②並行プロセス(2)

高度OS2015年度

問1　軽量プロセス

プロセスには，プログラムの実行に必要なシステムの資源が割当られる．では，CPU以外の資源は割り当てられず，親のプロセスから必要な資源を継承し，軽量プロセスとも呼ばれるものは，以下のどれか．（第1種　平成8年度改）

A. タスク
B. ジョブステップ
C. コンテキスト
D. スレッド
E. カーネル

プロセスの生成では，アドレス空間の生成などの処理が必要なため負荷が大きい．
一方，スレッドは，親プロセスのアドレス空間を使用するため，生成の負荷が小さい．
処理負荷が軽いために軽量プロセスとも呼ばれる．

問2　資源

前問の問題文で用いられた「資源」を意味する語として，最も適切なものは以下のどれか．

A. プログラム
B. CPU
C. メモリ
D. 計算に使用するもの
E. 地球環境から生まれるもの

資源には，ハードウェア資源とソフトウェア資源があり，コンピュータは，これらを複数使用して計算を実行する．

スライド(問3の添付ファイル)

問3，問4

P1:
:
(1) Xの内容をレジスタに読み込む．
(2) レジスタに2を加える．
(3) レジスタの内容をXに格納する．
:

P2:
:
(4) Xの内容をレジスタに読み込む．
(5) レジスタを1減じる
(6) レジスタの内容をXに格納する．
:

問9，問10

```
lock (A):
  A=OFF→nop;
  A=ON→発行元プロセスを待機状態に;
A=ON;
戻る;

unlock (A):
  待機プロセス無し→nop;
  待機プロセスあり→待機プロセスをレディ状態に;
A=OFF;
戻る;
```


問3　処理の競合(1)

変数Xを共有してスライド【添付ファイル】の処理を行う2つのプロセスP1，P2がある．

Xの初期値が2の時，P1が先に実行中となり，(1)を実行したところで，プリエンプションが発生し，P2が実行中となった．P2が(4)～(6)を実行して終了した後，P1が再開して残りの処理を行った．Xの値は幾つか．【数値のみを半角数字で解答欄に記入．】

答　4

次の次のスライドを参照

参考　プリエンプション無しの場合

Xの初期値＝2　(1)(2)(3)(4)(5)(6)の順に処理

順序	P1	ﾚｼﾞｽﾀ	X	P2
			2	
1	(1)Xをﾚｼﾞｽﾀに	2	〃	
2	(2)ﾚｼﾞｽﾀを＋2	4	〃	
3	(3)ﾚｼﾞｽﾀをXに	〃	4	
	終了		〃	開始
4		4	〃	(4)Xをﾚｼﾞｽﾀに
5		3	〃	(5)ﾚｼﾞｽﾀを－1
6		〃	3	(6)ﾚｼﾞｽﾀをXに

〃は値が変化しないことを表す．
レジスタ値の空欄は，OSの処理に使用され，種々の値をとる．

問3の処理による値の変化

Xの初期値＝2，(1)(4)(5)(6)(2)(3)の順に処理される

順序	P1	ﾚｼﾞｽﾀ	Xの値	P2
			2	
1	(1)Xをﾚｼﾞｽﾀに	2	〃	
	ﾌﾟﾘｴﾝﾌﾟｼｮﾝ(ﾚﾃﾞｨ) ﾚｼﾞｽﾀ退避「2」･[切替]		〃	実行中 [開始]
2		2	〃	(4)Xをﾚｼﾞｽﾀに
3		1	〃	(5)ﾚｼﾞｽﾀを－1
4		〃	1	(6)ﾚｼﾞｽﾀをXに
	[再開]		〃	[終了]
	ﾚｼﾞｽﾀ復元	2	〃	
5	(2)ﾚｼﾞｽﾀを＋2	4	〃	
6	(3)ﾚｼﾞｽﾀをXに	〃	4	

問4　処理の競合（2）

変数Xを共有してスライド[問3の添付ファイル]の処理を行う2つのプロセスP1，P2がある．

Xの初期値が2の時，P1が先に実行中となり，(1)(2)を実行したところで，プリエンプションが発生し，P2が実行中となった．P2が(4)(5)を実行したところで，ページフォールトが発生し，P1が再開して実行中となった．P1が残りを実行して終了後，P2が再開して残りの処理を行った．Xの値は幾つか．【数値のみ6半角数字で解答欄に記入．】

答　1

次のスライドを参照

問4の処理による値の変化

Xの初期値＝2，(1)(2)(4)(5)(3)(6)の順に処理される

順序	P1	ﾚｼﾞｽﾀ	Xの値	P2
			2	
1	(1)Xをﾚｼﾞｽﾀに	2	〃	
2	(2)ﾚｼﾞｽﾀを＋2	4	〃	
	ﾌﾟﾘｴﾝﾌﾟｼｮﾝ(ﾚﾃﾞｨ) ﾚｼﾞｽﾀ退避「4」･[切替]		〃	実行中 [開始]
3		2	〃	(4)Xをﾚｼﾞｽﾀに
4		1	〃	(5)ﾚｼﾞｽﾀを－1
	[再開]		1	ページフォールト レジスタ退避「1」･[切替]
	ﾚｼﾞｽﾀ復元	4	〃	
5	(3)ﾚｼﾞｽﾀをXに	4	4	
	[終了]		〃	[再開]
		1	〃	ﾚｼﾞｽﾀ復元
6			1	(6)ﾚｼﾞｽﾀをXに

問5　アクセス競合の防止

共有変数に対し，アクセスの競合が起こると不正な結果となる．これを防止するために，前問の(1)～(3)と(4)～(6)は実行が入り乱れないようにする必要がある．プログラムの中のこのような部分を指す用語を選択肢から選べ．【回答欄に記入．カタカナは全角文字．】

