

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-198001
(P2008-198001A)

(43) 公開日 平成20年8月28日(2008.8.28)

(51) Int.Cl. F I テーマコード (参考)
G06F 11/00 (2006.01) G06F 9/06 630B 5B176

審査請求 未請求 請求項の数 4 O L (全 16 頁)

(21) 出願番号 特願2007-33755 (P2007-33755)
(22) 出願日 平成19年2月14日 (2007.2.14)

(71) 出願人 000005108
株式会社日立製作所
東京都千代田区丸の内一丁目6番6号
(74) 代理人 110000442
特許業務法人 武和国際特許事務所
(72) 発明者 谷野 光宏
神奈川県横浜市戸塚区戸塚町5030番地
株式会社日立製作所ソフトウェア事業部
内
(72) 発明者 坂上 弘
神奈川県横浜市戸塚区戸塚町5030番地
株式会社日立製作所ソフトウェア事業部
内

最終頁に続く

(54) 【発明の名称】 修正パッチ選択支援システム及び支援方法

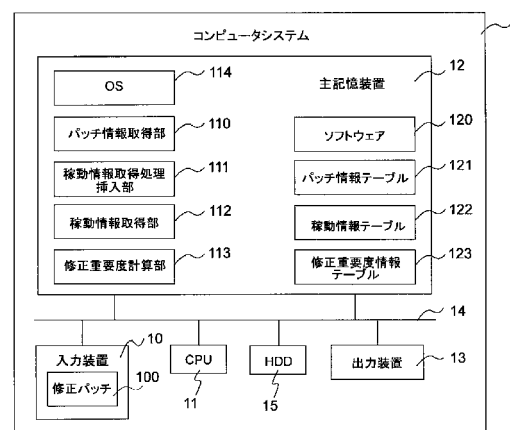
(57) 【要約】

【課題】 修正パッチの選択をソフトウェアや修正箇所の稼動状況に応じて支援する。

【解決手段】 ソフトウェア120に適用する修正パッチ100を入力装置10より入力し、修正箇所の修正要素名、修正位置、修正内容の組で表される修正情報を取得し、パッチ毎に修正要素名と修正位置との組による修正位置情報をパッチ情報テーブル121に登録するパッチ情報取得部110と、パッチ情報テーブル121からソフトウェア120の修正位置情報で示される挿入アドレスを計算してパッチ情報テーブル121に登録後、該当位置に稼動情報取得部112の呼び出し処理を挿入し、稼動情報の取得開始、終了を制御する稼動情報取得処理挿入部111と、稼動情報取得部112によって作成された稼動情報テーブル122と、パッチ情報テーブル121とから修正位置毎の稼働率と重要度を計算し、修正パッチ毎の重要度を示す修正重要度情報テーブル123を作成し、出力装置13に出力する。

【選択図】 図1

図1



【特許請求の範囲】**【請求項 1】**

コンピュータシステムにおいて利用されるソフトウェアに適用する修正パッチの選択を支援する修正パッチ選択支援システムにおいて、

入力された個々の修正パッチからパッチ名、修正対象のソフトウェアを構成する個々の要素の修正要素名及び修正位置からなる修正位置情報を取得し、パッチ情報を生成するパッチ情報取得手段と、

生成したパッチ情報からソフトウェアの修正位置情報で示される修正予定箇所に、稼働情報を取得する稼働情報取得手段を呼び出す処理を挿入し、稼働情報の取得の開始、終了を制御する稼働情報取得処理挿入手段と、

前記パッチ情報、稼働情報から修正位置、修正パッチ毎の稼働率と重要度とを計算し、その結果を出力する修正重要度計算手段とを前記コンピュータシステムに備えたことを特徴とする修正パッチ選択支援システム。

10

【請求項 2】

前記稼働情報取得手段は、前記ソフトウェアの稼働時に、前記修正予定箇所に挿入された稼働情報取得手段を呼び出す処理を契機に動作し、前記ソフトウェアの稼働情報を取得することを特徴とする請求項 1 記載の修正パッチ選択支援システム。

【請求項 3】

コンピュータシステムにおいて利用されるソフトウェアに適用する修正パッチの選択を支援する修正パッチ選択支援方法において、

20

パッチ情報取得手段と、稼働情報取得処理挿入手段と、稼働情報取得手段と、修正重要度計算手段とを前記コンピュータシステムに備え、

前記パッチ情報取得手段は、入力された個々の修正パッチからパッチ名、修正対象のソフトウェアを構成する個々の要素の修正要素名及び修正位置からなる修正位置情報を取得し、パッチ情報を生成し、

前記稼働情報取得処理挿入手段は、生成したパッチ情報からソフトウェアの修正位置情報で示される修正予定箇所に、稼働情報を取得する稼働情報取得手段を呼び出す処理を挿入し、稼働情報の取得の開始、終了を制御し、

前記修正重要度計算手段は、前記パッチ情報、稼働情報から修正位置、修正パッチ毎の稼働率と重要度とを計算し、その結果を出力することを特徴とする修正パッチ選択支援方法。

30

【請求項 4】

前記稼働情報取得手段は、前記ソフトウェアの稼働時に、前記修正予定箇所に挿入された稼働情報取得手段を呼び出す処理を契機に動作し、前記ソフトウェアの稼働情報を取得することを特徴とする請求項 3 記載の修正パッチ選択支援方法。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、修正パッチ選択支援システム及び支援方法に係り、特に、コンピュータシステムにおいて利用されるソフトウェアの予防保守を行う際に適用する修正パッチの選択を支援する修正パッチ選択支援システム及び支援方法に関する。

40

【背景技術】**【0002】**

コンピュータ上のソフトウェアを安定に稼働させるためには、ソフトウェアに対する修正パッチを適宜対策する必要がある。そして、ミッションクリティカルなコンピュータシステムに対する修正パッチの対策は、対策時のトラブルを防止するために、事前に適用するパッチを選択して行われることが多い。しかし、近年、コンピュータのシステム構成の多様化により、システムで利用されるソフトウェア数が増加し、修正パッチ量も増加の傾向にあるため、修正パッチの選択に伴う作業量が増大している。

【0003】

50

前述したような修正パッチの選択に伴う作業量の増大に対して、修正パッチ選択の支援を行う技術が、例えば、特許文献1等に記載されて知られている。この従来技術は、コンピュータシステムの構成情報をデータベースに登録し、その情報に基づいて対策する修正パッチを選択するための情報を提供することができるようにしたものである。また、他の従来技術として、修正パッチそれ自体に設定された重要度から修正パッチを選択するという方法も知られている。これらの方法は、いずれも、コンピュータシステムで利用されるソフトウェアの稼働状況と連動しない情報を用いて修正パッチの選択を行っているものである。

