



**【特許請求の範囲】****【請求項 1】**

コンピュータシステムを制御する OS の一部として、割り込みを発生する割り込み資源からの割り込みを制御する割り込み制御装置であって、

割り込み資源のうち OS が管理する第 1 タイプの割り込み資源および OS が管理しない第 2 タイプの割り込み資源のそれぞれに対応するアドレス情報を保持するテーブル手段と

第 1 タイプの割り込み資源が発生した割り込みに対しては、テーブル手段のアドレス情報に基づいて第 1 タイプの割り込み資源に共通する管理処理を含む第 1 の制御処理を起動し、第 2 タイプの割り込み資源が発生した割り込みに対しては、テーブル手段のアドレス情報に基づいて前記管理処理を含まない第 2 の制御処理を起動する起動手段と

10

を備えることを特徴とする割り込み制御装置。

**【請求項 2】**

前記割り込み制御装置は、さらに、アプリケーションから特定の割り込み資源とアドレス情報とを示す通知を受け付ける受付手段を有し、

前記テーブル手段は、受付手段によって受け付けられた通知に従って第 2 タイプの割り込み資源に対応するアドレス情報を保持する

ことを特徴とする請求項 1 記載の割り込み制御装置。

**【請求項 3】**

前記起動手段は、前記第 2 の制御処理において、コンテキストの退避と、割り込みを発生した第 2 タイプの割り込み資源に対応する割り込み処理と、退避していたコンテキストの復帰とを順に実行する

20

ことを特徴とする請求項 1 又は 2 記載の割り込み制御装置。

**【請求項 4】**

前記起動手段は、前記第 2 の制御処理において前記コンテキスト退避の開始前に他の割り込み入力を禁止し、前記第 2 の制御処理が完了するまでに前記割り込み入力の禁止を解除する

ことを特徴とする請求項 3 記載の割り込み制御装置。

**【請求項 5】**

前記割り込み制御手段は、さらに、仮想記憶用のページテーブルのうち実行中プロセスに対応するページデータをキャッシュする変換バッファと、

30

前記起動手段は、前記第 2 の制御処理において前記割り込み処理実行前に当該割り込み処理に対応するページデータを保持するよう前記変換バッファを更新する更新手段と

を備えることを特徴とする請求項 3 記載の割り込み制御装置。

**【請求項 6】**

前記受付手段は、さらにアプリケーションから、第 2 タイプの割り込み資源に対応する割り込み処理のプロセス ID を示す通知を受け付け、

前記更新手段は、受付手段によって受け付けられた通知に示されるプロセス ID に従って前記ページテーブルから前記ページデータを読み出して前記変換バッファに格納する

ことを特徴とする請求項 5 記載の割り込み制御装置。

40

**【請求項 7】**

前記テーブル手段は、さらに割り込み資源毎に割り込みレベルと前記第 1 タイプか第 2 タイプかを示す区分とを保持し、

前記起動手段は、発生した割り込みの割り込みレベルとテーブル手段に保持された区分とに基づいて第 1 タイプと第 2 タイプのいずれであるかを判別する

ことを特徴とする請求項 1 から 6 の何れかに記載の割り込み制御装置。

**【請求項 8】**

前記起動手段は、発生した割り込みの割り込みレベルと所定レベルの優先順位を比較して、前記第 1 タイプと第 2 タイプのいずれであるかを判別する

ことを特徴とする請求項 1 から 6 の何れかに記載の割り込み制御装置。

50

## 【請求項 9】

コンピュータシステムを制御する OS の一部として、割り込みを発生する割り込み資源からの割り込みを制御する割り込み制御方法であって、

アプリケーションから特定の割り込み資源とアドレス情報とを示す通知を受け付ける受付ステップと、

割り込み資源のうち OS が管理する第 1 タイプの割り込み資源に対応するアドレス情報を保持する管理テーブルに、受付手段によって受け付けられた通知に従って OS が管理しない第 2 タイプの割り込み資源に対応するアドレス情報を追加する追加ステップと、

第 1 タイプの割り込み資源が発生した割り込みに対しては、管理テーブルのアドレス情報に従って第 1 タイプの割り込み資源に共通する管理処理を含む第 1 の制御処理を起動し、第 2 タイプの割り込み資源が発生した割り込みに対しては、管理テーブルのアドレス情報に基づいて前記管理処理を含まない第 2 の制御処理を起動する起動ステップと

を有することを特徴とする割り込み制御方法。

10

## 【請求項 10】

請求項 9 記載の割り込み制御方法をコンピュータに実行させることを特徴とするプログラム。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

コンピュータシステムを制御する OS の一部として、割り込みを発生する割り込み資源からの割り込みを制御する割り込み制御装置に関する。

20

## 【背景技術】

## 【0002】

一般に、コンピュータシステムにおける割り込み制御は、OS において各割り込み資源からの割り込みを一元的に管理することで処理される。

図 10 は、従来におけるコンピュータシステムにおける割り込み発生時に、OS のカーネル空間からユーザプロセス空間への遷移関係を示す図である。同図のように、割り込みは、カーネル空間 602 内において、割り込みを受け付ける割り込み処理部からスケジューラによる割り込みのスケジュール処理の後、プロセス空間 601 内のユーザプロセスへ遷移する。

