AMD64の仮想化技術を利用した仮想マシンモニタの実装

金田 憲二

発表の概要

- AMD64の仮想化技術を利用した
  (非常に単純な)仮想マシンモニタを実装した
  - 簡単なプログラムを仮想マシン上で実行できる
  - 3000行程ほどのCプログラムからなる

※ アカデミックな内容についての発表というよりは、
ハウツーもの

このVMMの特長

- 実装が軽量である

  c.f.) Xenのソースコードは100,000行以上

➔ 使い勝手が良い（はず）

– 教育・学習での利用

– 今後の研究のための基盤

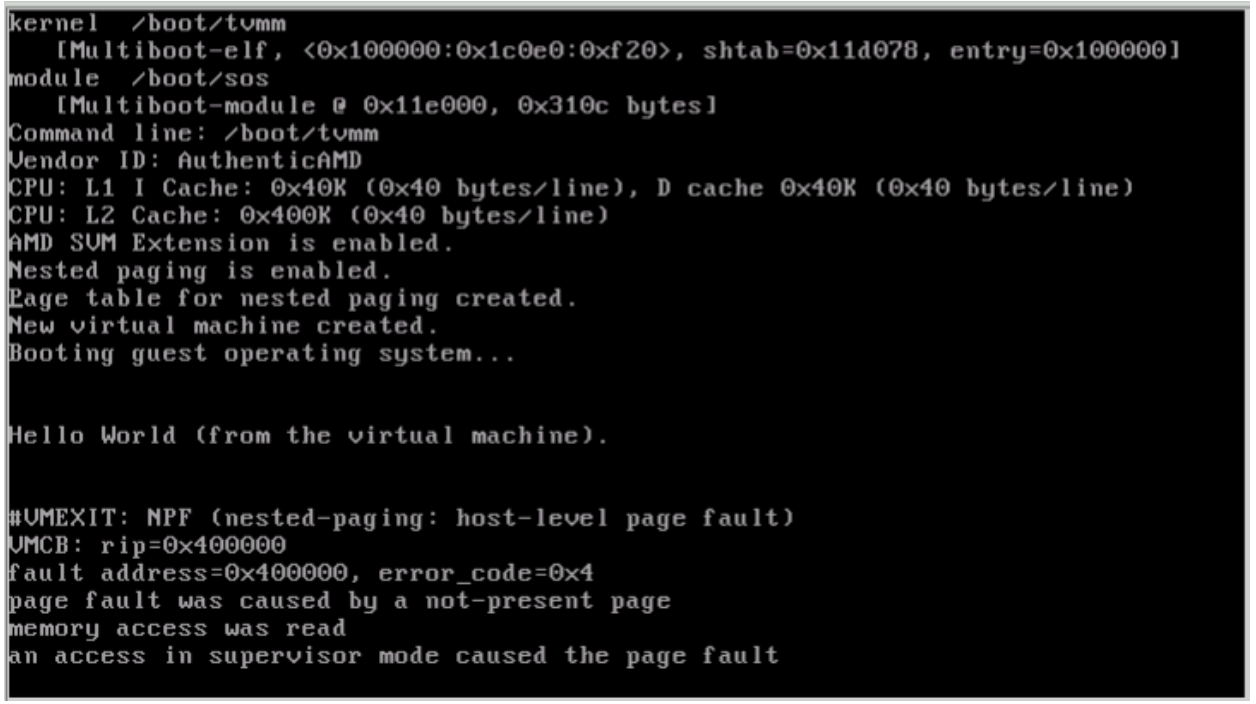
  - 検証（定理証明器、モデル検査）
  - メモリ安全な言語（Cyclone、TAL、Haskell）による記述

– …

VMMの動作デモ

- VGAに文字を出力するプログラムを仮想マシン上で実行

  ※AMD64のシミュレータSimNow上で実行



残りの発表の流れ

(Part 1) AMD64 Secure Virtual Machine

(Part 2) VMMの作り方

(Part 1)
AMD64 Secure Virtual Machine

VMMの実装に必要な処理は？

- プロセッサの仮想化
  - Sensitive命令の捕捉
  - 割り込みの転送
- メモリの仮想化
  - 独立したメモリ空間を個々のVMに提供
- …

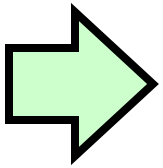
IA-32上でVMMを実装する場合

- プロセッサの仮想化
  - Sensitive命令の捕捉
  - 割り込みの転送
  - → カーネルの書き換え
  - → 動的バイナリ変換
- メモリの仮想化
  - 独立したメモリ空間を個々のVMに提供
  - → カーネルの書き換え
  - → hadow page table
- …

実装コスト大
(c.f., VMware、Xen)

