オペレーティングシステム
i386アーキテクチャ(2)

2005年10月27日

海谷 治彦

目次

- ページ
- 保護機能
  - セグメントの保護
  - ページの保護
- 割り込みと例外

ページング

- 0～4GBで表現されるリニアアドレスを物理アドレス(実際にマシンに搭載されたアドレス)に変換する仕組み.
- 実際のところ4GBなんてメモリをつんだマシンはめったに存在しないので必須.
- もし4GBメモリのマシンがあったとしても, i386上で, 複数のプロセスが同時に動作するOSを稼動させるには必要な技術.

ページ

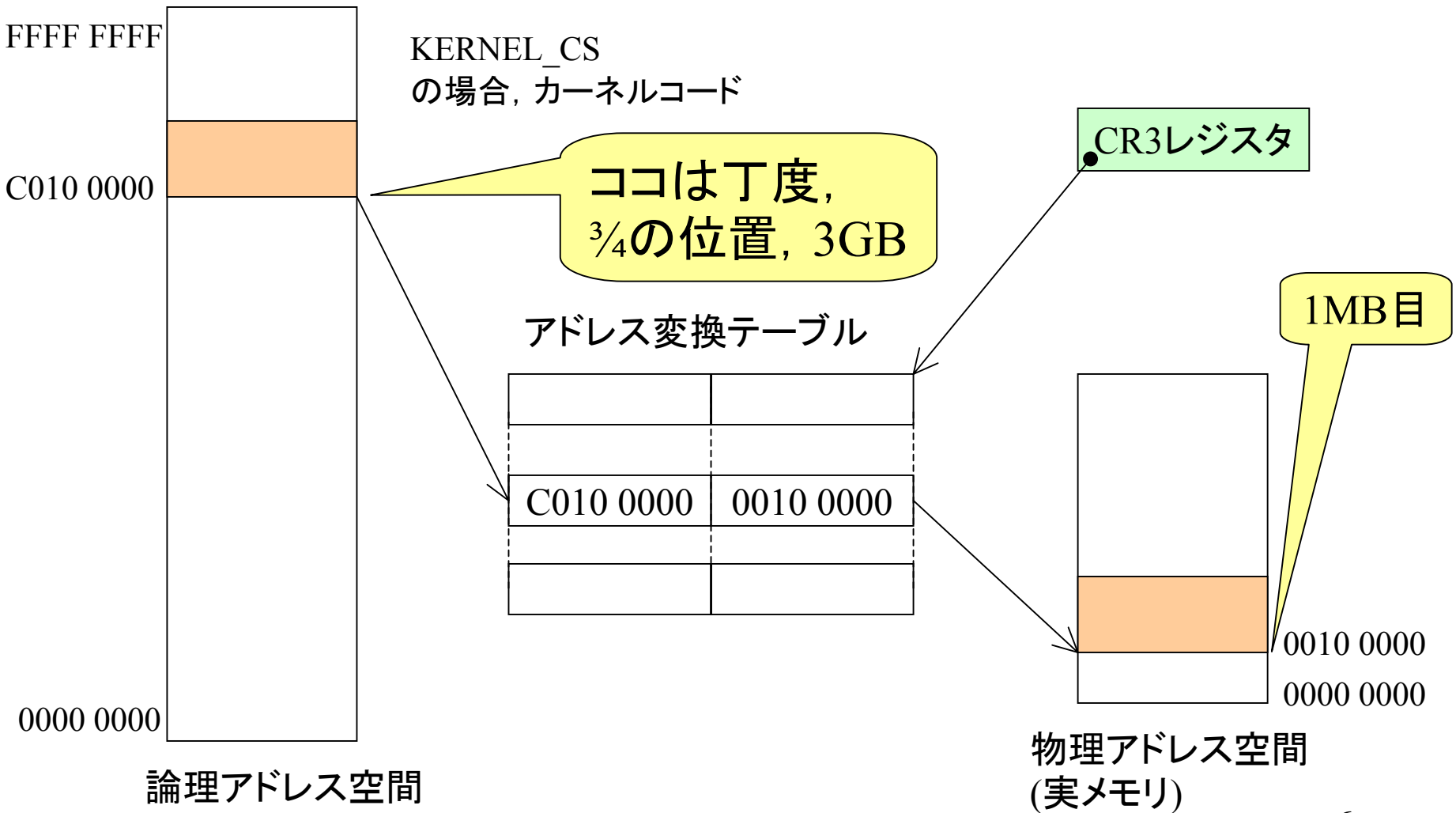
- セグメントを4KB(もしくは4MB)の固定長に分割した個々の部分のこと.
- リニアアドレスのある部分が物理アドレスのどの部分に対応するかは表で管理している.

