(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2013-533553
(P2013-533553A)
(43)公表日　平成25年8月22日(2013.8.22)

| (51)Int.Cl. | | F I | | テーマコード（参考） |
|---|---|---|---|---|
| G06F 11/34 | (2006.01) | G06F 11/34 | B | 5B042 |

審査請求 有　予備審査請求 未請求　（全 14 頁）

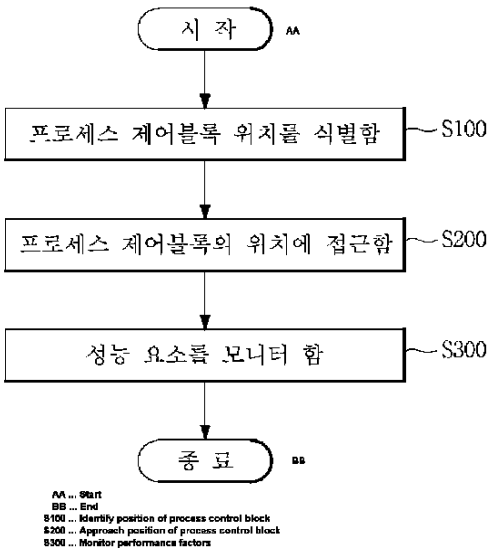
(21)出願番号　特願2013-518210 (P2013-518210)
(86)(22)出願日　平成22年9月7日 (2010.9.7)
(85)翻訳文提出日　平成24年11月16日 (2012.11.16)
(86)国際出願番号　PCT/KR2010/006068
(87)国際公開番号　WO2012/033237
(87)国際公開日　平成24年3月15日 (2012.3.15)

(71)出願人　512297583
ヒョンダイ　モーター　カンパニー
大韓民国，ソウル　１３７－９３８，ソチョ－グ，ヤンゼ－ドン　２３１
(71)出願人　512297594
キアモータース　コーポレーション
大韓民国，ソウル１３７－９３８，ソチョ－グ，ヤンゼ－ドン　２３１
(71)出願人　510054603
イファ　ユニーバーシティ－インダストリー　コラボレーション　ファウンデーション
大韓民国，ソウル　１２０－７５０，ソデムン－グ，デヒョン－ドン，１１－１，イファ　ユニバーシティ

最終頁に続く

(54)【発明の名称】システムテスト方法

(57)【要約】
【課題】システムの運用環境に影響を与えずに性能ボトルネックとその原因及び位置を識別するためにデータ収集するシステムテスト方法を提供する。
【０００８】　本発明は、プロセス制御ブロック（Ｐｒｏｃｅｓｓ　Ｃｏｎｔｒｏｌ　Ｂｌｏｃｋ）の位置を識別するステップと、プロセス制御ブロックの位置に接近するステップと、プロセス制御ブロックの性能要素をモニタリングするステップとを含み、前記プロセス制御ブロックは、カーネルメモリ内部に位置することを特徴とする。
。
【選択図】図１



【特許請求の範囲】
【請求項１】
プロセス制御ブロック（Ｐｒｏｃｅｓｓ　Ｃｏｎｔｒｏｌ　Ｂｌｏｃｋ）の位置を識別するステップと、
前記プロセス制御ブロックの位置に接近するステップと、
前記プロセス制御ブロックの性能要素をモニタリングするステップと
を含むことを特徴とするシステムテスト方法。
【請求項２】
前記プロセス制御ブロックは
カーネルメモリ内部に位置することを特徴とする請求項１に記載のシステムテスト方法。
【請求項３】
前記プロセス制御ブロックの位置に接近するステップにおいて、
仮想ドライバを用いて前記カーネルメモリに接近することを特徴とする請求項２に記載のシステムテスト方法。
【請求項４】
前記プロセス制御ブロックの性能要素をモニタリングするステップにおいて、
予め設定された時間間隔ごとに前記プロセス制御ブロックの性能要素を測定することを特徴とする請求項１に記載のシステムテスト方法。
【請求項５】
前記プロセス制御ブロックの性能要素をモニタリングするステップにおいて、
前記プロセス制御ブロックのプロセス及びスレッド（Ｔｈｒｅａｄ）に対する性能要素を測定することを特徴とする請求項２に記載のシステムテスト方法。
【請求項６】
前記プロセス制御ブロックの性能要素は
プロセッサ使用量、メモリ使用量、ページフォールト（Ｐａｇｅ　Ｆａｕｌｔ）中一つ以上を含むことを特徴とする請求項５に記載のシステムテスト方法。
【請求項７】
前記プロセス制御ブロックのプロセス性能要素は
アイディー（ＩＤ）、プロセスの状態、プロセスの優先順位、ヒープ（Ｈｅａｐ）使用量、プロセスの動作時間、使用時間、カーネル使用時間中一つ以上を含むことを特徴とする請求項６に記載のシステムテスト方法。
【請求項８】
前記プロセス制御ブロックのスレッド性能要素は
アイディー、運営状態（Ｒｕｎ　Ｓｔａｔｅ）、基本優先順位、現在優先順位、使用時間、カーネル使用時間、コールスタック（Ｃａｌｌ　Ｓｔａｃｋ）中一つ以上を含むことを特徴とする請求項６に記載のシステムテスト方法。
【請求項９】
前記プロセス制御ブロックは
プロセスの識別子、レジスタコンテクスト（Ｃｏｎｔｅｘｔ）、プロセスのアドレス、メモリ使用量、共有関数リスト、プロセスの資源（Ｒｅｓｏｕｒｃｅ）、プロセスの優先順位、プロセスの状態中一つ以上に対する情報を格納することを特徴とする請求項１に記載のシステムテスト方法。
【請求項１０】
前記プロセス制御ブロックの性能要素をモニタリングするステップは
カーネルに分析コードを挿入して前記プロセス制御ブロックの性能要素をモニタリングすることを特徴とする請求項１に記載のシステムテスト方法。
【請求項１１】
前記プロセス制御ブロックの性能要素をモニタリングするステップは
シミュレーションプログラムを用いて前記プロセス制御ブロックの性能要素をモニタ

