ていれべるなほん

haseham

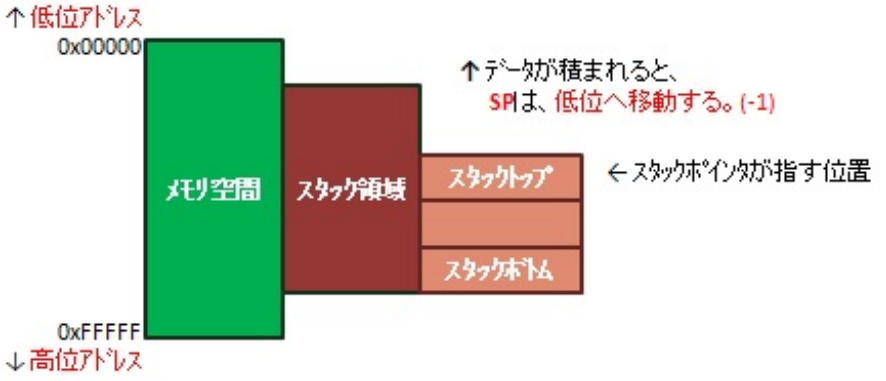
はじめに

　本書では、ﾒﾓﾘｱﾄﾞﾚｽに関する詳しい説明など、
OSやﾌﾟﾛｾｯｻの内部動作に関する部分を解説します。

　本書の内容は、C/C++や、Javaでﾌﾟﾛｸﾞﾗﾐﾝｸﾞする上では、
必須でもないのですが、ｱｾﾝﾌﾞﾗなどを併用するなど、
ﾊｲﾊﾟﾌｫｰﾏﾝｽが要求される場面では、必要となる情報です。

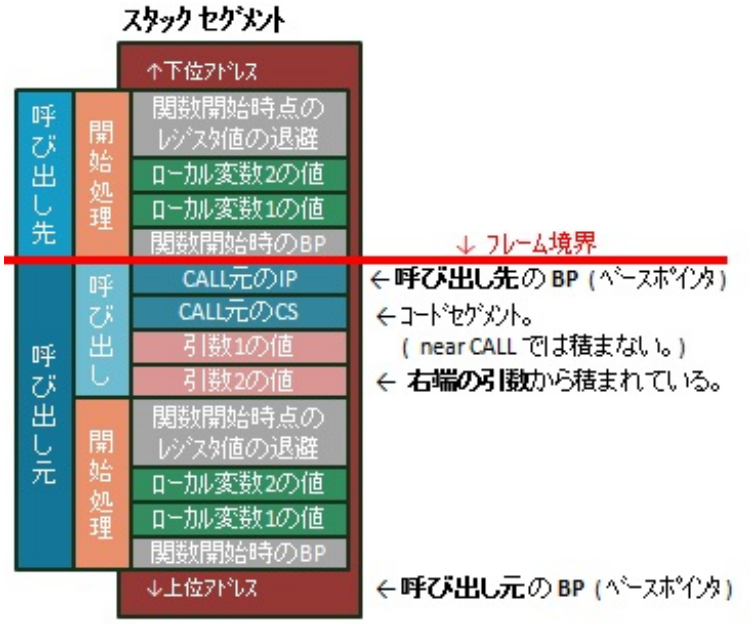
　情報工学科の学生や、卒業生であれば、
当然にして習う内容ではあると思われるのですが、
独習者にとっては、ﾈｯﾄ上のTipsｻｲﾄだけでは、情報が不足しており、
これを読めば、という本が、なかなか見当たらなかったため、
既存の良書や、Tipsｻｲﾄ、Intelのﾏﾆｭｱﾙの内容をまとめ、
初学者にも理解しやすくするために、本書を書き始めることとしました。

　とはいえ、まだ書きはじめたところであり、
抜けている点の方が多いのですが、
独学の初学者が、既存の技術資料を読み進める上でも、
何らかの助けにはなるのではないかと思い、
公開しながら、執筆する形式をとることとしました。



・低位ｱﾄﾞﾚｽが上向きに書かれていますが、↑
　これはｽﾀｯｸが、その名の通り、「積み上げる」ものだからです。

・本書では、この図から書きはじめたこともあって、
　他の図も、逆向きに書いています。



・SP（ｽﾀｯｸ ﾎﾟｲﾝﾀ）は、**最後に積まれたﾃﾞｰﾀ**を指しており、
関数が開始された時点では、BP（ﾍﾞｰｽﾎﾟｲﾝﾀ）と**同じ値**である。

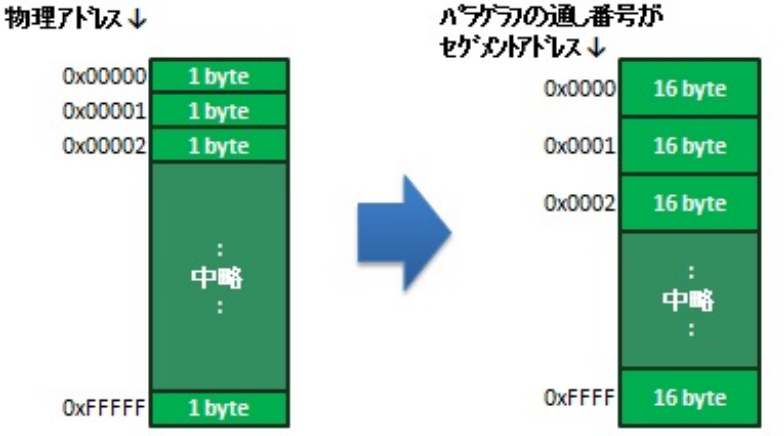
・**退避するﾚｼﾞｽﾀ値**は、BPと、**関数内で使用するもの**だけでよい。

・**関数の開始処理で確保した領域**は、**関数の最後**に、必ず**解放**する。
・**呼び出し時に引数を積んだ領域**も、**復帰直後**に、必ず**解放**する。

