

高度OS2011③並行プロセス(3)

③並行プロセス(3)

高度OS2013年度

添付ファイル(問1~10)

セマフォの定義

P(S):  
Sの値を1減らす;  
S≥0→nop;  
S<0→発行元プロセスを待機中に  
戻る;

V(S):  
Sの値を1増やす;  
S>0→nop;  
S≤0→待機中プロセスをレディに;  
戻る;

P1:

(1) P(S)  
(2) Xの内容をレジスタに読み込む。  
(3) レジスタに3を加える。  
(4) レジスタの内容をXに格納する。  
(5) V(S)

生産者:(以下を繰り返す)  
(1)ディスクreadし, nextpに入力;  
(2)P(empty);  
(3)buffer[in]=nextp;  
(4)in=(in+1) % n;  
(5)V(full);

P2:

(6) P(S)  
(7) Xの内容をレジスタに読み込む。  
(8) レジスタを1減じる  
(9) レジスタの内容をXに格納する。  
(10)V(S)

消費者:(以下を繰り返す)  
(6)P(full);  
(7)nextc = buffer[out];  
(8)out=(out+1) % n;  
(9)V(empty);  
(10)nextcのデータをプリンタにwrite;

高度OS2011③並行プロセス(3)

問1 セマフォ変数の使用方法

スライド<sup>【問1の添付ファイル】</sup>の上部に示したセマフォP(S), V(S)を用いて、排他制御を行う。一般的に、セマフォ変数の初期値は以下のどれにすれば良いか。

☒ A. 使用可能な共有資源の数  
B. システム全体のプロセス数  
C. 待機状態のプロセス数  
D. 使用している共有資源の数

セマフォは資源の数(残数)を管理するために用いる。  
初期値では、1つも使用していないので、使用可能な共有資源数になる。  
尚、B~Dは、処理中に変化する値であり、初期値には、なりえない

高度OS2011③並行プロセス(3)

問2 セマフォ変数の初期値

スライド<sup>【問1の添付ファイル】</sup>の左下部に示した2つのプロセスP1, P2の排他制御におけるセマフォ変数Sの初期値は以下のどれか。

A. 0  
☒ B. 1  
C. 2  
D. 3  
E. 10

セマフォ変数の初期値は使用可能な共有資源数である。  
この問題の場合、共有資源は変数Xであり、その数は1である。

高度OS2011③並行プロセス(3)

問3 プロセスの状態遷移

スライド<sup>【問1の添付ファイル】</sup>P1, P2に対してラウンドロビンスケジューリングを行う。P2, P1の生成が完了し、P2が先にCPUを割り当てられた。P2が(6)を実行した後のP1, P2の状態はそれぞれ何か(状態名:新規, レディ, 実行中, 待機, 停止)。

【P1, P2の順に空席のコンパ】、「で(実行中)」、状態名(遷移または空席のコンパ)を順番に記入。途中に空席などを入れると誤りとなる。例: 新規, レディ

答 レディ, 実行中

発生した事象または実行した命令	P1	P2	S
P1, P2生成	レディ	レディ	1
P2にCPU割当て	レディ	実行中	1
P2が(6)を実行	レディ	実行中	0

P2が(6)を実行したので、S=0となるが、S≥0なので、そのままP2に戻る。  
従って状態は変わらずに、レディ, 実行中

高度OS2011③並行プロセス(3)

問4 プロセスの状態遷移

前問に続いて、P2が(7)(8)を実行した後、量子時間経過の割り込みが発生した。OSの割り込み処理とスケジューリングが完了したあと、P1, P2の状態は何になるか。

【P1, P2の順に空席のコンパ】、「で(実行中)」、状態名(遷移または空席のコンパ)を順番に記入。途中に空席などを入れると誤りとなる。例: 新規, レディ

答 実行中, レディ

発生した事象または実行した命令	P1	P2	S
P1, P2生成	レディ	レディ	1
P2にCPU割当て	レディ	実行中	1
(6)を実行 (続いて(7), (8)を実行)	レディ	実行中	0
量子時間経過,		レディ	0
CPU割当て	実行中		

複数事象が続いて起こった場合、行数削減のため、下記のように記述することができる

量子時間経過, CPU割当て	実行中	レディ	0
----------------	-----	-----	---

P1, P2の状態が、実行中, レディにおいて、量子時間が経過したので、  
実行中のP2は、プリエンプション(CPU取り上げ)により、レディ状態となる。  
CPUが空いたので、レディ状態のP1は、CPUが割り当てられ、実行中となる。

