

【解答】

[設問1] オ

[設問2] a-イ, b-イ, c-キ

【解説】

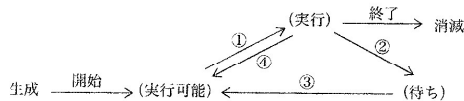
プロセス（又はタスク）のスケジューリングに関する問題である。スケジューリングに関する知識問題は午前試験での出題頻度が高い。ラウンドロビン方式や状態遷移図は基本情報技術者試験でも必須の内容なので、理解しておく必要がある。平成26年秋に同様の問題が出題されているので参考にするとよい。上位の応用情報技術者試験では定番の内容であり、今後も出題される可能性が高い内容である。

本問題ではラウンドロビン方式と優先度順方式を取り上げている。内容は示されているので、その手順に則ってタイムチャートなどを作成し、解決していけばよい。

設問2の優先度順方式では、プロセスの優先度が変化し、どのプロセスにCPUの実行権が割り当てられるかを正確に把握する必要がある。全てのプロセスが終了（消滅）するまで吟味すると時間を要するし、ミスも発生しやすいが、求められている内容はプロセスAの処理1が終了するまでの時刻となっているので、正確な追跡ができれば解答できる。一つ一つ丁寧に確認、検証していくことが求められる。

[設問1]

図3のプロセスの状態遷移について図Aで確認しておく。



図A プロセスの状態遷移

生成されたプロセスは、生成と同時に実行可能状態になり、開始の矢印に沿って遷移する。実行可能状態とは、いつでもCPUの実行権の割当てが可能な状態である。

次に、実行可能状態にある幾つかのプロセスの中から一つを取り出し、CPUの実行権を割り当て、①に沿って実行状態に遷移する。このとき、複数ある実行可能状態のプロセスのどれにCPUの実行権を割り当てるかは、スケジューリング方式によって決まる。

実行状態のプロセスに入出力が発生した場合は②に沿って待ち状態に遷移する。入出力が終了すると、③に沿って再び実行可能状態になる。

ここで、④の遷移は実行状態のプロセスを強制的に実行可能状態に戻す操作であり、これをプリエンプション（横取り）という。実行状態のプロセスは、タイムク

ウォンタムの経過によってプリエンプションされる。プリエンプションはラウンドロビン方式だけではなく、優先度順方式ではより高い優先度のプロセスが実行可能状態になったときに発生する。

実行が終了したプロセスは、終了の矢印に沿って消滅する。

プロセスXの処理時間及び待ち時間は表1に示されているが、タイムクウォンタムが20ミリ秒なので、処理1、処理2の処理時間を分割（実行を中断）し、図Aの①～④の遷移を当てはめてみると、表Aのようなになる。

表A プロセスXの分割と遷移

プロセス名 分割 遷移	処理時間及び待ち時間				
	処理1	入出力 待ち1	処理2	入出力 待ち2	処理3
X	30		50	30	10
分割	20	10	20	20	10
遷移	①④①	②③	①④①④①	②③	①

タイムクウォンタムが20ミリ秒なので、処理1は30=20+10に分割される。処理2は50=20+20+10に分割される。処理時間が終了するまでは、①と④及び①の遷移となる。また、入出力待ち1及び2では、②、③の遷移となる。処理3はタイムクウォンタム未満の処理時間なので、①から消滅へと遷移することになる。

このため、①は6回、②は2回、③は2回、④は3回なので、正解は（オ）である。解き方としては、表Aのような表を作るほかに、タイムチャートを作成し、処理1から時間の経過に沿って、①～④を書き込んで解いてもよいだろう。選択肢の内容から、ポイントは①と④の回数なので、特に注意する。

[設問2]

優先度順方式に関する内容である。優先度を正確に把握しながら解く必要があり、設問数が少ないので、ミスをするとな設問2で得点できないことになる。

・空欄a：プロセスXの処理をタイムクウォンタムで分割し、設問文の(1)～(3)に沿って優先度の推移を確認していくと表Bのようなになる。

- (1) プロセスが生成された場合、優先度3を与える。
- (2) タイムクウォンタムが経過して、実行を中断した場合、優先度が1のときは優先度1、優先度が2～5のときは、1段階下げた優先度を与える。
- (3) 入出力が完了した場合、優先度5を与える。

