OS 第9回予習資料　　第9回の授業までに空欄を埋めて予習　　質問は事前＋当日受付けます

第9回　デマンドページング

（前回の復習） メモリの容量が足りない　⇒　オーバーレイ　⇒　①プログラミングの手間＋②環境変わると書換え必要

9－1．　デマンドページングの仕組を説明してみよう

デマンドページングの場合は、メモリのイメージはメインメモリ上ではなく、（　　）に置かれている。

このイメージと、メインメモリと、両方とも同じ大きさの（　　）に分割されている。

このイメージが分割されたものを（　　）、メインメモリが分割されたものを（　　）と呼ぶ。

≪ここで、メインメモリはまだ空っぽ（空白）であることに注意≫

CPUがイメージ上の欲しい部分にアクセスしようとすると、メインメモリ上には無いので、（　　）。

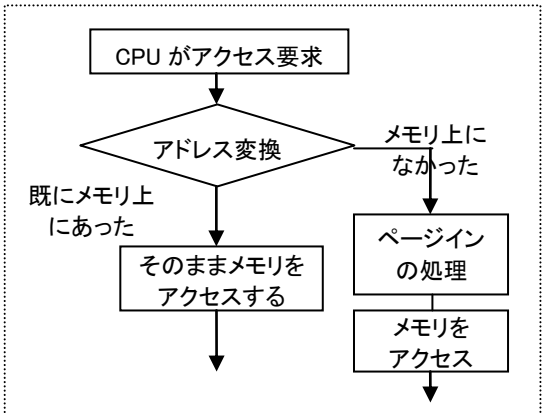
CPUが次々にアクセス要求を出すと、メインメモリ上には（　　）。

KIOSK の本屋のイメージだと、頻繁に売れる本は（　　）にあるので、渡すまでの時間は（　　）。滅多に売れない本は（　　）ので、渡すまでの時間は（　　）。平均のサービス時間を考えると、そこそこ短くて済む（後でちゃんと計算する）。

このとき、メインメモリ上のブロックは（　　長）であり、かつアドレスの連続性は（　　）。 そのためにフラグメンテーションが（　　）。

9－2．　ページ・インの処理

デマンドページングの全体の処理の流れを右の図のように描くことができるでしょう。



(1) アドレス変換とは、どういう処理でしょうか？

誰が ……

何を ……

どのように変換？ ……

(2) 「すでにメモリ上にあった」「メモリ上になかった」とはどういう意味？

……

……

(3) ページ・インの処理の具体的な内容・手順（何を、どういう順番でするか）を説明してください

……

……

……

4) ページ・インされた結果、メインメモリ上に存在するページは、

① メインメモリのアドレス（＝物理アドレス）で（アドレス順に）見たときはどのように並んでいますか

……

② 仮想空間上のアドレス（＝仮想アドレス）で（アドレス順に）見たときはどのように並んでいますか

……

(5) デマンドページングの全体の流れの中で、ハードウェアで行う部分と、ソフトウェア（プログラム）で行う部分を、それぞれ指摘してください。

……

……

……

(6) ページフォルト割り込みについて

① ページフォルト割り込みとは何ですか？説明してください。

(こんな時に起こる)

(こんな割り込み処理をする)