30

## 【0003】

図 11 は、従来のコンピュータシステムにおける割り込み制御装置の構成を示す図である。同図では、割り込みを発生する割り込み資源としての周辺デバイス、割り込みコントローラ、CPU、OS、アプリケーションプロセスを含む階層別に示している。図 12 は、図 11 における処理の流れを各構成要素のレジスタ構成に着目して示す動作説明図である。図 11 における動作を図 12 を用いて、より具体的に説明する。

## 【0004】

割り込みコントローラ (INTC) 502 が、各種周辺デバイス (I/O 装置) 501 a、501 b 等からの割り込み発生を検出し、該当割り込み要因に対応する要因番号レジスタ 502 a のフラグをセットする。当該割り込み要因がソフトウェア設定によりマスクされていなければ、割り込みコントローラ 502 は CPU 503 に割り込み発生を通知する。

40

## 【0005】

CPU 503 は、割り込みレベルに応じた割り込みベクタレジスタ 503 a を有し、割り込みコントローラ 502 からの割り込み通知を受理した後、該当割り込み要因がアサインされている割り込みベクタレジスタから、割り込みベクタを読み出し、OS が管理している割り込みエントリ関数 505 を実行する。この割り込みエントリ関数 505 を実行することにより、割り込み要因の判別や、ユーザの割り込みハンドラへ遷移する。その際、スケジューラ 506 は、他の割り込みや他のプロセスとの間でスケジュール調整を行い、調整後に割り込み要因に対応する割り込みハンドラ 507 a または 507 b 等を起動し、

50

割り込み要求に対応する処理を行う（または対応する処理を呼び出す）。

【 0 0 0 6 】

以上の動作により、従来の割り込み制御装置は一連の割り込み制御を行うことが一般的である。

ところで、一般的なOSは特定の割り込み資源だけをOSの管理外とする場合を想定しておらず、そうするには特別な機構が必要である。そのための特別な機構として特許文献1では、複数のOSと、OSから独立した割り込み管理プログラムによる実現方法が開示されている。

【 0 0 0 7 】

特許文献1によれば、OSから独立した割り込み管理プログラムが、一方のOSが管理する割り込み資源を横取りして他方のOSに振り分ける方法と、OSの実行状態により割り込みハンドラを起動するタイミングを決定してそれぞれの割り込みハンドラを起動するタイミングを決定する方法による、割り込み制御方法が開示されている。この割り込み制御方法によれば、発生した外部割り込みを割り込み管理プログラムで一旦処理し、複数のOSに割り込み処理を振り分け、振り分けた先のOSで割り込み処理を実行することが可能である。また割り込み処理が複数のOS間で多重化された場合も、割り込み管理プログラムで割り込み状態を管理することで、排他制御を可能としている。

【特許文献1】特開2001-216172号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 8 】

しかしながら、従来の割り込み制御装置によれば、特定の割り込み資源だけをOSの管理外とする場合に、割り込みが発生してから割り込みに対応するアプリケーションが実行されるまでの応答性、つまりリアルタイム性を保証することができないという問題がある。

具体的には、オペレーティングシステム(OS)上の特定の割り込み資源を、他のプログラムにより分離、独立して管理する場合には、以下のような問題がある。

【 0 0 0 9 】

(1) 特許文献1では、割り込み管理プログラムが、割り込み処理のOS間での多重化を調停するため、前記割り込み管理プログラムで割り込み状態を管理することが必須となる。例えば、図12に示したスケジューラ506が特許文献1では、さらに複雑なスケジュール調整を行う割り込み管理プログラムに相当する。この割り込み管理プログラムを新規に実装して処理を実行することにより、従来OSで実装されている割り込みエントリ関数に加えて割り込み状態の管理処理が増える分、リアルタイム性が悪くなる。

【 0 0 1 0 】

(2) 特許文献1に関わらず、特定の割り込み要因に対応するユーザの割り込みハンドラ関数を、OSのアドレス空間(カーネル空間)ではなくユーザプロセス空間に配置し、配置された割り込みハンドラ関数へ割り込み応答性を阻害することなく遷移する仕組みが従来存在しない。図10に示したカーネル空間602からプロセス空間601へ遷移する過程において、スケジューラの応答性はOS性能に依存する。汎用OSでは、割り込み発生後にアプリケーションへ遷移するまでの時間保証がない、すなわちリアルタイム保証がないことが課題になる。

【 0 0 1 1 】

(3) 特許文献1では、CPUの割り込み資源を複数のOSで分割管理するために、少なくとも2つ以上のOSが必要である構成であり、1つのOSとそのOS上で動作するアプリケーションプログラム間で割り込み資源分割を行うケースが想定されていない。

上記の課題に鑑み本発明は、汎用OSにおいて特定の割り込み資源からの割り込みに対してリアルタイム性を保証する割り込み制御装置、その制御方法およびプログラムを提供することを目的とする。特に、本発明は、組み込み機器に広く必要とされるリアルタイム性を汎用OS上で保証、実現する割り込み制御装置、その制御方法およびプログラムを提

