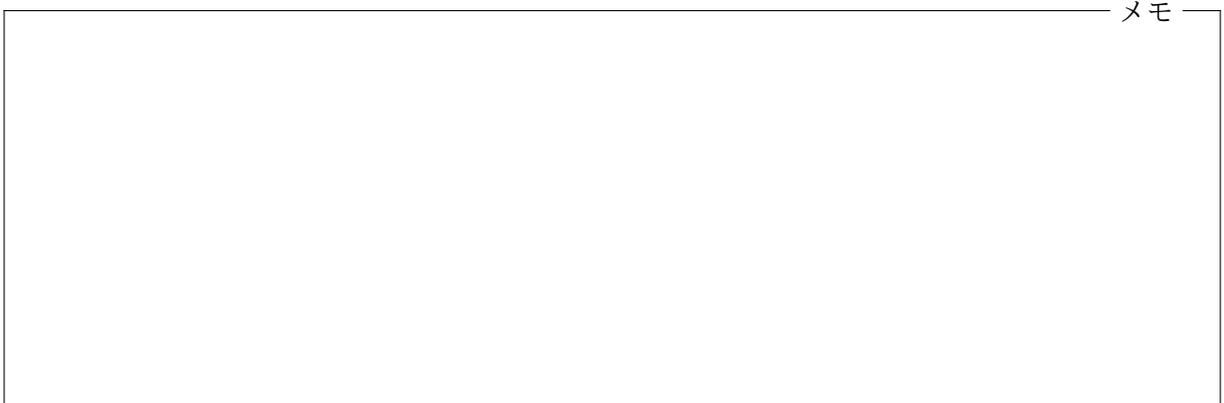


今回の内容

5.1 メモリ管理	5-1
5.2 CPU のアドレス変換機構	5-3

5.1 メモリ管理

計算機システム上でプロセス (プログラム) が実行される時に必要となるもっとも重要な資源は CPU とメモリ (主記憶装置) です。この2つなしでプロセスを実行することはできません。プロセスは、CPU がメモリ (主記憶装置) 中に記憶された機械語命令を読み取っていくことで実行されます。また、CPU にデータ転送命令を実行させることで、メモリ中に記憶された特定のデータにアクセスすることができます。第2回で説明したように、一つのプロセスがデータの記憶に使用することのできる記憶領域全体をそのプロセスのメモリ空間と呼び、そのメモリ空間中の特定の記憶場所を指定するのに使用されるアドレス (番地) の取り得る範囲をアドレス空間と呼びます。アドレス空間には、実際には使用できない (記憶装置内の記憶領域と対応していない) アドレスも含まれていますが、このことを無視して「メモリ空間」という言葉を「アドレス空間」と同じ意味で用いることもあります。このメモリ空間とアドレス空間の管理を行うのが OS のメモリ管理機能です。OS のメモリ管理の役割と機能は、大きく分けると、アドレス空間の多重化、メモリ空間の拡大、データの保護という3つの側面を持っています。



アドレス空間の多重化

PC やスマートフォンなどに使われる OS では、複数のプロセスが同時に実行されますが、このとき各プロセスが使用するメモリの記憶場所が重ならないようにする必要があります。このため、このような OS は、CPU のアドレス変換機構を利用して、各プロセスが使用する仮想アドレスをメモリモジュール中の物理的な記憶場所を指定するための物理アドレスに変換することで、異なるプロセスが使用する記憶場所を分離します。

仮想アドレス空間はプロセス毎に用意され、プロセスが異なれば、同じ仮想アドレスでも異なる物理アドレスに対応するようになります。これによってアドレス空間の多重化が実現できます。OS がプロセスを「実行中の状態」にする際には、その都度、そのプロセス固有のアドレス変換が行われるように CPU のアドレス変換機構を設定します。

メモリ空間の拡大

PCなどのOSは、ハードディスクやSSDなどの補助記憶装置を利用して、メモリ(主記憶装置)の実際の記憶容量以上のメモリ空間をプロセスに提供する働きも持っています。

機械語命令を実行していく過程で、CPUはいろいろなメモリアドレスへアクセスしますが、メモリ全体のいろいろなアドレスに全くでたらめな順番でアクセスすることは稀で、短い時間(たとえば $10\mu\text{s}$)だけを見れば、アクセスするデータはいくつかのアドレスの周辺に偏るのが普通です。CPUは機械語プログラムを読み取って、その指示を実行していきませんが、このときCPUは(分岐などがない限り)連続するアドレスを順にアクセスしていきます。また、プログラムで扱われる一般のデータに関しても、ひとまとまりのデータは連続するアドレスに置かれますから、このひとまとまりのデータを処理する場合にも、CPUがアクセスするアドレスは集中します。

