

2015 年度 New グループ研究テーマ (案)(修正版)

2015/4/20

藤田将輝

1 はじめに

2015 年度 New グループ研究テーマ (案) に変更点があるため，本資料では 2015 年度修正版を示す．

2 変更点

2015 年度 New グループ研究テーマ (案) との差異を以下に示す．

- (1) 新 B4 テーマについて，(テーマ 3) 自由な起動と再起動はすでに完了しているテーマであるため，新 B4 の研究テーマから削除する．また，このテーマ内に示していた研究項目である「OS ノード環境の高速な複製手法の実現」は，(テーマ 5) OS 間の連携の内容に含まれると考えるため，(テーマ 5) に移動する．
- (2) (テーマ 5)OS 間の連携内の，「Mint におけるヘルスチェック機能の設計」については，(テーマ 7)Mint の利用の内容に含まれると考えるため，(テーマ 7) に移動する．
- (3) (テーマ 7)Mint の利用内の，「Xen modified Mint」について，Xen は一般的に使用される機会が減少しているため，Xen を研究テーマの対象とした際に Mint の恩恵を強調しにくい．このため，(テーマ 7) から削除する．
- (4) 第 270 回 New 打ち合わせにより，新テーマについての議論が行われたため，新テーマを追加する．具体的には (テーマ 7)Mint の利用に，「カーネル開発環境の構築」と「カーネルの make，および reboot の自動化」を追加する．

3 New グループ研究テーマ

現在の New グループ研究テーマと課題について，以下で説明する．New グループの研究テーマは以下の 7 つに大別される．

(テーマ 1) Mint の基本機能

マルチコア計算機上で複数のカーネルが独立に走行する Mint を実現する．

(テーマ 2) 用途別の OS の混載

Linux 系の OS に Mint 方式による変更を加えることで，複数のカーネルが 1 台の計算機上で独立に動作することを示す．

(テーマ 3) 自由な起動と終了

マルチコア計算機上で複数の Linux 系 OS が目的に応じて起動し，終了することができる．

(テーマ 4) 資源分配

Mint 方式で動作する Linux 系 OS に効率良く計算機資源を分配する．

(テーマ 5) OS 間の連携

Mint 方式を用いて，同一計算機で走行する複数 OS ノード間で連携し，処理を行う．

(テーマ 6) VM との比較

1 台の計算機上で複数の OS が同行する方式として仮想計算機方式が広く利用されている．そこで，Mint 方式と VM を比較することで，Mint の有用性について明確にする．

(テーマ 7) Mint の利用

Mint を有効に利用した環境を考察し，作成する．

4 来年度在籍する学生の研究テーマ

2015 年度の New グループに所属する Master が担当している研究テーマを以下に示す．なお，テーマの担当者は右に記載する．

(テーマ 1) Mint の基本機能

(1) 多コアプロセッサへの Mint の適用 (中村)

単一の OS で多くのコアを効率良く利用することは困難である．そこで，複数の OS でコアを分割統治する方法の 1 つに VM(Virtual Machine) 方式があるが，これには仮想化によるオーバヘッドが発生する．ここで，Mint を用いて複数のコアを分割統治する．しかし，Mint は各 Linux が最少構成としてメモリを一部，コアを 1 つ以上，ファイルシステムを持つための I/O デバイスを 1 つ以上占有する．また，1 台の計算機に搭載可能な I/O デバイスの数には物理的な制約がある．この問題への対処として，ファイルシステムを持つための I/O デバイスを占有しない OS ノードの走行を実現する．

(テーマ 4) 資源分配

(1) 割り込みルーティングの変更によるデバイス移譲 (増田)

現在 Mint は Loadable Kernel Module(LKM) を利用したデバイスの移譲が可能である．しかし，LKM を利用したデバイスの移譲時間はミリ秒単位のオーダとなっている．このため，タイムスライス間隔でのデバイス移譲は不可能である．そこで，MSI を利用することで，マイクロ秒単位でのデバイス移譲を可能にする．

(テーマ 7) Mint の利用

(1) デバッグ支援環境の作成 (藤田)

