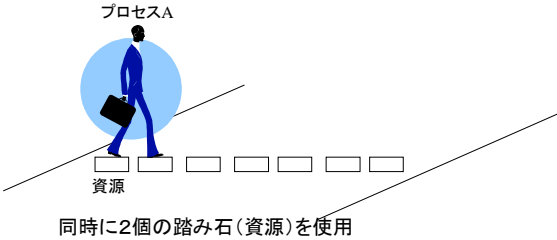
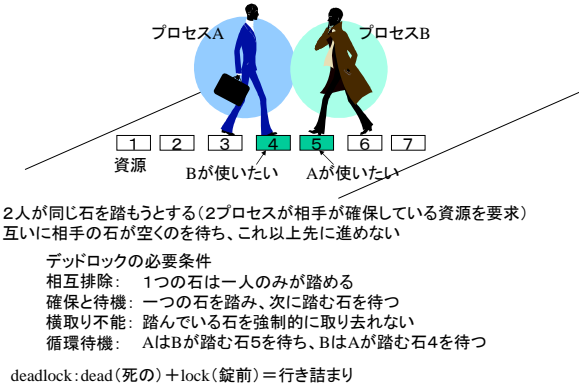


第6回 デッドロック(1)

川の横断

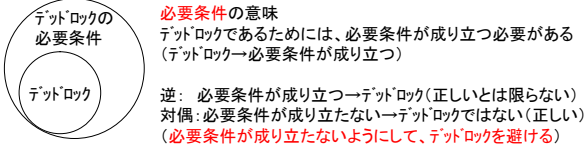


川の横断でのデッドロック

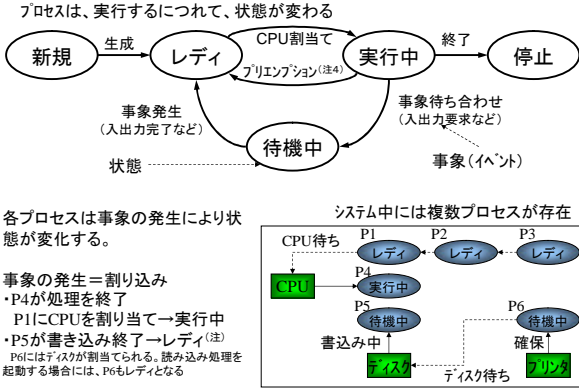


重要:デッドロックの必要条件

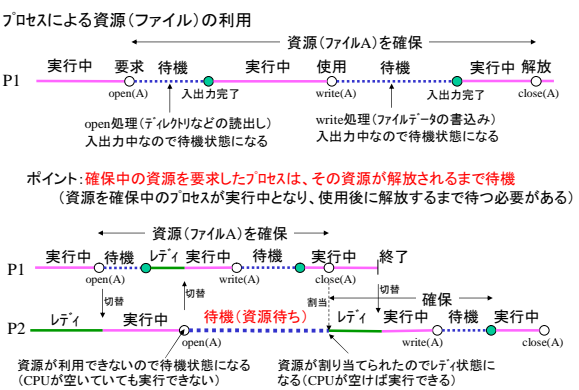
相互排除	1度に1プロセスのみが確保した資源を利用できる
確保と待機	プロセスが少なくとも1つの資源を確保し、他のプロセスが確保した資源を更に確保しようと待つ
横取り不能	資源は、それを確保しているプロセスによってのみ解放される
循環待機	確保と待機におけるプロセス間の関係に循環性がある(確保と待機の条件を含む)