ほとんどのプログラムにおいて、そのプログラムをCPUが実行した際に発生するメモリアクセスには、次のような傾向が見られます。

時間的局所性 1つのアドレスへのアクセスは、時間的に集中する傾向がある。つまり、あるアドレスへのアクセスがあったとすると、近い将来、同じアドレスへのアクセスが発生する可能性が高く、ここしばらくアクセスされていないアドレスについては、近い将来にアクセスされる可能性は低い。

空間的局所性 一定の時間内のメモリアクセスは、いくつかのアドレスの周辺に集中する傾向がある。つまり、あるアドレスへのアクセスがあれば、近い将来、その周辺のアドレスへのアクセスが発生する可能性が高く、遠く離れたアドレスへのアクセスが発生する可能性は低い。

つまり、短い時間だけをとって見ると、メモリ空間全体が均等にアクセスされていることは稀で、一部の範囲のアドレスにしかアクセスされていないのが普通です。この性質を利用して、そのプロセスが最近頻繁にアクセスしている部分にだけ主記憶装置(メモリ)を割り当てておき、あまりアクセスされていない部分のデータはハードディスクやSSDなどの補助記憶装置を使って記憶しておくようにします。補助記憶装置の方に記憶されている部分のアドレスに対するアクセスが起

こると、主記憶装置(メモリ)の一部分をその部分に割り当て直してから、補助記憶装置に退避してあったデータを主記憶装置(メモリ)上に戻し、それから CPU によるメモリへのアクセスを続行します。

主記憶装置(メモリ)は CPU から高速にアクセスすることはできますが、その容量は限られています。計算機システムで多くのプロセスが並行して動作していると、各プロセスが利用できるメモリの量は限られたものになってしまいがちですが、機械語プログラムの実行にさしあたり関係のない部分のデータを、主記憶装置から、より低速ではあるが容量に余裕のある補助記憶装置に移動することで、主記憶装置の実際の記憶容量以上のメモリ空間を各プロセスに提供することができるわけです。



データの保護

OS のメモリ管理のもう一つの側面は、メモリ中のデータの保護です。CPU のアドレス変換機構に付属するアクセス制限機能を利用して、実行中のユーザプロセスが、カーネルや異なるユーザプロセスが記憶しているデータを正当な権限なく覗き見たり、書き換えたりすることができないようにします。また、自分自身のプロセスのメモリ空間であっても、特定のアドレス領域のデータに書き込みできないようにしたり、(機械語プログラムとして)実行できないようにすることもできます。



5.2 CPU のアドレス変換機構

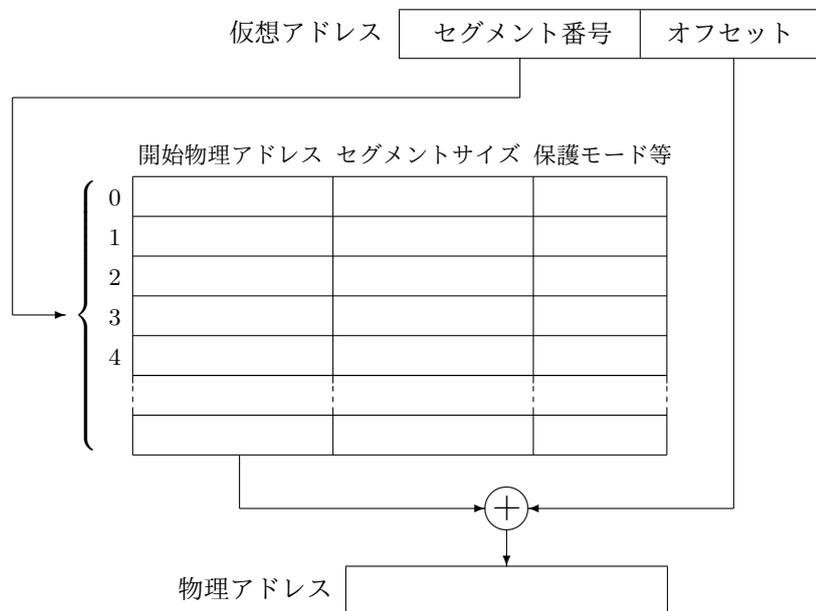
CPU がメモリにアクセスする際には、機械語命令によって指定された仮想アドレスが CPU に備わっているアドレス変換機構によって物理アドレスに変換されてメモリモジュールに伝達されます。オペレーティングシステムはこの機構を利用して、アドレス空間の多重化、メモリ空間の拡大、データの保護を実現します。