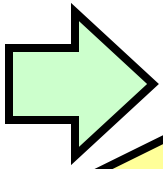
AMD64上でVMMを実装する場合

- プロセッサの仮想化
  - Sensitive命令の捕捉
  - 割り込みの転送



  - VMRUN命令
  - #VMEXIT例外

- メモリの仮想化
  - 独立したメモリ空間を個々のVMに提供



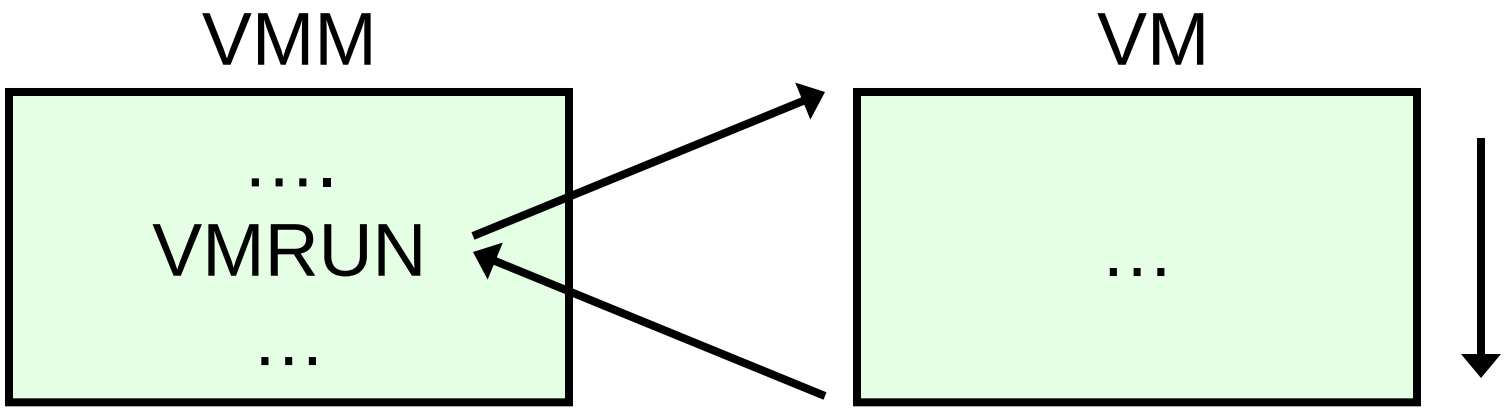
Nested paging

- …

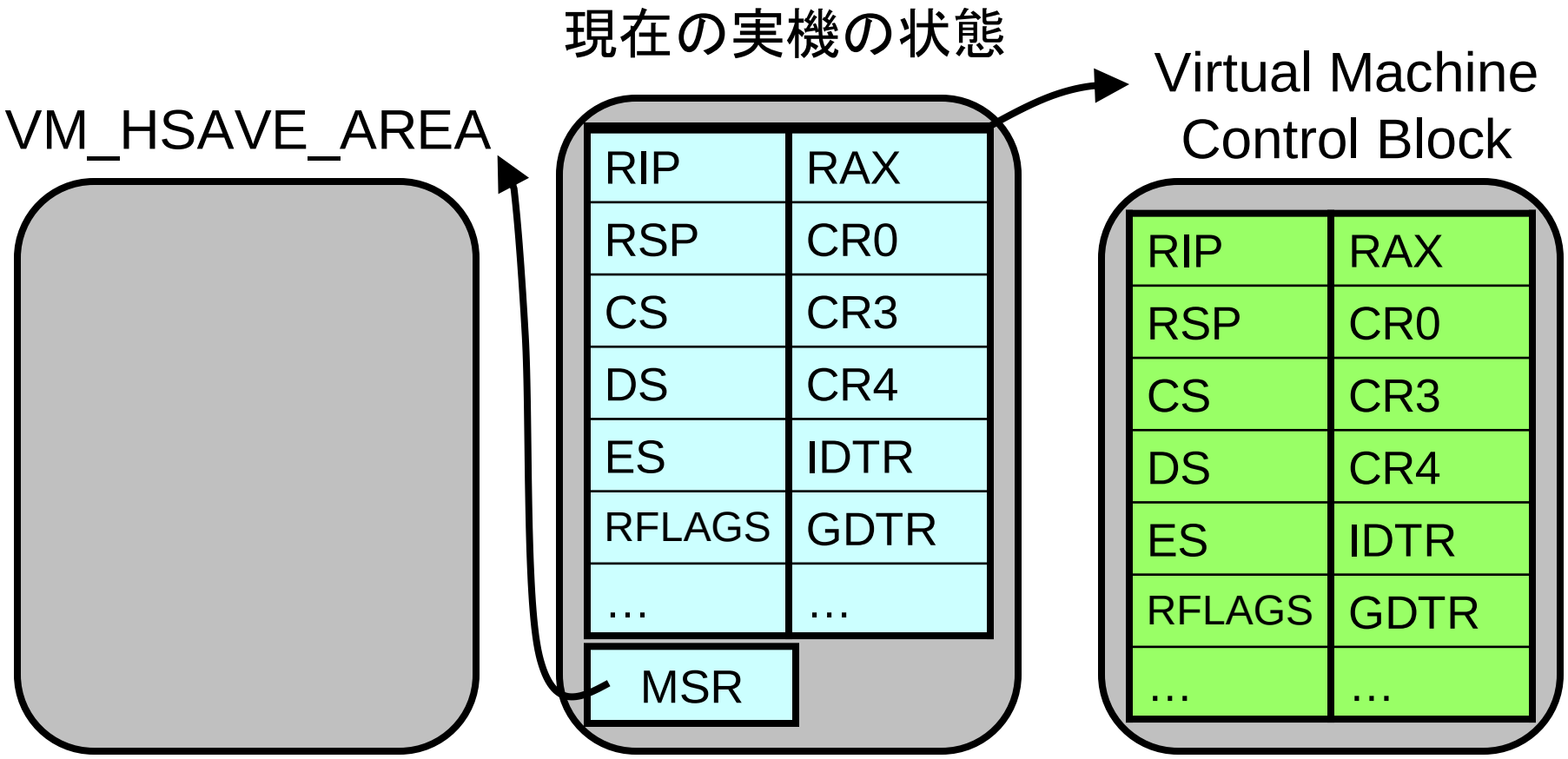
実装コスト小
（カーネルの書き換え不要）

VMRUN命令とは？

- VMMからVMに制御を切り替える命令
  - エミュレーションが必要な命令をVMが実行したり、例外が発生したりすると、VMMに制御が戻る