リングすることを特徴とする請求項１に記載のシステムテスト方法。
【請求項１２】
前記プロセス制御ブロックの性能要素をモニタリングするステップは
レジスタを含むハードウェアを用いて前記プロセス制御ブロックの性能要素をモニタリングすることを特徴とする請求項１に記載のシステムテスト方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、システムテスト方法に係り、より詳しくは、システムの運用環境に影響を与えずに、性能ネックとその原因及び位置を識別するためのデータを収集するシステムテスト方法に関する。
【背景技術】
【０００２】
最近自動車のような機械産業分野でも電子制御技術に対する関心が大きい。電子制御技術の核心はエンベデッドソフトウェアであり、これらソフトウェアの品質が全体製品の品質に大きな影響を及ぼす。
従来も品質の一助にするソフトウェアテスト技術とテスト道具等が開発された。しかし、エンベデッドソフトウェア固有の特徴等のため、これらをそのままエンベデッドソフトウェアテストに適用するには様々な問題があった。
【０００３】
第一に、エンベデッドソフトウェアは特定のハードウェアと機能に合わせて製作されるソフトウェアである。殆どのエンベデッドソフトウェアは、メモリのようなシステム資源に余裕のない制約的条件に最適化されて動作するように設計される。したがって、ターゲット環境で動作するエンベデッドソフトウェアは、ホスト環境で動作する一般のソフトウェアと比べた場合、システム運用時に資源の制約が甚だしいとの問題点がある。
【０００４】
第二に、エンベデッドソフトウェアは、メニュー選択のような使用者命令による動作よりも電子的シグナルや通信プロトコルのような多様な類型の外部入力による動作が大きな比率を占める。したがって、使用者命令によって主に動作する従来のソフトウェアテスト技術とテスト道具等をエンベデッドソフトウェアテストにそのまま適用し難いとの問題点がある。
【０００５】
第三に、従来のソフトウェアテスト技術は、性能テスト時に高いプロセッサ使用率が持続される場合にのみ性能ボトルネックを表す症候と見做す。この場合、重要なコンポーネントのプロセッサ使用率が基本性能以下に落ちる場合は異常症候と判断しない。したがって、システムの性能を正確にテストすることができない問題点がある。
【発明の概要】
【発明が解決しようとする課題】
【０００６】
本発明は、システム運用時に必要な資源の使用を最小限にしてシステムの運用環境に影響を与えずに性能ボトルネックとその原因及び位置を識別するためのデータ収集するシステムテスト方法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】
【０００７】
本発明は、プロセス制御ブロック（Ｐｒｏｃｅｓｓ　Ｃｏｎｔｒｏｌ　Ｂｌｏｃｋ）の位置を識別するステップと、プロセス制御ブロックの位置に接近するステップと、プロセス制御ブロックの性能要素をモニタリングするステップとを含むことを特徴とする。
【発明の効果】
【０００８】
本発明によれば、システム運用に必要な資源を最小限用いてシステムの運用環境に影

