

(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2007-226800

(P2007-226800A)

(43) 公開日 平成19年9月6日(2007.9.6)

(51) Int. Cl.

F I

テーマコード (参考)

G06F 9/44 (2006.01)

G06F 9/44 530S

G06F 9/50 (2006.01)

G06F 9/46 462A

審査請求 有 請求項の数 10 O L (全 16 頁)

(21) 出願番号 特願2007-40392 (P2007-40392)
 (22) 出願日 平成19年2月21日 (2007.2.21)
 (31) 優先権主張番号 10-2006-0018282
 (32) 優先日 平成18年2月24日 (2006.2.24)
 (33) 優先権主張国 韓国 (KR)

(71) 出願人 390019839
 三星電子株式会社
 Samsung Electronics
 Co., Ltd.
 大韓民国京畿道水原市八達区梅灘洞416
 番地
 (74) 代理人 100070150
 弁理士 伊東 忠彦
 (74) 代理人 100091214
 弁理士 大貫 進介
 (74) 代理人 100107766
 弁理士 伊東 忠重
 (72) 発明者 趙 雄 石
 大韓民国ソウル特別市廣津区紫陽2洞60
 O-14 5層

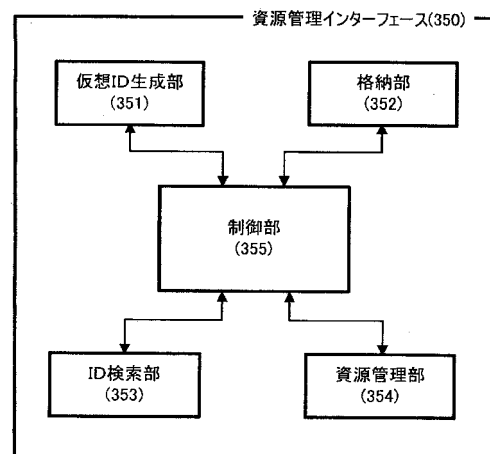
(54) 【発明の名称】 多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する装置およびその方法

(57) 【要約】

【課題】 多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する装置およびその方法に関し、より詳しくは、多重ジャバアプリケーション環境で仮想のIDを用いて資源の割当および解除を行うことによって、資源をより効率的に用いることのできる装置およびその方法を提供する。

【解決手段】 ジャバアプリケーション環境で仮想IDを用いて資源を管理する装置は、アプリケーションから資源の割当および資源の解除が要請されれば、運営体系に資源の割当および資源の解除を要請する資源管理部と、前記運営体系から割当てられた資源に対応する仮想IDを生成する仮想ID生成部と、前記アプリケーションが前記仮想IDを用いて資源の解除を要請すれば、前記仮想IDの存在有無を検索するID検索部を含む。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

アプリケーションから資源の割当および資源の解除が要請されれば、運営体系に資源の割当および資源の解除を要請する資源管理部と、

前記運営体系から割当てられた資源に対応する仮想 ID を生成する仮想 ID 生成部と、

前記アプリケーションが前記仮想 ID を用いて資源の解除を要請すれば、前記仮想 ID の存在有無を検索する ID 検索部とを含む多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置。

【請求項 2】

前記資源管理部は、前記 ID 検索部の検索の結果、仮想 ID が存在する場合、前記仮想 ID に対応する資源の解除を要請する請求項 1 に記載の多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置。 10

【請求項 3】

前記仮想 ID は、前記運営体系から資源が割当てられる度に生成される固有 ID である請求項 1 に記載の多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置。

【請求項 4】

前記アプリケーションから伝送されたアプリケーション ID、仮想 ID、および割当てられた資源を含むテーブルを作成し、所定の資源が解除された場合に前記解除された資源が含まれたテーブルを削除する制御部と、 20

前記作成されたテーブルを格納する格納部とを含む請求項 1 に記載の多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置。

【請求項 5】

アプリケーションから資源の割当が要請されれば、前記要請により運営体系に資源の割当を要請するステップと、

前記割当てられた資源に対応する仮想 ID を生成するステップと、

前記アプリケーションから伝送されたアプリケーション ID、前記生成された仮想 ID および割当てられた資源を含むテーブルを作成するステップと、

前記アプリケーションに仮想 ID および割当てられた資源を伝送するステップとを含む多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法。 30

【請求項 6】

前記仮想 ID は、前記運営体系から資源が割当てられる度に生成される固有 ID である請求項 5 に記載の多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法。

【請求項 7】

アプリケーションから仮想 ID を用いて資源の解除が要請されるステップと、

前記アプリケーションから伝送された仮想 ID が含まれたテーブルが存在するかを検索するステップと、

前記検索の結果、前記伝送された仮想 ID が含まれたテーブルが存在すれば、運営体系に前記仮想 ID に対応する資源の解除を要請するステップと、 40

前記資源が解除されれば前記仮想 ID が含まれたテーブルを削除するステップとを含む多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法。

【請求項 8】

前記検索の結果、前記伝送された仮想 ID が含まれたテーブルが存在しない場合、資源の解除要請を行わない請求項 7 に記載の多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法。

【請求項 9】

前記仮想 ID は、前記アプリケーションに割当てられた資源に対応する固有 ID である請求項 7 に記載の多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法。

【請求項 10】

前記テーブルには所定の資源を用いるアプリケーションの ID、資源に対応する仮想 ID、および割当てられた資源が含まれる請求項 7 に記載の多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置およびその方法に関し、より詳しくは、多重ジャバアプリケーション環境で仮想 ID を用いて資源の割当および解除を行うことによって、資源をより効率的に用いることのできる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置およびその方法に関するものである。

10

【背景技術】

【0002】

ジャバ (Java (登録商標)) は、作成されたコードをプラットフォームに独立して行うオブジェクト指向プログラミング言語である。ここで、プラットフォームはプログラムが実行されるハードウェアまたは運営体系のようなソフトウェア環境をいう。ジャバで作成されたコードはジャバコンパイラ (Compiler) によりジャババイトコード (Byte Code) にコンパイルされ、ジャババイトコードは多様なハードウェア基盤プラットフォームにポータリングされたジャバ仮想マシン (Java (登録商標) Virtual Machine) により実行される。

