# 3N-3　ディスクアクセスのリアルタイムスケジューリング方式

帆波 幸二†　毛利 公一†　吉澤 康文†
†東京農工大学工学部

## 1. はじめに

計算機の高性能化と，ネットワークインフラの拡大と高速化に伴い，動画像や音声などの大容量メディアを扱うマルチメディアシステムが普及してきた．これらマルチメディアシステムでは，必要とする計算機資源が適時的に提供される必要があるため，リアルタイムスケジューリングが必要となる．また，ディスクやイーサネットといった入出力デバイスもリアルタイムに利用できるように管理する必要がある．そこで，本論文では，入出力操作を伴うリアルタイムプロセス(以下RTプロセスと示す)のスケジューリングを可能とする機構を提案する．

本スケジューラで提供される機能は，主に次のものである．

(1) 周期的にプロセスにCPUを割り当てる
(2) RTプロセスと通常プロセスの混在を可能とする
(3) 入出力完了待ちの間もCPUをアイドルさせることなくケジュール可能とする
(4) RTプロセスによる入出力デバイスの使用が通常プロセスによって妨げられることがない

以下，本論文では，2章でプロセスモデル，3章で本機構の概要，4章で考察について述べ，5章で全体のまとめについて述べる．

## 2. プロセスモデル

スケジュールの対象となるプロセスのモデルとしては次のものを想定している．

(1) 周期的RTプロセス
(2) 非周期的RTプロセス
(3) 通常プロセス

RTプロセスは，連続メディア処理のように，一定周期毎に一定時間CPU資源を必要とするものと，単に即時性を求めるだけのものとで性質が異なるので，(1)と(2)のように分けた．

---

Disk Access Scheduling for Real-Time Operating Systems
Koji Honami, Koichi Mouri, Yasufumi Yoshizawa
Faculty of Engineering, Tokyo University of Agriculture and Technology

## 3. スケジューリング機構の概要

### 3.1. 構成

1章で示した機能を実現するためには，CPU資源管理と，入出力デバイス管理の両方からのアプローチが必要となる．CPU資源管理は，RTプロセスと，通常プロセスの混在を可能とするため，リアルタイムスケジューラと非リアルタイムスケジューラの二段に分割した構成にした．リアルタイムスケジューラで使用するアルゴリズムについては3.2節で述べる．入出力デバイスの管理は，RTプロセスに，使用するデバイスとその期間を宣言する義務を持たせ，宣言された期間中，他プロセスからの当該デバイスの利用を制限するという仕組みで，RTプロセスのデバイス利用をリアルタイムに行えるものとしている．

### 3.2. スケジューリングアルゴリズム

RTプロセスのスケジューリングとしては固定優先度決定方式で最適である RM(Rate Monotonic)[1]をベースにDS(Deferrable Server)[2]を適用することで，周期的RTプロセス，非周期的RTプロセスの両方をスケジュール可能である．

しかし，RM，DS共に入出力操作などによるプロセスの待ち状態を想定していない．待ち状態の間もCPUを占有していると考え，待ちそのものを無視してスケジュールする事も可能であるが，待ち状態が解除されるまで，長時間にわたってCPUがアイドル状態となってしまう．本機構では，この問題を解決するために，処理時間の繰延べ機構というものを考案し，待ちの間に他プロセスの処理ができるようにした．

この機構は，CPU時間を，必要としない間はそれを必要としている他プロセスに貸し与え，必要となったときに返してもらうという仕組みである．このように，CPUの処理時間を必要に応じて，繰延べして受け取れるようにすることで，CPUのアイドルを生じさせずに，需要にあったCPUの割当てが可能となる．この機構は，CPU時間を必要としない時にだけ貸し与える仕組みであるので，本来のスケジューリングを崩してしまう心配もない．

### 3.3. 入出力デバイス管理

入出力デバイスの使用は，一種のリソース共有であると考えることができる．デバイスに対する処理は中断できない．したがって，リアルタイムに適したデバイス管理を行わないと，RT プロセスの入出力要求が，通常プロセスの入出力操作により後回しにされる事態が起こる．特に，本機構では，RT プロセスが待ち状態の間に通常プロセスもスケジュールしてしまうため，この現象の起こる確率が大きなものとなっている．この問題は，リアルタイム同期問題における優先度逆転現象によく似ている．そこで，デバイスを使用する区間を臨界領域と考えて，優先度シーリングを利用した制御をすることで，低優先度プロセスによるブロッキングを押さえることに可能にした．(図 1，図 2 参照)