10

20

30

40

50

供することを目的とする。

【課題を解決するための手段】

【0012】

上記課題を解決するために、本発明の割り込み制御装置は、コンピュータシステムを制御するOSとして、割り込みを発生する割り込み資源からの割り込みを制御する割り込み制御装置であって、割り込み資源のうちOSが管理する第1タイプの割り込み資源およびOSが管理しない第2タイプの割り込み資源のそれぞれに対応するアドレス情報を保持するテーブル手段と、第1タイプの割り込み資源が発生した割り込みに対しては、テーブル手段のアドレス情報に基づいて第1タイプの割り込み資源に共通する管理処理を含む第1の制御処理を起動し、第2タイプの割り込み資源が発生した割り込みに対しては、テーブル手段のアドレス情報に基づいて前記管理処理を含まない第2の制御処理を起動する起動手段とを備える。

10

【0013】

この構成によれば、第1タイプの割り込み資源に対してはリアルタイム性には劣るものの汎用性の高い割り込み処理を起動し、第2タイプの割り込み資源に対しては割り込みにすばやく応答しリアルタイム性に優れた割り込み処理を起動することができる。

ここで、前記割り込み制御装置は、さらに、アプリケーションから特定の割り込み資源とアドレス情報とを示す通知を受け付ける受付手段を有し、前記テーブル手段は、受付手段によって受け付けられた通知に従って第2タイプの割り込み資源に対応するアドレス情報を保持する構成としてもよい。

20

【0014】

この構成によれば、アプリケーションはリアルタイム性を必要とする任意の割り込み資源を第1タイプから第2タイプに自由に設定変更することができる。

ここで、前記起動手段は、前記第2の制御処理において、コンテキストの退避と、割り込みを発生した第2タイプの割り込み資源に対応する割り込み処理と、退避していたコンテキストの復帰とを順に実行する構成としてもよい。

【0015】

ここで、前記起動手段は、前記第2の制御処理において前記コンテキスト退避の開始前に他の割り込み入力を禁止し、前記第2の制御処理が完了するまでに前記割り込み入力の禁止を解除する構成としてもよい。

30

この構成によれば、第2タイプの割り込み資源に対応する割り込み処理においてリアルタイム性を要する処理が完了するまでは他の割り込み入力を禁止するので、リアルタイム性をより確実に確保することができる。

【0016】

ここで、前記割り込み制御手段は、さらに、仮想記憶用のページテーブルのうち実行中プロセスに対応するページデータをキャッシュする変換バッファと、前記起動手段は、前記第2の制御処理において前記割り込み処理実行前に当該割り込み処理に対応するページデータを保持するよう前記変換バッファを更新する更新手段とを備える構成としてもよい。

【0017】

ここで、前記受付手段は、さらにアプリケーションから、第2タイプの割り込み資源に対応する割り込み処理のプロセスIDを示す通知を受け付け、前記更新手段は、受付手段によって受け付けられた通知に示されるプロセスIDに従って前記ページテーブルから前記ページデータを読み出して前記変換バッファに格納する構成としてもよい。

40

この構成によれば、仮想記憶用のいわゆるTLB(Translation Look aside Buffer)と呼ばれる変換バッファを更新するので、割り込み処理をプロセス空間に容易に配置することができる。

【0018】

ここで、前記テーブル手段は、さらに割り込み資源毎に割り込みレベルと前記第1タイプか第2タイプかを示す区分とを保持し、前記起動手段は、発生した割り込みの割り込み

50

レベルとテーブル手段に保持された区分とに基づいて第１タイプと第２タイプのいずれであるかを判別する構成としてもよい。

ここで、前記起動手段は、発生した割り込みの割り込みレベルと所定レベルの優先順位を比較して、前記第１タイプと第２タイプのいずれであるかを判別する構成としてもよい。

#### 【００１９】

この構成によれば、任意の割り込みレベルを第１から第２タイプに変更し、あるいは、所定レベルよりも低い（または高い）割り込みレベルの割り込み資源を第２タイプに設定することができる。

#### 【発明の効果】

10

#### 【００２０】

以上説明したように、本発明の割り込み制御装置によれば、リアルタイム性には劣るものの汎用性の高い割り込み処理と、割り込みにすばやく応答しリアルタイム性に優れた割り込み処理とを混在させることができる。また、アプリケーションが指定する任意の割り込みレベルにおいて、第２タイプＯＳの管理処理を介さないでユーザー独自の割り込み処理を起動するので、制御方法の実現が可能である。

#### 【００２１】

また、アプリケーションが管理する第２タイプの割り込み資源とＯＳで管理する第１タイプの割り込み資源との間の割り込み処理の排他制御については、調停機構を別途設ける必要がなく、調停機構自体による処理遅延も発生しないため、応答性を阻害することなく排他制御が可能である。

20

さらに、アプリケーションが管理する割り込み処理関数をカーネル空間とは別空間となるアプリケーションのユーザープロセス空間に実装することを可能としている。第２タイプの割り込み資源に対応する割り込み処理、つまり割り込み処理関数（ユーザープロセス空間）への遷移は、リアルタイム性を保証することができる。

