循環パイプライン計算機のオペレーティングシステム　ＣＰＸ
－その概要とオブジェクトモデル－

4P-4

佐藤三久・　深沢紀博・　Paul Spee・　後藤英一・・・・・
新技術開発事業団・　理化学研究所・・　東京大学・・・

1．はじめに

本稿では、メモリ共有型マルチプロセッサ・システムのオペレーティングシステムであるＣＰＸ[1][2]の基本的な概念について述べる。我々は現在、循環パイプライン計算機(Cyclic Pipeline Computer;ＣＰＣ)ＦＬＡＴＳ２[3]を開発中である。ＣＰＣはパイプラインを複数の命令列で共有しているためメモリ共有型並列計算機として機能する。ＣＰＸはＦＬＡＴＳ２のオペレーテングシステムである。

以下２章ではＣＰＸの設計目標を、３章ではＣＰＸのオブジェクトモデルについて、さらに４章では実例としてＣＰＸのカーネル機能を使ったＸＩＮＵ[4]の実現について述べる。

2．ＣＰＸの設計目標

ＣＰＸの第一の目標は、メモリ共有型のＣＰＣにとって効率の良い、汎用の並列計算機用のオペレーティングシステムを提供することである。

つぎにオペレーティングシステムのカーネル機能を独立させ、オペレーティングシステムの大部分の機能をユーザ・プログラムとして実現可能にすることによって、様々なオペレーティングシステムの設計手法を試しその結果を検討することである。同様のアプローチは実験的な並列計算機用オペレーティングシステムであるＨＹＤＲＡ[5]でも採用されている。また、ＭＡＣＨオペレーティングシステム[6]においても、メモリ・マネージメントのページング処理をユーザ・プログラムで定義できるようになっている。

ＣＰＸでは、特権モードで実行されるカーネル部分がプロセッサやメモリ、デバイスなどのハードウエアを仮想化し、それらはオブジェクト指向のインターフェースでユーザプログラムに提供される。ユーザプログラムでは、オブジェクト・モデルにもとずいてカーネルのオブジェクトを管理し、操作する事によってオペレーティング・システムの機能を実現・拡張できる。

3．ＣＰＸのオブジェクト・モデル

あるリソースのインスタンスを抽象化したものをオブジェクトと呼ぶ。全てのオブジェクトは通信ポートを持っており、オブジェクトはそのポートによって表される。またオブジェクトはそのポートを通じて送られてくるメッセージに対して定義された操作(behavior)を実行する。その操作には大きく分けて以下の三種類がある[7]:

（１）既知のオブジェクトに対してメッセージを送る。
（２）新規のオブジェクトを生成する。（ポートをつくる）
（３）内部状態(representation)を変更する。

オブジェクト間のメッセージ通信はバッファ付きの非同期通信で行なわれる。このバッファ付きの非同期通信では全てのメッセージがバッファリングされているため、メッセージを送信した時の受け取り側の状態による遅れがなく効率がよい。また同期通信もバッファ付きの非同期通信の特殊な場合であると考えることができる。

並列性は一つのメッセージへの応答として複数のメッセージを送信することにより発生する。マルチプロセッサの場合は、これによって物理的な並列性を向上させることができる。

メッセージ通信の場合、通常のコール・オペレーションとは異なり、呼びだし側のオブジェクトが呼び出された側のオブジェクトからの応答メッセージを待つ必要はない。そのかわり、応答メッセージを継続して処理する別のオブジェクト(continuation object)が作ることができる。そのオブジェクトはcontinuation objectが応答メッセージを待っているあいだにもまた別のメッセージの処理を行なうことができる。

4．カーネルの基本機能

カーネルはポートによって、オブジェクト間のメッセージ通信の機能をあたえる。実際にはカーネルによって管理されているオブジェクトに送信されるメッセージのキューである。送信・受信操作は仮想的なハードウエアの命令である。

さらに、カーネルはメモリ空間やプロセッサ、デバイスなどハードウエアの仮想化したオブジ

An Operating System for the CPC - CPX
Mitsuhisa Sato[1], Norihiro Fukazawa[1], Paul Spee[1], Eiichi Goto[1,2]
[1] Research Development Corp. of Japan, [2] University of Tokyo