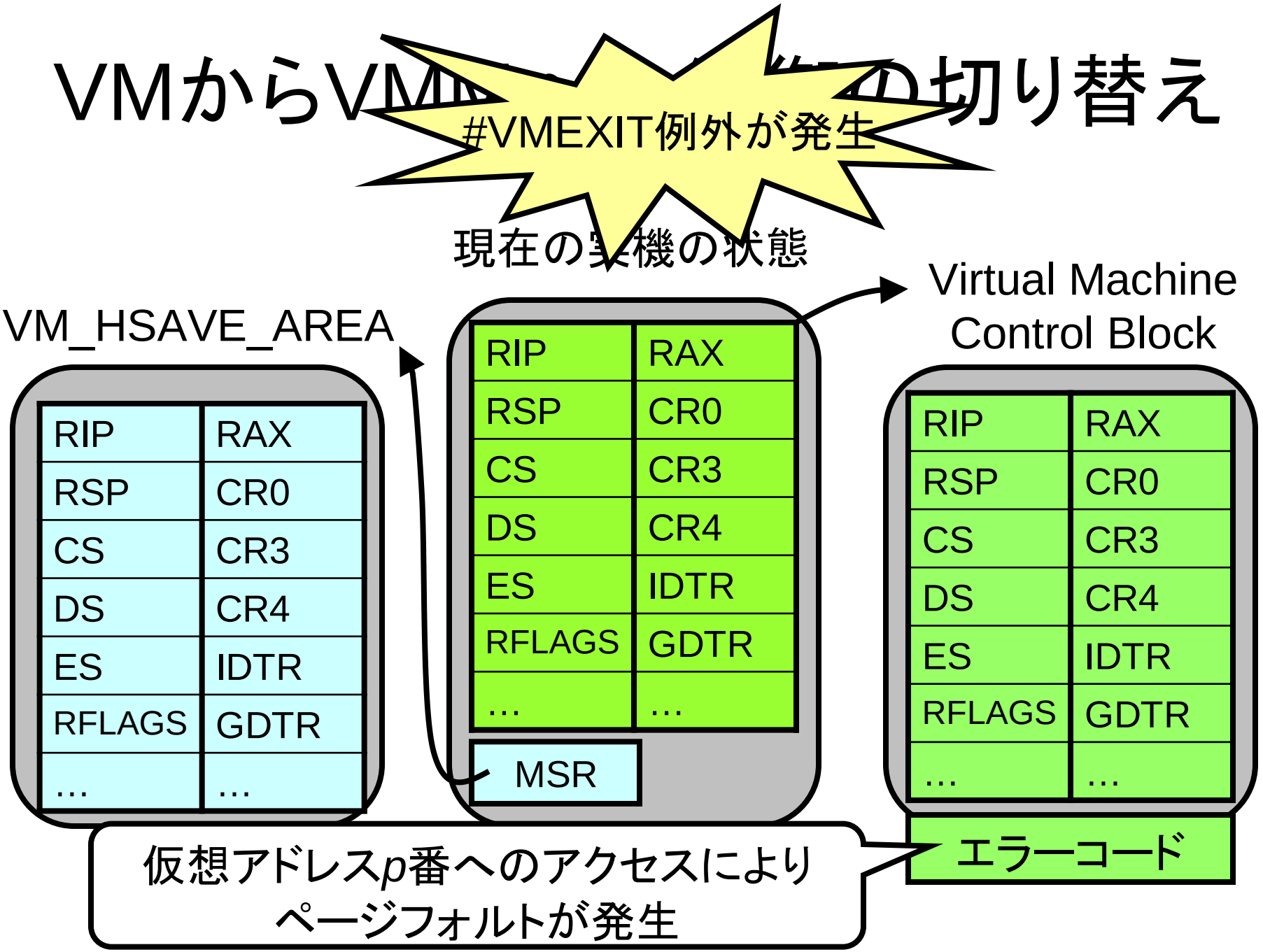


VMMからVMへの制御の切り替え



※ 全てのレジスタが保存・復元されないことに注意

VMからVMMへの切り替え



割り込みの仮想化

- VM実行中に割り込みが発生すると
  - #VMEXITが発生し、ホストに割り込みが転送される

  or
  - VMに直接割り込みが転送される

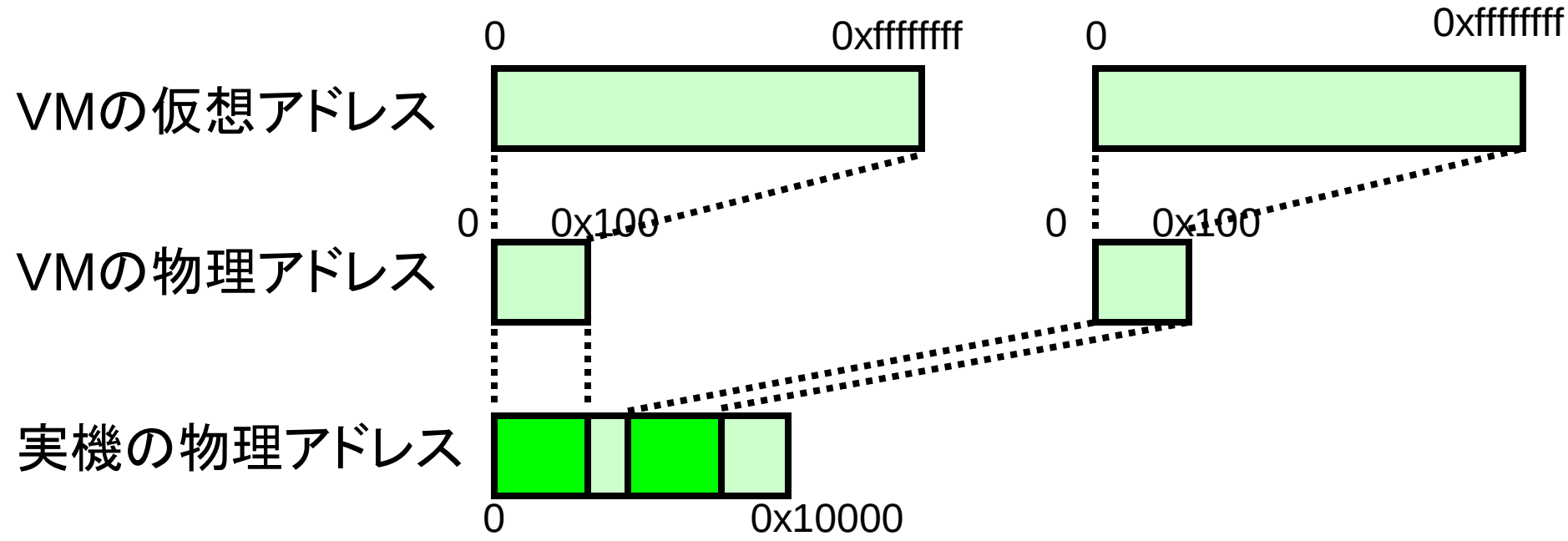
  ※VMRUN実行時に選択可能

VMRUN命令の使用例

```
for (;;) {

    /* VMM からVMに制御を移す */
    VMRUN;

    /* #VMEXIT が発生 */

    switch (error code) {
        /* sensitive命令や例外のエミュレーション */
    }
}
```