響を与えずにシステムをテストすることができる。
　また、本発明はシステムの外部から入力されるデータに基づきシステムをテストすることができる長所がある。
　さらに、本発明はプロセッサの使用率が基準以下に落ちる場合でもボトルネック現象なのか否かをテストすることができる長所がある。
【図面の簡単な説明】
【０００９】
【図１】本発明に係るシステムテスト方法を示すフローチャートである。
【図２】本発明に係るシステムテスト方法を具現するプログラムコードである。
【図３】本発明に係るシステムテスト方法を行うシステムの構造を示す図である。
【発明を実施するための形態】
【００１０】
　本発明に係るシステムテスト方法は、プロセス制御ブロック（Ｐｒｏｃｅｓｓ　Ｃｏｎｔｒｏｌ　Ｂｌｏｃｋ）の位置を識別するステップと、プロセス制御ブロックの位置に接近するステップと、プロセス制御ブロックの性能要素をモニタリングするステップとを含む。
　以下、図を参照して、本発明の多様な実施形態を具体的に説明する。
　本発明に係るシステムテスト方法で、プロセス制御ブロック（Ｐｒｏｃｅｓｓ　Ｃｏｎｔｒｏｌ　Ｂｌｏｃｋ）は運用体制がシステムに運用されているプロセス等の実行情報を管理する資料構造を意味する。
　例えば、プロセス制御ブロックは、システムで運用中のプロセス等の最新実行情報を実時間で管理するカーネルの資料構造を意味する。
【００１１】
　本発明に係るシステムテスト方法は、システムのメモリに発生した欠陥を見付けるため、プロセス制御ブロックを介しメモリと関連する関数テーブルをフッキングする。
　特に、本発明に係るシステムテスト方法は、プロセス制御ブロックのデータのうち性能ボトルネックとその原因を分析するため、ページフォールト率とプロセッサ使用率のようなシステム実行情報等をハッキングしてシステム性能を分析することができる。
　プロセス制御ブロックハッキングは、性能分析に必要なデータ収集をプロセス制御ブロック１ヶ所に集中させることにより、テストによるシステム性能低下を最小にすることができる。
　さらに、本発明は、前記のようなシステム性能低下を最小にしながら、同時にシステム運用環境での性能テスト要求事項を満足させることができる。
【００１２】
　例えば、本発明に係るシステムテスト方法は、次のような条件を仮定して行うことができる。
［システムテスト条件］
１．テスト範囲：全体システム内の全てのハードウェアとソフトウェアコンポーネント等が作動する状況のシステムに対する性能テスト。
２．テスト方式：システム実行方式を保障するためｒｅ－ｃｏｍｐｉｌｅ、ｒｅ－ｌｉｎｋ、ｒｅ－ｅｘｅｃｕｔｅされない方式でのランタイムテスト。
３．テスト対象：デバッギング情報のような付加的なコードを含まないテスト対象ソフトウェアの原本ソース／二進コードの変更なくシステムにローディングされたバイナリーコードを対象にしたテスト。
４．性能データ：性能ボトルネックの原因として、プロセッサだけでなくメモリ、Ｉ／Ｏ、ネットワーク資源まで考慮することができる性能データ収集。
【００１３】
５．ボトルネック位置追跡：性能ボトルネックの位置で関数のようにソースレベル分析が可能な位置データ収集。
６．ハードウェア独立性：テストのための別途のハードウェア追加や依存性なく、システ

ム運用環境と同一のハードウェア条件でのテスト。
７．ソフトウェア独立性：テストのため特別に製作されたカーネル（Ｉｎｓｔｒｕｍｅｎｔｅｄ Ｋｅｒｎｅｌ）や仮想マシンを用いない、システム運用環境と同一のソフトウェア条件でのテスト。
８． 性能遅延率：実時間動作を保障するため、テストによるシステム性能遅延率最小化。
９． コード追加率：制限された資源内で動作するため、テストによるコード追加率最小化。

【００１４】
しかし、このような条件は、システム運用環境に伴いいくらでも変更することができ、本発明は、このようなシステム運用環境に制限されずに適用できる。
プロセス制御ブロックは、システムに運用されるプロセスの実行情報を盛り込んでいるカーネル資料構造である。プロセス制御ブロックには、性能ボトルネックを判断するプロセッサ使用量と余裕メモリサイズのような性能要素等も含まれており、カーネルによりこれら値は常に一番最近の値に維持される。
本発明に係るシステムテスト方法は、この値等をハッキングするエージェント開発のための性能分析に用いられる。

