

安全な組込み用 OS の試作

—セパレーションカーネルに基づくセキュリティ機能の検討—
 Prototyping of a Secure Operating System for Embedded Systems

—Examination of Security Functions based on the Separation Kernel—

望月 祐希†
 Yuuki Mochizuki

野口 健一郎†
 Kenichiro Noguchi

1. はじめに

本研究室では、安全な組込みシステムを実現するために、セパレーションカーネルベースのオペレーティングシステム(OS) OS-Kを試作中である[1][2]。このOSに、セパレーションカーネルが満たすべき基準として米国 National Security Agency が定義する Separation Kernel Protection Profile (SKPP)[4]を参考に、セキュリティ機能—資源(プロセス、メモリ、デバイス)のより明確な分類、その資源に対するより明確な情報の流れの制御、ファイルアクセス権限の種類追加—の検討および試作を行った。

2. 背景

2.1 セパレーションカーネル

セパレーションカーネルは、複数の独立したパーティション空間と、パーティション間の通信路を提供するためのOS構成方式で、John Rushbyによって提案された[3]。

2.2 Separation Kernel Protection Profile (SKPP)

SKPPは、セパレーションカーネルのセキュリティ機能と保証要求を規定する。米国 National Security Agencyによって、高い頑健性を持つために満たすべき基準として定義された[4]。

3. セキュリティ機能の検討

表1にOS-Kのセキュリティ機能とそれに対応するSKPPの主な要求を示す。また、1~4について対応状況を次に述べる。

表1 OS-Kのセキュリティ機能とSKPPの要求

	OS-Kのセキュリティ機能	対応するSKPPの要求
1	独立したパーティション空間	資源の分離
2	全てのファイルの暗号化	信頼された配達
3	ロードモジュールの完全性検証	信頼された初期化
4	パーティション間の通信制御	情報の流れ制御
5	形式手法による仕様検証	形式的な仕様記述

(1) 資源の分離

OS-Kでは、独立したパーティション空間を実現しているが、パーティションへの資源(プロセス、メモリ、デバイスなど)の割当てを明確にしていない。

(2) 信頼された配達

OS-Kでは、ロードモジュールやパーティションのプロセスが扱うファイルなど、全てのファイル(起動時に実行するカーネル復号化プログラムロードモジュールを除く)に対して暗号化および認証コードを付加している。

ただし、全てのファイルを暗号化することはSKPPの要求を越えている。

(3) 信頼された初期化

OS-Kでは、組込み機器の起動時にカーネルの復号化プログラムを実行し、次にカーネルで初期化を行っている。このため、カーネルの実行開始直後は安全な状態にない。

(4) 情報の流れ制御

OS-Kでは、パーティション間の通信制御を構成ベクタから読み込み、動的に制御している。ただし、デバイスへのアクセス制御は静的である。

4. 資源の分離

パーティションへの資源の割当てを明確に行うために、次の改善を行う。

(1) 資源をプロセス、メモリ、デバイスに分類する。

(2) 各パーティションに必要な資源のみを割当てる。

(3) 割当てる資源とその内容を構成ベクタに記述する。

図1に、割当ての例を示す。各パーティションには異なるプロセスとメモリ空間が割当てられ、パーティションCにはデバイスが割当てられている。

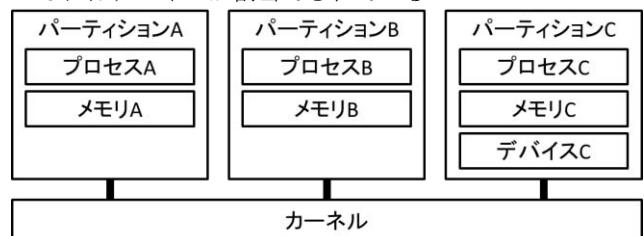


図1 パーティションへの資源の割当て例

デバイスへのアクセスは、I/OやメモリマップドI/Oへのアクセスである。そのため、I/Oかメモリが実際の資源となる。図1のデバイスCがI/Oポート0番を使って操作されるのであれば、構成ベクタにはパーティションCがI/Oポート0番を持つように記述する。

5. 信頼された配達

OS-Kは、OSに必要なファイル(カーネルロードモジュール、構成ベクタ、プログラムロードモジュール)とパーティションが扱うファイルを、別々のディスクパーティションで扱っている。このため、二つのロード機能を使用して、開発環境で組込み機器のディスクへ格納する。概要を図2に示す。

ロード機能1は従来通り、OSに必要なファイルの暗号化を行い、認証コードを付加して格納する。初期化機能ロードモジュールは、組込み機器の起動時に実行するため、暗号化を行わない。

† 神奈川大学大学院理学研究科情報科学専攻

ロード機能2は、パーティションが扱うファイルを、暗号化および認証コードの付加を行う。ただし、ファイル毎に暗号化を行うか否かを選択するよう改善する。また、ファイルにパーティション毎のアクセス権限を設定する。