  ⇒ アドレス変換テーブル

アドレス変換テーブル

- 前述のようにリニアアドレスのある部分が物理アドレスのどの部分に対応するかを記述した表.
- 無論，メモリ内に記入される.
- ある瞬間に利用されているテーブルは1つだが，変更もできる.
- i386ではCR3レジスタにこのテーブルがあるアドレスが記入されている.

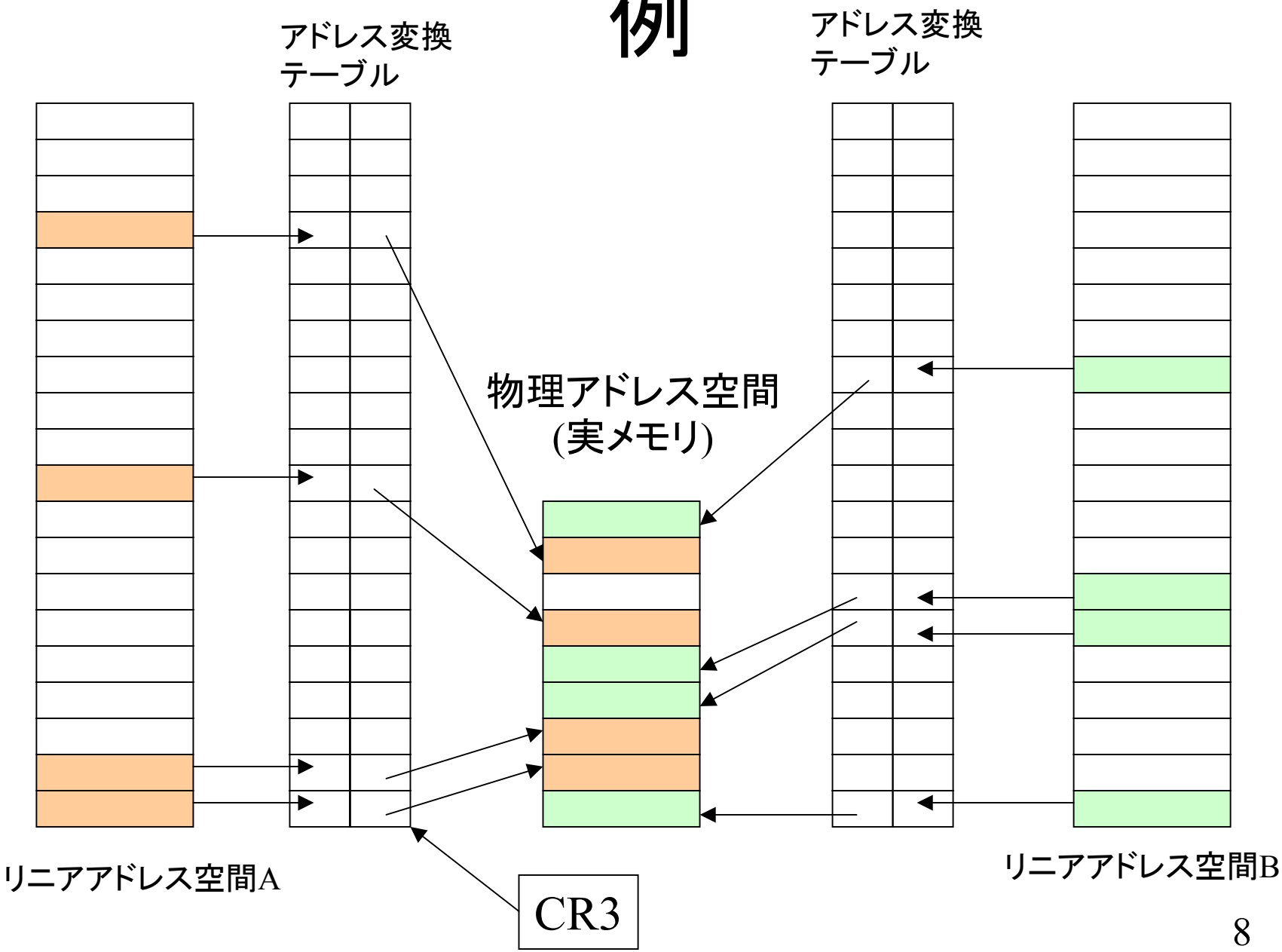
アドレス変換テーブルの例



アドレス変換テーブルとプロセス

- Linuxではプロセス毎にアドレス変換テーブルを作る.
- さらに，プロセス間で同じ物理アドレスを使わないようにテーブル内の値を設定する.
  - 意図的に共有させることもできる.
- ユーザープロセスがこのテーブル群やCR3の内容は変更できないので安心.
  - テーブル群はKERNEL_DSに書かれる.
  - CR3はKERNEL_CS内のコードでしか変更できない.
- 実際のテーブルは効率化のため，二段階テーブルになっているが，それについては後日に.

例


アドレス変換
テーブル
アドレス変換
テーブル
物理アドレス空間
(実メモリ)
リニアアドレス空間A
リニアアドレス空間B
CR3

i386の3つのアドレス記述法

- 論理アドレス: マシン語でのアドレス指定に使う形式．セグメントとオフセットの対で表現.
- リニアアドレス: 4GBのメモリ空間を素直に表現した形式．32ビット
- 物理アドレス: メモリチップの実アドレス．表現は32ビットだが，上限は実際に搭載しているメモリ量に依存.

