

C 並行処理プログラムのテスト充分性評価システムの試作

1 J-8

川口豊*, 伊東栄典*, 古川善吾**, 牛島和夫*

* 九州大学工学部情報工学科, ** 九州大学情報処理教育センター

1. はじめに

並行処理プログラムの特徴は、プロセス間通信や同期によって動作を変えることである。そのために、並行処理プログラムのテストにおいては、プロセス間通信、同期を考慮したテスト法が必要である。

C 言語では、OS のシステムコールを用いることによって並行処理を実現する。したがって、C 並行処理プログラムでは、OS のライブラリで準備されているそれぞれのプロセス間通信や同期の方法に対応したテスト法を考案する必要がある。これまでに、UNIX SYSTEM V 上で実装されているセマフォと BSD UNIX 上に実装されているソケットとに対して、テスト基準を提案してきた [1]。

本稿では、セマフォに対応したテスト基準 [1] に基づくテスト充分性評価システム「セマフォ被覆率測定系」を試作したので報告する。

2. セマフォテスト基準

2.1 被覆率

まず、被覆率を定義する。被覆率はテスト充分性評価の指標となる値である。

[定義 1] 被覆率 C

$$C = \frac{|W|}{|M|}$$

ただし、 W はプログラム実行時に発生する事象の集合、 M はプログラムに含まれる測定事象の集合、 $|\cdot|$ は集合の要素数を表す。

2.2 セマフォ命令順序対テスト基準

セマフォに対しては、P 命令、V 命令という 2 つの操作が許されている。ここで、P 命令、V 命令の動作を考えると、P 命令の場合プロセスが停止するかどうか、V 命令の場合セマフォの値を変化させるかどうかというように、P 命令、V 命令ともに 2 つの異なる動作がある。テストを行なう際にはこの違いをきちんと区別して、それぞれの場合に正しく動作することを確認する必要がある。そこで、セマフォ命令を次のように定義する。ここで、*semop* は、UNIX 上で P 命令、V 命令を実行するシステムコールである。

[定義 2] セマフォ命令 *semcom* は次の 2 つ組である。

$$\text{semcom}(\text{semid}) = (\text{semop}(\text{semid}), b)$$

$$b = \begin{cases} 1 \dots & \left\{ \begin{array}{l} \text{セマフォ値} = 0, \text{または} \\ \text{プロセスキューが空でない} \end{array} \right. \\ 0 \dots & \left\{ \begin{array}{l} \text{セマフォ値} > 0, \text{または} \\ \text{プロセスキューが空} \end{array} \right. \end{cases}$$

A prototype of testing reliability evaluation system for C concurrent programs

Y. Kawaguchi, E. Itou, Z. Furukawa, K. Ushijima

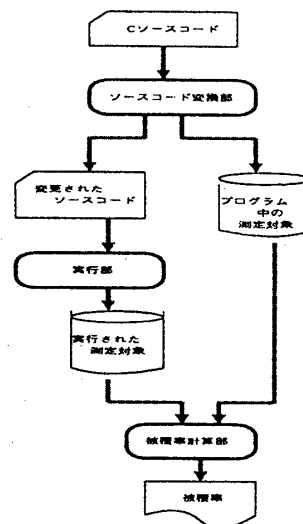


図 1: セマフォ命令被覆率測定系の概要

さらに、セマフォ命令の順序対をセマフォ命令順序対と定義する。

プログラムに含まれるセマフォ命令順序対の集合を測定対象とするテスト基準がセマフォ命令順序対テスト基準である。セマフォ命令順序対テスト基準を満足するテスト法は、完全通信不良を必ず発見することができる。完全通信不良とは、通信におけるすべてのデータに対して必ず誤りを発生する通信不良のことである。

今回試作したテスト充分性評価システム「セマフォ被覆率測定系」は、セマフォ命令順序対テスト基準に基づく被覆率を計算する。ただし、異なったテストデータに対して得られた被覆率を累積する機能はまだ実現していない。

3. セマフォ被覆率測定系

セマフォ被覆率測定系は、C のソースコードを入力とし、セマフォ命令順序対テスト基準に基づく被覆率の値を出力する。測定系の概要を図 1. に示す。測定系はソースコード変換部、実行部、被覆率計算部から構成される。以下、それぞれの部分について説明する。

3.1 ソースコード変換部

ソースコード変換部は、プログラムに含まれるすべてのセマフォ命令を抽出する。同時に、実行されたセマフォ命令を検出するためにソースコードを変更する。C のソースコードを入力として、プログラムに含まれる測定対象（この場合は、セマフォ命令）と実行されたセマフォ命令を検出するためのコード（探針）を挿入したソースコードとを出力する。