CPU 内に用意されるアドレス変換機構には、それぞれセグメント方式とページング方式と呼ばれる2つの代表的な方式が存在します。

セグメント方式 セグメント方式では、各プロセスの仮想アドレス空間をセグメントと呼ばれるいくつかの部分に分割します。セグメントごとに(0番地から始まる)アドレスの空間を用意し、セグメントの番号とそのセグメント内でのアドレス(オフセット)の対が仮想アドレスとして用いられます。セグメントのアドレスの空間の大きさはセグメントごとに異なり、プロセスの実行途中で増減させることが可能となっています。セグメント方式では、各セグメントに対して

- (1) 開始物理アドレス
- (2) セグメントの大きさ(セグメントサイズ)
- (3) セグメントの保護モード¹等の付加情報

を記憶した表を用意しておき、仮想アドレスから抽出したセグメント番号に基づいて、この表を参照し、CPUが物理アドレスへの変換を行います²。



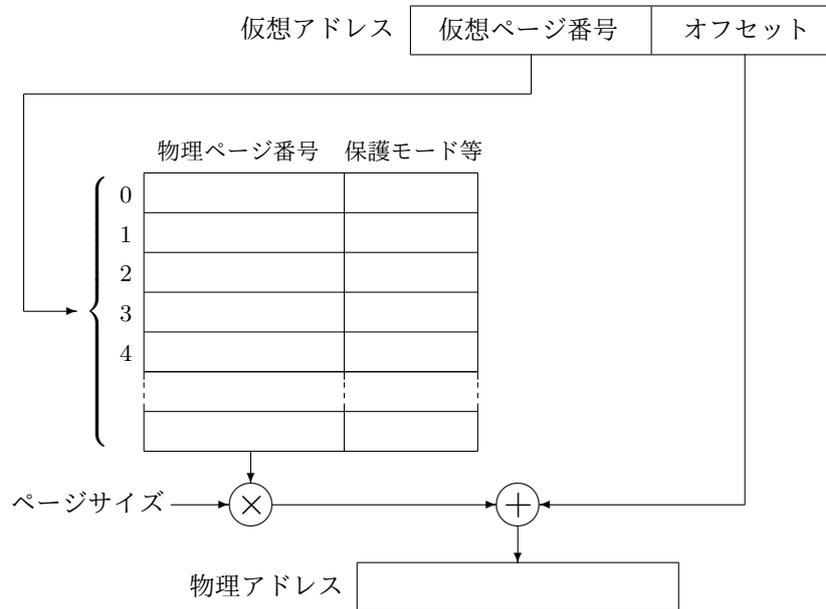
セグメント方式で得られたアドレスをそのまま物理アドレスとして用いるのではなく、さらに次節のページング方式によって変換した結果を物理アドレスとして使用するCPUもあります。



¹ そのセグメント内のデータを保護するための設定(書き込み不可など)などの情報

² オフセットがセグメントサイズを超えている場合は、通常、例外が発生します

ページング方式 仮想アドレス空間や物理アドレス空間を、ページと呼ばれる数 KiB の大きさ (たとえば 4096 byte)³の区画に分割し、ページごとに仮想アドレス空間のページ (仮想ページ) が物理アドレス空間のどのページ (物理ページ) に対応するかを表にして管理します。この表には、セグメント方式の場合と同様に、その仮想ページの内のデータを保護するための保護モードが記憶されるのが普通です。



セグメント方式でもページング方式でも、アドレス変換を行うための表は、通常メモリ中に記憶されますが、CPU がメモリへアクセスする度に、この表を見に行く (これもメモリへのアクセスとなります) のは非効率なので、第1回で解説したキャッシュメモリと同様に、表の (よく使用されている) 一部は、そのコピーが CPU 内に置かれるようになっており、高速なアドレス変換ができるようになっています。ページング方式でのアドレス変換表の CPU 内キャッシュは、**TLB** (Translation Lookaside Buffer) と呼ばれます。

³この大きさをページサイズと呼びます、複数の異なるページサイズを使用できる CPU もあります