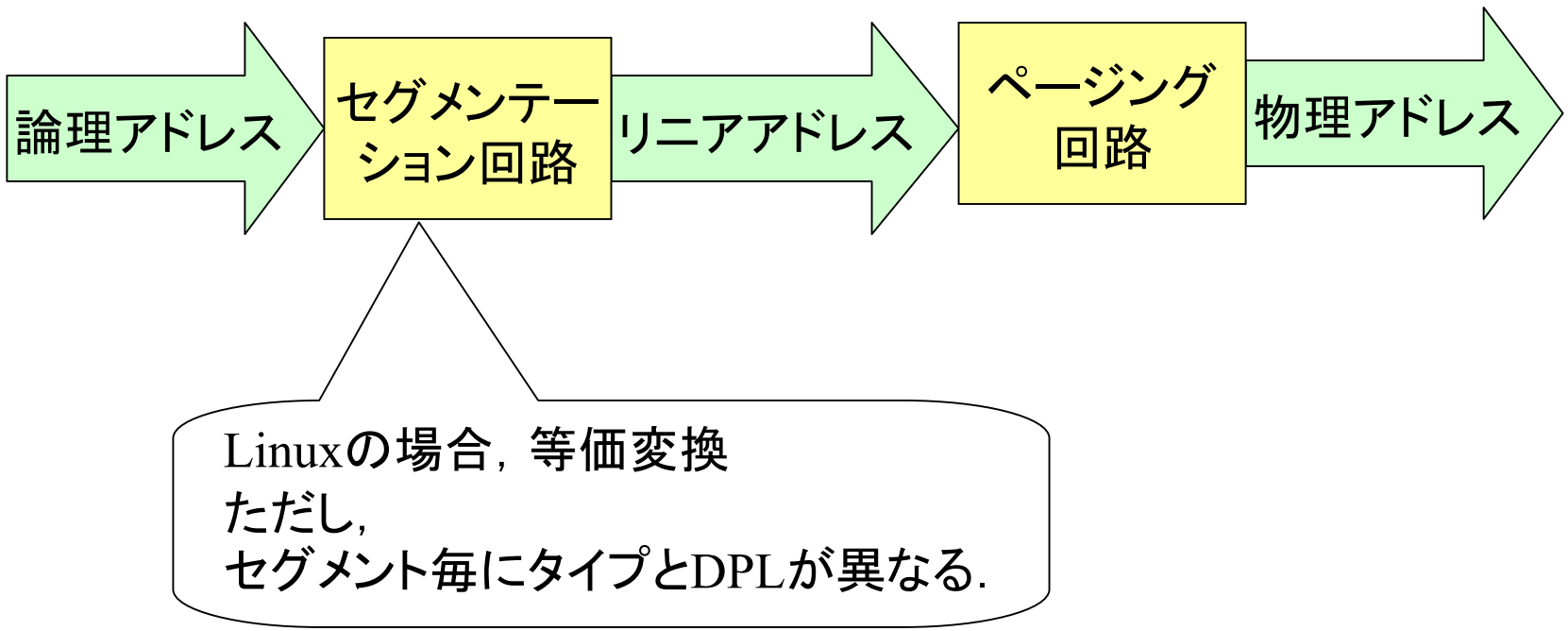
Linuxの場合，4Gサイズのセグメントを使うので，

論理アドレス = リニアアドレス

文献5 p.44

アドレス変換

マシン語が解釈されて，実際のメモリ回路までたどり着くには，以下のような二段階の変換が行われる.



保護機能

- 特権レベルとはセグメント内に書かれたプログラムの実行権限の強さを規定するもの.
- i386は4種類の特権レベルを割り当てられるが, Linuxでは以下の2つのみを使う.
  - レベル0 なんでもできる.
  - レベル3 特権命令が実行できない.
- 保護はセグメント段階とページ段階の二段階で行われる.