問5 プロセスの状態遷移

前問に続いて、P1が(1)を実行した。OSの割り込み処理とスケジューリングが完了したあと、P1、P2の状態は何になるか。【P1、P2の順に全角のコンマ、1で既切り、状態名(漢字または全角のカタカナ)を解答欄に記入。途中に空白などを入れたら誤りとなる。例: 待機、レディ】

答 待機、実行中

P2がクリティカルセクション((7)~(9))の実行途中のため、Sの値が0になっている。P1が(1)を実行したため、S=-1となった(事象待合わせ)。OSはP1を待機状態にする。CPUが空いたので、OSはレディ状態のP2にCPUを割り当て実行中にする。

発生した事象または実行した命令	P1	P2	S
P1、P2生成	レディ	レディ	1
P2にCPU割当て	レディ	実行中	1
(6)を実行 (続いて(7)(8)を実行)	レディ	実行中	0
量子時間経過、CPU割当て	実行中	レディ	0
P1が(1)を実行(事象待合わせ)、CPU割当て	待機	実行中	-1

問5以降の処理

発生した事象または実行した命令	P1	P2	S
P1、P2生成	レディ	レディ	1
P2にCPU割当て	レディ	実行中	1
P2が(6)を実行 (続いて(7)(8)を実行)	レディ	実行中	0
量子時間経過、CPU割当て	実行中	レディ	0
P1が(1)を実行(事象待合わせ)、CPU割当て	待機	実行中	-1
P2が(9)を実行	待機	実行中	-1
P2が(10)を実行(事象発生)	レディ	実行中	0
P2が終了、CPU割当て	実行中	停止	0
P1が(2)~(5)を実行	—	実行中	1

問5までの処理

P2が再開し、(10)を実行するとセマフォ値が0になる(OSはP1をレディ状態にする)。P2が終了し、CPUが空いたので、OSはP1を実行中にする。P1が再開して(2)~(5)を実行し、最後の(5)により、セマフォ値Sが初期値の1に戻る。以上により、クリティカルセクションの実行が入り乱れないので、計算結果は正常になる。

問6 セマフォ変数の初期値

スライド【問1の回答ファイル】の右下部に記述した生産者、消費者の排他制御におけるセマフォ変数empty、fullの初期値は幾つか、但し、buffer[]の大きさは2で、開始時は空きとする。【empty、fullの順に整数値を全角のコンマ、1で既切り、半角数字で解答欄に記入】

答 2、0

セマフォ変数の初期値は使用可能な共有資源数である。この問題の共有資源はバッファ(buffer[])である。生産者にとっての使用可能な資源は空きバッファなので初期値empty=2。消費者にとっての使用可能な資源はデータが入力されたバッファなので初期値full=0。

問7 セマフォによる排他制御

スライド【回答ファイル】の右下部に記述した生産者、消費者の生成が完了し、生産者にCPUが割り当てられた。生産者は(1)を実行するために、ディスクreadのシステムコールを発行した。OSは消費者を実行中にし、消費者は(6)を実行した。OSのスケジューリングが終わった時点でのプロセスの状態は何になるか。【生産者、消費者の順に状態名を全角のコンマ、1で既切り、解答欄に記入】

答 待機、待機

発生した事象または実行した命令	生産者	消費者	empty	full
生産者、消費者生成	レディ	レディ	2	0
生産者にCPU割当て	実行中	レディ	2	0
(1)を実行(read要求)、CPU割当て	待機	実行中	2	0
(6)を実行(事象待ち合わせ)	待機	待機	2	-1

(1)のディスクread(入出力要求)により、OSは生産者を待機状態にする。CPUが空いたので、OSは、消費者にCPUを割り当てる。

消費者の(6)実行によりfullが1減少(0→-1)。セマフォ変数full<0なのでOSは発行元の消費者を待機状態にする)

問8 セマフォによる排他制御

前問に続いて、生産者のreadが完了して生産者が実行中になり、(2)~(5)を実行した。(5)に対するOSの処理が終わった後、生産者、消費者の状態とセマフォ変数の値はどのようになっているか。【生産者、消費者、empty、fullの順に状態名と整数値を全角のコンマ、1で既切り、解答欄に記入。状態名は漢字または全角のカタカナ、整数値は半角数字。】

