

第2章 プロセスとスレッド 演習問題

1. 図2-2では、3つのプロセスの状態が示されています。理論上、3つの状態があれば、各状態から2つずつ、合計6つの遷移が考えられます。しかし、図には4つの遷移しか示されていません。示されていない遷移の一方または両方が発生するような状況はありますか。
2. 割り込みではなく、ハードウェアでプロセスの切り替えを行う高度なコンピュータアーキテクチャを設計します。CPUはどのような情報を必要としますか。ハードウェアによるプロセス切り替えがどのように機能するか説明しなさい。
3. 現在のすべてのコンピュータにおいて、割り込みハンドラは少なくともその一部がアセンブリ言語で書かれています。それはなぜですか。
4. 割り込みやシステムコールがOSに制御を移す際、割り込まれたプロセスのスタックとは別に、カーネル用のスタック領域が一般的に使用されます。それはなぜですか。
5. あるコンピュータシステムには、主記憶に5つのプログラムを保持するのに十分な領域があります。これらのプログラムは、半分の時間はI/O待ちでアイドル状態です。CPU時間の何割が無駄になりますか。
6. あるコンピュータには4GBのRAMがあり、そのうちOSが512MBを占有しています。プロセスはすべて256MB（簡単のため）で、同じ特性を持つとします。99%のCPU利用率を目標とする場合、許容できる最大のI/O待ちの割合はどれくらいですか。
7. 2つのジョブがあり、それぞれ20分間のCPU時間を必要とするとします。これらが同時に開始された場合、順次実行すると最後のジョブが完了するまでにどれくらいの時間がかかりますか。並列実行した場合はどうですか。50%のI/O待ちを仮定しなさい。
8. 多重プログラミング度が6（つまり、同時に6つのプログラムがメモリにある）のシステムを考えます。各プロセスがその時間の40%をI/O待ちに費やすと仮定します。CPU利用率はどうなりますか。
9. インターネットから2GBの大きなファイルをダウンロードしようとしています。そのファイルは複数のミラーサーバーから利用可能で、各サーバーはファイルの一部を提供できます。リクエストではファイルの開始バイトと終了バイトを指定すると仮定します。ダウンロード時間を改善するためにスレッドをどのように利用できるか説明しなさい。
10. 本文では、図2-11(a)のモデルはメモリ内にキャッシュを使用するファイルサーバーには適していないと述べました。それはなぜですか。各プロセスが独自のキャッシュを持つことは可能ですか。
11. マルチスレッドのプロセスがforkした場合、子プロセスが親のすべてのスレッドのコピーを取得すると問題が発生します。元のスレッドの1つがキーボード入力を待ってブロックしていたとします。今や、各プロセスに1つずつ、2つのスレッドがキーボード入力を待っていることになります。この問題はシングルスレッドのプロセスでも発生しますか。
12. 図2-8には、マルチスレッドのWebサーバーが示されています。ファイルから読み取る唯一の方法が通常のブロッキングreadシステムコールである場合、そのWebサーバーにはユーザーレベルスレッドとカーネルレベルスレッドのどちらが使われていると考えられますか。その理由も述べなさい。
13. 本文では、マルチスレッドのWebサーバーについて説明し、それがシングルスレッドのサーバーや有限状態機械のサーバーより優れている理由を示しました。シングルスレッドのサーバーの方が優れている状況はありますか。例を挙げなさい。
14. 図2-12では、レジスタセットはプロセスごとの項目ではなく、スレッドごとの項目としてリストされています。それはなぜですか。マシンにはレジスタセットが1つしかないのに。
15. なぜスレッドはthread_yieldを呼び出して自発的にCPUを明け渡すことがあるのでしょうか。定期的なクロック割り込みがないため、CPUを二度と取り戻せないかもしれないのに。
16. スレッドがクロック割り込みによってプリエンプト（横取り）されることはありますか。もしあるなら、どのような状況ですか。もしないなら、なぜですか。
17. シングルスレッドのファイルサーバーとマルチスレッドのサーバーを使ってファイルを読み取る場合を比較します。データの要求を受け取り、それをディスパッチし、必要な処理を行うのに12ミリ秒かかります（必要なデータがブロックキャッシュにある場合）。ディスク操作が必要な場合（1/3の確率で発生）、追加で75ミリ秒が必要で、その間スレッドはスリープします。サーバーがシングルスレッドの場合、1秒あたり何件のリクエストを処理できますか。マルチスレッドの場合はどうですか。

