⑥デッドロック(2)

2015年度

問1　デッドロックの防止

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. 資源型に対し，昇順に番号を付け，番号順に要求するようにプログラミングさせることで，循環待機の条件が成立しないようにする．
B. 他のプロセスが使用中であっても，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. 資源が割当てられるまで待機中のプロセスから，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. 資源が割当てられるまで待機中のプロセスから，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．

Bは，実現不能．C，Dは実現が困難．
Aがもっとも簡単なため，「防止」はこの方法を採用．但し，プログラムが自由に作れない，資源の利用効率が低下するという問題点がある．

問2　デッドロックの防止

スライド[添付ファイル]表1のように資源r1，r2，r3を排他的に占有して処理を行うプロセスP1～P4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P1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．（ソフトウェア開発　平成14年度）

A. プロセスP2
B. プロセスP3
C. プロセスP4
D. プロセスP2，P3
E. プロセスP2，P4
F. プロセスP3，P4
G. プロセスP2，P3，P4

プロセス名	資源r1	資源r2	資源r3
プロセスP1	1	2	3
プロセスP2	3	2	1
プロセスP3	1	2	3
プロセスP4	2	3	1

デッドロックの防止は，循環待機が発生しないように，資源の要求順序を決め，それを守るようにプログラムを作成するものである（資源を要求する順序が同じプロセス同士では，循環待機が発生しない．逆に，順序が違えば，循環待機が発生する可能性がある）．例えば，資源r1,r2,r3の順に要求するようにP2，P4のプログラムを変更することで，P1とのデッドロックが防止できる．
（資源の確保順は，昇順である必要は無く，「防止」の方法は1通りではない）．

スライド（問2の添付ファイル）

問2　表1　プロセスが資源を要求する順序

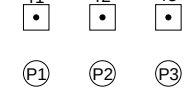
プロセス名	資源r1	資源r2	資源r3
プロセスP1	1	2	3
プロセスP2	3	2	1
プロセスP3	1	2	3
プロセスP4	2	3	1

問9，10　図1　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:
(1) r1要求
(2) r2要求
・・・・
(3) r1解放
(4) r2解放

P2:
(5) r2要求
(6) r1要求
・・・・
(7) r2解放
(8) r1解放

P3:
(9) r3要求
(10) r2要求
・・・・
(11) r3解放
(12) r2解放

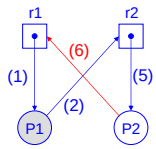


処理の条件　資源型の実体は1プロセスのみが確保できる
（複数のプロセスが同時に確保できない）．
OSは要求された資源が空いていれば，無条件に割当てる．

問3　デッドロックの発生

今，以下の動作を行うプロセスP1，P2が並行して動作している．
プロセスP1:(1)r1要求→(2)r2要求→(3)r2解放→(4)r1解放
プロセスP2:(5)r2要求→(6)r1要求→(7)r1解放→(8)r2解放
資源r1，r2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？（要求資源が未使用であれば，割り当てるものとする）

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



C.の場合，P1は資源待ちで待機状態．実行可能なP2は，この後(6)でr1を要求するようにプログラムされているため，資源割り当てグラフに循環ができる．
（他の選択肢では循環ができない）
B，Dの場合は，資源待ちで待機状態のプロセスがあるが，実行可能なプロセスのプログラムは，資源を解放するようになっており，デッドロックとはならない．
A，Eは一方のプロセスのみが実行しており，デッドロックにはなりえない．

問4　デッドロックの回避

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．

デッドロックの回避では，割当て時点では，デッドロックではなくても，以後，デッドロック発生の可能性があれば資源を割当てない．従って，Bの判定は誤り．
（割当てても，安全な順序が存在するかどうかを銀行家ｱﾙｺﾞﾘｽﾞﾑで判定する．
安全な順序が無い場合は，将来，ﾃﾞｯﾄﾞﾛｯｸが発生する可能性があるため，
資源を割当てない．）

問5　デッドロックの回避

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．

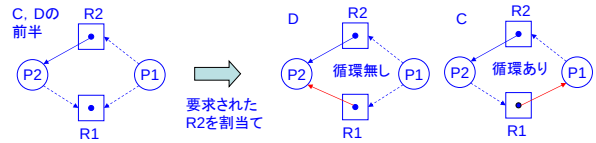
A. 資源を要求する度ではなく，定期的に銀行家アルゴリズムを実行する．
B. 資源割り当てグラフの資源型の実体の数を1個にし，割当てると循環ができるかどうかをチェックする．
C. 資源割当てグラフの資源型を省略した待ち合せグラフを作成し，割当てると循環ができるかどうかチェックする．
D. 資源割り当てグラフに要請枝を追加し，割当てると循環ができるかどうかをチェックする．

次のスライドを参照

問6　デッドロックの回避

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．

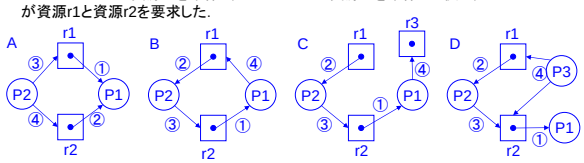


BもDと同様に，循環はできない．Aは，r1，r2共に空きなので循環はできない．

問7　デッドロックの検出

デッドロックが発生するのは，以下のどれか．（第1種　平成11年度改）

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．



循環が発生

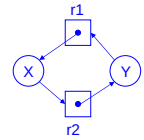
Cの④の要求枝は，すぐに割当て枝になる

問8　デッドロックの検出

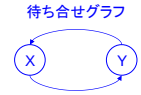
資源型の実体の数が全て1の場合，デッドロックの検出は，比較的簡単な方法で実行できる．その方法は以下のどれか．

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．

資源割り当てグラフ



待ち合せグラフ



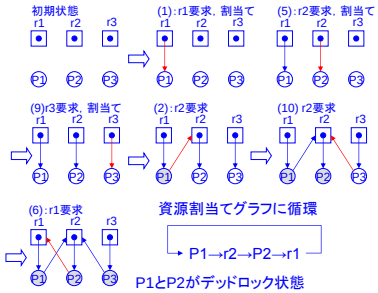
待ち合わせグラフの方が循環のチェックが簡単である

デッドロックの検出では，資源の要求時にはチェックをせず，定期的に循環をチェックする．資源の要求の頻度に比べて，チェックの回数が極めて少なくできるので，処理負荷が少ない．

問9　資源割り当てグラフ

スライド【問2の添付ファイル】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→r1）
B. 要求枝（P1→r2）
C. 要求枝（P2→r1）
D. 要求枝（P2→r2）
E. 要求枝（P3→r2）
F. 要求枝（P3→r3）
G. 割当て枝（r1→P1）
H. 割当て枝（r2→P1）
I. 割当て枝（r1→P2）
J. 割当て枝（r2→P2）
K. 割当て枝（r2→P3）
L. 割当て枝（r3→P3）
M. 枝は存在しない



資源割当てグラフに循環

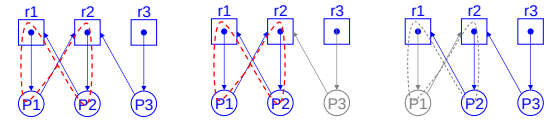
P1→r2→P2→r1

P1とP2がデッドロック状態
そのため，P3も動けなくなっている．

問10　デッドロックの回復

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．



P1とP3で循環

P3を消しても循環が残る

P1かP2を消すと循環が消える