#### 【００２２】

また、本発明の割り込み制御方法およびプログラムについても上記と同様の手段、作用および効果を有する。

#### 【発明を実施するための最良の形態】

#### 【００２３】

30

#### （実施の形態１）

以下、本発明の割り込み制御方式の実施形態について、図面を参照しながら説明する。

#### < 割り込み制御装置の構成 >

図１は、本発明の実施の形態１における割り込み制御装置のシステム構成を示すブロック図である。同図では、アプリケーションと、ＯＳと、ＣＰＵと、割り込みコントローラと、外部デバイスと階層的に示している。図１に示した従来のシステム構成と異なる点は、アプリケーションが自プロセスからＯＳに対してユーザーで管理したい割り込みレベル、すなわちＯＳの管理外とする割り込みレベルを通知する機構としてユーザ管理割り込み番号通知部１１２を備える点と、アプリケーションが準備する拡張割り込みエントリ関数１０８と呼ぶ関数アドレスを通知する機構として拡張割り込みエントリ関数通知部１１３を備える点と、ＯＳ内に前記アプリケーションが準備する拡張割り込み関数のアドレスの通知を受け取る割り込みベクタ更新部１１０を備える点とである。拡張割り込みエントリ関数１０８およびユーザハンドラ１０９はＯＳの一部としてではなくアプリケーションとして準備される。

40

#### 【００２４】

図１に示した割り込み制御装置について、処理の流れを各構成要素のレジスタ構成に着目して示す図２を用いて、より具体的に説明する。

割り込みコントローラ１０２が、各種周辺デバイス（Ｉ／Ｏ装置）１０１ａ、１０１ｂ等からの割り込み発生を検出し、該当割り込み要因に対応する要因番号レジスタ１０２ａのフラグをセットする。当該割り込み要因がソフトウェア設定によりマスクされていなく

50

れば、割り込みコントローラ 102 は CPU 103 に割り込み発生を通知する。

#### 【0025】

CPU 503 は、割り込みレベルに応じた割り込みベクタレジスタ 103a を有している。割り込みベクタレジスタ 103a は優先度の高い順にレベル 0 からレベル n までの割り込みレベルに対応するアドレスを保持する割り込みレベルレジスタを備える。

図 2 の例では、割り込みベクタレジスタ 103a においてレベル 0 からレベル m までの割り込みレベルレジスタは第 1 タイプの割り込み用であり、それぞれ共通割り込みエントリ関数 105 の先頭アドレスを保持しているものとする。また、レベル m + 1 からレベル n までの割り込みベクタレジスタは第 2 タイプの割り込み用であり、それぞれ拡張割り込みエントリ関数 108 の先頭アドレスを保持しているものとする。ここで、第 1 タイプの割り込みとは、従来技術と同様に OS によって共通の管理処理（主としてスケジューラ 106 のスケジュール処理）を経由して割り込み処理（ユーザの割り込みハンドラ 107a、107b など）が起動される汎用性に優れた割り込みをいう。第 2 タイプの割り込みとは、アプリケーションが管理する割り込みであって上記の管理処理を経由しないで割り込み処理（ユーザの割り込みハンドラ 109a、109b など）が起動されるリアルタイム性を保証される割り込みをいう。

10

#### 【0026】

割り込みコントローラ 102 からの割り込み通知を受理した後、該当割り込み要因がアサインされている割り込みベクタレジスタのアドレスに従って共通割り込みエントリ関数 105、拡張割り込みエントリ関数 108 の何れかが呼び出される。同図の例では、第 1 タイプの割り込みである場合は、共通割り込みエントリ関数 105 が呼び出され、第 2 タイプの割り込みである場合は、拡張割り込みエントリ関数 108 が呼び出されることになる。

20

#### 【0027】

共通割り込みエントリ関数 105 の実行によって、割り込み要因の判別や、ユーザの割り込みハンドラ 107a や 107b への遷移がなされる。その際、スケジューラ 106 は、他の割り込みとの多重化調停や他のプロセスとのスケジュール調整を行い、調整後に割り込み要因に対応する割り込みハンドラ 107a または 107b 等を起動する。割り込みハンドラ 107a や 107b が呼び出されることにより割り込み要求に対応する割り込み処理が実行される（または呼び出される）。

30

#### 【0028】

一方、拡張共通割り込みエントリ関数 108 の実行によって、割り込み要因の判別や、ユーザの割り込みハンドラ 107a や 107b への遷移がなされる。その際、スケジューラ 106 の調整を経由しないので、割り込み要因に対応する割り込みハンドラ 109a や 109b が遅延することなく起動される。割り込みハンドラ 109a や 109b が呼び出されることにより割り込み要求に対応する割り込み処理が実行される（または呼び出される）。

#### 【0029】

<アプリケーションからの登録処理>

図 3 は、ユーザーアプリケーションで準備する割り込み処理ルーチン（つまり拡張割り込みエントリ関数 108）のアドレスを CPU 内の割り込みベクタレジスタへ登録する処理を示すフローチャートである。同図の手順は、リアルタイム性を必要とする任意の割り込み要因を第 2 タイプの割り込みに設定する場合に、例えばアプリケーションの起動時の最初に実行される。

40

#### 【0030】

まず、実行中のアプリケーション（以下単にアプリケーションと呼ぶ）は、OS が管理する割り込み資源の中から任意のリアルタイム性を必要とする任意の割り込み資源の資源番号を選択する（S301）。図 2 の例では、割り込み資源として、割り込み要因（周辺デバイス名称又は番号）および割り込みレベルを選択する。この場合、割り込み要因と割り込みレベルが 1 対 1 または多対 1 で対応していれば、選択すべき資源番号は割り込みレ