20

【0003】

図 1 a と図 1 b は、従来の多重ジャバアプリケーションの構造を示すブロック図である。

【0004】

図 1 a に示すように、多重ジャバアプリケーション (Multiple Java (登録商標) Applications) 10 は、アプリケーション管理者 (Application Manager) 11、アプリケーション (Application) 12 a、12 b、12 n、ミドルウェア (Middleware) 13、ジャバ仮想マシン (Java (登録商標) Virtual Machine) 14、および運営体系 (Operating System) 15 を含んで構成される。ここで、アプリケーションは、ジャバアプリケーション 12 a、12 b、12 n を含む。

30

【0005】

アプリケーション管理者 11 は、アプリケーションの状態を切替えることによってアプリケーションのライフサイクル (Life Cycle) を管理する。例えば、OCA P (Open Cable Application Platform)、ACAP (Advanced open Cable Application Platform) 等のミドルウェア 13 を介してアプリケーション管理者 11 はアプリケーションの動作を管理する。

【0006】

ジャバ仮想マシン 14 は、アプリケーションに含まれたクラスをメモリに搭載してクラスの実行コードを運営体系 15 の命令コードに変換し、アプリケーションに割当てられた資源を管理する。

40

【0007】

ジャバ仮想マシン 14 は、図 1 b に示すようにクラスをロードするクラスローダ (Class Loader) 16、ジャストインタイムコンパイラ (Just-In-Time Compiler) 17、インタープリタ (Interpreter) 18、および実行時間にリソース管理を担当するガーベッジコレクタ (Garbage Collector: GC) 19 を含んで構成される。

【0008】

50

インタープリタ 18 はクラスの実行コードを実行するための構成要素であって、仮想マシン命令語を認識して命令を行う役割をし、ジャストインタイムコンパイラ 17 はクラス実行中に仮想マシン命令語を CPU の機械語にあらかじめ変換して速度を向上させる機能を行う。

【 0 0 0 9 】

ガーベッジコレクタ 19 は、ガーベッジオブジェクト（資源を用いないオブジェクト）を捜し出し、ガーベッジオブジェクトが用いたヒープ資源を運営体系に返還することによって、該当ヒープ資源が異なるアプリケーションにより再使用されるようにする。

【 0 0 1 0 】

換言すれば、ジャバプラットフォームにおいて資源管理は、ガーベッジコレクタ 19 によって実行されるものであり、ガーベッジコレクタ 19 はオブジェクト（Object）が参照（reference）されるか否かによって該当オブジェクトの資源の割当または資源の解除を管理するものである。ところで、多重ジャバアプリケーションのようなライフサイクルを有する多重ジャバアプリケーションにおいて複数のアプリケーションが実行される場合、アプリケーション上で資源の解除のための作業が正しく実行されなければ、アプリケーションが終了した状態でもガーベッジコレクタ 19 はオブジェクトが実際に有効（alive）な状態であるかを確認できないため、適切な資源の解除を行うことができないという問題点がある。

【 0 0 1 1 】

一方、アプリケーションで直接資源を割当および解除する場合を説明する。まず、アプリケーションで資源を用いる場合にジャバ仮想マシン 14 を介して直ちに運営体系 15 に資源の割当および解除を要請し、資源の割当に成功するようになれば直ちにその資源を用いるようになる。ここで、運営体系 15 から資源の割当および解除が失敗した場合には該当作業を終了する。

【 0 0 1 2 】

一方、上記のようにアプリケーションが直接資源を解除する場合とは異なって、資源の解除をしない場合には、オブジェクトが回収される時にガーベッジコレクタ 19 とファイナライザ（finalizer）によって資源が解除される。

【 0 0 1 3 】

すなわち、ガーベッジコレクト（Garbage collecting）が開始すれば、ガーベッジコレクタ 19 はすべてのオブジェクトの参照有無を検査して参照されないオブジェクトを検査する。この時、参照されないオブジェクトのうちのファイナライズが宣言されたオブジェクトを探す。

【 0 0 1 4 】

次に、ガーベッジコレクタ 19 は、ファイナライザ（finalizer）が宣言されているすべてのオブジェクト（object）のファイナライズを実行した後、ファイナライズによってさらに生成される参照されないオブジェクトを再び調べる。次に、参照されないオブジェクトをすべて解除してガーベッジコレクタ 19 の動作を終了する。

【 0 0 1 5 】

しかし、ジャバアプリケーションで明らかに資源を回収しない場合、用いた資源はただガーベッジコレクタ 19 によってのみ回収され、ここで、アプリケーションが終了した後にもガーベッジコレクタ 19 が動作しなかったとすれば、再使用可能な資源が動作の終わったアプリケーションによって先に獲得されて用いることができないという問題点がある。

【 0 0 1 6 】

一方、従来の資源管理インターフェースを介して資源が管理される例を説明する。まず、アプリケーションが終了する場合、アプリケーション管理者 11 の資源管理インターフェースに該当アプリケーションの資源を解除することを要求し、資源管理インターフェースは該当アプリケーションの資源を解除する役割を行う。

【 0 0 1 7 】

この時、各アプリケーションが用いる資源は、各々のオブジェクトによって維持および用いられる。したがって、資源管理インターフェースは資源を有するオブジェクトとオブジェクトが属しているアプリケーションに対する情報とを有し、アプリケーションの終了時に該当アプリケーションのオブジェクトが有する資源を呼び出してシステムに返却するのが一般的な方法である。

【0018】

しかし、オブジェクトを介して資源を解除する場合にJNIのような方法を介してネイティブ(native)資源を解除するべきであるが、これはスタックマシン(stack machine)のジャバの場合に新しいスタックを積んで進めなければならない作業であるため性能の低下をもたらし得る。しかし、性能の低下を避けるために、アプリケーションのIDとオブジェクトではない資源のみ資源管理インターフェースが維持するようになれば、資源管理インターフェースが資源を解除した後にもオブジェクトには資源に対する情報が残るようになる。

