71 デバッグ中変数値表示部, 72 カーソル又はマウス, 81 ステップ実行命令通知部, 91 行番号, 92 変数値, 101 各種情報通知部, 102 各種情報受信部, 103 各種抽出情報抽出部, 111 行番号通知部, 112 変数値通知部, 121 行番号受信部, 122 変数値受信部, 131 行番号抽出部, 132 変数値抽出部, 221 break実行命令結果受信部, 222 run実行命令結果受信部, 241 step実行命令結果受信部, 242 next実行命令結果受信部, 243 cont実行命令結果受信部, 244 step, next, cont以外のC-DBGのコマンドに対応する実行命令<C-DBGの全てのコマンドに対応する実行命令>結果受信部, 281 C-DBG実行命令結果受信部, 282 アップロード実行命令結果受信部, 283 make実行命令結果受信部, 301 ディレクトリ/ソースファイル群一覧表示部, 302 ディレクトリ/ソースファイル群一覧表示実行部, 311 ソースファイル名, 321 選択ファイル表示部, 331 クリック, 332 ソースファイル表示, 341 コンパイル結果表示部, 351 コマンド実行命令結果表示部, 401 breakポイント, 421 クリック, 500 デバッグ実行命令, 510 ステップ実行命令, 521 break実行命令, 522 run実行命令, 541 step実行命令, 542 next実行命令, 543 cont実行命令, 544 step, next, cont以外のC-DBGのコマンドに対応する実行命令<C-DBGの全てのコマンドに対応する実行命令>, 581 ログオン及びC-DBG起動, 582 編集ファイルアップロード実行命令, 583 make実行命令, 600 デバッグ実行命令結果, 610 ステップ実行命令結果, 621 break実行命令結果, 622 run実行命令結果, 641 step実行命令結果, 642 next実行命令結果, 643 cont実行命令結果, 644 step, next, cont以外のC-DBGのコマンドに対応する実行命令<C-DBGの全てのコマンドに対応する実行命令>結果受信部, 681 C-DBG実行命令結果, 682 アップロード実行命令結果, 683 make実行命令結果, 722 run釦, 741 step釦, 742 next釦, 743 cont釦, 744 step, next, cont以外のC-DBGのコマンドに対応する<C-DBGの全てのコマンドに対応する>釦, 760 コマンド実行命令に対応する釦, 770 デバッグ実行命令に対応する釦, 781 C-DBG釦, 791 押下, 821 break実行命令通知部, 822 run実行命令通知部, 841 step実行命令通知部, 842 next実行命令通知部, 843 cont実行命令通知部, 844 step, next, cont以外のC-DBGのコマンドに対応する実行命令<C-DBGの全てのコマンドに対応する実行命令>通知部, 881 C-DBG実行命令通知部, 88

10

20

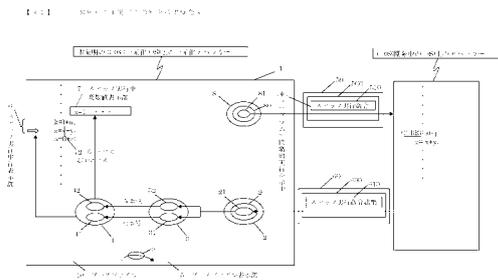
30

40

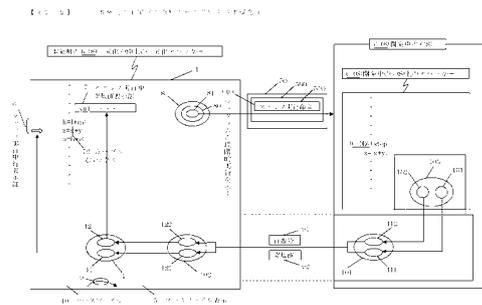
50

2 更新実行命令通知部, 883 make実行命令通知部

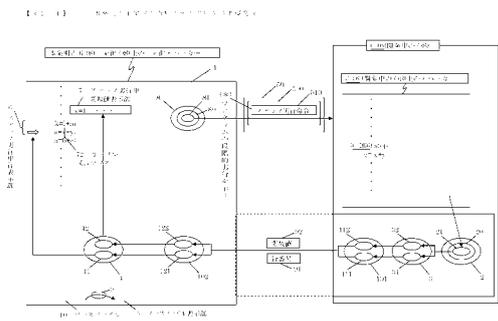
【図1】



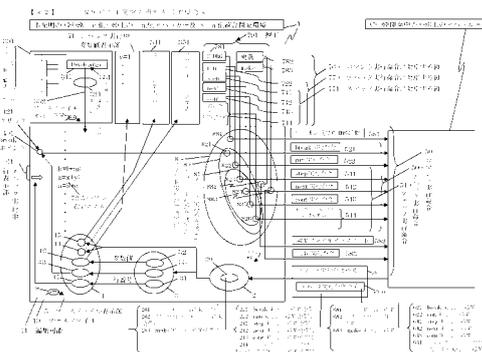
【図1-2】



【図1-1】



【図2】



フロントページの続き

- (56)参考文献 特開平05-313876(JP,A)
特開2001-356908(JP,A)
特開平09-120366(JP,A)
特開平08-305606(JP,A)
特開2001-051871(JP,A)
仲島晶,リアルタイム・シンボリック・デバッガ(RView),情報処理学会研究報告,社団法人情報処理学会,1990年5月8日,第90巻,第31号,p.1-8
ベック・ザレイティアン,Microsoft Visual C++5.0 オーナーズマニュアル,株式会社アスキー,1998年2月11日,第1版,p.455-457
OS IV AF II JCオプション説明書 V01用,富士通株式会社,2001年6月30日,第4版,p.21-24
賢く使おうオフィス・OS系 ウィンドウズユーザーのためのリナックス超入門,YOMIURI PC,読売新聞東京本社,2004年6月1日,第9巻,第6号,p.122-125

(58)調査した分野(Int.Cl.,DB名)

G06F 11/28 P, G06F 9/06 620M