50

ベルだけでよい。アプリケーションは、さらに選択した割り込み資源の番号をOSに通知する(S302)。次に、前記選択した割り込みレベルに対応する割り込みベクタテーブルにアドレスを格納するための、拡張割り込みエントリ関数の名称(又は先頭アドレス)をOSに通知する(S303)。

#### 【0031】

この通知を受けてOS中の割り込みベクタ更新部110は、通知された資源番号に対応するベクタレジスタがベクタテーブル中にあれば(S304)、さらにアプリケーションは、拡張割り込みエントリ関数の先頭アドレス(自プロセス内の任意のアドレスでもよい)をOSに通知する(S305)。OSは通知されたアドレスをベクタテーブル内のベクタレジスタに格納し(S306)、本処理を終了する。もし、上記ステップS304において、アプリケーションから通知された資源番号に対応するベクタレジスタがベクタテーブル中になければエラー処理を行って(S307)本処理を終了する。

10

#### 【0032】

これによりアプリケーションが任意に選択した割り込み資源について、拡張割り込みエントリ関数のアドレスがベクタテーブルに格納されることになる。

#### < 割り込み制御処理 >

図4は、図3での登録処理後、実際に割り込みが発生した場合の割り込み制御処理を示すフローチャートである。

#### 【0033】

周辺I/Oなどの割り込み資源から割り込みが発生した場合、その割り込みを管理する割り込みコントローラ102が、該当割り込み要因の割り込み発生をCPUへ通知する(S401)。CPUが該当割り込み要因がアサインされている割り込みベクタレジスタへ遷移する(S402)。通知された割り込み要因に応じた、CPU内の割り込みベクタレジスタが選択される(S403)。

20

#### 【0034】

遷移先の割り込みベクタレジスタが、前記図3で登録された拡張割り込みエントリ関数108のアドレスである場合には、その拡張割り込みエントリ関数108が実行される(S405)。それ以外の割り込みベクタレジスタであれば、既存のOSの共通割り込みエントリ関数105が実行される(S404)。

この結果、図2に示したように、割り込みベクタレジスタ内に格納される割り込みエントリ関数のアドレスは、既存のOSで管理する共通割り込みエントリ関数105のアドレスだけでなく、アプリケーションで準備される拡張割り込みエントリ関数108のアドレスも存在しているので、アプリケーション独自にリアルタイム性のよい割り込み処理が可能となる。図2の例では、割り込みレベルがレベルmよりも大きい場合には、第2タイプの割り込みとして拡張割り込みエントリ関数108が起動される。

30

#### 【0035】

以上のようにして、割り込み制御装置は、第2タイプの割り込みに対して、任意の割り込みレベルにおいてOS内部の管理処理を介さずに、ユーザー関数独自の割り込み処理を優れた応答性で起動することができる。

なお、図4に示した割り込み制御の代わりに図5に示す割り込み制御処理を実行してもよい。図5は、図4と比べて、ステップS405の代わりにステップS504～S508を実行する点が異なっている。図4と同じステップ番号は同じなので説明を省略し、異なる点を中心に説明する。

40

#### 【0036】

S504～S508の処理は、第2タイプの割り込みにより呼び出される拡張割り込みエントリ関数による割り込み処理、または拡張割り込みエントリ関数により呼び出される割り込み処理を示している。

まず、拡張割り込みエントリ関数の起動により、まず他の割り込み処理の入力を禁止する(S504)。この禁止は、割り込みコントローラ102のマスケジスタの設定を変更することにより、マスカブルな割り込みの全部または一部でよい。次に、現在実行中の

50



プロセスのコンテキスト（CPUの各種制御レジスタのデータなど）を退避し（S505）、割り込みを発生した資源に対応するユーザの割り込みハンドラ109a等を選択して実行し（S506）、ユーザの割り込みハンドラの終了後に退避したコンテキストを復帰し（S507）、割り込み入力禁止を解除する（S508）。

#### 【0037】

この割り込み制御処理によれば、他の割り込み入力を禁止するので、より確実に割り込み処理を実行することが可能になる。その結果、リアルタイム性をより確実に保証することができる。なお、割り込み禁止およびその解除は、割り込み処理において必要とされるリアルタイム性に行うようにしてもよい。例えば、S504にて禁止した後、リアルタイム処理が完了した時点で解除してもよい。

10

#### 【0038】

このように、アプリケーションが管理する第2タイプの割り込みとOSで管理する第1タイプの割り込みの排他制御が可能である。図2の例では、第1タイプの割り込みと第2タイプの割り込みとの間の多重化の調停自体は、第1タイプと第2タイプとが割り込みレベルにより分割されているのでOSを介さずにCPUが実現している。このため第1タイプの割り込みと第2タイプ割り込みを調停する機構を別途設ける必要がなく、第2タイプの割り込み処理に移行するのに処理遅延も発生しない。

#### 【0039】

また、図2では、発生した割り込みがレベルmより上であるかそれ以下であるかによって、第1タイプか第2タイプかを区別する例を示したが、割り込みレベル個別に第1タイプか第2タイプかを設定する構成としてもよい。この場合に、割り込みレベル毎に第1タイプと第2タイプの何れかに任意に対応付ける管理テーブルの一例を図6に示す。