10

【0019】

ここで、アプリケーションIDと資源のみを管理する資源管理インターフェースがアプリケーションの終了と共に該当アプリケーションの資源を強制的に運営体系に返却する場合、運営体系は返却された資源を他のアプリケーションに再使用することができるように割当てる。

【0020】

この時、ガーベッジコレクタが実行されてファイナライザが実行され、オブジェクトに記録されていた資源を再び解除するようになれば、再使用されていた資源を用いていたアプリケーションは誤作動するようになる問題点がある。

20

【0021】

特許文献1は、アプリケーションに資源を提供する少なくとも1つ以上のライブラリ部とコーネル部とを備えるアプリケーション実行装置であり、前記コーネル部はアプリケーションが終了すれば終了したアプリケーションがいずれであるかを資源に提供する各ライブラリ部に通知する手段を有し、前記各ライブラリ部は通知手段から通知された終了アプリケーションに提供した資源を回収する回収手段を含むアプリケーション実行装置を開示しているが、これはアプリケーションが終了すればライブラリ部によって不要な資源が回収されるものであって、仮想IDを用いて資源を割当および解除する技術は全く開示されていない。

30

【特許文献1】日本公開特許第2002-259146号公報

【発明の開示】

【発明が解決しようとする課題】

【0022】

本発明は、多重ジャバアプリケーション環境において仮想IDを用いて資源の割当および解除を行うことにより、資源をより効率的に利用することができるようにすることにある。

【0023】

本発明の目的は、以上で言及した目的に制限されず、言及していないまた他の目的は下記記載によって当業者が明確に理解できるものである。

40

【課題を解決するための手段】

【0024】

前記目的を達成するために、本発明の一態様に係わるジャバアプリケーション環境で仮想IDを用いて資源を管理する装置は、アプリケーションから資源の割当および資源の解除が要請されれば、運営体系に資源の割当および資源の解除を要請する資源管理部と、前記運営体系から割当てられた資源に対応する仮想IDを生成する仮想ID生成部と、前記アプリケーションが前記仮想IDを用いて資源の解除を要請すれば、前記仮想IDの存在有無を検索するID検索部とを含む。

【0025】

50

また、本発明の他の態様に係わる多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する方法は、アプリケーションから資源の割当が要請されれば、前記要請により運営体系に資源の割当を要請するステップと、

前記割当てられた資源に対応する仮想IDを生成するステップと、前記アプリケーションから伝送されたアプリケーションID、前記生成された仮想IDおよび割当てられた資源を含むテーブルを作成するステップと、

前記アプリケーションに仮想IDおよび割当てられた資源を伝送するステップとを含む。

【0026】

また、本発明のまた他の態様に係わる多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する方法は、アプリケーションから仮想IDを用いて資源の解除が要請されるステップと、前記アプリケーションから伝送された仮想IDが含まれたテーブルが存在するかを検索するステップと、前記検索の結果、前記伝送された仮想IDが含まれたテーブルが存在すれば、運営体系に前記仮想IDに対応する資源の解除を要請するステップと、前記資源が解除されれば前記仮想IDが含まれたテーブルを削除するステップとを含む。

10

【発明の効果】

【0027】

本発明の多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する装置およびその方法によれば、次のような効果が1つあるいはそれ以上ある。

20

【0028】

多重ジャバアプリケーション環境で仮想IDを用いて資源の割当および解除を行うことによって、重複解除を行う誤りを防止する利点がある。

【0029】

また、多重ジャバアプリケーション環境でアプリケーション終了時にオブジェクトではない使用された資源だけを解除させることによって、資源が不足した環境で早く資源を再使用することができる利点がある。

【発明を実施するための最良の形態】

【0030】

その他、実施形態の具体的な事項は、詳細な説明および図面に含まれている。

30

【0031】

本発明の利点および特徴、そしてそれらを達成する方法は、添付する図面と共に詳細に後述する実施形態を参照すれば明確になる。しかし、本発明は以下にて開示する実施形態に限定されず、互いに異なる多様な形態によって実施され、単に本実施形態は本発明の開示が完全なものとなるようにし、本発明が属する技術分野で通常の知識を有する者に発明の範疇を完全に知らせるために提供されるものであって、本発明は請求項の範疇によってのみ定義されるものである。明細書の全体に亘り、同一参照符号は同一構成要素を示す。

【0032】

以下、添付した図面を参照して、本発明の好ましい実施形態を詳細に説明する。

【0033】

図2は、本発明の一実施形態に係わる多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する装置において、アプリケーションによって資源が用いられる例を示す図面である。

40

【0034】

多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する装置は、少なくとも1つ以上のアプリケーション210、220を用いて資源を用いることができる。ここで、アプリケーション210、220は、ライフサイクル(Life Cycle)を有する多重ジャバアプリケーション環境でのジャバアプリケーションであり得るが、例えば、エクセレット(Xlet)、ミドレット(Midlet)またはアプレット(Applet)であり得る。

50

【 0 0 3 5 】

また、アプリケーション 2 1 0、2 2 0 によって用いられる資源は、メモリ資源 2 5 1、ファイル資源 2 5 2、およびネットワーク資源 2 5 3 のうちの少なくともいずれか 1 つであり得る。

【 0 0 3 6 】

図に示すように、第 1 アプリケーション 2 1 0 が 2 つのメモリ資源 2 5 1 および 1 つのネットワーク資源 2 5 3 を用い、第 2 アプリケーション 2 2 0 が 1 つのメモリ資源 2 5 1、2 つのファイル資源 2 5 2 および 1 つのネットワーク資源 2 5 3 を用いていることを示す。

【 0 0 3 7 】

すなわち、各々のアプリケーション 2 1 0、2 2 0 は互いに異なる複数の資源を同時に用いることができ、同一の資源を複数用いることができる。また、1 つのアプリケーションによって、使用中である資源をまた他のアプリケーションが用いることもできる。