答 実行中、レディ、1、0

発生した事象または実行した命令	生産者	消費者	empty	full
生産者、消費者生成	レディ	レディ	2	0
生産者にCPU割り当て	実行中	レディ	2	0
(1)を実行(read要求)、CPU割当て(消費者)	待機	実行中	2	0
(6)を実行(事象待ち合わせ)	待機	待機	2	-1
read完了、CPU割り当て(生産者)	実行中	待機	2	-1
(2)を実行(emptyが1減少)、続いて(3)(4)を実行	実行中	待機	1	-1
(5)を実行(事象発生)	実行中	レディ	1	0

生産者による(5)V(full)の実行により、full(セマフォ変数)の値が1増加(-1→0)。full=0となったためOSは待機状態の消費者プロセスをレディ状態にする

問9 プロセスの状態遷移

前問に続いて、生産者は(1)を実行し、さらに、消費者は(7)~(10)を実行した。生産者、消費者の状態とセマフォ変数の値はどのようになっているか。【生産者、消費者、empty、fullの順に状態名と整数値を全角のコンマ、1で既切り、解答欄に記入。状態名は漢字または全角のカタカナ、整数値は半角数字。】

答 待機、待機、2、0

発生した事象または実行した命令	生産者	消費者	empty	full
生産者、消費者生成	レディ	レディ	2	0
生産者にCPU割り当て	実行中	レディ	2	0
(1)を実行(read要求)、CPU割当て	待機	実行中	2	0
(6)を実行(事象待ち合わせ)	待機	待機	2	-1
read完了。(2)~(5)を実行(事象発生)	実行中	レディ	1	0
(1)を実行(read要求)、CPU割当て(消費者)	待機	実行中	1	0
(7)~(10)を実行(write要求)	待機	待機	2	0

生産者は(1)のread要求を実行し、待機状態になる。消費者は前問の処理でレディ状態となっているため、CPUを与えられて実行中となり、(7)~(10)を実行する。(9)でemptyが1→2となり、(10)のwrite要求により、待機状態になる。

注: 前回の授業では、消費者の生成は、生産者の処理が1回終わった後として説明した。実際には、本問のように両方とも同時に生成されるので、初期の動作は複雑になる。

問10 セマフォによる排他制御

前問に続いて、(A)生産者、(B)生産者、(C)生産者、(D)消費者、(E)生産者の順で実行中状態となった。(C)における生産者の実行中状態が終わった段階で、生産者、消費者の状態とセマフォ変数の値はどのようになっているか。【生産者: 消費者: empty: fullの値に状態名と実行値を全角のコンマ「,」で区切り、縦書きに記入。状態名は漢字または全角カタカナ。実行値は半角数字。】

答 待機, 待機, -1, 2

発生した事象または実行した命令	生産者	消費者	empty	full
問8の処理が終わった時点	待機	待機	2	0
(A) read完了, CPU割り当て(生産者)	実行中	待機	2	0
(A) (2)~(5) を実行	実行中	待機	1	1
(A) (1)を実行(read要求)	待機	待機	1	1
(B) read完了, CPU割り当て(生産者)	実行中	待機	1	1
(B) (2)~(5) を実行	実行中	待機	0	2
(B) (1)を実行(read要求)	待機	待機	0	2
(C) read完了, CPU割り当て(生産者)	実行中	待機	0	2
(C) (2)を実行(事象待ち合わせ)	待機	待機	-1	2

生産者は(A)、(B)では、(1)のディスクreadにより実行中から待機状態となる(実行中状態が終わる)。一方(C)では、(2)P(empty)の実行によりemptyが0→-1になり、生産者は待機状態となる。

問10以降の処理

前問に続いて、(D)、(E)によって状態のセマフォ変数は以下のように変化する。

発生した事象または実行した命令	生産者	消費者	empty	full
問10の実行が終わった時点	待機	待機	-1	2
(D) write完了, CPU割り当て(消費者)	待機	実行中	-1	2
(D) (6)~(8)を実行	待機	実行中	-1	1
(D) (9)を実行(事象発生)	レディ	実行中	0	1
(D) (10)を実行(write要求)	レディ	待機	0	1
(E) CPU割り当て(生産者)	実行中	待機		
(E) (2)~(5)を実行	実行中	待機	0	2
(E) (1)を実行(read要求)	待機	待機	0	2