18. ユーザースペースでスレッドを実装する最大の利点は何ですか。最大の欠点は何ですか。
19. 図2-15では、スレッドの生成とスレッドによって出力されるメッセージがランダムな順序で入り混じっています。これを「スレッド1生成、スレッド1メッセージ出力、スレッド1終了、スレッド2生成、スレッド2メッセージ出力、スレッド2終了」という厳密な順序に強制する方法はありますか。もしあるなら、どうすればよいですか。もしないなら、なぜですか。
20. スレッドにおけるグローバル変数の議論で、変数自身ではなく、変数へのポインタを格納するためにcreate_globalというプロシージャを使用しました。これは不可欠ですか、それともプロシージャは値そのものでうまく機能しますか。
21. スレッドが完全にユーザースペースで実装され、ランタイムシステムが1秒に1回クロック割り込みを受け取るシステムを考えます。あるスレッドがランタイムシステム内で実行中にクロック割り込みが発生した場合、どのような問題が起こり得ますか。それを解決する方法を提案できますか。
22. あるOSには、ファイル、パイプ、またはデバイスから安全に読み取れるかを事前に確認するためのselectシステムコールのようなものがないとします。しかし、ブロックされたシステムコールを中断させるアラームクロックを設定することはできます。これらの条件下で、ユーザースペースにスレッドパッケージを実装することは可能ですか。議論しなさい。
23. turn変数を使ったビジーウェイトの解決策（図2-23）は、2つのプロセスが共有メモリを持つマルチプロセッサ（つまり、共通のメモリを共有する2つのCPU）で実行された場合でも機能しますか。
24. 図2-24に示されているピーターソンの相互排除問題の解決策は、プロセスのスケジューリングがプリエンプティブ（横取り有り）の場合に機能しますか。ノンプレンプティブ（横取り無し）の場合はどうですか。
25. 2.3.4節で議論された優先度逆転問題は、ユーザーレベルスレッドで発生する可能性はありますか。それはなぜですか、あるいはなぜ発生しないのですか。
26. 2.3.4節で、高優先度プロセスHと低優先度プロセスLが無限ループに陥る状況を説明しました。優先度スケジューリングの代わりにラウンドロビンスケジューリングが使われた場合も、同じ問題が発生しますか。議論しなさい。
27. スレッドを持つシステムでは、ユーザーレベルスレッドが使われる場合、スタックはスレッドごとに1つですか、それともプロセスごとに1つですか。カーネルレベルスレッドが使われる場合はどうですか。説明しなさい。
28. コンピュータを開発する際、通常はまず1命令ずつ実行するプログラムによってシミュレートされます。マルチプロセッサでさえ、このように厳密に逐次的にシミュレートされます。このような同時イベントがない状況で、競合状態が発生することは可能ですか。
29. 生産者消費者問題は、1つの共有バッファに書き込む（または読み取る）複数の生産者と消費者がいるシステムに拡張できます。各生産者と消費者が自身のスレッドで実行されると仮定します。図2-28で示されたセマフォを使った解決策は、このシステムで機能しますか。
30. 2つのプロセスP0とP1が関わる相互排除問題に対する以下の解決策を考えます。変数turnは0に初期化されていると仮定します。プロセスP0のコードは以下の通りです。/* その他のコード */while (turn != 0) { } /* 何もせず待つ */クリティカルセクション /* ... */turn = 1; /* その他のコード */プロセスP1については、上記のコードの0を1に、1を0に置き換えます。この解決策が、正しい相互排除の解決策に必要なすべての条件を満たしているか判断しなさい。
31. 割り込みを無効にできるOSは、どのようにセマフォを実装できますか。
32. バイナリセマフォと通常の機械語命令だけを使って、カウンティングセマフォ（任意の値を保持できるセマフォ）をどのように実装できるか示しなさい。
33. システムに2つのプロセスしかない場合、それらを同期させるためにバリアを使う意味はありますか。それはなぜですか、あるいはなぜですか。
34. 同じプロセス内の2つのスレッドは、スレッドがカーネルによって実装されている場合、カーネルセマフォを使って同期できますか。ユーザースペースで実装されている場合はどうですか。他のプロセスのスレッドはそのセマフォにアクセスできないと仮定します。あなたの答えを議論しなさい。
35. モニタ内の同期は、条件変数とwaitおよびsignalという2つの特別な操作を使用します。より一般的な同期形式は、任意のブール述語をパラメータとして持つ単一のプリミティブwaituntilを持つことです。例えば、次のように言えます。waituntil $x < 0$ or $y + z < n$ signalプリミティブはもはや不要になります。この方式は明らかにHoareやBrinch Hansenのものより一般的ですが、使われていません。なぜですか。（ヒント：実装につい

て考えなさい。)

