

# 「Mint オペレーティングシステムを用いた割り込み処理のデバッグ支援環境の提案」の要約

2014/5/12

藤田将輝

## 1 はじめに

本資料では山本の特別研究報告書である「Mint オペレーティングシステムを用いた割り込み処理のデバッグ支援環境の提案」[?] の要約を示す。

## 2 仮想計算機を用いた OS デバッグ手法

### 2.1 構成

仮想計算機方式にはベアメタル型とホスト OS 型の 2 種類ある。以下でその概要について示す。

#### (1) ベアメタル型

ベアメタル型ではハードウェア上でハイパーバイザが走行し、ハイパーバイザ上でデバッグ支援 OS とデバッグ対象 OS が走行する。

#### (2) ホスト OS 型

ホスト OS 型ではハードウェア上でホスト OS が走行し、ホスト OS 上でデバッグ支援アプリケーションとハイパーバイザが走行する。また、ハイパーバイザ上でデバッグ対象 OS が走行する。

### 2.2 デバッグ手法の概要

仮想計算機を用いた OS デバッグ手法で、割り込み処理のデバッグが可能な手法として割り込み挿入手法とロギング/リプレイ手法がある。以下にその概要を示す。

#### (1) 割り込み挿入手法

割り込み挿入手法では、プログラマがデバッグ対象 OS のコードにハイパーコールを挿入する。デバッグ対象 OS の走行時、ハイパーコールを挿入したコード位置に単一の割り込みを発生させ、割り込みを再現することで、デバッグを支援する。

#### (2) ロギング/リプレイ手法

ロギング/リプレイ手法ではロギングとリプレイにより、割り込み処理を含めたバグ発生時点までの処理の流れを再現することで、デバッグを効率化する。ロギングとはデバッグ対象の処理の流れを保存することで、リプレイとはロギングで保存した情報から処理の流れを再現することである。

## 2.3 問題点

仮想計算機を用いたデバッグ手法の問題点を以下に示す．

### (1) 実計算機で発生する間隔での複数の割り込みが困難

割り込み挿入法ではデバッグ対象 OS のコードにハイパーコールを挿入し、割り込みを発生させるため、コードが実行されるタイミングは OS の処理速度に依存する．このため調整によって CPU へ発生する間隔で複数の割り込み（以下、実割り込み）が発生する間隔を調整することは困難である．ロギングではデバッグ対象 OS とハイパーバイザの間の処理の遷移や再現情報の格納による処理負荷が発生する．このため、実割り込みがロギング中に発生しないと考えられる．ロギング中に実割り込みが発生しない場合、実割り込みを再現するための再現情報を保存できない．このため、実割り込みの発生が困難である．

### (2) 任意のタイミングでの割り込み発生が困難

ロギングリプレイ手法では、ロギング時に発生した割り込みに対する処理をリプレイ時に確認できる．しかし、任意のタイミングで割り込みを発生させるためには、再現情報として割り込みを発生させるアドレスと分岐回数をプログラマが用意しなければならない．これらの指定が困難であるため、任意のタイミングで割り込みを発生させることが困難である

## 3 Mint オペレーティングシステム

### 3.1 Mint の設計方針

Mint は 1 台の計算機上で複数の Linux を独立に走行させる方式である．Mint の設計方針として、以下の 2 つがある．

- (1) すべての Linux が相互に処理負荷の影響を与えない．
- (2) すべての Linux が入出力性能を十分に利用できる．

### 3.2 Mint の構成

Mint では、1 台の計算機上で CPU、メモリ、およびデバイスを分割し、各 OS が占有する．Mint の構成例を以下で説明する．

- (1) CPU  
コア単位で分割し、各 OS ノードがコアを 1 つ以上占有する
- (2) メモリ  
空間分割し、各 OS ノードが分割領域を占有する．
- (3) デバイス