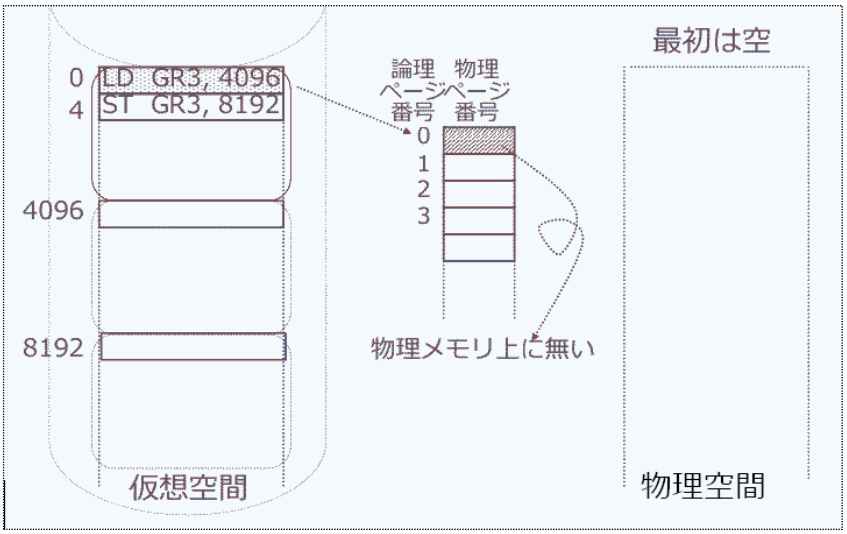
② (脱線話題 ～ 興味のある人だけ) 通常の割り込み(入出力割込みなど)では、割り込み発生時に実行していた命令はその最後まで実行し、次の命令に移るときに(次の命令の代わりに)割り込み処理プログラムの先頭へジャンプし、割り込み処理終了後は次に実行するはずだった命令の先頭に復帰します。 ところが、ページフォルト割り込みの場合は、割り込み発生時に実行途中であった命令の実行は中断し、割り込み処理プログラムへジャンプし、割り込み処理終了後は中断した命令の先頭からもう一度その命令をやり直します。何で対応が違うのか、理由を説明してください。

9－3. 実際の動作イメージ

ここでは、実際の命令実行の中でデマンドページングがどのように動くのかを、かなり細かい目で理解して欲しいと思います。

右の図はデマンドページングの状況を表しています。左側が仮想空間で、0～3 番地に LD 命令が、4～7番地にST命令が入っています。右側は物理空間で、最初は空です。

CPU が 0 番地から実行を始めるとしましょう。その時起こることを、書き出してみてください。



①CPU は 0 番地にある命令を実行しようとする

②CPU は次の命令に進む(＝4 番地にある命令を実行しようとする)

９－５．　デマンドページングの性能モデル

(1) ヒット率とは何か、説明してください

ミス率とは何ですか。

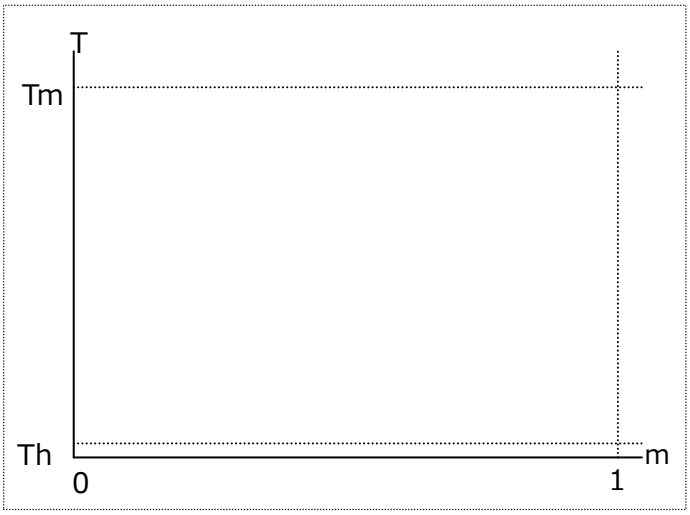
(2) デマンドページングにおける、アクセス時間の期待値は、どのような式で書けるか、説明してください

（なぜその式になるのか、説明してみてください）

９－６ / ９－７ / ９－８．　ミス率/ヒット率と局所性

(1) ミス率/ヒット率と性能の関係を考えてみましょう。

まず、1回のアクセスにかかる時間のモデルを設定します。ヒットしたときはメインメモリにアクセスするだけなので、アクセス時間 Th は半導体メモリの時間程度になりますが、簡単のために仮に Th＝1ナノ秒（＝10^{-9}秒）としましょう。他方、ミスしたときはハードディスクから読み出してこなければならないので、アクセス時間 Tm はディスクのアクセス時間程度になりますが、簡単のため Tm＝10ミリ秒（＝10^{-2}秒）としましょう。 これで、ヒット時とミス時のそれぞれのアクセス時間が決まりました。