【 0 0 3 8 】

アプリケーションは随時発生して消滅し得るが、特定アプリケーションによって占有された資源が該当アプリケーションの消滅によって解除されなければ、以後に発生したアプリケーションはその資源を用いられない場合もある。

【 0 0 3 9 】

よって、本発明の仮想 ID を用いて資源を管理する装置は、各アプリケーションに割当てられた資源に対する使用および解除有無に対する情報を仮想 ID (Virtual ID: VID) を用いて管理することによって、アプリケーションから資源の解除要請時に直ちに該当資源を解除することによって、他のアプリケーションが該当資源を再使用することができるようにするものである。

【 0 0 4 0 】

例えば、ジャバ環境において、仮想 ID を用いて資源を管理する装置は、所定アプリケーションが資源の割当てを要請する場合、運営体系から資源が割当てられた後に仮想 ID を生成し、生成された仮想 ID および割当てられた資源をテーブルに格納した後に該当アプリケーションで資源および生成された仮想 ID を伝送する。

【 0 0 4 1 】

また、アプリケーションが資源を解除する場合の運営体系に該当資源の解除を要請した後、該当資源に対応する仮想 ID が格納されたテーブルを削除することによって、他のアプリケーションが資源を再使用することができるようにする。以下、図 3 での多重ジャバアプリケーション環境に対する説明を後述する。

【 0 0 4 2 】

図 3 は、本発明の他の実施形態に係わる多重ジャバアプリケーション環境の構造を示すブロック図である。

【 0 0 4 3 】

多重ジャバアプリケーション 3 0 0 は、アプリケーション管理者 3 1 0、アプリケーション (3 2 1、3 2 2、および 3 2 n)、ミドルウェア 3 3 0、ジャバ仮想マシン 3 4 0、資源管理インターフェース 3 5 0、および運営体系 3 6 0 を含んで構成される。ここで、アプリケーションはジャバアプリケーションを含む。

【 0 0 4 4 】

アプリケーション管理者 3 1 0 は、アプリケーションの状態を切替えることによってアプリケーションのライフサイクルを管理する。ここで、アプリケーション管理者 3 1 0 は、OCA P (Open Cable Application Platform)、ACA P (Advanced open Cable Application Platform) 等のミドルウェア 3 3 0 を介してアプリケーションの動作を管理することができる。

【 0 0 4 5 】

ジャバ仮想マシン 3 4 0 は、アプリケーションに含まれたクラスをメモリに搭載し、

10

20

30

40

50

クラスの実行コードを運営体系 360 の命令コードに変換する。

【0046】

また、ジャバ仮想マシン 340 は、多重ジャバアプリケーション 300 に存在するアプリケーションの固有 ID を生成する。ここで、アプリケーション ID 生成部 351 は、アプリケーションのスレッド (Thread) ID を用いてアプリケーション ID を生成することができる。

【0047】

資源管理インターフェース 350 は、ライフサイクルを有する多重ジャバアプリケーション 300 上での資源の割当および資源の解除を行う。

【0048】

例えば、資源管理インターフェース 350 は、割当てられた資源に対応する仮想 ID を生成した後、資源の割り当てを要請したアプリケーションから伝送されたアプリケーション ID、仮想 ID、および運営体系 360 から割当てられた資源を 1 つのテーブルとして作成および格納する。

【0049】

次に、該当アプリケーションに割当てられた資源、および生成された仮想 ID を伝送する。

【0050】

この後、所定アプリケーションが仮想 ID を用いて資源の解除を要請すれば、運営体系 360 に伝送された仮想 ID に対応する資源の解除を要請し、該当仮想 ID が含まれたテーブルを削除する。

【0051】

また、資源管理インターフェース 350 は、多重ジャバアプリケーション 300 内に別に備えられることもでき、ジャバ仮想マシン 340 内に備えられることもできる。本発明では多重ジャバアプリケーション 300 内に別に備えられた構成を例にあげて説明する。以下、図 4 で資源管理インターフェース 350 に対する説明を後述する。

【0052】

図 4 は、本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置の内部ブロック図である。

【0053】

図に示すように、資源管理インターフェース 350 は、仮想 ID 生成部 351、格納部 352、ID 検索部 353、資源管理部 354、および制御部 355 を含んで構成される。

【0054】

この時、本実施形態で用いられる「～部」という用語は、ソフトウェアまたは FPG A または ASIC のようなハードウェア構成要素を意味し、「～部」はある役割を行う。ところが「～部」は、ソフトウェアまたはハードウェアに限定される意味ではない。「～部」は、アドレッシングできる格納媒体にあるように構成することもでき、1 つまたはそれ以上のプロセッサを再生させるように構成することもできる。したがって、一例として「～部」はソフトウェア構成要素、オブジェクト指向ソフトウェア構成要素、クラス構成要素およびタスク構成要素のような構成要素と、プロセス、関数、属性、プロシージャ、サブルーチン、プログラムコードのセグメント、ドライバ、ファームウェア、マイクロコード、回路、データ、データベース、データ構造、テーブル、アレイ、および変数を含む。構成要素と「～部」の中で提供されている機能は、さらに小さい数の構成要素および「～部」に結合したり、追加的な構成要素と「～部」にさらに分離したりすることができる。それだけでなく、構成要素および「～部」は、デバイスまたは保安マルチメディアカード内の 1 つまたはそれ以上の CPU を再生させるように実現することもできる。

【0055】

仮想 ID 生成部 351 は、所定アプリケーションに資源を割当てる時、該当資源に付与される固有な仮想 ID (VID) を生成する。ここで、仮想 ID は、資源管理インターフ

