プロセッサ性能を可変提供するためのスケジュール法

1V－6

谷口　秀夫
九州大学工学部

１．はじめに

ハードウェアの性能向上とともに、計算機のソフトウェア処理時間は短縮の一途をたどっている。これにより、複雑な処理を行うことができるようになり、いろいろな使いやすいサービスが実現されている。例えば、１０年前までは、ハードウェアの支援なしには考えられなかったマルチウィンドの環境が、現在ではソフトウェアだけで実現されている。

しかし、ハードウェアの性能向上は、いくつかの問題を生んでいる。例えば、種々の性能を持つプロセッサの存在は、プロセッサ性能に依存した処理を面倒なものにしている。Ｉ／Ｏ制御において制御間隔を調整する処理は、プロセッサ性能に合わせた調整が必要になっている。もう１つの例としては、プロセッサ性能が高過ぎ、プログラムの処理時間が速過ぎることである。文字の表示速度を人間に合わせて調整することがある。このように、ハードウェア性能が高くなるにつれて、様々な問題が発生すると思われる。したがって、今後は、ハードウェア性能に惑わされず、人間の感覚の時の流れに合わせた計算機利用が必要になる。

このような背景により、ハードウェア性能に関係なくサービス処理時間を保証するＯＳの研究を進めている。ここでは、サービスの処理時間を自由に設定できるために、要求に応じたプロセッサ性能を提供するスケジュール法について述べる。

２．サービス処理時間の決定要因とＯＳへの要求

２．１　決定要因

サービス処理は、いつ始まり（開始時刻）、どのくらいの割合で処理を行い（処理速度）、いつ終わるか（終了時刻）、により動きを把握できる。また、どのくらいの時間で処理を行うか（処理時間）により動きを把握することもできる。さらに、サービスの処理内容を見ると、その処理速度は一定ではなく可変であることが好ましい。これは、処理が速いほどよい部分や外界との接触にあわせて適当な処理の速度を求める部分が混在しているためである。ここで、外界とは、プロセッサの周辺にある装置や人間のことである。

したがって、サービス処理は、開始時刻と処理量および処理速度を要因として、その動作が決定されると考えられる。この時、処理時間は、処理量と処理速度により決定できる。また、終了時刻は、開始時刻と処理時間により決定できる。

２．２　ＯＳへの要求

ここでは、サービス処理の処理量は応用プログラム（以降、ＡＰと略す）で把握していると仮定する。したがって、サービス処理を決定する要因から、ＯＳに対しては、以下の要求がある。

（１）処理の開始時刻を設定できること
（２）処理速度に相当する性能の設定ができること

さらに、サービスの処理速度は可変であることが好ましいため、

（３）性能を変更できること

がＯＳに望まれる。また、ＯＳの制御処理への要求として、

（４）要求の性能に合わせた処理の均一性

が必要である。

３．制御方式

３．１　時刻の制御

時刻を制御しＡＰに提供する。提供する機能は、ＯＳへの要求とＡＰからの利用しやすさを考慮し、以下のものとする。

（１）ＯＳへの要求を受けて、サービス処理の開始時刻の設定機能を提供する。
（２）時刻は絶対時刻で提供する。相対時刻の提供は、絶対時刻からの変換により可能である。
（３）サービス処理の異常や処理量が不明確なサービスの走行を制御するため、サービス処理の終了時刻の設定機能を提供する。

時刻を制御するに当たり、その確度を向上させるため、以下の制御を行う。

（Ａ）プロセッサの即時割り当て

開始時刻や終了時刻になると、プロセッサを即時に割り当て処理を開始または終了させる。多くのＯＳで行われているように単にプロセスのレディーキューにつなぐだけでは、本当の実行開始の時刻を保証できない。そのため、単にプロセスのレディーキューにつなぐのではなく、プロセッサを割り当てて実行する。このために、プロセスの実行優先度を越えたスケジュール制御を行う。

Scheduling Mechanism for Supplying Variable Performance of Processor
Hideo TANIGUCHI
Faculty of Engineering, Kyushu University

（B）２重要求の排除

既に要求されている開始時刻や終了時刻と同じ時刻に対する開始時刻や終了時刻の設定要求は、受け付けず排除する。これにより、開始処理や終了処理の開始をより正確に行える。

（C）終了時刻による強制終了

終了時刻になっても存在するプロセスは、強制的に終了させる。これは、サービスの異常動作などを防ぐだけではない。プロセスのハードウェア使用量を明確にすることにより、性能の確保も行える。ただし、強制終了を行う処理量は、性能を制御する上で考慮が必要である。