プロセスの状態遷移

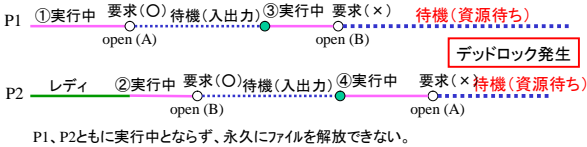


資源の確保と解放

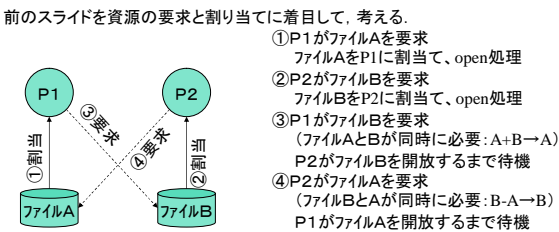


デッドロック発生例

P1:ファイルA内のデータ + ファイルB内のデータ→ファイルAに格納
P2:ファイルB内のデータ + ファイルA内のデータ→ファイルBに格納
・P1、P2ともにファイルAとファイルBを同時かつ排他的(他のプロセスから)に使用。
①P1が実行中(P2はレディ)。P1がファイルAを要求。OSは空いているので割当てる。
P1はopen(入出力処理)のため待機状態になる。OSはP2にCPUを割当てる。
②P2が実行中、ファイルBを要求。OSは空いているので割当てる。
P2はopen処理のため待機状態になる。
③P1のopen処理が完了。CPUが空いているため実行中になり、ファイルBを要求。
OSはファイルBが使用中なので、P1を待機状態にする(CPUを取りあげる)
④P2のopen処理が完了。CPUが空いているため実行中になり、ファイルAを要求。
OSはファイルAが使用中なので、P2を待機状態にする(CPUを取りあげる)

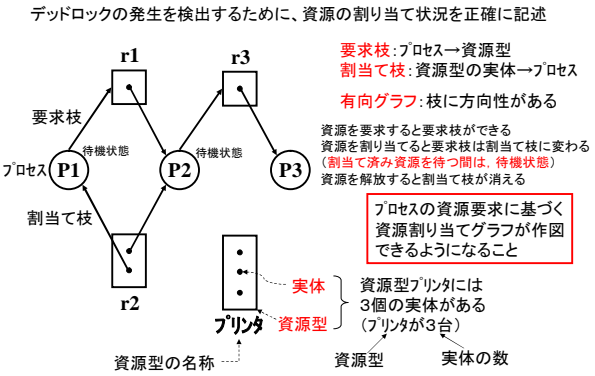


資源の割り当てとデッドロック

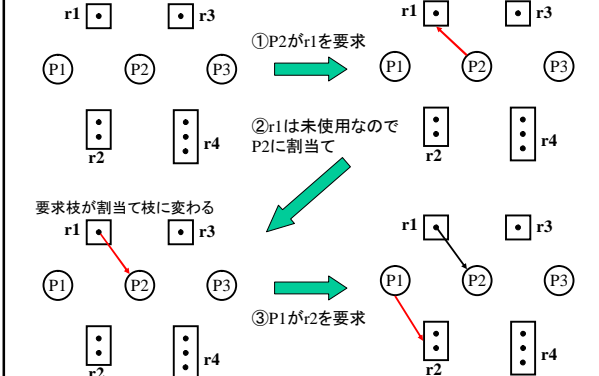


デッドロックの定義
集合内のどのプロセスも、集合内の他プロセスによってのみ引き起こされる事象(注1)を待つ(注2)
注1:システムコールを発行して、確保した資源(ファイル)を解放する事象。
注2:しかし、待機状態のプロセスは実行(システムコールの発行)ができない。
→P1、P2は永久に処理が終わらない。しかも、資源を確保したままなので、システムに悪影響をおよぼす(他のプロセスもファイルA、Bを使えない)。