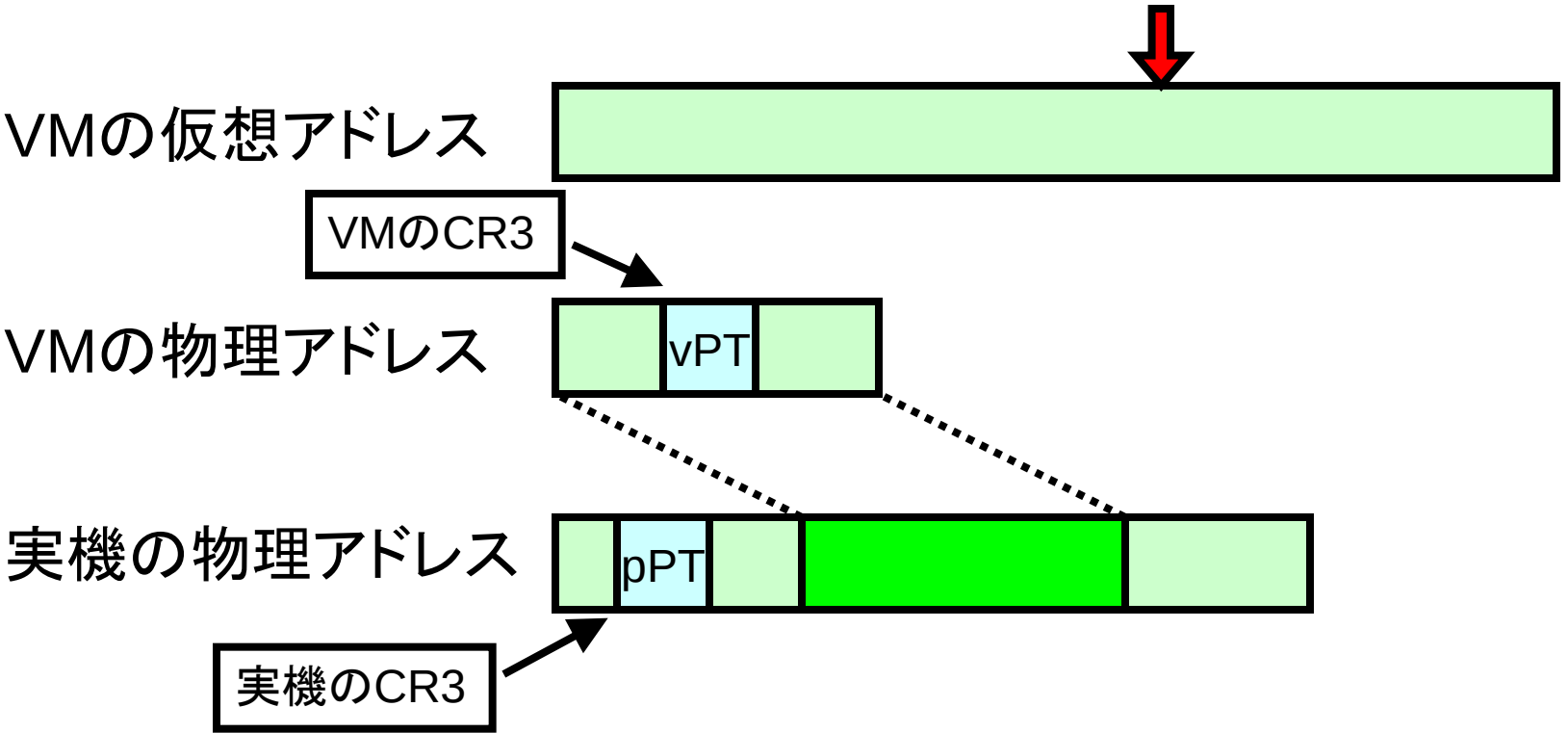
Nested Pagingとは？

- VMの物理アドレスを実機の物理アドレスへ対応付ける機構
  - 複数のVMにそれぞれ独立した物理アドレス空間を提供できる



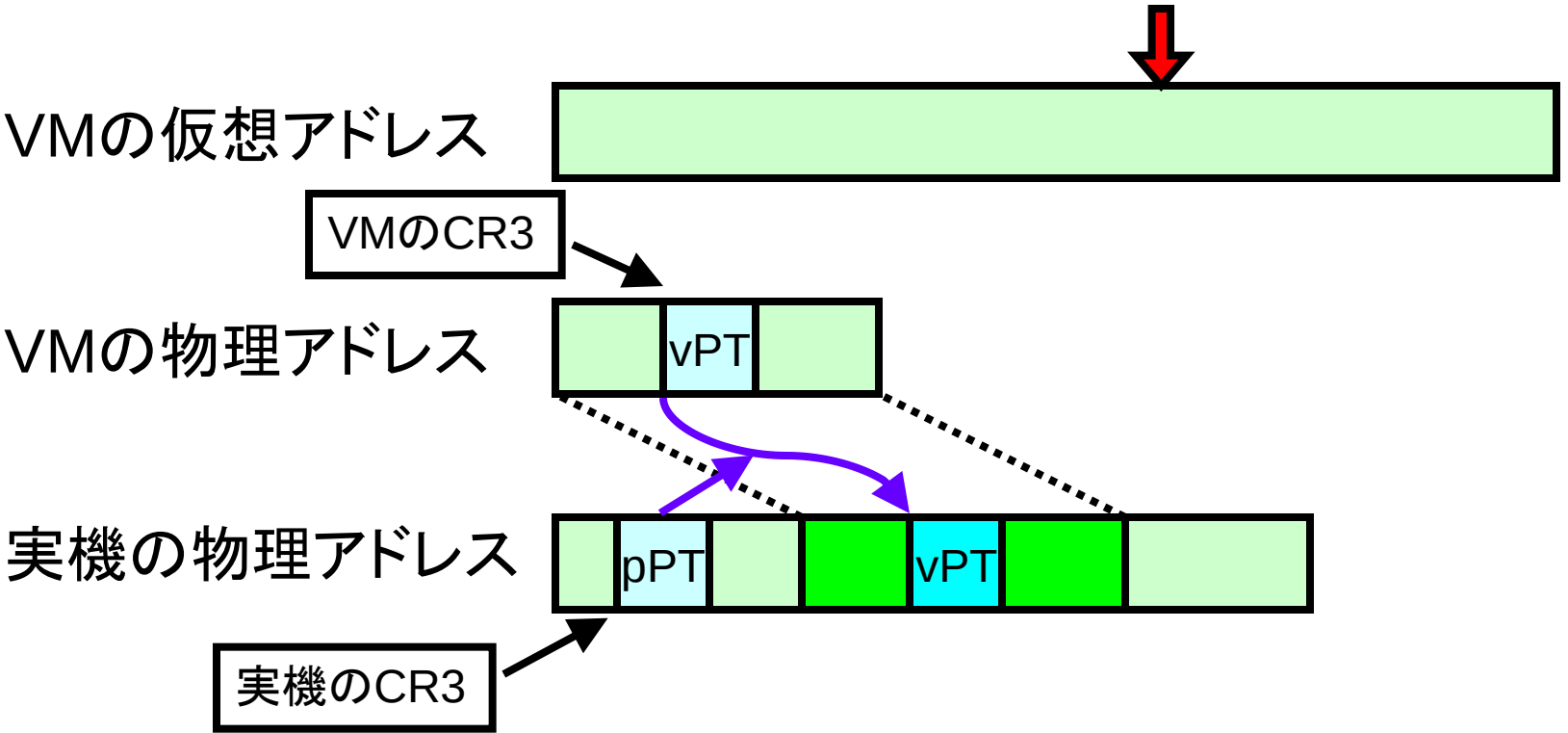
Nested Paging有効時のアドレス変換 (1/4)