36. あるファストフード店には4種類の従業員がいます。(1)注文係、(2)調理師、(3)袋詰め担当、(4)レジ係です。各従業員は通信する逐次プロセスと見なせます。彼らはどのような形のプロセス間通信を使っていますか。このモデルをUNIXのプロセスと関連付けなさい。
37. メールボックスを使ったメッセージパッシングシステムがあるとして、満杯のメールボックスに送信しようとしたり、空のメールボックスから受信しようとした場合、プロセスはブロックしません。代わりにエラーコードが返されます。プロセスはそのエラーコードに対し、成功するまで何度も再試行します。この方式は競合状態を引き起こす可能性がありますか。
38. CDC 6600コンピュータは、プロセッサ共有と呼ばれる興味深いラウンドロビンの一形態で、最大10個のI/Oプロセスを同時に処理できました。プロセスの切り替えは各命令後に行われ、命令1はプロセス1から、命令2はプロセス2から、という具合でした。プロセスの切り替えは特別なハードウェアによって行われ、オーバーヘッドはゼロでした。競争がない場合にプロセスが完了するのにT秒かかるとすると、n個のプロセスでプロセッサ共有が使われた場合、どれくらいの時間が必要ですか。
39. 以下のCコードを考えます。void main() { fork(); fork(); exit(); } このプログラムの実行時に、いくつの子プロセスが生成されますか。
40. ラウンドロビンスケジューラは通常、実行可能なすべてのプロセスのリストを保持し、各プロセスはリストに1回だけ現れます。もしプロセスがリストに2回現れたらどうなりますか。これを許可する理由は考えられますか。
41. プロセスがCPUバウンドかI/Oバウンドかの尺度は、ソースコードを分析して決定できますか。実行時にこれをどのように決定できますか。
42. ラウンドロビンスケジューリングアルゴリズムにおいて、タイムクォンタムの値とコンテキストスイッチの時間が互いにどのように影響するか説明しなさい。
43. あるシステムの測定によると、平均的なプロセスはI/Oでブロックする前にTの時間だけ実行されます。プロセスの切り替えにはSの時間が必要で、これは事実上無駄（オーバーヘッド）になります。クォンタムQのラウンドロビンスケジューリングについて、以下の各場合のCPU効率の式を求めなさい。
a. $Q = \infty$ (b) $Q > T$ (c) $S < Q < T$ (d) $Q = S$ (e) Qがほぼ0
44. 5つのジョブが実行を待っています。それらの予想実行時間は9、6、3、5、およびXです。平均応答時間を最小化するためには、どのような順序で実行すべきですか。（あなたの答えはXに依存します。）
45. 5つのバッチジョブAからEが、ほぼ同時にコンピュータセンターに到着しました。それらの推定実行時間はそれぞれ10、6、2、4、8分です。それらの（外部で決定された）優先度はそれぞれ3、5、2、1、4で、5が最高優先度です。以下の各スケジューリングアルゴリズムについて、平均プロセスターンアラウンド時間を求めなさい。プロセスの切り替えオーバーヘッドは無視します。
a. ラウンドロビン (b) 優先度スケジューリング (c) 先着順 (10, 6, 2, 4, 8の順で実行) (d) 最短ジョブ優先 (a) については、システムが多重プログラミングされており、各ジョブがCPUの公平なシェアを得ると仮定します。(b)から(d)については、一度に1つのジョブだけが実行され、それが完了するまで続くと仮定します。すべてのジョブは完全にCPUバウンドです。
46. CTSSで実行されるあるプロセスが完了するのに30クォンタム必要です。最初に（全く実行される前に）スワップインされるのを含め、何回スワップインされなければなりませんか。
47. 周期5ミリ秒、1回のCPU時間が1ミリ秒の2つの音声通話と、周期33ミリ秒、1回のCPU時間が11ミリ秒の1つのビデオストリームを持つリアルタイムシステムを考えます。このシステムはスケジューリング可能ですか。
48. 上記の問題について、もう1つのビデオストリームを追加しても、システムはスケジューリング可能ですか。
49. $a = 1/2$ のエージングアルゴリズムが実行時間を予測するために使われています。過去4回の実行時間（古いものから順に）は40、20、40、15ミリ秒です。次の時間の予測値は何ですか。
50. あるソフトリアルタイムシステムには、それぞれ50、100、200、250ミリ秒の周期を持つ4つの周期的イベントがあります。これらの4つのイベントがそれぞれ35、20、10、xミリ秒のCPU時間を必要とするとして、システムがスケジューリング可能であるためのxの最大値は何ですか。
51. 食事する哲学者の問題で、次のプロトコルが使われるとします。偶数番号の哲学者は常に左のフォークを取ってから右のフォークを取ります。奇数番号の哲学者は常に右のフォークを取ってから左のフォークを取ります。このプロトコルはデッドロックのない操作を保証しますか。
52. あるリアルタイムシステムは、それぞれ6ミリ秒ごとに実行され、1回のバーストで1ミリ秒のCPU時間を消費す

る2つの音声通話と、25フレーム/秒で、各フレームが20ミリ秒のCPU時間を必要とする1つのビデオを処理する必要があります。このシステムはスケジューリング可能ですか。

