80386マイクロプロセッサにおけるリンケージフォールト処理の一方式

4F−10

中村浩之，田中広幸，森永智之，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

近年，計算機の利用形態は多様化し，目的に合わせてデバイスを追加したりハードウェア構成を変更して利用するようになってきている．計算機システムの専用化に合わせて，オペレーティングシステム（OS）にも多様な計算機資源を管理することが望まれている．

このような要求に答えるためには，OS には機能の拡張性や構成の柔軟性が必要である．我々の研究室では，目的に応じた機能をダイナミックリンクを用いて追加し，構成するOS を作成している[1]．しかし，最近のマイクロプロセッサ（CPU）はダイナミックリンクの実現機構を持たないので，リンケージフォールト機能は持っていない．

本稿では，ダイナミックリンク機能を持たない CPU でリンケージフォールト処理を行う方法を述べる．さらに，80386 の保護機能を使ったリンケージフォールト処理についても述べる．

2．ダイナミックリンク機能

ダイナミックリンク環境では，モジュール間のリンクはリンクテーブルを使って行われる．モジュール間にまたがる手続き呼出しやデータ参照などは，図１のように，リンクテーブルを介した間接アクセスによって行われる[2][3]．

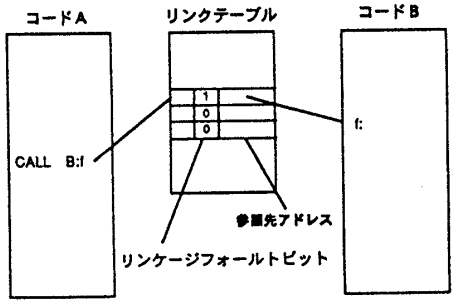

図１：ダイナミックリンクによる手続き呼出し

Multics の GE645 のようにダイナミックリンクの実現機構を持った計算機では，リンクテーブル中にリンケージフォールトビットを持っている．リンクテーブルの参照時にこのビットがリンク未確定を示していれば，リンケージフォールトが発生して，OS 側に制御を渡すことができる．

しかし，80386 をはじめとする現在の CPU のメモリ管理は，ページフォールト等の機能は有するが，リンケージフォールト機能を持っていない．これらの CPU 上でダイナミックリンクを行うためには，リンケージフォールトの代わりとなる方法を考えなければならない．

以下に，我々の考案したリンケージフォールト処理の方式を示す．

3．リンケージフォールト処理の方式

3．1　ページフォールトの利用

現在 CPU のほとんどはページフォールトを持っているので，まず，これを利用したリンケージフォールト処理を考える．

CPU の不在ページエントリ中にはユーザ（OS 開発者）に提供された領域があり，ユーザはここを自由に使うことができる．ここにリンクテーブルの構成ページであることを示す情報を入れておけば，発生したページフォールトがリンケージフォールトかどうかを判別できる．リンク確定後はこの領域をページスワップ用の情報として利用する．

ページフォールトを使ったリンケージフォールト処理の流れは次のようになる．

（１）初期段階ではリンクテーブルを構成するページをすべて未割当てとしておく．

（２）変数参照や関数呼出しのためにリンクテーブルに始めてアクセスすると，ページフォールトが発生する．

（３）例外処理部ではページフォールトの発生したページが未確定リンクテーブルのものかどうかを判別する．

（４）リンクテーブルのページであればリンクの確定処理を行った上で，ページを割り当てる．この際，同一ページ中にある外部名のリンクはすべて解決する．

（５）ページフォールトから復帰すると正確なアドレスが書き込まれたリンクテーブルが割り当てられている．

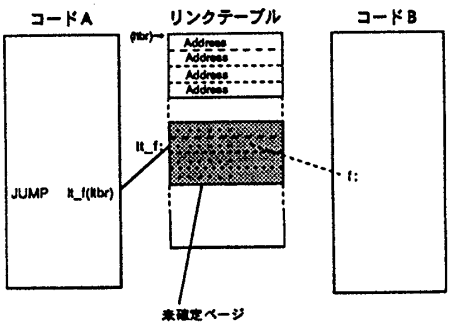

図２：ページフォールトを使った処理

この方式は，リンケージフォールト機能を持たない CPU アーキテクチャでもダイナミックリンクを実現できる．ただし，ページ単位でリンクを解決するので，リンクテーブルをページで整合し，各ページもリンクテーブルのレコードで整合しておく必要がある．

A Method of Linkage Fault Processing for the 80386 Micro Processer.