- 2種類のページテーブルを利用
  - VMの仮想アドレス ➔ VMの物理アドレス
  - VMの物理アドレス ➔ 実機の物理アドレス



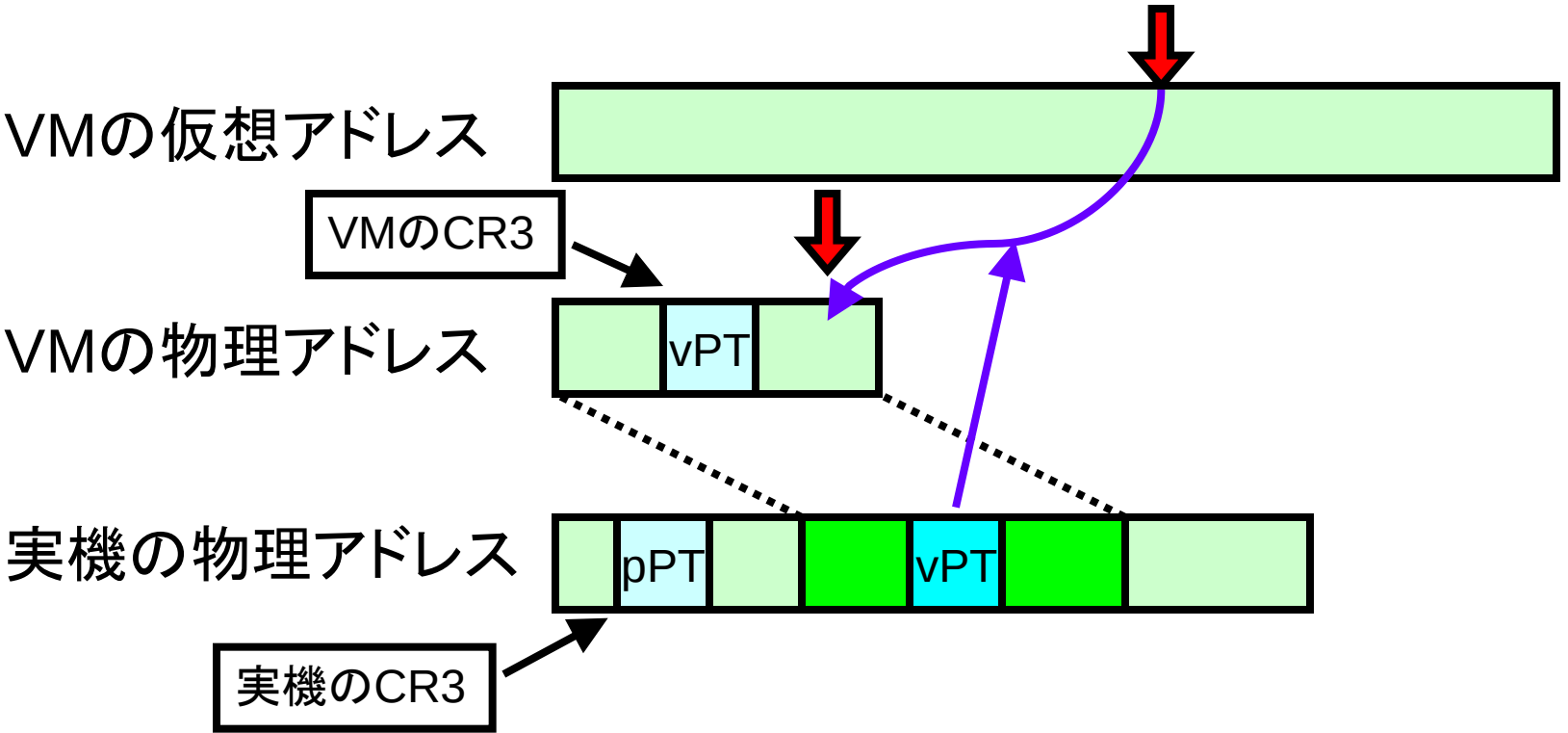
Nested Paging有効時のアドレス変換 (2/4)