20

また、図6および図2において、図3に示した登録手順なしに、予めそれぞれの割り込みエントリ関数を登録するようにしてもよい。

#### 【0040】

##### （実施の形態2）

図7は、本発明の実施の形態2における割り込み制御装置のシステム構成を示すブロック図である。同図は、図1のシステム構成と比較して、アプリケーションが自プロセスのIDをOSに対して通知するプロセスID通知部114と、OS内に前記アプリケーションから通知されたプロセスIDに基づいてCPUのTLBを更新するTLB更新部117を備える点が主として異なっている。また、同図では、CPU内に仮想記憶用の仮想アドレスと実アドレスとの対応表であるページテーブル115と、ページテーブル115から実行中のプロセスに対応するページデータを一時的にキャッシュするTLB116を備える点を明記している。

30

#### 【0041】

図7の割り込み制御装置において、アプリケーションが割り込みエントリ関数を登録する処理を、図8に示すフローチャートを用いて説明する。図8のフローチャートでは、図3に示したユーザーで準備する割り込み処理ルーチン（拡張割り込みエントリ関数）をCPU内の割り込みベクタレジスタへ登録する手順と、プロセスIDの通知手順とを示している。図3と同じ番号のステップは同じ処理内容なので説明を省略し、異なるステップを中心に説明する。

40

#### 【0042】

アプリケーションは、通知した資源番号に対応するベクタレジスタがベクタテーブル中にある場合（S304）、自プロセスのプロセスIDをAPI（Application Program Interface）を通じてOSから取得し（S801）、取得したプロセスIDをOSに通知する（S802）。この通知を受けたOSはTLB更新部117内に通知されたプロセスIDと割り込みレベルとを記憶する。

#### 【0043】

これにより、アプリケーションからOSへのプロセスIDの通知と、OS内のTLB更新部117でのプロセスIDの記憶とが完了する。

50

次に図 9 は、図 8 での登録処理後に実際に割り込みが発生した場合の割り込み制御処理を示すフローチャートである。図 9 は、図 5 に示した割り込み制御処理と比較して、ステップ S 5 0 5 と S 5 0 6 の間にステップ S 9 0 1 が追加されている点が異なっている。図 5 と同じ番号のステップは同じ処理内容なので説明を省略し、異なる点を中心に説明する。

#### 【 0 0 4 4 】

拡張割り込みエントリ関数の実行において、コンテキストを退避 ( S 5 0 5 ) した後、T L B 更新部 1 1 7 に T L B の更新を指示し、この指示を受けた T L B 更新部 1 1 7 は通知されたプロセス I D のページデータをページテーブルから読み出して、T L B 1 1 6 に格納する ( S 9 0 1 )。このように T L B 1 1 6 をプロセス I D に基づいて更新することにより、ユーザープロセス空間アドレスが参照可能となる。

10

#### 【 0 0 4 5 】

以上により、アプリケーションが管理する第 2 タイプの割り込みに対応する割り込み処理関数をカーネル空間とは別空間となるアプリケーションのユーザープロセス空間に実装することが可能となる。

また、割り込み発生後、拡張割り込みエントリ処理 (カーネル空間) からアプリケーションが管理する割り込み処理関数 (ユーザープロセス空間) へ遷移するが、カーネル内のスケジューラ 1 0 6 を通らないため、割り込み応答性を阻害することがない。

#### 【 0 0 4 6 】

なお本発明は、上述した各実施例に限定されるものではなく、その要旨を逸脱しない範囲で、種々変形して実施することができる。

20

#### 【産業上の利用可能性】

#### 【 0 0 4 7 】

本発明は、コンピュータシステムを制御する O S の一部として、割り込みを発生する割り込み資源からの割り込みを制御する割り込み制御装置に適し、特に、コンピュータシステム上の割り込み資源を、O S とアプリケーションとが別々にそれぞれの割り込みを管理する割り込み制御装置、割り込み制御方法、およびプログラムに適している。

