

三層クライアント／サーバ・システムにおけるミドルウェアの開発

(3) メッセージキューの実装

4F-5

成田 秀明 鈴木勇至 松本 諭 関根 徹

松下電器産業(株) マルチメディアシステム研究所

1. はじめに

今回我々は、三層クライアント/サーバモデルを採用したシステム(本システム)を開発した^[1]。本システムでは、各種ユーザ端末であるクライアントから送られた要求メッセージがサーバにより集約されてデータベースを載せたリソースマネージャに送られる。そして、返された応答メッセージが再びサーバを経由してクライアントに送り返される仕組みとなっている。

一般に、プロセス間でメッセージを蓄積交換する場合にはキューが用いられるが、本システムにおいてもサーバ上のクライアント制御プロセス(エージェント)とリソースマネージャ制御プロセス(トレーダ)との間の要求メッセージと応答メッセージの蓄積交換に独自に開発したキューを用いている。

本稿では、開発したキューについてその概要と構成について報告する。

2. キューの概要

今回開発したキューは、異なるプロセス間でデータを蓄積交換する機能をC言語の関数ライブラリ(キューAPI)により提供している。次にキューの特徴とキューAPIについて述べる。

2.1 キューの特徴

キューの特徴を次に示す。

- キューAPIでは、プロセス間での排他制御を行なっており、キューに対する複数プロセスの同時アクセスが起こっても矛盾なく処理できる。
- 本システムでは、取り扱う要求メッセージや応答メッセージが可変長であるために、取り扱うデータの長さはキューの持つリソースの制限の範囲内で任意の長さを許している。
- キューAPIは、SystemV系UNIXオペレーティングシステムの提供するプロセス間通信(IPC)ファシリティを用いて実装しているため、多くのプラットフォームで動作可能である。

2.2 キューAPI

キューAPIによるプロセス間のデータのやりとりは、基本的に、任意のHEAP制御ユニット(HCU)から割り当てられたメモリブロックにデータを入れ、指定したFIFO制御ユニット(FCU)を介してそれをやりとりする形で行なわれる。

HCUはヒープの役割を果たす単位であり、FCUは一般に言う「キュー」という単位である。HCUおよびFCUについては後述する。

次に、キューAPIに含まれる主な関数を示す。

| | |
|------------|--------------------|
| alloc_fcuc | FCUの割り当て |
| free_fcuc | FCUの解放 |
| alloc_blk | HCUからのメモリブロックの割り当て |
| free_blk | HCUへのメモリブロックの解放 |
| enq_blk | FCUへのデータの追加 |
| deq_blk | FCUからのデータの取り出し |

3. キューの構成と動作

本システムのキューは、SystemV系UNIXオペレーティングシステムの提供するIPCファシリティである共有メモリとセマフォを利用して構成される。IPCファシリティとしては、前述の二つの他にメッセージキューがあるが、今回は利用していない。それは、SystemVメッセージキューは、基本的に固定長メッセージの交換において効果的なものであり、可変長メッセージを交換する場合には、データの取り出し操作の繰り返しが発生する可能性があるという点で不利と判断したからである。

キューは二つの共有メモリセグメント、HEAPセグメントとFIFOセグメントから構成される(図1)。キューでは、共有メモリセグメントに保持する情報は、仮想メモリアドレスに依存していないため、これら両セグメントは任意の仮想メモリアドレスにアタッチ可能となっている。これにより、プロセスのメモリ管理上の制約を少なくしている。

Developing a Middleware in the Three Tier Client-Server System

(3) Message Queue

Hideaki NARITA, Yuji SUZUKI, Satoru MATSUMOTO, Toru SEKINE

Matsushita Electric Industrial Co., Ltd.

