

第9回の課題について

デッドロックが起きないようにするにはどのような点に気をつければよいか、p.10の例にもとづいて説明せよ

起きないようにする、すなわち「防止」の観点から、以下のような点が考えられます。

- ◆ 複数の資源を必要とする処理では、一度にまとめて資源を確保する（「確保・待ち」の回避）
講義資料 p.10では、資源 A と B をプロセス 1 と 2 がそれぞれ取り合っていました。たとえば、プロセス 1 が A と B を一度に確保してしまえば、プロセス 2 が B を確保することはなく、デッドロックは起きません
- ◆ 複数の資源を必要とするときは、確保する順序を同一方向にする（「循環待ち」の回避）
講義資料 p.10では、プロセス 1 が A→B の順、プロセス 2 が B→A の順で取りにいったので、一方を確保した後に他方を待つことでデッドロックになっています。どちらも A→B の順で確保すれば、A を確保できなかったプロセス（この場合はプロセス 2）が B を確保することはなく、デッドロックになりません
- ◆ 使用済みの資源は直ちに解放する
これは、発生しないことを保証するものではありませんが、常に気をつけるべきことで、デッドロックの発生確率を下げることができます

などです。

デッドロックは、以下の 4 条件が同時に成り立つと起こり得ます。

- （１）相互排除 … プロセスは必要とする資源を相互排除する
- （２）確保・待ち … プロセスは必要とする資源を待っている間、ある資源をすでに確保している
- （３）横取り不可 … 確保している資源は自ら解放するまで横取りされない
- （４）循環待ち … 資源の待ちが循環している

デッドロックの防止について、上記の条件の中で相互排除に関しては任意の資源について発生しないようにすることは不可能ですので、残りの三つについて成立しないようにするものが課題の解答になります（「横取り不可」の回避は必ずしも可能なものではありません）。

また、デッドロックが発生する条件があらかじめ分かっているならば、決してデッドロック状態に陥らないことを保証するアルゴリズムを実現することができます。しかし、将来の資源要求についての情報が必要であり、これは基本的に困難です。

【提出いただいた解答へのコメント】

『デッドロックの必要条件を(いずれかが成り立たないようにするなど)回避する』ことが基本ですが、それによって、たとえば講義スライドで発生したデッドロックが起こらないことまで説明されるとよいのではと思います。

資源の確保を順序付けすることを示された解答が多く、講義の範囲ではそれが妥当だと思います。さらに、A と B をまとめて確保することなども含め、複数の方法を提示された方もいらっしゃいます。上記で説明した観点に近いものを考察いただいたと思います。

デッドロックの検出に関して検討なさった方もいらっしゃいました。資源の割り当てをグラフで表して、そこにサイクルがあるか調べるなどの検出アルゴリズムがあるようです。

デッドロックが発生しているときに、何らかの方法でそれを解除するという策もあるかもしれませんが、実際にデッドロックであるかどうかを判定することが難しいのではと考えられます。また、あるプロセスの共有資源を処理の途中で強制的に解放するとした場合、それで計算の整合性(共有資源の状態)が保証されるのかという問題もありそうです。

参考（令和2年度 秋期 応用情報技術者試験問題）

三つの資源 X ～ Z を占有して処理を行う四つのプロセス A ～ D がある。各プロセスは処理の進行に伴い、表中の数値の順に資源を占有し、実行終了時に三つの資源を一括して解放する。プロセス A と同時にもう一つのプロセスを動かした場合に、デッドロックを起こす可能性があるプロセスはどれか。

プロセス	資源の占有順序		
	資源 X	資源 Y	資源 Z
A	1	2	3
B	1	2	3
C	2	3	1
D	3	2	1

選択肢

ア B、C、D

イ C、D

ウ C だけ

エ D だけ

A と資源の確保順序が逆になっているもの (Z→X や Y→X など) は、C と D