

ハイレベルの排他・同期機構



ハイレベルの待ち

- 状態遷移を使った待ちは、
（ハイレベル）プログラミング上の概念として
提供されている
- 例：セマフォ
モニター
その他、多数提案されている

⇒ セマフォのみを紹介する



セマフォ

- 基本となるものであり
- 歴史的にも有名
- 最近どこまで使われるかは不明



東邦大



セマフォ

- 考え方
バイナリセマフォ ~ ロックとほぼ同じ
計数セマフォ
~ こちらが基礎で、
バイナリは計数を0/1だけにしたもの
- E. Dijkstra (オランダ人の高名な先生) が発案
⇒ 考え方が高尚 & ことばが耳慣れない、が
要するにロックの拡張のようなもの



東邦大



セマフォ

- 共通変数 s (セマフォ変数) = カウンター と
- 2つの操作 P操作 (ロック) とV操作 (解放)



東邦大



セマフォ

- 共通変数 s (セマフォ変数) = カウンター と
- 2つの操作 P操作 (ロック) とV操作 (解放)
- P操作 : (ロック操作に相当)
 - $s > 0$ なら、 $s = s - 1$ (資源は確保できた)
 - $s \leq 0$ なら、事象待ちで s が正になるのを待つ



東邦大



セマフォ

- 共通変数 s (セマフォ変数) = カウンター と
- 2つの操作 P 操作 (ロック) と V 操作 (解放)
- P 操作 : (ロック操作に相当)
 $s > 0$ なら、 $s = s - 1$ (資源は確保できた)
 $s \leq 0$ なら、事象待ちで s が正になるのを待つ
- V 操作 : (ロック解放操作に相当)
待っているプロセスがあれば、待ちを解除する
無ければ、 $s = s + 1$



東邦大



セマフォ

- セマフォ変数 s の初期値は正の数とする
仮に3としよう
この後の動作を追いかけてみよう



東邦大



セマフォ

- セマフォ変数 s の初期値は正の数とする
仮に3としよう
- プロセスAがP操作 \Rightarrow Aは資源確保、 $s=2$



東邦



セマフォ

- セマフォ変数 s の初期値は正の数とする
仮に3としよう
- プロセスAがP操作 \Rightarrow Aは資源確保、 $s=2$
- プロセスBがP操作 \Rightarrow Bは資源確保、 $s=1$
- プロセスCがP操作 \Rightarrow Cは資源確保、 $s=0$



東邦



セマフォ

- セマフォ変数 s の初期値は正の数とする
仮に3としよう
- プロセスAがP操作 \Rightarrow Aは資源確保、 $s=2$
プロセスBがP操作 \Rightarrow Bは資源確保、 $s=1$
プロセスCがP操作 \Rightarrow Cは資源確保、 $s=0$
プロセスDがP操作 $\Rightarrow s \leq 0$ なのでDは待たされる



東邦



セマフォ

- セマフォ変数 s の初期値は正の数とする
仮に3としよう
- プロセスAがP操作 \Rightarrow Aは資源確保、 $s=2$
プロセスBがP操作 \Rightarrow Bは資源確保、 $s=1$
プロセスCがP操作 \Rightarrow Cは資源確保、 $s=0$
プロセスDがP操作 $\Rightarrow s \leq 0$ なのでDは待たされる
- その後、BがV操作(解放) \Rightarrow 待っているDを起動



東邦



セマフォ

- セマフォ変数 s の初期値は正の数とする
仮に **3** としよう
- プロセスAがP操作 \Rightarrow Aは資源確保、 $s=2$
プロセスBがP操作 \Rightarrow Bは資源確保、 $s=1$
プロセスCがP操作 \Rightarrow Cは資源確保、 $s=0$
プロセスDがP操作 $\Rightarrow s \leq 0$ なのでDは待たされる
- その後、BがV操作(解放) \Rightarrow 待っているDを起動
- つまり、最大 **3** 人が使える (**3** 人しか使えない)



東邦大



セマフォ

- セマフォ変数 s の初期値は正の数とする
仮に **3** としよう
- プロセスAがP操作 \Rightarrow Aは資源確保、 $s=2$
プロセスBがP操作 \Rightarrow Bは資源確保、 $s=1$
プロセスCがP操作 \Rightarrow Cは資源確保、 $s=0$
プロセスDがP操作 $\Rightarrow s \leq 0$ なのでDは待たされる
- その後、BがV操作(解放) \Rightarrow 待っているDを起動
- つまり、最大 **3** 人が使える (**3** 人しか使えない)
 s の初期値で決まる



東邦大



バイナリセマフォ

- ではバイナリセマフォは？
s の値が 0か1



東邦大



バイナリセマフォ

- ではバイナリセマフォは？
s の値が 0か1 初期値を1にすると
最大1人が使える（1人しか使えない）
つまり、ロックと同じ効果になる



東邦大



セマフォの脱線

- セマフォ (Semaphore) の意味は？
腕木信号や手旗信号などを意味するらしい
腕木によって通行を許可したり禁じたりする
イメージがあるのだろうか
- 操作の名前が P と V の理由？
英語なら Signal/Wait とか、Lock/Unlock では？
⇒ Dijkstra 先生はオランダ語を使ったらしい
(ウィキペディアによると更に諸説ある)



東邦大



セマフォの見方・評価

- ハイレベルのプログラミングに対応
細かい実装を隠し、機能をユーザに使わせる
- その割には、機構は単純で分かりやすい
- 実際の高級言語には
もう少し「高級」な機構が使われることが多い



東邦大



セマフォのまとめ

- セマフォ変数 s と、PとVの操作から成る
- s の初期値(>0)が最大同時利用者数になる
 s 人まで同時に利用できる、超えると待たせる
- P操作 = ロック操作に相当
 s を1つ減らす。もし $s \leq 0$ なら待つ。
- V操作 = 解放操作に相当
もし待っている人がいれば待ちを解除。
 s を1つ増やす
- s 人同時利用 $\Rightarrow s$ を0/1にすれば単なるロック



セマフォの原理が
理解できましたか？



↓
次へ