【特許文献1】特開2005-99967号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

前述した従来技術は、ソフトウェアの修正箇所の稼働状況に応じた情報を提供していないため、稼働率が低いソフトウェアやコンピュータシステムの使用状況では動作しない箇所のソフトウェアに対する修正パッチの対策を除外することができず、逆に、稼働率が高いが修正パッチに与えられた重要度が低い修正パッチが対策されない等、コンピュータシステムの稼働状況に応じた修正パッチの選択を支援することができないという問題点を有している。

【0005】

本発明の目的は、前述したような従来技術の問題点を解決し、ソフトウェアの修正箇所の稼働状況に連動したデータを取得して、コンピュータシステムの使用状況に基づいて最適に修正パッチの選択を行うことを支援することができるようにした修正パッチ選択支援システム及び支援方法を提供することにある。

【課題を解決するための手段】

【0006】

本発明によれば前記目的は、コンピュータシステムにおいて利用されるソフトウェアに適用する修正パッチの選択を支援する修正パッチ選択支援システムにおいて、入力された個々の修正パッチからパッチ名、修正対象のソフトウェアを構成する個々の要素の修正要素名及び修正位置からなる修正位置情報を取得し、パッチ情報を生成するパッチ情報取得手段と、生成したパッチ情報からソフトウェアの修正位置情報で示される修正予定箇所に、稼働情報を取得する稼働情報取得手段を呼び出す処理を挿入し、稼働情報の取得の開始、終了を制御する稼働情報取得処理挿入手段と、前記パッチ情報、稼働情報から修正位置、修正パッチ毎の稼働率と重要度とを計算し、その結果を出力する修正重要度計算手段とを前記コンピュータシステムに備えたことにより達成される。

【発明の効果】

【0007】

本発明によれば、コンピュータシステムの稼働情報から、修正箇所の稼働状況に基づいて修正パッチの重要度を自動的に計算することができるため、コンピュータシステムの利用状況に則した最適な修正パッチの選択を支援することができるので、ユーザは、ソフトウェアに対して最適な修正パッチを施すことが可能となる。

【発明を実施するための最良の形態】

【0008】

以下、本発明による修正パッチ選択支援システム及び支援方法の実施形態を図面により詳細に説明する。

【0009】

図1は本発明の一実施形態による修正パッチ選択支援システムが構築されるコンピュータシステムの構成を示すブロック図である。図示コンピュータシステムは、パッチされるソフトウェアを有して、そのソフトウェアを用いて処理を行うコンピュータシステムであってよい。

【0010】

10

20

30

40

50

図示コンピュータシステム 1 は、P C に代表される情報処理装置であり、入力装置 1 0 と、C P U 1 1 と、主記憶装置 1 2 と、出力装置 1 3 と、H D D 1 5 とを備え、これらがバス 1 4 を介して接続されて構成されている。そして、主記憶装置 1 2 には、O S 1 1 4、コンピュータシステム 1 上で動作しているソフトウェア 1 2 0 が格納されていると共に、本発明により設けられるパッチ情報取得部 1 1 0、稼働情報取得出力挿入部 1 1 1、稼働情報取得部 1 1 2、修正重要度計算部 1 1 3 等を実現するプログラム、及び、パッチ情報テーブル 1 2 1、稼働情報テーブル 1 2 2、修正重要度情報テーブル 1 2 3 が格納されている。前述したプログラムは、磁気ディスク等により構成される外部記憶装置としての H D D 1 5 に格納されていて、主記憶装置 1 2 にロードされ、O S 1 1 4 の下で実行されることにより、対応する各機能を実現する。また、前述した各テーブルは、前述の各機能部により作成される。

【0011】

前述したように構成される本発明の実施形態において、コンピュータシステム 1 の記憶装置 1 2 上で動作しているソフトウェア 1 2 0 に適用する修正パッチ 1 0 0 は、その修正パッチを選択する際に、入力装置 1 0 から入力される。修正パッチ 1 0 0 は、ソフトウェア 1 2 0 の開発者等から記録媒体あるいは図示しないネットワークを介して提供され、入力装置 1 0 から入力される。

【0012】

そして、パッチ情報取得部 1 1 0 は、入力された修正パッチ 1 0 0 から個々の修正個所の修正要素名、修正位置、及び、修正内容の組で表される修正情報を取得し、パッチ毎に得られた修正要素名と修正位置との組で表される修正位置情報をパッチ情報テーブル 1 2 1 に登録する。稼働情報取得処理挿入部 1 1 1 は、作成されたパッチ情報テーブル 1 2 1 内のソフトウェア 1 2 0 の修正位置情報から挿入アドレスを計算し、その挿入アドレスをパッチ情報テーブル 1 2 1 に登録後、該当位置に稼働情報取得部 1 1 2 の呼び出し処理を挿入し、また、稼働情報の取得開始、終了を制御する。稼働情報取得部 1 1 2 は、稼働情報テーブル 1 2 2 を作成し、修正位置毎の稼働情報を取得し、その稼働情報を稼働情報テーブル 1 2 2 に格納する。また、修正重要度計算部 1 1 3 は、稼働情報取得部 1 1 2 によって作成された修正位置毎の稼働データを記録する稼働情報テーブル 1 2 2 と、作成されたパッチ情報テーブル 1 2 1 との内容により修正位置毎の稼働率と重要度とを計算し、それに基づいて修正パッチ毎の重要度を示す修正重要度情報テーブル 1 2 3 を作成し、その内容を表示装置、印刷装置等による出力装置 1 3 に出力する。

【0013】

パッチ情報テーブル 1 2 1、稼働情報テーブル 1 2 2、修正重要度情報テーブル 1 2 3 は、主記憶装置 1 2 上に作成され、C P U 1 1 に制御されて O S 1 1 4 上で動作するパッチ情報取得部 1 1 0、稼働情報取得処理挿入部 1 1 1、稼働情報取得部 1 1 2、修正重要度計算部 1 1 3 によって作成、更新、または、参照される。

【0014】

図 2 はソフトウェア 1 2 0 と修正パッチ 1 0 0 との関係例について説明する図である。

【0015】

ソフトウェア 1 2 0 は、複数の要素の集合で構成され、複数の要素の個々の要素は、要素名を持つ。図示例において、ソフトウェア 1 2 0 は、要素 A 2 0 1、要素 B 2 0 2 の 2 つの要素により構成されている。