デバイス単位で分割し，各 OS ノードが指定されたデバイスを占有する．

## 4 設計

### 4.1 設計の目的

Mint を用いた割り込み処理のデバッグ支援環境を設計する目的について以下に示す．

#### (1) 実割り込みを発生させる環境の提供

割り込みの発生感覚に依存するバグを確認するためには，実割り込みを発生させる必要がある．しかし，仮想化を用いたデバッグ実割り込みを発生させることが困難である．そこで Mint を用いた割り込み処理のデバッグ支援環境では，デバッグ対象 OS へ実割り込みを発生させる環境を提供する．

#### (2) 任意のタイミングで割り込みを発生させる環境の提供

デバッグの際，デバッグ対象処理のバグの有無やバグの発生個所を確認するために，デバッグ対象の処理を繰り返し実行する．しかし割り込み処理は割り込みが非同期的に発生するため，繰り返し実行することが困難である．また，仮想計算機を用いたデバッグ手法では，任意のタイミングで割り込みを発生させることが困難である．Mint を用いた割り込み処理のデバッグ支援環境では，任意のタイミングで割り込みを発生できる環境を提供する．

### 4.2 設計における課題

Mint を用いた割り込み処理のデバッグ支援環境における課題について以下に示す．

#### (1) ハイパーバイザを用いないデバッグ支援環境の提供

実割り込みを発生させる環境を実現するため，ハイパーバイザによる処理負荷のない環境が必要となる．このため，ハイパーバイザを用いないデバッグ支援環境を提供する．

#### (2) デバッグ対象 OS がデバッグ支援機構の処理負荷の影響を受けないデバッグ支援環境の提供

デバッグ対象 OS がデバッグ支援機構の処理機構の処理負荷を受けると，実割り込みが発生しない．このため，デバッグ対象 OS がデバッグ支援機構の処理負荷の影響を受けないデバッグ支援環境を提供する

#### (3) CPU への実割り込みの発生

割り込み挿入手法のようにデバッグ対象 OS のコードにハイパーコールを挿入して割り込みを発生させると，実割り込みの発生が困難である．実割り込みを発生させる環境を実現するため，デバッグ対象 OS の CPU へ割り込みを発生させる．

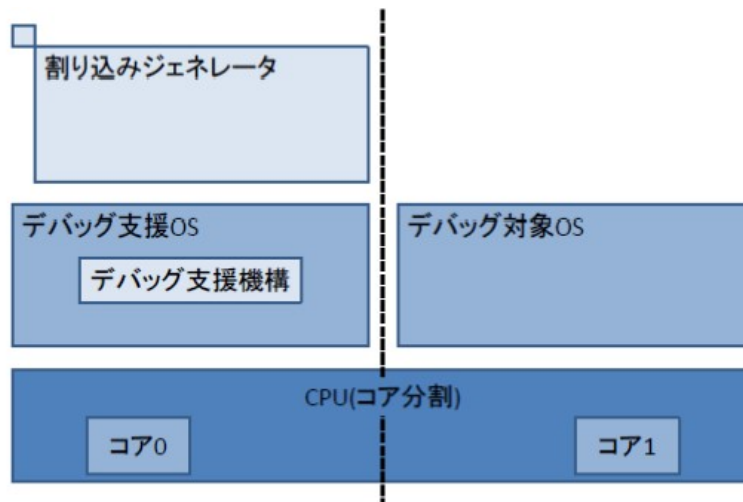


図 1 Mint を用いた割り込み処理のデバッグ支援環境の構成

### 4.3 設計における課題への対処

Mint を用いた割り込み処理のデバッグ支援環境における課題への対処を以下に示す。

(1) Mint を用いたデバッグ対象 OS とデバッグ支援 OS の同時走行

課題 (1) と (2) への対処として Mint を用いてデバッグ対象 OS とデバッグ支援 OS を同時走行させる。Mint は仮想計算機を用いないため、ハイパーバイザによる処理負荷が発生しない。また、デバッグ対象 OS がデバッグ支援 OS の影響を受けることがないため、デバッグ対象 OS がデバッグ支援機構の影響を受けない。

(2) Inter-Processor Interrupt の送信

課題 (3) への対処として、Inter-Processor Interrupt(以下、IPI) を利用する。実際に CPU へ割り込みを発生させるため、実割り込みや任意のタイミングでの割り込みを発生させられる。

