

2次元アドレスを管理する核OSの開発

4F-9

森永智之, 中村浩之, 早川栄一, 並木美太郎, 高橋延匡  
(東京農工大学 工学部 電子情報工学科)

1. はじめに

最近の計算機の用途の多様化によって, 計算機に接続される外部デバイス, 計算機の扱う資源が多様化している. 最近では, 従来にないデバイスが接続され, 音声や画像といった様々な資源を計算機で管理する局面がある. このために, オペレーティングシステム(OS)でも目的に応じた資源管理を行う必要がある.

したがって, 目的に応じた様々なOSを容易に構築することが必要であると考え, そこで我々は,

- ・既存のOSを2層に分け, 下の層(以下, 核OS)をOS開発の共通基盤として利用し, 上の層(以下, OS層)で目的別OSを実現する
- ・OS層でダイナミックリンク[1][2]の機構を採用し, これを用いてOSを構成することで目的別OSの生成を容易にする
- ・OS層での資源管理と保護の容易さから, セグメンテーションを核OSで提供する

という方法をとることにした.

核OSの上にダイナミックリンクを用いてOSを構成することで, 様々な目的別OSを容易に構築することが本研究の目的である. このためには, OSの構成要素に対して論理的な単位を設定し, 保護を行うことができる2次元アドレスを提供することが有効である. 本稿では, 主として核OSの行うメモリ管理について述べる.

2. 核OSの設計

様々な目的別OSを実現するための共通基盤と, その動作環境を提供するために, 核OSでは次のような設計方針をとった.

(1) OSをタスクとして仮想化する

核OSは, OSをタスクとして管理し, タスクに対してプロセスを仮想化して提供することにする.

(2) 2次元アドレスを提供する

核OSでは, OSに対して2次元アドレスを提供することにする.

(3) 核OSを保護する

本システムは, ユーザ, OS, 核OSという3層から構成される. そこで, リングプロテクションを用いて, それぞれの層の間の保護を行う.

(4) 外部デバイスの仮想化は行わない

核OSではごく基本的な, メモリ, 割込み, プロセッサといった資源について管理を行う. そして, これ以外の資源に

ついては, 核OSでは管理せず, OS層で管理を行うことにする.

次に, 核OSで管理する資源について述べる. 核OSでは, OSをタスクとして管理する. また, 主記憶を仮想化して, セグメントという形でタスクに提供する. タスクは, 独自の割込みハンドラを設定可能である. タスクが物理ページを確保する場合に, OSごとに排他制御を行うために, 核OSではセマフォを提供することにした.

また, 核OSが管理しない資源としては, 外部デバイス, 物理ページがある. デバイスドライバは, OS層に置かれ, タスクがダイナミックリンクの機構を用いて呼び出す. 物理ページに関しても同様である. これらを表1に示す.

表1 核OSの管理する/しない資源

核OSの管理する資源	核OSの管理しない資源
タスク	外部デバイス
セグメント	物理ページ
割込み	
セマフォ	

ダイナミックリンクの実行環境では, タスクは複数のセグメントをもつ(図1). しかし, これらの管理は核OSでは行わない. セグメントのリンク状況, 共有状況, タスク消去時のセグメントの解放などについては, OS層にあるページクファイルシステム(bfs)で管理する.

3. システムの全体構成

このシステムは, 次のような3つの層から構成される. それぞれの層について説明を示す.

(1) 核OS層: 核OSのセグメントが置かれる. 核OSの管理する資源管理のための表もここに置かれ, 他の層から保護する.

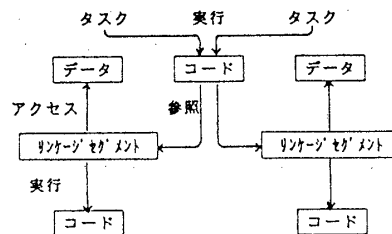


図1 タスクとセグメントの関係

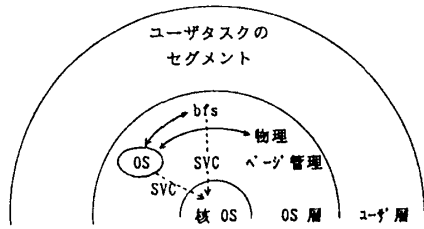


図 2 システムの全体構成

(2) OS 層： 核 OS が管理するタスクの層であり、bfs、デバイスドライバ、OS 自身の他に、物理ページ管理などのセグメントがおかれる。これらはダイナミックリンクでリンクされることによって利用される。

