

オペレーティングシステム試験問題

(配点：(1) 15 点, (2) 20 点, (3-1) 8 点, (3-2) 7 点, (4) 10 点, (5-1) 8 点, (5-2) 6 点, (5-3) 6 点,
(6-1) 8 点, (6-2) 6 点, (6-3) 6 点)

【注意】問題(1)～(3)と(4)～(6)はそれぞれ別の答案用紙に解答すること。

- (1) 読取り速度 10 枚/秒のカードリーダーと、プリント速度 20 ページ/分のプリンタが各 1 台ある単一 CPU の計算機システムを考える。このシステムで、以下の 3 つのジョブを実行するとき、最後に終了するジョブの終了時刻を最短にするスケジューリングを図で示し、各ジョブの終了時刻を答えよ。ただし、各ジョブは時刻 0 で到着し、各ジョブは入力・計算・出力の処理をこの順番で実行し、それらの処理中では横取りは生じないものとする。

ジョブ	入力カード枚数	計算時間	出力ページ数
ジョブ A	100 枚	300 秒	20 ページ
ジョブ B	100 枚	10 秒	100 ページ
ジョブ C	1000 枚	20 秒	10 ページ

- (2) 単一 CPU のマルチプログラミング環境におけるプロセスのスケジューリングについて考える。マルチプログラミング環境では、通常、複数の実行可能なプロセスがレディ(ready)キューに登録され、CPU が空いたときキューから 1 つプロセスが取り出されて実行される。プロセス P1～P4 が到着時刻と処理時間が以下の表のように与えられているとき、

- ・ 到着順(First Come First Served: FCFS)
- ・ 処理時間順(Shortest Processing Time First: SPT)
- ・ 残余処理時間順(Shortest Remaining Time First: SRT)
- ・ ラウンドロビン(Round Robin: RR)

のそれぞれのスケジューリング方式について、各プロセスのターンアラウンドタイム(プロセスが到着してから終了するまでの時間)とそれらの平均値を求めよ。ただし、時刻 0 で CPU は空いているものとする。また、ラウンドロビンでは、タイムスライスを 10 として、新規到着プロセスは到着後の次のタイムスライスで実行され、実行中のプロセスが終了した場合は次のタイムスライスまで待つて別のプロセスをディスパッチするものとする。

プロセス	到着時刻	処理時間
P1	0	50
P2	10	10
P3	16	3
P4	26	8

(3) 単一 CPU のマルチプログラミング環境で、次の 2 つのプロセス P1 と P2 が並行に動作している状況を考える。

P1

```
wait(s1);           ①
buffer = a1[0];     ②
signal(s2);         ③
wait(s1);           ④
buffer = a1[1];     ⑤
signal(s2);         ⑥
```

P2

```
wait(s2);           ⑦
a2[0] = buffer;     ⑧
signal(s1);         ⑨
wait(s2);           ⑩
a2[1] = buffer;     ⑪
signal(s1);         ⑫
```

上のプログラムで、buffer は整数型変数、a1, a2 は大きさ 2 の整数型の配列(a1[0] と a1[1] には既に初期値が設定されている)で、s1, s2 はセマフォ(初期値は、s1=1, s2=0)であり、buffer, s1, s2 は P1 と P2 で共有されているものとする。また、wait と signal はセマフォについての操作であり、それぞれ次のような動作をする。

wait(s): セマフォ s の値が正なら値を 1 減らして次の文へ進む。s の値が 0 ならブロックする(プロセスの実行が中断する)。

signal(s): 以前にセマフォ s で wait を実行してブロックしているプロセスがあるとき、そのプロセスのうち 1 つの実行を再開させる(レディキューにつなぐ)。ブロックしているプロセスがないときは、s の値を 1 増やす。

このとき以下の小問に答えよ。ただし、以下では初期状態として、プロセス P1, P2 が共にレディキューにつながれており、最初に P1 がディスパッチされるものとする。また、プロセスのスケジューリングではプリエンプション(横取り)はなく、wait 操作によるブロック以外はプロセスの実行は中断しないものとする。

(3-1) プロセス P1, P2 の文①～⑫はどのような順番で実行されるか答えよ。

(3-2) P2 のプログラムで s1 と s2 を入れ替えて、⑦、⑩の行を wait(s1)に、⑨、⑫の行を signal(s2)に置き換えると、P1, P2 の実行はどのようなになるか答えよ。