【0016】

修正パッチ 1 0 0 は、ソフトウェア 1 2 0 を構成する個々の要素の中で修正を行う要素の要素名を示す修正要素名と修正要素の先頭からの変位で表される修正位置とからなる修正位置情報とその位置に対策される修正内容とにより構成される。修正情報は、修正要素名、修正位置、修正内容の順に記録されている。修正位置は、修正要素名の次のデータに記録されている。修正要素名、修正位置を示す情報としてユニークな識別子が設定されている。図 2 に示す例では、修正要素名を示す識別子を T A G 1、修正位置を示す識別子を T A G 2 と表している。修正パッチ 1 0 0 には複数の修正情報が存在し得る。すなわち、

同一の修正要素に対する異なる修正情報が同一の修正パッチ、または、複数の修正パッチに含まれていることがある。図2に示す例の場合、修正パッチ100は、修正パッチA21、修正パッチB23の2つの修正パッチにより構成されている。

【0017】

修正パッチA21は、ソフトウェア120の要素A201に対する修正情報A1(210)、修正情報A2(218)と、ソフトウェア120の要素B202に対する修正情報B1(214)とが含まれている。また、修正情報A1(210)は、修正情報として、修正要素名211と修正位置212と修正内容213とを含んで構成される。修正情報A2(218)は、修正情報として修正要素名219と修正位置220と修正内容221とを含んで構成される。修正情報B1(214)は、修正情報として、修正要素名215と修正位置216と修正内容217とを含んで構成される。

10

【0018】

また、修正パッチB23に含まれるソフトウェア120の要素B202に対する修正情報B2(231)は、修正情報として修正要素名232と修正位置233と修正内容234とを含んで構成される。

【0019】

図2に示す例において、ソフトウェア120を構成する要素A201、要素B202の先頭アドレスX、Yは、各要素が格納されている主記憶装置12の先頭アドレスを表しており、このアドレスは、要素名を入力することによりOS114によって取得することができる。

20

【0020】

修正パッチA21の修正情報A1(210)は、修正要素名211が要素A201、修正位置が+1000番地となっているため、修正要素名A201の先頭アドレスX番地をOSから求め、+1000番地目のアドレス=X+1000番地に、修正内容A1(213)が対策される。同様に、修正情報A2(218)は、修正要素名A201の先頭アドレスX番地から+1100番地目のアドレス=X+1100番地に、修正内容A2(221)が対策される。同様に、要素B202に対しても、修正パッチA21の修正情報B1(214)から、修正要素名B202の先頭アドレスY番地から+100番地目のアドレス=Y+100番地に、修正内容B1(217)が対策される。同様に、修正パッチB23の修正情報B2(231)から、修正要素名B202の先頭アドレスY番地から+300番地目=Y+300番地に、修正内容B2(234)が対策される。

30

【0021】

図3はパッチ情報取得部110での処理動作を説明するフローチャート、図4は図3に示す処理によって生成されるパッチ情報テーブル121の構成例を示す図である。図3に示す処理は、パッチ情報取得部110が、図2で説明した修正パッチ100の構成に基づいて修正パッチ名を取得し、パッチ情報テーブル121に登録し、取得した個々の修正パッチの修正情報をパッチ情報テーブル121に登録し、また、修正パッチ100の修正位置毎の稼働情報を後述する稼働情報取得部112での処理で取得するために必要な修正位置情報を修正パッチから取得する処理である。

【0022】

まず、図4を参照してパッチ情報テーブル121の構成例について説明する。パッチ情報テーブル121は、修正パッチ100の名称を登録する修正パッチ名400と、そのパッチに含まれる修正要素名及び修正位置からなる修正位置情報401と、後述の処理で使用する挿入アドレス402と、稼働情報取得開始時刻403と、稼働情報取得終了時刻404とを1組のレコードとして構成される。修正位置情報401及び挿入アドレス402は、対応する修正パッチ毎に作成される。

40

【0023】

次に、図3に示すフローを参照してパッチ情報取得部110での処理動作を説明する。

【0024】

(1) 修正パッチ100は複数存在し得るため、パッチ情報取得部110は、図1に示す

50

入力装置 10 から該当する修正パッチ名を順に入力し、修正パッチ名の入力が完了したか否かを判定し、完了した場合、ここでの処理を終了する（ステップ 300、301）。

【0025】

(2) ステップ 301 の判定で、修正パッチ名の入力が完了していなかった場合、入力される修正パッチ名を、図 4 に示したパッチ情報テーブル 121 の修正パッチ名 400 の欄に登録する（ステップ 302）。

【0026】

(3) 次に、修正パッチに含まれる複数の修正位置情報（修正位置情報と修正内容とを含む）からなるパッチ情報を図 1 で示す入力装置 10 から入力し、修正パッチ情報の入力が完了したか否かを判定し、修正パッチ情報の入力が完了していた場合、ステップ 300 からの処理に戻って次の修正パッチの入力処理からの処理を繰り返す（ステップ 303、304）。

【0027】

(4) ステップ 304 の判定で、修正パッチの入力が完了していなかった場合、ステップ 303 の処理で入力された修正要素名に対応する識別子が含まれているか否かを判定し、入力データに識別子が含まれていなかった場合、ステップ 303 からの処理に戻って該当する修正パッチの入力処理からの処理を繰り返す（ステップ 305）。

【0028】

(5) ステップ 305 の判定で、入力された修正要素名に対応する識別子が含まれていることを検出した場合、入力データから修正要素名を取得し、次のデータを入力して、そのデータから修正位置を取得する（ステップ 306、307）。

【0029】

(6) 次に、パッチ情報テーブル 121 の修正パッチ名に対応する修正位置情報 401 の欄に、先に求めた修正要素名と修正位置情報とを登録し、以上の処理が完了した時点で、ステップ 303 からの処理に戻って該当する修正パッチの入力処理からの処理を繰り返す（ステップ 308）。

【0030】

前述したようなパッチ情報取得部 110 での処理により、図 4 に示したパッチ情報テーブル 121 の修正パッチ名 400 の欄、修正位置情報 401 の欄への登録が行われることになる。

【0031】

図 5 は稼動情報取得処理挿入部 111 での処理動作を説明するフローチャートであり、次に、これについて説明する。この処理は、図 3 に示すフローによる処理で作成したパッチ情報テーブル 121 の修正位置情報 401 に基づいて対象ソフトウェアの稼動情報を取得する処理を登録する挿入アドレスを計算し、その位置に稼動情報取得部 112 の呼び出し処理を挿入し、また、対象ソフトウェアの主記憶装置 12 への読み込みや実行開始、終了を制御することにより修正位置毎の稼動情報の取得を制御する処理である。

【0032】

(1) まず、稼動情報取得処理挿入部 111 は、対象のソフトウェア 120 が既に主記憶装置 12 上に存在するか否か、すなわち、対象のソフトウェア 120 が動作中であるか否かを判定する（ステップ 500）。