- 2種類のページテーブルを利用
  - VMの仮想アドレス ➔ VMの物理アドレス
  - VMの物理アドレス ➔ 実機の物理アドレス



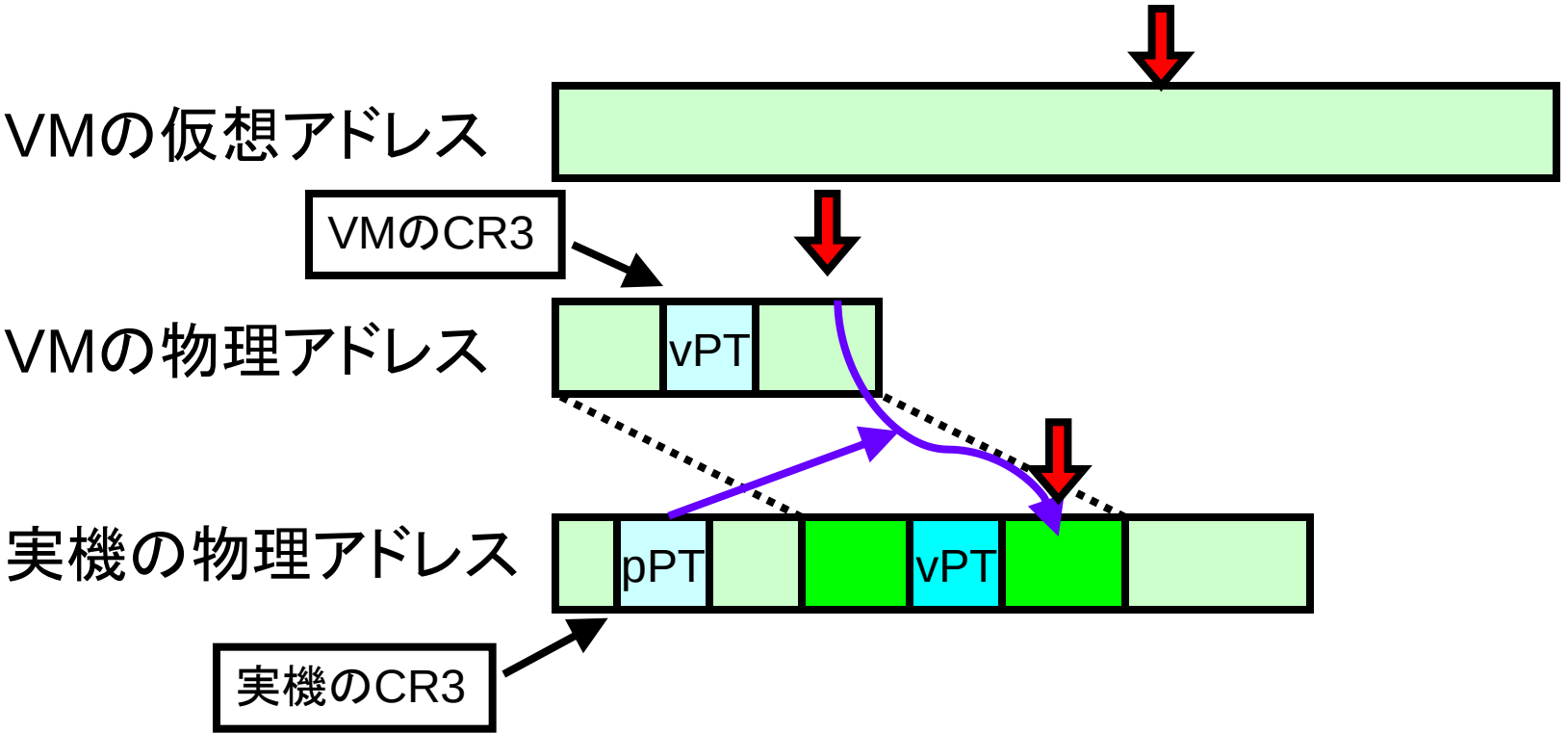
Nested Paging有効時のアドレス変換 (3/4)

- 2種類のページテーブルを利用
  - VMの仮想アドレス ➔ VMの物理アドレス
  - VMの物理アドレス ➔ 実機の物理アドレス



Nested Paging有効時のアドレス変換 (4/4)

