

プロセスの実行優先度を調整して応答速度を上げる

リアルタイム性の向上

小林 明

4-1 リアルタイム性向上に関する基礎知識

Linuxは、リアルタイム性を追求した設計はされていません。そのため、RTOS(リアルタイムOS)のように「定められた期限内に確実にタスク切り替えの完了を保証する」ようなハード・リアルタイムは実現できません。しかし、スケジューラが他のプロセスよりも優先してCPUを割り当てるリアルタイム・プロセスを作ることができます。

プロセスの実行優先度の調整は、優先度とスケジューリング・ポリシーで決定されます。

プロセスの実行優先度を調整する2つのパラメータ

● 優先度

カーネル内において優先度は1～139の値を取り、数値が小さいほど実行優先度が高いことを意味します。1～99を静的優先度、100～139を動的優先度に分類して管理しています。

▶ 静的優先度

静的優先度は、プロセス実行時に決定されます。また、プロセス動作中であってもchrtコマンドなどを用いて優先度の変更を行うことができます。

▶ 動的優先度

動的優先度は、実行中のプロセスにおいて、CPU使用量やnice値(ユーザ・レベルのプロセス優先度)を参照して動的に決定されます。ユーザは、動的優先度を直接変更することはできませんが、nice値を変更して間接的に変更できます。nice値は-20～+19の値を取り、数値が小さいほど実行優先度が高くなります。

● スケジューリング・クラス/ポリシー

スケジューラは、スケジューリング・クラス、スケジューラ・モジュールの階層で拡張可能な構造になっています。スケジューラ・モジュールはスケジューリング・ポリシーの詳細をカプセル化します。

▶ スケジューリング・クラス

Linuxのスケジューリング・クラスを表1に示します。

▶ スケジューリング・ポリシー

Linuxは、プロセスにCPU時間を割り当てる方法を管理するため、いくつかのスケジューリング・ポリシーを提供します(表2)。これらのポリシーにより、Linuxはプロセスが必要とするCPU時間をうまく管理して、重要なタスクが優先されるようにしながら

表1 Linuxのスケジューリング・クラス
ここに示すのは主なスケジューリング・クラス

スケジューリング・クラス	説明
デッドライン	SCHED_DEADLINE 最も優先度が高く、リアルタイム・アプリケーションに特に適している(カーネル・バージョン3.14で追加)
リアルタイム	SCHED_FIFO / SCHED_RR (POSIX 準拠) sched_priorityにより1(低)～99(高)の優先度の設定が可能。 タイム・シェアリングより優先度が高い
タイムシェアリング [Completely Fair Scheduler (CSF)]	SCHED_OTHER (SCHED_NORMAL) / SCHED_BATCH / SCHED_IDLE sched_priorityは0でリアルタイム・スレッドより優先度が低い。 SCHED_IDLE以外はnice(ユーザ・レベルのプロセス優先度)で通常タスクの優先度を変更できる。 nice値のデフォルトは0で、-20(高)～+19(低)の範囲で設定が可能