## 4.4 Mint を用いた割り込み処理のデバッグ支援環境の構成

Mint を用いた割り込み処理のデバッグ支援環境の構成例を以下に示す．このデバッグ支援環境では，ハードウェア上でデバッグ対象 OS とデバッグ支援 OS が走行する．また，デバッグ支援 OS にデバッグ支援機構を実装し，この機構を利用する割り込みジェネレータをアプリケーションとして実装する．割り込みジェネレータ，デバッグ支援機構，デバッグ支援 OS，およびデバッグ対象 OS について図 1 に示し，以下で説明する．

### (1) 割り込みジェネレータ

プログラマが割り込み情報を指定する際に利用するアプリケーションである．

### (2) デバッグ支援機構

割り込みジェネレータから通知される割り込み情報をもとに，デバッグ対象 OS へ IPI を送信する機構である．

### (3) デバッグ支援 OS

デバッグ支援機構を実装して，走行している OS である．

### (4) デバッグ対象 OS

デバッグの対象となる OS である．

## 4.5 Mint を用いた割り込み処理のデバッグ支援環境における処理流れ

Mint を用いた割り込み処理のデバッグ支援環境における処理流れを図 2 に示し以下で説明する．

### (1) 割り込み情報の指定

プログラマが割り込みジェネレータを用いて割り込み情報を指定する

### (2) 割り込み情報の通知

割り込みジェネレータからデバッグ支援 OS のデバッグ支援機構へ割り込み情報を通知する．

### (3) IPI 送信要求

割り込み情報をもとに，デバッグ支援 OS のデバッグ支援機構からコア 0 へ IPI 送信要求を行う．

### (4) IPI の送信

コア 0 からコア 1 へ送信する．

### (5) 割り込み処理の開始

デバッグ対象 OS が割り込み処理を開始する．

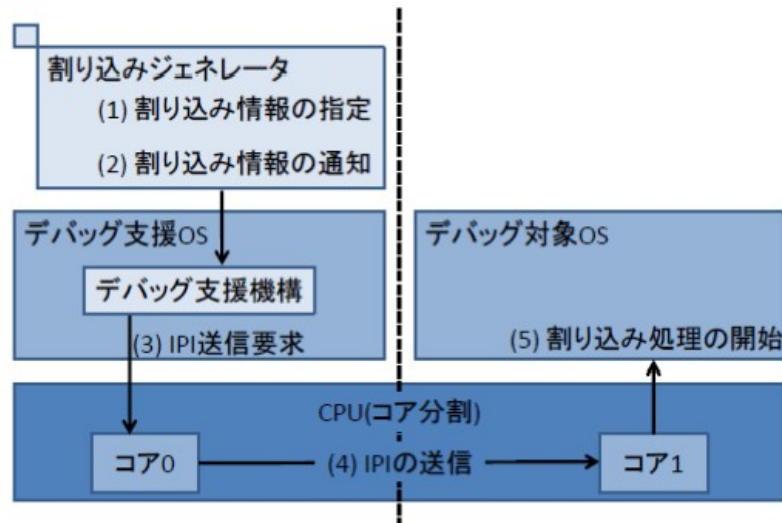


図 2 Mint を用いた割り込み処理のデバッグ支援環境におけるデバッグ処理の流れ

## 5 実装

### 5.1 実装例

デバイスの 1 つである NIC のパケット受信割り込みを発生させる環境を実現する．これにより，Mint における割り込み処理のデバッグ支援環境で，割り込み処理が再現できることを示す．実装例として，NIC がパケット受信時に発生させる割り込み（パケット受信割り込み）に対する NIC ドライバの割り込み処理のデバッグ支援環境を実装する．

### 5.2 実装における課題

NIC ドライバの割り込み処理を対象としたデバッグ支援環境の実装における課題を以下に示す．

#### (1) NIC なしで NIC ドライバを利用できる環境の提供

NIC がパケット受信割り込みや，パケット格納が行われない環境で NIC ドライバの割り込み処理を確認する必要がある．ただし，本研究では考察しない．

(2) 受信バッファへのパケット格納

パケット割り込み発生後，NIC ドライバは受信バッファからパケットを取得し，取得したパケットを用いて割り込み処理を行う．このため，NIC の代わりに受信バッファへパケットを格納する．