(3) ユーザ層： OS の管理するユーザータスクのセグメントが置かれる層である。

核 OS を含めたシステムの全体構成を図 2 に示す。

4. メモリ管理の設計方針

核 OS では、2 次元アドレス空間を提供する。しかし、核 OS がページングなどについても管理を行うと、OS 層で自由な OS を実現できない。また、OS 間の保護を実現し、安全性を保証することも大切である。このことから、核 OS では、次のようなメモリ管理の方針をとった。

(1) OS が自由にページングを行えるようにする。

ページングを行うか、また、どのようなページングアルゴリズムを採用するかなどについては、OS で自由に決定できるようにする。

(2) 核 OS でセグメントの管理を行う。

核 OS では、セグメントの管理を行う。これによって、OS 間の共有セグメントなどを実現することにする。

5. メモリ管理

5.1 セグメンテーション

核 OS では、OS ごとの保護を行うために、それぞれの OS に固有のセグメントと、全システムでグローバルなセグメントを提供することにする。これによって、OS 間でセグメントを共有し、デバイスドライバなどを共有することが可能になる。核 OS が提供するメモリ空間を図 3 に示す。

核 OS では、OS に対してセグメントの生成/消去といった SVC を提供し、OS に対してメモリをセグメントという形で仮

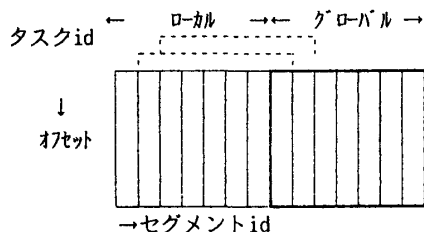


図 3 核 OS が提供するメモリ空間

想化して提供する。さらに、OS に対してはセグメント id を提供する。セグメントの定義情報を保持するテーブルなどは、核 OS で保護を行う。

つまり、OS は、概念的には次のような SVC を用いてセグメントを生成することができる。

セグメントid = SVC\_セグメントを生成する(保護属性, 大きさ)

しかし、この SVC によって、セグメント物理ページが確保されるわけではない。物理ページは、タスクが物理ページ管理部をダイナミックリンクすることによって確保することになる。

5.2 ページング

核 OS はページテーブルをあらかじめ確保しておき、SVC を用いてページテーブルをアクセスさせる。これによって、ページテーブルに書込まれる値を核 OS がチェックすることができ、安全性を確保できる。

核 OS では、一般に多くみられる 2 段階のアドレス変換機構のインタフェースをとらずに、1 段階のページテーブルとして仮想化する方法をとった。つまり、概念的には次のような SVC を用いる。

SVC\_ページテーブルアクセス(タスクid, セグメントid, インデクス)

これによって、ページテーブルを 1 ページの大きさに制限せず、セグメントの大きさに応じた可変長のテーブルとして仮想化することができ、さらに、マシンアーキテクチャに依存しにくい、簡潔なインタフェースを提供することができる考えた。

このように、ページテーブルのアクセスに SVC を用いる方法では、必然的にオーバーヘッドは増加することが考えられる。これについては、OS のページングの頻度にも依存することが考えられ、実際に性能測定を行うことを考えている。

6. 最後に

本稿では、2 次元アドレスを管理する核 OS のメモリ管理について述べた。現在、インテルの 80386 プロセッサを採用したマシン上で実現を行っている。核 OS が完成した段階で、我々の研究室で開発している OS/omicron[3] の第 3 版を実現するつもりである。

参考文献

[1] Sigeru Motobayasi, Takasi Masuda and Nobumasa Takahashi: "THE HITAC 5020 TIME SHARING SYSTEM", 'Proc. ACM 24th, Nat. conf. 24, 419-429

[2] E.I.Organick, 菊池豊彦, 佐々木彬夫: "MULTICS システム (上) (下)", 共立出版, 1973

[3] 高橋延匡: "研究プロジェクト総説 OS/omicron の開発" 情報処理学会オペレーティングシステム研究会報告 39-5, 1988