【0033】

(2) ステップ 500 の判定で、対象のソフトウェア 120 が記憶装置 12 上に存在していた場合、当該ソフトウェア 120 の実行を終了させ、主記憶装置 12 上からそのソフトウェア 120 を消去する（ステップ 501）。

【0034】

(3) ステップ 500 の判定で、対象のソフトウェア 120 が記憶装置 12 上に存在しなかった場合、あるいは、ステップ 501 の処理で、そのソフトウェア 120 を主記憶装置 12 から消去した後、対象のソフトウェア 120 を主記憶装置 12 に読み込む（ステップ 502）。

10

20

30

40

50

【 0 0 3 5 】

(4) 次に、後述の処理で作成する稼動情報テーブル 1 2 2 が既に存在するか否かを判定し、存在する場合、この稼動情報テーブル 1 2 2 の全ての情報を消去して初期化する（ステップ 5 0 3、5 0 4）。

【 0 0 3 6 】

(5) ステップ 5 0 3 の判定で、稼動情報テーブル 1 2 2 が存在しなかった場合、あるいは、ステップ 5 0 4 の処理で、当該テーブルを初期化した後、図 3 に示す処理で作成したパッチ情報テーブル 1 2 1 から修正位置情報 4 0 1 を順に入力し、入力が完了したか否かを判定する（ステップ 5 0 5、5 0 6）。

【 0 0 3 7 】

(6) ステップ 5 0 6 の判定で、修正位置情報 4 0 1 の全ての入力が完了していなかった場合、ステップ 5 0 5 の処理で入力された修正位置情報 4 0 1 から対応するソフトウェア 1 2 0 の修正箇所を示すアドレスである修正アドレスを計算するため、修正位置情報 4 0 1 内の修正要素名の先頭アドレスを OS 1 1 4 から求める（ステップ 5 0 7）。

【 0 0 3 8 】

(7) 次に、ステップ 5 0 7 の処理で求めた修正要素名の先頭アドレスの値に、修正位置を加算することにより挿入アドレスを決定し、その内容をパッチ情報テーブル 1 2 1 の挿入アドレスの欄 4 0 2 に登録する（ステップ 5 0 8）。

【 0 0 3 9 】

(8) ステップ 5 0 8 の処理で決定した挿入アドレスで示されるソフトウェア 1 2 0 の該当位置に稼動情報取得部 1 1 2 の呼び出し処理を挿入する。説明している本発明の実施形態では、この挿入処理は OS の機能を用いて実現しており、この機能は、修正アドレスで示す位置のソフトウェア 1 2 0 の処理を改変せずに処理を挿入することができる。以上の挿入処理が完了した時点で、ステップ 5 0 5 からの処理に戻って次の修正位置情報の入力処理からの処理を繰り返す（ステップ 5 0 9）。

【 0 0 4 0 】

(9) ステップ 5 0 6 の判定で、修正位置情報 4 0 1 の全ての入力が完了した場合、すなわち、全ての修正位置情報の入力と、稼動情報取得部の呼び出し処理の挿入とが完了した場合、パッチ情報テーブル 1 2 1 の稼動情報取得開始時間 4 0 3 の欄に現在の時刻を設定し、その後、対象ソフトウェア 1 2 0 を実行させ、その終了を待つ（ステップ 5 1 0 ~ 5 1 2）。

【 0 0 4 1 】

(10) ソフトウェア 1 2 0 が終了した場合、稼動情報取得処理挿入部 1 1 1 は、ステップ 5 1 2 でのソフトウェア実行完了待ちの処理を終了する。その後、パッチ情報テーブル 1 2 1 の稼動情報取得終了時刻 4 0 4 の欄に現在の時間を設定する。以上の処理を完了した段階で、稼動情報取得処理挿入部 1 1 1 は処理を終了する（ステップ 5 1 3）。

【 0 0 4 2 】

図 6 は稼動情報取得部 1 1 2 での処理動作を説明するフローチャート、図 7 は図 6 に示す稼動情報取得部 1 1 2 の処理によって生成される稼動情報テーブル 1 2 2 の構成例を示す図である。図 6 に示す処理は、対象ソフトウェア 1 2 0 に挿入した稼動情報取得部 1 1 2 の呼び出し処理を経由して起動される稼動情報取得部 1 1 2 が修正位置毎の稼動情報を、対象ソフトウェア 1 2 0 の稼動状況に基づいて取得する処理である。このため、稼動情報取得部 1 1 2 は、対象ソフトウェア 1 2 0 が処理を終了するまでの間に何回でも起動されることになる。説明する例では、稼動情報取得部 1 1 2 は、1 秒間に複数回起動されるものとしている。

【 0 0 4 3 】

まず、図 7 を参照して、稼動情報取得部 1 1 2 の処理によって生成される稼動情報テーブルの構成について説明する。稼動情報テーブル 1 2 2 は、図 7 に示すように、挿入アドレス 4 0 2 で表される修正位置毎の稼動情報として、最終実行時刻 7 0 0 と、稼働時間 7 0 1 との組を 1 つのレコードとして保持するテーブルである。最終実行時刻 7 0 0 は、挿

10

20

30

40

50

入アドレス402に対応する修正位置が最後に稼動した際の時刻を1秒単位に保持している。稼動時間701は、修正個所が動作した時間を1秒単位に保持している。この稼動時間701は、該当の修正位置に対応する稼動情報取得部112が毎秒1回以上呼び出された場合、1秒単位に加算される。

【0044】

次に、図6に示すフローを参照して稼動情報取得部112での処理動作を説明する。

【0045】

(1) 稼動情報取得処理挿入部111での図5により説明したステップ511のソフトウェアの実行開始処理によって対象ソフトウェア120が実行を開始すると、稼動情報取得部112の呼び出し処理を挿入した修正個所を通過する毎に稼動情報取得部112が動作し、稼動情報取得部112は、呼び出し時の呼び出し元アドレスを、ソフトウェアへの挿入アドレス402を設定する際に利用したOSの該当機能により求める。ここで求めた呼び出し元アドレスは、パッチ情報テーブル121、及び、稼動情報テーブル122に登録される挿入アドレス402と等価である(ステップ600)。

10

【0046】

(2) 次に、呼び出しアドレスと一致する稼動情報テーブル122を検索し、該当するレコードが存在するか否かを判定する。このとき、該当するテーブルが存在しなかった場合、稼動情報テーブル122を新たに作成し、作成した稼動情報テーブル122に新たなレコードを作成して、最終実行時刻700の欄に0を設定する(ステップ601~604)。

20

【0047】

(3) ステップ602の判定で、稼動情報テーブル122が存在した場合、あるいは、ステップ604の処理で当該テーブルを生成した場合、該当する稼動情報テーブル122から最終実行時刻700を求め、次に、稼動情報取得部112が呼び出された時刻を取得する(ステップ605、606)。