Hiroyuki NAKAMURA, Hiroyuki TANAKA, Tomoyuki MORINAGA, Eiichi HAYAKAWA, Mitarou NAMIKI and Nobumasa TAKAHASHI

Tokyo University of Agriculture and Technology

3．2　80386 不在セグメント例外の利用

上の方式では一回のフォールトでページ中の名前をすべて解決しなければならない．このため，同一ページ中に一度も参照されない外部名があるとそれらも一緒にリンク処理が行われてしまう．外部名一つごとにフォールトが発生すれば，このようなことはない．

80386 では，不在セグメント例外を利用すれば，外部名ごとのフォールトを発生させることができる．それは，次のような流れの処理になる．

（１）あらかじめ決まったセレクタをリンケージフォールト用と定め，リンクテーブルのアドレス格納領域を，そのセレクタと適当なオフセット値で初期化しておく．セレクタの示すセグメントのデスクリプタは不在属性としておく．

（２）リンクが未確定のときにこの領域からアドレスを取り出すと，命令実行中に不在セグメント例外が発生し，例外処理ハンドラに制御が移る．このとき，例外処理ハンドラには例外を発生したセレクタの値が渡される．

（３）例外処理ハンドラは渡されたセレクタからこの例外がリンケージフォールトによるものなのかどうかを判定する．

（４）リンケージフォールト用のセレクタであれば，リンク処理を行い，リンクテーブルに正しい二次元アドレスを書き込む．別のセレクタであれば不在セグメント例外と判定し，適切な処理を行う．

（５）処理が終了し，例外を起こしたプログラムに復帰すると，例外を起こした命令を最初から再実行する．今度は正しいアドレスがリンクテーブルに書き込まれているため，命令は正しく実行される．

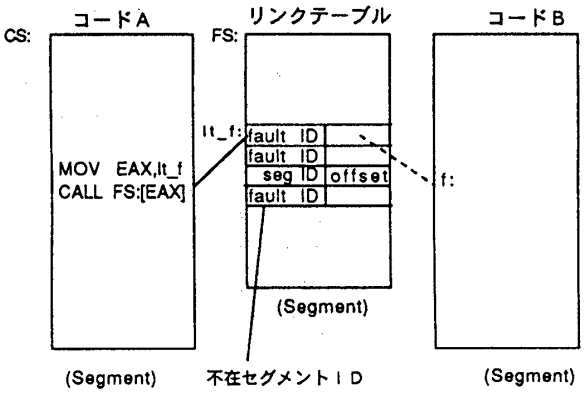

図３：不在セグメントフォールトを使った処理

これにより，外部名ごとにリンケージフォールトを発生させて，リンクを張ることができる．また，一つ以上のリンケージフォールト用セレクタを使用できる．これは例えば，特定のライブラリ用のフォールトセレクタを用意してモジュール検索を簡単にすること等に利用できる．

4．言語Cとダイナミックリンク

我々は開発言語を言語Cとし，ダイナミックリンクを行う言語Cの実行環境の設計を行った[4]．その際，リンケージフォールトに関して，言語Cの＆演算子が問題になった．

例えば，図４の左のように＆演算子が使われていたとする．最初，我々は不正アドレスを初期値としてリンクテーブル中に格納し，参照時にバスエラー例外を発生させることでリンケージフォールト処理を行おうと考えていた．しかし，その方法では (2) の位置でフォールトが発生するため，リンクテーブルの書換えを行えない．また，不正アドレスが変数に格納されているため，フォールトからの復帰後も不正アドレスでのアクセスが行われてしまう．正しいアドレスを取り出し，リンクテーブルの内容を更新するためには，(1) のアドレスを取り出す命令でリンケージフォールトが発生しなくてはならない．

```
int a;                mov  reg,lt_a(ltbr)    (1)
int extn_f(){         mov  p(sp),reg
    int *p, b;        mov  reg,p(sp)         (2)
    p = &a;           mov  b(sp),(reg)
    b = *p;
}
  言語Ｃソース                機械語
```

図４：＆演算子の問題

この問題は，前述の方式では発生しない．両方ともリンクテーブルにアクセスした時点でフォールトが発生するので，＆演算子では正しくアドレスが取り出される．

また，char *p = &a のように，初期化式中で＆演算子が使われることもある．このような場合は，ロード時にあらかじめリンクし，p の内容も正しい値で初期化する．

5．おわりに

本稿では，ダイナミックリンクの実現機構を持たない CPU 上で，ページフォールトを利用したリンケージフォールト処理を行う方法を述べた．また，80386 の不在セグメント例外を利用したリンケージフォールト処理について述べた．

参考文献：

[1] 早川，並木，高橋：“手書きインタフェースを支援する OS OS/omicron 第４版の構成”，第４回コンピュータシステムシンポジウム論文集 pp.35-4 ，1992-10

[2]F.Corbato and V.Vyssotsky：“Introduction and Overview of the Multics System”，FJCC，pp.185-196

[3]本林，益田，勝枝，高橋：“２次元番地付方式による HITAC5020 TSS の 特徴”，情報処理学会誌 Vol.9，No.6(1968)，pp.317-325

[4] 中村，田中，早川，並木，高橋：“80386 用 OS/omicron 開発のための言語Ｃ処理系の実行環境の設計”情報処理学会第44回全国大会 2F-10 ，1992-3