10

20

30

40

50

エース350が運営体系360から割当てられた資源に対応するように生成されることであって、すなわち資源が割当てられる度に生成される固有IDである。また、資源は、メモリ資源、ファイル資源、およびネットワーク資源のうちの少なくとも1つを含むことができる。すなわち、アプリケーションは、メモリ資源、ファイル資源、およびネットワーク資源のうちの少なくとも1つを用いることができるものであり、各々の資源を複数用いることもできる。ここで、アプリケーションは、エクスレット、ミドレットまたはアプレットのように所定のライフサイクルを有するコンテインメントフレームワーク環境でのジャバアプリケーションであることができる。

【0056】

例えば、所定アプリケーションから資源の割当てが要請されれば、運営体系360から資源が割当てられて、仮想ID生成部351は割当てられた資源に対する仮想IDを生成する。

【0057】

格納部352は、仮想ID生成部351が生成したアプリケーションIDおよび仮想ID、運営体系360から割当てられた資源、およびアプリケーションを介して伝送されたアプリケーションIDが対応して作成されたテーブルを格納する。ここで、格納部352に格納されたテーブルは、資源の割当ておよび解除されることによって更新され得る。以下、図5で格納部352に格納されたテーブルに対する説明を後述する。

【0058】

ここで、格納部352は、ハードディスク、フラッシュメモリ、CFカード(Compact Flash(登録商標)Card)、SDカード(Secure Digital Card)、SMカード(Smart Media(登録商標)Card)、MMCカード(Multimedia Card)またはメモリスティック(Memory Stick)等の情報の入出力が可能なモジュールをいう。

【0059】

ID検索部353は、アプリケーションおよびガーベッジコレクションが仮想ID(またはアプリケーションID)を用いて資源の解除を要請した場合、格納部352に格納されたテーブル内に伝送された仮想ID(またはアプリケーションID)が存在するかを検索する。

【0060】

例えば、アプリケーションおよびガーベッジコレクションが仮想IDを用いて所定の資源の解除を要請する場合、ID検索部353は伝送された仮想IDが含まれたテーブルを検索する。

【0061】

また、アプリケーションがすべての資源の解除を要請する場合、ID検索部353はすべての資源の解除を要請したアプリケーションのIDが含まれたテーブルを検索する。

【0062】

資源管理部354は、アプリケーションの資源の割当て要請によって運営体系360から所定の資源が割当てられる。

【0063】

また、資源管理部354は、アプリケーションおよびガーベッジコレクションから所定の資源の解除が要請されれば、ID検索部353の検索結果に基づいて該当資源の解除有無を決定する。

【0064】

例えば、アプリケーションおよびガーベッジコレクションが仮想IDを用いて所定の資源の解除を要請すれば、資源管理部354はID検索部353から仮想IDの存在有無を伝送される。

【0065】

この時、検索の結果、該当仮想IDが存在する場合、資源管理部354は運営体系360に該当資源の解除を要請し、検索の結果、該当仮想IDが存在しない場合資源管理部3

10

20

30

40

50

54は運営体系360に該当資源の解除を要請しない。

【0066】

制御部355は、アプリケーションから伝送されたアプリケーションID、仮想ID生成部351を介して生成された仮想ID、および資源管理部354を介して割当てられた資源が対応されているテーブルを作成し、資源管理部354により所定の資源が解除されれば、解除された資源が含まれたテーブルを削除する。

【0067】

また、制御部355は、資源管理インターフェース350を構成する各機能性ブロック(351~354)の動作を制御する。

【0068】

図5は、本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する装置が管理するテーブルの例を示す図面である。

【0069】

図に示すように、資源管理インターフェース350の格納部352には、複数のテーブル510、520が格納されており、テーブル内には資源を要請したアプリケーションのID情報521、仮想ID情報522、および割当てられた資源523が格納されている。

【0070】

例えば、資源管理インターフェース350がアプリケーションから所定の資源の割当てを要請されれば、資源管理インターフェース350の資源管理部354は運営体系360に資源の割当てを要請して資源が割当てられ、仮想ID生成部351は割当てられた資源に対する固有な仮想IDを生成する。ここで、アプリケーションIDは資源を要請したアプリケーションから伝送される。

【0071】

次に、制御部355は、アプリケーションID521、仮想ID522、および資源523を1つのテーブル(例えば、テーブル2520)に作成した後、格納部352に格納する。

【0072】

図6は、本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想IDを用いて資源を管理する方法中の資源を割当てる過程を示すフローチャートである。

【0073】

まず、アプリケーション321が資源管理インターフェース350に資源の割当てを要請すれば(S600)、資源管理インターフェース350の資源管理部354は運営体系360に資源の割当てを要請する(S610)。この時、アプリケーション321は自身のアプリケーションIDを共に伝送する。

【0074】

次に、制御部355は、運営体系360から資源が割当てられたかを判断する(S620)。判断の結果、資源が割当てられた場合、仮想ID生成部351は割当てられた資源に対応する固有IDを生成する(S630)。

【0075】

次に、制御部355は、アプリケーションID、仮想ID、および運営体系360から割当てられた資源を含むテーブルを作成した後、作成されたテーブルを格納部352に格納する(S640)。

【0076】

次に、割当てられた資源およびそれに対応する仮想IDを該当アプリケーション321に伝送する(S650)。

【0077】

一方、運営体系360から資源が割当てられたかを判断した結果、資源の割当てがなされない場合(S620)、制御部355は資源管理部354を介して再要請を行ったり、アプリケーション321に資源の割当てが失敗したことを知らせたりする。

10

20

30

40

50

【 0 0 7 8 】

図 7 a と図 7 b は、本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法中の資源を解除する過程を示すフローチャートである。ここで、図 7 a はアプリケーションから資源の解除要請された場合、資源管理インターフェースで資源の解除を行う過程を示す図面であり、図 7 b はガーベッジコレクタから資源の解除が要請された場合、資源管理インターフェースで資源の解除を行う過程を示す図面である。