- 2種類のページテーブルを利用
  - VMの仮想アドレス ➔ VMの物理アドレス
  - VMの物理アドレス ➔ 実機の物理アドレス

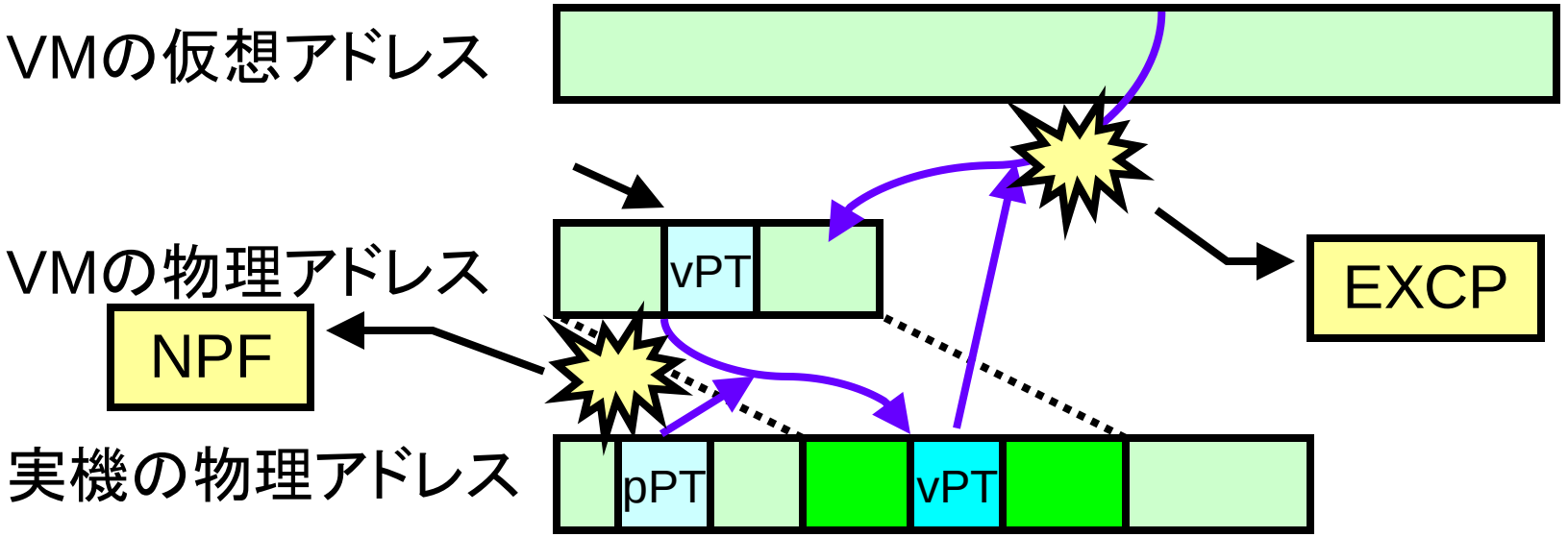


ページングレジスタの仮想化

- CR0、CR3などの制御レジスタの複製を生成
- VMRUN実行後、制御レジスタへのアクセスは、その複製へのアクセスに**自動的に**変換

ページフォルトの仮想化

- どの段階でのフォルトかに応じて、#VMEXITのエラーコードが異なる



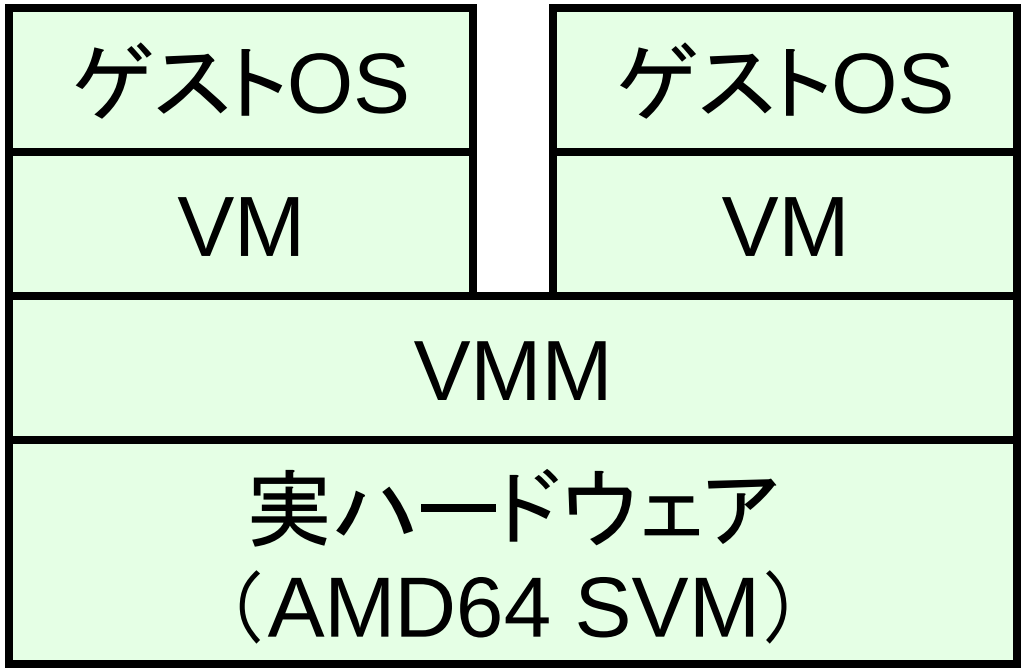
AMD64が提供するその他の命令

- VMSAVE、VMLOAD
  - VMRUNでは保存・復元されない状態を保存・復元
    例）セグメントレジスタのキャッシュ
- VMMCALL
  - 明示的にVMからVMMに切り替える
- STGI、CLGI
  - Global Interrupt Flag (GIF)のセット・クリア
    - VMRUNの前後で実行

(Part 2)
VMMの作り方

VMMのアーキテクチャ

- VMMは、実ハードウェア上に直に存在



VMMの動作の流れ

1. ハードウェアの初期化
   - OSの起動時の処理とほぼ同様
2. VMの作成
   - Nested pagingの設定など
3. ゲストOSの起動
   - VMRUN命令を実行