#### 【図面の簡単な説明】

#### 【 0 0 4 8 】

【図 1】実施の形態 1 における割り込み制御装置の構成を示すブロック図である。

30

【図 2】レジスタ構成に着目して処理の流れを示す動作説明図である。

【図 3】拡張割り込みエントリ関数を登録する処理を示すフローチャートである。

【図 4】割り込み制御処理を示すフローチャートである。

【図 5】他の割り込み制御処理を示すフローチャートである。

【図 6】割り込みレベル毎に第 1 タイプと第 2 タイプの何れかを対応付ける管理テーブル例を示す図である。

【図 7】実施の形態 2 における割り込み制御装置の構成を示すブロック図である。

【図 8】拡張割り込みエントリ関数を登録する処理を示すフローチャートである。

【図 9】割り込み制御処理を示すフローチャートである。

【図 10】従来の O S のカーネル空間からユーザープロセス空間への割り込み制御の遷移関係を示す図である。

40

【図 11】従来のコンピュータシステムにおける割り込み制御装置の構成を示す図である。

【図 12】図 11 における処理の流れを各構成要素のレジスタ構成に着目して示す動作説明図である。

#### 【符号の説明】

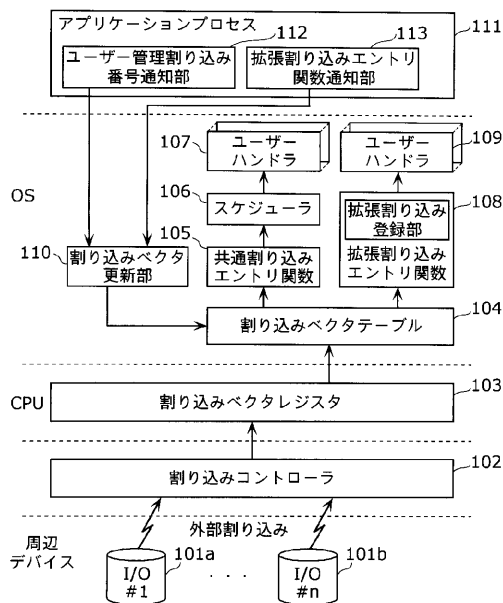
#### 【 0 0 4 9 】

- 1 0 2      割り込みコントローラ
- 1 0 2 a    割り込み要因番号レジスタ
- 1 0 3      C P U

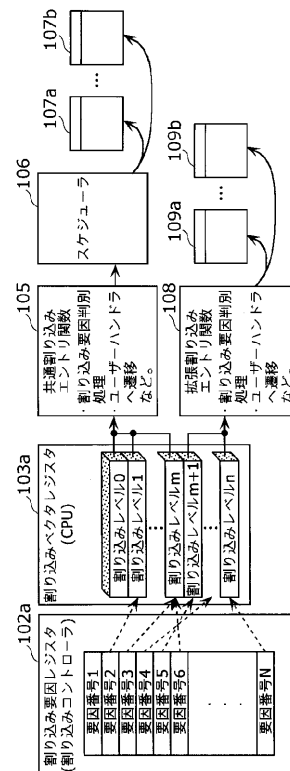
50

- 1 0 3 a 割り込みベクタレジスタ
- 1 0 5 共通割り込みエントリ関数
- 1 0 6 スケジューラ
- 1 0 7 割り込みハンドラ
- 1 0 8 拡張割り込みエントリ関数
- 1 0 9 ユーザ割り込みハンドラ
- 1 1 0 割り込みベクタ更新部
- 1 1 2 ユーザ管理割り込み番号通知部
- 1 1 3 拡張割り込みエントリ関数通知部
- 1 1 4 プロセスID通知部
- 1 1 5 ページテーブル
- 1 1 6 T L B
- 1 1 7 T L B 更新部

