

## 問010046解説

### ◆解答

設問1 ウ

設問2 イ

設問3 a キ b カ c ウ

### ◆解説

OSにおけるプロセスのスケジューリングに関する問題である。

### 3 状態の遷移

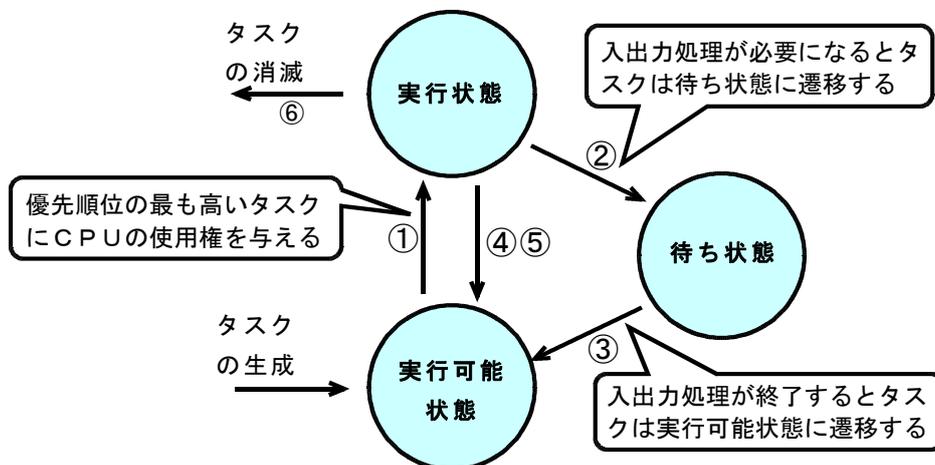
- ① 実行可能状態のタスクのうち、最も優先順位の高いタスクを選択する。

選択されたタスクにCPUの制御権が与えられ、実行状態に移される。ディスパッチャは、TCBの待ち行列を調べ、優先順位の高いタスクを実行可能状態タスクの中から選択するプログラムであり、この処理機能を実行することをディスパッチングという。

- ② 実行状態のタスクで入出力オペレーションが必要になると、そのタスクは要求した入出力動作が終了するまで待ち状態となる。

- ③ 入出力オペレーションが終了すると、タスクは実行可能状態となる。

入出力オペレーションが終了すると、終了を知らせる信号が入出力装置からCPUに送られる。この信号を受け取ってCPUは実行中のタスクを中断し、入出力待ちのタスクを実行可能状態にする。この機能は入出力割込によって行われる。



- ④ 実行中のタスクよりも優先順位の高いタスクが実行可能状態になった場合、実行状態のタスクは実行を中断し、実行可能状態に移り、その後、優先順位の高いタスクにCPUの制御権を渡す。

- ⑤ タイムシェアリングシステムでは、接続されている端末に数十ミリ秒単位毎にCPU時間を配分し、この時間を経過する毎に実行中のタスクを実行可能状態に戻す。このTSSの機能はタイマ割込機能と呼ばれる。

- ⑥ タスクの実行が完了すると、TCBはTCB待ち行列から削除される。この時、該当タ

スクが使用していたシステム資源は全て解放される。

## スケジューリング

複数のタスクを効率的に切り換えてシステム資源を有効に利用し、スループットやレスポンスの向上を実現するためにスケジューリングが用いられる。適正なスケジューリングを行うためには、トリガと優先順位が重要になる。トリガにはイベントドリブン方式、タイムスライス方式、両者の組合せの方式がある。

## イベントドリブン方式

システムの状態に変化が生じたときCPUを別のプロセスに切り替える方式である。切替のイベントには、入出力の完了、入出力要求の発生、新しいプロセスの到着、プロセスの完了等がある。イベントドリブン方式はマルチプログラミングの基本概念でCPUの利用効率を向上させる。

## タイムスライス(時分割)方式

システムの状態変化と無関係に周期的にプロセスを切り替える方式で、各プロセスに割り当てられたタイムスライス(一定時間)を利用して行う。一般的にはタイムスライスの大きさは数十ミリ秒から数百ミリ秒の範囲を用いる。

## プリエンプションとノンプリエンプション

### ① プリエンプション

ある処理中に他の処理要求が発生すると、発生した処理要求を実行するために、現在実行中の処理を一時中断する。この中断処理をプリエンプションという。優先度の高い処理が発生すると、優先度の低い処理を中断し、処理を切り替える。優先度の高い処理が完了すると、中断されていた優先度の低い処理を再開する。処理を切り替えることをスケジューリング処理といい、これによって実行順序を制御する。この実行順序の制御にプリエンプションが利用される。

### ② ノンプリエンプション

ノンプリエンプションは、一台のコンピュータで同時に複数の処理をこなすマルチタスク処理の実現方法のうち、OSがCPUを管理しない方式である。実行中の各アプリケーションソフトが、自分が処理を行わない空き時間を自発的に開放することによって、他のアプリケーションソフトと同時実行できるようにする方式である。

## タスク実行制御方式

タスク実行制御方式に、プリエンプション方式とノンプリエンプション方式がある。

プリエンプション方式は、実行中のタスクが中断されて、優先度の高いタスクに切り替わり、優先度の高いタスクの処理が終了すると、中断されたタスクの処理を再開する方式である。

ノンプリエンプション方式は、実行中のタスクが終了、または自発的にCPUを解放して、

新しいタスクの処理が開始される方式である。この方式では中断が発生しない。

### 優先順位的方式

#### ① 到着順方式（FCFS）

実行可能状態になった順に実行権を切り替える。実行を開始すると自ら待ち状態になるか完了するまで実行を続け、他のタスクを実行状態にすることができないノンプリエンプティブ方式である。

