3P-3 共有ライブラリを用いたUNIXカーネルの動的変更に関する一考察

°松田 晃一, 川又 滋, 有野 康仁, 高橋 久
(日本電気株式会社 C&C共通ソフトウェア開発本部 基本システム開発部)

1. はじめに

UNIXは静的なOSである. すなわち, OSに変更があると, そのたびごとに再コンパイル, 再立ち上げが必要である. これは, 特にUNIXを新規のCPUを用いたシステムに移植する場合には, かなり重要な問題になる. たとえば, UNIXの移植作業の多くの部分を占めるI/O装置等の周辺装置の移植のためのドライバの開発は, 一般に, OSが安定してサービスが提供できるまで頻繁に変更が行われる.

本論文では, I/Oドライバ等の既にインタフェースの決っている部分(デバイススイッチ以下の部分)を, OSそのものの再コンパイル, 再立ち上げすることなしに変更するメカニズムについて述べる.

これは, 既存の機能を用いて, OSそのものの改造を最小限にして, 限定された範囲であるがOS内部の機能の動的な変更を可能にする方法である.

ここで用いる基本的なメカニズムは, 共有ライブラリと共有メモリであり, 対象とするUNIXは, SystemV R3.0以降のversionである.

2. 共有メモリと共有ライブラリ

基本となる共有メモリと共有ライブラリの機能について簡単に説明する.

- 共有メモリ
  アプリケーションの制御で, 特定のアドレスにメモリをアタッチすることができる. このメカニズムによって, アプリケーションから動的にメモリの割り当てを行うことができる.
- 共有ライブラリ
  共有メモリのメカニズムによって共有ライブラリを用いたアプリケーションは, ライブラリを変更があっても再コンパイルすることなしに実行することができる.

3. 基本的なアプローチ

基本的なアプローチとしては, デバイススイッチ以下のドライバ(図中odd)を共有ライブラリとして登録し, それを用いてアドレス解決を行ったUNIXカーネルをコンパイルする. ドライバ自体は, 後から共有メモリを用いて(図中attachment tool), OS空間内に置く(アタッチする).

ドライバを変更した場合は, 変更したドライバ(図中ndd)を新たに共有ライブラリとして, OSのアドレス空間にアタッチする. 変更したドライバで使用しているOS内の関数等とのアドレス解決はOSのシンボルテーブルを用いて行う.

こうすることで, デバイスドライバ自体の変更があっても, OS自体を再コンパイル, 再立ち上げすることなしに動的な変更を行うことができる.

このアプローチの全体の流れを図1に示す.

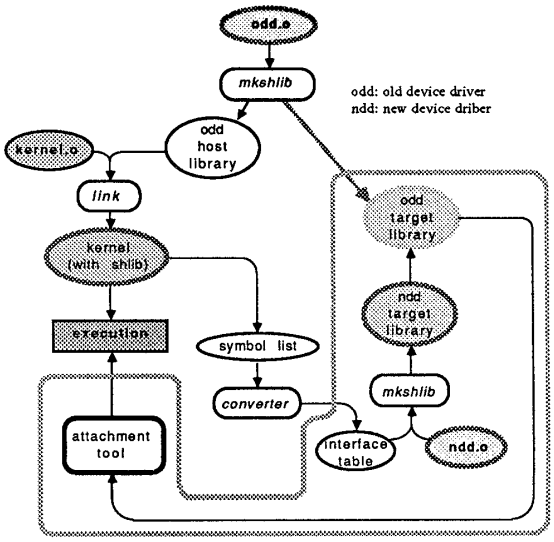

図1 UNIXカーネル作成の流れ

4. 実現に必要な機能

基本的に以下の新規機能が必要である.

(1) 実行可能な共有メモリが取れる.

(2) 共有メモリをOSのアドレス空間上にアタッチできる.

(3) OSと新しく変更したドライバ(ndd)からOS内のシンボルが参照できる.
  OSからドライバへの参照は共有ライブラリで解決できるが, ndd内で用いているOS内の関数等の参照は別の方法で解決しなくてはならない. これは, OSのシンボルテーブルをアドレス解決が図れるような形に変換することで解決する.

(4) 実行可能な共有メモリをページングの対象から外す.

(5) 実行可能な共有メモリに物理的に連続なメモリが割付けられる.

5. まとめ

共有メモリと共有ライブラリを用いた動的なOSの機能変更について述べた. 現在, このようなメカニズムを持ったUNIXを当本部で開発中のUOV(System V + 4.3BSDの機能の一部)上で開発中である.

6. 参考文献

- UNIX System V プログラマ・ガイドリリース 3.1

A study for dynamic modification of UNIX kernel using shared library

Kouichi Matsuda, Shigeru Kawamata, Yasuhito Arino, Hisashi Takahashi
C&C Common Software Development Laboratory, NEC corporation