

次の問1から問7までの7問については、この中から5問を選択し、選択した問題については、答案用紙の選択欄の(選)をマークして解答してください。

なお、6問以上マークした場合には、はじめの5問について採点します。

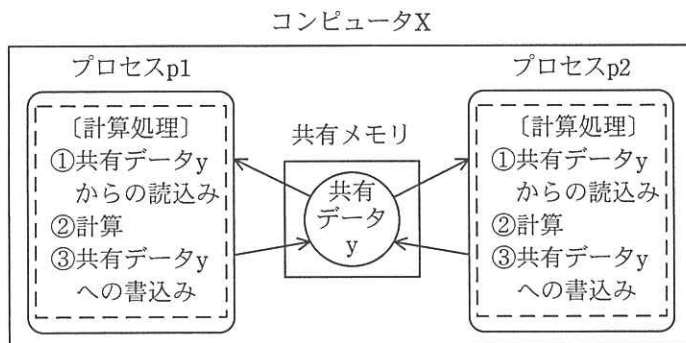
問1 プロセスの排他制御に関する次の記述を読んで、設問1～3に答えよ。

複数のプロセスが、共有するデータ（以下、共有データという）を書き換える処理を、並行して実行する場合がある。このようにプロセス間でデータを共有する方法の一つとして、プロセス間で共有するメモリ（以下、共有メモリという）に共有データを格納する方法がある。ここでは、CPU が一つで共有メモリをもつコンピュータ X 上で、二つのプロセス p1, p2 が共有メモリを使用して並行に処理を実行する場合を考える。

プロセス p1, p2 が共有データ y に対して計算処理をする場合を、図 1 に示す。ここで、各プロセスの計算処理は、次のとおりである。

〔計算処理〕

- ① 共有データ y の値を読み込む。
- ② 読み込んだ値を用いて計算する。
- ③ 計算結果を y に書き込む。



注記 “——→” は共有データへのアクセス（読み込み又は書き込み）を表す。

図1 共有メモリによるデータの共有

設問1 図1に示すプロセス p1, p2 が共有データ y に対して次の処理を実行する場合を考える。

プロセス p1 : y の値を 2 だけ増加させる。

プロセス p2 : y の値を 1 だけ減少させる。

プロセスの実行前の共有データ y の値が 5 であり, y に対する排他制御をしないとき, プロセス p1, p2 が並行に 1 回だけ処理を実行した直後において, y が取り得ない値を, 解答群の中から選べ。

解答群

ア 4

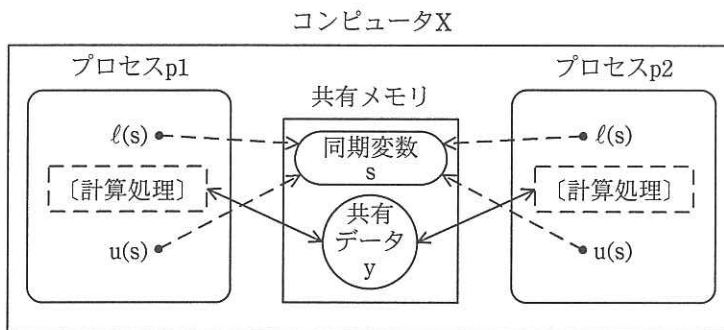
イ 5

ウ 6

エ 7

設問2 次の記述中の  に入れる正しい答えを, 解答群の中から選べ。

プロセスの排他制御の仕組みとして, 共有データがいずれかのプロセスに確保されている状態 (以下, 確保状態という) 又はどのプロセスにも確保されていない状態 (以下, 解放状態という) のいずれかの状態をもつ同期変数を使用する方法を考える。図1のプロセス p1, p2 の計算処理で使用する共有データ y に対して同期変数 s を用いて排他制御する場合を, 図2に示す。



注記 “ $\longleftrightarrow$ ” は共有データへのアクセスを表す。  
 “ $\bullet\text{---}\rightarrow$ ” は同期変数  $s$  への操作を表す。

図2 同期変数  $s$  を用いた排他制御

図 2 において、同期変数の状態を変更する関数  $l$  と  $u$  があり、同期変数を引数で指定する。各プロセスは、共有データを排他制御して計算処理をする場合、関数  $l$  の呼出し、計算処理、関数  $u$  の呼出しの順番で処理を実行する。このとき、共有データ  $y$  を排他制御するために、関数  $l$  と  $u$  の引数として同期変数  $s$  を指定する。

関数  $l$  の操作内容は  処理であり、関数  $u$  の操作内容は  処理である。ここで、“同期変数の状態を調べて、変更する処理”は中断のない処理として実行されるものとする。ただし、プロセスが待ちの状態になったら、CPU は別のプロセスに割り付けられるものとする。

### 解答群

- ア  $s$  の状態が解放状態ならば確保状態にし、確保状態ならば解放状態になるまで待ってから確保状態にする
- イ  $s$  の状態が解放状態ならば確保状態にし、確保状態ならば何もしない
- ウ  $s$  の状態が解放状態ならば確保状態になるまで待ち、確保状態ならば解放状態になるまで待つ
- エ  $s$  の状態が解放状態ならば確保状態になるまで待ち、確保状態ならば何もしない
- オ  $s$  の状態が確保状態ならば解放状態にし、解放状態ならば何もしない
- カ  $s$  の状態が確保状態ならば解放状態になるまで待ち、解放状態ならば何もしない

設問 3 プロセス  $p_1$ ,  $p_2$  が使用する共有データが二つあり、共有データ  $y_1$  に対して同期変数  $s_1$  を用いて排他制御し、共有データ  $y_2$  に対して同期変数  $s_2$  を用いて排他制御する場合を考える。プロセス  $p_1$  が、 $y_1$  の確保、 $y_2$  の確保、 $y_2$  の解放、 $y_1$  の解放の順序で同期変数を操作するとき、プロセス  $p_2$  が  $y_1$ ,  $y_2$  の確保と解放を行う順序によってはデッドロックが発生する可能性がある。デッドロックが発生する可能性のあるプロセス  $p_2$  の操作の順序を、解答群の中から選べ。

### 解答群

- ア  $y_1$  の確保、 $y_1$  の解放、 $y_2$  の確保、 $y_2$  の解放
- イ  $y_1$  の確保、 $y_2$  の確保、 $y_2$  の解放、 $y_1$  の解放
- ウ  $y_2$  の確保、 $y_1$  の確保、 $y_1$  の解放、 $y_2$  の解放
- エ  $y_2$  の確保、 $y_2$  の解放、 $y_1$  の確保、 $y_1$  の解放