(3) 受信バッファ状態の更新

NIC の代わりに受信ディスクリプタの受信バッファ状態を更新する．

(4) パケット受信割り込みの発生

NIC の代わりに NIC ドライバへのパケット受信割り込みを発生させる．

(5) パケット受信割り込みの発生間隔の調整

連続して発生するパケット受信割り込みに対する割り込み処理を確認する際，パケット受信割り込みの発生間隔を調整する必要がある．ただし，本研究では考察しない．

### 5.3 実装における課題への対処

課題 (2)(3)(4) の対処を以下に示す．

(1) デバッグ支援 OS によるパケット生成と格納

課題 (2) への対処として，デバッグ支援 OS がパケットを生成し，格納する．

(2) 受信バッファ状態の更新

課題 (3) への対処として，デバッグ支援 OS が受信ディスクリプタの受信バッファ状態を更新する．

(3) デバッグ支援 OS による IPI の送信課題 (3) への対処として，デバッグ支援 OS からデバッグ対象 OS への IPI の送信によりパケット受信割り込みを発生させる．

### 5.4 NIC ドライバの割り込み処理を対象としたデバッグ支援環境の構成

パケット受信割り込みに対する NIC ドライバの割り込み処理を対象としたデバッグ支援環境の構成例について図 3 に示し，以下で説明する．

(1) 割り込みジェネレータ

プログラマが割り込み情報を指定する際に用いるアプリケーションである．

(2) デバッグ支援 OS

デバッグ支援機構を実装し，走行する Linux である．

(3) デバッグ対象 OS

Mint を用いて走行するための改変を加えた Linux である．

(4) デバッグ支援機構

デバッグ支援機構のシステムコールである．デバッグ支援機構として以下の機能を持つ．

(A) パケットの生成

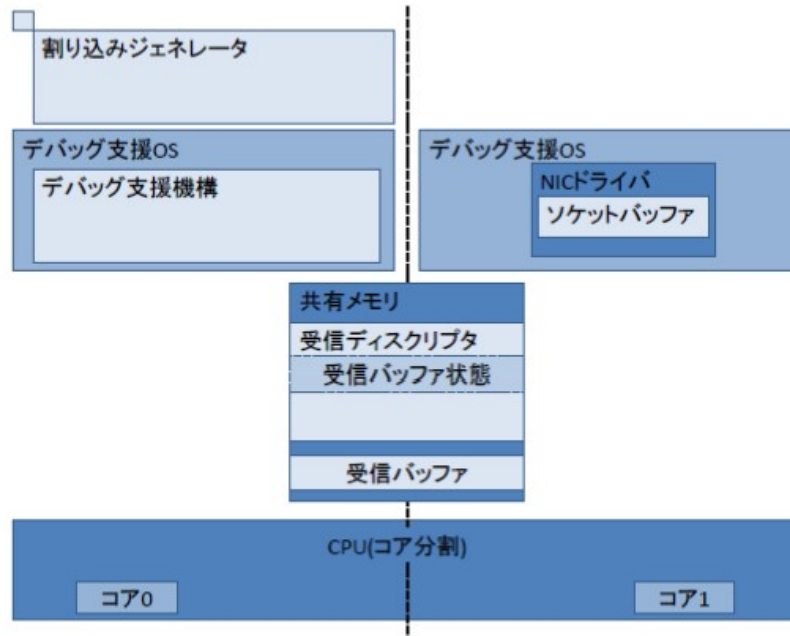


図 3 NIC ドライバの割り込み処理を対象としたデバッグ支援環境の構成

- (B) パケットの格納
- (C) 受信バッファ状態の更新
- (D) 割り込み情報をもとにした IPI の送信

#### (5) NIC ドライバ

デバッグ対象 OS の保持する NIC ドライバである。

