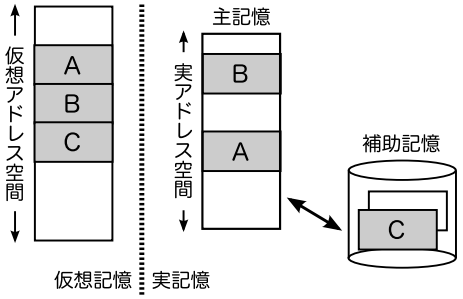
目次

★12　メモリの管理と仮想記憶（前回のつづき）

★12.2　仮想記憶

★12.2.1　仮想記憶とは

仮想記憶 (virtual memory) とは，実在する主記憶のメモリ空間（アドレス空間）と切り離された仮想的・論理的なメモリ空間のことである．仮想記憶につけた番地を**仮想アドレス**といい，仮想アドレスで指定されるメモリ空間を**仮想アドレス空間**という．また，実在の主記憶の番地とメモリ空間を**実アドレス**, **実アドレス空間**という．現代の OS では，プロセスを仮想記憶上で動作させる方式が広く用いられている．この方式には，次のような利点がある．



1. **プログラミングの手間が省ける**
   仮想記憶方式では，個々のプログラムは仮想アドレスを用いてメモリアクセスする．仮想アドレスから実アドレスへの変換は，ハードウェアや OS によって自動的に行われる．そのため，プログラムの側では，実際にコンピュータに搭載されている主記憶の容量や，自分が扱うメモリ領域が主記憶中のどこに置かれているかといったことに気を使う必要がない．
2. **主記憶の容量を超えるような大きなアドレス空間でも扱いやすい**
   プロセス毎のアドレス空間のサイズの合計，あるいは一つのプロセスのアドレス空間のサイズが，実際にコンピュータに搭載されている主記憶の容量を超えることができる (☆ 1)．これは，主記憶に入りきらない分を補助記憶装置上に置くことで実現される．あるプロセスのアドレス空間の全体または一部を主記憶／補助記憶のどこに置くかは OS が自動的に決定して管理してくれる．
3. **記憶保護が容易になる**
   例えば，仮想アドレス空間をプロセス毎に独立に持たせるようにしておけば，あるプロセスが間違えて他のプロセスのアドレス空間にアクセスしてしまうようなことを防げる．このように，仮想記憶を用いると**記憶保護**がきちんとできる (☆ 2)．

☆ 1) ［発展］仮想記憶の実用化の前にも，プログラムのメモリ領域をいくつかの部分に分割して一部を補助記憶に追い出す**オーバレイ**という方式でこのようなことは実現されていた．スワッピングも目的は同様である．

☆ 2) きちんと記憶保護をしていないと，応用プログラムが間違えて OS の領域を書き換え，システム全体がクラッシュしてしまうこともある．

★ 12.2.2 ページングとアドレス変換

仮想記憶では，アドレス空間を一定サイズの**ページ** (☆ 3) に分割してこれを単位として扱う方式が代表的である．これを**ページング**という．ページング方式の場合，以下に示すように，仮想アドレスも実アドレスも「**ページ番号**」と「**ページ内オフセット**」(ページ内での位置を表す) から構成され，**ページテーブル**と呼ばれる表を用いて仮想アドレスから実アドレスへの変換が行われる (☆ 4)．ページテーブルの管理は OS が行う．

☆ 3) ページ，ページング: page, paging．1 ページのサイズは 4KB のことが多い．

☆ 4) このような変換はハードウェアで高速に実行できるようになっていることが多い．

仮想記憶における
アドレス変換の例

仮想アドレス: 8bit
実アドレス: 5bit
ページサイズ: 4 語

アドレスの下位 2bit がページ内オフセット，
上位 bit がページ番号

仮想アドレス
000000 00
000000 01
000000 10
000000 11
000001 00
000001 01
000001 10
000001 11
:
111111 10
111111 11

ページテーブル

仮想ページ番号	有効 bit	実ページ番号
000000	1	101
000001	1	000
000010	0	
:	:	:
111111	0	

● 仮想アドレス 00000110 へのアクセス
→ 実アドレス 00010 に対応付けられる
● 仮想アドレス 00001001 へのアクセス
→ ページフォルト → そのページを主記憶へ
→ ページテーブル更新 → 実アドレスを…

実アドレス	内容
000 00	ADDA GR1,GR2
000 01	:
000 10	123
000 11	'H'
001 00	
001 01	
001 10	
001 11	
:	:
111 10	
111 11	111

一般に，全ての仮想ページを主記憶上に置いておくことはできず，一部のページは補助記憶上に置かれることになる．アクセス先のページが主記憶上にない (対応する実ページ番号が存在しない) ことを**ページフォルト** (page fault) という．そのようなページはページテーブルの「有効 bit」で区別される (☆ 5)．

UNIX ではプロセス毎に独立した仮想アドレス空間を持つようになっている．すなわち，2 つのプロセスの同じ仮想アドレスがそれぞれ異なる実アドレスに対応づけられる．

☆ 5) この授業では省略したが，キャッシュメモリでもこの例と類似の仕組みが用いられる．調べてみよう．

★ 12.2.3 ページの置き換え

主記憶がいっぱい (全ての実ページが仮想ページに対応づけられた状態) のときにページフォルトが発生したら，いずれかのページを補助記憶に追い出して，必要なページを主記憶上にコピーしなければならない．補助記憶へのアクセスは主記憶に比べて桁違いに遅いのが普通なので，このようなページの置き換えはなるべく少ない方がよい (☆ 6)．そのため仮想記憶では，主記憶から追い出すページの選択法を工夫することが重要となる．

このページ選択のアルゴリズムとしては，**LRU 法** (☆ 7) がよく用いられる．これは，「最近いちばん使われていないページを置き換える」というものである．

☆ 6) 主記憶容量が少ないと，ページ置き換えが頻発して通常の処理がままならなくなる，**スラッシング**という状態に陥ることがある．

☆ 7) LRU: Least Recently Used．LRU 法は，キャッシュメモリの解説時に登場した「メモリアクセスの時間的局所性」を活かすアルゴリズムといえる．

Q1. 主記憶上に A,B,C,D,E という 5 ページが存在しており，最近のアクセスを古い方から並べると C,E,B,A,E,D,C という順だった．この状況でページフォルトが発生した場合，LRU 法ではどれが新しいページに置き換えられるか．また，続けて再度ページフォルトが発生した場合，今度はどれが置き換えられるか．