【0048】

(4) 次に、稼動情報取得部112が呼び出された時刻と、先に求めた最終実行時刻700との差の時間が1秒を越えているか否かを判定し、越えていなかった場合、ここでの処理を終了する(ステップ607)。

【0049】

(5) ステップ607の判定で、稼動情報取得部112が呼び出された時刻と、先に求めた最終実行時刻700との差の時間が1秒を越えていた場合、最終実行時刻700を呼び出された時刻に更新し、稼動時間701に1秒を加算して、ここでの処理を終了する。以上の処理がソフトウェア120の実行終了まで、稼動情報取得部112が呼び出される毎に繰り返される(ステップ608、609)。

30

【0050】

図8は修正重要度計算部113での処理動作を説明するフローチャート、図9は修正重要度計算部113の処理によって作成される修正重要度情報テーブル123の構成例を示す図、図10は修正重要度計算部113がパッチ毎の重要度を決定するために用いる修正重要度テーブル1000の構成例を示す図である。図8に示す処理は、パッチの重要度を設定するため、稼動情報テーブル122に格納されている修正パッチ毎の各修正個所の稼動時間から稼動率と重要度とを計算し、それらの重要度のうち最も高いものから該当する修正パッチの重要度900を設定する処理である。

40

【0051】

修正重要度計算部113がパッチ毎の重要度を決定するために用いる修正重要度テーブル1000には、図10に示すように、パッチにより修正される箇所の稼動率1001に対応した5段階の重要度900が設定されており、修正重要度計算部113は、この修正重要度テーブル1000の内容により、図9に示すような修正パッチ名400毎に重要度900を対応させたレコードを持つ修正重要度情報テーブル123を作成する。

【0052】

50

次に、図 8 に示すフローを参照して、修正重要度計算部 1 1 3 での処理動作を説明する。

【 0 0 5 3 】

(1) 修正重要度計算部 1 1 3 は、パッチ情報テーブル 1 2 1 を入力し、入力が完了したか否か判定し、完了していた場合、後述する修正重要度情報テーブル 1 2 3 の作成が終了したとして、修正重要度情報テーブル 1 2 3 を出力装置に出力して処理を終了する (ステップ 8 0 0、8 0 1、8 1 3)。

【 0 0 5 4 】

(2) ステップ 8 0 1 の判定で、パッチ情報テーブル 1 2 1 の入力が完了していなかった場合、入力したパッチ情報テーブル 1 2 1 から修正パッチ名を求め、求めたパッチの重要度の初期値として 1 を設定する (ステップ 8 0 2、8 0 3)。

10

【 0 0 5 5 】

(3) 次に、当該修正パッチ名で登録された挿入アドレス 4 0 2 をパッチ情報テーブル 1 2 1 から順に入力し、入力が完了したか否かを判定する (ステップ 8 0 4、8 0 5)。

【 0 0 5 6 】

(4) ステップ 8 0 5 の判定で、挿入アドレス 4 0 2 の入力を完了していた場合、重要度の算出が終了したのものとして、修正パッチ名と重要度とを修正重要度情報テーブル 1 2 3 に登録し、その後、ステップ 8 0 0 からの処理に戻って処理を繰り返す (ステップ 8 1 2)。

【 0 0 5 7 】

(5) ステップ 8 0 5 の判定で、挿入アドレス 4 0 2 の入力が完了していなかった場合、入力された挿入アドレス 4 0 2 と一致する稼働情報テーブル 1 2 2 の情報を入力し、入力が完了したか否かを判定し、稼働情報テーブル 1 2 2 の情報の入力が完了していなかった場合、次の挿入アドレス 4 0 2 を入力するステップ 8 0 4 からの処理に戻って処理を繰り返す (ステップ 8 0 6、8 0 7)。

20

【 0 0 5 8 】

(6) ステップ 8 0 7 の判定で、稼働情報テーブル 1 2 2 の情報が入力された場合、当該稼働情報テーブル 1 2 2 から登録された修正箇所の稼働時間 7 0 1 を取得し、取得した稼働時間 7 0 1 から修正箇所の稼働率を計算する。稼働率は、パッチ情報テーブル 1 2 1 の稼働情報取得終了時刻 4 0 4 に記録された時刻からパッチ情報テーブル 1 2 1 の稼働情報取得開始時刻 4 0 3 に記録された時刻を引いたソフトウェア 1 2 0 全体の実働時間を算出し、稼働時間 / 実働時間の計算を行うことにより算出する (ステップ 8 0 8、8 0 9)。

30

【 0 0 5 9 】

(7) 次に、算出した稼働率と修正重要度テーブル 1 0 0 0 に定義された稼働率 1 0 0 1 の範囲とを比較し、一致する重要度を求め、求めた重要度と修正パッチの重要度 9 0 0 とを比較し、修正パッチの重要度が小さい場合、修正パッチの重要度を先に計算した重要度に更新し、パッチ名に対応する挿入アドレス 4 0 2 を入力するステップ 8 0 4 からの処理に戻って処理を繰り返す (ステップ 8 1 0、8 1 1)。

【 0 0 6 0 】

前述した処理において、修正パッチの重要度 9 0 0 は、重要度 1 で初期化されているため、前述した処理での判定と重要度の更新とを繰り返すことにより、修正パッチに含まれる修正情報の中で最も稼働率が高い修正情報の重要度が登録される。対応する全ての挿入アドレス 4 0 2 に対する処理が完了した場合、前述で説明したように、ステップ 8 1 2 の処理で、設定した重要度と該当修正パッチ名とが修正重要度情報テーブル 1 2 3 に登録される。前述の処理で全ての修正パッチの重要度を修正重要度情報テーブル 1 2 3 に登録した時点で、修正重要度情報テーブル 1 2 3 の内容は、前述で説明したように、ステップ 8 1 3 の処理で、図 1 で説明した出力装置 1 3 に出力される。

40

【 0 0 6 1 】

次に、前述までに説明した本発明の実施形態での一連の処理をまとめて具体的に説明する。

50

【 0 0 6 2 】

ベンダから提供された修正パッチ 1 0 0 は、入力装置 1 0 を通してコンピュータシステム 1 に取り込まれる。ここで、修正パッチ 1 0 0 は、修正パッチ A 2 1、修正パッチ B 2 3 等の複数のパッチから構成されるパッチ群である。このような修正パッチが入力されると、まず、パッチ情報取得部 1 1 0 は、図 3 により説明したフローに示す処理に従い、最初の修正パッチ名として、修正パッチ A 2 1 を取り込む。ここで、パッチ情報取得部 1 1 0 は、修正パッチ名が入力されたか否かを判断し、今回は入力があるため次の処理へ進む。修正パッチ名として、修正パッチ A 2 1 が入力されると、パッチ情報テーブル 1 2 1 の修正パッチ名 4 0 0 の欄に修正パッチ A 2 1 が登録される。