【００１５】
以下、性能ボトルネックの原因を分析するために必要な性能要素及び関連したプロセス制御ブロック構造について説明する。
性能は、システムやコンポーネント等が与えられたシステムの制約条件内で機能を行う程度を意味する。性能テストは、特定性能要求事項をシステムが満足するかに関する評価である。
このような性能テストを通じて性能ボトルネックとその原因を分析し、これを解決することによりシステムの性能を改善することができる。
システムの性能ボトルネックはメモリ、Ｉ／Ｏ デバイス、ネットワークのような制限された資源等に対する競争で、システムの性能が低下する現象である。システム性能ボトルネックの原因は、資源不足、共有資源競合、資源独占、資源の誤った構成、或いは資源の誤った動作のように多様である。

【００１６】
先ず、メモリが原因となる性能ボトルネックは、使用可能なメモリが不足する時に発生することがあり、メモリ不足症侯を識別することができる核心的なメモリ性能要素としてページフォールトとメモリ使用量がある。
本発明に係るシステムテスト方法は、多様な性能ボトルネック等を識別することができる。
先ず、本発明に係るシステムテスト方法は、ページフォールトを基準にメモリボトルネックを判断することができる。

【００１７】
例えば、ページフォールトが高いとメモリボトルネックであり得る。ページフォールトとは、プログラムが自分のアドレス空間には存在するものの、システムのメモリには現在ないデータやコードに接近を試みた場合に発生する現象である。ページフォールトが発生すると、運営体制はそのデータをメモリに持って来て恰もページフォールトが全然発生しないもののようにプログラムが引き続き作動するようにする。運営体制のページフォールトに対する例外処理のためアプリケーションの処理時間は遅延され、全体システム性能に影響を与える。

【００１８】
さらに、本発明に係るシステムテスト方法は、メモリ使用量を介し性能ボトルネックを識別することができる。システムのメモリは物理メモリ使用量、仮想メモリ使用量があり、プロセス別にヒープメモリ使用量で区分することができる。本発明に係るシステムテスト方法は、このようなメモリ使用量等の和を基準に性能ボトルネックを判断する。

さらに、本発明に係るシステムテスト方法は、プロセッサ使用量（或いはＣＰＵ使用量）を介し性能ボトルネックを識別することができる。

例えば、本発明に係るシステムテスト方法は、プロセッサ使用量が高く維持されながら余裕メモリのある場合はＣＰＵにボトルネックがあると判断することができる。その反面、プロセッサ使用量が高くメモリも枯渇したのであれば、性能問題はＣＰＵよりもメモリボトルネックと判断することができる。

【００１９】

さらに、本発明に係るシステムテスト方法は、プロセス使用量（Ｐｒｏｃｅｓｓｏｒ　Ｕｓａｇｅ）に従い性能ボトルネックを識別することができる。プロセス使用量はシステムの実行時間で、全体ＣＰＵ使用量から待機時間（Ｉｄｌｅ　Ｔｉｍｅ）を除いた時間を意味する。

さらに、本発明に係るシステムテスト方法は、使用時間（Ｕｓｅｒ　Ｔｉｍｅ）に従い性能ボトルネックを判断することができる。使用時間は、実行が使用者空間で留まった時間を意味し、まさにアプリケーションの実行時間を意味する。

【００２０】

さらに、本発明に係るシステムテスト方法は、カーネル使用時間（Ｋｅｒｎｅｌ　Ｔｉｍｅ）に従い性能ボトルネックを判断することができる。カーネル使用時間は実行がカーネル空間で留まった時間で、カーネルのサービス処理時間を意味する。　プロセス制御ブロックは、ランタイムにプロセス等を制御するための目的で運営体制カーネルが管理する資料構造である。一般に、プロセス制御ブロックにはプロセス識別子、レジスタコンテクスト、プロセスのアドレス空間、プロセスのメモリ使用量、共有関数リスト、プロセスが所有する資源等、プロセスの優先順位と状態などの実行情報が格納されている。

【００２１】

図１は、本発明に係るシステムテスト方法を示すフローチャートである。

図２は、本発明に係るシステムテスト方法を具現するプログラムコードである。

図１に示す通り、本発明に係るシステムテスト方法は、プロセス制御ブロックの位置を識別するステップ（Ｓ１００）と、プロセス制御ブロックに接近するステップ（Ｓ２００）と、プロセス制御ブロックの性能要素をモニタリングするステップ（Ｓ３００）とを含む。

【００２２】

プロセス制御ブロックの位置を識別するステップ（Ｓ１００）で、プロセスが生成され消滅する時、そのプロセス制御ブロックも共に生成され消滅するためプロセス制御ブロックが格納される位置は決定されていない。