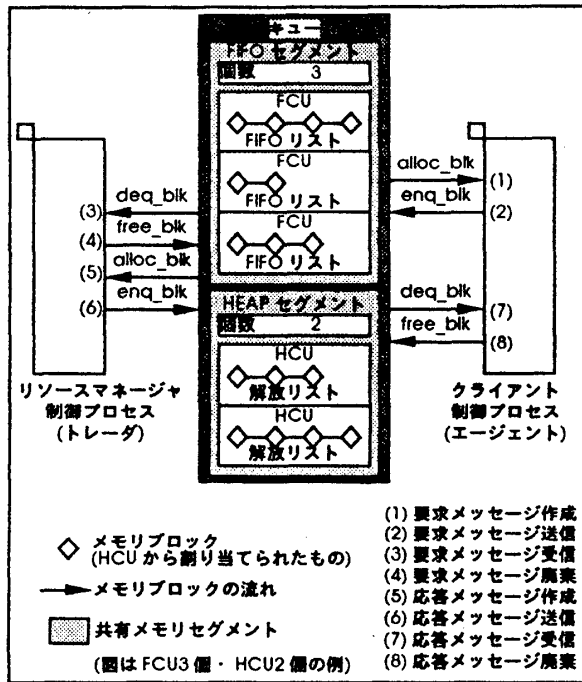


図1 キューの構成と使用例を示す概念図

3.1 HEAP セグメント

HEAP セグメントは HCU の配列とその個数情報とで構成される。この個数はキュー作成時のコンフィギュレーションにより決定される。

HCU は、任意の長さのメモリブロックを HCU 内部の領域から割り当てるものである。HCU から割り当てられるメモリブロックは、プロセスの仮想メモリアドレスを用いず、HCU における相対的なアドレスによりポイントされる。

HCU は次の要素から構成される。(a) 解放リストの先頭ブロックの位置情報の更新作業を排他的に行なうためのロックとして使用するセマフォ。(b) 解放リストの先頭ブロックの位置情報。(c) 割り当てるメモリブロックを切り出すヒープ領域。

解放リストとは解放されたメモリブロック (解放ブロック) の連結リストである。

メモリブロックの割り当ては、解放リストをたどり、所望のサイズを満足するメモリブロックを解放リストからはずすか、または一部を切り取る形で行なわれる。

メモリブロックの解放は、解放リストに当該ブロックを挿入する形で行なわれる。その時にもし隣接する解放ブロックがあれば、それらを一つの解放ブロックとして解放リストを再構成する。

HCU 作成直後は、ヒープ領域全体が一個の解放ブロックであり、解放リストの唯一のブロックとなる。なお、ヒープ領域の大きさはキュー作成時のコンフィギュレーションにより決定される。

3.2 FIFO セグメント

FIFO セグメントは FCU の配列とその個数情報とで構成される。この個数はキュー作成時のコンフィギュレーションにより決定される。

FCU は次の要素から構成される。(a)FIFO リストの先頭および終端ブロックの位置情報の更新作業を排他的に行なうためのロックとして使用するセマフォ。(b)FIFO リストに追加されたメモリブロックの個数を保持するカウンタとして使用するセマフォ。(c)FIFO リストに追加可能なメモリブロックの個数を保持するカウンタとして使用するセマフォ。(d)FIFO リストの先頭および終端ブロックの位置情報。

FIFO リストとは、任意の HCU から割り当てられたメモリブロックを先入れ先出し形式でバッファリングするための連結リストである。

FCU へのデータの追加は (1) 追加可能なメモリブロックの個数を保持するカウンタを 1 デクリメント (wait) し、(2)FCU をロックし、(3)FIFO リストの終端にデータを連結し、(4)FCU をアンロックし、(5) 追加されたメモリブロックの個数を保持するカウンタを 1 インクリメント (post) する形で行なわれる。

FCU からのデータの取り出しは (1) 追加されたメモリブロックの個数を保持するカウンタを 1 デクリメント (wait) し、(2)FCU をロックし、(3)FIFO リストの先頭のデータを取り出し、(4)FCU をアンロックし、(5) 追加可能なメモリブロックの個数を保持するカウンタを 1 インクリメント (post) する形で行なわれる。

4. おわりに

本稿では、開発したキューについてその概要および構成と動作について報告した。

今回開発したキューでは、共有メモリ上のメモリブロックをプロセス間でやりとりし、データの転送回数を減らすことにより、高速処理を実現している。特にプロセス間でやりとりするデータが大きい場合に、SystemV メッセージキューを用いて同様の機能を実現した場合に比べ、はるかに良い処理速度を出すことを確認した。

今後は、キューに対して運用系の機能とともに障害系の機能を追加することにより、信頼性を向上させる予定である。

【参考文献】

[1] 鈴木他、「三層クライアント/サーバ・システムにおけるミドルウェアの開発」、情報処理学会第 51 回全国大会講演論文集