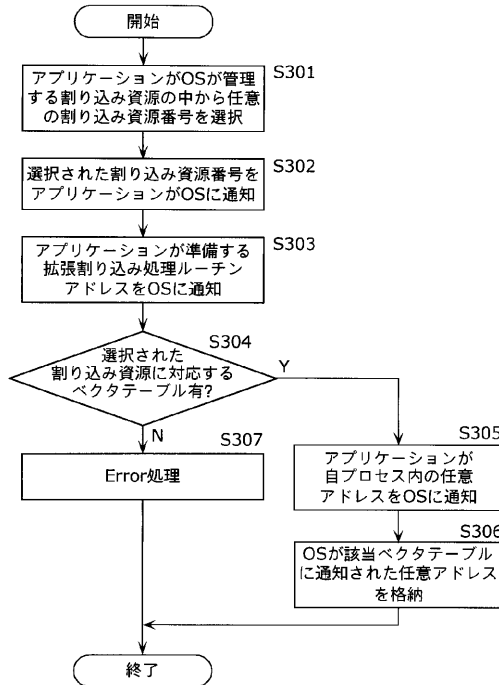
【図 1】



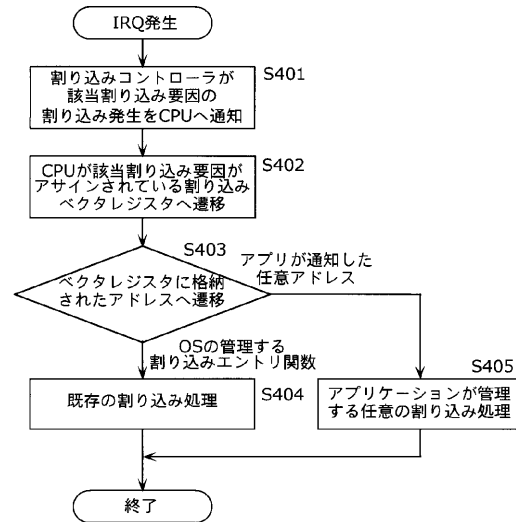
【図 2】



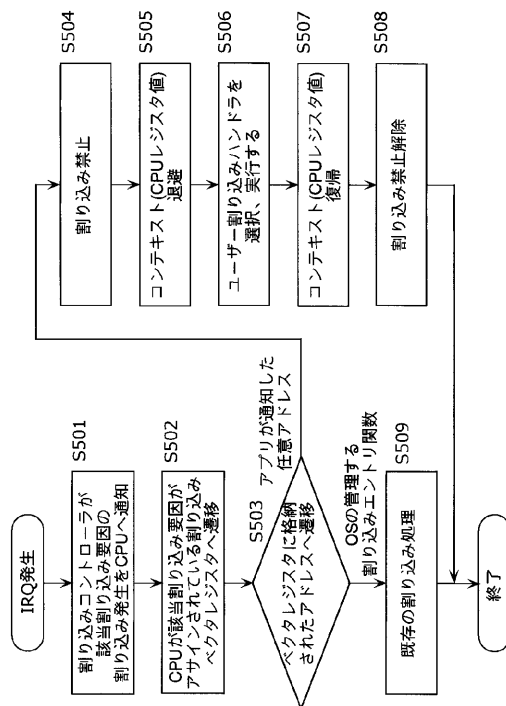
【図 3】



【図 4】



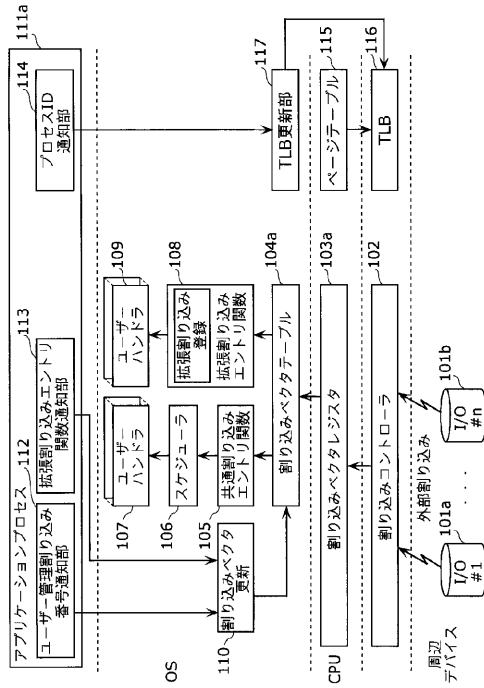
【図 5】



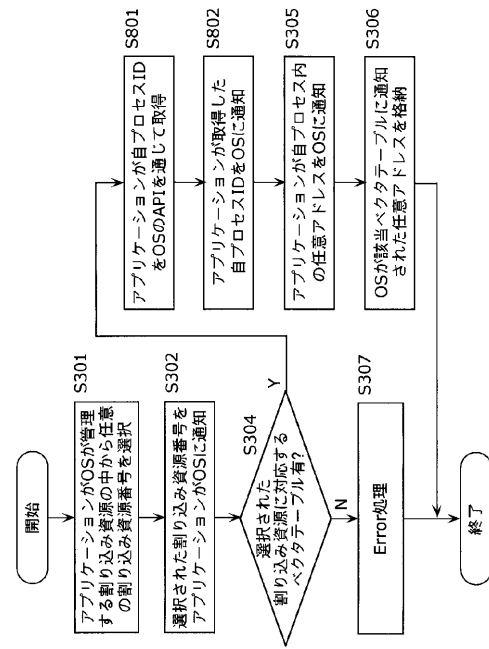
【図 6】

割り込みレベル	割り込みエントリ関数
0	共通割り込みエントリ
...	共通割り込みエントリ
i	共通割り込みエントリ
i+1	拡張割り込みエントリ
...	拡張割り込みエントリ
j	拡張割り込みエントリ
j+1	共通割り込みエントリ
...	共通割り込みエントリ
k	共通割り込みエントリ

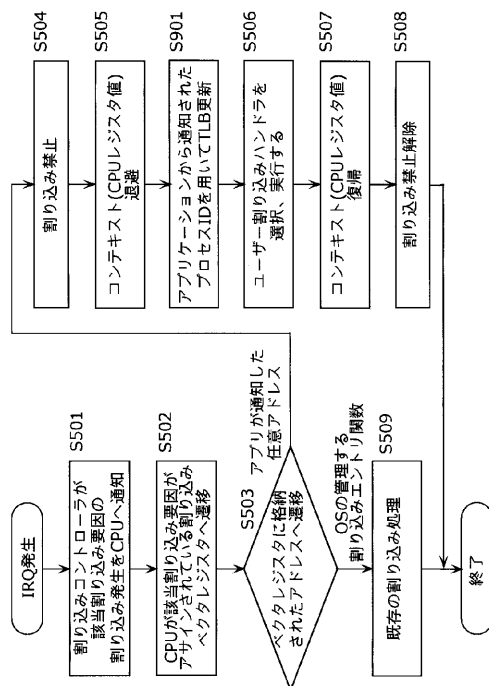
【図 7】



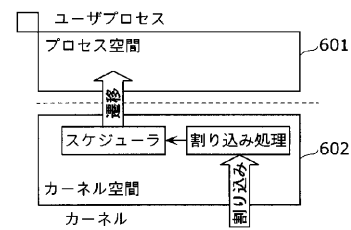
【図 8】



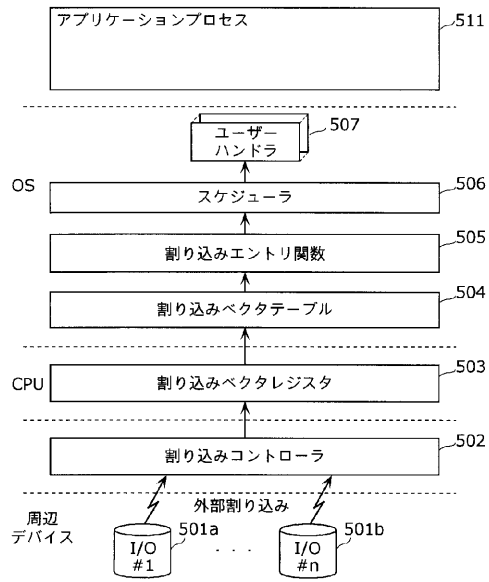
【図 9】



【図 10】



【図 1 1】



【図 1 2】

