(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2007 年11 月8 日 (08.11.2007)

PCT

(10) 国際公開番号
WO 2007/125911 A1

(51) 国際特許分類:
G06F 11/28 (2006.01) *G06F 21/22* (2006.01)

(21) 国際出願番号: PCT/JP2007/058838

(22) 国際出願日: 2007 年4 月24 日 (24.04.2007)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2006-118881 2006 年4 月24 日 (24.04.2006) JP

(71) 出願人 (米国を除く全ての指定国について): 松下電器産業株式会社 (MATSUSHITA ELECTRIC INDUSTRIAL CO.,LTD.) [JP/JP]; 〒5718501 大阪府門真市大字門真１００６番地 Osaka (JP).

(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 前田 学 (MAEDA, Manabu). 松島 秀樹 (MATSUSHIMA, Hideki). 井藤 好克 (ITO, Yoshikatsu).

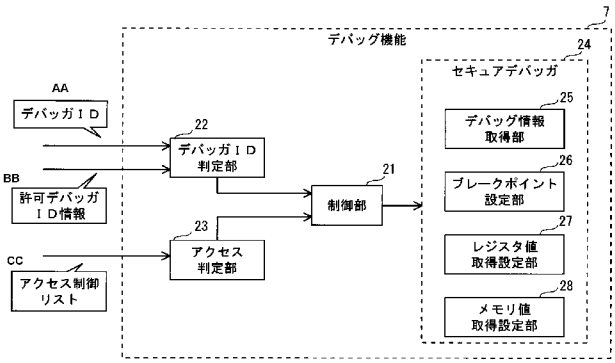
(74) 代理人: 中島 司朗, 外(NAKAJIMA, Shiro et al.); 〒5310072 大阪府大阪市北区豊崎三丁目２番１号淀川５番館６Ｆ Osaka (JP).

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL,

[続葉有]

(54) **Title:** DATA PROCESSING DEVICE, METHOD, PROGRAM, INTEGRATED CIRCUIT, AND PROGRAM GENERATING DEVICE

(54) 発明の名称: データ処理装置、方法、プログラム、集積回路、プログラム生成装置



AA DEBUGGER ID
BB PERMIT DEBUGGER ID INFORMATION
CC ACCESS CONTROL LIST
7 DEBUG FUNCTION
22 DEBUGGER ID JUDGING SECTION
23 ACCESS JUDGING SECTION
21 CONTROL SECTION
24 SECURE DEBUGGER
25 DEBUG INFORMATION ACQUIRING SECTION
26 BREAKPOINT SETTING SECTION
27 REGISTER VALUE ACQUISITION SETTING SECTION
28 MEMORY VALUE ACQUISITION SETTING SECTION

(57) **Abstract:** A data processing device controls the execution of debugging on a program by a debugger. The program includes a validation value for judging whether the debugging can be performed and an access control list indicating whether an access to the program is enabled for each part of the program. The data processing device acquires the ID of the debugger, and the validation value and the access control list included in the program, and judges whether the debugging can be performed according to the result of the comparison between the debugger ID and the validation value. If part of a program to be debugged is indicated as accessible in the access control list, the data processor permits the access, and if not, it does not permit any access.

(57) 要約: データ処理装置は、デバッガによるプログラムに対するデバッグ処理の実行を制御する。プログラムには、デバッグ処理の可否を判定するための検証値と、プログラムへのアクセスの可否をプログラムの部分ごとに示すアクセス制御リストが含まれる。データ処理装置は、デバッガのデバッガＩＤと、プログラムに含まれる検証値およびアクセス制御リストを取得する。デバッガＩＤと検証値との比較結果に応じて

[続葉有]

WO 2007/125911 A1

SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

明 細 書

データ処理装置、方法、プログラム、集積回路、プログラム生成装置

技術分野

[0001] 本発明は、プログラムの保護に関し、特に、プログラムのデバッグ処理の実行を制御する技術に関する。

背景技術

[0002] 著作権管理のためのプログラムのような、不正な解析などの不正行為がなされるのが望ましくないプログラム(以下、「保護プログラム」という)を保護する技術が広く用いられている。保護が不十分であると、プログラムの権利者に限らず様々な方面で損害が発生しうるからである。

例えば、暗号化されたディジタルコンテンツの復号処理を行って再生するプログラムを不正者が不正に解析して改ざんすることができると、ディジタルコンテンツが不正に利用されるおそれが出てくる。すなわち、不正者がディジタルコンテンツを不正に再生したり、ディジタルコンテンツの複製回数や再生回数を制限していても、その制限を無効化したりする。

[0003] 不正者による不正な解析等からプログラム等のデータを保護する技術として、下記の非特許文献1には、外部からの不正アクセスを防止する機構を有するセキュアドメインを構築し、セキュアドメインにおいて処理を行うセキュアモードと、セキュアドメインを用いないで処理を行う通常モードとを備え、通常モードとセキュアモードとを切り換えて動作するLSI(Large Scale Integration)技術が開示されている。この技術によると、セキュアモード時においてのみ保護プログラムを動作させることで、保護プログラムを不正な解析等から保護することができる。

[0004] ところで、セキュアドメインにおいて動作するプログラムを開発する場合、動作の検証や不具合の修正を行うために前記プログラムのデバッグが必要となる。かといって、誰でも前記プログラムをデバッグできるようにすると、不正者がデバッグして不正な解析等の足がかりとされてしまうので、デバッグ可能な者を前記プログラムの開発者などの関係者のみに限定する必要がある。

[0005] そこで、従来は、所定の認証コードを知っている者のみが、セキュアドメインにおいて動作しているプログラムをデバッグ可能とする技術が用いられている(下記の特許文献1参照)。具体的には、特許文献1の技術は、認証コードを用いた認証に成功した者のみが、デバッグ処理を実行できることとしている。

こうすることで、認証コードを知らない者は、認証に失敗するので、セキュアドメインにおいて動作するプログラムを不正者がデバッグすることを防ぐことができる。

特許文献1:特開2004－171565号公報

非特許文献1:TrustZone - Integrated Hardware and Software Security(http://www.arm.com/pdfs/TZ Whitepaper.pdf)

発明の開示

発明が解決しようとする課題

[0006] 上記特許文献1の技術によると、プログラムの開発関係者が例えば自社内のみの特定のグループに限られている場合は、認証コードの管理が適切になされている限り、プログラムの保護を実現することができる。

一方、近年では、プログラム開発の負担を和らげるために、複数のプログラムがそれぞれ別の権利者により開発され、これらプログラムが連係して動作する場合がある。例えば、OMA DRM(Open Mobile Alliance Digital Rights Management)規格で配信された音楽コンテンツを、SDメモリカード(Secure Digital Memory Card)にSD-Audio規格でエクスポートする場合、OMA DRM規格に基づいた処理を行うプログラムとSD-Audio規格に基づいた処理を行うプログラムとがそれぞれ別の権利者によって開発され、これらプログラムが連係して動作する。

[0007] これら連係動作する複数のプログラムの開発時に不具合が発生し、連係動作にかかるプログラムの少なくとも一部が保護プログラムである場合、開発者は、保護プログラムと別のプログラムとを同時にデバッグする必要がある。そうすると、これらプログラムの権利者がそれぞれのプログラムで異なる場合、ある権利者の保護プログラムを、他の権利者側の開発者がデバッグすることが必要となる。

[0008] しかし、上記特許文献1の技術は、セキュアドメインにおいて動作するプログラムをデバッグして良いか否かを、認証コードを用いて制御している。認証コードを知って

いる者は、セキュアドメインにおいて動作する全てのプログラムをデバッグできる。

そうすると、前記他の権利者側の開発者は、認証コードを知っていると、連係動作にかかるプログラムに限らず、あらゆる保護プログラムをデバッグすることができる。その結果、デバッグされることを前記ある権利者が望まない保護プログラムをも、前記他の権利者側の開発者は、その保護プログラムを入手することができればデバッグできてしまう。したがって、前記ある権利者は、連係動作するプログラム群を共同開発する場合、デバッグされることを望まない他の保護プログラムが解析等されて機密情報が外部に流出するかもしれないという多大なリスクを抱えることとなる。このようなリスクを伴うために、連係動作するプログラム群を、プログラムを保護しつつそれぞれ異なる権利者が共同で開発するのは困難である。

[0009] そこで、本発明は、プログラムを保護しつつ、異なる権利者が共同でプログラムを開発することを容易にするデータ処理装置、データ処理方法、集積回路、デバッグ処理の実行を制御するプログラム、プログラム生成装置を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、本発明は、デバッグ処理部によるデバッグ処理の実行を制御するデータ処理装置であって、前記デバッグ処理部を識別する識別子を取得する第1取得手段と、不正アクセスから保護された状態にあるデバッグ対象プログラムの所定部分に含まれる検証値を取得する第2取得手段と、前記デバッグ対象プログラムから取得された前記検証値と前記第1取得手段により取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定手段と、許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御手段とを備えることを特徴とする。

発明の効果

[0011] 本発明のデータ処理装置は、取得した検証値とデバッグ処理部の識別子とを比較して、比較結果に応じてデバッグ処理部によるデバッグ処理の実行の可否を制御する。すなわち、ある検証値をプログラムの権利者が当該プログラムに含ませると、当該プログラムのデバッグ処理が許可されるデバッグ処理部の識別子が定まる。

したがって、当該プログラムの権利者は、当該プログラムのデバッグ処理を行うことが可能なデバッグ処理部の識別子を、自身がそのプログラムに含ませた検証値によって指定することができる。

[0012] つまり、当該プログラムの権利者は、当該プログラムのデバッグ処理が可能な識別子を有するデバッグ処理部を、自身が希望するように限定することができるので、当該識別子を有さないデバッグ処理部を保持している利害関係者等に当該プログラムが不要にデバッグされることを回避することができる。

ところで、プログラムの権利者としては、当該プログラムを他の権利者等にデバッグさせることを許容するとしても、そのプログラムのデバッグを制限したいことがある。例えば、当該プログラムに秘匿しておきたい情報が含まれている場合は、その秘匿部分が暴露されると様々な方面で損害が発生しうるため、その秘匿部分については解析されないようにしたいものである。

[0013] そこで、前記デバッグ対象プログラムの前記所定部分には、前記デバッグ対象プログラムを構成する各部分についてアクセスが許可または不許可であることを示すアクセス制御リストが含まれ、前記第2取得手段は、前記デバッグ対象プログラムの前記所定部分に含まれる前記アクセス制御リストを取得するアクセス制御リスト取得部を含み、前記データ処理装置は、さらに、前記取得した前記アクセス制御リストに基づいて、前記デバッグ対象プログラムの一部分のアクセスが許可されているかを判定するアクセス判定手段を備え、前記制御手段は、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていないと判定された第1の場合、前記一部分のデバッグ処理の実行を禁止し、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていると判定された第2の場合、前記一部分のデバッグ処理を前記デバッグ処理部に実行させることとしてもよい。

[0014] これにより、プログラムの権利者は、当該プログラムにおいて秘匿しておきたい情報が含まれている部分をデバッグされないようにすることができる。

具体的には、前記アクセス制御リストに示される前記部分のそれぞれは、前記デバッグ対象プログラムをメモリにロードする場合におけるロード先メモリアドレスの前記部

分それぞれに対応するアドレス範囲を示し、前記アクセス制御リストには、前記部分それぞれに対応する前記アドレス範囲のそれぞれについて、アクセスの可否が対応づけられており、前記アクセス判定手段は、前記デバッグ対象プログラムをメモリにロードする場合におけるロード先メモリアドレスの前記一部分に対応するアドレス範囲について、前記アクセス制御リストにおいて対応づけられているアクセスの可否を参照することにより前記判定を行うこととしてもよい。

[0015] これにより、特定のメモリアドレスに配置される秘匿情報に対するデバッグを制限することができる。

この他に、前記アクセス制御リストに示される前記部分のそれぞれは、前記デバッグ対象プログラムに含まれるシンボルを示し、前記アクセス制御リストには、前記シンボルのそれぞれについて、アクセスの可否が対応づけられており、前記アクセス判定手段は、前記デバッグ対象プログラムの前記一部分に含まれるシンボルについて、前記アクセス制御リストにおいて対応づけられているアクセスの可否を参照することにより前記判定を行うこととしてもよい。

[0016] これにより、特定のシンボルへのデバッグを制限することができ、シンボルを指定するという簡単な処理で、シンボルが秘匿情報を扱う場合に秘匿情報を保護することができる。

ところで、利害関係者によってプログラムの保護の強度を柔軟に設定したい場合がある。例えば、関連会社の有するデバッグ処理部に対しては、広くデバッグ処理の実行を許可し、部外者のデバッグ処理部に対しては、秘匿情報を含めてデバッグ処理の実行を制限したい場合などである。

[0017] そこで、前記デバッグ対象プログラムの前記所定部分には、前記検証値が複数含まれ、前記アクセス制御リストは、前記所定部分に1以上含まれ、前記アクセス制御リストのそれぞれは、前記検証値の少なくとも1つと対応づけられており、前記判定手段は、前記検証値のそれぞれについて前記取得した識別子と比較して前記判定を行い、前記アクセス判定手段は、前記判定手段により許可されていると判定された前記検証値に対応づけられている前記アクセス制御リストに基づいて前記判定を行うこととしてもよい。

[0018] これにより、検証値それぞれに対応する識別子を有するデバッグ処理部ごとに、デバッグ対象プログラムのデバッグ可能な部分を指定することができる。すなわち、プログラムの権利者は、デバッグ処理部の識別子に応じて、プログラムの保護する部分を設定することができる。

また、前記データ処理装置は、さらに、表示部を備え、前記制御手段は、前記第1の場合、前記一部分のデバッグ処理が禁止された旨を示す表示を前記表示部に行わせ、前記第2の場合、前記デバッグ処理の結果を前記表示部に表示させる表示制御部を含むとすることが望ましい。

[0019] こうすることで、デバッグを行う者は、デバッグが許可されたか否かを知ることができる。

上述のデータ処理装置においては、判定手段による判定の仕方は、前記判定手段は、前記検証値と前記識別子とを比較し、前記検証値と前記識別子とが一致した場合に、前記デバッグ処理が許可されていると判定する、とするとよい。

[0020] これにより、プログラムの権利者はデバッグ処理の実行を許可するデバッグ処理部を前記検証値によって指定することができる。また、前記検証値と前記識別子とが一致するか否かという簡単な演算によって判定を行うので、判定に要する時間が短くて済む。

この他に、判定の仕方としては、前記判定手段は、比較値を不正アクセスから保護された状態で記憶する比較値保持部を含み、前記検証値と前記識別子とを演算子として用いた所定の演算を行い、その演算結果が、前記記憶している前記比較値と一致した場合に、前記デバッグ処理が許可されていると判定する、としてもよい。

[0021] こうすると、仮に前記検証値が何らかの不正な手段により暴露されたとしても、所定の演算と比較値とを用いて前記判定を行うため、前記検証値からは、デバッグ処理が許可されているデバッグ処理部の識別子を得ることが難しい。すなわち、上述の構成によると、プログラムの保護の強度を上げることができる。

このデータ処理装置は、具体的には、前記データ処理装置は、外部からの不正アクセスを防止する機構を備えたセキュアドメインを有し、前記データ処理装置は、動作モードとして、通常モードとセキュアモードとを備え、前記通常モードと前記セキュ

アモードとを切り替えて動作し、前記セキュアモード時においてのみ前記セキュアドメインを用いて動作し、前記データ処理装置は、さらに、前記通常モードと前記セキュアモードとを切り替える切替部を備え、前記通常モードで動作するプログラムは、前記切替部を経由して所定の処理の要求を前記セキュアモードで動作するプログラムに通知することで前記セキュアモードで動作するプログラムにアクセス可能であり、前記デバッグ対象プログラムは、前記セキュアドメインにおいて記憶され、前記第2取得手段は、前記セキュアドメインにおいて前記デバッグ対象プログラムから前記検証値の前記取得を行い、前記判定手段は、前記セキュアドメインにおいて前記判定を行う、という構成で実現することもできる。

[0022] 上述の構成によると、前記デバッグ対象プログラムはセキュアドメインにおいて記憶され、また、前記検証値は、前記第2取得手段により、セキュアドメインにおいて取得される。

したがって、前記検証値は、データ処理装置がこれを取得する過程において、利害関係者や不正者等、何人も知得することが難しい。そのため、不正者等は、プログラムを入手したとしても、プログラムのデバッグ処理が可能な識別子を有するデバッグ処理部を特定することが困難となるので、プログラムが不正にデバッグされる可能性を小さくすることができる。また、プログラムに前記検証値が含まれているので、デバッグの可否を制御するための情報を予めデータ処理装置に記憶させておく必要もない。

[0023] この構成において、前記デバッグ処理部は、前記セキュアドメインの外部にあって前記通常モードにおいて動作し、前記データ処理装置は、さらに、前記セキュアモードにおいてデバッグ処理を実行するセキュアデバッガを備え、前記セキュアデバッガは、前記セキュアドメインに含まれ、前記デバッグ処理部は、前記デバッグ対象プログラムのデバッグ処理要求を出力し、前記制御手段は、前記デバッグ処理部が前記デバッグ処理要求を出力すると、前記判定手段に前記判定を行わせ、許可されていないと判定されたとき、前記デバッグ処理要求にかかる前記デバッグ対象プログラムに対する前記セキュアデバッガによるデバッグ処理を禁止することとするとよい。

[0024] 上述の構成によると、通常モードで動作するデバッグ処理部と、セキュアモードで動

作するセキュアデバッガとを備えているので、デバッグ処理部が不正に改ざん等されても、セキュアデバッガには影響がなく、セキュアモードで動作するプログラムへの不正な解析を防ぐことができる。

また、この構成においても、前記デバッグ対象プログラムの前記所定部分には、前記デバッグ対象プログラムを構成する各部分についてアクセスが許可または不許可であることを示すアクセス制御リストが含まれ、前記第2取得手段は、前記デバッグ対象プログラムの前記所定部分に含まれる前記アクセス制御リストを取得するアクセス制御リスト取得部を含み、前記データ処理装置は、さらに、前記取得した前記アクセス制御リストに基づいて、前記デバッグ対象プログラムの前記一部分のアクセスが許可されているかを判定するアクセス判定手段を備え、前記アクセス判定手段は、前記セキュアドメインにおいて前記判定を行い、前記制御手段は、前記デバッグ処理部が出力した前記デバッグ処理要求にかかる前記デバッグ対象プログラムについて、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていないと判定された場合、前記セキュアデバッガによる前記一部分の前記デバッグ処理の実行を禁止し、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていると判定された場合、前記一部分のデバッグ処理を前記セキュアデバッガに実行させることとしてもよい。

[0025] これにより、プログラムの権利者は、当該プログラムにおいて秘匿しておきたい情報が含まれている部分をデバッグされないようにすることができる。

また、プログラムが連係して動作する場合には、以下の構成とするとよい。

すなわち、前記デバッグ処理部は、前記通常プログラム、および、前記通常プログラムと連係する前記デバッグ対象プログラムに対してデバッグ処理を行う機能を有し、前記デバッグ処理部による前記デバッグ対象プログラムのデバッグ処理は、前記デバッグ処理部がデバッグ処理要求を出力し、出力された前記デバッグ処理要求に対して前記セキュアデバッガにより行われたデバッグ処理の結果を前記切替部を介して前記デバッグ処理部が受け付けることにより行われることとしてもよい。

[0026] これにより、連係して動作するプログラムが通常モードで動作するものであってもセキュアモードで動作するものであっても通常モードで動作するデバッグ処理部でデバ

ッグ処理を同時に実行できるので、連係動作するプログラムを効率良く開発できる。

ところで、通常モードで動作しているプログラムからは、セキュアモードで動作しているプログラムに直接アクセスできないので、セキュアモードにおいて通常モードで動作しているデバッグ処理部は、いつからセキュアモードにおいてデバッグ対象プログラムが動作したかがわからない。そのため、プログラムの開発者にとって、セキュアモードで動作するデバッグ対象プログラムのデバッグが困難となっている。

[0027] そこで、前記デバッグ処理部は、自デバッグ処理部がアタッチされた通常プログラムを識別するプロセス識別子を出力し、前記セキュアデバッガは、前記デバッグ処理部から出力された前記プロセス識別子に示される通常プログラムと連係動作するデバッグ対象プログラムのエントリポイントにある命令をブレーク命令へと変更するとしてもよい。

こうすることで、デバッグ対象プログラムが動作を開始すると、ブレーク命令によりいったん動作が止まるので、プログラムの開発者に対してデバッグ処理の実行のための設定等を行わせることが可能となる。

[0028] また、前記データ処理装置は、前記セキュアモードにおいて前記デバッグ対象プログラムの実行中にデバッグ例外が検出されると、前記切替部を介して前記デバッグ処理部にデバッグ例外の発生を通知し、前記デバッグ処理部は、前記切替部から前記デバッグ例外の発生の通知を受け付けると、デバッグ処理の実行結果を示すデバッグ情報の取得要求を出力し、前記セキュアデバッガは、前記デバッグ対象プログラムについて前記判定手段が肯定的な判定をしているときに前記デバッグ情報の取得要求を受け付けると、前記デバッグ対象プログラムのデバッグ処理を実行してデバッグ情報を取得し、取得したデバッグ情報を、前記切替部を介して前記デバッグ処理部に出力するとするとよい。

[0029] こうすることで、プログラムの開発者は、デバッグ処理が許可されていれば、セキュアモードで動作するプログラムに対するデバッグ処理の結果を知ることができる。

また、前記データ処理装置は、さらに、前記通常プログラムのデバッグ処理の結果を第1の表示領域に表示する第1の結果表示部と、前記通常プログラムとの連係にかかるデバッグ対象プログラムのデバッグ処理の結果を、前記第1の表示領域とは異な

る第2の表示領域に表示する第2の結果表示部とを備え、前記第1および第2の結果表示部は、前記連係にかかるデバッグ対象プログラムと前記通常プログラムとが連係して動作しているとき、前記第1の表示領域および前記第2の表示領域に、前記デバッグ対象プログラムと前記通常プログラムのデバッグ処理の結果を表示するとしてもよい。

[0030] これにより、プログラムの開発者は、通常プログラムとデバッグ対象プログラムの両方の動作を確認しながらデバッグを行うことができる。

また、前記通常モードにおいては、通常OSが動作し、前記セキュアモードにおいては、保護OSが動作し、前記通常プログラムは、前記通常OSが生成するプロセスとして前記通常モードにおいて動作し、前記デバッグ処理部は、前記通常OSで動作するデバッガとして前記通常モードにおいて動作し、前記デバッグ対象プログラムは、前記保護OSが生成するプロセスとして前記セキュアモードにおいて動作し、前記セキュアデバッガは、前記保護OSの有する機能として実装されているとしてもよい。

[0031] また、前記制御手段は、前記デバッグ処理部が前記デバッグ処理要求を出力すると、前記判定手段に前記判定を行わせ、許可されていないと判定されたとき、前記デバッグ処理の実行の禁止を示すデバッグ処理不可通知を前記デバッグ処理部へ出力するとしてもよい。

