

- 【解答】
- 【設問 1】                    ヲ
- 【設問 2】                    aーウ, bーエ
- 【設問 3】                    エ

【解説】

プロセスの排他制御に関する問題である。ほぼ同内容の問題が平成 24 年秋に出題されている。この問題を解いていた受験者は高得点できたであろう。排他制御は、データベースシステムで取り上げることが多いが、午前試験を含め、出題頻度の高い重要項目の一つである。二重更新の回避のための排他制御の必要性や、排他制御を行った場合に生じるデッドロックなどは理解して試験に臨む必要がある。

本問は、平成 24 年秋の出題に「セマフォ変数」が加わった内容になっている。基情報技術者の午後の試験では初めて出題されたが、どちらかと言うと上級試験の範疇であり、知らなかった受験者もいるかもしれない。しかし、ロック法は試験対策上、必須の内容である。

セマフォ変数 S の値として、0 及び 1 を割り当て、0 のときは共有データがいずれかのプロセスに確保されているロック状態、1 のときは解放されているアンロック状態となっている。このように、他のプロセス (プログラム) からのアクセスを制御し、二重更新などによるデータの不具合 (正しい値とならない) を回避する仕組みであると理解できればよい。セマフォでは、ロック時を P 操作、アンロック時の操作を V 操作として実現している。

設問内容は難しくないが、設問数が少ないため、一つの解答に対する点数が高い。ケアレスミスに注意する必要がある。

【設問 1】

排他制御の必要性を考える設問である。二重更新による不具合の内容が理解できればよい。

共有データ Y に対して排他制御を行わずに、並行して各 1 回の更新処理を行う。

実行前の共有データ Y の値が 100

- 【更新処理 1】：共有データ Y の値を 30 増加させる。……プロセス p1
- 【更新処理 2】：共有データ Y の値を 50 減少させる。……プロセス P2

プロセス p1 の更新処理 1 の終了後に、プロセス p2 の更新処理 2 を行えば、共有データ Y の値は、100+30-50=80 となる。プロセス p2, p1 の順序で実行しても、100-50+30=80 となり、同じ結果になる。

共有データを更新する複数のプロセスが同時並行的に動くとき、処理のタイミソングによっては正しい値とならないことがある。図 A で示すように、実行順序を①～⑥とした場合、最終的な共有データ Y の値は⑥の 50 になってしまい、正しい結果が得られない。

また、⑥のプロセス P2 の書出しの後に、⑤のプロセス p1 の書出しを行った場合には、最終的な共有データ Y の値は⑤の 130 になってしまうことも想定できる。これを二重更新による不具合という。

以上から、共有データ Y が取り得る値は、正しい値の 80、及び誤った値の 50, 130 である。取り得ない値は「100」であり、(ウ) が正解である。

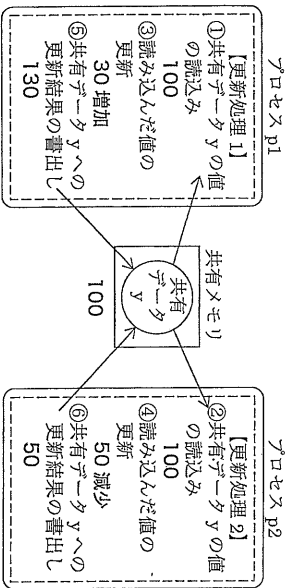


図 A プロセス p1, p2 による更新結果

プロセスの実行順序にかかわらず、常に正しい値となるよう処理が行わなければならない。つまり、p1→p2 でも、p2→p1 でも結果が同じ (直列化可能という) にならなければならない。これが、排他制御を行う理由である。

直列化可能について簡単に説明しておく。例えば、p1 が共有データ Y の値を 2 倍し、p2 が共有データ Y の値を 1 減少させる場合、p1→p2 では 2y-1, p2→p1 では 2 (y-1) で同じ結果にはならない。直列化可能というのは、どちらかの結果に一致するという意味である。

【設問 2】

排他制御の必要性は設問 1 で確認したように、共有データ (資源) の読み込みを同時並行的に行うプロセスに対して、同時には許さないことである。つまり、図 2 のプロセス p3 か、プロセス p4 のどちらかの処理が完全に終了した後に、次の処理を行えば誤りは発生しないことになる。そこで、プロセス p3 が先に共有データ Y の読み込みを行った場合、プロセス p4 からの読み込み要求があってもロックし、排除する。これを排他制御のロック (P 操作：1→0) という。その役割を担った変数がセマフォ S である。処理が終了した後は、アンロック (V 操作：0→1) を行い、共有データを解放すればよいことになる。

P 操作の 1→0, V 操作の 0→1 を知っていれば、容易に解答できるが、セマフォ変数 S の役割をよく考えて、選択肢の内容を吟味する。

- ・確保されている状態：S=0, 他のプロセスによって確保されている。ロック状態。
- ・解放されている状態：S=1, アンロック状態。

- ・空欄 a：前述したように、P 操作は S=1→0 の操作であるから、P 操作前の S の値は 1 である。したがって、(ウ) の「セマフォ変数 S の値が 1 であれば 0 に変更し、終了する。0 であれば 1 になるまで待った後、0 に変更して終了する」が正解である。
- (エ) は、「0 であれば何もせずに終了する」となっているが、共有データの