時刻を制御するに当たり、その精度を向上させるためには、時刻検出から当該プロセスをディスパッチまでのＯＳ処理時間の短縮が重要である。

３．２　性能の制御

プロセッサの性能を可変提供するには、

・プロセッサの割当方式

・プロセッサの途中放棄への対処

・プロセッサ性能の新規要求への対処

が課題である。ここでは、プロセッサの割当方式について、以降に述べる。

サービス処理に対しプロセッサの性能を調整するには、サービス処理の実行と停止を適度な間隔で繰り返すことが必要になる。このように、プロセッサの性能を調整し制御するには、プロセッサの実行を単位時間（これをタイムスロットと名付ける）に分割し、サービス処理にタイムスロットを割り当てる割合を調整することで可能である。この場合、タイムスロットをサービス処理に割り当てる方式が課題になる。

この方式は、大きく２つに分類できる。１つは、サービス処理を実行するタイムスロットを必要な周期で割り当てる方法である。これを周期割当方式と名付ける。もう１つは、連続する複数のタイムスロット（例えばN個の連続したタイムスロット。これをタイムブロックと名付ける）内で、サービス処理を実行するタイムスロット数を必要個数割り当てる方式である。これをブロック内割当方式と名付ける。タイムスロットとタイムブロックの関係を図１に示す。図１では、タイムスロットは時間ｔであり、タイムブロックは時間ｔ×Ｎである。また、各割当方式を図２に示す。図２（ａ）は、周期割当方式の例

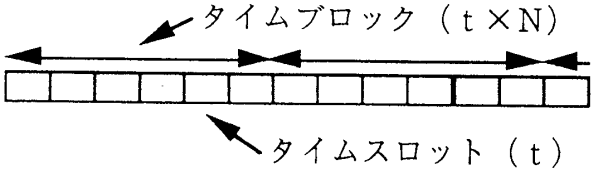
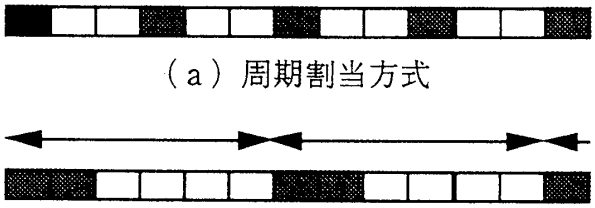


図１　タイムスロットとタイムブロック



（ａ）周期割当方式

（ｂ）ブロック内割当方式

図２　タイムスロットの割当方式

であり、２つおきにタイムスロットを割り当てた場合である。この場合、サービス処理は、本来のプロセッサの性能の３分の１の性能で実行される。図２（ｂ）は、ブロック内割当方式の例であり、Ｎ＝６である。タイムブロック内の２タイムスロットがサービス処理に割り当てられている。このため、（ａ）と同様に、サービス処理は、本来のプロセッサの性能の３分の１の性能で実行される。

ブロック内割当方式では、タイムブロック内での割当タイムスロットの位置は提供するプロセッサの性能に影響を与えない。しかし、タイムブロック毎に割当タイムスロットの位置を変えることによる長所はない。むしろ、タイムブロック内では同じ位置のタイムスロットを与える方が、サービス処理の均一性を高める。

周期割当方式は、サービス処理の均一性が高いものの、割当が均等でなくなることが多い。一方、ブロック内割当方式は、サービス処理の均一性を犠牲にして割当を行う方式である。

いづれの方式においても、確度や精度の確保は同様な対処になる。確度の確保は、時刻の制御と同様に「プロセッサの即時割り当て」が重要である。タイムスロットの獲得とともにサービス処理のプロセッサ実行を開始させる。精度の確保は、プロセス切り替え時間がタイムスロットの時間間隔に比べ十分小さいことである。つまり、

・タイムスロット＞＞プロセス切り替え時間

である。サービス処理から見て処理が連続的に動いているようにするためには、要求性能から決定されたタイムスロット間隔が処理の連続性を損なわない時間間隔より小さいことである。つまり、

・割当ﾀｲﾑｽﾛｯﾄ間隔＜連続性を損なわない時間間隔

である。

４．おわりに

プロセッサ性能を可変提供するため、時刻とプロセッサ性能の制御方式について報告した。

今後は、プロセッサ性能の制御における「プロセッサの途中放棄」と「プロセッサ性能の新規要求」への対処を検討する。さらに、検討結果を試作し評価する予定である。