

網向きオペレーティングシステムの
プロセス間通信システムの考察

和賀井 フミ子 田中英彦 元岡達
(東京大学 工学部)

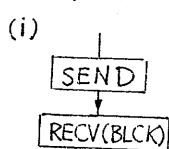
1 はじめに 私どもで、研究開発を行なっている実験用計算機網TECNETのOS(NOS)は、プロセスとそれ以外のシステム核とから成るが、そのシステム核には、次のような特徴がある。①システム核はプロセスが走る環境を作るとともに、プロセスが利用可能なプリミティブオペレーションを実行する。②プロセスごとの通信にはプリミティブを用いる。③プリミティブの一つであるプロセス間通信では、ユーザはそのプロセスの存在する計算機を意識しなくてもよい。プリミティブは、機械語の拡張であり、プロセスはこれを道具として用いている。プロセス通信の方式は、通信しあうプロセスがSEND-RECEIVEプリミティブを互いに申し合うことにより実際のデータ転送が行なわれる、という形を取る。

ここでは、NOSの利用実験で問題となったデッドロックの問題を取り上げ、その解析と解決法を示すとともに、網環境下でのプリミティブの幾つかを提案する。

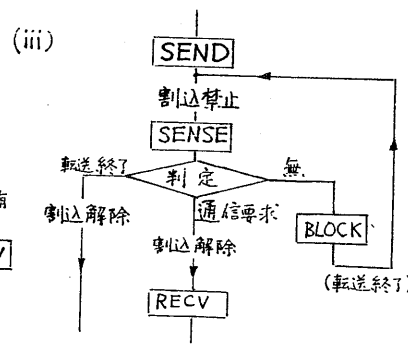
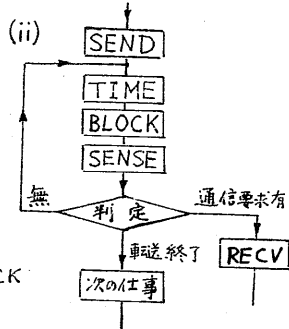
2 デッドロック 現在TECNETの要素計算機の一つであるOKITAC-4300CにはNOSが実装されているが、そこで用いているプリミティブは、SEND, RECEIVE, SENSE, SENSE2, TIME, CREATE REMOVE, START, BLOCK, STOPの10個である。このNOSを様々なに使用してみたところ、主に次の二つの問題が生じた。①通信しあうプロセスが互いにSEND-BLOCKを出し合い、ともにBlock状態になる。②プロセスごとの同期ずれ(BLOCKとSTARTプリミティブの時間ずれ)からいつまでもBlock状態が解けない。

3 状況分析及び対策 上記の問題は、単一の計算機システムでも問題になる事ではあるが、特に計算機網の場合は、プロセスの存在するシステムが互いに離れており、この種の問題が顕在化する。NOSは網向きOSであるので、相手プロセスの状態を必ずしも全面的には把握できない。ところが現在の版では、BLOCKプリミティブはプロセスを止めるのみ、STARTは単にプロセスを起動するだけで事象管理を行なっていない、これらの条件が重なった時に、上記のデッドロックが生ずる。これに対しては、次のような解決策が考えられる。

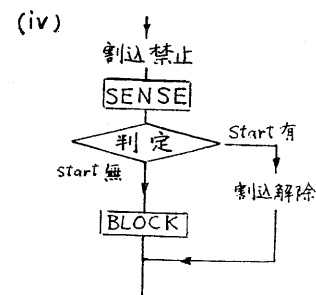
①に対して



* RECV(BLOCK)は RECEIVE and BLOCK を意味する



②に対して



(i)は、相手からSENDが出て支障のない時点でBLOCKするようにしたものであり、(ii)はTIMEプリミティブを用いて、或る時間経たら必ず起動がかかるようにしたものである。(i)も(ii)も片側のプロセスについて直せばよい。(iii)はBLOCKを行なう前にSENSEを行ない相手からの通信要求の有無を調べるもので、この場合は、両プロセスについて直す必要がある。(iv)では、プロセスをBlock状態にする前にSTARTの有無を調べる方法である。

4 結論 BLOCKもSTARTも、プロセスを止める、動かす、という基本的な機能だけを持ったプリミティブなので、単独で用いるには不十分な面もある。そこで、網向きの通信用プリミティブとしては、(i)のようにBLOCKの使用を制限して、BLOCKはRECEIVEオペレーション内のみで用いるしかたをするか、②従来のSEND-RECEIVEとTIMEを組み合わせる、または、③SENSE、BLOCKの組み合わせで事象を管理するプリミティブを新たに作る方法(従来のPOST/WAITに似たもの)等、が考えられる。

参考文献

田中・元岡 「研究用電子計算機網TECNET」 EC 73-57 (1973年12月)
 田中 他 「第16回情報処理学会一般講演 231, 232, 290」 (昭和50年)
 山内 「計算機網のオペレーティングシステム」 東大工修論(昭和52年)