これにより、デバッグ処理部を用いてデバッグしているユーザは、デバッグ対象プログラムのデバッグが許可されていない場合に、その旨を知ることができる。

[0032] また、上述のプログラムは、以下のようにして生成される。

すなわち、プログラム生成装置は、秘匿すべき保護情報を含んだプログラムを取得するプログラム取得手段と、取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成手段と、前記プログラムについて前記検証値生成手段で生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成手段とを備えることを特徴とする。

[0033] ここで、前記プログラム生成装置は、さらに、前記プログラムを構成する各部分についてアクセスが許可または不許可であることを示すアクセス制御リストを取得するアク

セス制御リスト取得部を含み、前記保護プログラム生成手段は、前記取得したアクセス制御リストを、前記プログラムに付加するアクセス制御リスト付加部を含むとしてもよい。

この構成によると、プログラムの開発者自身が前記検証値をプログラムに含ませるので、開発者の意図したようにプログラムのデバッグ処理の実行を制御することができる。

[0034] また、本発明は、デバッグ処理部によるデバッグ処理の実行を制御するデータ処理方法であって、前記デバッグ処理部を識別する識別子を取得する第1取得ステップと、不正アクセスから保護された状態にあるデバッグ対象プログラムの所定部分に含まれる検証値を取得する第2取得ステップと、前記デバッグ対象プログラムから取得された前記検証値と前記第1取得ステップにおいて取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定ステップと、許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御ステップとを含むことを特徴とするデータ処理方法でもある。

[0035] また、本発明は、デバッグ処理部によるデバッグ処理の実行の制御をデータ処理装置に行わせる、コンピュータ読み取り可能な制御プログラムであって、前記デバッグ処理部を識別する識別子を取得する第1取得ステップと、不正アクセスから保護された状態にあるデバッグ対象プログラムの所定部分に含まれる検証値を取得する第2取得ステップと、前記デバッグ対象プログラムから取得された前記検証値と前記第1取得ステップにおいて取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定ステップと、許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御ステップとを含むことを特徴とする制御プログラムでもある。

[0036] また、本発明は、デバッグ処理部によるデバッグ処理の実行を制御するデータ処理装置において用いられる集積回路であって、前記デバッグ処理部を識別する識別子を取得する第1取得部と、不正アクセスから保護された状態にあるデバッグ対象プロ

グラムの所定部分に含まれる検証値を取得する第2取得部と、前記デバッグ対象プログラムから取得された前記検証値と前記第1取得手段により取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定部と、許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御部とを含むことを特徴とする集積回路でもある。

[0037] また、本発明は、秘匿すべき保護情報を含んだプログラムを取得するプログラム取得ステップと、取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成ステップと、前記プログラムについて前記検証値生成ステップで生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成ステップとを含むことを特徴とするプログラム生成方法でもある。

[0038] また、本発明は、プログラムを生成する処理をプログラム生成装置に行わせるための、コンピュータ読み取り可能な制御プログラムであって、秘匿すべき保護情報を含んだプログラムを取得するプログラム取得ステップと、取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成ステップと、前記プログラムについて前記検証値生成ステップで生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成ステップとを含むことを特徴とする制御プログラムでもある。

[0039] また、本発明は、プログラムを生成するプログラム生成装置において用いられる集積回路であって、秘匿すべき保護情報を含んだプログラムを取得するプログラム取得部と、取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成部と、前記プログラムについて前記検証値生成手段で生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成部とを含むことを特徴とする集積回路でもある。

図面の簡単な説明

[0040] [図1]本発明の実施の形態1におけるデータ処理装置1の概略図。

[図2]デバッグ機能7の詳細なブロック図。

[図3]デバッガID判定部22の詳細なブロック図。

[図4]切替デバドラ13のブロック図。

[図5]暗号化された保護プログラム73を示す図。

[図6]アクセス判定部23が取得するアクセス制御リスト53のデータ構造の例を示す図。

[図7]デバッグ機能7の動作を示す図。

[図8]本発明の実施の形態1における、デバッグを行わない場合の通常プログラム12および保護プログラム8の実行のフローチャート。

[図9]通常プログラム12が、保護プログラム8の機能の実行を必要とする場合の動作を示すフローチャート。

[図10]通常プログラム12aが保護プログラム8aの使用を終了する場合の動作を示すフローチャート。

[図11]デバッガ14が、通常プログラム12と保護プログラム8とに対するデバッグ処理を行うための前処理を示すフローチャート。

[図12]保護プログラム8a実行中にブレークポイントによるデバッグ例外が発生し、デバッガ14を用いて保護プログラム8aに対してデバッグ処理を行うときのフローチャート。

[図13]本発明にかかる保護プログラム8の生成方法を説明するための図。

[図14]保護プログラム生成装置72の構成図。

[図15]保護プログラム生成装置72が、暗号化された保護プログラム73を生成する処理を示すフローチャート。

[図16]デバッガID管理サーバによるデバッガIDの管理方法を説明するための図。

[図17]デバッガID管理サーバ81がデバッガIDの管理に用いるデバッガID管理ファイル90のデータ構造を示す図。

[図18]プログラムの動作情報を表示するためのグラフィカルユーザインターフェース(GUI)を示す図。

[図19]本発明実施の形態5におけるキャラクタベースユーザインターフェース(CUI)

の表示方法の説明図。

符号の説明

[0041] 1 データ処理装置

2 LSI

3 切替機構

6 保護OS

7 デバッグ機能

8 保護プログラム

11 通常OS

12 通常プログラム

13 切替デバイスドライバ

14 デバッガ

15 デバッガ用切替デバイスドライバ

21 制御部

22 デバッガID判定部

23 アクセス判定部

24 セキュアデバッガ

25 デバッグ情報取得部

26 ブレークポイント設定部

27 レジスタ値取得設定部

28 メモリ値取得設定部

31 デバッガID比較部

32 デバッガID演算部

33 比較値保持部

41 切替操作部

42 要求振り分け部

43 通常要求受付部

44 デバッグ要求受付部

51 保護プログラム本体

52 許可デバッガID情報

53 アクセス制御リスト

54 復号化用ヘッダ情報

60 アクセス制御リスト

61 開始アドレス

62 終了アドレス

63 アクセス許可情報

64 シンボル名

65 アクセス許可情報

71 保護プログラムソースコード

72 保護プログラム生成装置

73 暗号化された保護プログラム

74 許可デバッガID格納ファイル

75 アクセス制御リスト格納ファイル

76 秘匿情報領域格納ファイル

77 コンパイラ

78 リンカ

79 保護プログラム化ツール

81 デバッガID管理サーバ

82 保護プログラム開発装置

83 保護プログラム解析装置

90 デバッガID管理ファイル

91 管理番号

92 デバッガID

93 保護プログラムの開発者名

94 連絡先

150 GUI

151 コード表示部

152 レジスタ表示部

153 メモリ表示部

154 シンボル表示部

155 ウォッチポイント表示部

156 コールスタック表示部

157 ウィンドウタイトル表示部

158 メニュー表示部

159 モード表示部

160 通常プログラム用デバッグウィンドウ

161 保護プログラム用デバッグウィンドウ

170 CUI

171 デバッグ処理結果の表示例

発明を実施するための最良の形態

[0042] 以下本発明の実施の一形態について、図面を参照しながら説明する。

1 実施の形態1

図1は、本発明の実施の形態1におけるデータ処理装置1の概略図である。データ処理装置1は、保護機構を持つLSI2と切替機構3、保護OS6、デバッグ機能7、保護プログラム8(8a、8b、・・)、通常OS11、通常プログラム12(12a、12b、・・)、切替デバイスドライバ13(以下、「切替デバドラ13」という)、デバッガ14、デバッガ用切替デバイスドライバ15(以下、「デバッガ用切替デバドラ15」という)から構成される。

[0043] 1.1 実施の形態1における各機能ブロックの説明

1.1.1 LSI2

図1において、LSI2は、プログラムの不正な解析や改竄から保護するための機構である保護機構を搭載している。保護機構は、外部からの不正アクセスを防止するハードウェア機構を有している。保護機構の具体的な例としては、外部からのアクセスを一時的に遮断する等が挙げられる。

[0044] LSI2は、動作モードとして保護モード(または、「セキュアモード」と呼ぶこととしても

よい)と通常モードとを備えており、保護モードと通常モードとを切り替えて動作する。動作モードの切り替えは、後述する切替機構3を用いて行う。

保護モードとは、保護機構によってプログラムが不正な解析や改ざんから保護される特殊なモードであり、保護OS6や保護プログラム8が動作する。一方、通常モードとは、保護機構によってプログラムが保護されない一般的なモードであり、通常OS11や通常プログラム12が動作する。

[0045] 保護モードから通常モードへの切替は、保護OS6が切替機構3を用いて行う。通常モードから保護モードへの切替は、通常OS11にある切替デバドラ13等が切替機構3を用いて行う。

1．1．2 切替機構3

切替機構3は、保護OS6や通常OS11からの、動作モードの切り替え指示を受け付けて、動作モード切り替えに必要な処理を行うためのハードウェア機構を有する。動作モードの切り替え処理は、例えば、非特許文献1に記載の技術を適用することができる。

[0046] この場合、切替機構3は、通常モードにおいても保護モードにおいても動作し、通常モードおよび保護モードがともにアクセス可能な記憶領域を有している。通常モードでLSI2が動作している場合に、通常モードで動作する通常プログラム12から保護モードで動作する保護プログラム8への要求等を、上記記憶領域にいったん格納させる。動作モードが切り替わってから、保護OS6や保護プログラム8が、前記格納された情報を読み出すことで、通常モードで動作するプログラムと保護モードで動作するプログラムとの間の通信を実現する。

[0047] 1．1．3 保護OS6

保護OS6は、LSI2が保護モードの時の、データ処理装置1の動作を制御するOSである。

保護OS6は、保護モード上で動作する保護プログラム8の管理(プロセス管理)やリソース管理、メモリ管理ユニット(MMU)を用いた保護プログラム間のアクセス制御、割り込み処理、切替機構3を用いた通常モードへの切替処理、保護プログラムをデバッグするためにデバッグ機能7を用いたデバッグ処理などを行う。

[0048] 1．1．4 デバッグ機能7

デバッグ機能7は、デバッガ14が保護プログラム8に対するデバッグ処理を行う際に、そのデバッグ処理の実行を制御する。

すなわち、デバッグ機能7は、デバッガ14が保護プログラム8をデバッグするときに、デバッガ14による保護プログラム8のデバッグが許可されているかどうかを判定する。判定の結果、許可されている場合には、デバッガ14からの要求に従い、デバッグ情報の取得やブレークポイントの設定、レジスタ値やメモリ値の取得・設定などの処理を保護プログラム8に対して行う。なお、デバッグ情報とは、プログラムのデバッグのための情報であり、オブジェクトファイル中のプログラムコードとソースコードの対応関係などを示す。また、デバッグ機能7は、停止フラグを用いて、保護プログラム8のデバッグ処理の前処理を行う。なお、デバッグ機能7の詳細については、後述する。

[0049] 1．1．5 保護プログラム8

保護プログラム8は、不正な解析や改ざんから保護しなければならない情報（以下、秘匿情報）を含むアプリケーションプログラムである。

秘匿情報の例としては暗号化されたディジタルコンテンツを復号するための復号鍵や復号アルゴリズム、再生やコピーに関する権利を格納する権利情報等がある。また、保護プログラム8は不正な解析を防止するため、実行が開始されるまでは暗号化された状態で保持されており、実行開始時に保護OS6によって復号化される。

[0050] 1．1．5．1 保護プログラム8の補足

保護プログラム8は、実行が開始されるまでは、図5に示す暗号化された保護プログラム73のように、暗号化された状態で保持されている。

図5に示すように、暗号化された保護プログラム73は、保護プログラム本体51と、許可デバッガID情報52と、アクセス制御リスト53と、復号化用ヘッダ情報54とから構成される。暗号化された保護プログラム73を、復号化用ヘッダ情報54を用いて復号すると、保護プログラム8が得られる。保護プログラム8は、保護プログラム本体51と、許可デバッガID情報52と、アクセス制御リスト53とからなる。

[0051] 1．1．5．2 保護プログラム本体51

保護プログラム本体51は、プログラムの実行コードである。

1．1．5．3 許可デバッガID情報52

許可デバッガID情報52は、保護プログラム8に対するデバッグ処理を許可するか否かを判定するための検証値である。この実施形態では、許可デバッガID情報52とは、保護プログラム8に対するデバッグ処理が許可されているデバッガの識別子（デバッガID）を示していることとする。すなわち、許可デバッガID情報52に示される値と同一のデバッガIDを有するデバッガが、保護プログラム8に対するデバッグ処理を行うことができる。デバッグ機能7の前記判定に際しては、この許可デバッガID情報52が用いられる。

[0052] 1．1．5．4 アクセス制御リスト53

アクセス制御リスト53は、保護プログラム8の所定領域に対するアクセスが許可されているか否かを示すリストである。アクセス制御リスト53とは、要するに、保護プログラム8を構成する各部分のそれぞれについて、アクセスを許可するか否かを対応づけたものである。アクセス制御リスト53の詳細は後述する。

[0053] 1．1．5．5 復号化用ヘッダ情報54

復号化用ヘッダ情報54は、暗号化された保護プログラム73の復号化に必要な情報を示す。例えば、暗号化に用いられたアルゴリズムや、保護プログラム8をメモリにロードするアドレスなどが復号化用ヘッダ情報54には含まれる。なお、暗号化されたプログラムに、復号化に必要な情報を付加してプログラムの復号を行う技術は従来広く知られており、本発明の主要な構成要件ではないため詳細な説明を省略する。

[0054] 1．1．5．6 各データの配置

保護プログラム8のうち、許可デバッガID情報52やアクセス制御リスト53や保護プログラム本体51は、どのように配置してもよい。具体的には、どのように許可デバッガID情報52等を配置しているかという情報を、ヘッダ等の情報としてプログラムに付加してもよい。

[0055] また、予め保護プログラムのどの部分に含まれるかを定義しておき、その定義に従ってデータ処理装置1のデバッグ機能7が許可デバッガID情報52を読み出すこととしてもよい。例えば、許可デバッガID情報52を示すビットの数（またはバイト数）と、アクセス制御リスト53を示すビットの数とを予め定義しておき、保護プログラム8の先頭

から所定ビットまでを許可デバッガID情報52とし、そこから後続する所定ビットまでをアクセス制御リスト53とする。

[0056] なお、許可デバッガID情報52とアクセス制御リスト53とからなる組が複数ある場合は、保護プログラムの先頭に、これら組がいくつあるかを示す情報を含ませておくとしてもよい。

また、これら許可デバッガID情報52等を、保護プログラム8内にどのように配置するかを、復号化用ヘッダ情報54に含めてもよい。デバッグ機能7は、これら許可デバッガID情報52等の保護プログラム8に占める位置を示す情報を読み取る等することで、許可デバッガID情報52やアクセス制御リスト53を取得することができる。

[0057] なお、保護プログラム8の詳細については、実施の形態2でその生成方法とともに詳しく説明する。

1．1．6 通常OS11

通常OS11は、LSI2が通常モードの時の、データ処理装置1の動作を制御するOSである。

[0058] すなわち、通常OS11は、通常モードで動作している時に、通常モードで動作する通常プログラム12の管理(プロセス管理)やリソース管理、割り込み処理等を行う。

1．1．7 切替デバドラ13

切替デバドラ13は、通常OS11のデバイスドライバとして動作し、通常プログラム12が保護プログラム8との通信を行うときに使用する。なお、後述するが、デバッガ14がデバッグ機能7との通信を行う場合は、デバッガ用切替デバドラ15が利用される。

[0059] 具体的には、切替デバドラ13は、通常プログラム12と保護プログラム8との間の通信データの受け渡し処理と、通常モードから保護モードへの切替処理を行う。通信データの受け渡し処理とは、要するに、通常プログラム12が出力するデータを受け付けて、切替機構3を経由して保護プログラム8へ出力し、また、保護プログラム8から出力されたデータを、切替機構3を経由して取得し、取得したデータを通常プログラム12へ出力する処理のことである。

[0060] 切替デバドラ13の詳細については後述する。

1．1．8 デバッガ用切替デバドラ15

デバッガ用切替デバドラ15は、通常OS11のデバイスドライバとして動作し、デバッガ14がデバッグ機能7との通信を行うときに使用する。

デバッガ用切替デバドラ15は、デバッガ14とデバッグ機能7の通信データの受け渡し処理と、通常モードから保護モードへの切替処理を行う。

[0061] 1．1．9 通常プログラム12

通常プログラム12(12a、12b、・・)は、通常OS11上で動作するアプリケーションプログラムである。

通常プログラム12は、切替デバドラ13を利用して保護モードで動作する保護プログラム8と通信を行い、保護プログラム8と連携して動作する。

[0062] 1．1．10 デバッガ14

デバッガ14は、通常プログラム12に対してデバッグ処理を行う機能と、保護プログラム8に対してデバッグ処理を行う機能とを有する。デバッガ14は、自身を識別するための識別子であるデバッガIDを有している。デバッグ機能7による、デバッグが許可されているか否かの判定に際しては、このデバッガIDが用いられる。なお、デバッガ14のデバッガIDの管理等に関しては、実施の形態3で詳しく説明する。

[0063] デバッガ14の通常プログラム12をデバッグする機能は、例えばLinux(登録商標)で使用されるGDBなどのアプリケーションデバッガと同様の機能とすることで実現できる。

また、保護プログラム8に対してデバッグ処理を行う機能とは、デバッガ14が、デバッガ用切替デバドラ15を介して保護OS6のデバッグ機能7と通信し、デバッグ機能7が保護プログラム8に対してデバッグ情報の取得やブレークポイントの設定、レジスタ値やメモリ値の取得・設定などのデバッグ処理を行い、そのデバッグ処理の結果を受け付ける機能のことである。

[0064] なお、以下の説明では、デバッガ14は、通常モードで動作する通常プログラム12にアタッチし、アタッチした通常プログラム(例えば、通常プログラム12a)、および、その通常プログラムと連携動作している保護プログラム(例えば、保護プログラム8a)に対してデバッグ処理を行うものとする。

なお、本実施の形態1におけるデバッガ14は、通常OS11上で動作するアプリケー

ションデバッガとしたが、これに限定するものではなく、例えばLinux(登録商標)で使用されるKGDB等のカーネルモードデバッガとし通常モードや保護モードで動作するデバイスドライバのデバッグを可能としてもよい。

[0065] 1．2 デバッグ機能7の詳細な説明

「1．1．4 デバッグ機能7」で説明したデバッグ機能7の詳細を、以下、説明する。

図2はデバッグ機能7の詳細なブロック図である。デバッグ機能7は、制御部21、デバッガID判定部22、アクセス判定部23、セキュアデバッガ24とを有している。セキュアデバッガ24は、デバッグ情報取得部25、ブレークポイント設定部26、レジスタ値取得設定部27、メモリ値取得設定部28とを有している。

[0066] なお、「1．1．5 保護プログラム8」で説明したように、保護プログラム8には許可デバッガID情報52とアクセス制御リスト53が含まれており、「1．1．10 デバッガ14」で説明したように、デバッガ14は、デバッガIDを有している。

なお、以下のデバッグ機能7の説明では、デバッグ対象の保護プログラムについて、どのプログラムがデバッグ対象か特定せず、デバッグ対象の保護プログラム8と総称して説明する。

[0067] 1．2．1 デバッガID判定部22

図2において、デバッガID判定部22は、デバッガ14が、デバッグ対象の保護プログラム8に対するデバッグ処理を許可されているか否かを判定する。

すなわち、デバッガID判定部22は、デバッガ14の持つデバッガIDと、デバッグ対象の保護プログラム8に含まれる許可デバッガID情報52とを取得する。取得したデバッガIDと許可デバッガID情報52とを比較する。その比較の結果に応じて、デバッグ対象の保護プログラム8のデバッグ(すなわち、デバッガ14によるデバッグ対象の保護プログラムに対するデバッグ処理の実行)がデバッガ14に許可されているか否か判定を行う。

[0068] 1．2．1．1 デバッガID判定部22の詳細な説明

図3はデバッガID判定部22の詳細なブロック図である。

図3に示すように、デバッガID判定部22は、デバッガID比較部31とデバッガID演算部32、比較値保持部33とを有している。デバッガID判定部22は、デバッガ14の

デバッガIDとデバッグ対象の保護プログラム8に含まれる許可デバッガID情報52に示される値とが一致しているかどうかを判定する。

[0069] 具体的に説明すると、図3に示すように、デバッガID演算部32は、デバッガ14のデバッガIDとデバッグ対象の保護プログラム8に含まれる許可デバッガID情報52を受け付ける。デバッガIDから許可デバッガID情報52に示される値を引く。引いた結果を演算結果としてデバッガID比較部31へ出力する。

デバッガID比較部31は、デバッガID演算部32の演算結果と、比較値保持部33の保持している比較値とを比較し、一致している場合に「デバッグ可」を制御部21に通知し、一致しなかった場合に「デバッグ不可」を制御部21に通知する。なお、比較値保持部33は、デバッガID演算部32の演算結果と比較するための比較値として"0"を保持している。

[0070] 1．2．1．2 デバッガID判定部22の補足説明

なお、本実施の形態1におけるデバッガID判定部22は、デバッガ14のデバッガIDとデバッグ対象の保護プログラム8に含まれる許可デバッガID情報52に示される値とが一致しているかどうかを判定するとしたが、デバッガIDの一致を判定することに限定するものではない。すなわち、デバッガID演算部32では減算以外に乗算や暗復号演算を行ってもよい。また、比較値保持部33では"0"以外の値を保持するとしてもよい。

[0071] 要するに、デバッガID判定部22は、デバッガ14の識別子と、デバッグ対象の保護プログラム8に含まれる検証値とを演算子として所定の演算を行い、その結果が、比較値保持部33に保持されている比較値と一致すれば「デバッグ可」と判定し、一致しなければ「デバッグ不可」と判定すればよい。

また、この他に、図示しないデバッガID保持部に予め保護プログラム8の許可デバッガID情報52に示される値を保持しておき、デバッガID判定部22がデバッガ14から受け付けたデバッガ14のデバッガIDと、デバッガID保持部に保持している値とを比較するとしてもよい。この場合、デバッガID演算部32は特に演算を行わない。

[0072] 1．2．2 アクセス判定部23

図2に戻って説明を続ける。

アクセス判定部23は、デバッガ14がデバッグ対象の保護プログラム8の所定領域に対してアクセスを要求している場合に、そのアクセスが許可されているか判定する。

すなわち、アクセス判定部23は、デバッグ対象の保護プログラム8からアクセス制御リスト53を取得し、取得したアクセス制御リスト53に基づいて、デバッガ14がアクセスしようとしている領域へのアクセスが許可されているかどうか判定を行う。