しかし、プロセッサ（例えば、ＣＰＵ）を占有する現行プロセス（Ｃｕｒｒｅｎｔ　Ｐｒｏｃｅｓｓ）に関するプロセス制御ブロックのｂａｓｅ　ａｄｄｒｅｓｓは、特定メモリ空間に管理されるか、場合に応じて予め固定されたアドレスで管理されるため、現行プロセスのプロセス制御ブロック情報が分かる。

例えば、図２に示したプログラムコードのｌｉｎｅ　７でのように現行プロセスのプロセス制御ブロックｂａｓｅ　ａｄｄｒｅｓｓをスタックポインタで計算し、ｌｉｎｅ　３１のように全てのプロセスのプロセス制御ブロックに接近することができる。

プロセス制御ブロックの位置に接近するステップ（Ｓ２００）で、プロセス制御ブロックはカーネルメモリ空間に存在する。

【００２３】

この場合、カーネル空間には保安（Ｓｅｃｕｒｉｔｙ）次元で接近を許容しないこともある。このようにカーネル空間に対する接近が遮断される時、下記プログラムコードのようにｐｓｅｕｄｏ　ｃｏｄｅｓを、カーネルアドレス空間を共有することができるよう仮想ドライバ形態に具現し、カーネルメモリ空間に存在するプロセス制御ブロックに接近することができる。

プログラム制御ブロックの性能要素をモニタリングするステップ（Ｓ３００）で、図２に示したプログラムコードのｌｉｎｅ　６１とｌｉｎｅ　６５のようにｔｉｍｅｒ－ｉ

ｎｔｅｒｒｕｐｔを用いて指定された時間間隔（ｅ．ｇ．１ｓｅｃ、１００ｍｓｅｃ）ごとに性能データを測定することができる。

【００２４】

さらに、本発明に係るシステム方法では、システム性能をシステム単位だけでなくプロセスとスレッドの単位にまで分析するため、図２に示したプログラムコードのｌｉｎｅ　３１とｌｉｎｅ　３３のように、プロセス制御ブロックのプロセスとスレッドリスト等を循環しながら性能要素を測定することができる。

この場合、システム性能ボトルネックが発生する位置を詳しく追跡するためスレッド別にコールスタック（Ｃａｌｌｓｔａｃｋ）情報を格納する。

【００２５】

各モニタリング単位別に測定する性能要素は次の通りである。

プロセス制御ブロックの性能要素はプロセッサ使用量、メモリ使用量、ページフォールト（Ｐａｇｅ　Ｆａｕｌｔ）中の一つ以上を含むことができる。

プロセス制御ブロックのプロセスに対する性能要素はアイディー（ＩＤ）、プロセスの状態、プロセスの優先順位、ヒープ（Ｈｅａｐ）使用量、プロセスの動作時間、使用時間、カーネル使用時間中一つ以上を含むことができる。

プロセス制御ブロックのスレッドに対する性能要素はアイディー、運営状態（Ｒｕｎ　Ｓｔａｔｅ）、基本優先順位、現在優先順位、使用時間、カーネル使用時間、コールスタック情報中一つ以上を含むことができる。

【００２６】

図３は、本発明に係るシステムテスト方法を行うシステムの構造を示す図である。

図３に示す通り、本発明に係るシステムテスト方法を行うシステムは、エージェント部１２０及びテスト管理部２１０を含む。

エージェント部１２０は、テストの対象となるターゲットシステムに含まれ、性能データを測定する役割を果たす。エージェント部１２０は、図１に示されたアルゴリズムを行う。

テスト管理部２１０は、ＰＣ（Ｐｅｒｓｏｎａｌ　Ｃｏｍｐｕｔｅｒ）のようなホストシステム２００に含まれ、エージェント部１２０から収集された性能データを分析して性能ボトルネックを検出しテストカバレッジを分析する。

【００２７】

具体的に、エージェント部１２０は、運用環境に配置されたターゲットシステムに共に搭載され、使用者のテスト開始と終了命令に従いシステムが運営されながら、周期的にシステム性能と関連するデータを測定する役割を果たす。

例えば、エージェント部１２０はＰｅｒｆＡｇｅｎｔ．ｄｌｌ及びＰｅｒｆＲＯＢＯ．ｅｘｅを含むことができる。

ＰｅｒｆＡｇｅｎｔ．ｄｌｌは、図１に示すアルゴリズムを具現する仮想のカーネルデバイスドライバで、性能テストのためカーネルのプロセス制御ブロック情報をハッキングする。

ＰｅｒｆＲＯＢＯ．ｅｘｅは、使用者のテスト開始と終了命令によりＰｅｒｆＡｇｅｎｔ．ｄｌｌの活性化可否を制御するテストサーバの役割を果たす。

テスト開始命令によりＰｅｒｆＡｇｅｎｔ．ｄｌｌが活性化されると、タイマ設定を介しシステム性能モニタリングモジュールを実行させ、使用者要求によりテストを終了する時も設定されたタイマを終了させることによりテストを終了する。