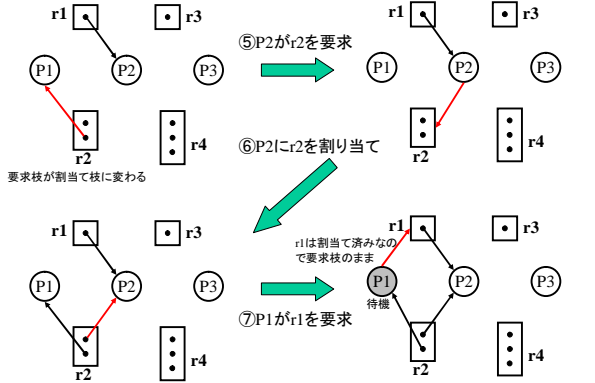
重要:資源割り当てグラフ



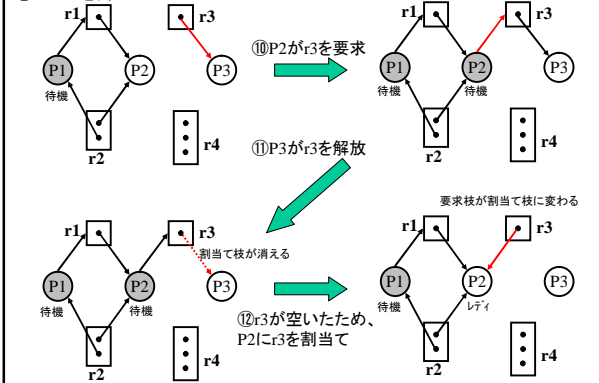
資源割り当てグラフ



資源割り当てグラフ



資源割り当てグラフ



資源割り当てグラフ(デッドロック)

P3がr2を要求
循環が発生
P1→r1→P2→r3→P3→r2→P1
P2→r3→P3→r2→P2

必要条件
デッドロック→循環あり
(循環待機)

循環なし→デッドロックでない
循環あり→デッドロックの可能性あり

矢印がぶつかっているものは、循環ではない
(P1→r1→P2→r2→P1は循環ではない)

循環有(デッドロックではない場合)

① P2がr2を要求

② P3がr2を解放

③ r2に空きができたP2にr2を割当て

矢印がぶつかっているものは、循環ではない

資源型の実体が全て1個

P1→r1→P2→r2→P3→r3→P1

循環があればデッドロックである

必要条件
デッドロック→循環待機

資源型の実体が1個の場合
逆: 循環待機→デッドロックも成り立つ

P1はr1が空かないと処理が終わらない(r3を解放できない)
P2はr2が空かないと処理が終わらない(r1を解放できない)
P3はr3が空かないと処理が終わらない(r2を解放できない)

デッドロックの必要条件(補足)

相互排除	1度に1プロセスのみが確保した資源を利用できる
確保と待機	プロセスが少なくとも1つの資源を確保し、他のプロセスが確保した資源を更に確保しようと待つ
横取り不能	資源は、それを確保しているプロセスによってのみ解放される
循環待機	確保と待機におけるプロセス間の関係に循環性がある(確保と待機の条件を含む)

相互排除の例
P1: 資源Xを要求(確保)→P2: 資源Xを要求(待機)
(資源Xを利用できるのはP1のみ。P1が資源解放するまで、P2は待たされる)

確保と待機の例
P1: 資源Xを要求(確保)→P2: 資源Yを要求(確保)→P2: 資源Xを要求(待機)
P2は資源Xを確保したまま資源Xを待つ

循環待機の例(実体の数が1個の場合、デッドロック)
P1: 資源Xを要求(確保)→P2: 資源Yを要求(確保)→P3: 資源Zを要求(確保)→P1: 資源Zを要求(待機)→P2: 資源Xを要求(待機)→P3: 資源Yを要求(待機)

デッドロック発生と解決法

デッドロックの**必要条件**が成立しないようにする
(O): (システムが固まったので、)利用者がマニュアルで再立ち上げ
全プロセスの処理を打ち切り、システムが資源を取り上げる(横取り不能の否定)

①**防止**: 資源の使用順序を統一してプログラムを作る(ファイルA→ファイルBの順で要求)
1方向なので、要求の循環が発生し得ない(循環待機の否定)

②**回避**: P1がファイルA、Bを使い終わるまで、P2の要求を待たせる
割り当て時に、循環待機が起こりえるかどうかを検査(循環待機の否定)

③**検出と回復**: システムの状態を検査し、デッドロックとなったP2をロールバックする
デッドロックとなったプロセスから資源を取り上げる(横取り不能の否定)