９－5(2)の式を使って考えます。ミス率 m を0から1まで動かしたときに、メモリアクセスの性能（アクセスにかかる時間 T）がどう変わるかを、①式に整理し、②グラフに描いてみてください。

(2) 実用を考えると、メモリアクセスの時間（期待値）T は、デマンドページングが無いとき（＝すべてヒットするとき）と大きく変わらない（大きく増えない）ことが条件になります。なぜなら、このメモリアクセス時間がシステムの性能を決めるからです。具体的な数値イメージを作ってみましょう。 もし、

① 総アクセス数の 1/2 がヒット、1/2 がミスだとすると、アクセス時間の期待値 T はヒット時 Th の何倍になるでしょうか

② メインメモリ（物理空間）の容量が、ハードディスク（仮想空間）の 1/10 であったとします。つまりメインメモリの 10 倍の大きさの仮想空間を用意したとします。 このとき、もしすべての仮想アドレス上にランダムにアクセスするとすると、単純には総アクセス数の 1/10 がヒット、9/10 がミスとなる（つまりヒット率が 1/10）と考えられます。 このときのアクセス時間の期待値 T はヒット時アクセス時間 Th の何倍でしょうか。

③ 逆に、期待値 T がヒット時アクセス時間 Th の2倍になるときのヒット率はどのぐらいでしょうか？ 1．5倍になるときのヒット率はどのぐらいでしょうか？

(3) アクセスの局所性とはどういうことか、説明してください。

(4) デマンドページングではページ単位で物理メモリにコピーするので、命令部分についてはページに含まれる1つの命令を1回アクセスすると、そのあとはそのページに含まれる残りすべての命令がヒットになります。 ページの大きさを4096 バイト、1つの命令の長さを(仮に固定長で)4 バイトであるとすると、ページの先頭にある命令にアクセスすると、疎のアクセスは「ミス」になりますが、それ以降の 1023(4096÷4−1)命令分のアクセスは「ヒット」になります。
この状況があるだけで、そのページ中の全部の命令を実行するためのアクセス回数(　　)回に対してミス回数は先頭の(　　)回になるので、ミス率は(　　)になります。

(5) もう少し細かく考えると、プログラム中にはよくループがあります。 もっとも内側のループの部分が、もっとも実行回数が多くなることは自明です。 もっとも内側のループが1ページの中に納まっていると、上の(4)のケースよりもっとページの中にアクセスがとどまっている割合が増えます。 仮に、1ページ(1024.命令)の中の 102 命令(1/10)を 1000 回回るループがあったとしましょう。 このページの先頭に飛び込んだ時に1回ページのミスを起こしてからこのページを出るまで(次のページミスが起こるまで)の実行命令数(この間はすべてヒット)を数えてみましょう。
(A) このページの中のループ以外の命令は、1023(ページ全体)—102(ループ部分)＝(　　)命令です。
(B) ループ内での命令のアクセス数は、102(命令)×1000(ループ回数)＝(　　)です。
(A)と(B)とを合計すると、このページ内のヒットアクセスの数は(　　)です。 これに対してミスは先頭の1命令だけですから、ミス率は (　　)になります。

というわけで、現実のプログラムのミス率はかなり低くなることが期待できます。 最終的には実測によって確認されています。
なお、命令実行に伴うメモリアクセスのうち、上記で取り上げたのは命令読み出しの部分だけで、命令実行フェーズでのデータのアクセスは数えていません。データアクセスは、命令読み出しに比べて規則性が少ない(先頭から順番にアクセスするという規則がない)ので、上記(4)(5)のような偏り(局所性)は出にくくなります。

９－９.　ワーキングセットの考え方を簡単に説明してください。