【 0 0 6 3 】

次に、パッチ情報取得部 1 1 0 は、修正パッチ A 2 1 の中から修正要素名を表す識別子を検索する。ここでは、修正要素 2 1 1 に要素 A 2 0 1 を表す識別子が含まれているため、修正要素名として要素 A 2 0 1 が取得される。次に、修正パッチ A 2 1 から修正要素に続くデータを読み込むことにより、修正位置 (+ 1 0 0 0 番地) (2 1 2) が取得される。取得した情報を、パッチ情報テーブル 1 2 1 の修正パッチ A 2 1 に対応する修正位置情報 4 0 1 の欄に、要素 A 2 0 1、修正位置 (+ 1 0 0 0 番地) (2 1 2) として登録する。修正パッチ A 2 1 を続けて検索すると、識別子としての修正要素名として要素 B 2 1 5 が得られる。パッチから続けてデータを入力すると修正位置 (+ 1 0 0 番地) (2 1 6) が得られる。これにより取得した情報を、パッチ情報テーブル 1 2 1 の修正パッチ A 2 1 に対応する修正位置情報 4 0 1 の欄に、要素 B 2 0 2、修正位置 (+ 1 0 0 番地) (2 1 6) として登録する。以上の処理をパッチに識別子がある間、さらに全ての修正パッチ 1 0 0 について繰り返すことにより、パッチに含まれる修正位置情報をパッチ情報テーブル 1 2 1 に登録し、パッチ情報テーブル 1 2 1 が完成する。

【 0 0 6 4 】

パッチ情報テーブル 1 2 1 が完成すると、次に、稼動情報処理挿入部 1 1 1 により図 5 に示すフローの処理が実行される。

【 0 0 6 5 】

稼動情報取得処理挿入部 1 1 1 は、前述のようにソフトウェア 1 2 0 に対して、パッチ情報テーブル 1 2 1 に登録されたパッチ毎の修正情報で示される修正位置に、稼動情報を取得するための処理を挿入し、ソフトウェア 1 2 0 の実行を行う。

【 0 0 6 6 】

まず、対象ソフトウェア 1 2 0 が主記憶装置 1 2 中に存在して動作中であるか否かを判断する。ここでは主記憶装置 1 2 中にソフトウェア 1 2 0 が存在しなかったとして、対象ソフトウェア 1 2 0 の読み込み処理に進む。次に、それまでの処理で作成した稼動情報テーブル 1 2 2 が存在するか否かを判断する。ここでは存在しなかったとする。

【 0 0 6 7 】

以上の初期化処理が終わると、次に、稼動情報取得処理挿入部 1 1 1 は、パッチ情報テーブル 1 2 1 から修正パッチの修正位置情報 4 0 1 を順に読み込んでいく。まず、修正位置情報 4 0 1 を入力すると、修正要素名として要素 A 2 1 1、修正位置として + 1 0 0 0 番地 (2 1 2) の組で表される修正情報が入力される。ここで、OS の機能から修正要素名の先頭アドレス X を取得する。先頭アドレス X と修正位置 (+ 1 0 0 0 番地) (2 1 2) とから、挿入アドレス X + 1 0 0 0 番地が得られる。得られたアドレスに、OS の機能を用いて稼動情報取得部 1 1 2 の呼び出し処理を挿入する。

【 0 0 6 8 】

次に、稼動情報取得処理挿入部 1 1 1 は、パッチ情報テーブル 1 2 1 から新たな修正位置情報 4 0 1 を入力し、修正要素名として要素 B 2 1 5、修正位置として + 1 0 0 番地 (2 1 6) の組で表される修正情報が入力される。ここでも同様に、OS の機能から修正要素名の先頭アドレス Y を取得する。先頭アドレス Y と修正位置 (+ 1 0 0 番地) (2 1 6) とから、挿入アドレス Y + 1 0 0 番地が得られる。得られたアドレスに、OS の機能を用いて稼動情報取得部 1 1 2 の呼び出し処理を挿入する。

10

20

30

40

50

【 0 0 6 9 】

以上の処理をパッチ情報テーブル 1 2 1 の全ての修正位置情報に対して実施することにより、全ての修正パッチの修正個所に稼働情報取得部 1 1 2 の呼び出し処理を挿入することができる。

【 0 0 7 0 】

稼働情報取得部 1 1 2 の呼び出し処理の挿入が完了すると、パッチ情報テーブル 1 2 1 の稼働情報取得開始時刻 4 0 3 の欄に現在時刻として x x 年 y y 月 z z 日 0 0 時 0 0 分 0 0 秒を記憶し、対象ソフトウェア 1 2 0 の実行を開始する。ソフトウェア 1 2 0 の実行が開始されると、稼働情報取得処理挿入部 1 1 1 は待ち状態となり、対象ソフトウェア 1 2 0 の終了を監視し、ソフトウェアが終了した時点の時刻 x x 年 y y 月 z z 日 0 0 時 1 0 分 0 0 秒を稼働情報取得終了時刻 4 0 4 の欄に登録する。

10

【 0 0 7 1 】

ソフトウェア 1 2 0 が実行されると、内部の個々の要素が動作する。その際、個々の要素に設定された稼働情報取得部 1 1 2 の呼び出し処理が実行され、それに伴い稼働情報取得部 1 1 2 が呼び出される。ソフトウェア 1 2 0 を実行中に、要素 A 2 0 1 の X + 1 0 0 0 番地が実行された場合、当該アドレスに挿入された稼働情報取得部 1 1 2 の呼び出し処理により、稼働情報取得部 1 1 2 に処理が移る。

【 0 0 7 2 】

稼働情報取得部 1 1 2 に処理が移ると、稼働情報取得部 1 1 2 は、まず、呼び出し元のアドレスを取得し、取得した呼び出し元アドレスを元にパッチ情報テーブル 1 2 1 の挿入アドレス 4 0 2 に、一致するアドレスを格納したレコードがあるか否かを検索する。該当するテーブルが存在する場合、挿入アドレス 4 0 2 を元に稼働情報テーブル 1 2 2 に新規のレコードを作成する。このとき、挿入アドレスに対応する最終実行時刻 7 0 0 を 0 で初期化する。次に、挿入アドレス 4 0 2 に対する最終実行時刻 7 0 0 を取得する。この場合、最終実行時刻 7 0 0 は初期値の 0 が得られる。続いて、稼働情報取得部 1 1 2 が呼び出された時刻を取得する。