読み込みを必要としているわけであるから、通常は待ち状態に入る。何もせず終了するということは、処理が進まないことになるから誤りである。

また、(ア), (イ) は、最初に「セマフォ変数 S の値が 0 であれば 1 に変更し」となっており、逆なので、正しくない。

- ・空欄 b：前述したように、V 操作は S=0→1 の操作であるから、(エ) の「セマフォ変数 S の値を 1 にして終了する」が正解である。

なお、(ア), (イ) は、「待った後」との記述があり、待ち状態ではないから該当しない。また、(ウ) は「S の値を 0 にして終了する」となっており、逆である。

【設問 3】

デッドロックとは、排他制御によって共有データ (資源) が確保できないようにロックされている状態である。他が使用できないために、お互いが待ち状態に陥り、処理が進まない。デッドロックの例を図 B に示す。×は確保できないことを示す。

①～④の順番で実行すると、プロセス p5 は③で y2 を確保しようとするが、プロセス p6 の②によって既に確保されているから、解放されるまで待ち状態となる。一方、プロセス p6 は④で y1 を確保しようとするが、y1 はプロセス p5 の①によって既に確保されているから、解放されるまで待ち状態となる。デッドロックが起きるポイントが、二つの資源を互いに逆の順番で確保しようとしたとき」になる。

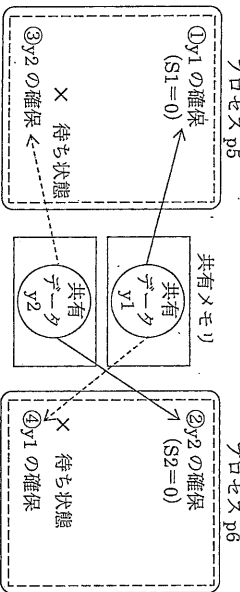
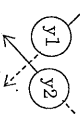


図 B デッドロックの例

「プロセス p6 が、共有データ y1, y2 に対する確保と解放を行う順序によってはデッドロックが発生する可能性がある」場合が問われている。デッドロックは、「二つの資源を互いに逆の順番で確保しようとしたとき」であり、該当しそうなものは (エ), (オ) である。なお、プロセス p5 から先に実行する場合と、プロセス p6 から先に実行する場合について考察しなければならぬ。

エ：プロセス p5 の実行順序：①y1 の確保, ②y2 の確保, ③y2 の解放, ④y1 の解放



プロセス p6 の実行順序：①y2 の確保, ②y1 の確保, ③y1 の解放, ④y2 の解放

先にプロセス p5 が①を実行し、プロセス p6 が①を実行すると、プロセス p5 は②を実行できない。プロセス p6 も②を実行できない。デッドロックが起きる可能性があることが分かる。

オ：プロセス p5 の実行順序：①y1 の確保, ②y2 の確保, ③y2 の解放, ④y1 の解放

プロセス p6 の実行順序：①y2 の確保, ②y2 の解放, ③y1 の確保, ④y1 の解放

先にプロセス p5 が①を実行した場合、プロセス p6 が最初に実行するのは①、②であり、処理は可能である。プロセス p5 も②、③を実行するが、どちらが先でも、処理は可能である。プロセス p6 は、プロセス p5 の④の後に、③、④を実行する。先にプロセス p6 が①、②を実行した場合でも、プロセス p5 が最初に実行するのは①である。②、③を実行するのは、プロセス 6 の①、②の後であり、可能である。プロセス p6 が③、④を実行するのは、プロセス 5 の④の後であり、可能である。デッドロックとはならない。

その他の選択肢についても、その可能性を吟味する。

ア：プロセス p5 の実行順序：①y1 の確保, ②y2 の確保, ③y2 の解放, ④y1 の解放

プロセス p6 の実行順序：①y1 の確保, ②y1 の解放, ③y2 の確保, ④y2 の解放

先にプロセス p5 が①を実行した場合、④が終わるまで、プロセス p6 は①を実行できない。プロセス p5 が④を実行したときには、プロセス p5 は全ての処理が終わるので、プロセス p6 は全ての処理が可能である。

先にプロセス p6 が①を実行した場合は、(ア)と同様である。

先にプロセス p5 が①を実行した場合は、②の後にプロセス p5 が①を実行することになる。このタイミソングで、プロセス p6 は先に④が終わるので、プロセス p5 は②、③の処理が可能である。したがって、デッドロックとはならない。

イ：プロセス p5 の実行順序：①y1 の確保, ②y2 の確保, ③y2 の解放, ④y1 の解放

プロセス p6 の実行順序：①y1 の確保, ②y2 の確保, ③y1 の解放, ④y2 の解放

先にプロセス p5 が①を実行した場合は、(ア)と同様である。

先にプロセス p6 が①を実行した場合は、④が終わるまで、プロセス p5 は①を実行できない。プロセス p6 が④を実行したときには、プロセス p6 は全ての処理が終わるので、プロセス p5 は全ての処理が可能である。したがって、デッドロックとはならない。

ウ：プロセス p5 の実行順序：①y1 の確保, ②y2 の確保, ③y2 の解放, ④y1 の解放

プロセス p6 の実行順序：①y1 の確保, ②y2 の確保, ③y2 の解放, ④y1 の解放

先にプロセス p5 が①を実行した場合は、(ア)と同様である。

先にプロセス p6 が①を実行した場合は、④が終わるまで、プロセス p5 は①を実行できない。プロセス p6 が④を実行したときには、プロセス p6 は全ての処理が終わるので、プロセス p5 は全ての処理が可能である。したがって、デッドロックとはならない。

(エ) が正解である。