18 仮想記憶におけるユーザインター

18–1 プログラマとのインターフェース

メモリ管理に関しては、
OSがプログラマに提供する機能は、システム運用やシステムプログラムの開発に関するものが幾つかあるだけである。 例えば次の通り。

- 動的なメモリ確保 · · · プログラム実行時にメモリ領域を確保する機能。
- 仮想記憶制御 · · · ページイン/ページアウトの明示的な指示、等。
- 共有メモリの利用 · · · System V系のIPC, ...

動的なメモリ確保：

プログラム実行時にメモリ領域を確保する機能。

- C言語の標準ライブラリ `malloc()`, `free()`, `realloc()`, `calloc()`, ...
- UNIXのシステムコール `brk()`, `sbrk()`,...

野口「オペレーティングシステム」9.4節によれば：

UNIXのシステムコール`brk()`は、プロセスに割り当てられたヒープ領域 (i.e. 動的なメモリ割当てのために用意された領域) を伸ばしたり縮めたりしたい時に使われるらしい。 `brk()`が呼ばれると、ヒープ領域の最上位アドレス (break値と言う; 先頭アドレスのこと?) が引数で与えられた値に変更される。

動的なメモリ確保のために`malloc()`が呼ばれると、`malloc()`ライブラリ関数は通常は予め用意されているヒープ領域から必要なメモリを動的に切り出してその先頭番地を返す。 しかし、ヒープ領域が空になることもあり、その時は`malloc()`の処理本体はシステムコール`brk()`を呼び出し、システムから追加のメモリ割当てを受ける。

仮想記憶制御：

システムプログラムの性能保証のために、次の様な機能をシステムプログラマに提供しているOSもある。

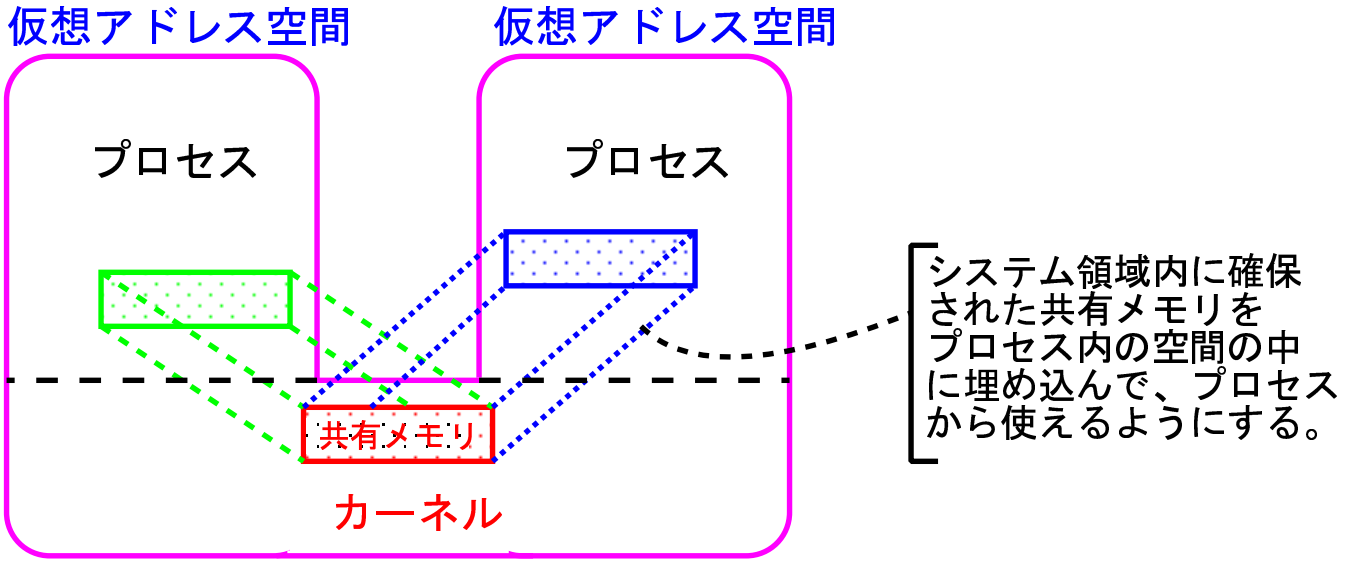
機能	用途
仮想アドレスと実アドレスを一致させる。	仮想記憶以前のプログラムを実行させたい場合などのため。
プロセス空間の全てに実ページを割り当て、ページングの対象外とする。	入出力装置との関係でページフォールトの遅延が許されない場合などのため。
指定領域を実ページに固定しページアウトの対象外にする。	ページフォールトによる遅延を防止。
指定領域を実ページにロードする。	ページフォールトによる遅延を防止。
指定領域をページアウトする。	指定領域を2度と参照しないことが分かっている場合。
指定領域の内容を解放する。	指定領域を2度と参照しないことが分かっている場合。

共有メモリの利用：

通常、1つのプロセスで確保したメモリは別のプロセスからはアクセス出来ない様になっている。 しかし、複数のプロセスが同一の実メモリにアクセスできる仕掛けが用意されていることがある。