【選択肢】
排他制御，マルチスレッド，マルチプログラミング，
クリティカルセクション，デッドロック

答　クリティカルセクション

クリティカルセクションは，共有資源の状態を変更する処理を含む．
そのため，排他制御が必要である．

問6　アクセス競合の防止

共有変数のアクセス競合を防止するために，アクセス中表示フラグを設け，入口区域で，フラグをチェックし，オフの場合は，フラグをオンにして実行し，出口区域でフラグをオフにする方法が用いられる．しかし，このままでは無駄なチェックを繰り返してCPUを浪費する(　　)という問題が発生する．
(　　)内に最適な用語を選択肢から選べ．【回答欄に記入．カタカナは全角文字．】

【選択肢】
繁忙待機，クリティカルセクション，
マシンチェック割り込み，システムコール

答　繁忙待機

CPUを占有してフラグをチェックし続けても，オンのままなので無駄である．
（相手のプロセスが実行されなければフラグはオフにならない）
忙しい思いをして，実質的に待機中と同じ効果しかないため，繁忙待機と言う．

問7　アクセス競合の防止

前問の問題点を防止するためには，以下のどの方法を用いればよいか．

A. フラグのチェック処理を関数呼び出しにする．
B. フラグの設定処理をOSのみが実行するようにする．
C. フラグをチェックする前に割込みを禁止にする．
D. フラグをチェックしてオンであれば，プロセスを待機状態にする．

フラグがオンである場合は，他のプロセスがクリティカルセクションを実行していた．
（その途中で，何らかの原因で処理を中断している）
この場合，これ以上チェック処理を繰り返しても無駄である．

このため，以下の処理を行う．
①チェックしたプロセスを待機状態にして，実行を停止させる．（繁忙待機の防止）
②フラグをオンにしたプロセスが実行中になる．
③クリティカルセクションを再開．それが終了し，フラグをオフにする．
④待機状態のプロセスをレディ状態にする．
⑤CPUが割り当てられると，フラグがオフになっており，ｸﾘﾃｨｶﾙｾｸｼｮﾝを実行する

問8　アクセス競合の防止

共有変数のアクセス競合防止の基本的な方法では，アクセス中表示フラグを設け，入り口区域で，フラグをチェックする．オフの場合は，フラグをオンにして実行し，出口区域でフラグをオフにする．但し，このフラグ自体が共有変数であり，その操作を排他的に実行する必要がある．これを実現する方法は，以下のどれか．

A. フラグのチェック処理を関数呼び出しにする．
B. フラグの操作をOSのみが実行するようにする．
C. フラグをチェックする前に割込みを禁止にする．
D. フラグをチェックしてオンであれば，プロセスを待機状態にする．

複数のプロセスによるフラグの操作が競合した場合に，問題が発生する．
これを防止するためには，フラグの操作（チェックと設定）処理をOS（1つのプロセス）のみが実行できるようにすれば良い．尚，OSは，フラグの設定処理よりも優先度が高い重要な処理を色々と行っており，フラグの操作が中断する可能性がある．この影響を排除するために，フラグのチェックと設定を1命令で実行する命令を設けたり，フラグの操作中のみ，割り込みを禁止するなどを行う．

問9　プロセスの状態遷移

システム内にラウンドロビンでスケジューリングされる4つのプロセスがあり，以下の状態であった．
P1：待機，P2：待機，P3：実行中，P4：レディ，　lock変数A=OFF
また，P3とP4は，共有資源があり，スライド【問3の添付ファイル】に示すlock，unlockを用いて排他制御を行っている．P3が発行したシステムコールlock(A)の完了直後に，量子時間が経過した．このあとの各プロセスの状態として適当なものはどれか．

A. P1：待機，P2：待機，P3：終了，P4：実行中
B. P1：待機，P2：待機，P3：レディ，P4：実行中
C. P1：実行中，P2：待機，P3：終了，P4：レディ
D. P1：待機，P2：待機，P3：実行中，P4：待機
E. P1：レディ，P2：待機，P3：実行中，P4：待機

レディ状態のプロセスは実行待ち列に並んで，CPUが空くのを待つ．ラウンドロビンスケジューリングでは，量子時間が経過すると，実行中のプロセスからCPUを取り上げ（プリエンプション），レディ状態にし，実行待ち行列の最後尾（P4の後ろ）に並べる．次に，実行待ち列の先頭のプロセス（P4）にCPUを割当て，実行中にする．

問10　プロセスの状態遷移

前問の状態遷移が行われた後，P4がシステムコールlock(A)を発行した．
この後の各プロセスの状態として，適当なものは以下のどれか．

A. P1：待機，P2：待機，P3：終了，P4：実行中
B. P1：待機，P2：待機，P3：レディ，P4：実行中
C. P1：実行中，P2：待機，P3：終了，P4：レディ
D. P1：待機，P2：待機，P3：実行中，P4：待機
E. P1：レディ，P2：待機，P3：実行中，P4：待機

前問により，P4が実行状態になっている．P4は，クリティカルセクションを実行するためにlockを発行した．しかし，P3が先にlock変数Aをオンにしている．そのためOSは，P4を待機状態にする．また，CPUが空くので，P3を実行中にする．
（P1，P2の状態は変わらない）
このあとの処理は以下のようになる．
P3がクリティカルセクションのを実行し，unlockシステムコールを発行．
OSはP4をレディ状態にする．
P3の終了または量子時間経過によりP4が実行中になる．
OSはlock命令からリターンし，P4が再開．P4はｸﾘﾃｨｶﾙｾｸｼｮﾝを実行．