問010041問題

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。

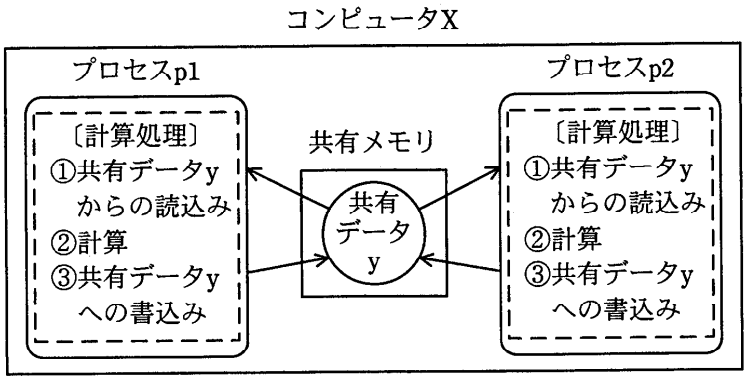
プロセスｐ１，ｐ２が共有データｙに対して計算処理をする場合を，図１に示す。ここで，各プロセスの計算処理は，次のとおりである。

〔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（読込み又は書込み）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ｐｌ，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。

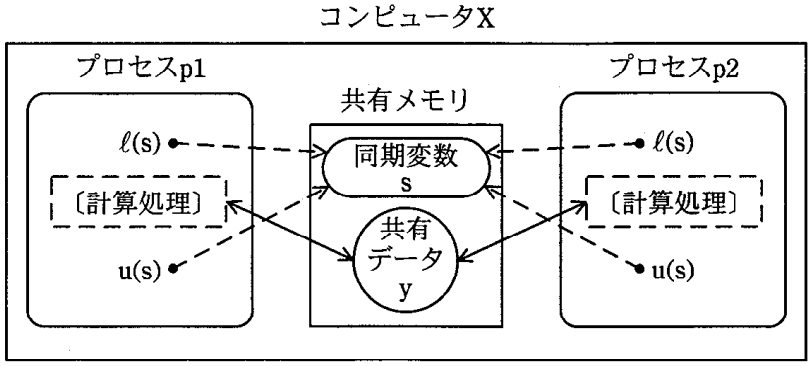
プロセスの実行前の共有データｙの値が５であり，ｙに対する排他制御をしないとき，プロセスｐ１，ｐ２が並行に１回だけ処理を実行した直後において，ｙが取り得ない値を，解答群の中から選べ。

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　“←→”は共有データへのアクセスを表す。
“●－－→”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［　ａ　］処理であり，関数ｕの操作内容は［　ｂ　］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ｐ１，ｐ２が使用する共有データが二つあり，共有データｙ１に対して同期変数ｓ１を用いて排他制御し，共有データｙ２に対して同期変数ｓ２を用いて排他制御する場合を考える。プロセスｐ１が，ｙ１の確保，ｙ２の確保，ｙ２の解放，ｙ１の解放の順序で同期変数を操作するとき，プロセスｐ２がｙ１，ｙ２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ｐ２の操作の順序を，解答群の中から選べ。

解答群

ア　ｙ１の確保，ｙ１の解放，ｙ２の確保，ｙ２の解放

イ　ｙ１の確保，ｙ２の確保，ｙ２の解放，ｙ１の解放

ウ　ｙ２の確保，ｙ１の確保，ｙ１の解放，ｙ２の解放

エ　ｙ２の確保，ｙ２の解放，ｙ１の確保，ｙ１の解放