### 5.5 NIC ドライバの割り込み処理を対象としたデバッグ支援環境における処理流れ

NIC ドライバの割り込み処理を対象としたデバッグ支援環境における処理流れについて図 4 に示し、以下で説明する。

#### (1) 割り込み情報の指定



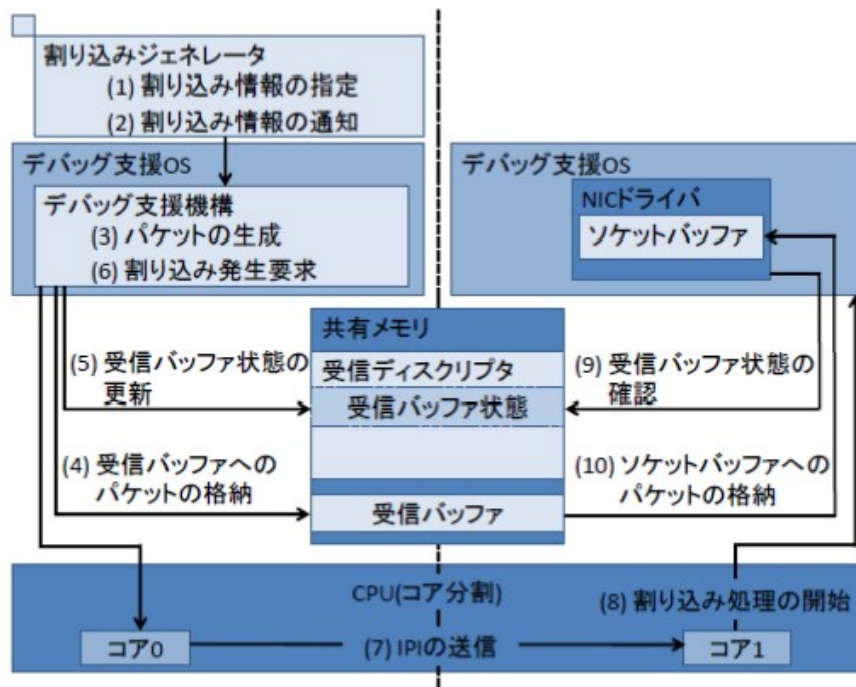


図 4 NIC ドライバの割り込み処理を対象としたデバッグ支援環境における処理流れ

プログラマが割り込みジェネレータを用いて割り込み情報を指定する。

(2) 割り込み情報の通知

割り込みジェネレータからデバッグ支援 OS のデバッグ支援機構へ割り込み情報を通知する

(3) パケット生成

デバッグ支援 OS のデバッグ支援機構がパケットを生成する。

(4) 受信バッファへのパケット格納

デバッグ支援 OS のデバッグ支援機構が受信バッファへのパケットを格納する

(5) 受信バッファ状態の更新

デバッグ支援 OS のデバッグ支援機構が受信バッファ状態を受信済み状態へ更新する。

(6) 割り込み発生の要求  
割り込み情報をもとに、デバッグ支援 OS のデバッグ支援機構からコア 0 へ割り込み発生要求を行う。

(7) IPI の送信

デバッグ支援 OS が保持するコア 0 からデバッグ対象 OS が保持するコア 1 へ IPI を送信する。

(8) 割り込み処理の開始

デバッグ対象 OS が割り込み処理を開始する。この際、デバッグ対象 OS が割り込みベクタ番号に対応した割り込みハンドラを実行する。

(9) 受信バッファの特定

受信ディスクリプタが保持する受信バッファ状態をデバッグ対象 OS の NIC ドライバが参照する。これにより、パケットが格納されている受信バッファを特定する。

(10) ソケットバッファへのパケット格納

受信バッファからソケットバッファへデバッグ対象 OS の NIC ドライバがパケットを格納する

## 6 終わりに

本資料では論文「Mint オペレーティングシステムを用いた割り込み処理のデバッグ支援環境の提案」の要約を示した。従来のデバッグ方法の概要と問題点を確認できた。さらにその問題点を Mint を用いて解決する方法を理解できた。これからは問題を解決するために必要な動作を調査し、解決に臨む。

## 参考文献

- [1] 山本凌平:Mint オペレーティングシステムを用いた割り込み処理のデバッグ支援環境の提案