

第1章 序論 演習問題

1. オペレーティングシステムの主な機能を2つ挙げなさい。
2. 1.4節では、9種類のオペレーティングシステムが説明されています。これらのシステムそれぞれについて、アプリケーションの例を挙げなさい（OSの種類ごとに1つ）。
3. タイムシェアリングシステムとマルチプログラミングシステムの違いは何ですか。
4. キャッシュメモリを使用するため、メインメモリは通常32バイトまたは64バイト長のキャッシュラインに分割されます。キャッシュライン全体を一度にキャッシュすることには、単一のバイトやワードを一度にキャッシュするのと比べてどのような利点がありますか。
5. 初期のコンピュータでは、読み書きされるデータの各バイトはCPUによって処理されていました（つまり、DMAはありませんでした）。このことは、マルチプログラミングにどのような影響を与えますか。
6. I/Oデバイスへのアクセスに関連する命令は、通常は特権命令です。つまり、カーネルモードでは実行できますが、ユーザーモードでは実行できません。これらの命令が特権命令である理由を説明しなさい。
7. 1960年代にIBM System/360メインフレームで導入された「コンピュータファミリー」という考え方は、現在では完全に時代遅れのものなのでしょうか、それとも生き続けているのでしょうか。
8. GUIが当初なかなか採用されなかった理由の一つは、それをサポートするために必要なハードウェアのコストでした。25行×80列の文字モノクロテキストスクリーンをサポートするには、どれくらいのビデオRAMが必要ですか。1200×900ピクセルの24ビットカラービットマップではどうですか。1980年当時の価格（\$5/KB）では、このRAMのコストはいくらでしたか。現在ではいくらですか。
9. オペレーティングシステムを構築する上での設計目標には、リソース使用率、適時性、堅牢性など、いくつかのものがああります。互いに矛盾する可能性のある2つの設計目標の例を挙げなさい。
10. カーネルモードとユーザーモードの違いは何ですか。2つの異なるモードを持つことが、オペレーティングシステムの設計にどのように役立つか説明しなさい。
11. 255GBのディスクには65,536個のシリンダがあり、1トラックあたり255セクタ、1セクタあたり512バイトです。このディスクにはいくつのプラッタとヘッドがありますか。平均シリンダシーク時間が11ミリ秒、平均回転遅延が7ミリ秒、読み取り速度が100MB/秒と仮定して、1つのセクタから400KBを読み取るのにかかる平均時間を計算しなさい。
12. 以下の命令のうち、カーネルモードでのみ許可されるべきものはどれですか。
 - a. すべての割り込みを無効にする。
 - b. 時刻時計を読み取る。
 - c. 時刻時計を設定する。
 - d. メモリマップを変更する。
13. あるシステムには2つのCPUがあり、各CPUは2つのスレッド（ハイパースレッディング）を持っています。3つのプログラムP0、P1、P2が、実行時間はそれぞれ5、10、20ミリ秒で開始されたとします。これらのプログラムの実行が完了するまでにどれくらいの時間がかかりますか。3つのプログラムはすべて100% CPUバウンドであり、実行中にブロックせず、一度割り当てられたCPUを変更しないものと仮定します。
14. あるコンピュータには4ステージのパイプラインがあります。各ステージがその処理にかかる時間は同じで、1ナノ秒です。このマシンは1秒あたり何命令を実行できますか。
15. あるコンピュータシステムには、キャッシュメモリ、メインメモリ（RAM）、ディスクがあり、オペレーティングシステムは仮想メモリを使用しています。キャッシュから1ワードにアクセスするのに1ナノ秒、RAMから1ワードにアクセスするのに10ナノ秒、ディスクから1ワードにアクセスするのに10ミリ秒かかります。キャッシュヒット率が95%で、メインメモリヒット率（キャッシュミスの後）が99%の場合、1ワードにアクセスするための平均時間はどれくらいですか。
16. ユーザープログラムがディスクファイルを読み書きするためにシステムコールを行う際、どのファイルを対象とするか、データバッファへのポインタ、そしてバイト数を指定します。その後、制御はオペレーティングシステムに移り、適切なドライバが呼び出されます。ドライバがディスクを起動し、割り込みが発生するまで終了すると仮定します。ディスクからの読み込みの場合、呼び出し元は（データがないため）ブロックされる必要があります。ディスクへの書き込みの場合はどうでしょうか。呼び出し元はディスク転送の完了を待ってブロックされる必要がありますか。

