

連続メディア処理向けタスクスケジューリング機能の開発

3D-1

鈴木 貴明 吉澤 康文
東京農工大学 工学部

1 はじめに

ビデオオンデマンド(VOD)や遠隔会議システムなど連続メディアを扱うシステムの処理は、同一の基本処理を周期的に繰り返す性質を持つ。これまでのオペレーティングシステム(OS)で、アプリケーション(AP)に対して周期的な計算資源割当てを保証する機能を持つものは少ない。計算機への負荷が少ない場合は周期的な計算資源の確保が可能だが、過負荷に陥った場合、資源確保の周期性が崩れ、動画のこま落ちなど、連続メディア処理特有の障害が出る。

また、OSがAPに対して提供する時間情報は貧弱な場合が多い。APは現在自分が処理している時刻などを把握することが不可能なため、きめ細かい時間制御などが困難であった。

本研究では、OSのタスクスケジューラを改良し、連続メディア処理に適したCPU時間配分を行う機構を開発する。また、APへの時間情報の提供を強化する方法としてシステムコールの処理時間を提供する。

2 周期的タスクのモデル化

周期的なCPU資源分配の対象となるタスクを周期的タスク、それ以外のタスクは非周期的タスクとして定義する。周期的タスクはパラメータとして周期と周期内に必要なCPU時間(必要処理時間と呼ぶ)をもち、生成時にこれらパラメータをスケジューラに対して申告する。この申告が受理され生成が成功すると、タスクは周期的なCPU時間確保が保証される。また同時に、1周期内に必要処理時間以上の処理が禁止される。必要処理時間を超過して処理を行った場合、そのタスクは強制終了される。

周期的タスクは1周期分の処理を終了した時点で、周期内処理の終了をスケジューラに伝達する機能を持つシステムコールを発行し、次の周期開始までスリープする。このスリープは周期開始時刻になると自動的に解除される(図1)。

3 スケジューラの改良方針

周期的タスク間の優先度決定は、周期の短いタス

クを優先するレートモノトニック方式で行う。周期的タスクの優先度は、常に非周期的タスクより高く設定し、周期的タスクの周期性を確保する。

周期的タスクに対して与えたCPU時間は各タスク毎に集計し、周期内で与えた総CPU時間が必要処理時間を超過した場合、そのタスクを終了させ、システム全体のCPU時間の不足を防止する。

また、各周期的タスクの周期終了時刻を監視し、周期終了時刻に達した周期的タスクにはウエイクアップ処理を行い、次周期の処理を開始させる。

周期毎に必要な処理時間は予め予約し、周期的タスクの要求CPU時間を一定値以下に押さえ、各周期的タスクへのCPU時間の不足を防止する。

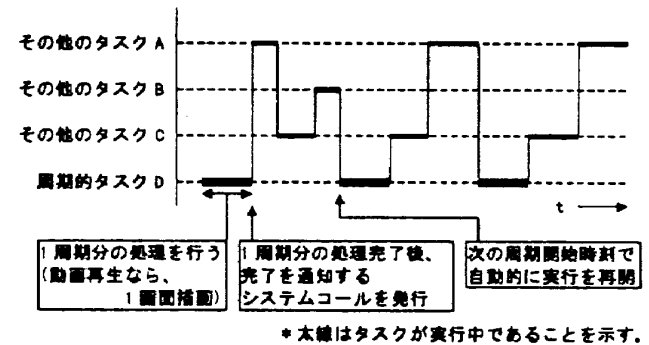
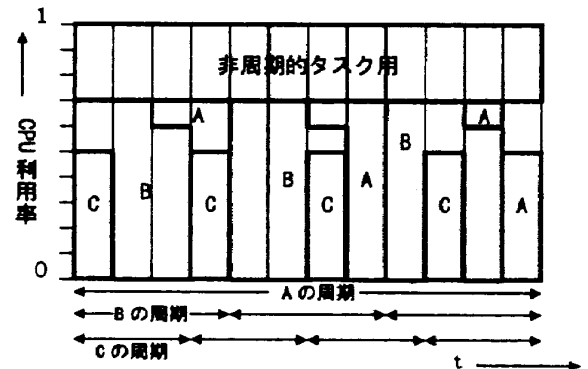


図1 周期的タスクモデル

周期的タスクA(周期12、必要処理時間1.8)と
周期的タスクB(周期4、必要処理時間1.5)がすでに存在し、
周期的タスクC(周期3、必要処理時間0.5)を新規に生成する場合の判定