[0073] 1．2．2．1 アクセス制御リスト53の詳細な説明

ここで、アクセス制御リスト53の詳細について説明する。

図6は、アクセス判定部23が取得するアクセス制御リスト53のデータ構造の例を示す図である。

アクセス制御リスト53は、アクセス制御を行う領域と、その領域に関するアクセス許可情報の2つの部分からなる。以下の説明では、メモリアドレスでアクセス制御を行う場合のアクセス制御リスト53aと、シンボルによってアクセス制御を行う場合のアクセス制御リスト53bとについて説明する。なお、シンボルとは、プログラムに含まれる変数や関数などを識別する識別子のことである。

[0074] 1．2．2．2 メモリアドレスでアクセス制御を行う場合

図6(a)は、アクセス制御を行う領域をメモリアドレスで指定するときのアクセス制御リスト53aのデータ構造であり、アクセス制御を行う領域は開始アドレスと終了アドレスで指定される。

図6(a)に示すように、アクセス制御リスト53aの1件のレコードは、開始アドレス61aと、終了アドレス62aと、アクセス許可情報63aとを含む。

[0075] 開始アドレス61aと終了アドレス62aとは、アクセス制御を行うメモリ領域の開始アドレスと終了アドレスとを示す。

アクセス許可情報63aは、開始アドレス61aと終了アドレス62aとに示されるメモリ領域へのアクセスを許可するか否かを示す。許可する場合は「アクセス可」、許可しない場合は「アクセス不可」を例えば1ビットの情報で示す。アクセス制御リスト53aには、開始アドレス61aおよび終了アドレス62aにより示されるメモリ領域の組を複数含んでおり、これら組のそれぞれに、アクセスを許可するか否かを対応づけて記憶している。

[0076] なお、リストの先頭には、開始アドレス61aと終了アドレス62aとにより示されていな

い領域を「default」と定義している。「default」の領域へのアクセスは、本実施形態では、「アクセス不可」としている。

また、開始アドレス61a等に示されるアドレスは、本実施形態では、相対的なアドレスである。すなわち、保護プログラム8の復号化用ヘッダ情報54には、保護プログラム8をメモリにロードする際のメモリアドレスが示されており、このメモリアドレスの先頭を0とした相対的なアドレスが、開始アドレス61a等に示されている。もちろん、開始アドレス61a等に示されるアドレスは、メモリの絶対アドレスとしてもよい。

[0077] 1．2．2．3 メモリアドレスでアクセス制御を行う場合の動作

アクセス判定部23は、アクセス制御リスト53aと保護プログラム8aのデバッグ情報を取得する。また、デバッガ14がアクセスを要求しているシンボルを、保護プログラム8aのデバッグ情報を用いてアドレスに変換する。

アクセス判定部23は、変換したアドレスが、アクセス制御リスト53a中の開始アドレス61aおよび終了アドレス62aにより示されるメモリ領域それぞれに含まれるかどうかをリストの上から順に判定する。含まれると判定されると、その領域に対応づけられたアクセス許可情報63aを取得し、アクセス許可情報63aに示される情報、すなわち「アクセス可」か「アクセス不可」かを制御部21に通知する。また、リスト中のメモリ領域に含まれなかった場合には、リスト先頭の「default」と対応づけられているアクセス許可情報63aに基づいて「アクセス可」か「アクセス不可」かを制御部21に通知する。

[0078] 1．2．2．4 シンボルによってアクセス制御を行う場合

図6(b)は、アクセス制御を行う領域をシンボル名で指定するときのアクセス制御リスト53bのデータ構造であり、アクセス制御を行う領域はシンボル名で指定される。

図6(b)に示すように、アクセス制御リスト53bの1件のレコードは、シンボル名64bと、アクセス許可情報65bとを含む。

[0079] シンボル名64bは、アクセス制御の対象となるシンボルの名称を示す。

アクセス許可情報65bは、シンボル名64bに示されるシンボルに対するアクセスを許可するか否かを示す。

図6(b)に示すように、アクセス制御リスト53bには、シンボルそれぞれについて、アクセスを許可するか否かが示されている。

[0080] なお、リストの先頭には、シンボル名64bに示されていないシンボルを「default」と定義している。「default」のシンボルへのアクセスは、本実施形態では、「アクセス不可」としている。

1．2．2．5 シンボルによってアクセス制御を行う場合の動作

アクセス判定部23は、アクセス制御リスト53bを取得する。デバッガ14がアクセスを要求しているシンボルの名称が、アクセス制御リスト53b中のシンボル名64bに示されるシンボル名と一致するかをリストの上から順に判定する。リスト中のシンボル名と一致した場合にはそのシンボル名に対応づけられたアクセス許可情報65bを取得し、アクセス許可情報65bに示される情報、すなわち「アクセス可」か「アクセス不可」かを制御部21に通知する。リスト中のシンボル名と一致しなかった場合には、リストの先頭の「default」と対応づけられているアクセス許可情報65bに基づいて「アクセス可」か「アクセス不可」かを制御部21に通知する。

[0081] 1．2．2．6 補足説明

なお、本実施の形態1におけるアクセス制御リスト53a、53bは、そのリストの先頭に、アクセス制御リスト53a、53bに示されていないシンボルを「default」として、「default」に対応づけてアクセス許可情報63a、65bを記憶することとしているが、これに限定するものではない。

[0082] すなわち、アクセス判定部23において、アクセス制御リスト53a、53bに含まれないメモリ領域やシンボル名の場合には常に「アクセス可」と判定するものとしてもよいし、逆に「アクセス不可」と判定するものとしてもよい。

また、デバッガ14は、シンボルによりアクセスを要求するとしたが、シンボルに限定するものではない。例えば、デバッガ14はメモリのアドレスによりアクセスする領域を指定してもよい。

[0083] この場合、上記の例ではアクセス制御リスト53aを用いたアクセス許可の判定において、デバッガ14から指定されたシンボルを一旦アドレスに変換する処理を行っていたが、デバッガ14から指定されたアドレスによって直接判定することとなる。

1．2．3 セキュアデバッガ24

セキュアデバッガ24は、デバッガ14からの依頼に応じて、様々なデバッグ処理を行

う。

[0084] セキュアデバッガ24は、デバッグ情報取得部25、ブレークポイント設定部26、レジスタ値取得設定部27、メモリ値取得設定部28を含んでいる。

デバッグ処理としては、デバッグ情報取得部25により、デバッグ対象の保護プログラム8からシンボル情報などのデバッグ情報を取得する処理を行う。

また、ブレークポイント設定部26により、デバッグ対象の保護プログラム8にブレークポイントを設定する処理を行う。

[0085] また、レジスタ値取得設定部27により、デバッグ対象の保護プログラム8が使用しているレジスタ値を取得し、あるいはデバッグ対象の保護プログラム8が使用するレジスタ値を設定する。

また、メモリ値取得設定部28により、デバッグ対象の保護プログラム8が使用しているメモリの値を取得し、あるいはデバッグ対象の保護プログラム8が使用するメモリの値を設定する。

[0086] 1．2．4 制御部21

制御部21はデバッガID判定部22とアクセス判定部23との判定結果に基づいて、デバッガ14によるデバッグ対象の保護プログラム8に対するデバッグ処理の実行が許可されているかを確認する。

確認の結果、デバッグ処理の実行が許可されていれば、制御部21は、デバッガ14からの依頼に応じてセキュアデバッガ24に含まれる各処理（デバッグ情報取得部25、ブレークポイント設定部26、レジスタ値取得設定部27、メモリ値取得設定部28）を呼び出す。

[0087] 制御部21は、デバッガID判定部22により、デバッグ対象の保護プログラム8をデバッガ14がデバッグ処理することが不許可と判定された場合や、アクセス判定部23により、デバッガ14によるアクセスを禁止されている領域へのアクセスと判定された場合には、デバッガ14から依頼された要求を処理しない。すなわち、セキュアデバッガ24に含まれる各処理を呼び出さない。なお、このとき、セキュアデバッガ24に含まれる各処理を呼び出さなかったことをデバッガ14に通知するために、デバッグ処理の実行が許可されなかったことを示すデバッグ処理不可通知をデバッガ14へ出力するこ

ととしてもよい。こうすることで、デバッガ14では、デバッグ処理の実行が許可されなかったことをデバッガ14のユーザに示す等の処理を行うことができる。

[0088] なお、上述のように、許可デバッガID情報52とアクセス制御リスト53とを用いてデバッグ処理の実行の制御を行うとして説明したが、保護プログラム8に含まれる許可デバッガID情報52に示されるデバッガIDは、1つでもよいし複数でもよい。デバッガIDを複数含ませることで、複数の開発者にデバッグ処理を許可することができる。例えば、複数の開発者が共同でプログラムを開発する場合などが考えられる。

[0089] また、許可デバッガID情報52とアクセス制御リスト53とを対応づけた組が、複数、保護プログラム8に含まれることとしてもよい。

もちろん、1のデバッガIDが含まれる許可デバッガID情報52を複数含み、それぞれの許可デバッガID情報52に、アクセス制御リスト53を対応づけることとしてもよい。この場合、アクセス制御リスト53は、それぞれの許可デバッガID情報52ごとに異なるアクセス制限を示すものであってもよい。こうすると、デバッガIDごとに、異なるアクセス制限を課すことができる。

[0090] 制御部21は、デバッガID判定部22に、保持している複数の許可デバッガID情報52に示されるデバッガIDそれぞれについて判定を行わせ、デバッグ許可と判定されたデバッガIDがあれば、アクセス判定部23に、そのデバッガIDに対応したアクセス制御リストによるアクセス判定を依頼する。すべてのデバッガIDがデバッグ不許可と判定された場合には、デバッガ14から依頼された要求を処理しない。

[0091] 1．3 切替デバドラ13の詳細な説明

「1．1．7 切替デバドラ13」で説明した切替デバドラ13の詳細を、以下、説明する。

図4は切替デバドラ13のブロック図である。切替デバドラ13は、切替操作部41、要求振り分け部42、通常要求受付部43、デバッグ要求受付部44とを有している。

[0092] 1．3．1 切替操作部41

切替操作部41は、通常モードで使用しているレジスタ値などを保存することで通常モードでのデータ処理装置の状態を退避し、その後、切替機構3を使用して通常モードから保護モードへ切り替える処理を行う。

さらに、保護モードから通常モードへ切り替わったときに、退避しておいた状態の復帰処理を行い、切り替え時に発生した保護モードからの要求を要求振り分け部42に通知する。なお、この要求としては、デバッグ例外によるデバッグ要求と、通常プログラム12への要求とがある。

[0093] 1．3．2 要求振り分け部42

要求振り分け部42は、保護モードからの要求が、保護モード動作中に発生したデバッグ例外によるデバッグ要求か、通常プログラム12への要求かを判断する。

判断の結果、デバッグ要求の場合にはデバッグ要求受付部44にデバッグ要求を通知し、通常プログラム12への要求の場合は通常要求受付部43に保護モードからの要求を通知する。

[0094] 1．3．3 通常要求受付部43

通常要求受付部43は、保護OS6や保護プログラム8等の保護モードで動作しているプログラムと、通常プログラム12との通信を仲介する。

すなわち、保護モードで動作しているプログラムからの要求を通常プログラム12へ通知したり、反対に通常プログラム12からの要求を保護モードで動作しているプログラムへ通知したりする。

[0095] 1．3．4 デバッグ要求受付部44

デバッグ要求受付部44は、保護プログラム8中に設定されたブレークポイントにより発生したデバッグ例外を、保護プログラム8と連係動作中の通常プログラム12をデバッグしているデバッガ14に通知する。

切替デバドラ13を上述の様な構成にすることにより、通常プログラム12と保護プログラム8との通信を仲介しつつ、保護プログラム8で発生したデバッグ例外を適切なデバッガ14に通知することが可能となる。こうすることで、保護プログラム8に含まれる秘匿情報を無関係なデバッガに通知することによる秘匿情報の流出を防ぐことができる。

[0096] 1．3．5 切替デバドラ13の補足

なお、本実施の形態1における切替デバドラ13は、切替操作部41以外に要求振り分け部42や通常要求受付部43、デバッグ要求受付部44から構成されるとしたが、こ

の様な構成に限定するものではない。例えば、切替デバドラ13は切替操作部41だけからなり、要求振り分け部42や通常要求受付部43、デバッグ要求受付部44は通常プログラム12にライブラリとして組み込まれていてもよい。この場合、要求振り分け部42や通常要求受付部43、デバッグ要求受付部44の動作は、通常プログラム12の実行中に呼び出されるライブラリが行うこととなる。

[0097] 1．4 動作

続いて、データ処理装置1の動作について説明する。

以下の説明では、まず、本発明の特徴でもある、デバッグ機能7の動作を「1．4．1 デバッグ機能7の動作」で説明する。

次に、全体の動作として、デバッグ処理を行わない場合の動作を「1．4．2 デバッグ処理を行わない場合の動作」で説明する。すなわち、通常プログラム12と保護プログラム8との連係動作を説明する。

[0098] 続いて、デバッガ14が、通常プログラム12と保護プログラム8とに対してデバッグ処理を行う場合の動作を、その前処理とあわせて「1．4．3 デバッグ処理の前処理」「1．4．4 デバッグ処理」で説明する。

なお、保護プログラム8は保護モードで動作するプログラムのため、通常モードで動作するプログラムおよびデバッガは保護プログラム8へ直接にはアクセスできない。そのため、セキュアデバッガ24が保護プログラム8へのデバッグ処理を行って、デバッガ14は、その結果を受け付ける。

[0099] 1．4．1 デバッグ機能7の動作

図7は、デバッグ機能7の動作を示す図である。

デバッガID判定部22は、デバッガ14のデバッガIDと保護プログラム8の許可デバッガID情報52とに基づいて、デバッガ14による保護プログラム8のデバッグが許可されているかどうか判定する(S101)。

[0100] 制御部21は、デバッガID判定部22の判定結果に基づいて処理を切り替える(S102)。すなわち、デバッガID判定部22の判定結果が「デバッグ不可」であった場合には(S102:NO)、デバッグ処理を中止する。

デバッガID判定部22の判定結果が「デバッグ可」であった場合には(S102:YES)

、アクセス判定部23は、デバッガ14がアクセスしようとしている保護プログラム8の領域が、デバッガ14によるアクセスが許可されているかを、保護プログラム8のアクセス制御リスト53に基づいて判定する(S103)。

[0101] 制御部21は、アクセス判定部23の判定結果に基づいて処理を切り替える(S104)。すなわち、アクセス判定部23の判定結果が「アクセス不可」であった場合には(S104:NO)、デバッグ処理を中止する。

アクセス判定部23の判定結果が「アクセス可」であった場合には(S104:YES)、デバッグ機能7は、セキュアデバッガ24の各処理部(デバッグ情報取得部25、ブレークポイント設定部26、レジスタ値取得設定部27、メモリ値取得設定部28)を呼び出して処理を行わせる(S105)。

[0102] 1.4.2 デバッグ処理を行わない場合の動作

図8は、本発明の実施の形態1における、デバッグを行わない場合の通常プログラム12および保護プログラム8の実行のフローチャートである。

1.4.2.1 保護プログラム8のロード処理

まず、保護プログラム8のロード処理についての説明を行う。なお、以下では、通常プログラム12と保護プログラム8とは、連係して動作するものとする。通常プログラム12は、通常OS11のデバイスドライバである切替デバドラ13と保護OS6を経由して、保護プログラム8を呼び出す。以下の説明では、通常プログラム12aが、保護プログラム8aと連係して動作する場合を例にして説明する。なお、通常プログラム12bが動作する場合や、保護プログラム8bが連係して動作する場合も同様である。

[0103] 図8に示すように、まず通常プログラム12aが起動される。起動された通常プログラム12aは、動作モードを保護モードに切り替えるための前処理として、切替デバドラ13をopenする(ステップS201)。openするとは、切替デバドラ13が、保護OS6等の、保護モードで動作するプロセスと通信可能な状態にするということである。

通常プログラム12aは、保護プログラム8aを動作させるために、暗号化された保護プログラムを指定し、その指定にかかる暗号化された保護プログラムをメモリへロードさせる要求を、切替デバドラ13を経由して、保護OS6へ通知する(ステップS202)。なお、ここでは、暗号化された保護プログラムを復号すると、保護プログラム8aが生成

されることとする。

[0104] 保護OS6は、暗号化された保護プログラムのロードの要求を受け付けて、暗号化された保護プログラムの復号化用ヘッダ情報から、ロードに必要な情報を取得する(ステップS203)。ロードに必要な情報には、暗号化された保護プログラムに含まれる、保護プログラム本体のロード先アドレスなど、暗号化された保護プログラムの復号化に必要な情報などが含まれる。

[0105] 保護OS6は、取得にかかる、ロードに必要な情報に基づいて、暗号化された保護プログラムを復号化する。復号により得られた保護プログラム8aを、保護モードで管理しているメモリ領域へロードし(ステップS204)、保護プログラム8aを実行できる状態にする。

保護プログラム8aが実行できる状態になると、保護OS6から切替デバドラ13を経由して通常プログラム12aへリターンする(ステップS205)。すなわち、通常プログラム12aにプロセスの実行権が移り、通常プログラム12aの実行が再開される。

[0106] 1.4.2.2 保護プログラム8の実行

続いて、通常OS11において通常プログラム12aが実行している時に、通常プログラム12aが保護プログラム8aの有する機能の実行が必要になった場合の処理を説明する。

図9は、通常プログラム12が、保護プログラム8の機能の実行を必要とする場合の動作を示すフローチャートである。

[0107] なお、図9に示す処理の説明においては、上述の図8で示した処理が既に行われていて、保護プログラム8が実行できる状態にあるとする。

通常プログラム12aは、保護プログラム8aの実行の要求を、切替デバドラ13を経由して、保護OS6へ通知する(ステップS206)。前記実行の要求には、保護プログラム8aが行うべき命令や処理が示されている。

[0108] 保護OS6は、保護プログラム8aの実行の要求を受け付けて、保護プログラム8aを実行し、前記実行の要求に応じた処理を行う(ステップS207)。

保護プログラム8aが処理を終えると、保護プログラム8aから保護OS6、切替デバドラ13を経由して、通常プログラム12aへリターンする(ステップS208)。保護プログラ

ム8aの処理結果を通常プログラム12aが利用することがあれば、切替機構3や切替デバドラ13を経由して保護プログラム8aと通常プログラム12aとの間で処理結果の受け渡しが行われる。

[0109] 保護プログラム8aの有する機能の実行が必要となるたびに、上述のステップS206からステップS208の処理が行われる。

1．4．2．3 保護プログラム8の使用の終了

続いて、通常プログラム12aが保護プログラム8aの使用を終了する場合の動作を説明する。

[0110] 図10は、通常プログラム12aが保護プログラム8aの使用を終了する場合の動作を示すフローチャートである。

図10に示すように、通常プログラム12aが保護プログラム8aの使用を終了すると、通常プログラム12aは、保護プログラム8aの削除の要求を、切替デバドラ13を経由して、保護OS6へ出力する(ステップS209)。なお、削除の要求には、削除の対象となる保護プログラム8が示されている。本実施形態では、削除の要求により保護プログラム8aを削除することとする。

[0111] 保護OS6は、削除の要求を受け付けて、保護プログラム8aを削除する(ステップS210)。その後、保護OS6から切替デバドラ13を経由して通常プログラム12aへリターンする。この削除の操作により、保護プログラム8aの機能は再び保護プログラム8aがロードされるまで使用できなくなる。また、平文の状態である保護プログラム8aがメモリから無くなるため、不正解析を受けにくくすることができる。

[0112] その後、通常プログラム12aにおいて、保護モードへ切り替わる必要がなくなれば、切替デバドラ13をcloseする(S211)。closeするとは、切替デバドラ13が保護OS6等との通信を行わない状態にすることである。

1．4．3 デバッグ処理の前処理

次に、通常プログラム12と保護プログラム8とに対して、デバッガ14がデバッグ処理を行う場合の動作を説明する。まず、デバッグ処理の前処理を説明する。

[0113] 1．4．3．1 前処理

図11は、デバッガ14が、通常プログラム12と保護プログラム8とに対するデバッグ

処理を行うための前処理を示すフローチャートである。

なお、以下の説明では、通常プログラム12aと保護プログラム8aとが連係して動作するものとし、デバッガ14は、通常プログラム12aと保護プログラム8aとに対してデバッグ処理を行うこととする。

[0114] デバッガ14は、プログラム開発者のアタッチの操作を受け付けて、通常プログラム12aに対してデバッグ処理を行うために通常プログラム12aにアタッチされる(S301)。

デバッガ14は、デバッグ処理実行時に保護モードで動作しているデバッグ機能7と通信するために、デバッガ用切替デバドラ15をopenする(S302)。

[0115] デバッガ14は、デバッグ機能7に通常プログラム12aのプロセスIDを通知するために、デバッガ用切替デバドラ15を経由して、通常プログラム12aのプロセスIDを、保護モードで動作するデバッグ機能7に通知する(S303)。

デバッグ機能7は、通知されたプロセスIDを保存し、保護プログラム8aを実行するときに、実行開始直後に保護プログラムを停止するかどうかを示す停止フラグを有効にする(S304)。なお、停止フラグを有効にする処理を行う理由は後述の「1．4．3．2 前処理の補足」で説明する。停止フラグは、1ビットの情報であり、保護機構内でレジスタやメモリ等の記憶領域に記憶される。

[0116] デバッガ14は、プログラム開発者から、デバッグ処理にかかる操作を受け付けて、通常プログラム12aに対するデバッグ処理を行う。必要な処理を終えると、デバッガ14は、プログラム開発者からプログラムの実行再開の操作を受け付けて、デバッグ対象である通常プログラム12aの実行を再開する(S305)。

実行が再開された通常プログラム12aは、切替デバドラ13をopenし、保護プログラム8aのロード処理(S306)と、保護プログラム8aの実行処理(S307)とを切替デバドラ13を経由して保護OS6に要求する。

[0117] なお、ステップS306のロード処理は、図7に示したステップS202、S203、S204とほぼ同様である。また、ステップS307の実行処理は、図8のステップS206、S207、S208とほぼ同様である。異なるのは、デバッガ14動作時には、通常プログラム12aがデバッグされているかどうかをデバッグ機能7が判断するために、各処理で受け渡しするデータと一緒に通常プログラム12aのプロセスIDが通知される(S303)、という

点である。

[0118] また、保護OS6が保護プログラム8aの実行の要求を受け付けたとき、保護OS6は、デバッグ機能7に前処理の実行を要求する(S308)。デバッグ機能7は、要求を受け付けて前処理を行う。ここで、前処理とは、デバッグ機能7が、プロセスIDに対応する停止フラグが有効であるか判断し(S309)、有効になっている場合は(S309:YES)、保護プログラム8aのエントリポイントにある命令をブレーク命令へ変更することをいう(S310)。なお、停止フラグが有効でなければ(S309:NO)、デバッグ機能7は、エントリポイントの命令を変更しない。

[0119] 保護OS6は、保護プログラム8aを実行する(S311)。

1.4.3.2 前処理の補足

なお、上記の動作では、ステップS304で、停止フラグを有効にさせているが、このような処理を行っているのは、要するにプログラム開発者が容易にデバッグできるようにするためである。