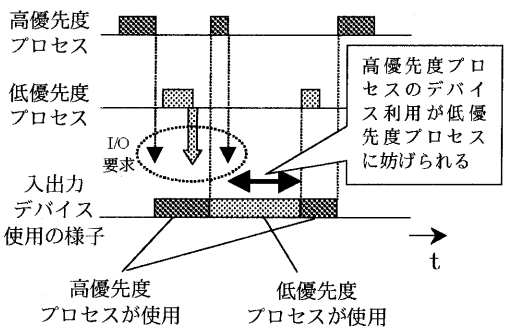


図 1　デバイス利用における優先度逆転

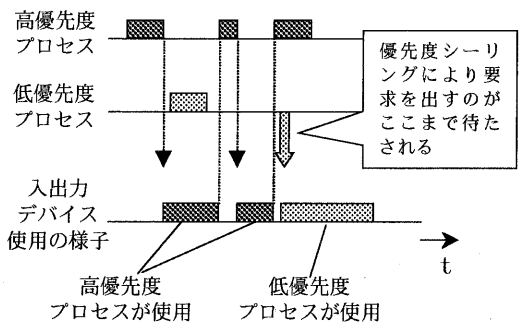


図 2　優先度シーリングによる制御結果

図 1 は，デバイス管理機構が無く，高優先度プロセスが待ち状態の間に低優先度プロセスが I/O 要求を出してしまい，優先度逆転が起こっている．

図 2 は，同じ状況で，デバイス管理機構を適用した場合である．待ちの間の I/O 要求を防ぐことで，高優先度プロセスの終了を早めることに成功している．

## 4. 考察

### 4.1. スケジューリング可能性

本機構におけるスケジューリング可能性判定式は現在のところ算出できていない．現段階で判明していることは，RT プロセスのスケジューリング可能性は DS のそれよりは低いものとなるということである．これは，入出力操作の完了待ちが DS によるリアルタイムスケジューリングに遅延を生じさせることに起因する．

遅延の算出を求めるためのアプローチとして，デバイスの制御に優先度シーリングを利用していることから，PCP(Priority Ceiling Protocol)[3]の判定式を参考にすることを考えている．

### 4.2. デッドラインミスの対処について

オーバーロードなどの原因によるデッドラインミスの発生は，他プロセスへ影響を波及させ，スケジューリングを大きく乱す結果となる．これを防ぐ方法として，デッドラインミスが生じた際には計算機資源を即座に取り上げる機構が考えられる．タイマ割込みの間隔で，常にデッドラインミスを監視し，いつでも資源を取り上げ可能にしておくことで，より安定した動作を実現することが可能である．

## 5. 終わりに

本稿では，入出力操作を伴う RT プロセスをスケジューリングする方法を述べた．今回提案した処理時間の繰延べ機構と，優先度シーリングによるデバイス管理機構により，入出力待ちの間も CPU をアイドルにすることなく利用することを可能とし，また，RT プロセスのデバイス利用が，通常プロセスに妨げられることも回避可能とした．さらに，スケジューラを二段階にすることにより，RT プロセスと通常プロセスを同時にスケジューリングすることも可能となった．

今後の課題としては，割込み処理や，スケジューラなど，OS 自体の処理にかかる時間を想定したスケジューリングを可能にすること，スケジューリング可能性の解析を進めることなどが挙げられる．

## 参考文献

[1] C. L. Liu and J. W. Layland: Scheduling Algorithm for multiprogramming a Hard Real Time Environment, J. of ACM, Vol. 20, No. 1, pp. 46-61 (1973)

[2] Jay K. Strosnider, John P. Lehoczky, Lui sha: The Deferrable Server Algorithm for Enhanced Aperiodic Responsiveness in Hard Real-Time Environments, IEEE Trans. on Computers, Vol. 44, No. 1, pp. 73-91 (1995)

[3] L. Sha, R. Rajkumar, J.P. Lehoczky: Priority Ceiling protocols, IEEE Trans. on Computers, Vol. 20, No. 9, pp.1175-1185, 1990