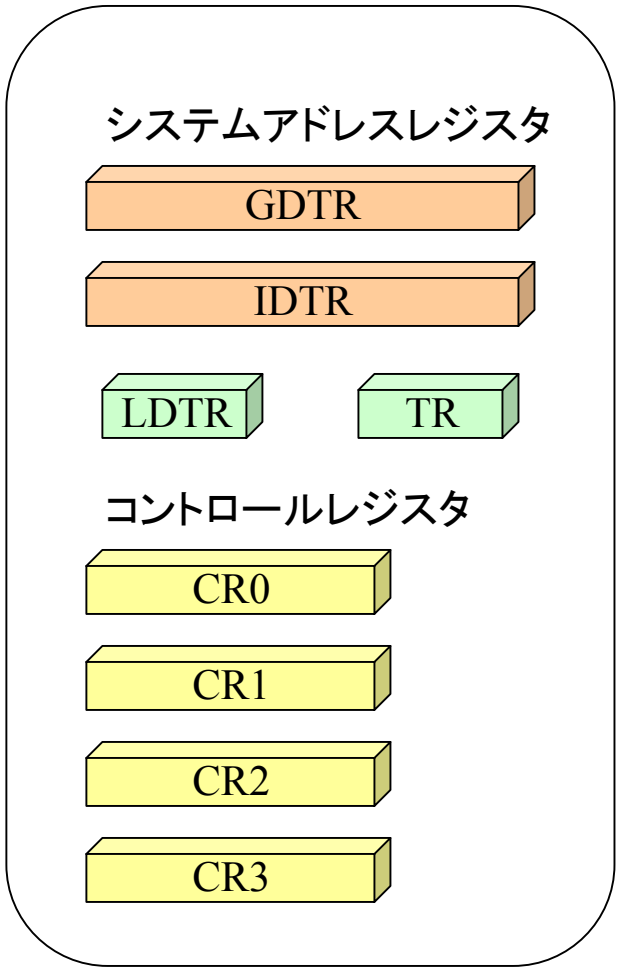
特権命令の例

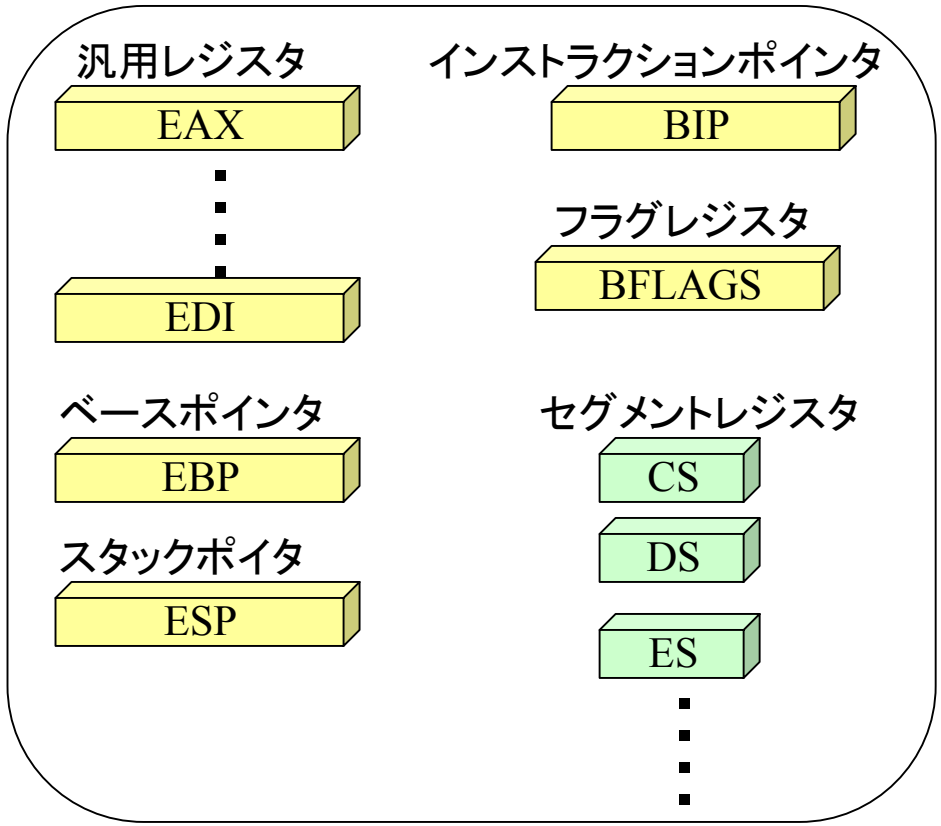
- HLT
  - システムを停止する.
- LGDT
  - GDTの値を設定
  - 要は利用しているセグメントの切り替え
- MOV CR_n
  - CR_nレジスタの値を設定
  - ページ変換テーブルの指定はCR_3

等

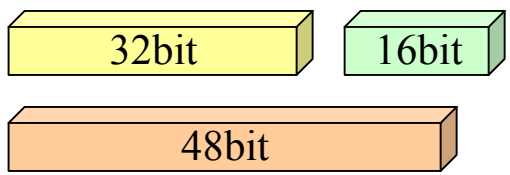
再録

i386のOS関係のレジスタ





凡例



割り込みと例外

- 現在，動作しているプログラムを停止し，事前に準備した別のプログラムを実行する仕組み.
- 割り込み・例外は発生原因によって分類されている.
- 大まかにいうと，
  - 割り込み: 優先度の高い機器等からの優先処理依頼.
  - 例外: 異常事態.

