デッドロックの対策

復習： デッドロックとは

- デッドロックとは
  待ちが引き起こす「待合い」(３すくみ)のこと

復習： デッドロックの (必要十分) 条件

（教科書） ４つの条件

- 相互排他（排除)があって
- 資源確保のとき、取れなければ待つ方式で
  （取れなければすぐあきらめて、今確保した
  資源も全部解放する方式ではなく）
- 資源の横取り不可で
  （現在の占有者から資源を横取りできない）
- 循環待ちであるとき
  （待ちがお互いにぐるっと循環している）

復習： 資源割付グラフでの循環

- 循環待ちは
  資源割付グラフを描くとわかる
  （資源割付グラフ上で循環しているとＮＧ）
- 資源割付グラフは、
  プロセス（楕円）と資源（四角）をノードとし
  ２種類のエッジを書く
  実線矢印＝プロセスが資源を要求している
  破線矢印＝資源がプロセスに割当済みである
  この上でサイクルが存在するとＮＧ

では、デッドロック対策です

東邦大学

デッドロック対策

- 対策には、３つの考え方

1.デッドロックが起きてしまってから、
　　起きたことを見つけて、何とか(回復)する

2.デッドロックにおちいる最後の資源待ち操作時に
　　見つけて、その操作をやめる
　　（これはそううまくいかない）

3.デッドロックが起きる条件が成立しないように、
　　あらかじめプログラムを工夫する

東邦大学

デッドロック対策１

- デッドロックが起きてしまってから、
  起きたことを見つけて、何とか(回復)する

デッドロック対策１

- デッドロックが起きてしまってから、
  起きたことを見つけて、何とか(回復)する
- 見つける方法：
  資源割付グラフを描いて、循環を検出

デッドロック対策１

- デッドロックが起きてしまってから、
  起きたことを見つけて、何とか(回復)する
- 見つける方法：
  資源割付グラフを描いて、循環を検出
- 回復の方法：
  検出の時点では３すくみを解くことはできない
  （既に一部の資源は確保して、操作を始めて
  しまっているから）
  誰かの処理をあきらめて、確保前まで戻す
  （データベースではロールバックと呼ぶ）

デッドロック対策２

- デッドロックに陥る前の最後の資源待ち操作時に
  見つけて、その操作をやめる
  ⇒　これはそううまくいかない

デッドロック対策２

- デッドロックに陥る前の最後の資源待ち操作時に
  見つけて、その操作をやめる
  ⇒　これはそううまくいかない
- 見つける方法：　資源割付グラフでできる

デッドロック対策２

- デッドロックに陥る前の最後の資源待ち操作時に
  見つけて、その操作をやめる
  ⇒　これはそううまくいかない
- 見つける方法：　資源割付グラフでできる
- 回避の方法：
  グラフ上で循環になる待ち操作を見つけた時点では、既に別の資源を確保して操作を始めている。この処理をあきらめて確保前まで戻す必要がある

デッドロック対策２

- デッドロックに陥る前の最後の資源待ち操作時に
  見つけて、その操作をやめる
  ⇒　これはそううまくいかない
- 見つける方法：　資源割付グラフでできる
- 回避の方法：
  グラフ上で循環になる待ち操作を見つけた時点では、既に別の資源を確保して操作を始めている。この処理をあきらめて確保前まで戻す必要がある
  ⇒
  結局、起きてしまってからの回復と同じになる

デッドロック対策３

- デッドロックが起きる条件が成立しないように、
  あらかじめプログラムを工夫する
- いくつかの方法がある

デッドロック対策３

- デッドロックが起きる条件が成立しないように、
  あらかじめプログラムを工夫する
- いくつかの方法がある

1. 利用する資源を１つのロック単位にする
2. 資源に順番を付け、必ずその順でロック/解放する

デッドロック対策３

- デッドロックが起きる条件が成立しないように、
  あらかじめプログラムを工夫する
- いくつかの方法がある

1. 利用する資源を１つのロック単位にする

複数資源を使うのだが、それをまとめて一括でロックしてしまう。

⇒　一方は確保し他方が取れず待つ、ということがない

デッドロック対策３

- デッドロックが起きる条件が成立しないように、
  あらかじめプログラムを工夫する
- いくつかの方法がある

1. 利用する資源を１つのロック単位にする

   複数資源を使うのだが、それをまとめて一括でロックしてしまう。

   ⇒　一方は確保し他方が取れず待つ、ということがない

   欠点：１プロセスしか動けない時間が長くなる

デッドロック対策３

- デッドロックが起きる条件が成立しないように、
  あらかじめプログラムを工夫する
- いくつかの方法がある

1. 利用する資源を１つのロック単位にする
2. 資源に順番を付け、必ずその順でロック/解放する

   逆順が無いので、循環ができない

デッドロック対策３

- デッドロックが起きる条件が成立しないように、
  あらかじめプログラムを工夫する
- いくつかの方法がある

1. 利用する資源を１つのロック単位にする
2. 資源に順番を付け、必ずその順でロック/解放する

逆順が無いので、循環ができない
欠点：その順番でプログラムするのは難しい

デッドロック対策３

- デッドロックが起きる条件が成立しないように、
  あらかじめプログラムを工夫する
- いくつかの方法がある

1. 利用する資源を１つのロック単位にする
2. 資源に順番を付け、必ずその順でロック/解放する

どちらも、使える場面が限られていて
あまり現実的ではない

デッドロックのまとめ

- デッドロックとは
- デッドロックの起こる（必要十分）条件は
- 資源割付グラフで、循環待ちがあるかがわかる
- デッドロックの対策には
  - 発生してから戻す・発生しそうなものを戻す
  - 最初から起こらないようにプログラムする

デッドロックのまとめ

- デッドロックとは
- デッドロックの起こる（必要十分）条件は
- 資源割付グラフで、循環待ちがあるかがわかる
- デッドロックの対策には
  - 発生してから戻す・発生しそうなものを戻す
  - 最初から起こらないようにプログラムする

対策はそれぞれの場合によって選択される

デッドロックが
理解できましたか？

○

次へ

×