

CSP、DP、Ada の比較

石畑 清 (東京大学・理学部・情報科学科)

メッセージ交換をベースとする 3 種類の並行処理記述法を比較する。以下の議論は、おむね次の 2 点の論文にもとづいているが、一部筆者の意見を交えてある。

- J. Welsh, A. Lister, E. J. Salzman, A comparison of two notations for process communication, Lecture Notes in Computer Science, Vol. 79, pp. 225-254 (1979).
- J. Welsh, A. Lister, A comparative study of task communication in Ada, Software - Practice and Experience, Vol. 11, pp. 257-290 (1981).

1 比較の対象、方法

比較の対象とするのは次の 3 種類である。

- CSP (communicating sequential processes, (2))
- DP (distributed processes, (1))
- Ada の並行処理機能 (3)

これらは、従来の共有メモリーを念頭においた並行処理基本命令を排除し、メッセージ交換を通信および同期の手段として使うという共通点を持つ。しかし、メッセージ交換の方法や制御構造の書き方などは大きく異なっている。

ここでは、この 3 種類をプログラムの書きやすさの点から比較、検討する。比較は、CSP と DP を提案した論文 (2) および (1) にあげられている例題に対するプログラムをそれぞれの言語で書き、それに対する観察を主体として行なう。なお、上記論文ではこのほかに、こうして書いたプログラムの行数その他を数えて、プログラムの複雑さを定量的に評価しようと試みているが、この部分は省略する。

2 CSP と DP の比較

- CSP ではメッセージの受け手が送り手の名前を指定するのに対し、DP では指定しない。送り手の名前は、再交信やユーザーの資格の判定などのために必要なことが多く、このような場合 DP は著しく不便である。逆に、送り手の名前を意識する必要がない場合には便利である。
- CSP ではメッセージの受け渡しがすめば両プロセスは独立に走る。これに対し、DP では手続き本体の実行が終わるまで手続きを呼んだプロセスが束縛される。このため、両プロセスを本当に並行に走らせたい場合に余計な手間を強いられる。
- DP は複数の仕事の順序をデータを使って間接的に記述する。このため、条件の記述がプログラム中に分散して理解しづらい。実行順序をデータで記述するのはモニターと同様であるが、モニターと違って条件変数を使わず、普通の数値型や論理型を使う。このため、wait はプログラム中に明示されるが、signal は通常の代入となんら変わ

らず、よりいっそう理解を困難にしている。これに対し、CSP は制御の流れで仕事の順序を記述する。

- DP の手続きは reentrant であり、それぞれの呼び手ごとの処理が他の呼び手から独立している場合、記述が楽になる。これに対し、CSP では呼び手ごとの閉じた scope を作れないので面倒になる。
- DP はモニターのようにもっぱら他プロセスにサービスするようなケースを記述するのに適しており、他プロセスとの交信と自プロセス独自の処理が複雑にからみ合ったケースは CSP の方が容易に記述できる。

3 Adaの並行処理

- Ada の accept 文からは相手のタスク名を指定しない。タスクの資格を判定する場合は private 型を使うが、alias や共有変数などに対する配慮が必要である。
- 交信の順序は、DP と違って制御の流れで表現する。このため、accept 文の中でする仕事とその外でする仕事を使い分けることによって、呼び手を束縛する期間、ひいては並行処理の度合を容易に指定できる。一方、CSP と違ってデータの双方向の転送が可能であり、見通しの良い記述が可能である。
- 交信に関係しない制御構造には非決定的な要素は含まれていない。これに対して、CSP、DP では全制御構造を非決定的な記法で統一している。
- CSP と違って、select 文の交信を伴わない選択肢 (else 部) に条件をつけることができない。しかし、これは本質的ではなく、文を適当に入れ替えることによって等価なプログラムを書ける。
- CSP は複数の選択肢とその条件を配列のような形式でまとめて書くことを許すが、Ada にこの機能はない。このため、いくつかの entry のうち特定の部分集合だけを受け付けて、ほかは受け付けないという場合を記述しにくい。これが Ada の並行処理の最大の問題点である。

参考文献

- (1) P. Brinch Hansen, Distributed processes: a concurrent programming concept, Communications of the ACM, Vol. 21, No. 11, pp. 934-941 (1979).
- (2) C. A. R. Hoare, Communicating sequential processes, Communications of the ACM, Vol. 21, No. 8, pp. 666-677 (1979).
- (3) U. S. Department of Defense, Reference manual for the Ada programming language, 1980.