Mint を用いて，OS のデバッグ支援環境を実現する．既存の OS のデバッグ支援環境として仮想計算機を用いたものがある．しかし，仮想計算機を用いたものはハイパーバイザへの処理遷移に伴う処理負荷が発生する．このため，実割り込みの再現が困難になる．Mint は OS 間の独立性が保たれているため，一方の処理負荷を受けずに割り込み処理の再現が可能になる．

5 新 B4 の研究テーマ

2014 年度 New グループに所属する Bachelor に割り振る研究テーマ案を以下に示す．なお，各テーマにおいて，2013 年度に担当者が確定していたものは，担当者名を右に記載する．

(テーマ 2) 用途別 OS の混載

(1) 組み込み環境への対応

1 台の計算機上で複数の OS を同時走行させる手法として，仮想計算機方式がある．仮想計算機方式を用いる場合，ハイパーバイザの処理によるオーバヘッドが発生する．このため，仮想計算機方式はリアルタイム性が要求される組み込み環境には適していない．一方 Mint は，実計算機と同じ環境で複数 OS を同時走行させることができる．そこで，組み込み環境で Mint を実現する．

(テーマ 4) 資源分配

(1) Mint における資源の可視化 (池田騰)

Mint において，CPU コア，メモリ，および入出力デバイスを各 OS ノードがどのように占有しているのかをユーザに提示する機能を実現する．この際，CPU コアとメモリの資源管理インタフェースと連携する．

(2) Mint におけるデバイス認識の動的変更 (池田騰)

現在，Mint 方式ではある OS ノードが占有するデバイスを他 OS ノードから隠蔽することで，デバイスの分配を実現している．デバイスの隠蔽を OS ノード走行中に動的に変更することで，デバイスの動的な分配を実現する．

(3) USB デバイスへの対応

現在，Mint において各 OS ノードが USB デバイスを占有して使用することを想定していない．このため，Mint で USB デバイスを使用可能にする．

(テーマ 5) OS 間の連携

(1) Mint におけるヘルスチェック機能の設計 (天野)

Mint におけるヘルスチェック機能を設計する．ヘルスチェック機能は攻撃対象 OS から独立に動作するセキュリティシステムである．セキュリティシステムが攻撃対象 OS から独立していることで，攻撃対象 OS が乗っ取られた場合でも，セキュリティシステムを無効化される危険が少ない．Mint におけるヘルスチェック機能を完成させるためには，以下の 5 つの課題を完了させる必要がある．

(A) コアを一時停止する機能の実装

(B) 一時停止したコアを再開する機能の実装

(C) ハッシュ値を取得する機能の実装

(D) サービス用 OS を起動中に検査する機能の実装

(E) LKM 検査機能の実装

以上の 5 つの課題の内，(A) コアを一時停止する機能の実装と (B) 一時停止したコアを再

開する機能の実装は完了している．また，必須の機能ではないが，TPM(Trusted Platform Module) を使用して後続 OS を起動し，ヘルスチェック機能の正当性を向上させる必要がある．

(2) OS ノード間の高速通信機能の実現 (仲尾)

Mint において，OS ノード間の高速な通信機能を実現する．具体的には，Mint 独自のデバイスドライバとして，仮想 NIC ドライバを作成することで，同一計算機上の OS ノード間で高速な通信機能を実現する．

(3) OS ノード環境の高速な複製手法の実現

Mint において，すでに起動している OS ノードの環境を高速に複製する手法を実現する．OS ノード環境の高速な複製手法を実現するためには，以下の課題を完了させる必要がある．

(A) すでに走行しているカーネルの指定領域へのコピー

(B) 初期化ルーチンの変更

(C) コアの走行環境の再現

(テーマ 7) Mint の利用

(1) 開発環境の構築

Mint を用いて開発環境を構築する．具体的には，監視する OS と複数の監視対象 OS を走行させる．監視対象 OS は，各々異なるカーネルを有しており，これらの OS の動作を監視する OS により，監視する．

