B-035

形式手法を用いた組込み用 OS の試作
—B メソッドによる仕様検証実験—
Design of an operating system for embedded systems using a formal method
- An experiment on the verification of specification with the B method -

川守田 慶† 野口 健一郎†
Kei Kawamorita Kenichiro Noguchi

1. はじめに

近年、組込み機器の多機能化、高機能化が急速に進んでいることから、組込み機器用 OS の信頼性及びセキュリティ向上が重要な課題になっている。現在、信頼性及びセキュリティの高い組込み用 OS の実現を目指して、セパレーションカーネル方式を利用したアーキテクチャを検討し、それを試作中である。本論文では、試作中の機能の一部の仕様を、代表的な形式手法である B メソッドを用いて記述し、整合性の検証を行った結果を報告する。

2. 組込み用 OS の信頼性とセキュリティの強化

これまで、信頼性及びセキュリティの高い組込み用 OS の実現を目指して、セパレーションカーネル上でクライアントサーバモデルの環境を構成するアーキテクチャを提案した[1]。

セパレーションカーネルは、John Rushby によって提案された方式[2]で、複数の独立したパーティション(仮想マシン)空間とパーティション間の通信路を提供する。近頃、組込み用に適用できるものとして注目されている。本研究では、カーネルを小さく単純なものにして正当性の証明を容易にすることと、プロセスの信頼度に応じて異なるパーティション上で動作させることでシステムのセキュリティを向上させることを狙いとして、セパレーションカーネルを採用することにした。

試作にあたって、セパレーションカーネルの仕様を B メソッドを用いて記述し、正当性の証明を通じて仕様を検証することによってバグの少ないカーネルの実現を目指している。

B メソッドは、Jean-Raymond Abrial らによって開発された形式手法に基づいたソフトウェア開発手法[3,4,5]で、仕様からプログラム作成までを段階的詳細化によってサポートしている。B メソッドでは仕様と詳細化の各段階で整合性を検証する方法と、下位段階への詳細化の正当性を検証する方法が提供されている。

3. 提案アーキテクチャの試作

現在、提案したアーキテクチャ中のセパレーションカーネル部分を IA-32(Intel Architecture-32)アーキテクチャのプロセッサ[6]上で試作中である(図 1)。セパレーションカーネルの機能の中で、実線で囲んだ機能は実装済みのものを、点線で囲んだ機能は実装中のものを表す。

試作中のアーキテクチャの特徴を次に述べる。

(1) ユーザプロセスはいずれかのクライアントパーティション上で動作させる。

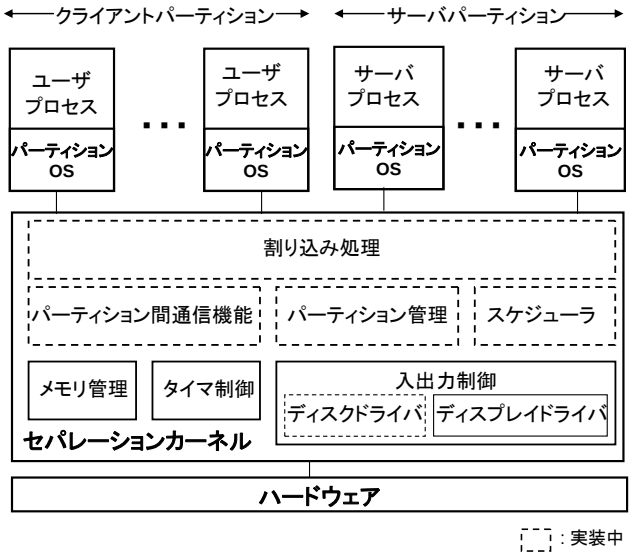

図 1: 提案アーキテクチャの概要

(2) セパレーションカーネルはパーティション間のメッセージ通信機能を提供するために、以下のカーネルコールを持つ。

send：宛先パーティションへメッセージを送信し応答を待つ

receive：メッセージを受信する。メッセージが到着していない場合は待ち状態になる。

reply：受信したメッセージの送信元へ応答メッセージを送信する

(3) セパレーションカーネルは、パーティションのドメインに応じて通信可能なパーティションを制御する。

(4) プロセスからは、パーティション OS が持つシステムコールのみ直接利用できるようにし、セパレーションカーネルが提供するカーネルコールは直接利用できないようにする。

(5) パーティションがセパレーションカーネルのメモリ領域に直接アクセスできないようにするために、IA-32 が持つリング保護を利用する。セパレーションカーネルには特権レベル 0 を、サーバ用パーティション OS には 1 を、クライアント用パーティション OS には 2 を、プロセスには 3 を割り当てる。

(6) パーティション間のメモリ保護を実現するために、パーティション毎に LDT(Local Descriptor Table)を割り当て、そこに各パーティションのセグメントを登録する。これによって、特権レベルが同一であっても、他のパーティションの LDT を直接参照することができない。

† 神奈川大学大学院理学研究科情報科学専攻

4. 仕様の検証

4.1 仕様の記述

セパレーションカーネルの機能の中で、メッセージ通信機能、スケジューラ、タイマ制御の仕様を B で記述した(約 480 行)。記述の中の主な構成要素とそれらの関係を図 2 に示す。