17. トラップ命令とは何ですか。オペレーティングシステムにおけるその使用法を説明しなさい。
18. タイムシェアリングシステムにおいて、なぜプロセステーブルが必要なのですか。単一ユーザーでUNIXやWindowsを実行しているパーソナルコンピュータシステムでも必要ですか。
19. 空ではないディレクトリにファイルシステムをマウントしたい理由がありますか。もしあるなら、それは何ですか。
20. 以下のシステムコールそれぞれについて、失敗する原因となる条件を挙げなさい：fork、exec、unlink。
21. 以下のリソースを共有するために、どの種類の多重化（時間、空間、または両方）が使用できますか：CPU、メモリ、ディスク、ネットワークカード、プリンタ、キーボード、ディスプレイ。
22. `count = write(fd, buffer, nbytes);` という呼び出しは、countにnbytes以外の値を返すことがありますか。もしあるなら、それはなぜですか。
23. ファイルディスクリプタfdを持つファイルに、3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5 というバイト列が含まれているとします。以下のシステムコールが実行されます。 `lseek(fd, 3, SEEK_SET); read(fd, &buffer, 4);` ここでlseek呼び出しは、ファイルの3バイト目にシークします。readが完了した後、bufferには何が含まれていますか。
24. 10MBのファイルがディスクの同じトラック（トラック50）の連続したセクタに保存されているとします。ディスクアームは現在、トラック番号100の上にあります。このファイルをディスクから取得するのにどれくらいの時間がかかりますか。アームをあるシリンダから次のシリンダへ移動するのに約1ミリ秒、ファイルの先頭が保存されているセクタがヘッドの下に回転してくるのに約5ミリ秒かかると仮定します。また、読み取りは200MB/秒の速度で行われるとします。
25. ブロックスペシャルファイルとキャラクタスペシャルファイルの根本的な違いは何ですか。
26. 図1-17の例では、ライブラリプロシージャはreadと呼ばれ、システムコール自体もreadと呼ばれています。これら両方が同じ名前であることが不可欠ですか。もしそうでなければ、どちらがより重要ですか。
27. 現代のオペレーティングシステムは、プロセスのアドレス空間をマシンの物理メモリから分離しています。この設計の利点を2つ挙げなさい。
28. プログラマにとって、システムコールは他のライブラリプロシージャの呼び出しと同じように見えます。プログラマがどのライブラリプロシージャがシステムコールになるかを知ることは重要ですか。どのような状況で、そしてなぜですか。
29. 図1-23は、多くのUNIXシステムコールには対応するWin32 APIがないことを示しています。Win32に対応するものがないとリストされている各コールについて、UNIXプログラムをWindowsで実行するために変換するプログラマにとってどのような影響がありますか。
30. ポータブルなオペレーティングシステムとは、あるシステムアーキテクチャから別のアーキテクチャへ何の変更もなしに移植できるものです。完全にポータブルなオペレーティングシステムを構築することがなぜ実行不可能であるかを説明しなさい。移植性の高いオペレーティングシステムを設計する際に持つであろう2つの高レベルなレイヤーを説明しなさい。
31. ポリシーとメカニズムの分離が、マイクロカーネルベースのオペレーティングシステムの構築にどのように役立つか説明しなさい。
32. 仮想マシンはさまざまな理由で非常に人気が出ています。しかし、いくつかの欠点もあります。1つ挙げなさい。
33. 単位変換を練習するための質問です：
 - a. 1ナノ年は何秒ですか。
 - b. マイクロメートルはしばしばミクロンと呼ばれます。1メガミクロンはどれくらいの長さですか。
 - c. 1ペタバイトのメモリには何バイトありますか。
 - d. 地球の質量は6000ヨタグラムです。これは何キログラムですか。
34. 図1-19に似ているが、実際に動作してテストできるだけのコードを含むシェルを作成しなさい。入力と出力のリダイレクション、パイプ、バックグラウンドジョブなどの機能を追加してもよいでしょう。
35. 安全にクラッシュさせて再起動できる個人用のUNIXライクなシステム（Linux、MINIX 3、FreeBSDなど）が利用できる場合、無限に子プロセスを作成しようとするシェルスクリプトを書き、何が起るか観察しなさい。実験を実行する前に、シェルにsyncと入力してファイルシステムのバッファをディスクにフラッシュし、ファイルシステムを破壊しないようにしてください。仮想マシン内で安全に実験を行うこともできます。注意：共有システムでこれを試す場合は、まずシステム管理者の許可を得てください。結果はすぐに明らかになるため、捕まる可能性が高く、制裁が科されるかもしれません。

36. UNIXライクなシステムまたはWindowsディレクトリの内容を、UNIXのodプログラムのようなツールで調べて解釈してみてください。（ヒント：これをどのように行うかは、OSが何を許可するかに依存します。うまくいくかもしれない一つの方法は、あるオペレーティングシステムでUSBスティックにディレクトリを作成し、次にそのようなアクセスを許可する別のオペレーティングシステムを使用して生のデバイスデータを読み取ることです）。