20

【 0 0 7 3 】

ここで、稼働情報取得部 1 1 2 は、最終実行時刻 7 0 0 と稼働情報取得部 1 1 2 が呼び出された時刻とを比較し、呼び出し時刻が最終実行時刻 7 0 0 より 1 秒以上経過している場合は、稼働情報テーブル 1 2 2 の最終実行時刻 7 0 0 を現在時刻に更新し、稼働時間に 1 秒加算する。最後に、呼び出し元アドレスの次のアドレスに戻る処理を行うことにより、稼働情報取得部 1 1 2 での処理が終了する。以上の処理が、ソフトウェア 1 2 0 を実行している間、稼働情報取得部 1 1 2 の呼び出し処理が登録された修正位置を通る毎に実行され、稼働情報テーブル 1 2 2 の情報が更新されていく。

30

【 0 0 7 4 】

次に、修正重要度計算部 1 1 3 の処理が開始され、前述までの処理で得られた稼働情報テーブル 1 2 2 の稼働情報から個々の修正情報の稼働率を計算し、そこからパッチの重要度を計算する。修正重要度計算部 1 1 3 は、まず、パッチ情報テーブル 1 2 1 の入力を行い入力が完了したか否かを判定する。この場合、入力は未完了のため次のパッチ名の取得処理へ進む。パッチ名が得られるとパッチの重要度の初期値として 1 を設定する。

40

【 0 0 7 5 】

次に、修正重要度計算部 1 1 3 は、個々のパッチの重要度を稼働情報テーブル 1 2 2 の稼働時間 7 0 1 を元に計算する処理を行う。まず、パッチ情報テーブル 1 2 1 から修正パッチ名に対応する挿入アドレス 4 0 2 を取得する。ここでは、挿入アドレス X + 1 0 0 0 番地が得られる。次に、この挿入アドレス X + 1 0 0 0 番地を元に稼働情報テーブル 1 2 2 を検索すると、この修正位置情報の稼働時間 7 0 1 から 1 0 0 秒が得られる。ここで、ソフトウェア 1 2 0 が実際に実行されていた時間を計算するために、パッチ情報テーブル 1 2 1 から実行開始時刻である稼働情報取得開始時刻 4 0 3 と、実行終了時刻である稼働情報取得終了時刻 4 0 4 を取得する。

【 0 0 7 6 】

50

ここで得られる稼働情報取得開始時刻403は、 xx 年 yy 月 zz 日00時00分00秒であり、稼働情報取得終了時刻404は、 xx 年 yy 月 zz 日00時10分00秒である。この終了時刻から開始時刻を引いた10分、すなわち、600秒がソフトウェアの実働時間であるため、この修正情報の稼働率は、 $100 / 600$ で16%と計算される。稼働率16%は、修正重要度テーブル1000では重要度2に設定されているため重要度が2と決定される。ここで得られた重要度と、修正パッチA21の現在の重要度=1とを比較し、より大きい重要度である重要度2が修正パッチA21の重要度として選択される。以上の処理を繰り返すことにより、最終的に修正パッチA21の重要度は、要素A201と修正位置(+1100番地)(220)の稼働時間から計算される重要度4となる。パッチの重要度が決定すると、パッチ名とパッチの重要度が修正重要度情報テーブル123に登録される。このような処理を全てのパッチに対して繰り返し実施することにより全てのパッチの重要度が決定される。

10

【0077】

最後に、以上から得られた修正重要度情報テーブル123を表示装置等の出力装置13に出力する。作業者は、表示される修正重要度情報テーブル123の内容からどの修正パッチが重要であるかを知って、修正パッチを選択することができる。

【0078】

前述した本発明の実施形態での各処理は、プログラムにより構成し、本発明が備えるCPUに実行させることができ、また、それらのプログラムは、FD、CDROM、DVD等の記録媒体に格納して提供することができ、また、ネットワークを介してデジタル情報により提供することができる。

20

【図面の簡単な説明】

【0079】

【図1】本発明の一実施形態による修正パッチ選択支援システムが構築されるコンピュータシステムの構成を示すブロック図である。

【図2】ソフトウェアと修正パッチとの関係例について説明する図である。

【図3】パッチ情報取得部での処理動作を説明するフローチャートである。

【図4】図3に示す処理によって生成されるパッチ情報テーブルの構成例を示す図である。

【図5】稼働情報取得処理挿入部での処理動作を説明するフローチャートである。

30

【図6】稼働情報取得部での処理動作を説明するフローチャートである。

【図7】図6に示す稼働情報取得部の処理によって生成される稼働情報テーブルの構成例を示す図である。

【図8】修正重要度計算部での処理動作を説明するフローチャートである。

【図9】修正重要度計算部の処理によって作成される修正重要度情報テーブルの構成例を示す図である。

【図10】修正重要度計算部がパッチ毎の重要度を決定するために用いる修正重要度テーブルの構成例を示す図である。

【符号の説明】

【0080】

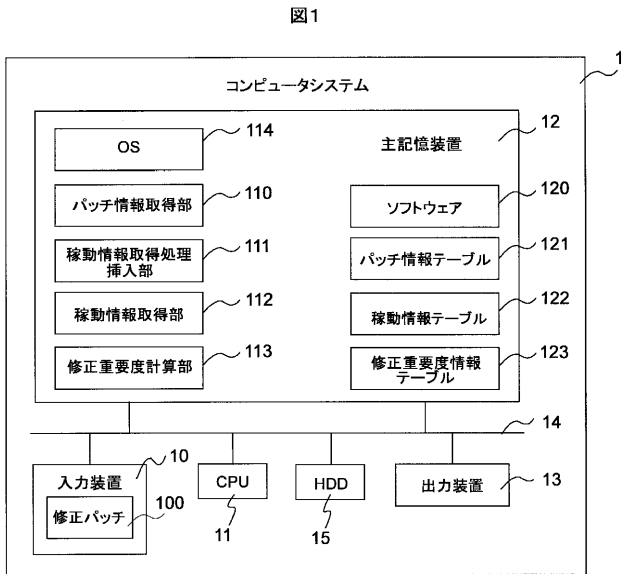
40

- 1 コンピュータシステム
- 10 入力装置
- 11 CPU
- 12 主記憶装置
- 13 出力装置
- 14 バス
- 15 HDD
- 100 修正パッチ
- 110 パッチ情報取得部
- 111 稼働情報取得処理挿入部