【 0 0 7 9 】

図 7 a を参照して、アプリケーションの要請により資源の解除を行う過程を説明する。10
先ず、アプリケーション 3 2 1 が仮想 ID を用いて資源管理インターフェース 3 5 0 に資源の解除を要請すれば (S 7 1 0)、資源管理インターフェース 3 5 0 の ID 検索部 3 5 3 はアプリケーション 3 2 1 から伝送された仮想 ID が含まれたテーブルを検索する (S 7 1 2)。ここで、アプリケーション 3 2 1 が仮想 ID を用いて資源の解除を要請するのは、複数のアプリケーションで同一の資源を用いる場合、所定の資源を用いる各々のアプリケーションは資源に対する固有仮想 ID を有し、仮想 ID で資源の解除を要請する場合、資源管理インターフェース 3 5 0 は該当仮想 ID に対する資源のみを解除することによって、他のアプリケーションが用いている他の仮想 ID で提供されていた資源は解除しないようにするためである。

【 0 0 8 0 】

次に、資源管理部 3 5 4 は、ID 検索部 3 5 3 を介して検索された仮想 ID が含まれた20
テーブルを参照し、対応する資源に対する解除を運営体系 3 6 0 に要請する (S 7 1 4)

【 0 0 8 1 】

次に、制御部 3 5 5 は、該当資源が解除されたかを判断し (S 7 1 6)、判断の結果、資源が解除された場合、該当資源が格納されたテーブルを格納部 3 5 2 で削除する (S 7 1 8)。

【 0 0 8 2 】

一方、該当資源の解除有無を判断した結果、資源が解除されていない場合 (S 7 1 6)、資源管理部 3 5 4 は運営体系 3 6 0 に該当資源の解除を再び要請する。

【 0 0 8 3 】

図 7 b を参照して、ガーベッジコレクタの要請により資源の解除を行う過程を説明する

先ず、ガーベッジコレクタで仮想 ID を用いて資源管理インターフェース 3 5 0 に資源の解除を要請すれば (S 7 5 0)、資源管理インターフェース 3 5 0 の ID 検索部 3 5 3 はガーベッジコレクタから伝送された仮想 ID が含まれたテーブルが存在するかを検索する (S 7 5 2)。ここで、ガーベッジコレクタが仮想 ID を用いて資源の解除を要請するのは、複数のアプリケーションで同一の資源を用いる場合、所定の資源を用いる各々のアプリケーションは資源に対する固有仮想 ID を有して、仮想 ID に資源の解除を要請する場合、資源管理インターフェース 3 5 0 は該当仮想 ID に対する資源だけを解除することによって、他のアプリケーションが用いている他の仮想 ID で提供されていた資源は解除40
しないためである。

【 0 0 8 4 】

検索の結果、ガーベッジコレクタが伝送した仮想 ID が含まれたテーブルが存在すれば (S 7 5 4)、資源管理部 3 5 4 は検索された仮想 ID が含まれたテーブルを参照して、対応する資源に対する解除を運営体系 3 6 0 に要請する (S 7 5 6)。

【 0 0 8 5 】

次に、制御部 3 5 5 は該当資源が解除されたかを判断し (S 7 5 8)、判断の結果、資源が解除された場合、該当資源が含まれたテーブルを格納部 3 5 2 で削除する (S 7 6 0)。

【 0 0 8 6 】

一方、該当資源の解除有無を判断した結果、資源が解除されていない場合（Ｓ７５８）、資源管理部３５４は運営体系３６０に該当資源の解除を再び要請する。

【００８７】

また、検索の結果、ガーベッジコレクタが伝送した仮想ＩＤが含まれたテーブルが存在しない場合（Ｓ７５４）、制御部３５５はすでに解除された資源に対してガーベッジコレクタが解除を要請することと判断して、ガーベッジコレクタの要請を行わない（Ｓ７６２）。

【００８８】

したがって、資源管理インターフェース３５０は、仮想ＩＤを用いてすでに解除された資源に対する重複解除を行わないことによって、同一の資源を用いているアプリケーションのプログラムが非正常に動作するようになる誤りが発生しないようにする。

【００８９】

図８は、本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想ＩＤを用いて資源を管理する方法中のすべての資源を解除する過程を示すフローチャートである。

【００９０】

まず、アプリケーション３２１が資源管理インターフェース３５０に用いたすべての資源の解除を要請する（Ｓ８００）。この時、アプリケーション３２１は自身のアプリケーションＩＤを共に伝送する。

【００９１】

次に、資源管理インターフェース３５０のＩＤ検索部３５３は資源解除を要請したアプリケーションのＩＤが含まれたテーブルを検索する（Ｓ８１０）。

【００９２】

検索の結果、アプリケーションのＩＤが含まれたテーブルが存在すれば（Ｓ８２０）、資源管理部３５４は検索されたテーブルを参照して、アプリケーションＩＤに対応する資源に対する解除を運営体系３６０に要請する（Ｓ８３０）。

【００９３】

次に、制御部３５５は該当資源が解除されたかを判断し（Ｓ８４０）、判断の結果、資源が解除された場合、該当資源が格納されたテーブルを格納部３５２で削除する（Ｓ８５０）。

【００９４】

一方、該当資源の解除有無を判断した結果、資源が解除されない場合（Ｓ８４０）、資源管理部３５４は運営体系３６０に該当資源の解除を再び要請する。

【００９５】

図９は、本発明のまた他の実施形態により多重ジャバアプリケーション環境で資源の割当および解除を行うアプリケーション、資源管理インターフェース、および運営体系の動作を示す図面である。

【００９６】

アプリケーションＡ９１０が資源管理インターフェース３５０に資源の割当を要請すれば、資源管理インターフェース３５０は運営体系３６０に資源の割当を要請する。この時、アプリケーションＡ９１０は自身のアプリケーションＩＤを共に伝送する。

【００９７】

次に、資源管理インターフェース３５０に運営体系３６０から資源ａが割当てられれば、資源管理インターフェース３５０は資源を要請したアプリケーションのＩＤ（例えば、アプリケーションＡＡ）９３１、割当てられた資源ａに対応する仮想ＩＤ（例えば、０１０１）９３２、および割当てられた資源ａ９３３を含むテーブル（例えば、テーブル１）を作成して格納する。