・上図は、C言語のｺﾝﾊﾟｲﾗが出力する一般的な形式です。

・C言語からｱｾﾝﾌﾞﾗのﾌﾟﾛｼｰｼﾞｬを呼び出す場合は、
このC言語の形式で、ﾌﾟﾛｼｰｼﾞｬ・ｽﾀｯｸを書く必要があります。

ﾊﾟﾗｸﾞﾗﾌ



・物理ﾒﾓﾘ空間を、16 byte ずつに分割したものを「パラグラフ」という。

・セグメントは、このパラグラフをまとめたものであり、
パラグラフの通し番号がセグメントアドレスである。

・セグメントアドレスを16倍すると、対応する物理アドレスが求まる。
（16進数でいうと、1ケタくり上げて、1ケタ目を0にすると求まる。）

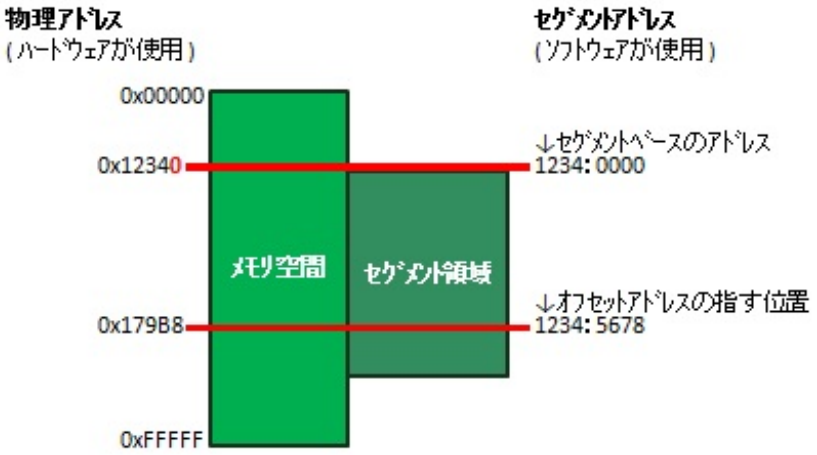
・さらにこれにオフセットアドレスを加算すれば、
オフセットアドレスに対応する物理アドレスも求まる。

ｾｸﾞﾒﾝﾄの種類

CPUによる各種ｾｸﾞﾒﾝﾄへの アクセス

・CPUは、命令に応じて、**対象のｾｸﾞﾒﾝﾄを切り替え**てｱｸｾｽしている。

ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽ空間



・ｾｸﾞﾒﾝﾄの方式は、ﾘｱﾙﾓｰﾄﾞと、ﾌﾟﾛﾃｸﾄﾓｰﾄﾞとで大きく異なる。

・ﾘｱﾙﾓｰﾄﾞでは、ｾｸﾞﾒﾝﾄｱﾄﾞﾚｽのところに、ｾｸﾞﾒﾝﾄﾍﾞｰｽが配置されている。

・ﾌﾟﾛﾃｸﾄﾓｰﾄﾞでは、ｾﾚｸﾀ値が配置されている。

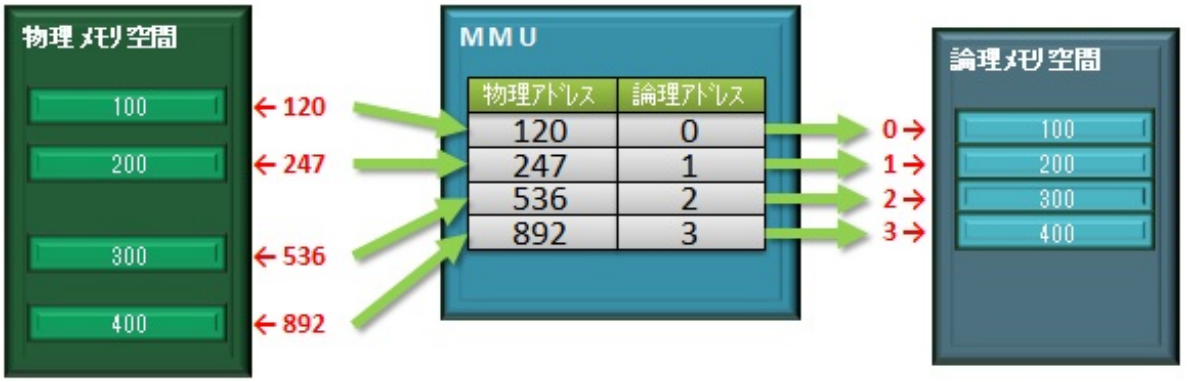
・ｾﾚｸﾀ値は、16bit値で、うち、下位2bitには、特権ﾚﾍﾞﾙが配置されている。

・ｾﾚｸﾀ値は、ﾃﾞｨｽｸﾘﾌﾟﾀ・ﾃｰﾌﾞﾙのindexとなっており、
ｾｸﾞﾒﾝﾄのﾍﾞｰｽｱﾄﾞﾚｽは、ｾｸﾞﾒﾝﾄ・ﾃﾞｨｽｸﾘﾌﾟﾀに配置されている。

(このとき、ｾﾚｸﾀ値の下位2bitは、0で埋める。)