表B プロセスXの優先度の推移

プロセス名 分割 優先度の推移	処理時間及び待ち時間				
	処理1	入出力 待ち1	処理2	入出力 待ち2	処理3
X	30	30	50	30	10
分割	20	10	20	20	10
優先度の推移	3	2	5	4	3

したがって、優先度の推移は、「3→2→5→4→3→5」の（イ）である。

・空欄b, c：プロセスがA～Cの三つあり、複雑な内容になるが、求められているのは最初の流れだけなので、タイムチャートを作成しながら正確にプロセスの動きを把握する。

プロセスAの処理1の開始から終了する時間までを追跡する。プロセスA～Cの処理1に着目すると、タイムクウォンタムが20ミリ秒であるため、プロセスA, Bの分割時間は20, 10となる。このことを踏まえて、生成時刻に沿って吟味すると、図Bのようなになる。

- (1) プロセスAが生成されると実行可能状態になり、優先度3のキューに登録されると同時にCPUの実行権が与えられ実行される。
- (2) プロセスAが実行されてから10ミリ秒後にプロセスBが生成され、実行可能状態になり、優先度3のキューに登録されるが、CPUは同じ優先度のプロセスAが実行中のため実行可能状態となる。
- (3) プロセスAの処理1（20ミリ秒）が終了すると、優先度は3から2になり、キューに登録される。一方、実行可能状態であるプロセスBの優先度は3で高いため、プロセスBにCPUの実行権が与えられ実行される。
したがって、空欄bの解答は「20」ミリ秒後の（イ）である。
- (4) プロセスAが生成されてから30ミリ秒後にプロセスCが生成され、優先度3のキューに登録されるが、同じ優先度のプロセスBが実行中のため実行可能状態になる。
- (5) プロセスBの処理1（20ミリ秒）が終了すると、優先度は3から2になる。この時点で、優先度2のキューには、プロセスA, Bの順に登録される。一方、待ち状態であるプロセスCの優先度は3で高いため、プロセスCにCPUの実行権が与えられ実行される。
- (6) プロセスCの処理1（20ミリ秒）が終了したとき、優先度は3から2になるが、優先度2のキューにはプロセスA, Bが登録されており、最後尾に登録される。先頭はプロセスA（先入れ先出し）なので、処理1の残りの10ミリ秒がCPUで実行され、プロセスAの処理1の30ミリ秒全てが終了することになる。
したがって、空欄cの正解は、「70」ミリ秒後の（キ）である。

経過時刻 (ミリ秒)	0	10	20	30	40	50	60	70
CPU割当て		A		B		C		A
プロセスA 処理時間		↑ 処理1 20						↑ 処理1 10
優先度の推移		生成 3 → 2						2 → 1
プロセスB 処理時間			↑ 処理1 20					
優先度の推移			生成 3 → 2					
プロセスC 処理時間				↑ 処理1 20				
優先度の推移				生成 3 → 2				

注記1：↑は各プロセスの生成時刻を示す。

注記2：処理時間はタイムクウォンタムで分割している。

注記3：→はプロセスが生成されてから処理が開始されるまで実行可能状態で待つ。

注記4：優先度の推移は「処理の開始時→処理の終了時」を示す。

図B プロセスAの処理1の開始から終了まで

【参考】

この問題では、コンピュータのCPUは一つとするとあるが、現在ではスマートフォンなどでも8個程度のCPU（マルチコア）が一般的になっている。この場合、プロセスは同時に最大8個が実行状態となると考えれば同じようなスケジューリング方式を適用することができる。また、設問2の優先度順方式は、現実の多くのOSで採用されているスケジューリング方式のひな型であり、動的にCPUを多く使うプロセスの優先度を低くし、入出力が多いプロセスの優先度を高くすることによって、全体のスループットを高めることができる。