

学生番号 _____ 氏名 _____

問 1

デッドロックの「防止」は、デッドロックの必要条件の全てが成立しないようにするものである。その考え方は、以下のどれか。

- A. 資源を昇順に要求するように制限条件をつけて、プログラミングさせることにより、循環待機の条件が成立しないようにする。
- B. 資源を使用中のプロセスがいても、要求された資源を割り当てることで、相互排除の条件が成立しないようにする。
- C. 資源待ちのため待機するプロセスについては、確保した資源を取り上げ、確保と待機の条件が成立しないようにする。
- D. 資源待ちのため待機するプロセスについては、確保した資源を取り上げ、横取り不能の条件が成立しないようにする。

問 2

資源 r_1 , r_2 , r_3 を排他的に占有して処理を行うプロセス $P_1 \sim P_4$ がある。各プロセスは処理の進行に伴い、3つの資源を表中の数値の順に要求・確保し、実行終了後に一括して解放する。プロセス P_1 とデッドロックの関係を起こす可能性のあるプロセスはどれか。

- A. プロセス P_2 B. プロセス P_3 C. プロセス P_4
- D. プロセス P_2, P_3 E. プロセス P_2, P_4 F. プロセス P_3, P_4
- G. プロセス P_2, P_3, P_4

スライド(添付ファイル:問2, 9, 10)

問2 プロセスが資源を要求する順序

プロセス名	資源 r_1	資源 r_2	資源 r_3
プロセス P_1	1	2	3
プロセス P_2	3	2	1
プロセス P_3	1	2	3
プロセス P_4	2	3	1

問9, 10 プロセスのプログラムおよび資源割当てグラフの初期値

P1:	P2:	P3:	r_1	r_2	r_3
(1) r_1 要求	(5) r_2 要求	(9) r_3 要求	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(2) r_2 要求	(6) r_1 要求	(10) r_2 要求	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
....			
(3) r_1 解放	(7) r_2 解放	(11) r_3 解放	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
(4) r_2 解放	(8) r_1 解放	(12) r_2 解放	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

処理の条件

資源型の実体は1プロセスのみが確保できるものとする(複数のプロセスが同時に確保できない)。

OSは要求された資源が空いていれば、無条件に割り当てるものとする。

問 3

今、以下の動作を行うプロセス P_1 , P_2 が並行して動作している。

プロセス P_1 : (1) r_1 要求→(2) r_2 要求→(3) r_2 解放→(4) r_1 解放

プロセス P_2 : (5) r_2 要求→(6) r_1 要求→(7) r_1 解放→(8) r_2 解放

資源 r_1 , r_2 は2プロセスが同時に確保することはできない。

この2プロセスがデッドロックとなる可能性があるのは、最初3ステップが、以下のどの順序で処理された場合か？

- A. (1)(2)(3) B. (1)(2)(5) C. (1)(5)(2) D. (5)(6)(1) E. (5)(6)(7)

問 4

デッドロックの回避は、プロセスが資源を要求した時、デッドロックの可能性をチェックする。その処理の方法に関する以下の説明で、不適当なものはどれか。（紛らわしいので注意）

- A. 資源が他のプロセスに確保されていなくても割当てないことがある。
- B. 銀行家アルゴリズムにより、安全な順序を探す。
- C. 資源を割当てると、デッドロックになるかどうかを判定する。
- D. 資源要求の度にチェックが必要であり、処理負荷が大きい。

問 5

デッドロックの回避では、プロセスが資源を要求する度に銀行家アルゴリズムを実行する必要がある。しかし、資源型の実体の数が全て 1 個の場合、より簡単なチェック方法がある。以下のどれか。

- A. 資源を要求する度ではなく、定期的に銀行家アルゴリズムを実行する。
- B. 資源割り当てグラフの資源型の実体の数を 1 個にし、割当てると循環ができるかどうかをチェックする。
- C. 資源割り当てグラフの資源型を省略した待ち合せグラフを作成し、割当てると循環ができるかどうかチェックする。
- D. 資源割り当てグラフに要請枝を追加し、割当てると循環ができるかどうかをチェックする。

問 6

実体の数が 1 個の資源型 R1, R2 を使用するプロセス P1, P2 が並行して動作している。デッドロックの回避アルゴリズムを行うシステムが、資源が空いていても割り当てを行わないのは、以下のどの場合か。

（ヒント：この場合、資源割り当てグラフに要請枝を追加したグラフを作成し、要求された資源を割り当てると循環ができるかを検査すれば良い）

- A. R1 空き, R2 空きの状態で P2 が R2 を要求した。
- B. R1 空き, R1 は P1 に割当ての状態で P1 が R2 を要求した。
- C. R1 空き, R2 は P2 に割当ての状態で P1 が R1 を要求した。
- D. R1 空き, R2 は P2 に割当ての状態で P2 が R1 を要求した。

問 7

デッドロックが発生するのは、以下のどれか。

- A. プロセス P1 が資源 r1 と資源 r2 を確保した後に、プロセス P2 が資源 r1 と資源 r2 を要求した。
- B. プロセス P1 が資源 r2 を確保し、プロセス P2 が資源 r1 を確保した後に、プロセス P2 が資源 r2 を要求し、プロセス P1 が資源 r1 を要求した。
- C. プロセス P1 が資源 r2 を確保し、プロセス P2 が資源 r1 を確保した後に、プロセス P2 が資源 r2 を要求し、プロセス P1 が資源 r3 を要求した。
- D. プロセス P1 が資源 r2 を確保し、プロセス P2 が資源 r1 を確保した後に、プロセス P3 が資源 r1 と資源 r2 を要求した。

問 8

資源型の実体の数が全て 1 の場合、デッドロックの検出は、比較的簡単な方法で実行できる。その方法は以下のどれか。

- A. 待ち合せグラフに循環があるかどうかを定期的に調べる。
- B. プロセスが要求した資源を割当てると、待ち合せグラフに循環ができるかどうかを調べる。
- C. プロセスが資源を要求してきた時、資源割当てグラフに循環があるかを調べる。
- D. プロセスが要求した資源を割当てると、資源割当てグラフに循環ができるかどうかを調べる。

問 9

スライドは、プロセスのプログラムと初期状態の資源割り当てグラフを示している。

(1)(5)(9)(2)(10)(6)の順で処理が実行された場合、資源割当てグラフに存在する枝を以下から全て選べ。

- A. 要求枝 ($P1 \rightarrow r1$)
- B. 要求枝 ($P1 \rightarrow r2$)
- C. 要求枝 ($P2 \rightarrow r1$)
- D. 要求枝 ($P2 \rightarrow r2$)
- E. 要求枝 ($P3 \rightarrow r2$)
- F. 要求枝 ($P3 \rightarrow r3$)
- G. 割当て枝 ($r1 \rightarrow P1$)
- H. 割当て枝 ($r2 \rightarrow P1$)
- I. 割当て枝 ($r1 \rightarrow P2$)
- J. 割当て枝 ($r2 \rightarrow P2$)
- K. 割当て枝 ($r2 \rightarrow P3$)
- L. 割当て枝 ($r3 \rightarrow P3$)
- M. 枝は存在しない

問 10

前問において、終了させても、デッドロックの回復に無関係なプロセスはどれか。(ヒント：終了させられたプロセスは、資源割り当てグラフから消え、その要求枝、割り当て枝も消える。)

- A. デッドロックは発生していない。
- B. P1
- C. P2
- D. P3
- E. どれを終了させても、デッドロックは解消される。