

## 7F-2

## メッセージ交換における優先度逆転とその対策\*

三浦真 正田輝雄†

明治大学理工学部‡

## 1 はじめに

並行プログラムにおいては、複数のプロセス(タスク)が、共有変数を通して、あるいはメッセージの交換によって、互いの間の通信と同期を行なう。さらに、各プロセスには優先度(priority)が付与されているとする。(プロセスの優先度は現実的なシステムにおいて重要で、実時間処理の周期的なプログラムにおいては、短い周期のプロセスに高い優先度を与えるという方式を用いることが多い。)この状況のもとで、優先度の高いプロセスが低いプロセスに待たされるという優先度逆転(priority inversion)の現象が起ることが知られている。一例は、優先度の高いプロセスAが、低いプロセスCからのメッセージを待って停止(suspend)し、Cの優先度が低いため、Aが中間の優先度のプロセスBに追い越されてしまう状況である。これに対処するための方策がいくつか提案されている。ここでは、優先度逆転がどのような状況において起こるかを解析する。さらに、新しい対策法を提案する。(なお、優先度を整数で表わすものとし、値の大きいほど優先度が高いとする。)

一般に、メッセージ交換方式のうちでも、いくつか種類がある。第一に、同期(synchronous)と非同期(asynchronous)の区別があり、受け付けプロセスのキューには、同期方式の場合は呼出しプロセスが、非同期方式の場合はメッセージがはいる。さらに細かい区別として、キューが、あるかないか、その長さが1か限定長か不定長かがある。現在のAda(1983年規格)は同期方式で、キューは不定長である。優先度逆転の起る状況はこれらの通信・同期の方式によって異なる。

ここでは単一プロセッサ・システムの下で考える。なお、プロセスの優先度と、プロセスのスケジューリングという二つの概念は、互いに独立したものである。ここではスケジューリングはもっとも単純なもので、その時点で実行可能な、待ち状態(blocked)のプロセスの中の、最も優先度の高いものから一つ選んで実行するというものとする。

プロセスの優先度に関する議論は並行処理一般に関わり、言語からは独立である。しかしこの話はこれまで主として、現在進行中のAdaの言語改定にからんで議論されている。我々は種々のインプリメントや実験をCおよびAdaの上で行なっている。現在のAdaでは、プロセスの優先度は、静的にプログラマによって与える。Ada 9Xでは、スケジューリングの方針をユーザが選択できるようにすること、優先度を実行時に変えられるようにすることなどが検討されている。

## 2 優先度逆転

優先度逆転は基本的には、高優先度のプロセスが低優先度のプロセスに何かの理由で待たされているときに、中優先度のプロセスが先にスケジュールされて実行されるという状況である。

## 2.1 共有変数

この場合、後述の優先度継承法によって、またその拡張である優先度上限法によって避けることができることがわかっている。

共有変数を、それにセマフォが付随したものとし、さらにそれら全体を一つのプロセスと見なすと、あるプロセスAが共有変数にアクセスすることを、同期式メッセージ交換における、Aから共有変数のプロセスへメッセージを送ることと見なすことができ、同期式メッセージ交換の特殊の場合と考えることができる。共有変数方式では、共有変数にアクセスする側のみ待ち合わせがある。

## 2.2 同期型メッセージ交換

同期型メッセージ交換では、メッセージを、送る際と、受ける際と、両方で待ち合わせがありえる。送り手と受け手のプロセスの時間的順序と、優先度の上下には、次の四つの場合がある。

1. 低いプロセスが先に送る状態になり、高いプロセスが後で受取る状態になる。
2. 低いプロセスが先に受取る状態になり、高いプロセスが後で送る状態になる。
3. 高いプロセスが先に送る状態になり、低いプロセスが後で受取る状態になる。
4. 高いプロセスが先に受取る状態になり、低いプロセスが後で送る状態になる。

第一と第二の場合には優先度逆転は起らず、第三と第四の場合に起る。後述の優先度継承法を採用すると、第三の場合は優先度逆転は起らないようになるが、第四の場合にはやはり起る。

## 2.3 非同期型メッセージ交換

非同期型メッセージ交換では、受ける際のみ、待ち合わせがありえる。上の四つの場合のうち、第四の場合に優先度逆転が起る。優先度継承法を採用してもこの状況は変わらない。

## 2.4 メッセージ交換の入れ子の有無

一回のメッセージ交換の途中において、別のメッセージ交換をすることを許すかどうかということである。これを許す場合は優先度逆転に関してさらに複雑なことが起りえる(5節の例)。

\*Analysis and methods for priority inversion

†Makoto Miura, Teruo Hikita

‡Department of Computer Science, Meiji University

### 3 優先度継承, 優先度上限, 優先度つきメッセージ

メッセージ交換の際に, 関係するプロセスにどのように動的に優先度を付与するかというのが課題である。

優先度継承 (priority inheritance) は, 以前から提案されている自然な方式である。プロセスの呼出しを受け付けた (同期式), あるいはメッセージを待ち行列に入れた (非同期式), 受け付け側のプロセスは, その間は, 呼出し側のプロセスないしメッセージの優先度を (それが高いとき) 持つというものである。