  と分類できる.

i386の割り込みの発生源

- ハードウェア割り込み
  - キーボードやタイマー等の外部機器からの信号により発生. (INTRピンから受け取る)
  - メモリーエラー等により発生. (NMIピンから受け取る)
    - この割り込みのみマスク(受け取り拒否)できない.
- ソフトウェア割り込み
  - 割り込みを発生させるアセンブラ命令, INT を用いて, プログラム自身が発生させる.
    - 気分的にはサブルーチンコールに似ている.

i386例外の分類

対処法の違いにより分類

- フォルト: 割り込まれた命令を再実行する場合.
  - セグメントやページの保護違反等が原因.
- トラップ: 再実行しないもの.
  - サブルーチンコールのようなもの.
- アボート: 致命的なエラー.
  - プログラムもしくはOSの強制停止がありうる.

i386の割り込みと例外の識別

- 割り込みと例外は「割り込みベクタ」という番号によって識別される.
- 番号は 0 ～ 255 (8bit)の256個つけられる.
- 0 ～ 31 は intelが予約, 32以降は自由に使ってよい.
- 通常, それぞれの割り込み番号は,

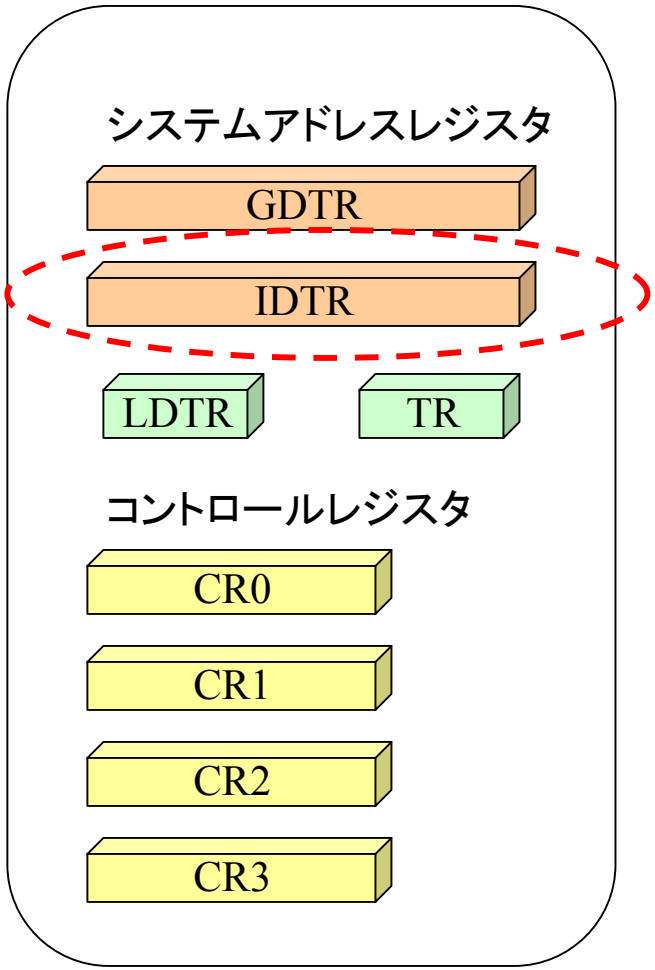
  INT X

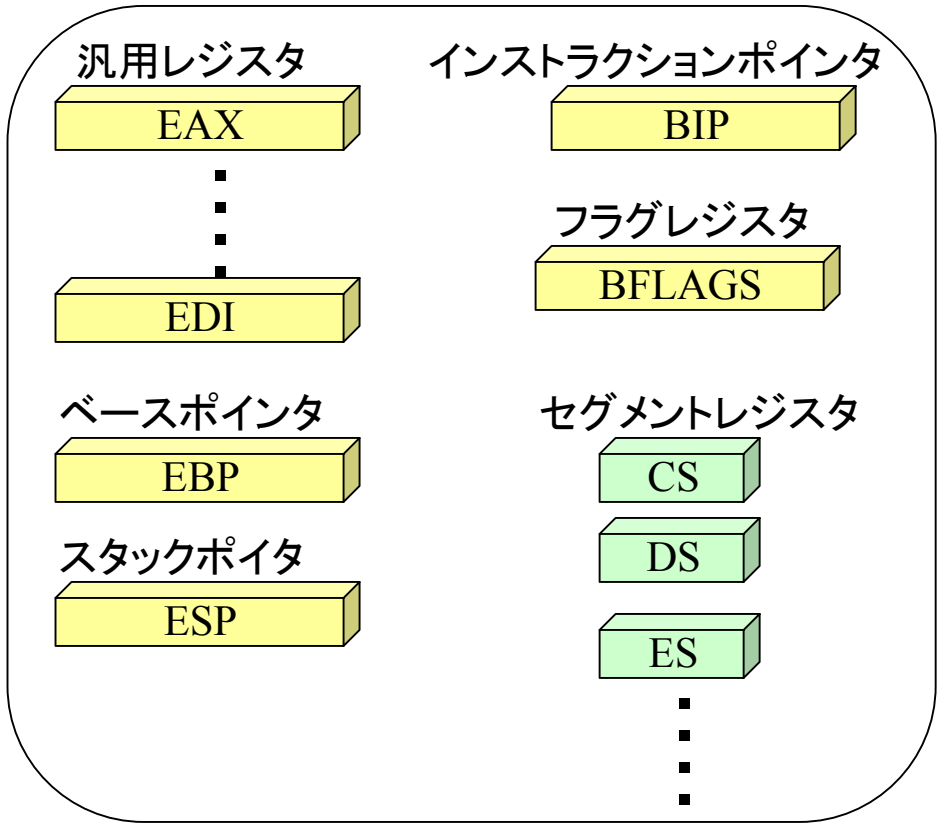
  とかいう番号で略記される. (実際, 割り込みを呼び出す場合のアセンブラ命令はこんな感じで書く)
- ベクタ番号128 (INT 0x80)は, Linuxではシステムコール実装のために使われる.

割込みディスクリプタ・テーブル

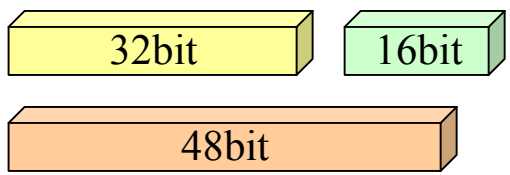
- IDT (Interrupt Description Table)
- 1個8Bのエントリを数個(≦256)列挙して，ある番号の割り込み・例外が起こったら，どのセグメントのどのコードを実行するかを定義している表.
- 表の先頭アドレスとはIDTRレジスタに記録されている.
- 後述の割込みゲートディスクリプタとトラップゲートディスクリプタの二種類のエントリがある.
- ベクトル番号(エントリの番号) 3, 4, 5, 128 だけはユーザーモードから割り込めるが，他はカーネルモードでないと利用できない.
  - 128はシステムコール実装のための利用.