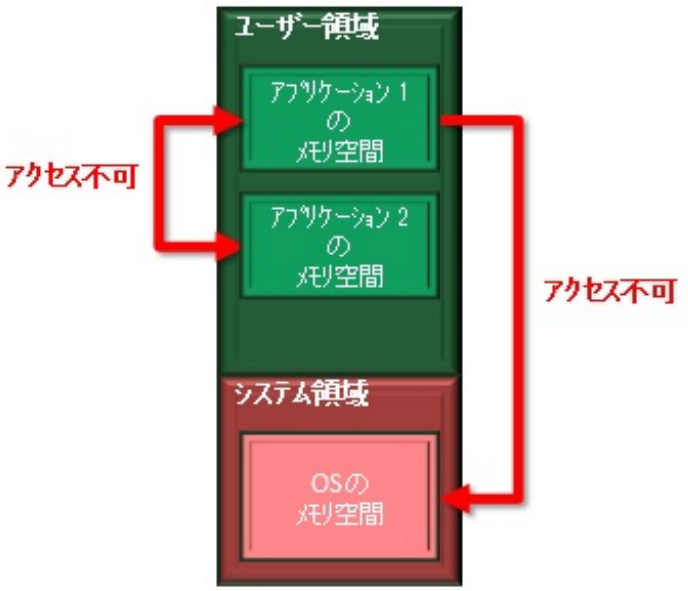
論理ｱﾄﾞﾚｽ空間

断片化したﾒﾓﾘ領域を、連続した論理領域に見立てることができる



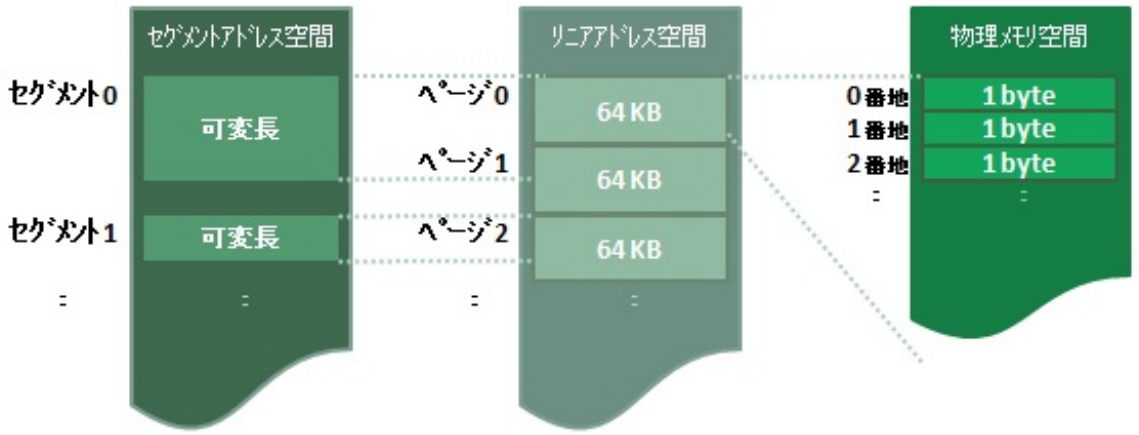
ﾒﾓﾘ保護

ﾀｽｸごとに割り当てられたﾒﾓﾘ空間

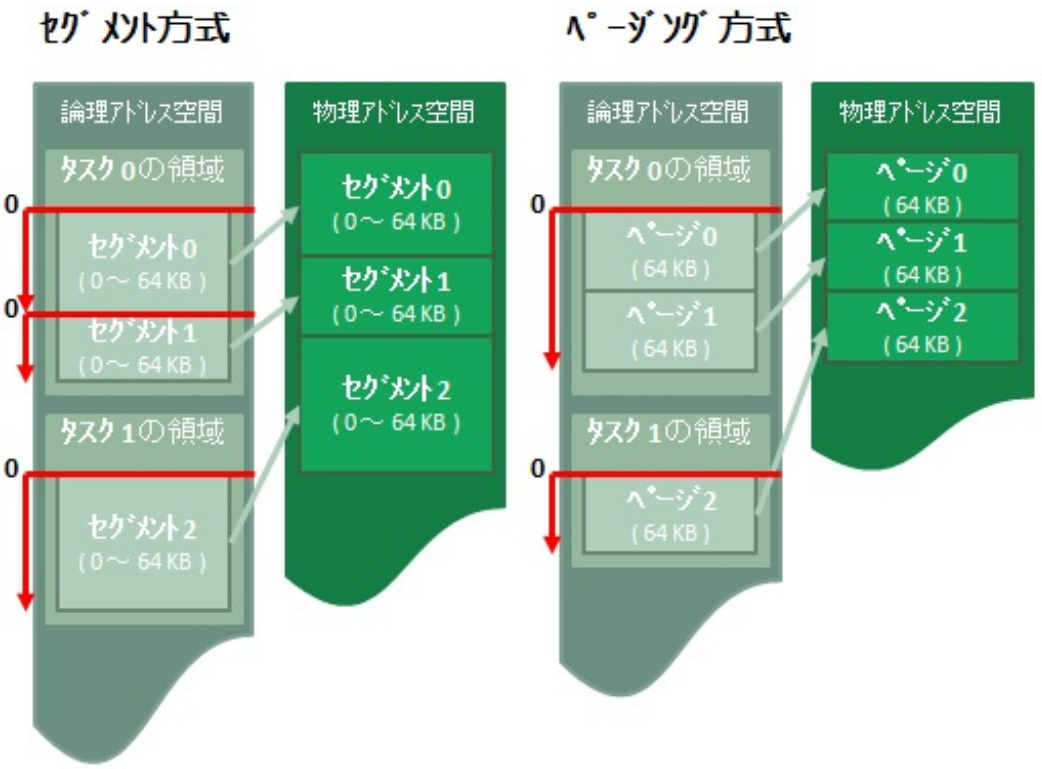


IA-32のﾍﾟｰｼﾞﾝｸﾞ

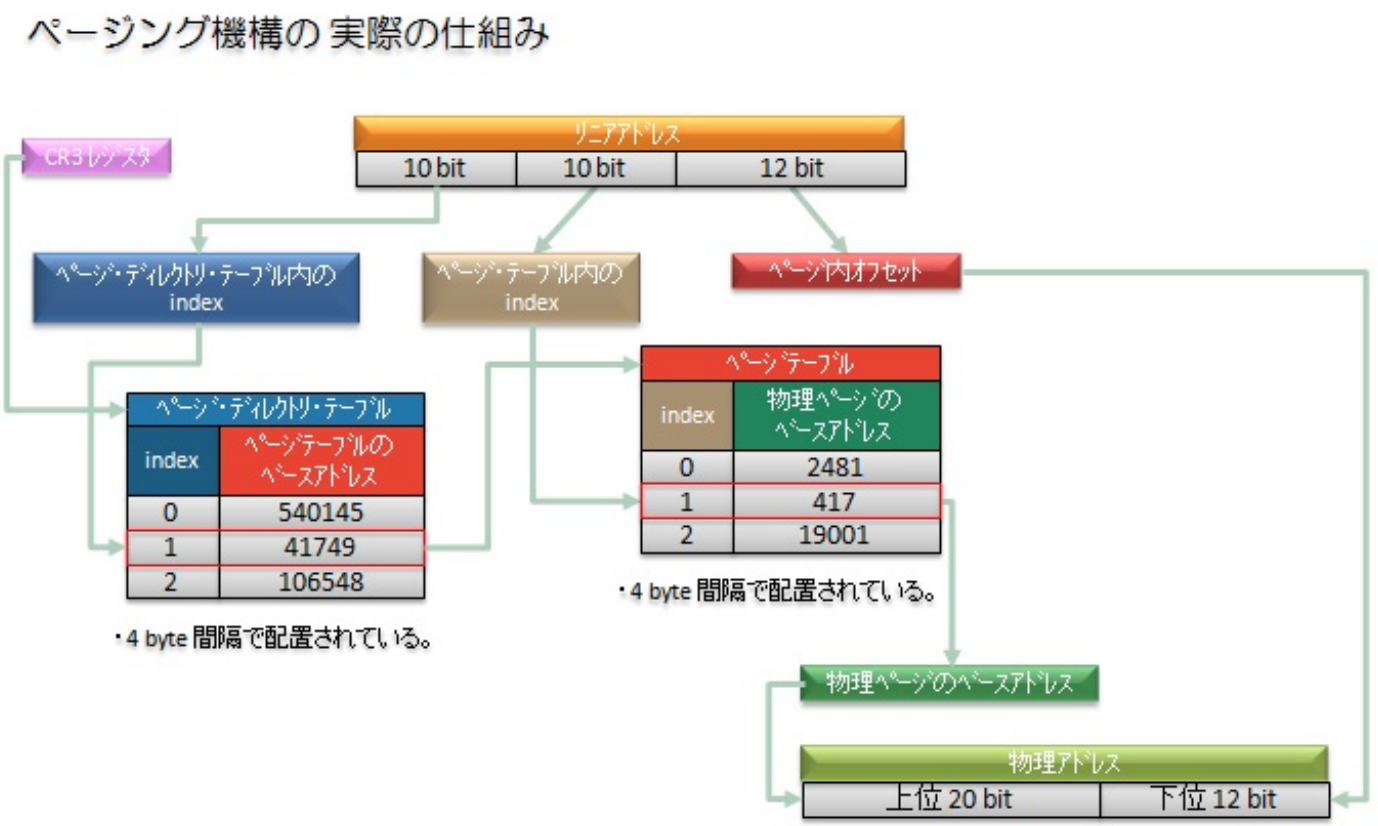
ｾｸﾞﾒﾝﾄは 複数のﾍﾟｰｼﾞをまたいで設置できる。



