4N-8

ＯＳ／omicron 第２版におけるタスク管理

鈴木茂夫，並木美太郎，高橋延匡
東京農工大学　工学部　数理情報工学科

１．はじめに

ＯＳ／omicron は，日本語処理を標準とした研究用オペレーティングシステムである．本稿では，このＯＳ／ｏ第２版におけるタスク管理について述べる．ＯＳ／ｏは，研究者がアプリケーションを含めた専用計算機システムを構築する際に，自由な形で利用できるアプリケーションオリエンテッドなＯＳとなることを目的としている．したがって，ＯＳの機能拡張，改造を自由に行なえるようにし，またタスクの実行スケジューリングに関する操作も全面的にユーザに開放する方針である．

ＯＳ／ｏでは，ユーザプログラムはもちろん，割込み処理プログラム，エラー処理プログラム，そしてＯＳ自身にいたるすべての処理プログラムをタスクという単位で統一的に管理する．そして，基本同期手段としてＰＶセマフォを採用しマルチタスク機能を実現している．

２．ＯＳ／ｏのシステム構成

ＯＳ／ｏでは，ＯＳの動的な機能拡張，およびデバイスハンドラの動的な登録機構を実現している．これらＯＳの拡張部分をユーザ拡張部と呼ぶ．ユーザアプリケーションは，ＯＳの核の呼び出しと同様の方法で，ユーザ拡張部の機能を呼び出すことができる．したがって，ＯＳ／ｏは，ファイル管理，タスク管理，メモリ管理を行なうＯＳの核，ユーザ拡張部層，およびユーザアプリケーション層の３層構成となる．

ユーザ拡張部の登録機構を利用して，現在，かな漢字変換入力機能，ウィンドウシステムなどの研究，開発が行なわれている．ユーザ拡張部に登録した機能は，動的に交換，削除が可能なことから，こうしたマンマシンインタフェース，およびデバイスインタフェースの研究が円滑に行なわれることが期待できる．

３．タスクとタスクフォース

ＯＳ／ｏでは，前述したようにすべての処理プログラムをタスクという単位で統一的に管理している．ＯＳの核，およびユーザ拡張部も同様である．タスクは，その動作環境として，プロシジャ，静的データ，ヒープ，スタックの４つの領域を主記憶上に持つ．これらの領域のうち，プロシジャ，静的データ，ヒープ領域を共有したタスク群の形態をタスクフォースと呼ぶ．タスクフォースの実現は，密な関係にあるタスク間での，柔軟でかつ高速な情報交換を可能とすることが主目的である．ユーザアプリケーション層において，タスクフォース，そしてタスクは，ともに動的に生成可能であり，その生成関係は木構造の親子関係となる．

ＯＳの核，およびユーザ拡張部はタスクフォースの形態で実現されている．ＯＳの核，ユーザ拡張部は，ユーザアプリケーション層の各タスクから同時に呼び出されることを考えなければならない．したがって，プロシジャ，静的データ，ヒープ領域についてはそれぞれ１つの領域を共有し，タスクとしての実行環境が，ＯＳの核，ユーザ拡張部を利用できるタスクの数だけ存在する一種のタスクフォースとなっている．

４．タスク間同期・通信手段

4.1 ＰＶセマフォ

前述したように，タスクフォースでは，静的データ領域を共有している．したがって，タスクフォース内のタスク間では，この領域を利用して自由なデータ交換を行なうことができる．この際の同期，排他制御手段として，ＯＳ／ｏでは最もプリミティブな同期命令であるＰＶセマフォをユーザプログラムに提供する．

ユーザは，共有データ及びセマフォを直接利用して，タスク間通信を行なうことができる．また，これら原始的な通信手段をもとに，より抽象度の高い通信方式，例えばメールボックス型のような通信方式をライブラリ等で実現し，それを用いてプログラミングすることも可能である．

4.2 メッセージ通信

同一のタスクフォースに属さないタスク間では，共有のデータ領域を持たないため，上記の方式でタスク間通信を行なうことはできない．そこで，タスクフォース間の通信手段として，同期式のメッセージ通信機能を実現した（図１参照）．メッセージの経路はタスクフォースごとに複数生成可能であり，そのとき得られるメッセージ識別子をもって任意長のデータの送受信を行なう．

メッセージ通信方式は，共有データによる通信と比較して，記述が容易という特徴を持つ反面，機能に制限が

Task Management of OS/omicron V2.0

Shigeo SUZUKI, Mitarou NAMIKI and Nobumasa TAKAHASHI
Tokyo University of Agriculture and Technology

あり，効率が劣るといった欠点を持つ．したがって，大量のデータ交換を必要とするプログラムの場合には，タスクフォース形態で実現することが望ましく，メッセージ通信機能は，データ交換の少ない粗な関係のタスク間において有効である．