【００９８】

次に、アプリケーションＡ９１０に仮想ＩＤ（例えば、０１０１）９３２、および資源ａ９３３を伝送する。

10

20

30

40

50

【0099】

この後、アプリケーションA910は、仮想ID（例えば、0101）932を用いて資源管理インターフェース350に資源の解除を要請する。ここで、資源管理インターフェース350は伝送された仮想ID（例えば、0101）932が含まれたテーブルを検索する。

【0100】

検索の結果、伝送された仮想ID（例えば、0101）932が含まれたテーブル（例えば、テーブル1）が存在すれば、資源管理インターフェース350は運営体系360に伝送された仮想IDに対応する資源の解除を要請する。

【0101】

次に、資源管理インターフェース350は伝送された仮想ID（例えば、0101）932が含まれたテーブル（例えば、テーブル1）を削除する。

【0102】

一方、アプリケーションB920が資源管理インターフェース350に資源の割当を要請すれば、資源管理インターフェース350は運営体系360に資源の割当を要請する。

【0103】

ここで、運営体系360は、以前に解除された資源aを資源管理インターフェース350に割当てて。

【0104】

次に、資源管理インターフェース350は、資源を要請したアプリケーションのID（例えば、アプリケーションBB）951、割当てられた資源aに対応する仮想ID（例えば、0501）952、および割当てられた資源a953を含むテーブル（例えば、テーブル2）を作成して格納する。

【0105】

次に、アプリケーションB920に仮想ID（例えば、0501）952、および資源a953を伝送する。

【0106】

この後、ガーベッジコレクタが仮想ID（例えば、0101）を用いて資源管理インターフェース350に資源の解除を要請すれば、資源管理インターフェース350は伝送された仮想ID（例えば、0101）が含まれたテーブルが存在するかを判断する。

【0107】

判断の結果、該当仮想ID（例えば、0101）が含まれたテーブルが存在しなければ、資源管理インターフェース350はすでに解除された資源に対してガーベッジコレクタが解除を要請することと判断し、ガーベッジコレクタの要請を行わない。

【0108】

ここで、すでに解除された仮想ID（例えば、0101）に対応する資源aに対する解除を行わないことによって、アプリケーションBで用いている資源aが解除されなくなるため、アプリケーションBのプログラムが非正常に動作するようになる誤りは発生しない。

【0109】

以上、添付した図面を参照して本発明の実施形態を説明したが、本発明が属する技術分野で通常の知識を有する者であれば、本発明がその技術的思想や必須の特徴を変更せず、他の具体的な形態によって実施することができるということを理解できる。したがって、以上で記述した実施形態はすべての面で例示的なものであり、限定的なものではないことを理解しなければならない。

【図面の簡単な説明】

【0110】

【図1a】従来の多重ジャバアプリケーションの構造を示すブロック図である。

【図1b】従来の多重ジャバアプリケーションの構造を示すブロック図である。

【図2】本発明の一実施形態に係わる多重ジャバアプリケーション環境で仮想IDを用い

10

20

30

40

50

て資源を管理する装置において、アプリケーションによって資源が用いられる例を示す図である。

【図 3】本発明の他の実施形態に係わる多重ジャバアプリケーション環境の構造を示すブロック図である。

【図 4】本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置の内部ブロック図である。

【図 5】本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する装置が管理するテーブルの例を示す図である。

【図 6】本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法中の資源を割当てる過程を示すフローチャートである。

10

【図 7 a】本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法中の資源を解除する過程を示すフローチャートである。

【図 7 b】本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法中の資源を解除する過程を示すフローチャートである。

【図 8】本発明のまた他の実施形態に係わる多重ジャバアプリケーション環境で仮想 ID を用いて資源を管理する方法中のすべての資源を解除する過程を示すフローチャートである。

【図 9】本発明のまた他の実施形態により多重ジャバアプリケーション環境で資源の割当および解除を行うアプリケーション、資源管理インターフェース、および運営体系の動作を示す図である。

20

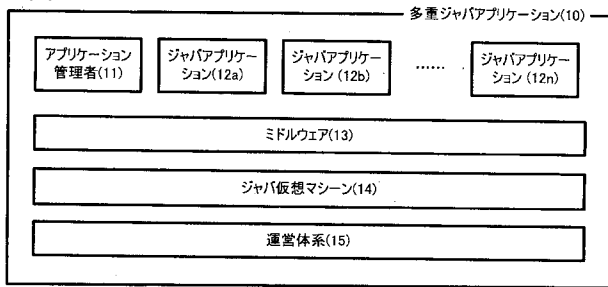
【符号の説明】

【 0 1 1 1 】

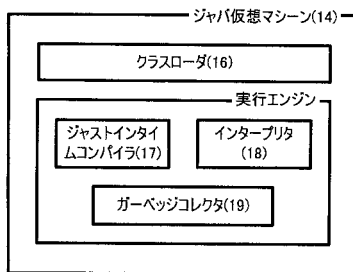
3 0 0	多重ジャバアプリケーション
3 1 0	アプリケーション管理者
3 4 0	ジャバ仮想マシン
3 5 0	資源管理インターフェース
3 5 1	仮想 ID 生成部
3 5 2	格納部
3 5 3	ID 検索部
3 5 4	資源管理部
3 5 5	制御部