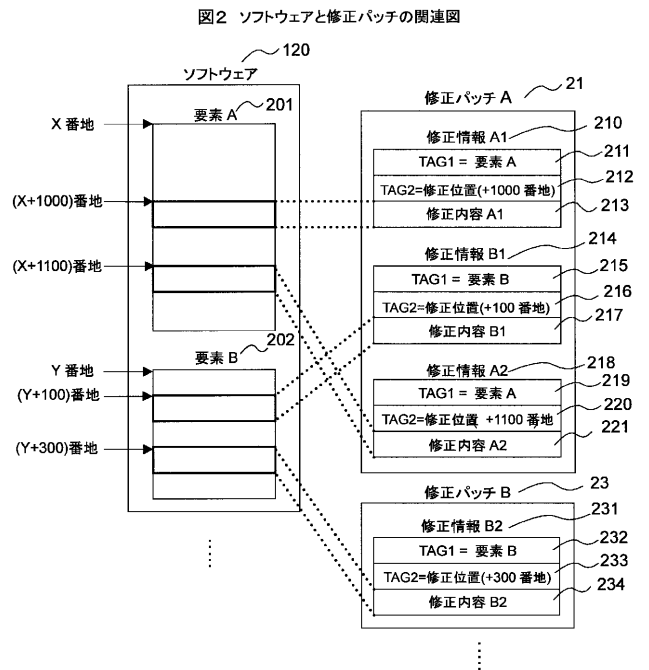
50

- 1 1 2 稼動情報取得部
- 1 1 3 修正重要度計算部
- 1 1 4 OS
- 1 2 0 ソフトウェア
- 1 2 1 パッチ情報テーブル
- 1 2 2 稼動情報テーブル
- 1 2 3 修正重要度情報テーブル

【 図 1 】

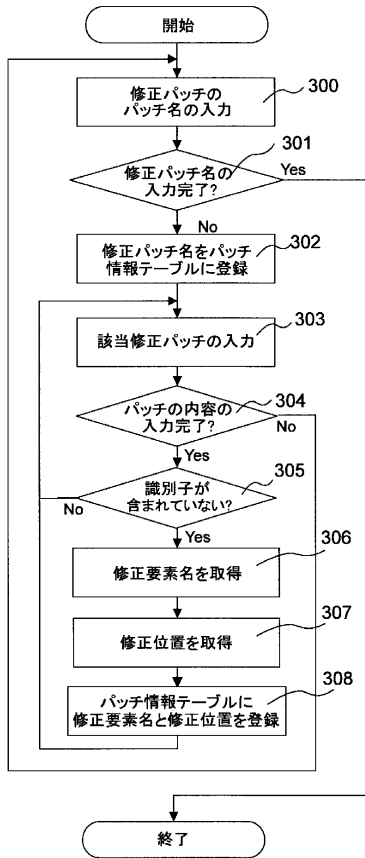


【 図 2 】



【 図 3 】

図3 パッチ情報取得部のフローチャート



【 図 4 】

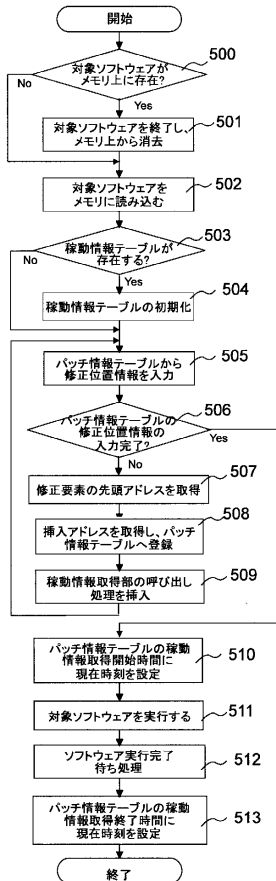
図4 パッチ情報テーブル

修正パッチ名	修正位置情報	挿入アドレス	稼働情報取得開始時刻	稼働情報取得終了時刻
パッチ A	要素 A	+1000 番地	xx 年 yy 月 zz 日 XX 時 YY 分 ZZ 秒	xx 年 yy 月 zz 日 XX 時 YY 分 ZZ 秒
	要素 B	+100 番地		
	要素 A	+1100 番地		
パッチ B	要素 B	+300 番地	Y+300 番地	
...		

121

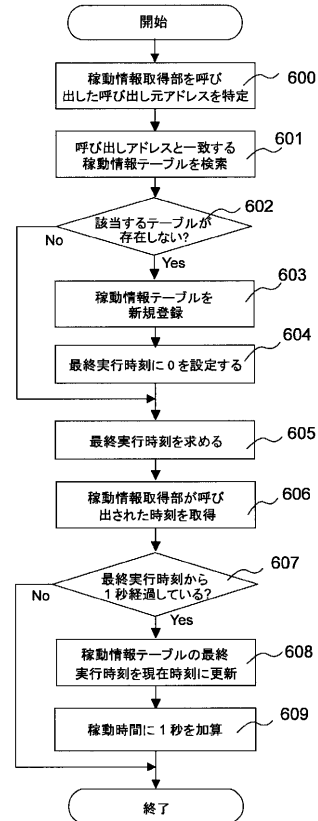
【 図 5 】

図5 稼働情報取得処理挿入部のフローチャート



【 図 6 】

図6 稼働情報取得部のフローチャート



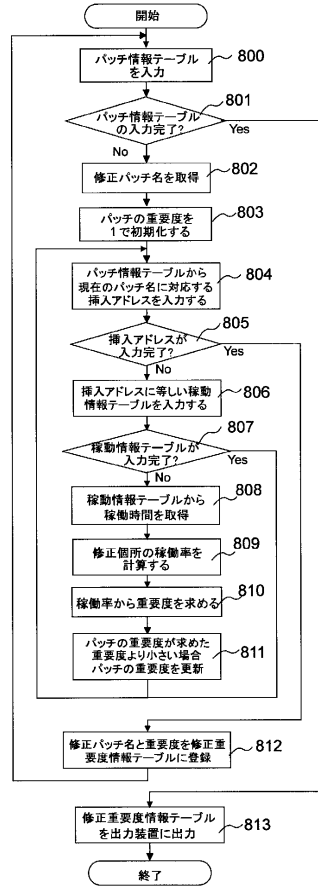
【 図 7 】

図7 稼働情報テーブルの作成例

挿入アドレス	最終実行時刻	稼働時間(秒)
X+1000 番地	xx 年 yy 月 zz 日 00 時 05 分 05 秒	100
Y+100 番地	xx 年 yy 月 zz 日 00 時 01 分 00 秒	5
X+1100 番地	xx 年 yy 月 zz 日 00 時 09 分 35 秒	450
....

【 図 8 】

図8 修正重要度計算部のフローチャート



【 図 9 】

図9 修正重要度情報テーブル

修正バッチ名	重要度
バッチ A	4
バッチ B	2
....

【 図 10 】

図10 修正重要度テーブル

稼働率(%)	重要度
0	1
1~30	2
31~60	3
61~90	4
90~100	5

フロントページの続き

(72)発明者 堀川 和雄

神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア事業部内

Fターム(参考) 5B176 EA01