(UNIXの場合 $\Longrightarrow$ 15.1節)

- 複数のプロセスで共有するデータを置くための領域 (システム共有領域と呼ぶ) をOS内に用意し、プログラムからの要求に応じてOSが共有メモリ領域の割り当て等を行う。



補足：

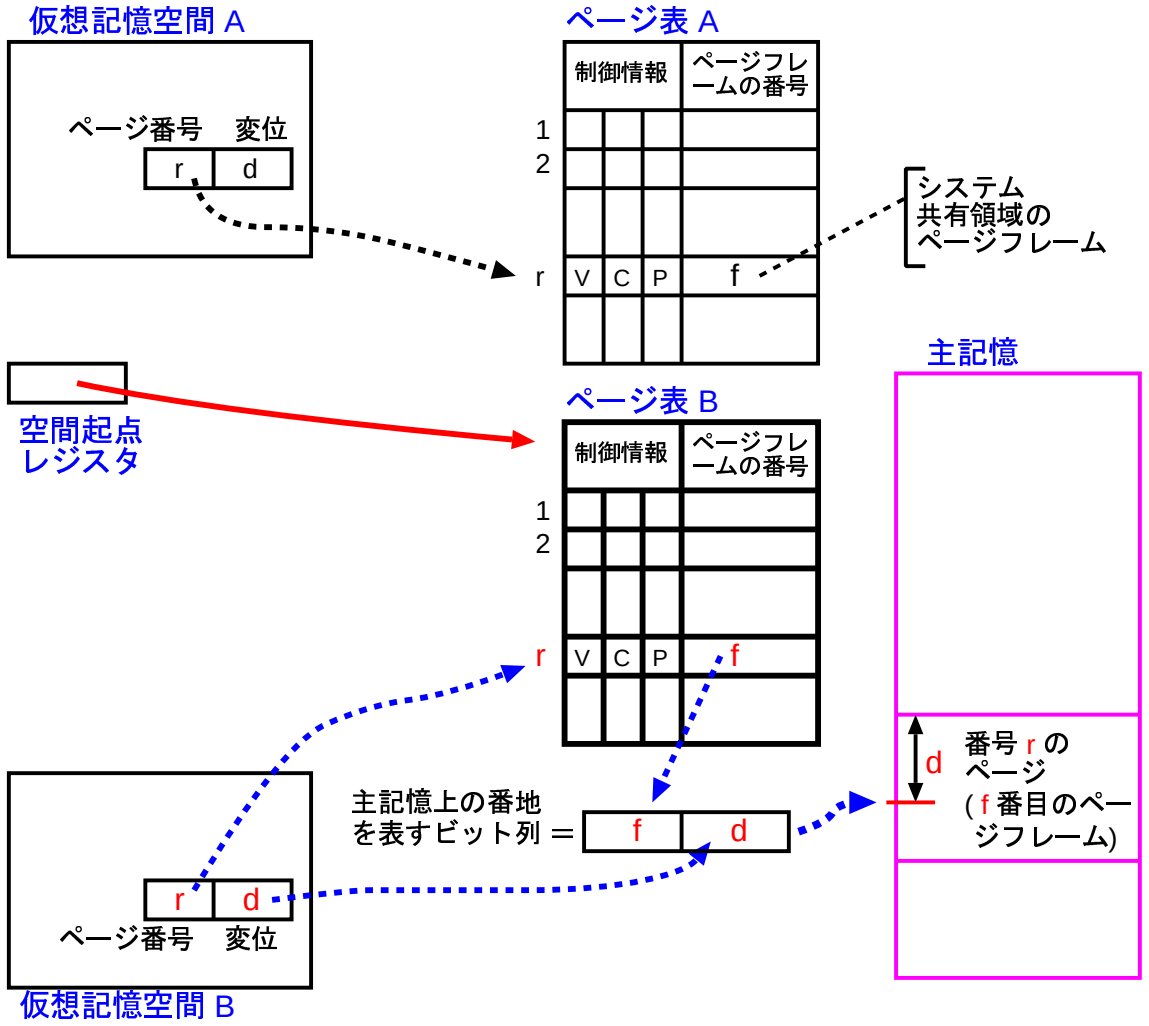
プロセス間のメモリ保護機構に抜け道を作るのは好ましくない。

⇨ 共有領域はOSが管理する領域内に確保するしかない。

- ページングによる仮想記憶の下で複数のプロセスがメモリを共有する仕掛けとしては、次の2つの方法が考えられる。

(方法1) プロセスの仮想アドレス空間の一部と共有メモリとの対応付けをプロセス毎に独立に行う。

⇨ 共有するのは実ページだけで、仮想メモリ空間上では(多分)違うアドレスに共有メモリが配置される。



(方法2) アドレス変換表のうち、システム共有領域に相当する部分を全く同一にしておく。

この場合、

管理を容易にするために、システム共有領域に関するアドレス変換表を別に用意し、その変換表そのものを共有してしまうのが良い。

⇨ 仮想メモリ空間上の同じアドレスに共有メモリが確保されることになる。