ｾｸﾞﾒﾝﾄとﾍﾟｰｼﾞﾝｸﾞ



ﾍﾟｰｼﾞﾝｸﾞ機構の仕組み



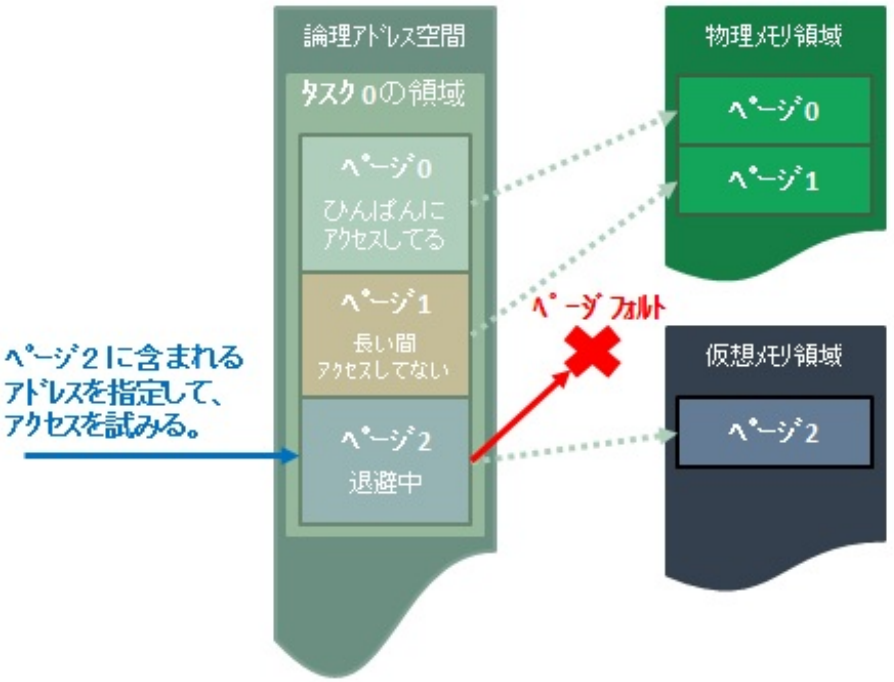
・ｾｸﾞﾒﾝﾄ機構は、ｾﾚｸﾀ値と、ｵﾌｾｯﾄを組み合わせて、
ﾘﾆｱｱﾄﾞﾚｽに変換する。

・ﾘﾆｱｱﾄﾞﾚｽは、ﾍﾟｰｼﾞﾝｸﾞが有効でない場合は、
そのまま物理ｱﾄﾞﾚｽとして利用される。

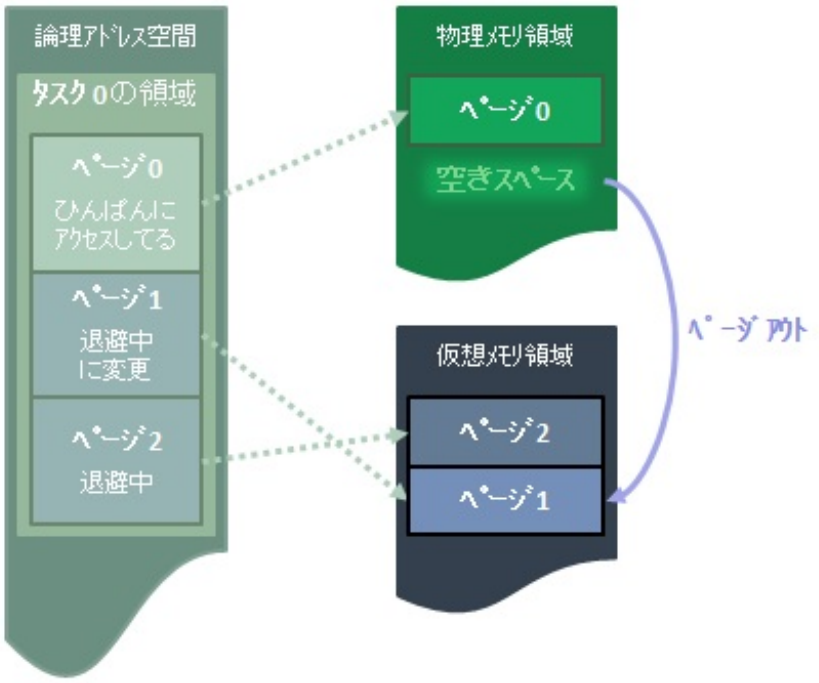
・ﾍﾟｰｼﾞﾃｰﾌﾞﾙを分割して管理しているのは、
ﾍﾟｰｼﾞﾃｰﾌﾞﾙ自身もﾍﾟｰｼﾞとして管理するためである。