【００２８】

一方、テスト管理部２１０は、収集された性能データを指定された格納媒体に格納することができる。テスト管理部２１０は、ホストシステム２００ログファイルを分析して運営タイムに発生した性能ボトルネックを検出する役割を果たす。

例えば、テスト管理部２１０が性能データを格納媒体に二進コードで格納するとすれば、テスト管理部２１０はテスト対象に対する二進実行コードと収集されたプロファイリングデータを入力にし、テストカバレッジと性能情報を出力するとき、コールスタック情

報に基づき何れの関数位置で問題が発生したのかの位置情報を共にディスプレイすることができる。

【００２９】

本発明に係るシステムテスト方法で、システムの性能をモニタリングする多様な方法が使用可能である。

先ず、カーネルに分析コードを挿入する方法が用いられる。この技術は、カーネルに分析コードを静的に或いは動的に挿入させる。

動的に挿入する技術ではランタイムにカーネルコードへ分析コードを挿入することができ、この場合、システムを再起動せずとも性能分析に必要なデータを収集するコードを挿入することができる。

静的に挿入する技術は、予め製作されたカーネルを用いる。

例えば、カーネル内部に重要なシステム性能要素をモニタリングする分析コードを植えておくことができる。性能モニタリングする場合、カーネル内部に予め挿入されている分析コードを介し性能データの送信を受ける。その結果、性能モニタリングするとき、プロセッサ使用量、メモリ使用量を含んで非常に多様なシステム性能要素の測定が可能である。

【００３０】

さらに、性能モニタリング技術としてシミュレータを利用する方法が用いられる。この方式は、開発段階初期のシステムの性能を検討する場合に有用である。

さらに、性能モニタリング技術としてハードウェアを用いることもできる。ハードウェア性能要素等は特に製作されたレジスタ集合を意味し、ソフトウェア基盤の性能要素に比べ少ないシステムオーバーヘッドでＣＰＵ、キャッシュ、メモリと関連した性能をモニタリングすることが可能である。この方式は、ソースコードや二進コードを修正しなくとも良い。

【産業上の利用可能性】

【００３１】

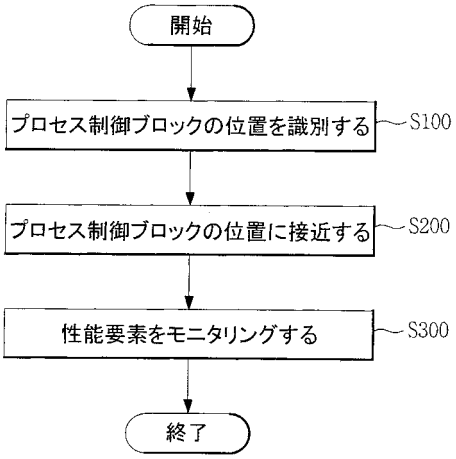
本発明は、システム運用に必要な資源の使用を最小限にしてシステムの運用環境に影響を与えずにシステムをテストすることができる。