[0120] 以下、このような処理を行っている理由を示すと、本実施の形態1では、デバッグを行わない場合の処理の説明にて述べたように、保護プログラム8aは通常プログラム12aに呼び出された時にロードされ、必要なくなるとメモリ上から削除される。ここで、本実施の形態1のプログラムは通常プログラム12aの実行から開始するため、プログラム開発者は、保護プログラム8aへ実行が切り替わったことを確認することが困難である。

[0121] また、本実施の形態1のデバッガ14は、通常プログラムにアタッチするものとしているので、プログラム開発者は、保護プログラム8aに対して直接ブレークポイントを設定することもできず、保護プログラム8aのデバッグが難しくなる。

そのため、プログラム開発者に保護プログラム8aへ処理が移ったことを知らせ、保護プログラム8aに対してブレークポイントの設定を可能とする機会等を与えるために、保護プログラム8aが読み出された際には処理が停止するものとしている。

[0122] 1.4.4 デバッグ処理

図12は、保護プログラム8a実行中にブレークポイントによるデバッグ例外が発生し、デバッガ14を用いて保護プログラム8aに対してデバッグ処理を行うときのフローチ

ャートである。

1．4．4．1 デバッグ処理

図12に示すように、保護プログラム8a実行中に、保護プログラム8aに設定されたブレークポイントによるデバッグ例外が発生すると、保護OS6にデバッグ例外が通知される(S401)。

[0123] 保護OS6は、デバッグ例外の通知を受け付けて、切替デバドラ13にデバッグ例外の発生を通知する(S402)。

切替デバドラ13は、デバッグ例外の発生の通知を受け付けると、デバッガ14にデバッグ処理を行わせるために、デバッグ要求受付部44によりデバッガ14へデバッグ処理実行を要求する(S403)。

[0124] デバッガ14は、デバッグ処理実行の要求を受け付けると、プログラム開発者へデバッグ情報を提供するために、デバッガ用切替デバドラ15を経由して、保護OS6にデバッグ情報の取得を要求する(S404)。このとき、デバッガ14のデバッガIDや、図示しないデバッグ用の通信領域の通知も保護OS6に要求する。ここで、デバッグ用の通信領域は、通常モードと保護モードとの両方がアクセス可能な領域であり、保護モードから通常モードへデバッグ情報を受け渡す際に使用する。

[0125] 保護OS6は、デバッグ機能7にデバッグ情報の取得を要求する(S405)。

デバッグ機能7は、保護プログラム8aに対するデバッグ処理と、保護プログラム8aの、デバッグ処理にかかる所定部分へのアクセスとが許可されているかどうかをデバッガID判定部22およびアクセス判定部23により判定する(S406)。

ステップS406において、デバッグ処理およびアクセスが許可されていたと判定された場合には(S406:YES)、デバッグ情報取得部25は、保護プログラム8aのデバッグ情報を取得し、デバッグ情報をデバッグ用通信用領域にコピーする(S407)。コピー後、デバッグ機能7は、デバッグ情報の取得の完了を、保護OS6、デバッガ用切替デバドラ15を経由してデバッガ14へと通知して、デバッグ機能7からデバッガ14へと処理が戻る(S408)。

[0126] デバッガ14は、デバッグ用通信用領域にコピーされたデバッグ情報を取得し、図示しない表示部に表示することでプログラム開発者にデバッグ情報を示す(S409)。

その後、プログラム開発者が、デバッグ情報を参照して必要な処理を終えると、デバッガ14は、プログラム開発者から所定の操作を受け付けて、デバッグ対象である保護プログラム8aの実行再開を、切替デバドラ13を経由して保護OS6に要求する(S410)。

[0127] 保護OS6は、実行再開の要求を受け付けて、保護プログラム8aの実行を再開する(S411)。

この後、再びデバッグ例外が発生した場合には、同様のフローで処理が行われる。

1.4.4.2 デバッグ処理の補足

以上の例では、保護プログラム8a実行中にブレークポイントによるデバッグ例外が発生した場合の処理について説明した。この他にも、デバッグ機能7を使ったデバッグ処理のパターンとしては、様々なものが考えられる。具体的には、プログラム開発者によってブレークポイントの設定処理やレジスタ値・メモリ値の設定や取得処理などの他のデバッグ処理が要求され、この要求に応じてデバッグ機能7がデバッグ処理を行う。これらの場合にも、上記と同様のフローでデバッグ機能7(厳密には、セキュアデバッガ24の持つ各機能部)により処理が行われるが、基本的な処理は同様であるので、詳細な説明は省略する。

[0128] なお、本実施の形態1においては、デバッグ例外発生時に、その例外発生にかかる保護プログラム8aに対するデバッグ処理を行えるとしたが、これは保護プログラム8に限定するものではない。デバッガ14がデバッグしている通常プログラム12aなどの通常プログラムに対するデバッグ処理も行えるとしてもよい。

2 実施の形態2

以下、実施の形態2について説明する。実施の形態2では、特に、保護プログラム8の生成方法、および、保護プログラム8を生成するプログラム生成装置について説明する。

[0129] 2.1 プログラムの生成方法の概略

図13は、本発明にかかる保護プログラム8の生成方法を説明するための図である。

保護プログラム8は、暗号化されて生成される。暗号化された保護プログラム73が生成されるのに、保護プログラムソースコード71と、保護プログラム生成装置72と、保

護プログラムに付加される情報である許可デバッガID格納ファイル74とアクセス制御リスト格納ファイル75、そして秘匿情報領域格納ファイル76が用いられる。

[0130] 図13に示す保護プログラムソースコード71は、保護プログラム8の動作を記述したソースコードである。

保護プログラム生成装置72は、保護プログラムソースコード71のコンパイルとリンクを行う。生成された実行ファイルに許可デバッガID情報とアクセス制御リストを付加し、暗号化を行う。さらに、復号化に必要な情報を復号化用ヘッダ情報として付加することで、暗号化された保護プログラム73を生成する。具体的には、後述する。

[0131] 暗号化された保護プログラム73は、保護プログラム生成装置72により生成されたプログラムである。

許可デバッガID格納ファイル74とアクセス制御リスト格納ファイル75は、それぞれ、保護プログラム8に対してデバッグが行われる時にデバッグ機能7やデバッガ14が使用する許可デバッガID情報、アクセス制御リストを含むデータである。

[0132] 秘匿情報領域格納ファイル76は、プログラム中の各情報の領域と、その領域に関する情報が秘匿情報か否かを示す秘匿情報区分から構成される。

保護プログラム開発者は、保護プログラムソースコード71を作成する。また、ソースコード中でデバッガからのアクセスを許可する領域と不許可にする領域とをアクセス制御リストとして作成し、アクセス制御リスト格納ファイル75に記す。さらに、秘匿情報の領域とそうでない領域を、秘匿情報領域格納ファイル76に記す。許可デバッガID格納ファイル74は、別途入手する。入手した許可デバッガID格納ファイル74と保護プログラムソースコード71、アクセス制御リスト格納ファイル75、秘匿情報領域格納ファイル76を入力として保護プログラム生成装置72を動作させる。その結果、暗号化された保護プログラム73が生成される。

[0133] 2．2 保護プログラム生成装置72の構成

図14は、保護プログラム生成装置72の構成図である。

保護プログラム生成装置72は、コンパイラ77と、リンカ78、保護プログラム化ツール79から構成される。

2．2．1 コンパイラ77

図14に示すコンパイラ77は、入力された保護プログラムソースコード71をコンパイルしてオブジェクトファイルを生成する。変数や関数の配置を示すシンボル情報や、オブジェクトファイル中のプログラムコードとソースコードの対応関係などをデバッグ情報として作成し、オブジェクトファイルに付加する。

[0134] 2.2.2 リンカ78

リンカ78は、コンパイラ77により生成されたオブジェクトファイルとライブラリとをリンクし、実行可能なファイルを生成する。さらに、リンカ78は、生成された実行可能なファイル中のどこにどの変数や関数を配置したかを示すシンボルファイルを生成する。

[0135] 2.2.3 保護プログラム化ツール79

保護プログラム化ツール79は、リンカ78が作成した実行可能ファイルのヘッダに、保護プログラム生成装置72に入力される許可デバッガID格納ファイル74に格納されている許可デバッガID情報と、アクセス制御リスト格納ファイル75に格納されているアクセス制御リストとを付加する。さらに、秘匿情報領域格納ファイル76に記載された各領域が秘匿情報であるか否かの情報をデバッグ情報に追加し、保護プログラムを生成する。

[0136] 保護プログラム化ツール79は、生成された保護プログラムを、保護OS6と保護プログラム生成装置72とで共通に保持している鍵により暗号化を行い、保護プログラムをロードするアドレスなどを復号化用ヘッダ情報として付加する。

なお、暗号化は、保護OS6と保護プログラム生成装置72とが共通に保持している鍵を用いる(いわゆる共通鍵暗号方式)こととしたが、互いに異なる鍵を保持することとなる公開鍵暗号方式などを用いてもよいことは言うまでもない。

[0137] なお、アクセス制御リストがシンボル名とアクセス許可情報の組という形式で与えられていた場合は、リンカ78が出力したシンボルファイルを使用して、シンボル名からそのシンボルが配置されている領域を求め、アクセス制御リストのシンボル名を、その領域(つまり、実行可能なファイル中のどこかを示す情報)へ変更してもよい。

なお、暗号化された保護プログラム73の構成は、実施の形態1において、図5を用いて説明した通りである。暗号化された保護プログラム73は、保護プログラム本体51と許可デバッガID情報52、アクセス制御リスト53、復号化用ヘッダ情報54から構成

され、許可デバッガID情報52とアクセス制御リスト53が保護プログラム本体51のヘッダに付加され、暗号化された構造になっている。復号化用ヘッダ情報54は復号に必要なデータを格納しているために暗号化はされない。

[0138] 2．3 保護プログラム生成装置72の動作

次に、保護プログラム生成装置72による、暗号化された保護プログラム73を生成する処理について説明する。

図15は、保護プログラム生成装置72が、暗号化された保護プログラム73を生成する処理を示すフローチャートである。

[0139] なお、保護プログラムソースコード71、アクセス制御リスト格納ファイル75、秘匿情報領域格納ファイル76は、プログラム開発者が既に作成しているものとする。すなわち、プログラム開発者は、保護プログラムソースコード71を記述する。また、ソースコード中でデバッガのアクセスを不可にする領域と可能にする領域を決定してアクセス制御リストを作成し、アクセス制御リスト格納ファイル75として記述する。また、秘匿情報が記されている領域を秘匿情報領域格納ファイルに記述する。

[0140] また、プログラム開発者は、許可デバッガID格納ファイル74を、別途入手しているものとする。

保護プログラム生成装置72の動作について説明すると、保護プログラム生成装置72は、保護プログラムソースコード71と、許可デバッガID格納ファイル74と、アクセス制御リスト格納ファイル75と、秘匿情報領域格納ファイル76とを入力として受け付ける(S501)。

[0141] 保護プログラム生成装置72は、コンパイラ77とリンカ78を用いて、入力された保護プログラムソースコード71のコンパイルとリンクを行う(S502)。

保護プログラム化ツール79は、保護プログラムソースコード71のコンパイルとリンクにより生成された保護プログラム本体51に、アクセス制御リスト格納ファイル75に格納されているアクセス制御リスト53と、デバッガID格納ファイル74に格納されている許可デバッガID情報52とを付加し(S503)、さらに、秘匿情報領域格納ファイル76に記載された各領域が秘匿情報であるか否かの情報をデバッグ情報に追加して、暗号化を行う(S504)。

[0142] 保護プログラム化ツール79は、暗号化されたプログラムに保護プログラム本体51のロード先アドレスなど復号に必要な情報を復号化用ヘッダ情報54として付加し、暗号化された保護プログラム73として出力する(S505)。

2．4 実施の形態2の補足説明

なお、上記の説明では、秘匿情報領域格納ファイル76への書き込みをプログラム開発者が行うものとしたが、これに限られない。例えば、コンパイラ等のコンピュータプログラムが自動的にこれらのアクセス制御リストや秘匿情報格納ファイルの作成等を行ってもよい。より具体的には、ソースコードに対して秘匿情報のある箇所に予め何らかの印をつけておき、コンパイラがその印の有無に従って秘匿情報格納ファイルへの書き込みを行ったりしてもよい。

[0143] また、上記の説明では、保護プログラムソースコードのコンパイルから保護プログラムの作成までの全ての処理を保護プログラム生成装置72が行っているが、これに限られない。例えば、保護プログラムの生成と、アクセス制御リスト等の付加とを異なる装置で行うとしてもよい。この場合、保護プログラム生成装置72は、保護プログラムソースコード71をコンパイルおよびリンクして保護プログラム本体51を生成する装置と、生成された保護プログラム本体51を取得してアクセス制御リスト等を付加する装置との組として構成されることとなる。上述の構成では、保護プログラム本体51を生成する装置は、入力として保護プログラムソースコード71のみがあれば十分である。

[0144] また、アクセス制御リスト等を付加する装置には、保護プログラム本体51が与えられるので、保護プログラムソースコード71の入力は不要である。このような構成だと、保護プログラム本体51の作成とアクセス制御リスト等の付加を別々に行えるので、それぞれの作業を別会社等に委託することでプログラムの開発効率の向上を図ることなどができる。

[0145] 3 実施の形態3

以下、実施の形態3について説明する。実施の形態3では、特に、デバッガIDをどのように管理するかについて説明する。なお、実施の形態3においては、デバッガIDは、デバッガID管理サーバによって管理されているものとして説明する。

図16は、デバッガID管理サーバによるデバッガIDの管理方法を説明するための

図である。

[0146] 3．1 構成の説明

3．1．1 デバッガID管理サーバ81

図16に示すデバッガID管理サーバ81は、デバッガIDを管理している。デバッガIDの管理は、後述するデバッガID管理ファイル90を用いて行う。

保護プログラム開発者が誰であるかに応じて（または、保護プログラム開発者の用いる保護プログラム開発装置82に応じて）デバッグ処理の実行可否を制御するためには、デバッガIDは、保護プログラム開発者ごと（または、保護プログラム開発装置82ごと）にそれぞれ異なる値にすることが望ましい。そのため、デバッガID管理サーバ81は、複数の保護プログラム開発者（または、保護プログラム開発装置82）に同じデバッガIDが付与されないように管理する必要がある。

[0147] デバッガID管理サーバ81は、デバッガIDを管理する会社（デバッガID管理会社）が持つサーバである（なお、「会社」としているが、これに限るものではなく、デバッガIDを管理する管理者であれば、会社以外の組織や個人であってもよいことは言うまでもない）。

デバッガID管理サーバ81は保護プログラム開発装置82からの要求により、過去に発行したデバッガIDと異なるデバッガIDを発行し、発行したデバッガIDを格納したデバッガID格納ファイルを保護プログラム開発装置82に提供する。また、デバッガID管理会社は、デバッガIDに対応したIDを持つデバッガを作成して保護プログラム開発装置82に提供する。

[0148] 3．1．2 保護プログラム開発装置82

保護プログラム開発装置82は、保護プログラム生成装置72を用いて保護プログラムを作成する。

具体的には、保護プログラム開発装置82は、保護プログラム解析装置83から、デバッガID格納ファイルを受け付ける。受け付けたデバッガID格納ファイルに示されるデバッガIDを取得して、許可デバッガID格納ファイル74として、実施の形態2で説明した保護プログラム生成装置72へ入力することで、取得したデバッガIDに示されるデバッガへのデバッグを許可する保護プログラム8を生成する。

[0149] なお、保護プログラム開発装置82は、保護プログラムを開発するプログラム開発者（個人もしくは組織）が所有している。

3．1．3 保護プログラム解析装置83

保護プログラム解析装置83は、保護プログラムに含まれる不具合の解析を行うための装置である。保護プログラム解析装置83は、個人もしくは組織が所有している。

[0150] 具体的には、保護プログラム解析装置83は、デバッガID管理サーバ81からデバッガIDを発行してもらい、入手したデバッガID格納ファイルを、解析対象の保護プログラムを開発した保護プログラム開発装置82に提供する。

デバッガID格納ファイルを提供した保護プログラム開発装置82が、デバッガID格納ファイルに基づいて保護プログラムを生成すると、保護プログラム開発装置82から、デバッグ可能な保護プログラムを入手し、その保護プログラムを解析する。

[0151] 3．2 デバッガID管理ファイル90のデータ構造

図17は、デバッガID管理サーバ81がデバッガIDの管理に用いるデバッガID管理ファイル90のデータ構造である。

デバッガID管理ファイル90の1件のレコードは、管理番号91と、デバッガID92と、保護プログラムの開発者名93と、連絡先94とから構成される。

[0152] 管理番号91には、デバッガID管理サーバ81で発行済みのデバッガIDを管理するための番号が格納される。

デバッガID92には、管理番号91で管理されている発行済みのデバッガIDの値が格納される。なお、デバッガID92は、各プログラム開発者（または、保護プログラム開発装置82のそれぞれ）を識別でき、かつ、デバッガのデバッグIDをなりすましてデバッグ処理を行うなりすまし攻撃を防ぐために、十分な長さの数値列とすることが望ましい。

[0153] 保護プログラムの開発者名93には、デバッガIDの発行を申請してきたプログラム開発者の名前が格納される。

連絡先94には、デバッガIDの発行を申請してきたプログラム開発者の連絡先が格納される。

3．3 実施の形態3の補足説明

なお、実施の形態3において、デバッガID管理サーバ81は、保護プログラム開発装置82にデバッガを提供するとしたが、デバッガの提供先を保護プログラム開発装置82に限定するものではない。例えば、保護プログラム解析装置83に提供してもよい。

[0154] また、デバッガID管理ファイル90は発行済みのデバッガIDを管理するためのファイルとしたが、発行済みのデバッガIDに限定するものではなく、未発行のデバッグIDも一緒に管理してもよい。

さらに、デバッガID管理ファイル90で発行済みデバッガIDの発行先のプログラム開発者の名前やその連絡先も管理するとしたが、発行済みのデバッガIDのみを管理するとしても良い。

[0155] また、デバッガは、必ずしもデバッガID管理会社から提供されるものではなく、デバッグ管理会社からデバッガの開発を委託された他者のサーバ等から提供されるものとしてもよい。この場合、その他者は、デバッガIDの情報を、デバッガID管理サーバ81から取得する。

4 実施の形態4

以下の実施の形態では、上述のデバッガ14等が行ったデバッグ処理の結果を、どのように表示するかについて説明する。

[0156] 実施の形態4は、実施の形態1のデバッガに対して、プログラムの動作情報をグラフィカルユーザインターフェースに表示する表示手段を付加したものである。デバッガ本体の機能は実施の形態1と同様であるので、説明を省略する。

ここで、プログラムの動作情報とは、デバッグ対象となっているプログラムをデバッグするために参照する情報、つまり、プログラムの動作に関連する情報のことを指す。具体的には、以下の例では、動作情報は、プログラムのコードや、プログラムを実行しているプロセッサの各レジスタの値、ローカル変数のシンボル名とその値、シンボルにより指定された変数とその値、メモリの使用量等の値、関数の呼び出し階層、および、保護モードと通常モードのどちらのモードで動作しているかの情報などを指す。なお、実施の形態1を例とすると、保護プログラム8の動作情報は、デバッガ14がデバッグ機能7を介して取得する情報に基づいて得られ、通常プログラム12の動作情

報は、デバッガ14が直接取得する情報に基づいて得られる。

[0157] 4．1 GUI150aの説明

図18は、プログラムの動作情報を表示するためのグラフィカルユーザインターフェース(GUI)を示す。

図18(a)には、デバッガ14が通常プログラム12にアタッチし、通常プログラム12をデバッグしているときの表示手段の画面構成であるGUI150aを示す。また、図18(b)には、通常プログラム12が保護プログラム8を実行し、通常プログラム12と保護プログラム8をデバッグしているときの表示手段の画面構成であるGUI150bを示す。

[0158] 図18(a)に示すように、GUI150aは、コード表示部151、レジスタ表示部152、メモリ表示部153、シンボル表示部154、ウォッチポイント表示部155、コールスタック表示部156、ウィンドウタイトル表示部157、メニュー表示部158で構成される。

コード表示部151は、デバッグ対象のプログラムのコードを表示するための表示部で、ソースコードやアセンブラコード、マシン語が表示される。

[0159] レジスタ表示部152には、プログラムを実行しているプロセッサの各レジスタの値が表示される。

メモリ表示部153には、メモリの値が表示される。

シンボル表示部154は、デバッグ対象のプログラムにおいて、停止した関数内で使用されているローカル変数のシンボル名とその値を表示する。

[0160] ウォッチポイント表示部155は、シンボルにより指定された変数とその値を表示する。

コールスタック表示部156は、デバッグ対象のプログラムにおいて、停止させられた関数が呼び出されるまでの呼び出し階層を表示する。

ウィンドウタイトル表示部157は、ウィンドウのタイトルを表示するための表示部で、デバッガやデバッグ対象のプログラムのプログラム名やプログラムの状態が表示される。

[0161] メニュー表示部158は、デバッガのメニューを表示する。メニューとしては、デバッグ対象のプログラムのオープンやアタッチ、デバッガの終了、デバッガの動作やプログラムの動作情報の表示方法を設定する設定画面の表示、デバッグ対象プログラムの

実行や中断、再開、ステップ実行等がある。

モード表示部159は、プログラムがブレークポイント等により停止した時に、プログラムが実行されていたモードを表示するための表示部であり、通常プログラム実行中に停止した場合には通常モードであることを示すために「通常モード」と表示し、保護プログラム実行中に停止した場合には保護モードであることを示すために「保護モード」と表示する。

[0162] 4．2 GUI150aの説明の補足

なお、実施の形態4においては、通常プログラム12と保護プログラム8のどちらでデバッグ例外が発生したかを表示するために、モード表示部に「通常モード」や「保護モード」と表示するとしているが、これに限られない。例えば、モードの表示を「通常モード」や「保護モード」の文字列に限定するものではなく、異なるモードであることが分かるような表示であれば、どのようなものでもよい。具体的な例を挙げると、アイコンなどであってもよい。さらに、表示箇所もモード表示部に限定するものではなく、ウィンドウ全体で表示してもよく、ウィンドウの色などで区別させるとしてもよい。また、モード表示用のウィンドウを別途設けるとしてもよい。

[0163] これらの表示部には、デバッガ14が通常プログラムにアタッチした直後や、通常プログラムがブレークポイント命令で停止し、デバッガ14によるデバッグが可能な状態の時に、実行を停止している通常プログラムの停止時点での各種状態や、デバッガ14の実行中にデバッガ14の実行状態が表示される。さらにレジスタやメモリ、変数等に設定されている値をユーザからの入力等により変更することも出来る。