(4) 外部断片化と内部断片化は、メモリシステムやファイルシステムにおける領域割当において、その実現方法によってたびたび発生するものであり、どちらも未使用領域があるにも関わらず、その領域が利用できないという問題を生じる。一般的に、ある領域割当方法において発生する外部断片化とは、どのような状況を指すか、また、どのような不具合が生じるかを簡潔に説明せよ。

(5) 仮想記憶におけるページング方式について、以下のプログラムを参照しながら、間に答えよ。なお、このプログラムが動作するコンピュータの条件として、以下を前提に考えよ。

(a) 仮想記憶としてページング方式が採用されており、プログラム実行開始時には割付け済みのページ枠はない。

(c) ページ置換えアルゴリズムとして LRU を採用し、変数 data はアドレス 0 におかれる。

(b) 整数型 (int) は 4 バイト、ページサイズは 2KB (1KB は 1,024B) とする。

(d) 変数 i, j については、レジスタにその記憶領域が割り当てられる。

(e) 命令フェッチに関するページ参照は考えなくてよい。

```
main(){
    static int data[256*256];
    register int i, j;
    for (i = 0; i < 256; i++)
        for (j = 0; j < 256; j++)
            data[i + 256*j] = 0;
}
```

(5-1) 上記プログラムのページ参照ストリングはどのようなになるか示せ。連続する同一ページへの参照に関しては、ページ枠のストリング列「3, 1, 1, 1, 2」を「3, 1, 2」などと略記してよい。

(5-2) ページフォールト回数を減らすために、プログラムに改良を加えよ。

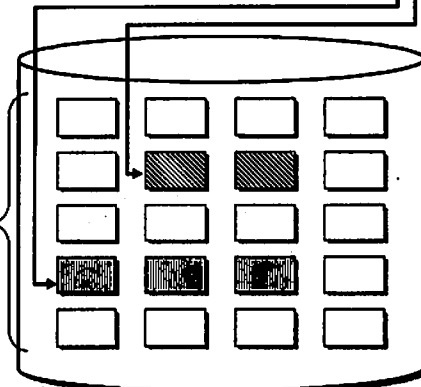
(5-3) 実際には LRU 方式をハードウェアで実現しようとするとは複雑になるので、簡易版が用いられている。どのようなものがあるか例を挙げて簡単に説明せよ。

(6) ファイルシステムを実装する方法にはいくつかある。例えば、単純な「連続割付け方式」では、ファイルをディスク上の連続したブロックに隙間なく格納し(図参照)、その場所を記憶するために、ディレクトリに先頭ブロックの位置のみを記録する。しかし、この方式には以下の欠点がある。ファイル生成時にその最大サイズを決める必要があるため、ディスク利用効率が悪くなる。また、ファイルが大きくなった場合は、より大きな領域を別に確保し、内容全体を移動しなければならない。さらに、ファイルが削除されて空き領域ができた場合にも、削除されたファイルのサイズ以下のものしかおけない。そこで、連続割付け方式以外の方法として、以下の3つの方法が考えられる。

ディレクトリ

ファイル名	ブロック数	開始
file1	2	●
file2	3	●

データ
ブロック



(a) ブロック内にポインタを持たせ、ブロックの連結リストにファイルを格納する割付け方式(「連結リスト割付け方式」)

(b) 上記の「連結リスト割付け方式」においてブロックに格納されたポインタのリンク情報を取り出して、別に索引としてメモリ上に置く(「連結リストの索引を用いた割付け方式」)

(c) 索引自体をファイル本体と同様にディスクに索引用ブロックとして置く(「索引ブロックを用いた割付け方式」)

以上の3つの方法について、それぞれ以下の小問に答えよ。

(6-1) 「連結リスト割付け方式」は、「連続割付け方式」のどのような欠点を解決できるか、2つ述べよ。一方、新たな問題が引き起こされる。順アクセス、直接アクセス、それぞれの場合について説明せよ。

(6-2) 「連結リストの索引を用いた割付け方式」を採用した場合に、ブロックサイズを大きくすることによる利点を述べよ。また、その時の問題点を示せ。

(6-3) 「索引ブロックを用いた割付け方式」では、1つのデータブロックに納まるような小さなファイルでも、索引ブロックとデータブロックと合わせて2回アクセスが必要となる。しかし、実際には平均アクセス回数は2ではなく1に近い。その理由を述べよ。