53. カーネルスレッドのスケジューリングにおいて、ポリシーとメカニズムを分離することが望まれるシステムを考えます。この目標を達成する手段を提案しなさい。
54. 食事する哲学者の問題の解決策（図2-47）において、なぜtake_forksプロシージャで状態変数がHUNGRYに設定されるのですか。
55. 図2-47のput_forksプロシージャを考えます。変数state[i]が、2回のtest呼び出しの後ではなく、その前にTHINKINGに設定されたとします。この変更は解決策にどのように影響しますか。
56. 読み手書き手問題は、どのカテゴリのプロセスが開始できるかに関して、いくつかの方法で定式化できます。この問題の3つの異なるバリエーションを慎重に説明し、それぞれがあるカテゴリのプロセスを優遇する（またはしない）ようにしなさい。各バリエーションについて、読み手または書き手がデータベースにアクセスする準備ができたときに何が起るか、またプロセスが終了したときに何が起こるかを述べなさい。
57. ファイルの最後の番号を読み取り、それに1を加えてファイルに追加することで、連続番号のファイルを生成するシェルスクリプトを書きなさい。同じファイルにアクセスするインスタンスを1つはバックグラウンドで、もう1つはフォアグラウンドで実行しなさい。競合状態が現れるまでにどれくらいかかりますか。クリティカルリージョンは何ですか。競合を防ぐようにスクリプトを修正しなさい。（ヒント：ln file file.lockを使ってデータファイルをロックしなさい。）
58. セマフォを提供するOSがあると仮定します。メッセージシステムを実装しなさい。メッセージを送受信するためのプロシージャを書きなさい。
59. 食事する哲学者の問題を、セマフォではなくモニタを使って解きなさい。
60. ある大学が、米最高裁判所の「分離すれども平等は本質的に不平等である」という原則を人種だけでなく性別にも適用することで、政治的に正しいことを示したいと考えています。しかし、伝統への配慮として、女性トイレにいるときは他の女性は入室できるが男性は入れず、その逆も同様であると定めています。各トイレのドアにあるスライド式の表示は、それが現在3つの状態のうちのどれにあるかを示します。・空き・女性在中・男性在中好きなプログラミング言語で、woman_wants_to_enter, man_wants_to_enter, woman_leaves, man_leavesというプロシージャを書きなさい。好きなカウンタや同期手法を使ってかまいません。
61. 図2-23のプログラムを、2つ以上のプロセスを扱えるように書き直しなさい。
62. スレッドを使い、共通のバッファを共有する生産者消費者問題のプログラムを書きなさい。ただし、共有データ構造を保護するためにセマフォや他の同期プリミティブは使用しないでください。各スレッドが必要な時にアクセスできるようにしなさい。満杯と空の状態を処理するためにsleepとwakeupを使いなさい。致命的な競合状態が発生するまでにどれくらいかかるか見てみましょう。例えば、生産者が時々数字を出力するようにするとよいでしょう。I/Oが競合状態に影響を与える可能性があるので、1分に1つ以上の数字を出力しないでください。
63. あるプロセスをラウンドロビンのキューに複数回入れることで、より高い優先度を与えることができます。あるプログラムの複数のインスタンスを実行し、それぞれがデータプールの一部を処理することも同じ効果があります。まず、数値のリストが素数かどうかをテストするプログラムを書きなさい。次に、そのプログラムの複数のインスタンスが同時に実行できるようにし、どの2つのインスタンスも同じ数値を処理しないようにする方法を考案しなさい。実際に、プログラムの複数のコピーを実行することで、リストをより速く処理できますか。注意：あなたの結果は、コンピュータが他に行っていることによって変わります。このプログラムのインスタンスのみを実行しているPCでは改善は期待できませんが、他のプロセスがあるシステムでは、この方法でCPUのより大きなシェアを獲得できるはずです。
64. この演習の目的は、与えられた数が完全数であるかどうかを判断するためのマルチスレッドの解決策を実装することです。Nが完全数であるとは、N自身を除くすべての約数の和がNである場合です。例として6と28があります。入力は整数Nです。出力は、その数が完全数であればtrue、そうでなければfalseです。メインプログラムはコマンドラインからNとPを読み取ります。メインプロセスはP個のスレッドを生成します。1からNまでの数はこれらのスレッドに分割され、2つのスレッドが同じ数を処理しないようにします。このセット内の各数について、スレッドはそれがNの約数であるかどうかを判断します。もし約数であれば、Nの約数を格納する共有バッファにその数を追加します。親プロセスはすべてのスレッドが完了するのを待ちます。ここでは適切な同期プリミティブを使用してください。親はその後、入力された数が完全数であるか、つまりNがそのすべての約数の和であるかを判断し、それに応じて報告します。（注意：検索する数を1からNの平方根までに制限するこ

とで、計算を高速化できます。)