i386のOS関係のレジスタ



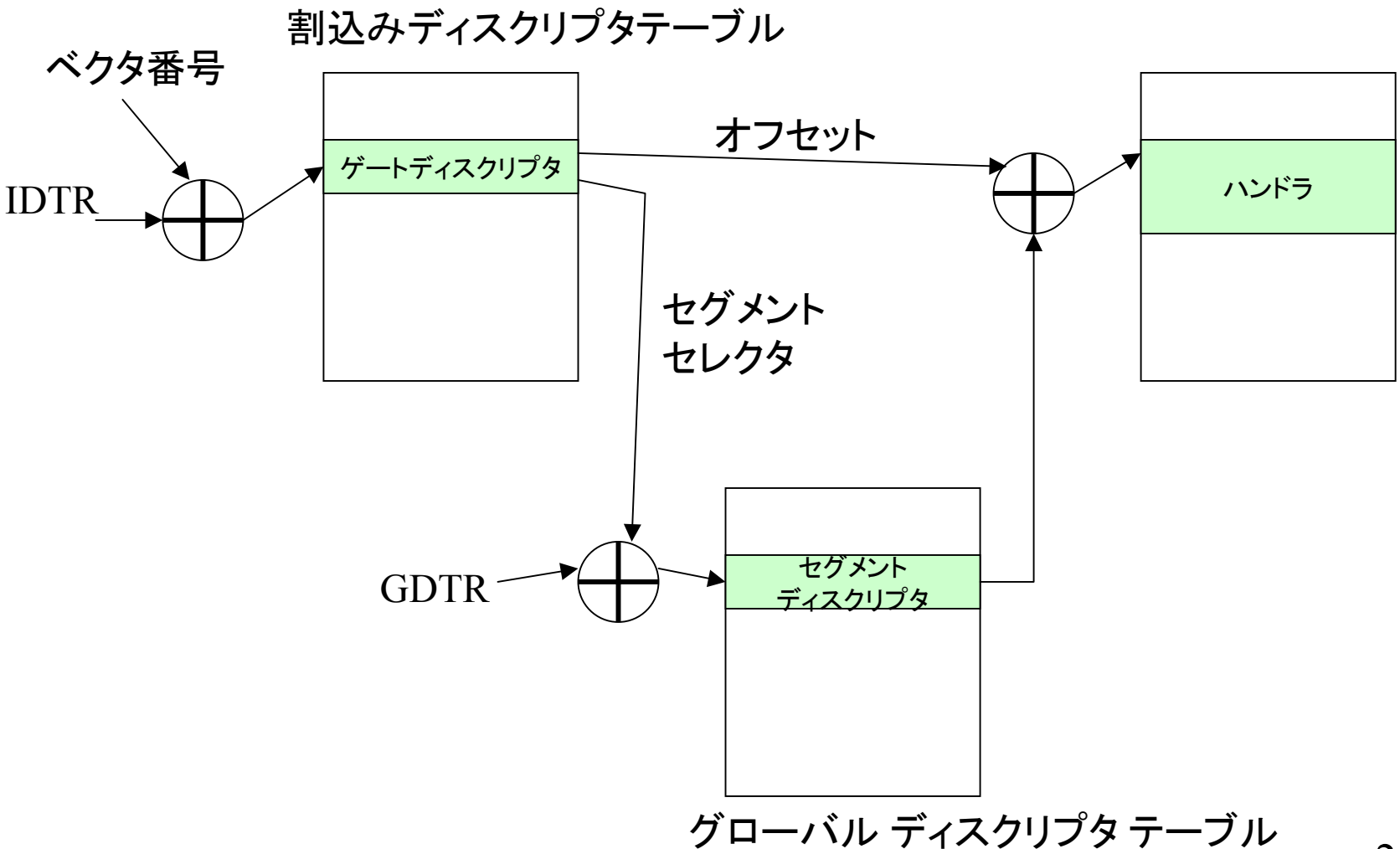


凡例



エントリの種類

- 割込みゲート ディスクリプタ
  - この割込み実行中には他の割込みを禁止する.
  - Linuxでは割込みを全てこちらのエントリ方式で処理する.
- トラップ ゲート ディスクリプタ
  - 他の割込みを禁止しない.
  - Linuxでは例外を全てこちらのエントリ方式で処理する.
  - ユーザーモードからアクセス可能な 3, 4, 5, 128番はこちらのエントリである.

もう1種類あるけどLinuxで使わないので略.

割込みハンドラ検索の概念図