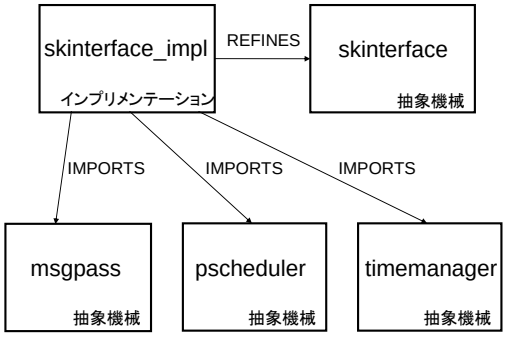

図 2 : 記述した仕様の主な構成要素

(1) カーネルコールの仕様は skinterface として記述し、skinterface_impl として詳細化した。skinterface_impl の各オペレーションは、内部でメッセージ通信機能(msgpass)やスケジューラ(pscheduler)、タイマ制御(timemanager)内のオペレーションを呼び出している。

(2) skinterface_impl の send、receive、reply オペレーションはそれぞれ内部で msgpass の sk_send、sk_reply、sk_receive オペレーションを呼び出している。

(3) sk_send (dst, msg, wmt)オペレーション(図 3)では、宛先パーティション ID である dst のパーティションが待ち状態にある(pstate(dst)=p_waiting)なら、dst へメッセージ msg を送信して、dst からの応答(reply の呼び出し)を最大待ち時間である wmt ミリ秒待つようにタイマを設定する。dst がまだ receive を発行していない(pstate(dst)=p_waiting でない)ならば、宛先にあるメッセージ待ち集合 rmsgs(dst)に自らのパーティション ID(current)を加える。そして、メッセージ ID を発行して dst へ渡し、current のパーティションに保存する。また、wm_t に 0 が指定された場合は、タイマを設定しない。dst で指定された宛先パーティションの ID が自身のパーティション ID だった場合や存在しないパーティションだった場合はエラーを出力する。

(4) sk_receive(wm_t)では rmsgs(current)にメッセージ送信要求があればパーティションからのメッセージを受信し、なければ wm_t で指定された時間メッセージを待つ。wm_t に 0 が指定された場合は、メッセージを待ち続ける。

(5) sk_reply(dst, mid, msg)では mid と dst にあるメッセージ ID が一致すれば、宛先パーティション dst へメッセージ msg を返信する。一致しなければエラーとする。

4.2 検証

B メソッドを用いて記述した仕様の整合性の検証を、B4free ツールを用いて行った。B4free は B メソッドのためのツールで、証明責務の生成や証明を支援している。B4free を用いて検証を行った結果、ツールによって生成された証明責務(72 個)のすべての証明を完了させることができた。

```
result <-- sk_send(dst, msg, wm_t) =
PRE  dst ∈ PID ∧ msg ∈ MESSAGE ∧ wm_t ∈ ℕ
THEN
  IF current ≠ dst ∧ dst ∈ allocated_pid
  THEN
    IF pstate(dst) = p_waiting
    THEN
      rmsg_ptr := rmsg_ptr <+ {dst↦msg} ||
      set_mid(dst) ||
      rmsg_src := rmsg_src <+ {dst↦current} ||
      smsg_dst := smsg_dst <+ {current↦dst} ||
      p_mode := p_mode <+ {current↦swait, dst↦ptwkup} ||
      IF wm_t > 0
      THEN
        timers := {dst} ⩤ timers <+{current↦wm_t+now}
      ELSE
        timers := {dst} ⩤ timers
      END ||
      result := dst_waiting
    ELSE
      rmsgs(dst) := rmsgs(dst) ∪ {current} ||
      set_smid ||
      smsg_dst := smsg_dst <+ {current↦dst} ||
      p_mode := p_mode <+ {current↦swait} ||
      IF wm_t > 0
      THEN
        timers := timers <+ {current↦wm_t+now}
      END ||
      result := dst_running
    END
  ELSE
    result := error
  END
END
```

図 3: sk_send オペレーションの B による記述

5. 評価

(1) 今回 B メソッドを用いてセパレーションカーネルの一部の仕様を記述し、その整合性を検証することができた。

(2) 仕様の正当性を証明するためには、整合性の検証だけでなく、記述した仕様が要求を満たしているかどうかの検証(妥当性の検証)も行うことが望ましい。

6. 今後の課題

(1) B を用いて記述した仕様の仕様実行による妥当性検証。仕様実行(仕様アニメーション)は、具体的なテストデータを用いて仕様をテスト実行することによって妥当性を検証する方法である。

(2) 記述した仕様の詳細化とその検証、実装、および評価。

参考文献

[1] 川守田 慶, 野口健一郎, 組込み用を目指した OS のセキュリティの研究 －MINIX 3 のセキュリティ機能の評価と強化の実験－, FIT2007.

[2] John Rushby. Design and Verification of Secure Systems. ACM Operating Systems Review, Vol15, No. 5, pp12-21, 1981.

[3] Jean-Raymond Abrial. The B-Book －Assigning programs to meanings, Cambridge University Press, 1996.

[4] 来間啓伸.『B メソッドによる形式的仕様記述』, 近代科学社, 2007.

[5] B Language Reference Manual Version 1.8.5, ClearSy.

[6] インテル, 『IA-32 インテル アーキテクチャ ソフトウェア・デベロッパーズ・マニュアル』.