各周期の最小公倍数の範囲で、周期の小さいタスクから必要処理時間を当てはめていく。すべてのタスクを収めることが可能な場合、スケジューリング可能と判定する。

図2 スケジューリング可能性判定の例

4 スケジューリング可能性判定

周期的タスクは生成時に周期と必要処理時間を申告するが、このとき、既存の周期的タスクのCPU

時間確保を妨げることなく、希望どおりのCPU時間割当てが可能かを判定する。図2に示すように、各タスクの周期の最小公倍数を範囲とする表を作成し、既存の周期的タスクがCPU時間の確保を妨げない場合、スケジューリング可能と判断する。

5 時間情報の提供

カーネルがタスクに対して提供する時間情報を強化する一環として、システムコールの処理時間を提供する。システムコールの処理が完了する毎に、ユーザが予め指定した変数に処理時間が自動的に格納される。

6 実装

実装はLinux Kernel 2.0.0上に行った。今回実現したシステムコールを次に示す。

period_start

呼出し元のタスクを周期的タスクに移行させる。引数として、周期と必要処理時間を取る。

period_end

周期的タスクとしての動作を終了し、非周期的タスクへ戻す。

period_skip

次の周期の開始までスリープさせる。次の周期の開始時刻にスケジューラがウェイクアップさせる。

syscalltime_start

システムコール処理時間の取得を開始する。引数として、処理時間の格納先アドレスを取る。

syscalltime_end

システムコール処理時間の取得を終了する。

実現規模を表1に示す。

表1 本研究の実現規模

	分類	行数
1	スケジューラの改良	700
2	時間情報の強化	150

7 評価

次の環境で、周期的タスクの周期性を評価した。

- マシン : Gateway2000 G6-200
- CPU : Pentium Pro 200MHz
- メモリ : 32Mbyte

CPUに対する負荷を変化させた場合の周期性を計測した。周期は10msとし、参考として既存のシステムコールを用いてスリープを繰り返す擬似的な周

期的タスクの周期性も測定した。負荷タスクは、空ループとスリープを繰り返し、計測タスクと同じ条件となるようにした。

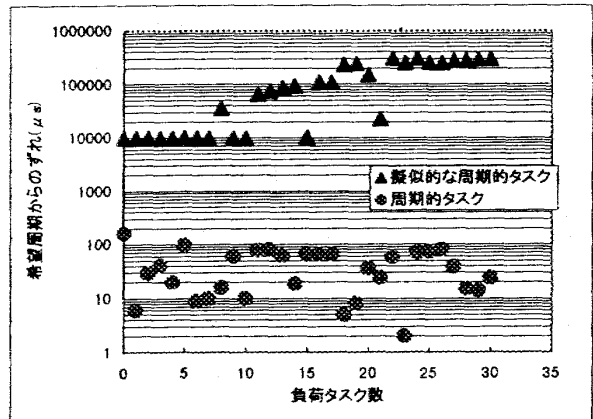


図3 周期10msでの周期性

8 スケジューリングモニタの実現

スケジューラを改良する際のスケジューリング動作確認を容易にするため、スケジューリングの様子を視覚化するスケジューリングモニタを作成した。次に示すOS内のイベントを取得し、視覚化する。

- ・タスクスイッチ
- ・システムコール発行
- ・割り込み発生

タイムスタンプの取得にはPentiumのクロックカウンタ取得命令を用いた。実行画面を図4に示す。

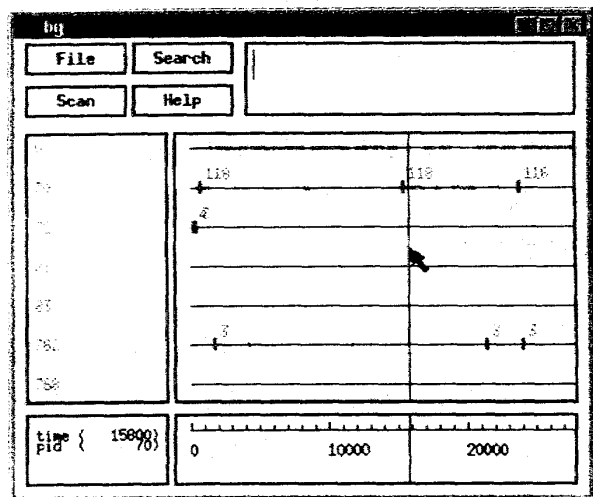


図4 スケジューリングモニタの実行画面

9 まとめ

連続メディア処理に向けた、周期的な資源割当てを保障するタスクスケジューリング機能を開発した。今後は複数タスクの協調スケジューリングについて研究を進める予定である。