・ｱｸｾｽ頻度の少ないﾍﾟｰｼﾞﾃｰﾌﾞﾙは、
ﾊｰﾄﾞﾃﾞｨｽｸ上の仮想領域に置かれる。

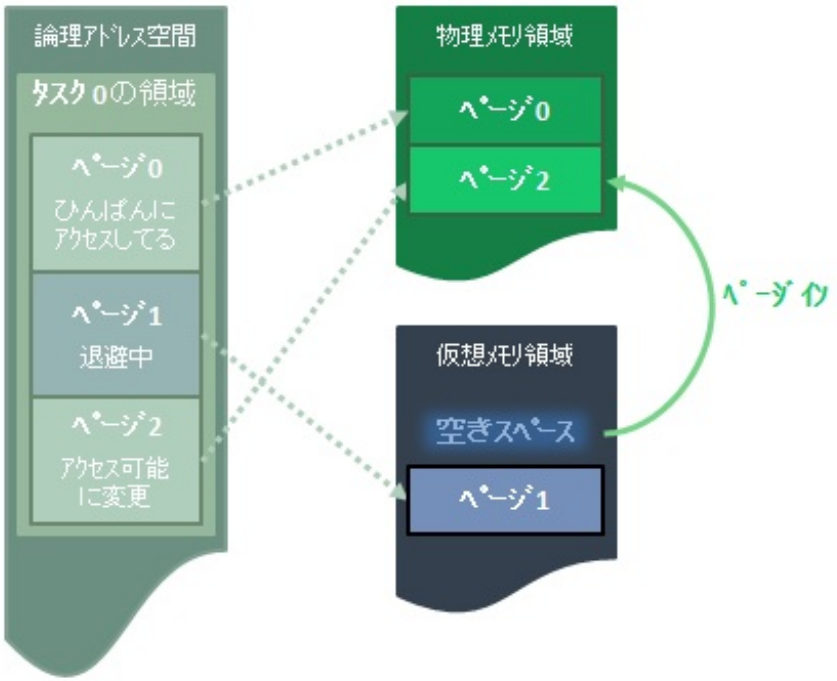
物理ﾒﾓﾘ上にないﾍﾟｰｼﾞ２に、ﾃﾞｰﾀをｾｯﾄしようとすると…



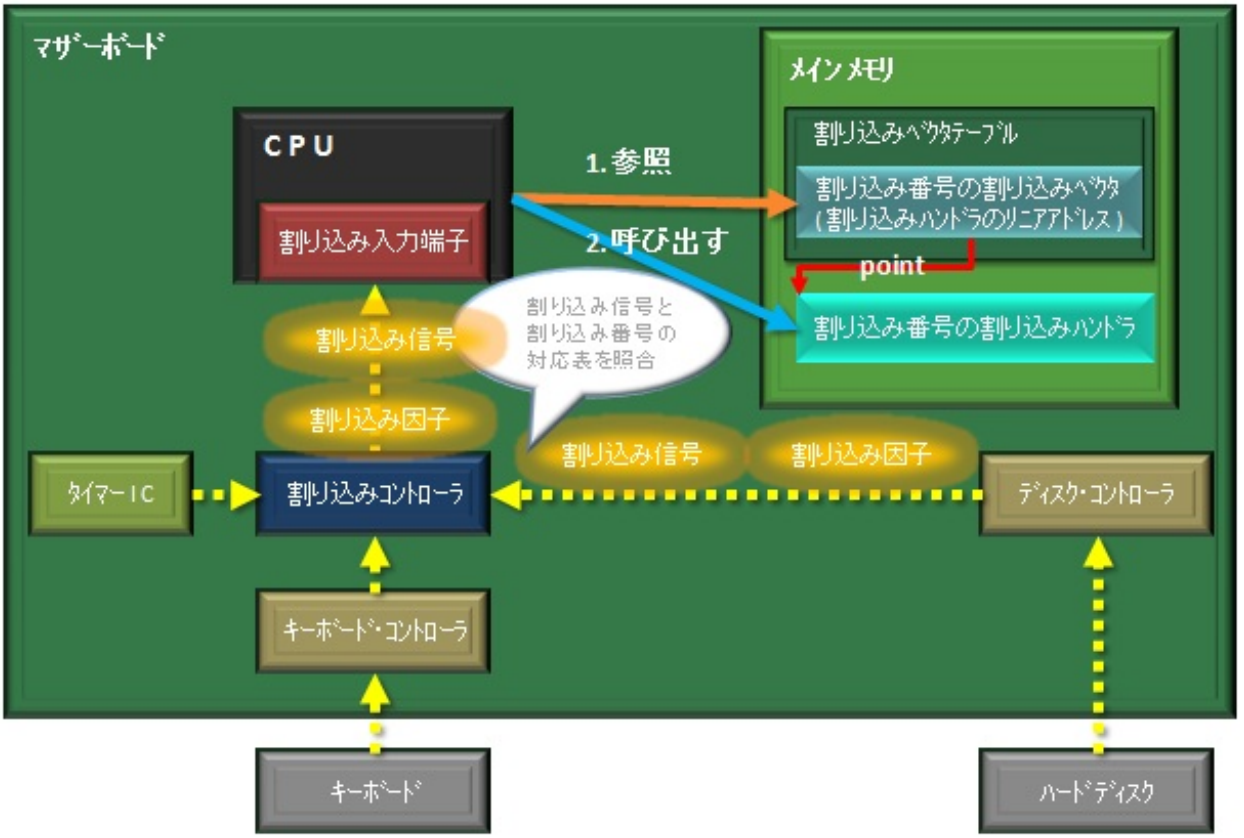
あまりｱｸｾｽしてないﾍﾟｰｼﾞ1を、仮想領域にﾍﾟｰｼﾞｱｳﾄする。



物理ﾒﾓﾘが１つ空いたので、ﾍﾟｰｼﾞ２を、ﾍﾟｰｼﾞｲﾝする。



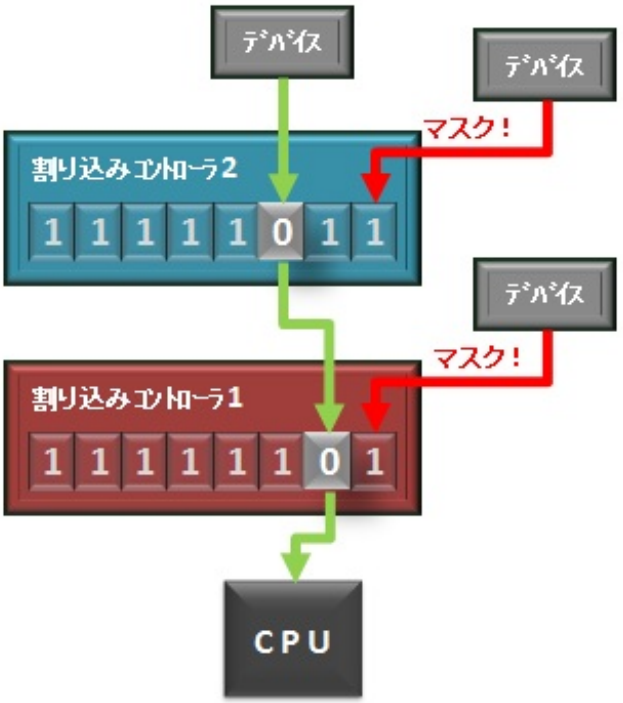
周辺機器からの割り込み信号の流れ



割り込みﾍﾞｸﾀ

・ﾘｱﾙﾓｰﾄﾞでは、こちらを使用する。