多段階の優先度継承は, 同期式メッセージ交換方式において考えることが妥当である。これを採用するかどうかで優先度逆転の起こる状況は異なってくる。

優先度上限 (priority ceiling) 方式はこれをさらに改良したもので, もとは共有変数方式に対して考案された [2]。あるプロセスの優先度を, 受け付け時において, そのプロセスを呼び出す可能性のあるプロセス全部の優先度の最大値とする。これは, 少しだけ優先度の高いプロセスへは実行が切り替わりにくくなるが, 優先度継承方式よりも待たされる可能性は減る。しかし適用の条件にいろいろの制限がある。

優先度は各プロセスに付与してあるのだが, 非同期通信方式の設定のもとで, 各メッセージにも優先度を付与するという方式が提案されている [1]。優先度の与え方はプログラマーの任意である。ここでも受け付け側プロセスはキューにあるメッセージの優先度を継承するものとする。この方式はプロセスの優先度を動的に随意に制御できるという利点があるが, 優先度の与え方によっては奇妙なことが起こりかねないという危険がある。

### 4 優先度逆転への対処法

メッセージ交換方式において優先度逆転を完全になくすための, 比較的単純な方法を二つ, 示す。第一のものはメッセージ送信の言語機能自体の変更を要する。第二のものはプログラマに処理を任せることになる。

#### 4.1 方法 1

メッセージ交換方式の, 先の第四の場合, どうしても優先度逆転が起こる。これを避けるには, 時間的に先にメッセージを待つ状態に入るプロセスが, そのことを, 相手のプロセスにメッセージを送って通知するとする。そのときに自分の優先度を知らせ, 相手のプロセスが (もし優先度の低い場合) 優先度を継承する。この方法は同期・非同期の両方で考えられる。

#### 4.2 方法 2

非同期式メッセージ交換において, プロセス A の優先度を  $a$ , プロセス B の優先度を  $b$  とし, A から B へ今, (非同期に) メッセージ M を送るとする。このとき, M の優先度  $m$  は, プログラマーが,  $a$  と  $b$  の間のいずれかの値を選んで指定するものとする ( $a \leq m \leq b$  または  $a \geq m \geq b$ )。そして, メッセージを受け取るプロセス B の優先度は, 優先度継承方式と同様とする。これは Elsom [1] の優先度つきメッセージの方式の枠内に入るものであるが, [1] は一般に過ぎると思われる。方法 2 の設定の理由は, (1) 優先度継承方式では高優先度のプロセスが急がない

メッセージを配布するという状況を記述できない。(2) 送る側のプロセスと受け付け側のプロセスの優先度の大きい方よりもさらに大きい値を指定するのは, 第三の高優先度のプロセスに思わぬ影響を及ぼす危険がある。(3)  $a$  と  $b$  の間に優先度を限ることで, 優先度の静的な解析が可能となり, 優先度上限方式も適用できるようになる。

### 5 例

一般のプロセスの状態を図 1 の 5 種に分ける。ブロックとサスペンドの違いは, ブロックは実行可能ではあるがスケジューラによって待ちにある状態, サスペンドは何かのイベント (たとえばメッセージの到着) を待っているため直ちには実行可能でない状態をいう。

例としてメッセージ交換の入れ子を考える。同期式において, 優先度の低いプロセスに高いプロセスがメッセージを送り, その処理途中においてさらに, 中間のプロセスのメッセージを受け付ける場合, 優先度継承方式のもとで, 新たな種類の優先度逆転がありえる。

以下の図 2 において, プロセス A, B, C, D, E の優先度はそれぞれ 5, 4, 3, 2, 1 であるとする。横軸は時間であるが, 幅は実際の時間を示すものではなく, 単に状態の推移を示す。プロセス E は, A からの呼出しを受け付ける処理の途中で (このとき E の優先度は 5), C からの呼出しを処理する。時刻 5 において優先度逆転が生じている。

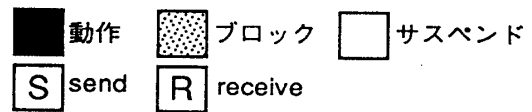


図1 プロセスの状態

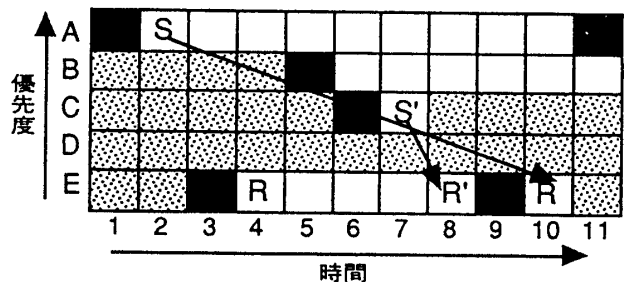


図2 メッセージの入れ子 (優先度継承)

### 参考文献

- [1] K. C. Elsom : Prioritized asynchronism in Ada9X, Proc. 4th International Workshop on Real-Time Ada Issues, in Ada Letters, vol. 10, no. 9, 1990.
- [2] L. Sha, R. Rajkumar and J. P. Lehoczky : Priority inheritance protocols: an approach to real-time synchronization, IEEE Trans. Comput., 39 (1990).