

Operating System

(CC-BY 4.0) 2024 Taniii.com

2024-08-03

01-7.pdf: 第7章 プロセスとその管理

1. OS がプロセスを管理する必要性を説明せよ

OS がプロセスを管理する必要性は、複数のプログラムを効率的に実行するためです。プロセス管理を通じて、システム資源（CPU、メモリ、I/O デバイスなど）を適切に分配し、プロセス間の干渉を防ぎ、システム全体の安定性と効率を向上させることができます。

2. プロセスとプログラムの違いをまとめよ

プログラムはディスク上に保存された静的な命令の集まりであり、プロセスはプログラムを実行するための動的な実行環境です。プログラムがメモリにロードされ、CPU により実行されると、プロセスとして扱われます。プロセスにはプログラムの実行状態やリソースの情報が含まれます。

3. 利用者から見たプロセスと物理的に見たプロセスの違いを説明せよ

利用者から見たプロセスは、複数のプログラムが並列に実行されているように見えます。一方、物理的に見たプロセスは、実際にはプロセッサが高速に切り替わりながら各プロセスを順次実行しているだけであり、同時に実行されているわけではありません。

4. プロセスを構成するものには何が含まれているか

プロセスを構成するものには、以下が含まれます：

- プロセッサの実行環境（レジスタの値や状態）
- メモリ空間（アドレス空間）
- 開かれているファイル
- ファイルポインタの位置やモード
- 親プロセスの情報
- 使用ユーザの情報

5. プロセッサの3つの状態とは 実行中 と 実行可能 と 待ち状態 である

プロセッサの3つの状態は以下の通りです：

- 実行中：プロセッサを割り当てられて実行中の状態
- 実行可能（レディ状態）：プロセッサの割り当てを待ち、いつでも実行できる状態
- 待ち状態：I/O の完了などの事象を待っており、実行できない状態

6. プロセスの状態が変化するのはどのようなときか説明せよ

プロセスの状態は以下のような場合に変化します：

- 実行中から待ち状態：I/O 操作の開始などの事象待ちの際に発生
- 実行中から実行可能状態：プリエンプションなどによりプロセッサが他のプロセスに割り当てられた際に発生
- 実行可能状態から実行中：スケジューラによりプロセッサが割り当てられた際に発生
- 待ち状態から実行可能状態：待ち事象が完了した際に発生

7. プロセススケジューラの役割を説明せよ

プロセススケジューラの役割は、システム内の全てのプロセスの状態を把握し、次に実行するプロセスを選択することです。これにより、システムの効率を最大化し、ユーザの使用感を向上させます。

8. I/O バウンドなプロセスとはどのようなプロセスか述べてよ

I/O バウンドなプロセスとは、頻繁に I/O 操作を行い、CPU の使用時間よりも I/O 待ち時間が長いプロセスのことです。例えば、テキストエディタやワープロなどが該当します。

9. CPU バウンドなプロセスとはどのようなプロセスか述べてよ

CPU バウンドなプロセスとは、ほとんど I/O 操作を行わず、CPU の使用時間が長いプロセスのことです。例えば、科学技術計算や物理演算、人工知能（AI）などが該当します。

10. ダイナミックディスパッチングを用いたときの利点を説明せよ

ダイナミックディスパッチングの利点は、I/O バウンドなプロセスの優先度を上げ、CPU バウンドなプロセスの優先度を下げること、システム全体の効率を向上させることです。これにより、プロセッサと入出力装置の並列処理が可能となり、トータルのプロセス処理時間が短くなります。

11. 多段フィードバックキューでは優先度はどのように決定されるか説明せよ

多段フィードバックキューでは、プロセスは初めに高い優先度が与えられますが、一定時間内に実行が終わらない場合、次のレベルに落とされます。待ち状態から戻ったプロセスには高い優先度が与えられるため、短いプロセスの処理が優先され、ダイナミックディスパッチングも実現されます。

12. (参考) プロセスの状態を区別しないとした場合、スケジューリングアルゴリズムにどのような影響を与えるか検討せよ

プロセスの状態を区別しない場合、スケジューリングアルゴリズムは全てのプロセスを一律に扱うことになり、待ち状態や実行可能状態の区別がなくなるため、システム効率が低下します。待ち状態のプロセスにも CPU が割り当てられることになり、無駄な CPU 時間が発生します。

13. (参考) ラウンドロビンでタイムスライスを 10ms などの短い時間に設定した場合、どのような利点・欠点が生じるか検討せよ

利点：

- 各プロセスに対する応答時間が短くなり、インタラクティブなユーザの体感速度が向上する。 欠点：

- プロセス切り替えのオーバーヘッドが増加し、システム全体の効率が低下する。

14. (参考) あるプロセスが、I/O バウンドであるか CPU バウンドであるか判定するための方法を検討せよ

I/O バウンドか CPU バウンドかを判定するためには、プロセスの I/O 待ち時間と CPU 使用時間を計測し、比較する方法があります。I/O 待ち時間が長ければ I/O バウンド、CPU 使用時間が長ければ CPU バウンドと判定できます。また、特定の期間内の I/O 操作の回数と CPU 使用回数の比率を計算する方法も有効です。

(CC-BY 4.0) 2024 Taniii.com