(2) カーネルの make，および reboot の自動化

カーネル開発を行う際，何度もカーネルを make し，再起動を行う必要がある．これらを自動化し，開発環境を構築する．また，カーネルの make と再起動にかかる時間と人間のモチベーションとの関係を調査し，考察する．

6 完了した研究テーマ

前年度までに完了した研究テーマをいかに示す．なお，担当者については研究テーマの右に記載する．

(テーマ 1) Mint の基本機能

(1) メモリ分配機能 (栗田)

各 OS ノードが占有するメモリ領域を操作し，各 OS に特定のメモリ領域を占有させる．

(2) CPU コア分割機能 (栗田)

各 OS ノードが占有する CPU コア以外のコアを使用しないように制限する．

(3) 入出力デバイス占有機能 (栗田・千崎)

各 OS ノードにおいて，任意の入出力デバイスを選択し，占有させる．また，占有していない入出力デバイスを使用しないようにする．

(4) OS ノードの起動 (栗田)

最初に起動した OS ノードからシステムコールを発行することでほかの OS ノードを起動

する．

(テーマ 2) 用途別 OS の混載

(1) Linux 系 OS の混載 (中原)

32bitLinux と 64bitLinux を混載する．

(テーマ 3) 自由な起動と終了

(1) Kexec を用いた OS ノードの起動 (中原)

Kexec を用いて最初に起動した OS ノードから他の OS ノードを起動する．

(2) 単一のカーネルイメージを用いた起動 (中原)

同一の Linux カーネルを起動する際に複数のカーネルイメージを用意することなくカーネルを起動する．

(3) ディスクレスブート機能 (天野)

初期 RAM ディスクを変更することで、ディスクを占有しない OS ノードを起動する．

(4) 起動の並列化 (中原)

複数 OS ノードを並列に起動する．

(テーマ 4) 資源分配

(1) Mint における実メモリ分配 (宮崎)

Mint では、カーネルメイク時に実メモリ分配を決め打ちで指定しておき、OS ノード起動時に OS ノード毎に処理を分岐している．よって、OS ノードメモリ構成を柔軟に設定可能にする．

(2) Mint における実メモリ移譲 (宮崎)

Mint において、Linux のメモリホットプラグ機能を利用することで Mint の OS ノード間で実メモリ移譲を可能にする．

(3) Mint における CPU コア移譲 (池田騰)

Mint において、Linux の CPU ホットプラグ機能を利用することで、Mint の OS ノード間で CPU コアを移譲可能にする．

(テーマ 6) VM との比較

(1) VM との性能比較 (池田剛，仲尾)

VM 方式との性能比較を行う．具体的には、Xen と KVM で比較する．

(2) Mint における KVM の評価 (仲尾)

同一計算機上に複数の OS を走行させる方式として、VM 方式が広く利用されている．このため、Mint 方式と VM 方式の処理性能を比較することで、Mint の性能について明らかにする．また、Mint 方式の OS ノードは通常の Linux として動作することから、Mint 方式で動作する OS ノードに VM 方式を導入した場合の処理性能を明らかにする．

(テーマ 7) Mint の利用

(1) KVM on Mint(仲尾)

近年、複数の OS が走行する技術の多段化が望まれている．しかし、既存技術による多段化は OS の性能低下が大きい．このため、最下段の走行方式に Mint を用いることで、性能低

下を抑制できることを示す．具体的には，KVM を Mint を用いて複数走行させた際の性能を評価した．

7 保留するテーマ

前年度までの研究テーマにおいて，保留するテーマについて以下に示す．なお，担当者については研究テーマの右に記載する．

(テーマ 2) 用途別 OS の混載

(1) Linux 系 OS の混載 (北川)

Mint において Linux をベースとする OS を混載させる．この際，Linux ベースの OS を混載するために必要な改修箇所と改修理由を明らかにする．また，OS 混載の更なる研究として ARM アーキテクチャ上に Mint を適用し，Android と Linux の混載を目指す．しかし，ARM 用の Kexec が正常に動作できることを確認していないため，保留とする．

8 おわりに

本資料では，2015 年度 New グループテーマ (案) の修正版について記述した．