[0164] 4．3 GUI150bの説明

図18(b)に示すように、GUI150bは、通常プログラム用デバッグウィンドウ160と保護プログラム用デバッグウィンドウ161から構成される。

通常プログラム用デバッグウィンドウ160は、通常プログラム12の各種情報を表示するためのウィンドウである。

[0165] 保護プログラム用デバッグウィンドウ161は、保護プログラムの各種情報を表示するためのウィンドウである。どちらのウィンドウ(通常プログラム用デバッグウィンドウ160、保護プログラム用デバッグウィンドウ161)もウィンドウ内には、図18(a)の各表示部(

コード表示部151、レジスタ表示部152、メモリ表示部153、シンボル表示部154、ウォッチポイント表示部155、コールスタック表示部156)が表示される構成になる。なお、図面が煩雑となるため、図18(b)では、これらの各表示部は載せていない。

[0166] この様な表示手段を持つデバッガを用いて、デバッガのユーザが通常プログラム12と保護プログラム8をデバッグするときには、デバッガの起動直後や、通常プログラムにアタッチした直後の状態では図18(a)に示すような画面構成になっている。

その後、通常プログラム12が保護プログラム8を実行し、保護プログラム8の実行中にブレークポイントによるデバッグ例外が発生した場合には、保護プログラムの各種情報を表示するために、図18(a)の画面を2つに分割し、図18(b)に示すような画面構成に変更する。

[0167] このとき、保護プログラム8のデバッグであることをユーザに知らせるために、ポップアップ表示を行い、さらにビープ音などの特定の音やユーザにより設定された音を鳴らし、モード表示部159に「保護モード」と表示することで、ユーザに保護モードであることを意識させる。

4.4 実施の形態4の補足

なお、実施の形態4において、保護プログラム8デバッグ時にデバッガのウィンドウが2つに分割されるとしたが、分割されることに限定するものではない。例えば、保護プログラムのデバッグ用に新しくウィンドウを生成してもよいし、タブやメニューにより通常プログラム用デバッグウィンドウ160と保護プログラム用デバッグウィンドウ161を切り替えられるようにしてもよい。

[0168] 保護プログラム8の実行中にデバッグ例外が発生し、デバッグ可能な状態になった場合には、保護プログラム用デバッグウィンドウ161にフォーカスが移り、保護プログラムに対してブレークポイントの設定やメモリ・レジスタ値の変更などが実施できる。さらに通常プログラム12に対しても同様の設定や変更の操作を行うことが可能である。

なお、実施の形態4において、保護プログラム8の実行中にデバッグ例外が発生した場合に保護プログラム8のデバッグと通常プログラム12のデバッグをデバッガのユーザが同時に行えるとしたが、同時にデバッグが行えることに限定するものではない。例えば、デバッグが行うことができる対象を保護プログラムのみに限定してもよく、通

常プログラムのデバッグウィンドウ160にアクセスできないように制限してもよい。

[0169] また、実施の形態1で述べたように、デバッガ14とデバッグ機能7を用いた保護プログラム8のデバッグは、デバッガ14によっては制限される。このため、保護プログラム8に対するデバッグ処理が許可されていない場合や、アクセス制御リストによりアクセスが許可されていない領域を表示しようとした場合など、各表示部に情報を表示できない場合がある。

[0170] この様に、表示できない情報を要求された場合には、表示できないことを、デバッガを使用しているユーザに知らせる必要がある。この様な表示できない情報があることを知らせる方法として、表示できない旨のポップアップ表示を行い、さらにビープ音などの特定の音やユーザにより設定された音を鳴らし、表示できない部分を「＊」など特定の文字列で表示する。

[0171] なお、上記では、表示できない部分を「＊」などの特定の文字列で表示するとしたが、特定の文字列で表示することに限定するものではなく、アイコンを表示してもよいし、何も表示しなくてもよい。また、表示できない部分の背景色を、表示できる部分の背景色と異なるようにしてもよい。

また、保護プログラム8をデバッグするときには、保護プログラム8の中の秘匿情報や、通常プログラム12と保護プログラム8との通信用の共有領域など、プログラムの開発や解析にあたって重要な情報や領域が存在する。このためデバッガ14では、この様な注意しなければならない情報や領域の表示方法を工夫することで、デバッガを使用しているユーザに注意を促すことが望ましい。

[0172] 秘匿情報の表示方法としては、デバッガ14が保護プログラム8に対するデバッグ処理の実行時に、保護プログラム8のデバッグ情報に含まれている秘匿情報領域に関する情報を取得し、各表示部に情報を表示する時に表示しようとしているコードやデータが秘匿情報領域に含まれているかどうかを判定する。

判定の結果、秘匿情報であった場合には、ユーザによる設定で秘匿情報を表示しないと設定されていた場合には、秘匿情報を表示せず、その部分が空白になった状態で表示される。

[0173] 一方、秘匿情報を表示すると設定されていた場合には、その秘匿情報を表示する

ときの文字の色をユーザにより設定された色(例えば赤色)にすることで、秘匿情報であることを強調する。さらに、情報を表示する時に秘匿情報を表示する旨のポップアップ表示を行い、ビープ音などの特定の音やユーザにより設定された音を鳴らす。この様に動作することで、ユーザに秘匿情報であると注意を促す。

[0174] なお、実施の形態4において、秘匿情報を表示しないと設定されていた場合にはその部分が空白になった状態で表示されるとしたが、空白の状態で表示することに限定するものではなく、秘匿情報が表示される部分を背景色と別の色で塗りつぶしてもよく、アイコンを用いてもよい。すなわち、ユーザから秘匿情報の内容が見えない状態に出来ればどのような表示方法でもよい。

[0175] 通常プログラム12と保護プログラム8とで共有する情報である共有情報の表示方法としては、デバッガ14が保護プログラム8に対するデバッグ処理の実行時に、保護OS6から通常モードと保護モードで情報を共有している領域に関する情報を取得し、各表示部に情報を表示する時に表示しようとしているコードやデータの領域が共有情報の領域に含まれているかどうかをチェックする。

[0176] 共有情報の領域であった場合には、その領域の情報を表示するときの文字の色をユーザにより設定された色(例えば黄色)にすることで、共有情報であることを強調する。さらに、情報を表示する時に共有情報を表示する旨のポップアップ表示を行い、ビープ音などの特定の音やユーザにより設定された音を鳴らす。この様に動作することで、ユーザに共有情報であると注意を促す。

[0177] なお、共有情報についても秘匿情報と同様に多様な表示方法を適用するとしてもよいが、共有情報と秘匿情報とが区別できるように秘匿情報とでは異なる表示方法を用いた方がよい。

なお、本実施の形態4において、秘匿情報や共有情報を表示するときの文字の色をユーザにより設定された色にするとしたが、文字の色を変更することに限定するものではなく、背景の色を変更してもよいし、太字や斜体など文字のスタイルを変更したり、下線や網掛けなど字飾りを付けてもよく、さらに、秘匿情報や共有情報全体を囲うなどしてもよい。

[0178] 5 実施の形態5

実施の形態5は、実施の形態1のデバッガに対して、実施の形態4とは異なり、プログラムの動作情報をキャラクタベースのユーザインタフェースに表示する表示手段を付加したものである。実施の形態5でも、デバッガ本体の機能は実施の形態1と同様であるので、説明を省略する。

[0179] 5．1 CUIの説明

図19は、本発明実施の形態5におけるキャラクタベースユーザインターフェース(CUI170)の表示方法の説明図である。

デバッガ14は、起動されると、コンソールと異なる、例えば「(dbg)」の様なプロンプトを表示し、デバッガ14が使用できる状態にあることを示す。この状態は、通常プログラム12がデバッグ可能であることを示している(デバッグ処理結果の表示例171a)。

[0180] 通常プログラム12が実行され、通常プログラム12により保護プログラム8が実行されると、実施の形態1で述べた動作に従い、一番最初の実行時に保護プログラムのエントリポイントに設定されたブレークポイントでデバッグ例外が発生される。

この時デバッガ14は、保護プログラム8がデバッグ可能であることを示すために、通常プログラム12のデバッグ時とは異なる、例えば「(dbg－sec)」の様なプロンプトを表示することでユーザが区別できるようにする。なお、プロンプトと、デバッガ14からのメッセージとの表示は、ユーザが区別できるように表示する。例えば、図示するように、括弧を用いてプロンプトを表すこととしてもよい。

[0181] また、デバッガ14は、保護プログラム8の実行中に停止した場合には、プロンプトやデバッガ14からのメッセージ、ユーザから入力された文字のスタイルを斜体に変更することで、保護プログラム8の実行中に停止していることを表示し、ユーザが区別出来るようにする(デバッグ処理結果の表示例171b)。

上記の状態では、実行が停止したプログラムを対象としたデバッグが可能となるが、実施の形態5では通常プログラム12と保護プログラム8とが連係動作している状態を対象としたデバッグを行うので、通常プログラム12の実行中に保護プログラム8をデバッグしたり、逆に保護プログラム8の実行中に通常プログラム12をデバッグできるようにすることが望ましい。

[0182] 以下、このような場合に用いるインタフェースについて説明を行う。

まず、通常プログラム12の実行中にデバッグ例外が発生し、通常プログラム12がデバッグ可能な状態の時、通常プログラム12のデバッグではなく、保護プログラム8のデバッグをデバッガ14のユーザが行いたい場合がある。この場合は、通常プログラム12がデバッグ可能な状態で、「secure」というコマンドを入力することにより、データ処理装置1が保護モードに移行して保護プログラム8がデバッグ可能な状態になる。

[0183] このとき、プロンプトは、「(dbg)」から「(dbg－sec)」へ変更されるが、通常プログラム12の実行中に停止したという状態は変わらないので、字体は斜体などに変化しない(デバッグ処理結果の表示例171c)。

逆に、保護プログラム8の実行中にデバッグ例外が発生し、保護プログラムがデバッグ可能な状態の時、保護プログラムのデバッグではなく、通常プログラムのデバッグが行いたい場合には、「normal」というコマンドを入力することにより、通常プログラムがデバッグ可能な状態になる。

[0184] このとき、プロンプトは、「(dbg－sec)」から「(dbg)」へ変更されるが、保護プログラム8の実行中に停止したという状態は変わらないので、字体は斜体で表示されたままであり、通常の字体に戻ったりはしない(デバッグ処理結果の表示例171d)。

デバッガ14のユーザが保護プログラム8のデバッグをしようとした場合に、デバッグが許可されていなかったり、アクセスが禁止されている領域の情報を表示しようとした場合には、コマンドを入力した時に、「Access invalid」などを表示することで、ユーザにデバッグが許可されていないことやアクセスが禁止されていることを通知する。

[0185] 5．2 実施の形態5の補足

なお、実施の形態5において、通常プログラム12がデバッグ可能か保護プログラム8がデバッグ可能かを区別するために表示されるプロンプトを変更するとしたが、プロンプトを変更することに限定するものではない。

例えば、プロンプトとは別にユーザから要求された処理が終わるたびに現在の状態を示す文字を表示してもよい。また、プロンプトやデバッガからのメッセージ、ユーザから入力された文字のスタイルを太字や斜体に変更するとしてもよいし、文字の色や背景色を変更するとしてもよいし、下線や網掛けなど字飾りを付けてもよい。

[0186] さらに、通常プログラム12の実行中に停止したか保護プログラム実行中に停止した

かを区別するために文字のスタイルを斜体に変更するとしたが、文字のスタイルを斜体にすることに限定するものではない。

例えば、太字などの他のスタイルに変更するとしてもよいし、文字の色や背景色を変更するとしてもよいし、下線や網掛けなど字飾りを付けてもよい。さらに、行頭に通常プログラム12の実行中に停止したのか保護プログラム8の実行中に停止したのかが分かるような文字列を表示してもよい。

[0187] また、通常プログラム12か保護プログラム8が停止し、ユーザがコマンドを入力可能になる度や通常プログラム12のデバッグと保護プログラム8のデバッグが切り替わる度に、以前の出力をすべて消すとしてもよい。

なお、本実施形態では、コマンドやメッセージの一例を以下のようにしている。

メッセージ「Change to Secure mode.」は、データ処理装置1が保護モードに移行したことを示すものである。メッセージ「Change to Normal mode.」は、データ処理装置1が通常モードに移行したことを示すものである。メッセージは、ユーザの入力したコマンドに応じた処理をデータ処理装置1が行った後、その処理の結果を通知するものである。

[0188] なお、コマンド「secure」は、保護モードに移行するためのコマンドで、コマンド「normal」は、通常モードに移行するためのコマンドであるとする。

なお、デバッガに対して入力できるコマンドや、コマンドに応じて出力されるメッセージは、各デバッガに応じて異なるため、詳細な説明は省略する。

6 補足

なお、本発明を上記実施の形態に基づいて説明してきたが、本発明は、上記の実施の形態に限定されないのはもちろんである。以下のような場合も本発明に含まれる。

(1)上述の実施の形態では、許可デバッガID情報52やアクセス制御リスト53は、保護プログラム8に含まれることとしたが、これに限らず、許可デバッガID情報52やアクセス制御リスト53を、データ処理装置の外部から取得することとしてもよい。この場合、取得にかかる許可デバッガID情報52等が、どの保護プログラム8と対応しているかを示す情報を併せて取得するとよい。

[0189] こうすることで、保護プログラム8に対するデバッグ処理実行の可否の判定を容易に行うことができる。

なお、許可デバッガID情報52等は、デバッグ可能なデバッガIDや、アクセス可能な領域を示すものであるため、解析されると、保護プログラム8の保護が図れなくなるため、不正アクセスによって暴露されないよう、データ処理装置において保護機構等の安全な領域で記憶する必要がある。

[0190] また、許可デバッガID情報52等の取得も、不正者の盗聴等により不正取得されないよう安全に行われる必要があることは言うまでもない。

(2)上記の実施の形態では、保護プログラム8や通常プログラム12は保護OS6や通常OS11上で動作するものとしたが、これに限られるものではない。例えば、OSを介さずに直接動作するものとしてもよい。

[0191] この場合、デバッグ機能7や切替デバドラ13およびデバッガ用切替デバドラ15はLSI2の機能等として設けられ、各OSが行っていた割り込みの監視等の処理も行うこととなる。また、デバッガ14も直接的にLSI2上のCPU等で実行可能な言語で作成されたものとすること等が考えられる。

(3)上記の実施の形態では、デバッガ14、切替デバドラ13、デバッガ用切替デバドラ15、デバッグ機能7等はLSI2上で動作するソフトウェアとして実装されているものとしたが、これに限られるものでない。例えば、LSI2の機能として実現したり、LSI2と相互に通信しあうハードウェア等の形で実現してもよい。また、各構成要素の一部のみをLSI2の機能として実現したり、ハードウェアとして実現したりしてもよい。

(4)上記の実施の形態3では、複数の保護プログラム開発装置82を所有する組織等のために、保護プログラム開発装置82を介して送信される要求に応じて、同一のデバッガIDを複数の保護プログラム開発装置82に付与したりしてもよい。

(5)上記の実施の形態4および実施の形態5では、図18および図19を参照してユーザインタフェースの例を紹介したが、ユーザインタフェースの表示は、図18および図19に示されるようなものに限られない。

[0192] すなわち、インタフェースを構成する一部の構成要素を表示しないとしてもよいし、各構成要素の表示位置を適宜入れ替えたり、文字等の書体を別のものにしたりして

もよいことは言うまでもない。

(6)上記の実施の形態では、デバッグ機能7は、デバッガIDの確認によりデバッグが許可されていることを確認し、アクセス制御リストの確認により要求している領域へのアクセスが許可されている場合のみ、デバッグを行うことが許可されるとしたが、いずれか一方の確認のみでデバッグを許可されるとしてもよい。

[0193] また、確認の順序もデバッガIDの確認の後にアクセス制御リストの確認を行うという順序に限らず、どちらを先に確認してもよい。

(7)上記の実施の形態では、デバッガ14は、通常プログラム12にアタッチし、その通常プログラム12と連係動作する保護プログラム8に対するデバッグ処理を行うものとしたが、これに限られるものではない。例えば、保護プログラム8に直接アタッチできるデバッガであるとしてもよい。

[0194] この場合、デバッガ14はロードされていないプログラムにアタッチできないので、保護プログラム8は予めメモリ上にロードされ、保護プログラム8で発生したデバッグ例外はデバッガ用切替デバドラ15を経由してデバッガ14に通知される。

さらに、データ処理装置が保護モードで使用可能な入出力装置を有している場合には、デバッガは通常モードではなく保護モードで動作するようにしてもよい。この場合、デバッグ機能7とデバッガ14は、デバッガ用切替デバドラ15を経由することなく、直接通信する。

(8)上記の実施の形態では、前処理において停止フラグを有効にすることで、保護プログラム8等の実行開始時に必ずプログラムが停止するものとしているが、これに限られるものではない。

[0195] 例えば、変形例(7)のようにデバッガ14が直接保護プログラム8にアタッチしてブレークポイントを設定できる場合や、通常プログラム12のみをデバッグしたい場合には、保護プログラム8の実行開始時に必ずプログラムが停止すると不便である。そのため、前処理時に常に停止フラグを有効にするのではなく、ユーザの選択により有効にするか否かを決定できるとしてもよい。

(9)上記の実施の形態では、デバッガIDの生成の仕方について特に言及していないが、以下のようにすることが考えられる。

[0196] 例えば、デバッガID管理サーバ81が乱数を生成してデバッガIDとしてデバッガに割り当てることとしても良い。

また、この他には、例えば、保護プログラム8の一部または全体のハッシュ値をデバッガIDとして用いてもよい。

この場合、デバッガID判定部22は、比較値保持部33に、デバッガ14のデバッガIDを保持しておく。デバッガ14からデバッグが要求されると、デバッガID演算部32は、デバッグ要求の対象となっている保護プログラム8のハッシュ値を計算し、計算結果と、比較値保持部33が保持している値とをデバッガID比較部31が比較することにより判定を行う。

[0197] これにより、保護プログラム8の中身が変化している場合にはハッシュ値とデバッガIDとの比較が成功しないため、保護プログラム8をアップデートすると自動的に古いデバッガによるデバッグを禁止することが出来る。

なお、この場合、実施の形態3における保護プログラム開発装置82は、デバッガID管理サーバ81に対して、保護プログラム8のうち、ハッシュ値が取られる箇所が完成した時点でその保護プログラム8を送信する。デバッガID管理サーバ81は、保護プログラム8のハッシュ値を計算し、そのハッシュ値をデバッガIDとして保護プログラム開発装置82に返送する。返送されたデバッガIDは、保護プログラム8の開発者のみが知りうる情報である。したがって、デバッガの作者は、この保護プログラム8のデバッグに対応したデバッガを作成するには、かかるデバッガIDを保護プログラム8の開発者から通知してもらい、通知により取得したデバッガIDを、デバッガ14に割り当てる必要がある。ただし、上記の場合でも、デバッガID管理サーバ81の所有者とデバッガの作者とが同一である場合は、デバッガの作者はデバッガIDの通知を受けなくともデバッガを作成できる。なぜなら、デバッガの作者は自身が所有するデバッガID管理サーバ81からデバッガIDを知ることが出来るからである。

[0198] また、上述のようにハッシュ値を用いる場合、ハッシュ値を計算することにより、保護プログラム8に対するデバッグ処理が可能なデバッガのデバッガIDが算出されるので、保護プログラム8に許可デバッガID情報52を含ませておく必要がない。

さらに、デバッガIDには、プログラム開発者ごとの値とプログラムごとの値とが含まれ

ることとしてもよい。この場合、プログラム開発者ごとの値のチェックによりプログラム開発者単位でのデバッグ許可／不許可を確認し、プログラム開発者単位でのデバッグ許可がある場合にのみ、プログラム単位の値をチェックすることによりプログラム単位でのデバッグ許可を確認する等とすることで、より細かなデバッグ制御を実現できる。

(10) 上述の実施の形態では、保護プログラム8a実行中にブレークポイントによるデバッグ例外が発生したことを検出して保護プログラム8aに対するデバッグ処理を行う場合を説明した。これに限らず、保護プログラム8の実行中に一般的なエラー割り込みが発生した場合に、その割り込みが検出されることでデバッガ14によるデバッグ処理が行われるとしても良い。一般的なエラー割り込みとは、例えば、保護プログラム8実行中に0による除算が発生した場合や、オーバーフローが発生した場合などがある。一般的なエラー割り込みが発生した場合に、保護OS6がデバッグ例外の発生を切替デバドラ13に通知する等の処理を行うことにより(S402等)、デバッガ14は、保護プログラム8に対するデバッグ処理を行うことができる。

(11) 上述の実施の形態では、保護プログラム8に対するデバッグ処理の実行の可否を制御することとしたが、これに限らず、通常プログラム12についても、デバッグ処理を許可するデバッガのデバッガIDを割り当てて、通常プログラム12に対するデバッグ処理の実行の可否を制御することとしてもよい。

(12) 上述した構成要素の一部、または全部をLSI等の集積回路として実現してもよい。この場合の集積回路はLSI2と同一の集積回路であってもよいし、異なる集積回路であってもよい。

[0199] なお、LSIは集積度の違いにより、IC、システムLSI、スーパーLSI、ウルトラLSIと呼称されることもあるが、システムLSI2を上記のいずれの集積度で実現した場合も本発明に含まれることは言うまでもない。また、LSI製造後に、プログラムすることが可能なFPGA(Field Programmable Gate Array)や、LSI内部の回路セルの接続や設定を再構成可能なリコンフィギュラブル・プロセッサを利用しても良い。

[0200] さらには、半導体技術の進歩または派生する別技術によりLSIに置き換わる集積回路化の技術が登場すれば、当然、その技術を用いて構成要素の集積化を行ってもよい。バイオ技術の適用等が可能性としてありえる。

(13)本発明は、上記に示す方法であるとしてもよい。また、これらの方法をCPUの処理として実現するプログラムであるとしてもよいし、前記コンピュータプログラムからなるディジタル信号であるとしてもよい。

(14)また、本発明は、前記コンピュータプログラムまたは前記ディジタル信号をコンピュータ読み取り可能な記録媒体、例えば、フレキシブルディスク、ハードディスク、CD－ROM、MO、DVD、DVD－ROM、DVD－RAM、BD(Blu－rayDisc)、半導体メモリなどに記録したものとしてもよい。また、これらの記録媒体に記録されている前記ディジタル信号であるとしてもよい。

(15)これらの実施の形態および変形例の組合せであってもよい。

産業上の利用可能性

[0201] 本発明にかかるデータ処理装置は、デバッグ処理の実行の可否を制御することでプログラムの保護を図ることができ、特に、プログラムを保護しつつ開発を支援するための装置として有用である。

請求の範囲

[1] デバッグ処理部によるデバッグ処理の実行を制御するデータ処理装置であって、

前記デバッグ処理部を識別する識別子を取得する第1取得手段と、

不正アクセスから保護された状態にあるデバッグ対象プログラムの所定部分に含まれる検証値を取得する第2取得手段と、

前記デバッグ対象プログラムから取得された前記検証値と前記第1取得手段により取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定手段と、

許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御手段とを備える

ことを特徴とするデータ処理装置。

[2] 前記デバッグ対象プログラムの前記所定部分には、前記デバッグ対象プログラムを構成する各部分についてアクセスが許可または不許可であることを示すアクセス制御リストが含まれ、