IOﾏｽｸﾚｼﾞｽﾀ



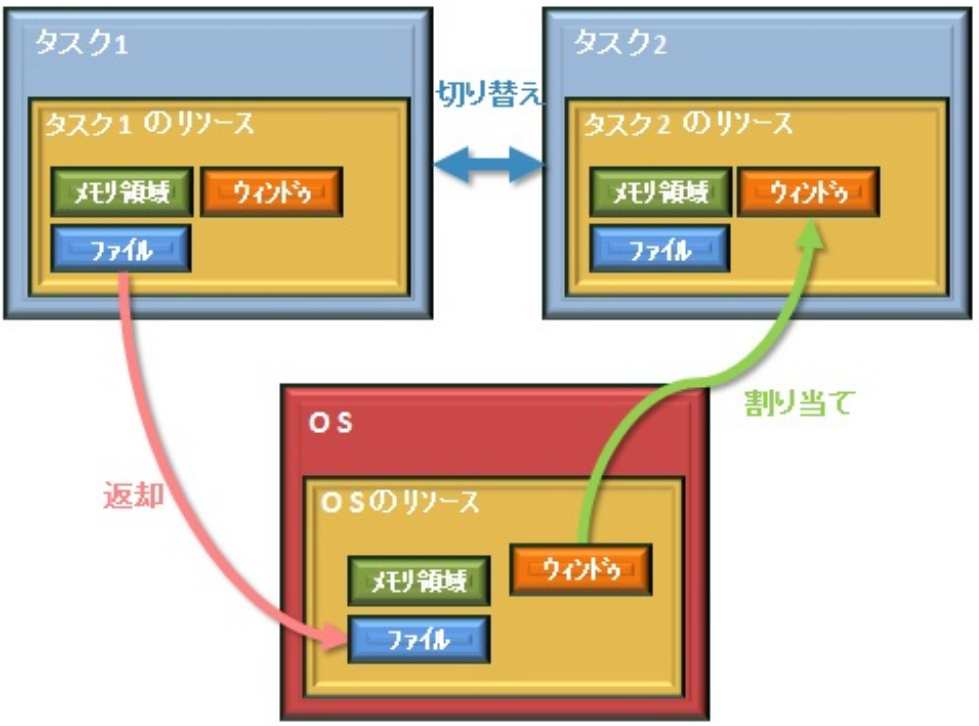
・**割り込みﾏｽｸﾚｼﾞｽﾀ**の**ﾋﾞｯﾄ**が1だと、**割り込み**が**ﾏｽｸ**される。

・ﾀｽｸ(仕事)は、ﾌﾟﾛｾｽ(処理) とも表現されている。

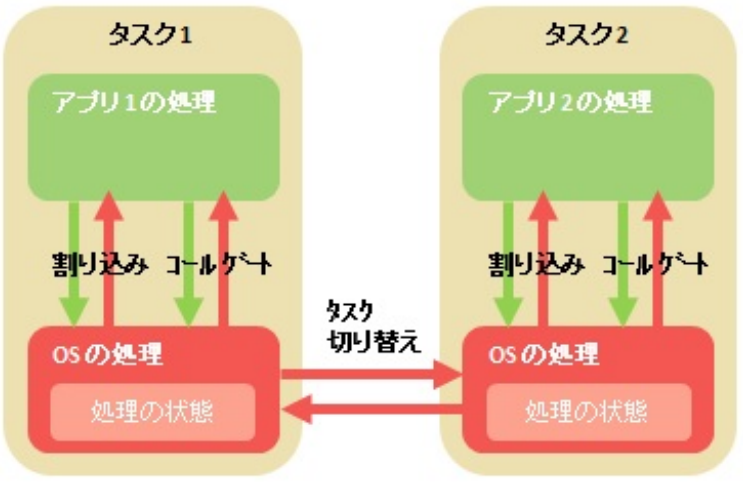
・処理系や、OSによって、呼び方が異なっていたり、
ｱｰｷﾃｸﾁｬを説明する上で、適切な方の表現が用いられている。

ﾀｽｸ切り替え

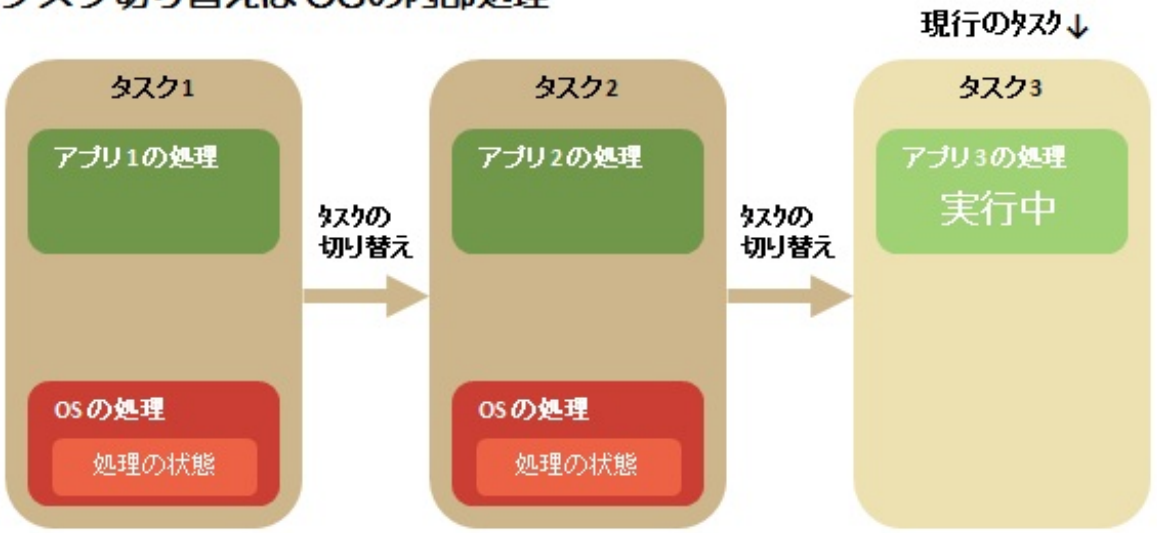
ﾀｽｸ(実行単位)**ごとに システムリソースを専有**(レンタルしている)