【符号の説明】

【００３２】

１００　ターゲットシステム
１１０　プロセス制御ブロック
２００　ホストシステム
２１０　テスト管理部

【図１】




【図２】

```
01  void * pPCB;                                   // PCB information
02  int uTimerId;                                  // Timer ID for performance monitoring
03  int uTimeOut = 1000;                           // Time-out value in milliseconds
04  DWORD GetKernelStackPoint();  // API for getting value of kernel stack point
05  int SetTimer(uElapse, lpTimerFunc);            // API for creating a timer with the specified time-out value
06  void KillTimer(uTimerId);                      // API for destroying the specified timer
07  void SaveLog(data, ...);              // API for saving data into log-file

08  void PERFAGENT_GetPCB(void)
09  {// Accessing the PCB memory space
10          pPCB = GetKernelStackPoint() & 0xFFFFE000; // e.g., in case of WinCE platform
11  }

12  void PERFAGENT_MemoryUsage()
13  { // Measuring memory usage
14          DWORD total = pPCB->NumPages * pPCB->SysPageSize;
15          DWORD unused = pPCB->FreePhysPage * pPCB->SysPageSize;
16          DWORD usage = total – unused;
17          Save(usage);
18  }

19  void PERFAGENT_HeapUsage(pProc)
20  { // Measuring heap usage
21          DWORD usage ← calculate heap usage using pProc->memPointer;
22          Save(usage);
23  }

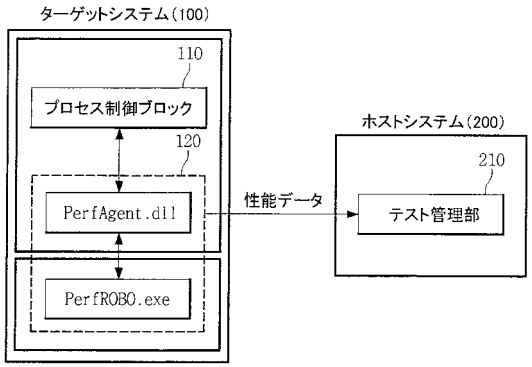
24  void PERFAGENT_CallStack(pThread)
25  { // Measuring thread's call stack
26          void * site ← dump callstack using pThread->pcstkTop;
27          Save(site);
28  }

29  void PERFAGENT_PCBHooking()
30  { // PCB hacking
31          PPROCESS pProc = pPCB->procList.pFwd;
32          while (pProc) {
33                  PTHREAD pThread = pProc->thrdList.pFwd;
34                  while (pThread) {
35                          // save thread status
36                          Save(pThread->dwId);
37                          Save(pThread->bBPrio, pThread->bCPrio);
38                          Save(pThread->bRunState, pThread->bWaitState);
39                          Save(pThread->dwUTime);
40                          Save(pThread->dwKTime);
41                          PERFAGENT_CallStack(pThread);
42                          pThread = pThread->pFwd;        // next thread
43                  }
44                  // save process status
45                  Save(pProc->dwId);
46                  Save(pProc->bPrio);
47                  Save(pProc->bState);
48                  Save(pProc->dwProcTime);
49                  PERFAGENT_HeapUsage(pProc);
50                  pProc = pProc->pFwd;            // next process
51          }
52          // save system status
53          Save(pPCB->dwProcTime);
54          Save(pPCB->nPageFault);
55          PERFAGENT_MemoryUsage();
56  }

57  // Performance Agent
58  void PERFAGENT_Start(void)
59  {
60          PERFAGENT_GetPCB();
61          uTimerId = SetTimer(uTimeOut, PERFAGENT_PCBHooking);
62  }

63  void PERFAGENT_Stop(void)
64  {
65          KillTimer(uTimerId);
66  }
```

【図３】



【国際調査報告】

**INTERNATIONAL SEARCH REPORT**

International application No.
**PCT/KR2010/006068**

A. CLASSIFICATION OF SUBJECT MATTER

***G06F 11/36(2006.01)i, G06F 11/30(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)

G06F 11/36; G06F 15/16; G06F 15/00; G06F 17/00; G06F 9/06

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean Utility models and applications for Utility models: IPC as above
Japanese Utility models and applications for Utility models: IPC as above

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

eKOMPASS (KIPO internal) & Keywords: process, control, block, data structure, search, thread, priority, monitoring

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X<br>Y<br>A | KR 10-2009-0081749 A (SAMSUNG ELECTRONICS CO., LTD.) 29 July 2009<br>See abstract; paragraphs [0027]-[0028], paragraphs [0067]-[0071]; claims 1,18 and figures<br>4,7 | 1-3,9,11<br>4,5<br>6-8,10,12 |
| Y<br>A | KR 10-2003-0041612 A (UNIT SYSTEMS CO., LTD.) 27 May 2003<br>See abstract; page 2 - page 3; claim 1 and figure 1 | 4,5<br>1-3,6-12 |
| A | KR 10-2008-0079343 A (KTFREETEL CO., LTD.) 01 September 2008<br>See abstract; paragraphs [0023]-[0031]; claim 5 and figure 1 | 1-12 |
| A | KR 10-2009-0001897 A (KTFREETEL CO., LTD.) 09 January 2009<br>See abstract; paragraphs [0019]-[0027]; claim 1 and figure 1 | 1-12 |

☐ 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 | Date of mailing of the international search report |
|---|---|
| 10 JUNE 2011 (10.06.2011) | **13 JUNE 2011 (13.06.2011)** |
| Name and mailing address of the ISA/KR<br>Korean Intellectual Property Office<br>Government Complex-Daejeon, 139 Seonsa-ro, Daejeon 302-701, Republic of Korea<br>Facsimile No. 82-42-472-7140 | Authorized officer<br><br>Telephone No. |

Form PCT/ISA/210 (second sheet) (July 2009)

**INTERNATIONAL SEARCH REPORT**
Information on patent family members

International application No.
**PCT/KR2010/006068**

| Patent document cited in search report | Publication date | Patent family member | Publication date |
|---|---|---|---|
| KR 10-2009-0081749 A | 29.07.2009 | NONE | |
| KR 10-2003-0041612 A | 27.05.2003 | NONE | |
| KR 10-2008-0079343 A | 01.09.2008 | NONE | |
| KR 10-2009-0001897 A | 09.01.2009 | NONE | |