前記第2取得手段は、前記デバッグ対象プログラムの前記所定部分に含まれる前記アクセス制御リストを取得するアクセス制御リスト取得部を含み、

前記データ処理装置は、さらに、前記取得した前記アクセス制御リストに基づいて、前記デバッグ対象プログラムの一部分のアクセスが許可されているかを判定するアクセス判定手段を備え、

前記制御手段は、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていないと判定された第1の場合、前記一部分のデバッグ処理の実行を禁止し、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていると判定された第2の場合、前記一部分のデバッグ処理を前記デバッグ処理部に実行させる

ことを特徴とする請求項1記載のデータ処理装置。

[3] 前記アクセス制御リストに示される前記部分のそれぞれは、前記デバッグ対象プログラムをメモリにロードする場合におけるロード先メモリアドレスの前記部分それぞれに対応するアドレス範囲を示し、

前記アクセス制御リストには、前記部分それぞれに対応する前記アドレス範囲のそれぞれについて、アクセスの可否が対応づけられており、

前記アクセス判定手段は、前記デバッグ対象プログラムをメモリにロードする場合におけるロード先メモリアドレスの前記一部分に対応するアドレス範囲について、前記アクセス制御リストにおいて対応づけられているアクセスの可否を参照することにより前記判定を行う

ことを特徴とする請求項2記載のデータ処理装置。

[4] 前記アクセス制御リストに示される前記部分のそれぞれは、前記デバッグ対象プログラムに含まれるシンボルを示し、

前記アクセス制御リストには、前記シンボルのそれぞれについて、アクセスの可否が対応づけられており、

前記アクセス判定手段は、前記デバッグ対象プログラムの前記一部分に含まれるシンボルについて、前記アクセス制御リストにおいて対応づけられているアクセスの可否を参照することにより前記判定を行う

ことを特徴とする請求項2記載のデータ処理装置。

[5] 前記デバッグ対象プログラムの前記所定部分には、前記検証値が複数含まれ、

前記アクセス制御リストは、前記所定部分に1以上含まれ、前記アクセス制御リストのそれぞれは、前記検証値の少なくとも1つと対応づけられており、

前記判定手段は、前記検証値のそれぞれについて前記取得した識別子と比較して前記判定を行い、

前記アクセス判定手段は、前記判定手段により許可されていると判定された前記検証値に対応づけられている前記アクセス制御リストに基づいて前記判定を行う

ことを特徴とする請求項2記載のデータ処理装置。

[6] 前記データ処理装置は、さらに、

表示部を備え、

前記制御手段は、前記第1の場合、前記一部分のデバッグ処理が禁止された旨を示す表示を前記表示部に行わせ、前記第2の場合、前記デバッグ処理の結果を前記表示部に表示させる表示制御部を含む

ことを特徴とする請求項2記載のデータ処理装置。

[7] 前記判定手段は、

前記検証値と前記識別子とを比較し、前記検証値と前記識別子とが一致した場合に、前記デバッグ処理が許可されていると判定する

ことを特徴とする請求項1記載のデータ処理装置。

[8] 前記判定手段は、比較値を不正アクセスから保護された状態で記憶する比較値保持部を含み、

前記検証値と前記識別子とを演算子として用いた所定の演算を行い、その演算結果が、前記記憶している前記比較値と一致した場合に、前記デバッグ処理が許可されていると判定する

ことを特徴とする請求項1記載のデータ処理装置。

[9] 前記データ処理装置は、外部からの不正アクセスを防止する機構を備えたセキュアドメインを有し、

前記データ処理装置は、動作モードとして、通常モードとセキュアモードとを備え、前記通常モードと前記セキュアモードとを切り替えて動作し、前記セキュアモード時においてのみ前記セキュアドメインを用いて動作し、

前記データ処理装置は、さらに、前記通常モードと前記セキュアモードとを切り替える切替部を備え、

前記通常モードで動作するプログラムは、前記切替部を経由して所定の処理の要求を前記セキュアモードで動作するプログラムに通知することで前記セキュアモードで動作するプログラムにアクセス可能であり、

前記デバッグ対象プログラムは、前記セキュアドメインにおいて記憶され、

前記第2取得手段は、前記セキュアドメインにおいて前記デバッグ対象プログラムから前記検証値の前記取得を行い、

前記判定手段は、前記セキュアドメインにおいて前記判定を行う

ことを特徴とする請求項1記載のデータ処理装置。

[10] 前記デバッグ処理部は、前記セキュアドメインの外部にあって前記通常モードにおいて動作し、

前記データ処理装置は、さらに、前記セキュアモードにおいてデバッグ処理を実行するセキュアデバッガを備え、前記セキュアデバッガは、前記セキュアドメインに含まれ、

前記デバッグ処理部は、前記デバッグ対象プログラムのデバッグ処理要求を出力し、

前記制御手段は、前記デバッグ処理部が前記デバッグ処理要求を出力すると、前記判定手段に前記判定を行わせ、許可されていないと判定されたとき、前記デバッグ処理要求にかかる前記デバッグ対象プログラムに対する前記セキュアデバッガによるデバッグ処理を禁止する

ことを特徴とする請求項9記載のデータ処理装置。

[11] 前記デバッグ対象プログラムの前記所定部分には、前記デバッグ対象プログラムを構成する各部分についてアクセスが許可または不許可であることを示すアクセス制御リストが含まれ、

前記第2取得手段は、前記デバッグ対象プログラムの前記所定部分に含まれる前記アクセス制御リストを取得するアクセス制御リスト取得部を含み、

前記データ処理装置は、さらに、前記取得した前記アクセス制御リストに基づいて、前記デバッグ対象プログラムの前記一部分のアクセスが許可されているかを判定するアクセス判定手段を備え、

前記アクセス判定手段は、前記セキュアドメインにおいて前記判定を行い、

前記制御手段は、

前記デバッグ処理部が出力した前記デバッグ処理要求にかかる前記デバッグ対象プログラムについて、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていないと判定された場合、前記セキュアデバッガによる前記一部分の前記デバッグ処理の実行を禁止し、前記判定手段により許可されていると判定され、かつ、前記アクセス判定手段により許可されていると判定された場合、前記一部分のデバッグ処理を前記セキュアデバッガに実行させる

ことを特徴とする請求項10記載のデータ処理装置。

[12] 前記デバッグ処理部は、前記通常プログラム、および、前記通常プログラムと連係

する前記デバッグ対象プログラムに対してデバッグ処理を行う機能を有し、

前記デバッグ処理部による前記デバッグ対象プログラムのデバッグ処理は、前記デバッグ処理部がデバッグ処理要求を出力し、出力された前記デバッグ処理要求に対して前記セキュアデバッガにより行われたデバッグ処理の結果を前記切替部を介して前記デバッグ処理部が受け付けることにより行われる

ことを特徴とする請求項10記載のデータ処理装置。

[13] 前記デバッグ処理部は、自デバッグ処理部がアタッチされた通常プログラムを識別するプロセス識別子を出力し、

前記セキュアデバッガは、前記デバッグ処理部から出力された前記プロセス識別子に示される通常プログラムと連係動作するデバッグ対象プログラムのエントリポイントにある命令をブレーク命令へと変更する

ことを特徴とする請求項12記載のデータ処理装置。

[14] 前記データ処理装置は、前記セキュアモードにおいて前記デバッグ対象プログラムの実行中にデバッグ例外が検出されると、前記切替部を介して前記デバッグ処理部にデバッグ例外の発生を通知し、

前記デバッグ処理部は、前記切替部から前記デバッグ例外の発生の通知を受け付けると、デバッグ処理の実行結果を示すデバッグ情報の取得要求を出力し、

前記セキュアデバッガは、前記デバッグ対象プログラムについて前記判定手段が肯定的な判定をしているときに前記デバッグ情報の取得要求を受け付けると、前記デバッグ対象プログラムのデバッグ処理を実行してデバッグ情報を取得し、取得したデバッグ情報を、前記切替部を介して前記デバッグ処理部に出力する

ことを特徴とする請求項12記載のデータ処理装置。

[15] 前記データ処理装置は、さらに、

前記通常プログラムのデバッグ処理の結果を第1の表示領域に表示する第1の結果表示部と、

前記通常プログラムとの連係にかかるデバッグ対象プログラムのデバッグ処理の結果を、前記第1の表示領域とは異なる第2の表示領域に表示する第2の結果表示部とを備え、

前記第1および第2の結果表示部は、前記連係にかかるデバッグ対象プログラムと前記通常プログラムとが連係して動作しているとき、前記第1の表示領域および前記第2の表示領域に、前記デバッグ対象プログラムと前記通常プログラムのデバッグ処理の結果を表示する

ことを特徴とする請求項12記載のデータ処理装置。

[16] 前記通常モードにおいては、通常OSが動作し、

前記セキュアモードにおいては、保護OSが動作し、

前記通常プログラムは、前記通常OSが生成するプロセスとして前記通常モードにおいて動作し、

前記デバッグ処理部は、前記通常OSで動作するデバッガとして前記通常モードにおいて動作し、

前記デバッグ対象プログラムは、前記保護OSが生成するプロセスとして前記セキュアモードにおいて動作し、

前記セキュアデバッガは、前記保護OSの有する機能として実装されている

ことを特徴とする請求項12記載のデータ処理装置。

[17] 前記制御手段は、前記デバッグ処理部が前記デバッグ処理要求を出力すると、前記判定手段に前記判定を行わせ、許可されていないと判定されたとき、前記デバッグ処理の実行の禁止を示すデバッグ処理不可通知を前記デバッグ処理部へ出力する

ことを特徴とする請求項10記載のデータ処理装置。

[18] 秘匿すべき保護情報を含んだプログラムを取得するプログラム取得手段と、

取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成手段と、

前記プログラムについて前記検証値生成手段で生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成手段とを備える

ことを特徴とするプログラム生成装置。

[19] 前記プログラム生成装置は、さらに、

前記プログラムを構成する各部分についてアクセスが許可または不許可であること

を示すアクセス制御リストを取得するアクセス制御リスト取得部を含み、

前記保護プログラム生成手段は、前記取得したアクセス制御リストを、前記プログラムに付加するアクセス制御リスト付加部を含む

ことを特徴とする請求項18記載のプログラム生成装置。

[20] デバッグ処理部によるデバッグ処理の実行を制御するデータ処理方法であって、

前記デバッグ処理部を識別する識別子を取得する第1取得ステップと、

不正アクセスから保護された状態にあるデバッグ対象プログラムの所定部分に含まれる検証値を取得する第2取得ステップと、

前記デバッグ対象プログラムから取得された前記検証値と前記第1取得ステップにおいて取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定ステップと、

許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御ステップとを含む

ことを特徴とするデータ処理方法。

[21] デバッグ処理部によるデバッグ処理の実行の制御をデータ処理装置に行わせる、コンピュータ読み取り可能な制御プログラムであって、

前記デバッグ処理部を識別する識別子を取得する第1取得ステップと、

不正アクセスから保護された状態にあるデバッグ対象プログラムの所定部分に含まれる検証値を取得する第2取得ステップと、

前記デバッグ対象プログラムから取得された前記検証値と前記第1取得ステップにおいて取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定ステップと、

許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御ステップとを含む

ことを特徴とする制御プログラム。

[22] デバッグ処理部によるデバッグ処理の実行を制御するデータ処理装置において用いられる集積回路であって、

前記デバッグ処理部を識別する識別子を取得する第1取得部と、

不正アクセスから保護された状態にあるデバッグ対象プログラムの所定部分に含まれる検証値を取得する第2取得部と、

前記デバッグ対象プログラムから取得された前記検証値と前記第1取得手段により取得された前記識別子とを比較し、その比較結果に応じて、前記デバッグ対象プログラムに対するデバッグ処理が許可されているか判定する判定部と、

許可されていないと判定されたとき、前記デバッグ対象プログラムに対するデバッグ処理の実行を禁止する制御部とを含む

ことを特徴とする集積回路。

[23] 秘匿すべき保護情報を含んだプログラムを取得するプログラム取得ステップと、

取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成ステップと、

前記プログラムについて前記検証値生成ステップで生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成ステップとを含む

ことを特徴とするプログラム生成方法。

[24] プログラムを生成する処理をプログラム生成装置に行わせるための、コンピュータ読み取り可能な制御プログラムであって、

秘匿すべき保護情報を含んだプログラムを取得するプログラム取得ステップと、

取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成ステップと、

前記プログラムについて前記検証値生成ステップで生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成ステップとを含む

ことを特徴とする制御プログラム。

[25] プログラムを生成するプログラム生成装置において用いられる集積回路であって、

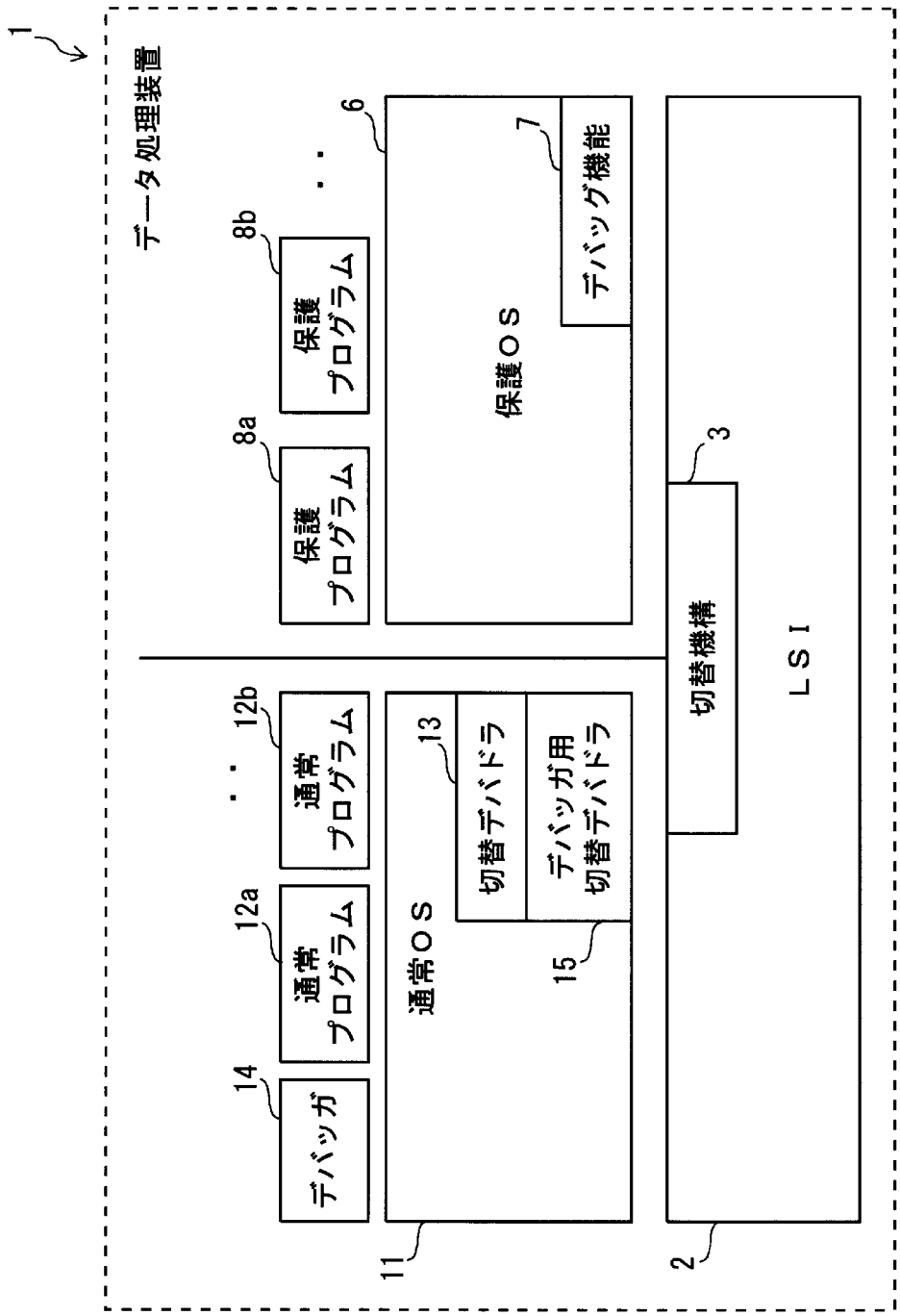
秘匿すべき保護情報を含んだプログラムを取得するプログラム取得部と、

取得した前記プログラムに対するデバッグ処理をデバッグ処理部の識別子に応じて許可するか否かを判定するための検証値を生成する検証値生成部と、

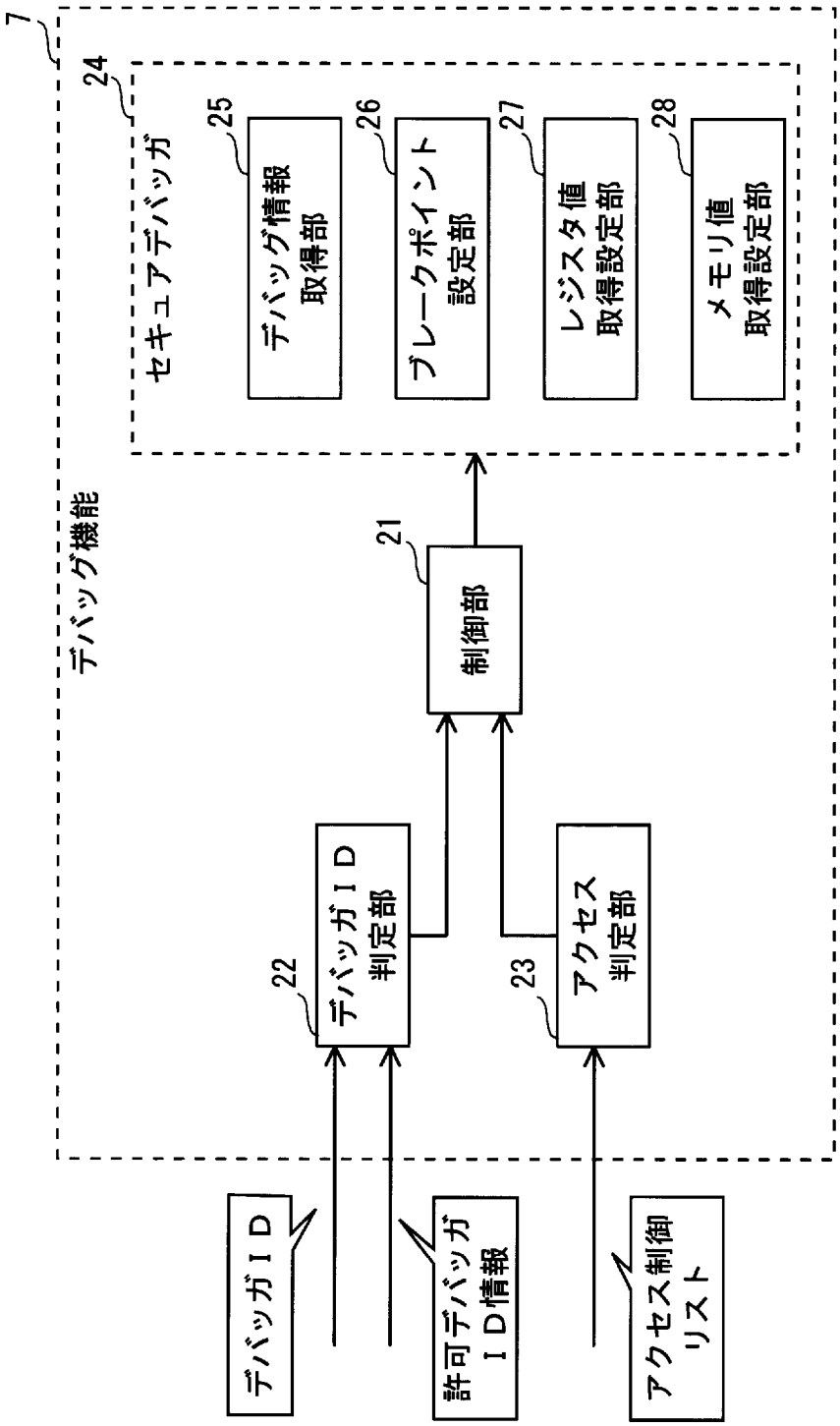
前記プログラムについて前記検証値生成手段で生成された検証値を前記プログラムに付加して保護プログラムを生成する保護プログラム生成部とを含む

ことを特徴とする集積回路。

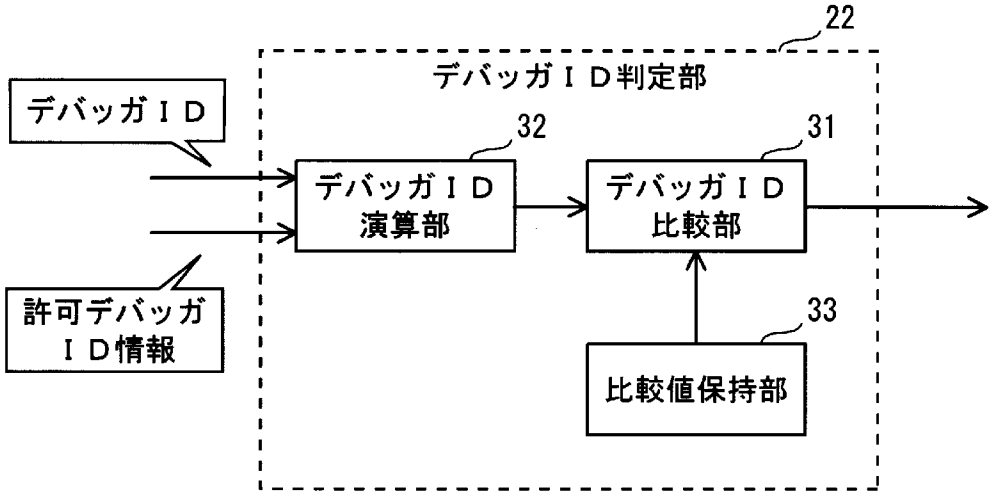
[図1]


1
データ処理装置
6
7
デバッグ機能
保護OS
8b
保護プログラム
8a
保護プログラム
3
切替機構
LSI
12b
通常プログラム
13
切替デバドラ
デバッガ用切替デバドラ
15
12a
通常プログラム
通常OS
14
デバッガ
11
2