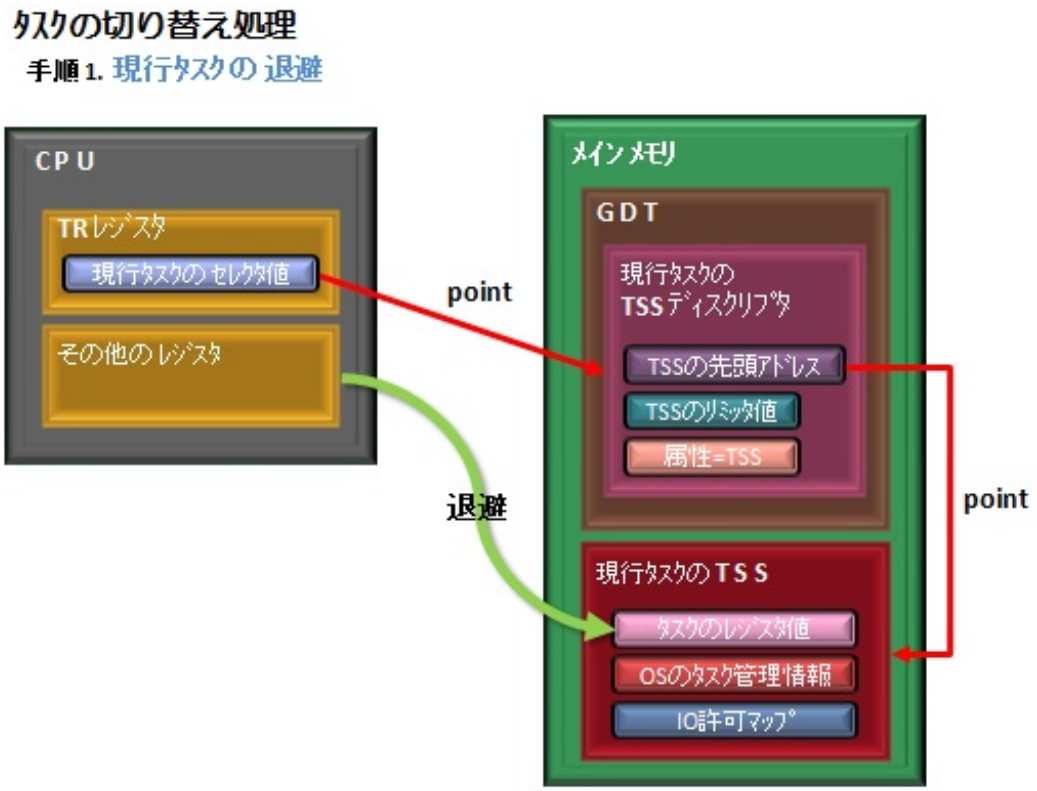
タスク 切り替え時の動作ﾚﾍﾞﾙ

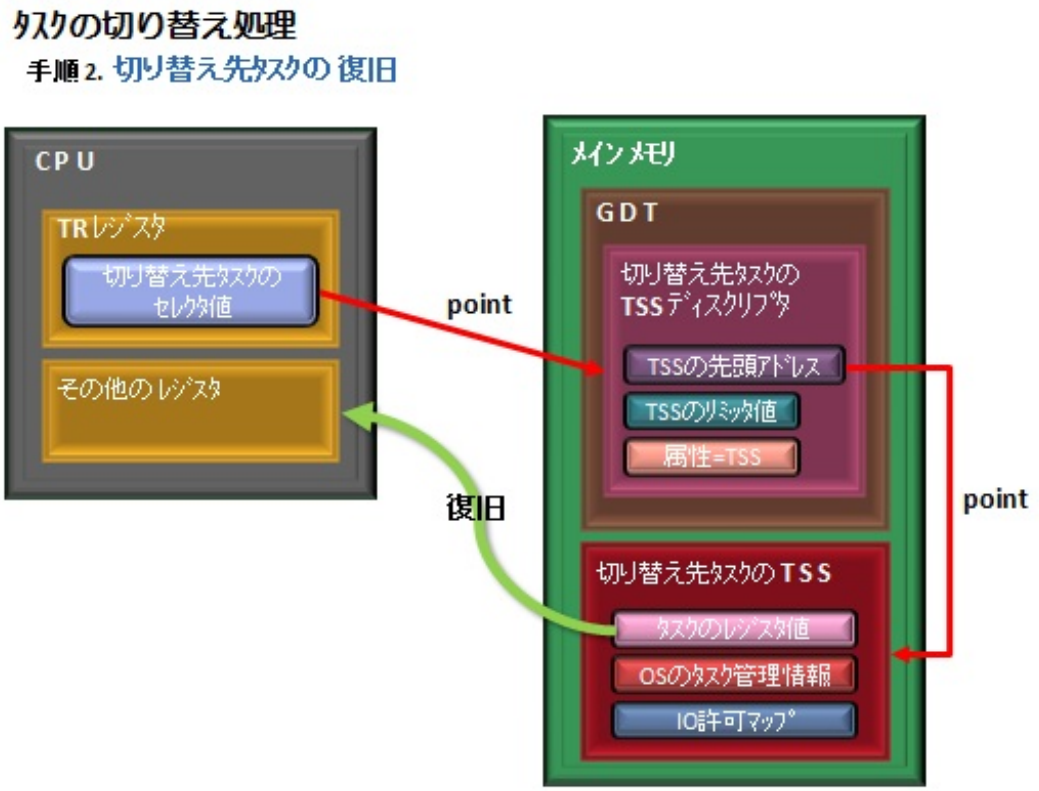


タスク切り替えはOSの内部処理



TSS



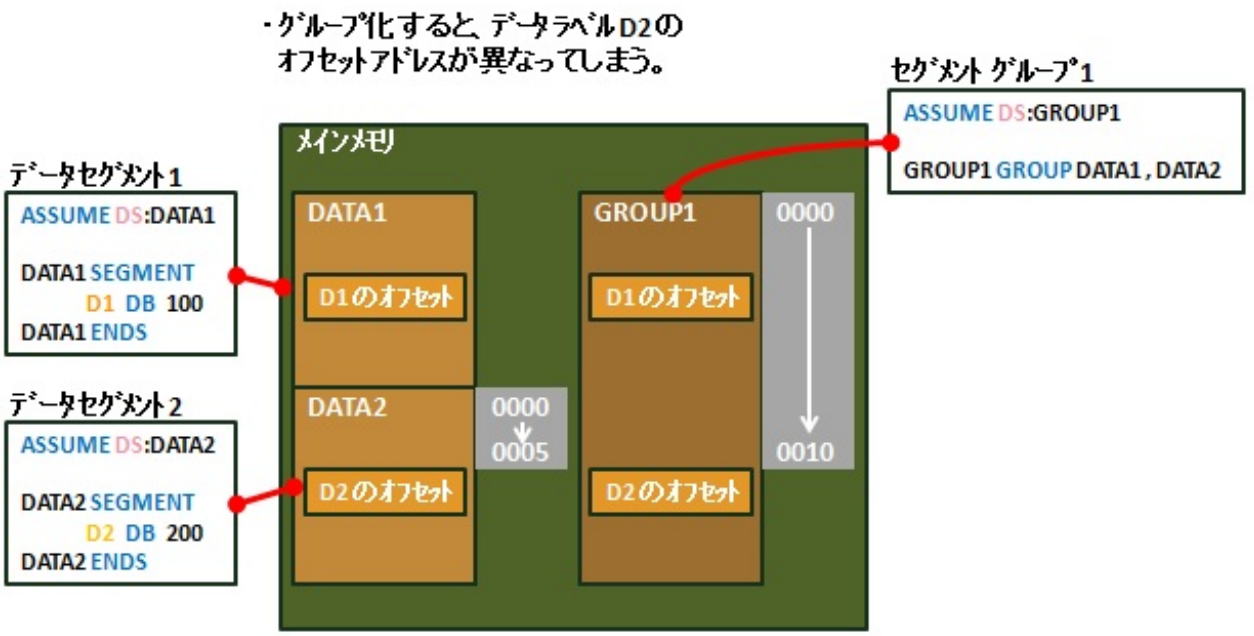


・ﾀｽｸが切り替わる時、
現行ﾀｽｸの状態(ﾚｼﾞｽﾀの値など)は、
そのﾀｽｸのTSS に保存される。

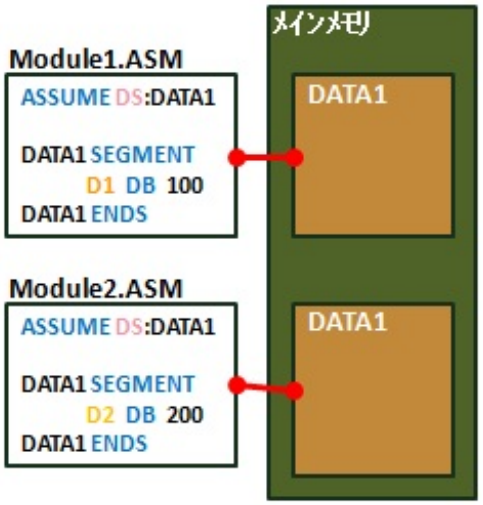
・そして、切り替わる先のﾀｽｸの状態が、

そのﾀｽｸのTSSから読み込まれる。

ASSUME擬似命令

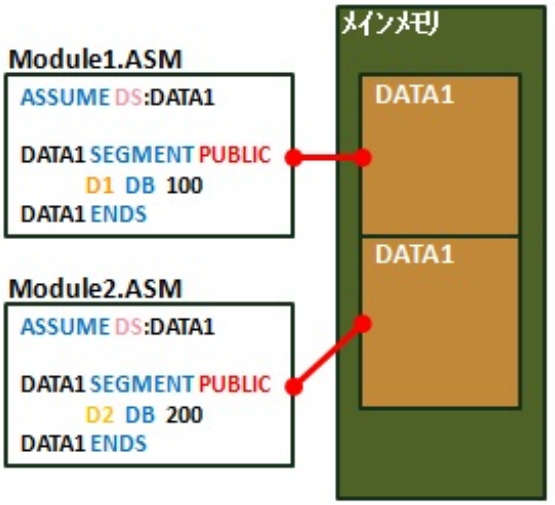

・グループ化すると、データラベルD2の
オフセットアドレスが異なってしまう。
セグメント グループ1
ASSUME DS:GROUP1
GROUP1 GROUP DATA1 , DATA2
メインメモリ
データセグメント1
ASSUME DS:DATA1
DATA1 SEGMENT