ェクトをカーネル・オブジェクトとして提供する。それらもまたポートとして表現され、メッセージ送受信によって操作する。

ＣＰＸではオペレーティングシステムの主な機能はユーザ・プログラムとして定義され、それぞれの機能を実現するプログラムはサブシステムまたはサーバと呼ばれる。

サーバは、仮想空間と複数のプロセッサによって実行されるプロセスである。サーバで定義されたオブジェクトは、サーバ内のオブジェクトに対応するポートによって表現される。サーバー内のプロセッサはオブジェクトに対応したポートに送られたメッセージに対して、処理を行なう。

カーネルは、基本的にハードウエアの仮想化をするのみであり、それよりも上位の概念である”プロセス”や”ファイル”は、ユーザプロセスであるサーバによるオブジェクトとして定義される。これによって、ユーザがシステムの基本的な機能を拡張できる。

５．ＣＰＸカーネル機能を用いたＸＩＮＵの実現

ＣＰＸカーネル機能をつかってオペレーティングシステムＸＩＮＵのＣＰＸカーネル上への実現を試みた。ＸＩＮＵの実現の方針は；

（１）プロセスごとに仮想計算機を割り付ける。

（２）ＸＩＮＵに対する変更を少なくするために、システム・コールは通常のコール命令で各々のプロセスが直接行なうような形にした。

（３）ＣＰＸではデバイスに対する入出力は割り込みではなく、メッセージ通信によっておこなわれる。したがって、デバイスからのメッセージをうけとるプロセスをつくって、デバイス・ハンドラを実行するようにした。

（４）仮想化されたプロセッサでは、相互排除はプロセッサ・ステータス(割り込み禁止)によって行なえないため、ＬＡＳ(load and store)命令によるスピン・ロックとセマフォ(Ｐ・Ｖオペレーション)によって行なう。

マルチプロセッサでは、複数の仮想プロセッサは物理的に同時に動く。ＸＩＮＵはもともとシングル・プロセッサ用のオペレーティング・システムであるため相互排除は割り込み禁止でおこなっていたがこれはもはや使えない。シングル・プロセッサでの割り込み禁止と同じ機能を実現するために、プロセスごとの情報をユーザプログラムの領域に格納し、プロセッサの管理を行なった。

前に述べたとおり仮想プロセッサは、ユーザに対してはポートとして表現されているため、テーブルとして管理しにくい。よって、これを容易にするために、ユーザの管理する情報（例えば、プロセスのidなど）をオブジェクトに格納しておくための機能(get client/put client)を加えた。これによって、ハードウエアのリソースをユーザプログラムのマネジャで効率的に管理することができる。

実現に当たっては、ＦＬＡＴＳ２のシミュレータ上にカーネル機能を組み込みそれを用いた。その結果ＸＩＮＵの全ての機能がＣＰＸカーネル上でユーザ・プログラムとしてオペレーティングシステムを構築できることが確認できた。

６．まとめ

マルチプロセッサ向けのオペレーティングシステムであるＣＰＸの基本的な概念であるオブジェクトモデルについて報告した。

またＸＩＮＵを用い、ＣＰＸカーネル機能の検証を行ない、ＣＰＸカーネル上でユーザ・プログラムとして基本的なオペレーティングシステムを構築できることを確認した。

今後はＣＰＸ本体の機能の実現とＵＮＩＸ*等更に高機能なオペレーテング・システムの構築を検討していく予定である。

*UNIX is a trademark of AT&T Bell Laboratories.

参考文献

[1] Sato,Spee,Fukazawa,and Goto. "CPX:Exploiting Concurrency on the CPC" Proceedings of the 6th Riken Symposium on Josephson Electronics, March 1989.

[2] Spee,Sato,Fukazawa,and Goto. "CPX:An Operating System Kernel for the CPC" Proceedings of the 6th Riken Symposium on Josephson Electronics, March 1989.

[3] Ichikawa:A study on a Cyclic Pipeline Computer: FLATS2, 修士論文、東京大学理学部情報科学科、1987

[4] Comer,D,"Operating System Design, the XINU Approach",prentice-hall

[5] Wulf, W.A. and Harbison,S.P., "HYDRA/C.mmp - An experimental Computer System", McGraw Hill, 1981.

[6] Young,M., Tevanian,A., Rashid,R., Golub,D., Eppinger,J., Chew,J. Bolosky, W Black,D. and Baron, R. "The Duality of Memory and Communication in the Implement-ation of a Multiprocessor operating system", Proceedings of the 11th ACM Symposium on Operating System Principles, Austin, Texas, November 1987

[7] Agha,G."An Overview of Actor Language",SIGPLAN NOTICE, Volume21, Number 10 1986