4.3 ポインタ通信

通常のタスクフォース間では，データを共有することができない．しかし，親子関係にあるタスクフォース間で，しかも親タスクフォースがそれを許可した場合にのみ，データの共有を許すことにした．共有するデータ領域は，親タスクフォースの静的データ領域，または子タスクフォースの静的データ領域であり，その指定は子タスクフォース生成時に親タスクフォースが行なう．このとき同時に，親子タスクフォース間で共有するセマフォの指定を行なう．データの共有を指定した場合には，親タスクフォースと子タスクフォースのポインタの基底アドレスが共通になる．したがって，共有データ領域を指すポインタ値を，メッセージ通信を利用して通信相手に送ることにより，共有データとセマフォを用いた高速で柔軟なデータ交換が可能となる（図２参照）．

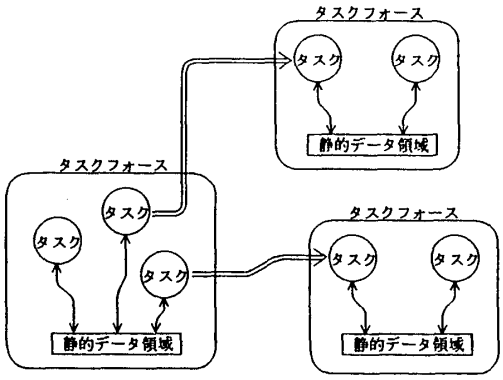

図１　タスク間通信
⟷ ＰＶセマフォを用いて直接アクセス
⟸ メッセージ通信

５．おわりに

ＯＳ／ｏは，現在第２版の実現を完了し，研究室内で実働中である．これにより，次に述べるような成果を得た．
(1) ＯＳの機能拡張，およびデバイスハンドラの登録を動的に行なう機構を実現した．(2) タスクフォースにより，高速で柔軟なタスク間通信が実現可能となった．(3) これらの成果により，種々のデバイスを用いたマンマシンインタフェースの研究，および並列処理言語などの研究の基盤となる環境を提供できた．

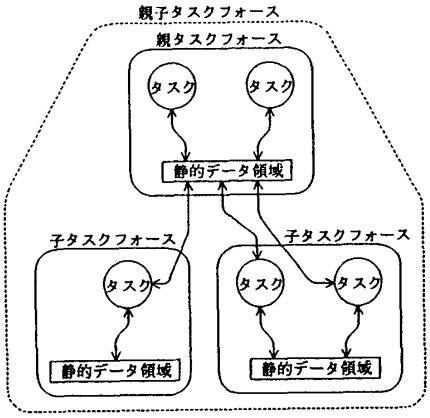

図２　親子タスクフォース間でのポインタ通信
親タスクフォースの静的データ領域を共有領域とした形態
⟷ ＰＶセマフォを用いて直接アクセス

参考文献

[1] 高橋延匡，“研究プロジェクト総説：ＯＳ／omicron の開発”，オペレーティング・システム研究会，39-5，昭和63年６月
[2] 鈴木茂夫，他，“ＯＳ／omicron における日本語プログラミング環境”，「コンピュータシステム」シンポジウム，昭和62年11月

表１　主なタスク管理ＳＶＣの仕様

関数名	引数	戻り値	機能
_create_tf	・モジュール名 ・生成情報パケット	子タスクフォース識別子	子タスクフォースを生成する
_go_tf	・子タスクフォース識別子	エラーコード	子タスクフォースを実行させる
_stop_tf	・親タスクフォースへの戻り値	エラーコード	自タスクフォースの実行を停止する
_exit_tf	・親タスクフォースへの戻り値	エラーコード	自タスクフォースの実行を終了する
_wait_tf	・子タスクフォース識別子 ・子タスクフォース情報パケット	子タスクフォース識別子	子タスクフォースの停止、終了を待つ
_kill_tf	・子タスクフォース識別子	エラーコード	子タスクフォースを削除する
_create_task	・エントリポイント ・スタックサイズ ・タスクの種類	子タスク識別子	子タスクを生成する
_go_task	・子タスク識別子	エラーコード	子タスクを実行させる
_stop_task	・親タスクへの戻り値	エラーコード	自タスクの実行を停止する
_exit_task	・親タスクへの戻り値	エラーコード	自タスクの実行を終了する
_wait_task	・子タスク識別子 ・子タスク情報パケット	子タスク識別子	子タスクの停止、終了を待つ
_kill_task	・子タスク識別子	エラーコード	子タスクを削除する
_create_sem	・セマフォの初期値	セマフォ識別子	セマフォを生成する
_delete_sem	・セマフォ識別子	エラーコード	セマフォを削除する
_P_op	・セマフォ識別子	エラーコード	P命令を実行する
_V_op	・セマフォ識別子	エラーコード	V命令を実行する
_create_mssg	・通信相手のタスクフォース識別子 ・通信モード(RECEIVE/SEND)	メッセージ識別子	メッセージ通信経路を生成する
_delete_mssg	・メッセージ識別子	エラーコード	メッセージ通信経路を削除する
_receive_mssg	・メッセージ識別子 ・バッファを差すポインタ ・バッファのサイズ	受信したメッセージサイズ	メッセージを受信する
_send_mssg	・メッセージ識別子 ・バッファを差すポインタ ・バッファのサイズ	送信したメッセージサイズ	メッセージを送信する
_set_ue_task	・モジュール名 ・ユーザ拡張部番号	エラーコード	ユーザ拡張部の登録を行なう