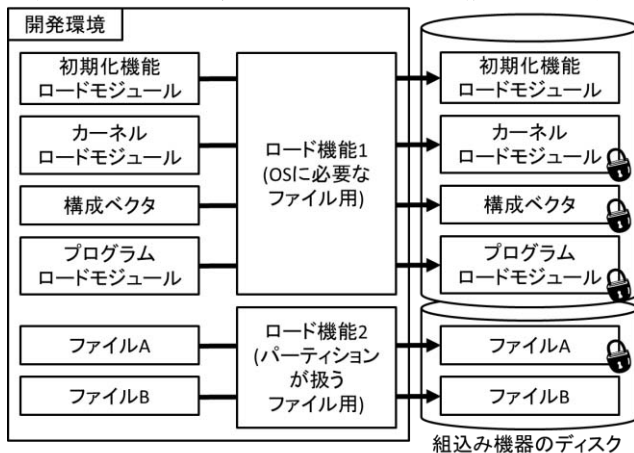


図2 組み込み機器のディスクの作成

6. 信頼された初期化

カーネルを安全な状態の下で実行するために、次の改善を行う。

- (1) 初期化は、カーネルではなく初期化機能により行う。
- (2) 構成ベクタを用いて、パーティションの作成と資源の割当てを行う。

概要を図3に示す。初期化機能は、組み込み機器を起動した管理者から鍵を入力し、これを用いて組み込み機器のディスクに格納したカーネルロードモジュール、構成ベクタ、プログラムロードモジュールを復号化、および、付加した認証コードを用いて完全性の検証を行う。検証が失敗した場合は、組み込み機器の起動を停止する。成功した場合は、組み込み機器の初期化と、構成データを用いてパーティションの作成と資源の割当てを行い、安全な状態でカーネルを実行する。

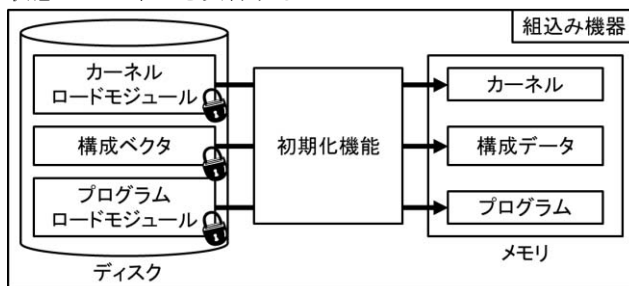


図3 システムの初期化

7. 情報の流れ制御

7.1 カーネルによる制御

初期化機能によって読み込んだ構成データを基に、カーネルによって資源を動的に制御するよう改善する。

プロセス間の情報の流れは、メッセージ通信で行うので、メッセージ通信機構によって制御する。

メモリとデバイス（つまり I/O かメモリ）へのアクセスは、CPU の保護機能によって制御する。これにより、メモリやデバイスへのアクセスは、カーネルを通さず直接行う事ができる。

7.2 ファイルサーバによる制御

OS-K では、ファイル操作はファイルサーバが制御する。ファイルサーバは、ディスクドライブを通してファイルシステムへアクセスする。そのため、ファイル操作に伴う情報の流れを許可する事が前提となる。例を図4に示す。

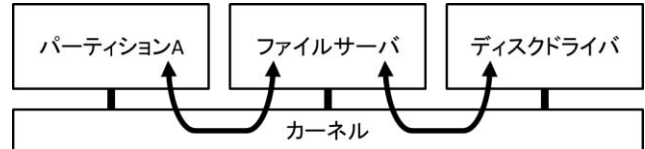


図4 ファイル操作に伴う情報の流れ

図4は、パーティションAがファイル操作を行うため、パーティションAとファイルサーバ、ファイルサーバとディスクドライブ間の情報の流れを許可する。

実際のファイルへのアクセス制御は、ファイルサーバがパーティション毎のアクセス権限を基に行う。

ファイルに暗号化の指定がされていた場合、ファイルサーバは暗号化および復号化を透過的に行う。

8. 試作

以下の試作を行った。

- (1) 構成ベクタ作成ツール
- (2) ロード機能1およびロード機能2
- (3) 初期化機能の一部（組み込み機器の初期化、鍵入力、構成データの配置）

構成ベクタ作成ツールは、人可読の構成ベクタを機械可読の構成ベクタに変換するツールである。

その他の改善については、試作中である。

9. 結言

SKPPを参照して、OS-Kのセキュリティ機能の見直しと検討を行った。資源をより明確に、分離、割当て、制御を行うことにより、より安全性を高められると考える。今後の課題は、次の通りである。

- (1) 改善のための試作の完了（初期化機能、構成データを用いた情報の流れ制御、ファイルサーバ）
- (2) POSシステムなどを想定したアプリケーションの試作
- (3) 信頼された回復、監査などのOS-Kで未対応のSKPPの機能の検討
- (4) 形式手法による仕様検証（カーネル部は検証済み[1]）

参考文献

- [1] Kei Kawamorita, Ryouta Kasahara, Yuuki Mochizuki, and Kenichiro Noguchi: Application of Formal Methods for Designing a Separation Kernel for Embedded Systems, World Academy of Science, Engineering and Technology Issue 68, pp. 506-514, July 2010.
- [2] 望月 祐希, 野口 健一郎: 組み込みOS用暗号化ロードモジュール機能及びセキュアファイルサーバの試作, FIT2010
- [3] John Rushby: The design and verification of secure systems, Eighth ACM Symposium on Operating System Principles (SOSP), pp. 12-21, 1981.
- [4] U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, Version 1.03, Jun 2007, Information Assurance Directorate.