국제조사보고서

국제출원번호
**PCT/KR2010/006068**

**A.** 발명이 속하는 기술분류(국제특허분류(IPC))

***G06F 11/36(2006.01)i, G06F 11/30(2006.01)i***

**B.** 조사된 분야

조사된 최소문헌(국제특허분류를 기재)
G06F 11/36; G06F 15/16; G06F 15/00; G06F 17/00; G06F 9/06

조사된 기술분야에 속하는 최소문헌 이외의 문헌
한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC
일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))
eKOMPASS(특허청 내부 검색시스템) & 키워드: 프로세스, 제어, 블록, 자료 구조, 검색, 스레드, 우선순위, 모니터링

**C.** 관련 문헌

| 카테고리* | 인용문헌명 및 관련 구절(해당하는 경우)의 기재 | 관련 청구항 |
|---|---|---|
| X<br>Y<br>A | KR 10-2009-0081749 A (삼성전자주식회사) 2009.07.29<br>요약; 단락 [0027]-[0028],단락 [0067]-[0071]; 청구항 1,18 및 도면 4,7 참조. | 1-3,9,11<br>4,5<br>6-8,10,12 |
| Y<br>A | KR 10-2003-0041612 A ((주)유니트시스템즈) 2003.05.27<br>요약; 페이지 2 - 페이지 3; 청구항 1 및 도면 1 참조 | 4,5<br>1-3,6-12 |
| A | KR 10-2008-0079343 A (주식회사 케이티프리텔) 2008.09.01<br>요약; 단락 [0023]-[0031]; 청구항 5 및 도면 1 참조 | 1-12 |
| A | KR 10-2009-0001897 A (주식회사 케이티프리텔) 2009.01.09<br>요약; 단락 [0019]-[0027]; 청구항 1 및 도면 1 참조 | 1-12 |

☐ 추가 문헌이 C(계속)에 기재되어 있습니다.

☒ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

"A" 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌

"E" 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌

"L" 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌

"O" 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌

"P" 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

"T" 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌

"X" 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.

"Y" 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.

"&" 동일한 대응특허문헌에 속하는 문헌

| 국제조사의 실제 완료일 | 국제조사보고서 발송일 |
|---|---|
| 2011년 06월 10일 (10.06.2011) | **2011년 06월 13일 (13.06.2011)** |

| ISA/KR의 명칭 및 우편주소 | 심사관 |
|---|---|
| 대한민국 특허청<br>(302-701) 대전광역시 서구 청사로 189,<br>정부대전청사<br>팩스 번호 82-42-472-7140 | 김종택<br>전화번호 042 481 5983 |

서식 PCT/ISA/210 (두 번째 용지) (2009년 7월)

국 제 조 사 보 고 서
대응특허에 관한 정보

국제출원번호
**PCT/KR2010/006068**

| 국제조사보고서에서 인용된 특허문헌 | 공개일 | 대응특허문헌 | 공개일 |
|---|---|---|---|
| KR 10-2009-0081749 A | 2009.07.29 | 없음 | |
| KR 10-2003-0041612 A | 2003.05.27 | 없음 | |
| KR 10-2008-0079343 A | 2008.09.01 | 없음 | |
| KR 10-2009-0001897 A | 2009.01.09 | 없음 | |

서식 PCT/ISA/210 (대응특허 추가용지) (2009년 7월)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,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,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PE,PG,PH,PL,PT,RO,RS,RU,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,ZA,ZM,ZW

(74)代理人 110000051
特許業務法人共生国際特許事務所
(72)発明者 チョエ，ビョン ジュ
大韓民国，ソウル １３５－７８８，カンナムーグ， アプグジョンードン ヒョダイアパート７４ドン１３０５ホ
(72)発明者 ソ，ジュ ヨン
大韓民国，ソウル １３８－２００，ソンパーグ，ムンジョンードン オリンピックフェミリタウン １０５ドン３０４ドン
(72)発明者 ヤン，スン ワン
大韓民国，ギョンギード ４３５－７０４，グンポーシ，グァンジョンードン サムソン ジャンミアパート，１１３７－２２０２
(72)発明者 オ，ジョン ソク
大韓民国，ギョンギード ４２３－８０１，グァンミョンーシ グァンミョン１ードン，７０－１３ボンジ，２Ｆ

Ｆターム(参考) 5B042 GA23 HH20 MB03 MC22 MC33