D1 DB 100
DATA1 ENDS
データセグメント2
ASSUME DS:DATA2
DATA2 SEGMENT
D2 DB 200
DATA2 ENDS
DATA1
D1のオフセット
DATA2
D2のオフセット
0000
0005
GROUP1
D1のオフセット
D2のオフセット
0000
0010

PRIVATE 属性の場合



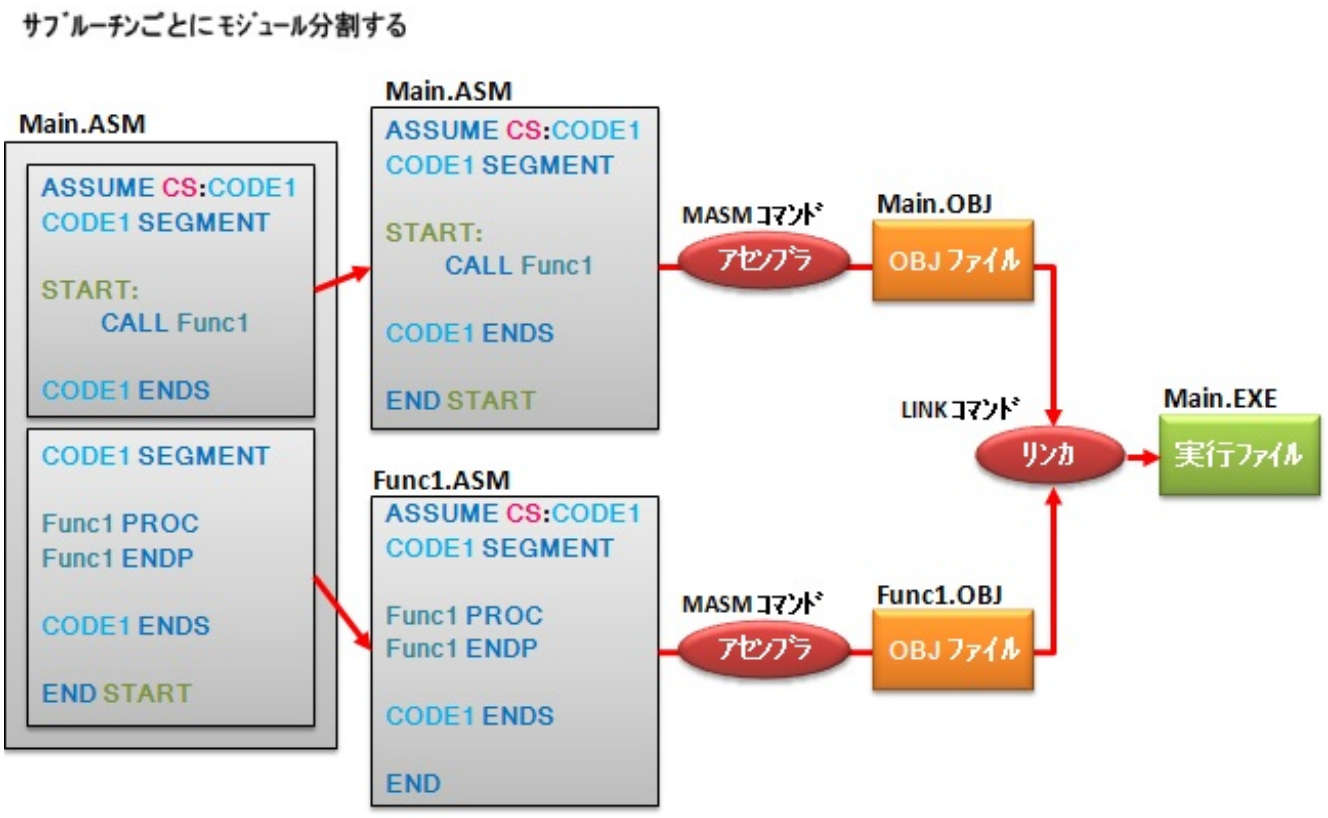
・同名のｾｸﾞﾒﾝﾄでも結合されない。

PUBLIC 属性の場合



・同名のｾｸﾞﾒﾝﾄであれは、結合される。

分割ｱｾﾝﾌﾞﾙ



ソースファイルごとに完結させ、個別にアセンブリできるように書く。
実行開始アドレスを指定するのは、メインモジュールだけでいい。

PUBLIC宣言とEXTRN宣言

外部参照の解決

・オフセットが確定すると、パブリック参照をしている箇所に配置された仮オフセットが書き換えられる。
・セグメントベースについては、リンクの段階では確定しておらず、実行時に確定される。

(※アドレス値は、わかりやすくするために、10進数で表記している。)