#### ② 処理時間順方式（SPT）

処理時間の短いプロセスから実行する方式で、対話型処理の平均応答時間を最小にする。新しいタスクが実行可能状態になるたびに、各タスクの残り時間を評価し、その残り時間の短いものから順に実行する。

プリエンプティブ方式と組み合わせて、新しいプロセスの優先権を高くする方法で利用される。

#### ③ 優先度順方式

TCBに設定している実行優先順位に従って実行を制御する。実行優先順位方式は、イベントの発生に伴って、実行中のタスクと実行可能状態になったタスクの優先度が比較されて、後者の優先度が高いと実行中のタスクは実行を中断する。実行を中断されたタスクは実行可能状態に遷移し、優先度の高いタスクが実行状態になる。

優先度比較のためのイベントが発生するのは、タスクが待ち状態から実行可能状態に遷移したときや新しくタスクが生成されたときになる。優先度の低いプロセスは順番が回ってこないスタベーションの状態になり、エイジングにより優先度を高める解決策が必要になる。

#### ④ ラウンドロビン方式

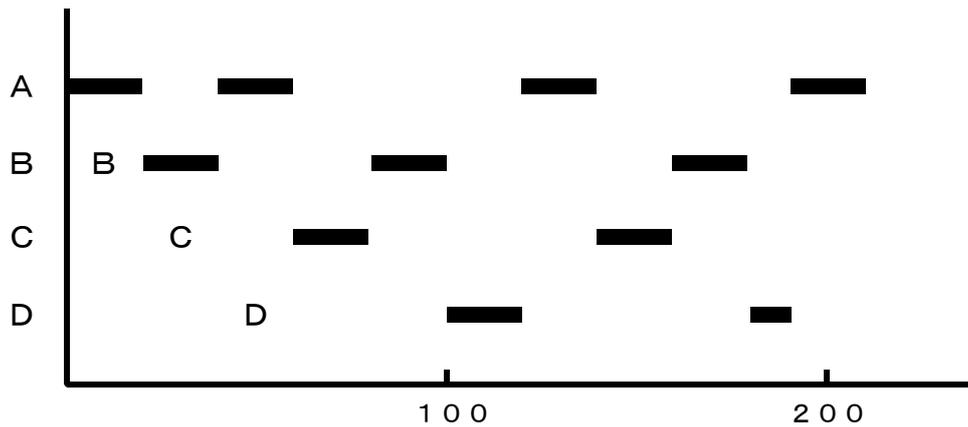
一定時間毎にプロセスの実行を切り替える方式で、プリエンプティブ方式のスケジュールになる。決められた時間内に処理が終了しないか、または待ち状態に遷移しない場合、タイマー割込により実行中のプロセスを中断し、実行可能状態の待ち行列の最後尾に回し、その待ち行列の先頭のプロセスにCPUの使用権を与える。この時の一定時間をタイムスライスまたはタイムクォンタムという。

この方式は公平な応答時間を保証し、タイムシェアリングシステムに利用される。プライオリティのあるラウンドロビンスケジュールリングでは、同一のプライオリティ内ではラウンドロビン方式を用いる。

### ラウンドロビン方式のスケジューリング

設問2の表1のプロセスの到着時刻、処理時間、およびタイムクォンタム20ミリ秒で実行した場合のスケジュールおよび待ち行列の状態を示すと次の図のようになる。

待ち行列のプロセスのうち次に実行されるのは、待ち行列の先頭のプロセスであり、実行状態から中断して待ち行列に入るプロセスは待ち行列の最後になる。プロセスDが最初に実行状態になった時の待ち行列の状態は、ACBのプロセスの並びとなる。



待ち行列

B	A	C	B	D	A	C	B	D	A
	C	B	D	A	C	B	D	A	C
		D	A	C	B	D	A	C	B

### ターンアラウンドタイムの計算

残余処理時間順方式による4プロセスのターンアラウンドタイムを計算すると、次の表のようになる。

A	120	110	110	110	110	110	110	110	110	110	110	110	110	110	110
B		90	80	70	70	70	70	70	70	70	70	70	70	60	50
C				60	50	40	40	40	40	30	20	10	0		
D						30	20	10	0						
A	110	110	110	110	110	100	90	80	70	60	50	40	30	20	10
B	40	30	20	10	0										
C															
D															

プロセスAのターンアラウンドタイムは300ミリ秒、プロセスBのターンアラウンドタイムは180ミリ秒、プロセスCのターンアラウンドタイムは90ミリ秒となる。

### 設問1

①～④のプロセスに関するa～dの遷移の回数を調べると次のようになる。

	(a)	(b)	(c)	(d)
① 前処理	○			
② データの入力			○	○
③ 計算処理	○			
④ データの出力			○	○

⑤ 後処理                    ○  
回 数                    3        0        2        2

遷移の組み合わせは、3 0 2 2となり、求める答えはウとなる。

### 設問2

解説欄のラウンドロビン方式のスケジューリングに示したスケジュール図の待ち行列の中の長方形で囲んだ内容が答えとなる。答えはA C Bとなり、求める答えはイとなる。

### 設問3

解説欄のターンアラウンドタイムの計算の箇所に、残余処理時間順方式による4プロセスの残余処理時間の推移を示した。1セル10ミリ秒として、A、B、Cの各プロセスのターンアラウンドタイムを求めると、プロセスAは300ミリ秒、プロセスBは180ミリ秒、プロセスCは90ミリ秒となる。aの求める答えはキ、bの求める答えはカ、cの求める答えはウとなる。