30

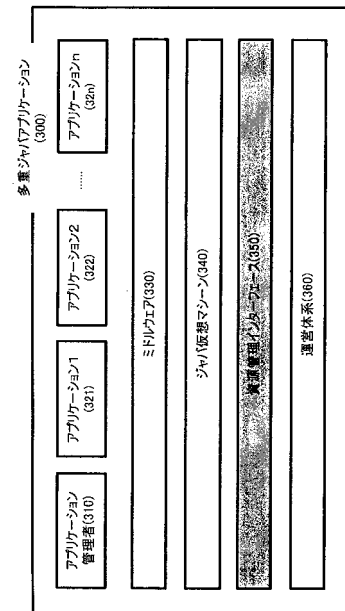
【図 1 a】



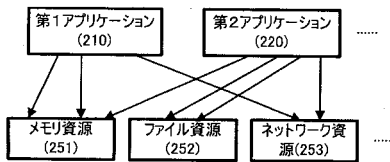
【図 1 b】



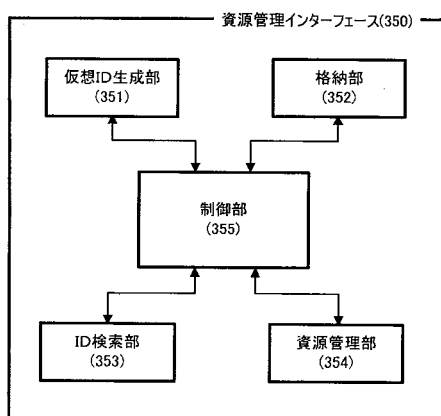
【図 3】



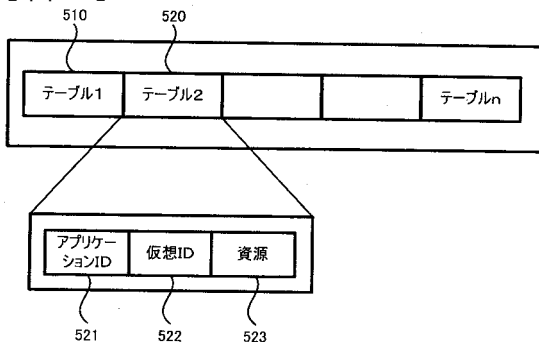
【図 2】



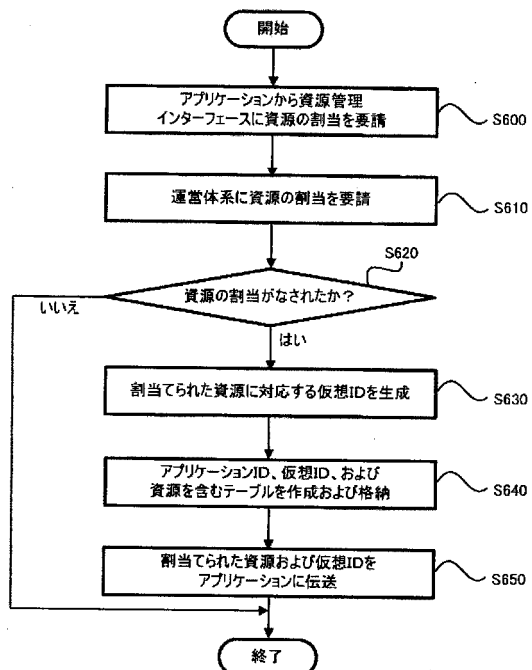
【図 4】



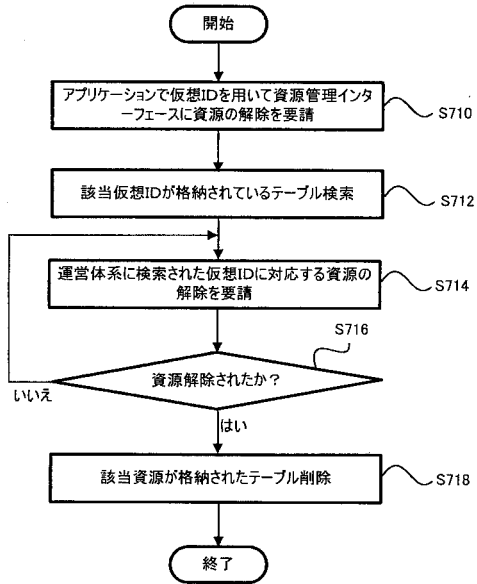
【図 5】



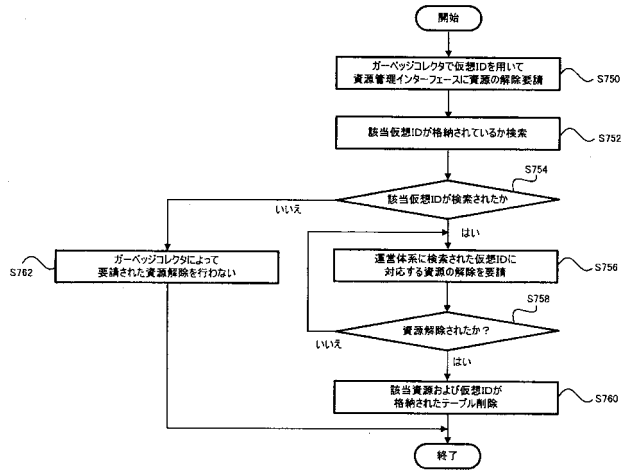
【図 6】



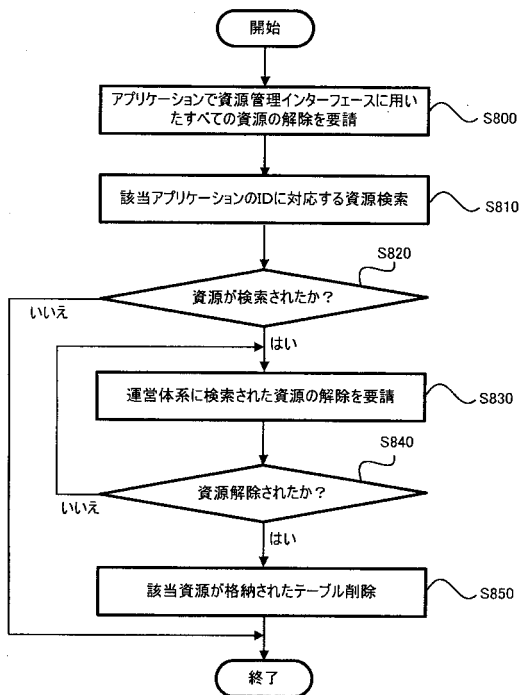
【図 7 a】



【図 7 b】



【図 8】



【図 9】