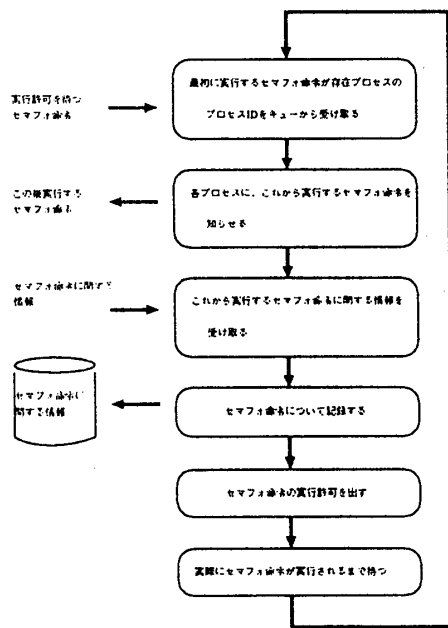


図 2: モニタの構成

3.2 実行部

この部分では、ソースコード変換部で変更されたソースコードをコンパイルした後に、被テストプログラムに入力データを与えて実行する。この実行に際して、検出および記録用プロセス（モニタプロセス）と探針との間の通信によって実際に実行されたセマフォ命令を記録する。モニタの構成を図 2. に示す。モニタの動作は、以下の通りである。

- 各プロセスからモニタとの通信許可を受け付ける。
- セマフォ命令を実行しようとするプロセスから、これから実行しようとしているセマフォ命令に関する情報を受け取る。
- セマフォ命令の実行許可を出す。
- 受け取った情報通りにセマフォ命令が実行されるまで待つ。（セマフォ命令が実行されたかどうかは、セマフォの値や待機中のプロセス数の変化から判断する。）

3.3 被覆率計算部

被覆率計算部では、プログラムに含まれるセマフォ命令順序対集合の要素数と、実際に実行されたセマフォ命令順序対集合の要素数とから、セマフォ命令順序対テスト基準に基づく被覆率を計算する。

4. 評価および議論

4.1 実行例

被覆率測定系に、「生産者-消費者問題」のプログラムを入力して実行した際の、検出されたセマフォ命令数、実行されたセマフォ命令順序対集合の要素数 $|M|$ 、被覆率の関係を表 1. に示す。このプログラムは、3 つのセマフォを使用している。それらのセマフォを利用するためのシステムコール `semop` はプログ

実行されたセマフォ命令数	$ W $	被覆率 (%)
15	9	9.4
39	10	10.4
215	10	10.4
2229	14	14.6
8790	15	15.6
25176	15	15.6

表 1: 被覆率と実行されたセマフォ命令数との関係

ラム中の 8 箇所で使用されている。また、プログラム中に含まれるセマフォ命令順序対集合 M の要素数は 96 である。

表 1. から、実行したセマフォ命令数が 10000 を越えても既に実行されたセマフォ順序対しか実行されていないことが分かる。

4.2 今後の課題

今後の課題として以下の点があげられる。

- 被覆率の値に関する検討
先の実行例では、実行されたセマフォ命令数が 8700 であっても約 25000 であっても被覆率は同じ 15% であった。このことから、2 つの課題が出てくる。1 つめは、セマフォ順序対の中に実行可能でないものが含まれている可能性があり、このセマフォ順序対テスト基準の意味をさらに検討する必要がある。2 つめは、スケジューリングの方法に依存して、これ以上被覆率を増加させることできない可能性がある。この場合には、実行されていないセマフォ順序対を強制的に実行させるための方法を考える必要がある。
- 他のプロセス間通信方法への対応
テスト充分性評価システムをソケットやメッセージ通信などセマフォ以外のプロセス間通信方法に対応させるために、今回試作したセマフォ被覆率測定系をどの程度変更しなければいけないかを検討する必要がある。

5. おわりに

これまでに提案してきた並行処理プログラムのプロセス間通信であるセマフォに対応したテスト基準に基づくセマフォ被覆率測定系を試作した。今後、実用的な C 並行処理プログラムの被覆率を測定して、テスト法の実用性を評価していく。また、セマフォ以外のソケットやメッセージ通信などのプロセス間通信や同期方法にも対応できるようにシステムの拡張を検討する。

参考文献

- [1] 伊東栄典, 川口豊, 古川善吾, 牛島和夫: C 並行処理プログラムのプロセス間通信に関するテスト充分性評価について, 情報処理学会研究会報告 Vol.93, No.13, 93-SE-90, pp.9-16, 1993.
- [2] 川口豊: C 並行処理プログラムのテスト充分性評価システムの試作, 九州大学卒業論文, 1993.