

# 1 G-2 ダイナミックリンク機能における高速プロセス生成方式

谷口 秀夫      箱守 聡

NTTデータ通信(株) 開発本部

## 1. はじめに

ダイナミックリンク機能は、プロセス間でのプログラムファイル共用や、プログラムの部分的変更の容易さ等の特徴を持つ。このため、大型計算機では古くから実現されており、最近、いくつかのワークステーションでも実現されている。しかし、実行形式プログラムが多くのファイルからなるため、プログラムのメモリ常駐を必要とする場合にはプロセス生成時間が遅いことが問題になる。

本稿では、大きなプロセスを多数生成する場合について、その処理の効率化を図る方式を述べる。

## 2. ダイナミックリンク機能

### 2.1 機能概要

ダイナミックリンク機能とは、応用プログラム(以降、APと略す)の実行形式プログラム(以降、ロードモジュールと呼ぶ)を分割し、プログラム実行時に関数呼出しや全体変数参照のアドレスを解決する機能である。したがって、APのロードモジュールは複数のファイル(以降、ロードセグメントと呼ぶ)からなる。

これに対し、関数呼出しや全体変数参照のアドレス解決を、プログラム実行以前つまりコンパイル時に行なう機能をスタティックリンク機能と呼んでいる。この場合、ロードモジュールは一つのファイルになる。

### 2.2 特徴

スタティックリンク機能と比べたダイナミックリンク機能の特徴を以下に示す。

〈長所〉

#### (1) ロードセグメントの共用

ロードセグメントをロードするメモリのアドレスを自由に変更できる。そのため、一つのロードセグメントを複数のAPプロセス間で共用できる。

#### (2) モジュール化

ロードセグメントの単位を意識したプログラム開発により、プログラムのモジュール化が進み易い。さらに、ロードセグメントの共用も、この傾向に拍車をかける。

#### (3) 部分的変更処理の容易化

APの機能追加や変更は、ロードセグメントの追加や対応する一部のロードセグメント改版でできる。そのため、実際にその機能をインストールする計算機上の処理が、容易になる。

〈短所〉

#### (A) プロセッサ負荷の増加

関数呼出しや全体変数参照がテーブルを介した形になるため、実行処理が重くなる。

#### (B) 分割損

一つのロードモジュールを複数のロードセグメントに分割するため、分割損がメモリや外部記憶装置に発生する。

#### (C) 制御の複雑化

関数呼出し等のアドレス解決をプログラム実行時に行なうため、オペレーティングシステム(以降、OSと略す)の処理が複雑になる。

## 3. プロセス生成処理

### 3.1 処理概要

ダイナミックリンク機能におけるプロセス生成の概要処理フローは、次のようになる。

- ① APの中で最初に起動されるロードセグメントを外部記憶装置からメモリにロードする。
  - ② APが必要とするロードセグメントを外部記憶装置からメモリにロードする。
  - ③ 関数呼出しや全体変数参照のアドレスを解決する。
  - ④ APに制御を移行する。
- 処理②③は、それを行なう契機により、各々二つの方式がある。一つは、プログラム生成時に行なう方

式である。もう一つは、プログラム走行時に必要となった時点で必要分だけ行なう方式である。

### 3. 2 要求と問題点

端末制御装置のように、APの大きさが大きく、同じAP機能のプロセスを多数生成する場合には、次の問題がある。

(問題1) 多くのロードセグメントを外部記憶装置からメモリにロードするため、外部記憶装置へのアクセスが多発する。

(問題2) 関数呼出しや全体変数参照のアドレスを解決する処理が多い。

したがって、効率的なプロセス生成の処理が必要になる。

### 3. 3 対処方式

#### 3. 3. 1 メモリロードセグメント

(問題1)を解決するため、ロードセグメントをメモリ上に置く方式を提案する。これを、メモリロードセグメントと呼ぶ。この方式は、3. 1節で述べた処理②③の実行契機の各々について適用できる。ここでは、処理②③をプログラム生成時に行なう場合を例に説明する。プロセス生成の処理フローを図1に示し、以下に説明する。

- ① APの中で最初に起動されるロードセグメントを外部記憶装置からメモリにロードする。
- ② APが必要とするロードセグメントがメモリロードセグメントに登録されているか調べる。
- ③ 登録されていないならば、外部記憶装置から当該ロードセグメントを読み出し、メモリロードセグメントとして登録する。
- ④ メモリロードセグメントの内容を複製してロード処理を行なう。
- ⑤ 関数呼出しや全体変数参照のアドレスを解決する。
- ⑥ APに制御を移行する。

上記の処理②③④は、APが必要とするロードセグメント分だけ繰り返し行なう。

メモリロードセグメントにより、最初のプロセス生成時間は長いものの、二つ目以降のプロセス生成時間は短い。これは、プロセス間で共用するロードセグメントについては、外部記憶装置からの読み出しが一度しか発生しないためである。つまり、装置におけるロードセグメントの読み出し回数は、高々その装置にあるロードセグメントの数である。

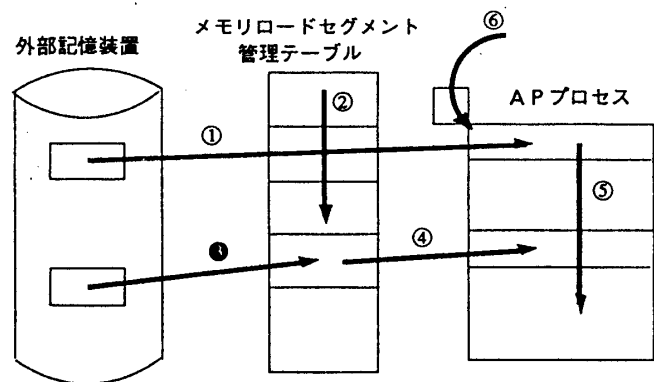


図1 メモリロードセグメント方式

#### 3. 3. 2 マルチプロセス生成機能の利用

著者らは、文献[1]で提案したマルチプロセス生成機能を利用し、(問題2)の解決を図った。マルチプロセス生成機能とは、多数のプロセスを一度に生成できる機能である。

マルチプロセス生成機能を利用したプロセス生成の概要処理フローを以下に示す。

- ① プロセス生成情報の通知
  - ② 外部記憶装置からのメモリへのロード
  - ③ 関数呼出しや全体変数参照のアドレス解決
  - ④ プロセス生成情報に基づき、一つのプロセスを複製して要求分のプロセスを生成
  - ⑤ APに制御移行
- 複数のプロセス生成処理④の前にアドレス解決処理③を行なうことにより、アドレスを解決する処理の量は、生成するプロセス数に関係なく一定である。つまり、プロセス生成時に一度でよい。

#### 4. おわりに

ダイナミックリンク機能を提供するOSにおいて、多数の大きなプロセスを高速に生成する方式を述べた。

ロードセグメントの原本をメモリ上に置く「メモリロードセグメント」により、外部記憶装置へのアクセス回数を削減できる。さらに、マルチプロセス生成機能の利用により、アドレス解決の処理量を抑えることができる。

今後は、本方式を実現し、評価する予定である。

#### <参考文献>

- [1] 谷口, 境: プロセス生成の高速化と並列化に関する検討, 信学技報, CPSY89-28.