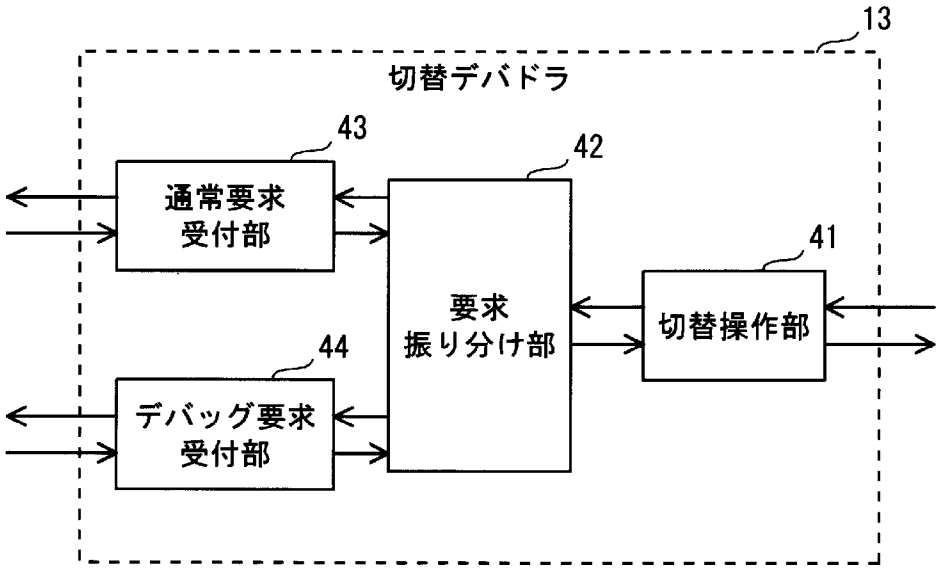
[図2]


7
デバッグ機能
24
セキュアデバッガ
25
デバッグ情報
取得部
26
ブレークポイント
設定部
27
レジスタ値
取得設定部
28
メモリ値
取得設定部
21
制御部
22
デバッガＩＤ
判定部
23
アクセス
判定部
デバッガＩＤ
許可デバッガ
ＩＤ情報
アクセス制御
リスト

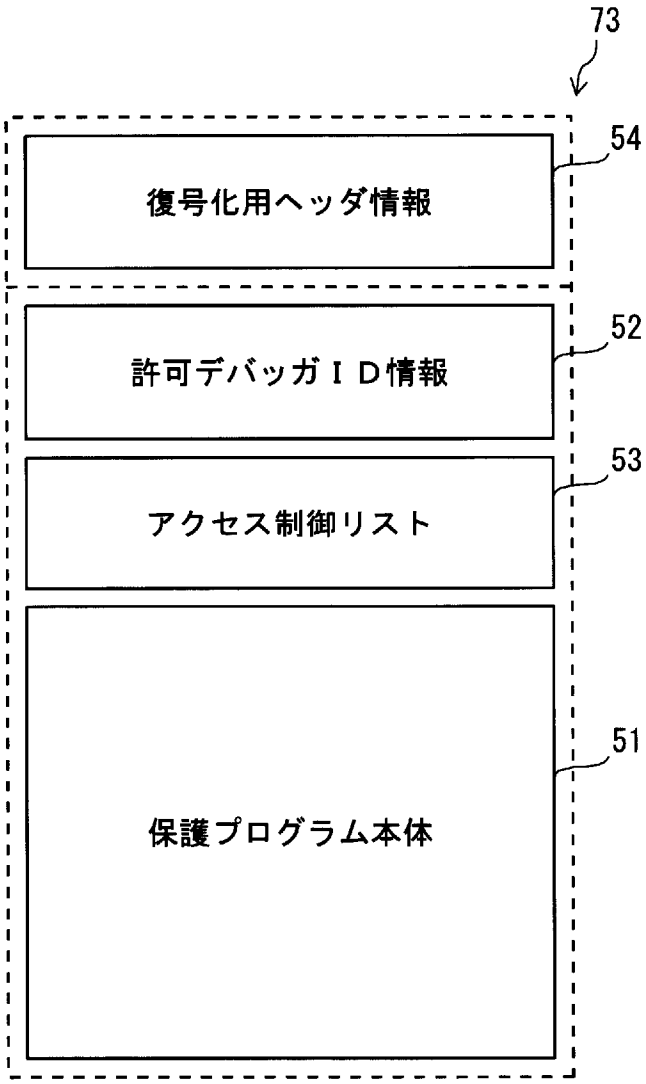
[図3]


22
デバッガＩＤ判定部
デバッガＩＤ
32
デバッガＩＤ
演算部
31
デバッガＩＤ
比較部
許可デバッガ
ＩＤ情報
33
比較値保持部

[図4]

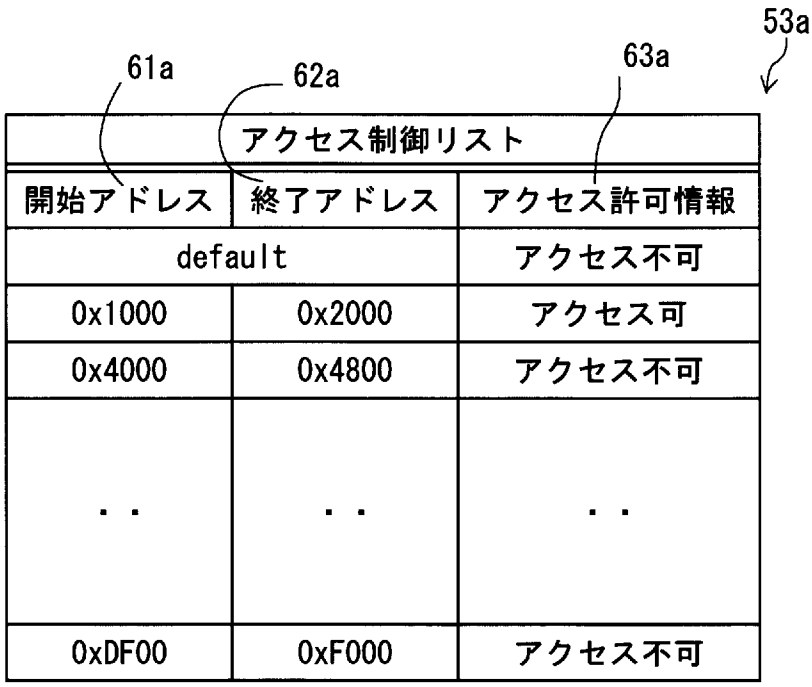

13
切替デバドラ
43
通常要求
受付部
42
要求
振り分け部
41
切替操作部
44
デバッグ要求
受付部

[図5]



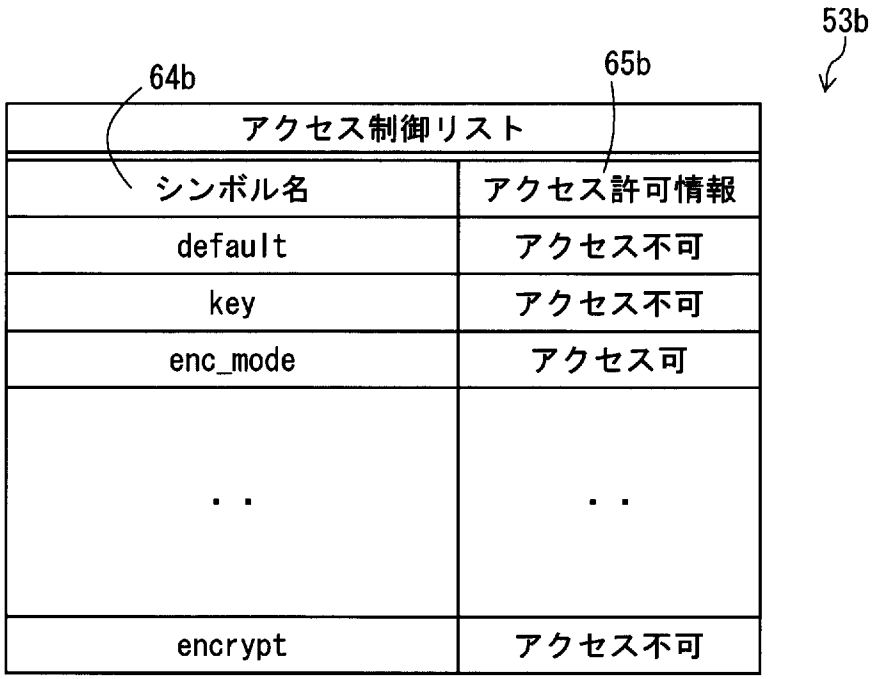
[図6]

(a)



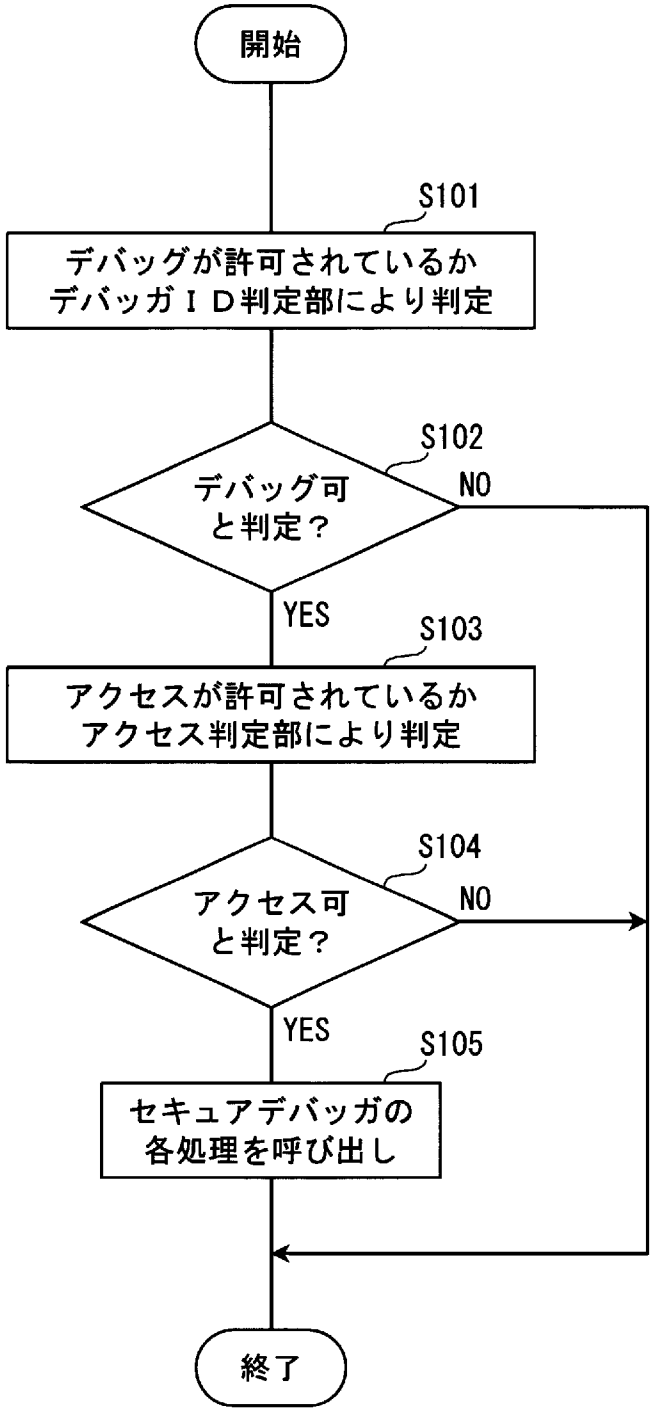
アクセス制御リスト		
開始アドレス	終了アドレス	アクセス許可情報
default		アクセス不可
0x1000	0x2000	アクセス可
0x4000	0x4800	アクセス不可
・・	・・	・・
0xDF00	0xF000	アクセス不可

(b)

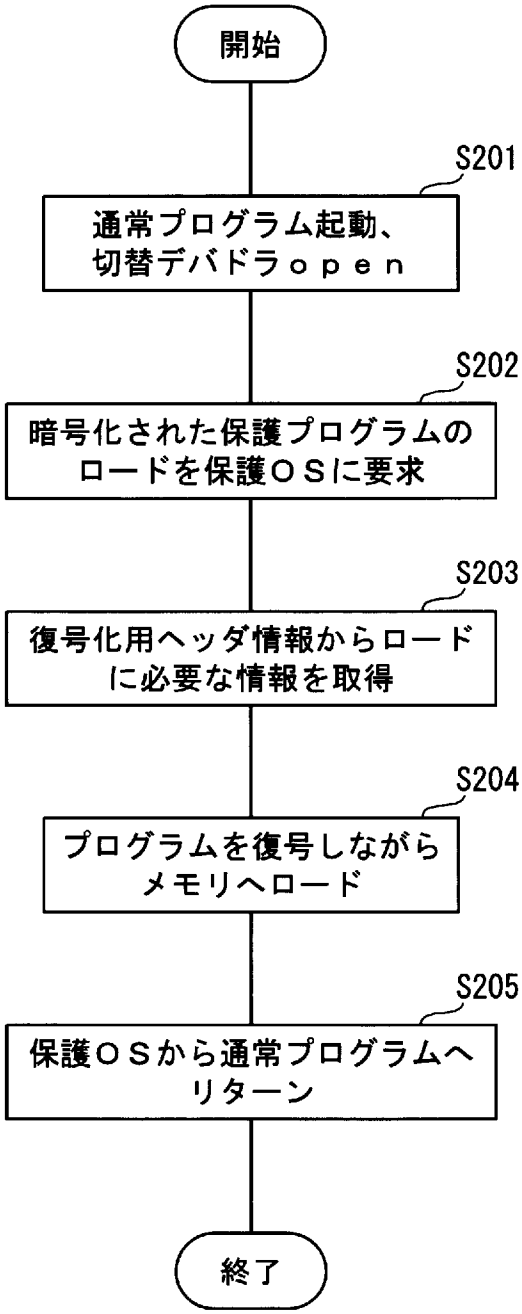


アクセス制御リスト	
シンボル名	アクセス許可情報
default	アクセス不可
key	アクセス不可
enc_mode	アクセス可
・・	・・
encrypt	アクセス不可

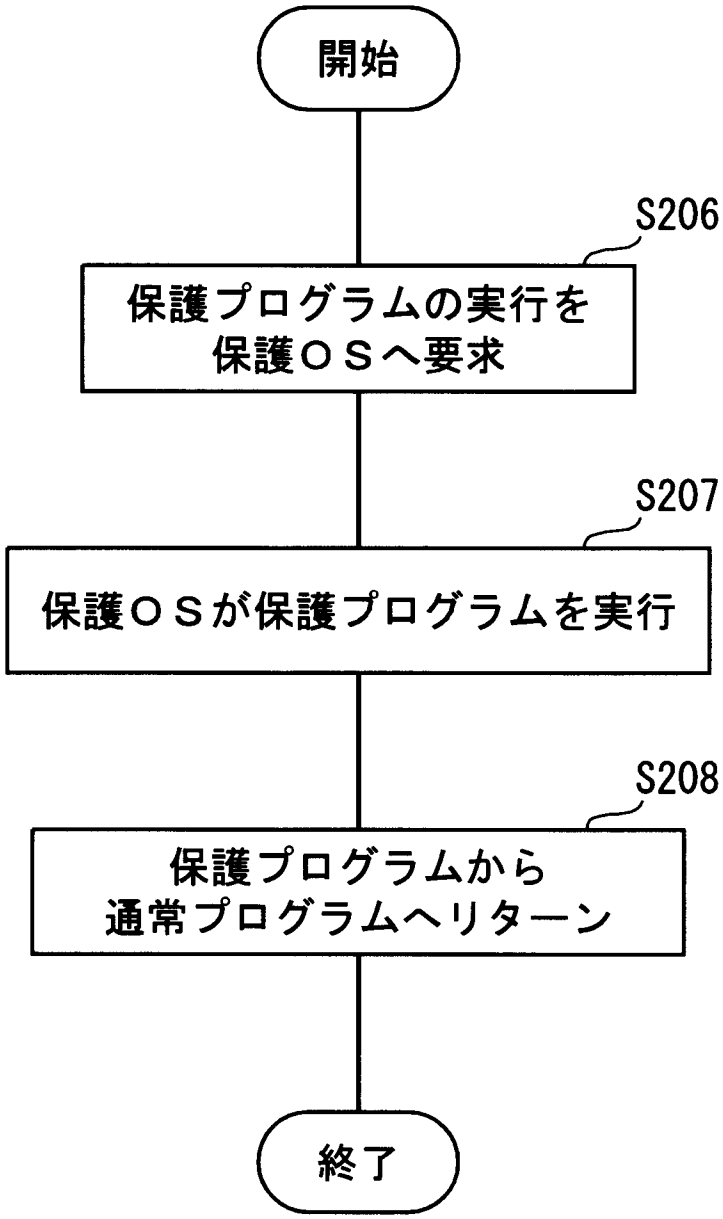
[図7]


開始
S101
デバッグが許可されているか
デバッガＩＤ判定部により判定
S102
デバッグ可
と判定？
NO
YES
S103
アクセスが許可されているか
アクセス判定部により判定
S104
アクセス可
と判定？
NO
YES
S105
セキュアデバッガの
各処理を呼び出し
終了

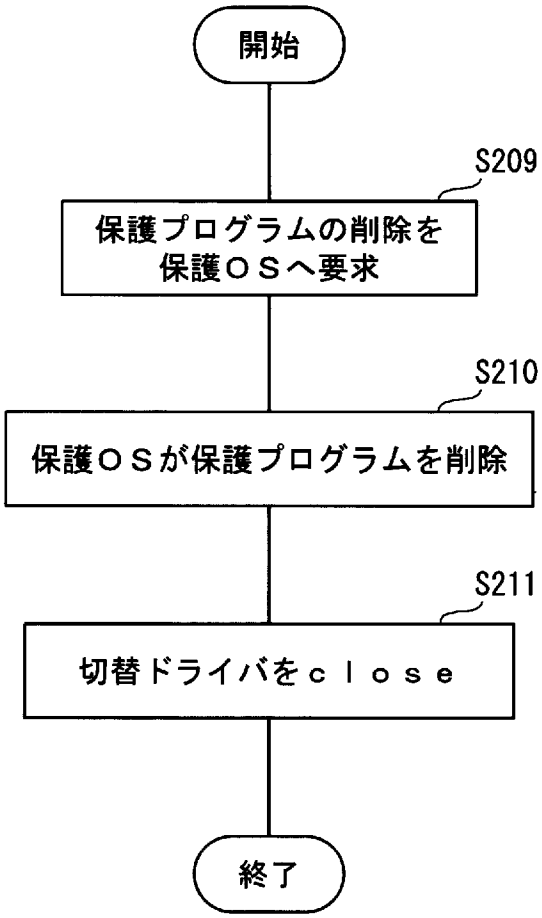
[図8]


開始
S201
通常プログラム起動、
切替デバドラｏｐｅｎ
S202
暗号化された保護プログラムの
ロードを保護ＯＳに要求
S203
復号化用ヘッダ情報からロード
に必要な情報を取得
S204
プログラムを復号しながら
メモリへロード
S205
保護ＯＳから通常プログラムへ
リターン
終了

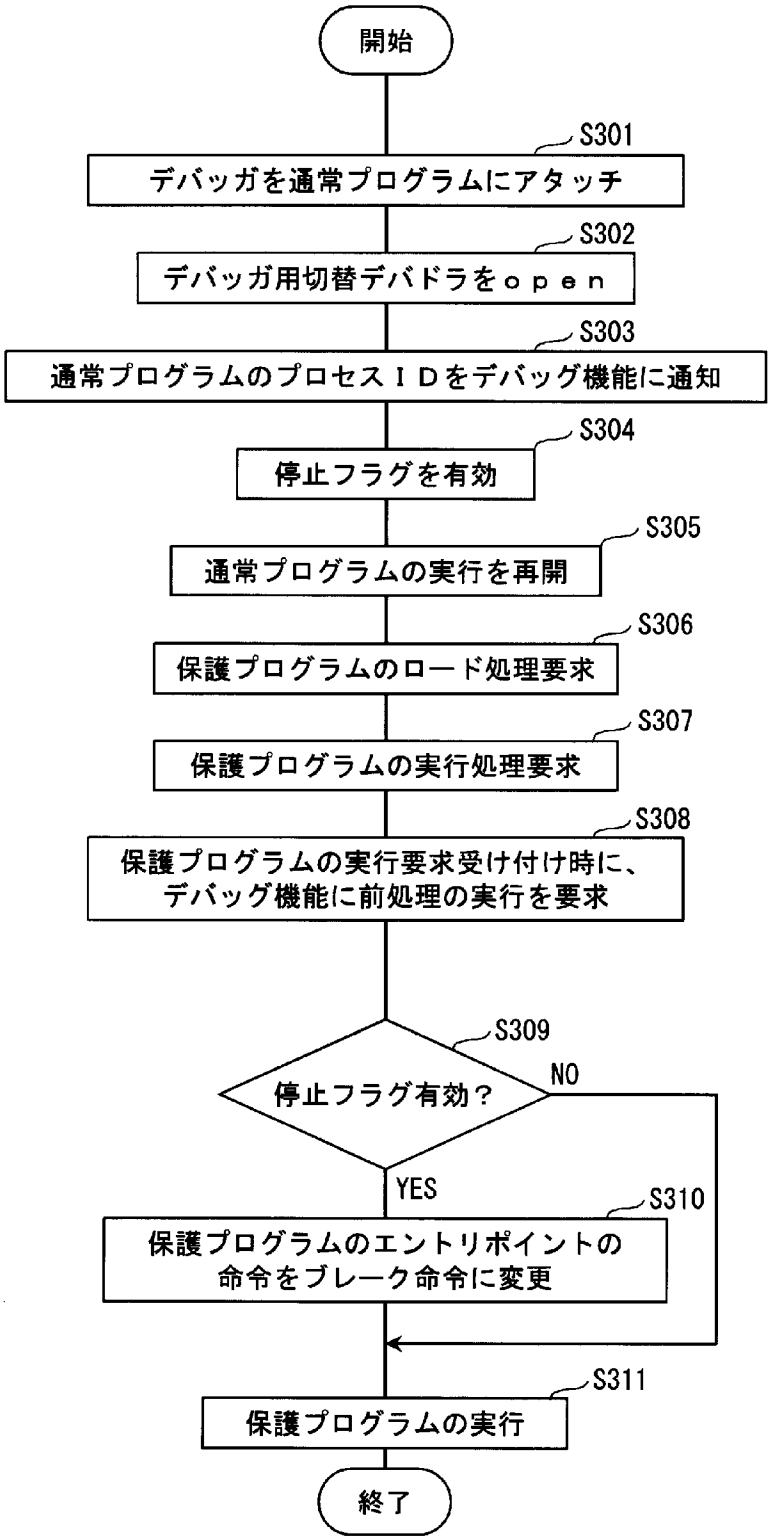
[図9]


開始
S206
保護プログラムの実行を
保護ＯＳへ要求
S207
保護ＯＳが保護プログラムを実行
S208
保護プログラムから
通常プログラムへリターン
終了

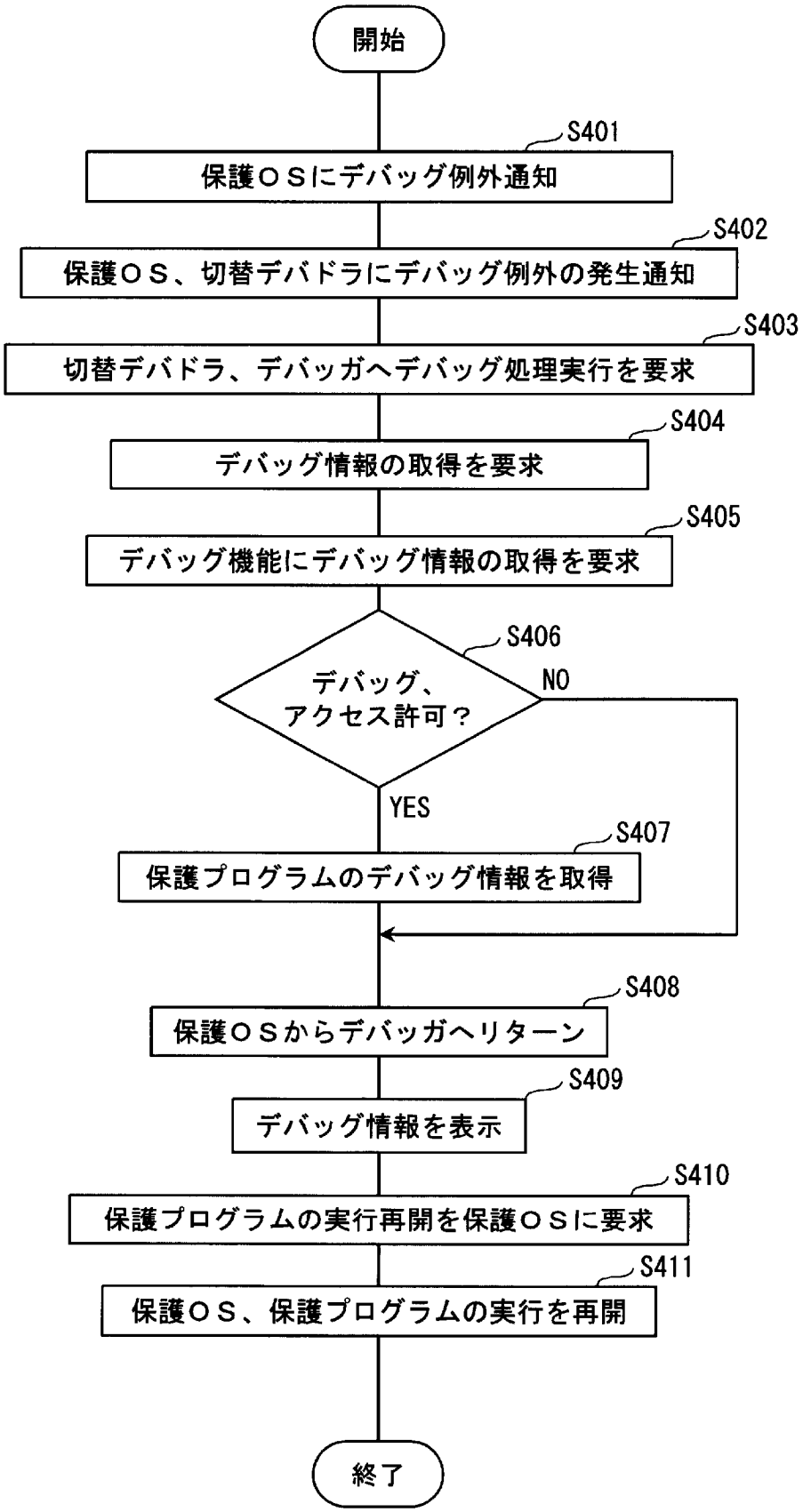
[図10]


開始
S209
保護プログラムの削除を
保護ＯＳへ要求
S210
保護ＯＳが保護プログラムを削除
S211
切替ドライバをｃｌｏｓｅ
終了

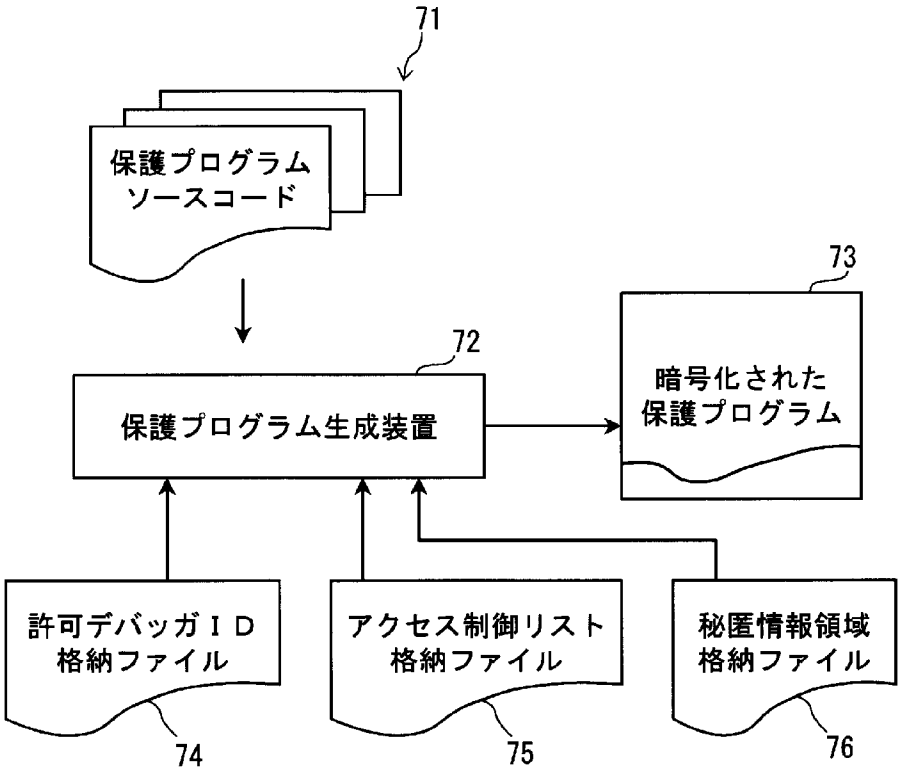
[図11]


開始
S301
デバッガを通常プログラムにアタッチ
S302
デバッガ用切替デバドラをｏｐｅｎ
S303
通常プログラムのプロセスＩＤをデバッグ機能に通知
S304
停止フラグを有効
S305
通常プログラムの実行を再開
S306
保護プログラムのロード処理要求
S307
保護プログラムの実行処理要求
S308
保護プログラムの実行要求受け付け時に、
デバッグ機能に前処理の実行を要求
S309
停止フラグ有効？
NO
YES
S310
保護プログラムのエントリポイントの
命令をブレーク命令に変更
S311
保護プログラムの実行
終了

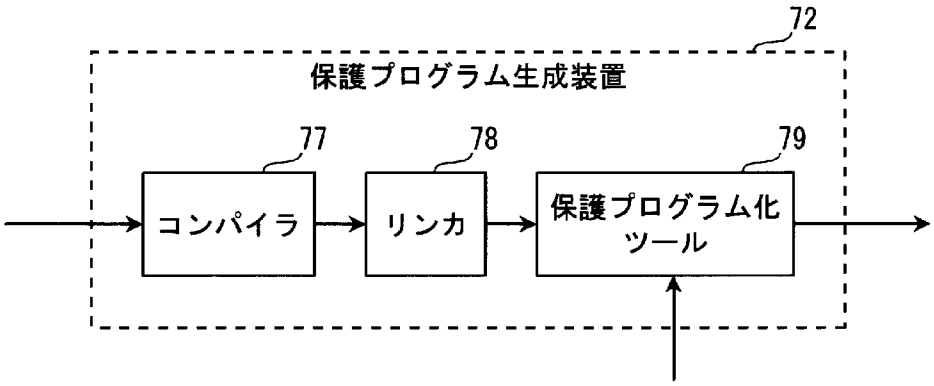
[図12]


開始
S401
保護ＯＳにデバッグ例外通知
S402
保護ＯＳ、切替デバドラにデバッグ例外の発生通知
S403
切替デバドラ、デバッガへデバッグ処理実行を要求
S404
デバッグ情報の取得を要求
S405
デバッグ機能にデバッグ情報の取得を要求
S406
デバッグ、
アクセス許可？
NO
YES
S407
保護プログラムのデバッグ情報を取得
S408
保護ＯＳからデバッガへリターン
S409
デバッグ情報を表示
S410
保護プログラムの実行再開を保護ＯＳに要求
S411
保護ＯＳ、保護プログラムの実行を再開
終了

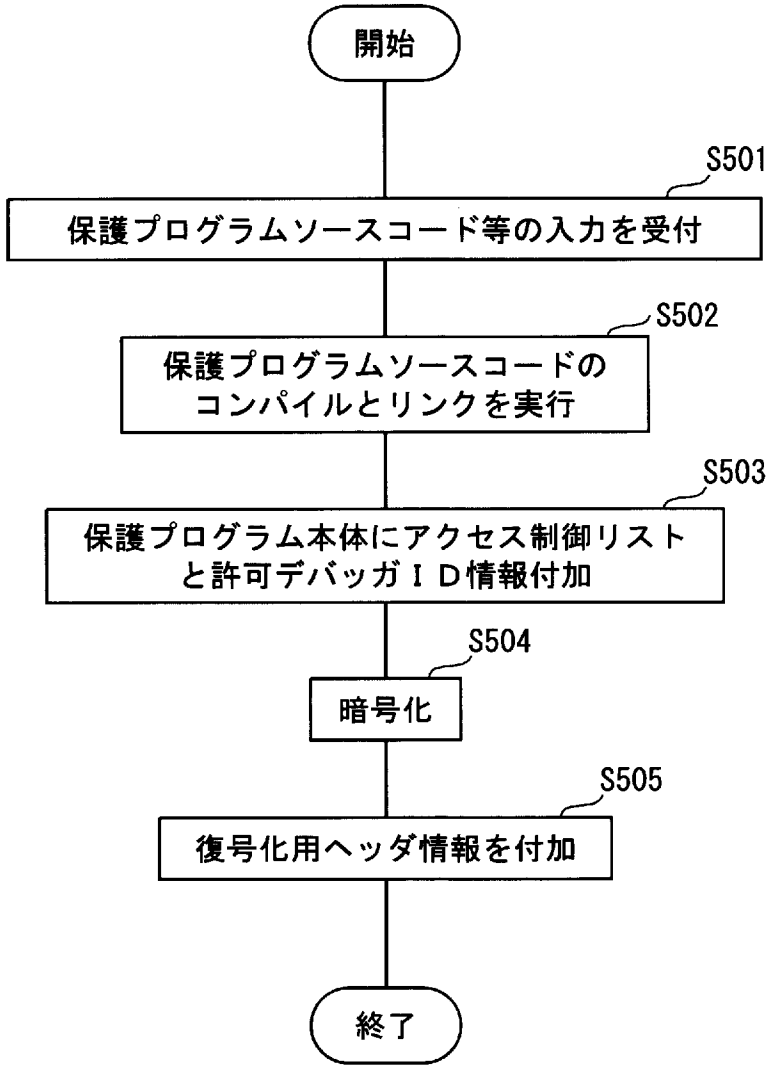
[図13]


71
保護プログラム
ソースコード
73
72
保護プログラム生成装置
暗号化された
保護プログラム
許可デバッガＩＤ
格納ファイル
アクセス制御リスト
格納ファイル
秘匿情報領域
格納ファイル
74
75
76

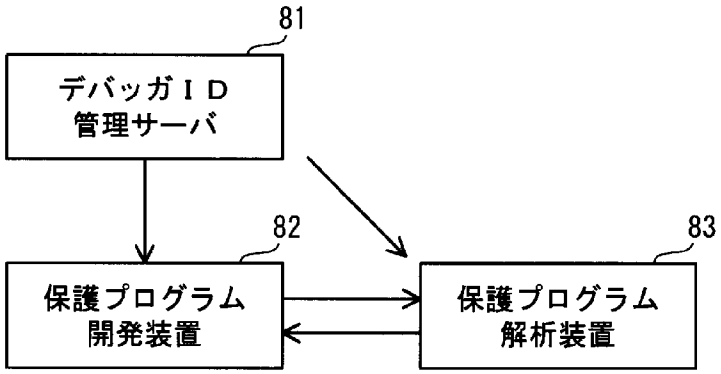
[図14]


72
保護プログラム生成装置
77
78
79
コンパイラ
リンカ
保護プログラム化
ツール

[図15]


開始
S501
保護プログラムソースコード等の入力を受付
S502
保護プログラムソースコードの
コンパイルとリンクを実行
S503
保護プログラム本体にアクセス制御リスト
と許可デバッガＩＤ情報付加
S504
暗号化
S505
復号化用ヘッダ情報を付加
終了

[図16]



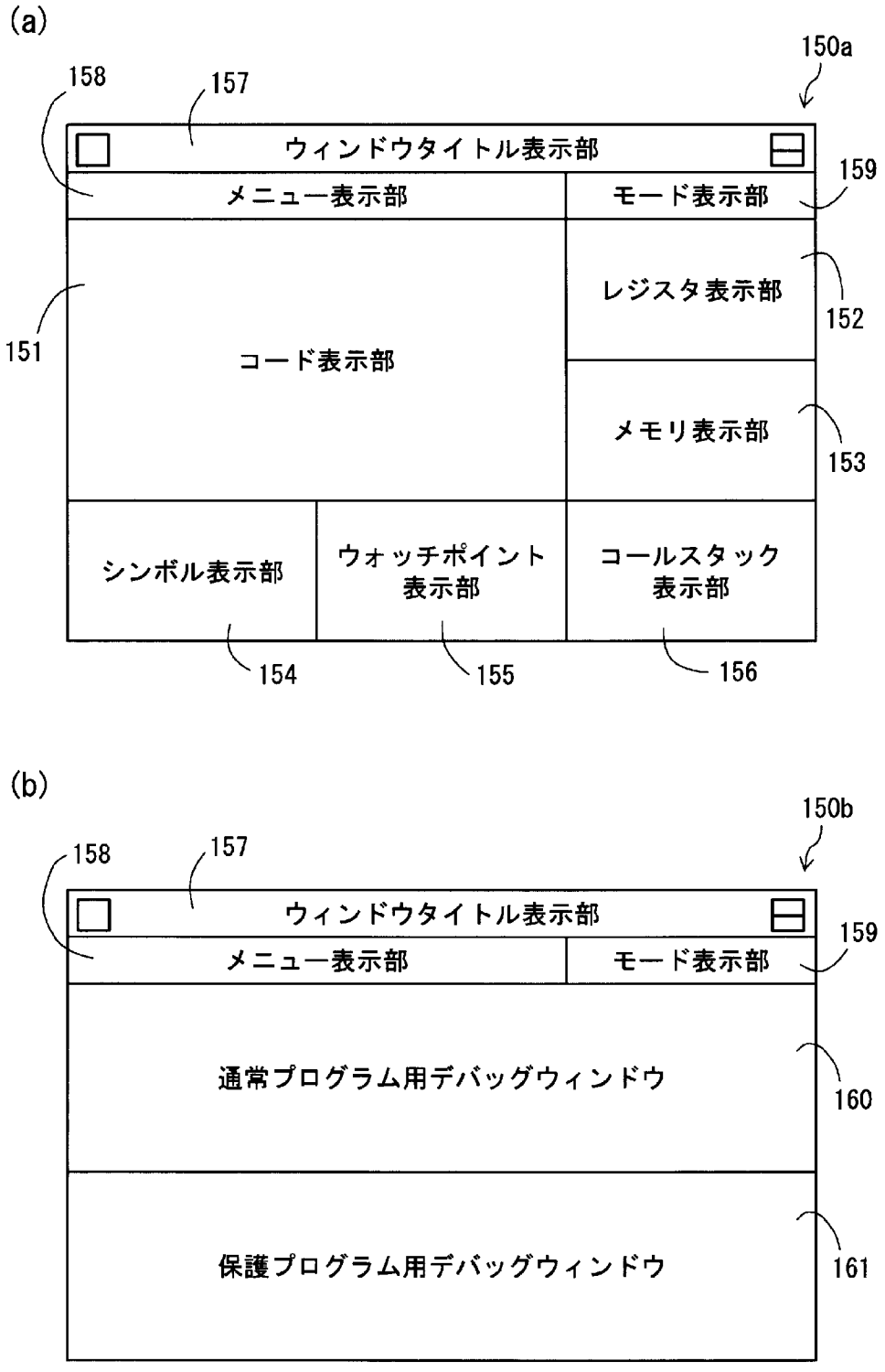
[図17]

90

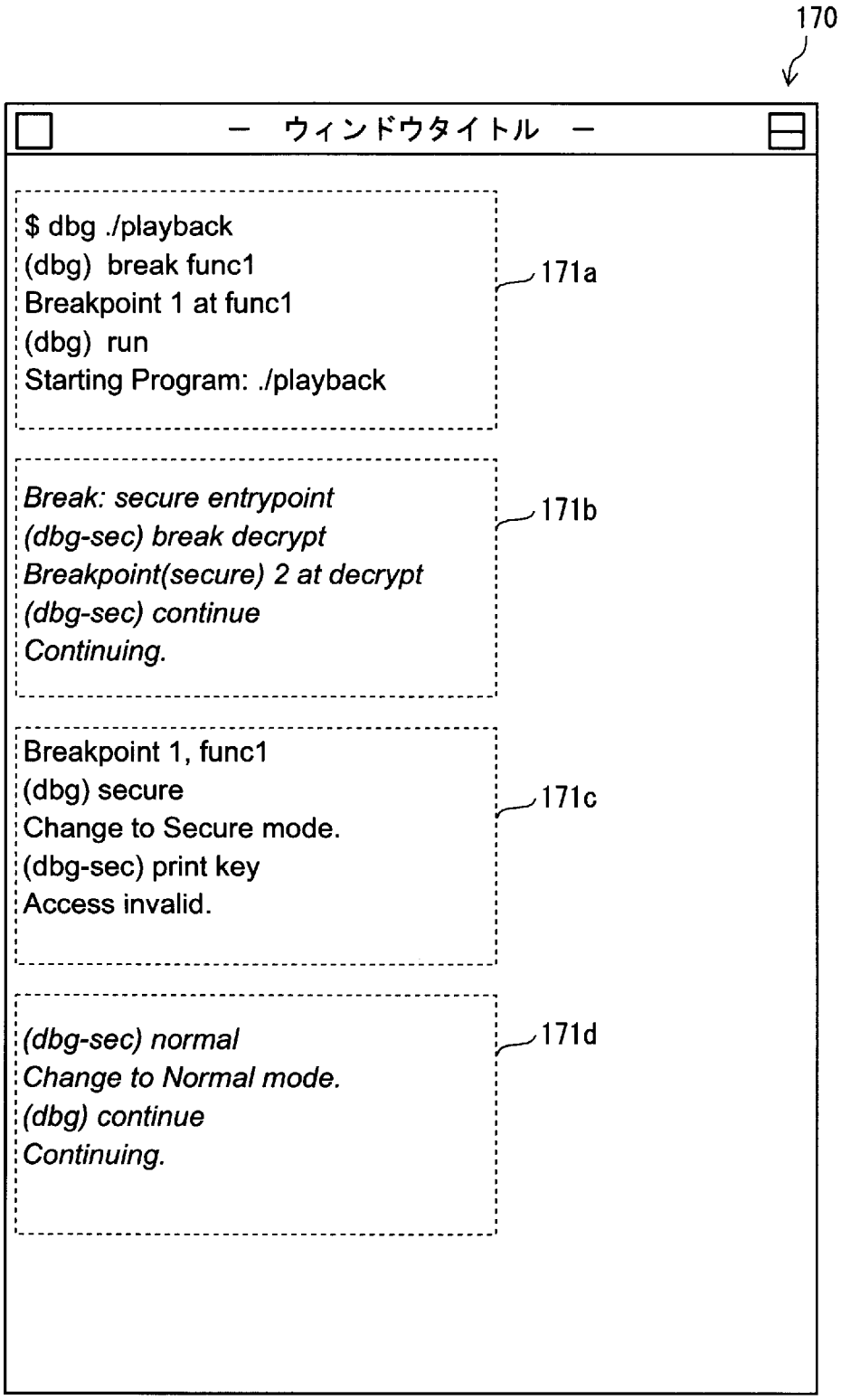
デバッガＩＤ管理ファイル

管理番号	デバッガＩＤ	保護プログラムの開発者名	連絡先
1	521473687	Ａ社	03-4567-89012
2	128793186	Ｂ社	06-7890-12345
・・	・・	・・	・・

[図18]


(a)
150a
158
157
ウィンドウタイトル表示部
メニュー表示部
モード表示部
159
レジスタ表示部
152
151
コード表示部
メモリ表示部
153
シンボル表示部
ウォッチポイント
表示部
コールスタック
表示部
154
155
156
(b)
150b
158
157
ウィンドウタイトル表示部
メニュー表示部
モード表示部
159
通常プログラム用デバッグウィンドウ
160
保護プログラム用デバッグウィンドウ
161

[図19]


170
ー ウィンドウタイトル ー
$ dbg ./playback
(dbg) break func1
Breakpoint 1 at func1
(dbg) run
Starting Program: ./playback
171a
Break: secure entrypoint
(dbg-sec) break decrypt
Breakpoint(secure) 2 at decrypt
(dbg-sec) continue
Continuing.
171b
Breakpoint 1, func1
(dbg) secure
Change to Secure mode.
(dbg-sec) print key
Access invalid.
171c
(dbg-sec) normal
Change to Normal mode.
(dbg) continue
Continuing.
171d

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2007/058838

A. CLASSIFICATION OF SUBJECT MATTER
G06F11/28(2006.01)i, *G06F21/22*(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F11/28, G06F21/22

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2007
Kokai Jitsuyo Shinan Koho	1971-2007	Toroku Jitsuyo Shinan Koho	1994-2007

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	WO 2005/091143 A1 (Matsushita Electric Industrial Co., Ltd.), 29 September, 2005 (29.09.05), Particularly, Par. Nos. [0022] to [0045] (Family: none)	1,6-10,17-25 2-5,11-16
A	JP 2004-509392 A (International Business Machines Corp.), 25 March, 2004 (25.03.04), Claims & EP 001368737 A	1-25

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search 12 July, 2007 (12.07.07)	Date of mailing of the international search report 24 July, 2007 (24.07.07)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００７／０５８８３８

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G06F11/28(2006.01)i, G06F21/22(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G06F11/28, G06F21/22

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００７年
日本国実用新案登録公報	１９９６－２００７年
日本国登録実用新案公報	１９９４－２００７年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X	WO 2005/091143 A1（松下電器産業株式会社）2005.09.29, 特に第２２－４５段落（ファミリ無し）	1,6-10,17-25
A		2-5,11-16
A	JP 2004-509392 A（インターナショナル・ビジネス・マシーンズ・コーポレーション）2004.03.25, 特許請求の範囲 & EP 001368737 A	1-25

☐　Ｃ欄の続きにも文献が列挙されている。

☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日 １２．０７．２００７	国際調査報告の発送日 ２４．０７．２００７
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員）　５Ｓ　３３５０ 永野　志保 電話番号　０３－３５８１－１１０１　内線　３５４６