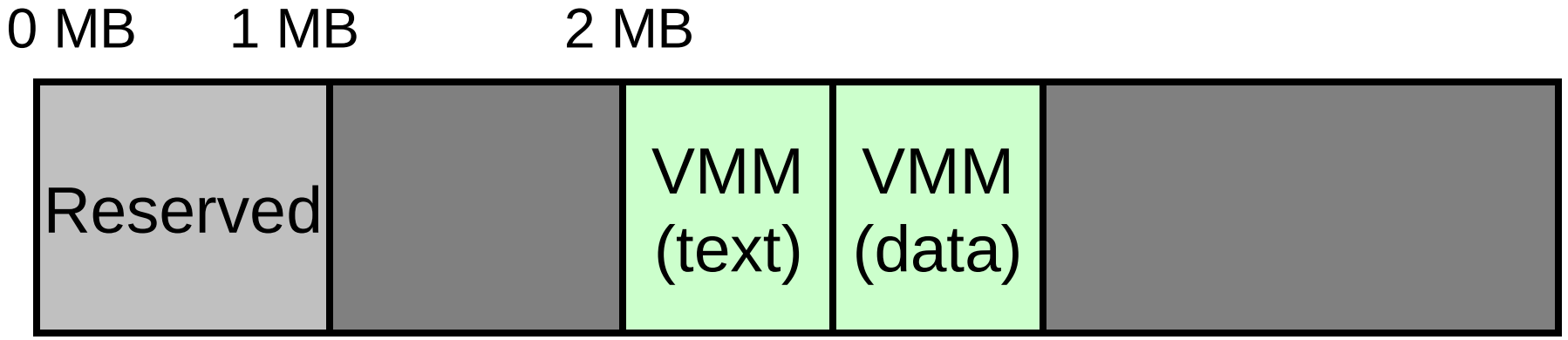
以降の発表では、
それぞれの処理を
順に説明していく

1．ハードウェアの初期化

- メモリを初期化する
  - ページテーブル、セグメントデスクリプタを初期化
  - ページのアロケーターを初期化
- CPUを初期化する

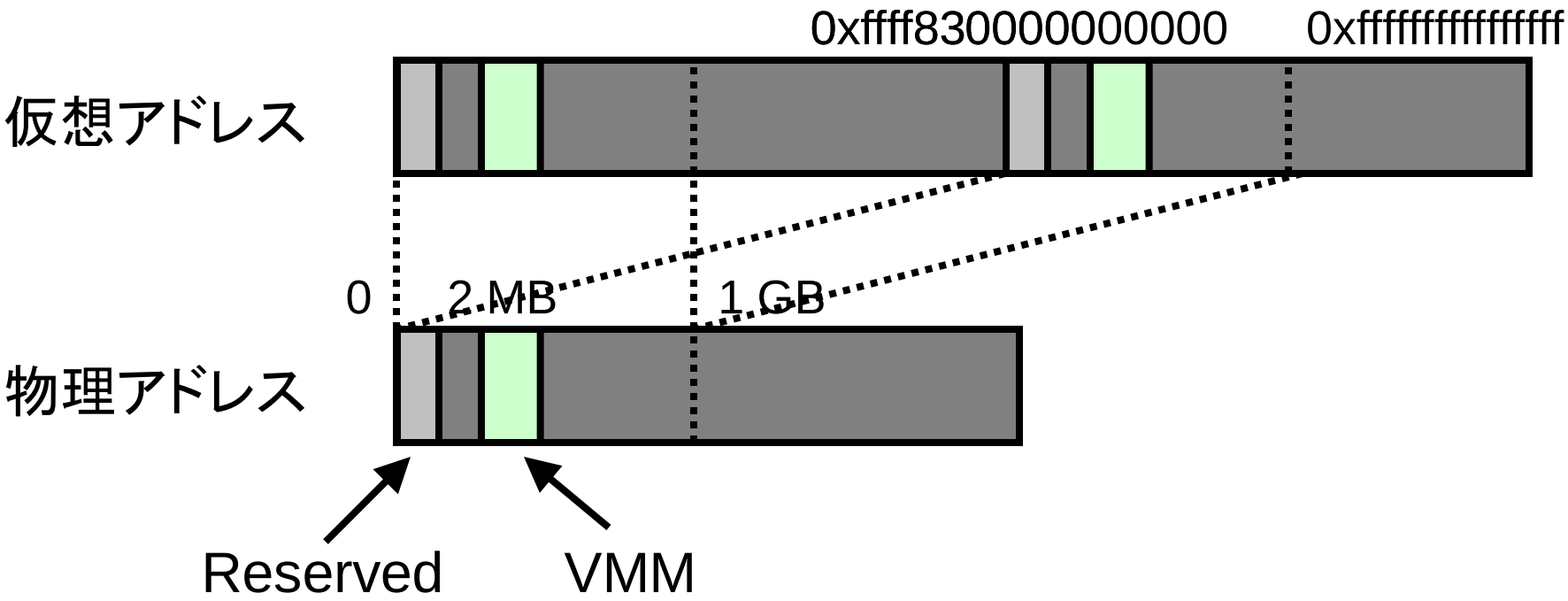
VMMの起動

- GRUBによってVMMが物理メモリ上にロードされる



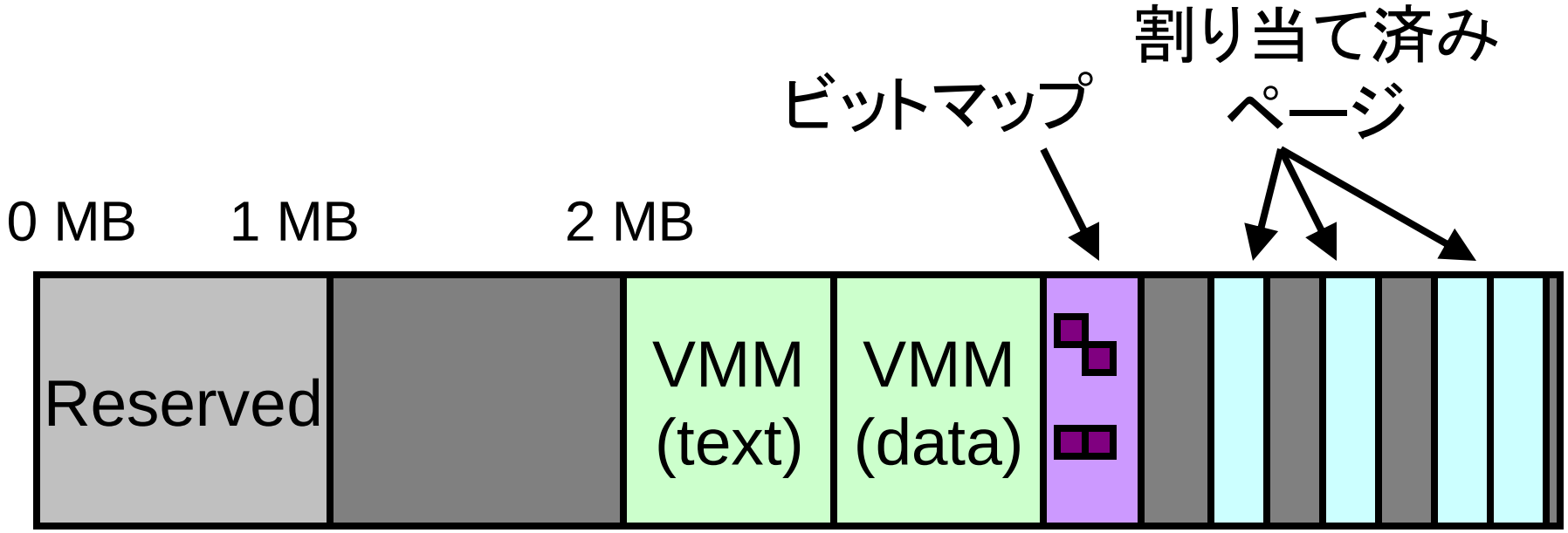
ページテーブル、セグメントデスクリプタの初期化

- ページングを有効にする
  - 64ビットモード
  - 物理アドレスの0~1 GBを、2つの仮想アドレスにマップ



ページのアロケーターの初期化

- 物理ページを動的に確保・解放できるようにする
  - GRUBから、使用可能なメモリ領域を教えてもらう
  - ページの使用・未使用を記憶するビットマップを用意



CPUの初期化

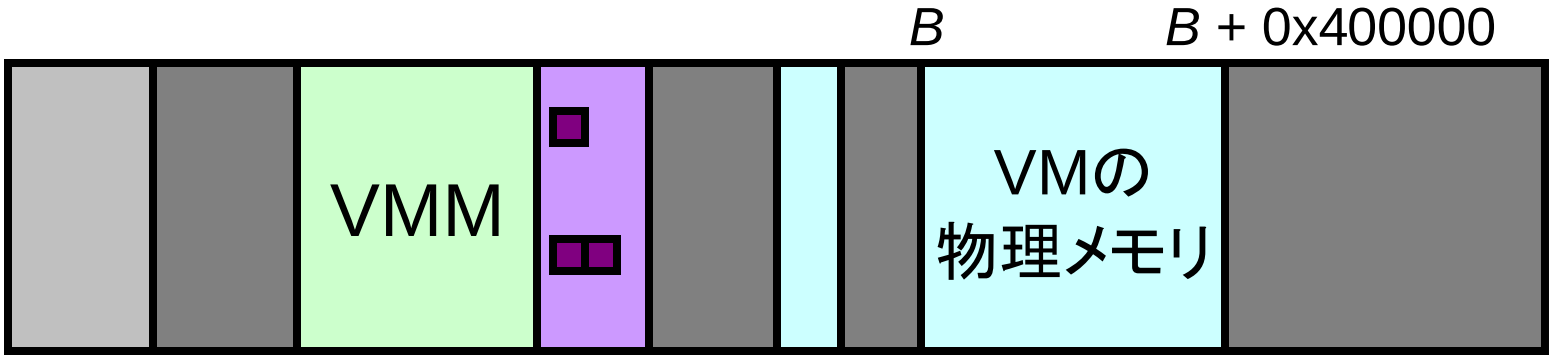
- 以下を検査する
  - AMDアーキテクチャか？
  - 仮想化機能をもつか？
- 可能であれば、仮想化機能を有効にする
  - VMMの状態を保存するための領域を確保
    - 前述のページアロケーターを使用して
  - そのページの物理アドレスをMSRレジスタに格納

2. VMの作成

- VMの物理メモリを確保
- ゲストOSをロード
- Nested pagingを設定
- レジスタを初期化

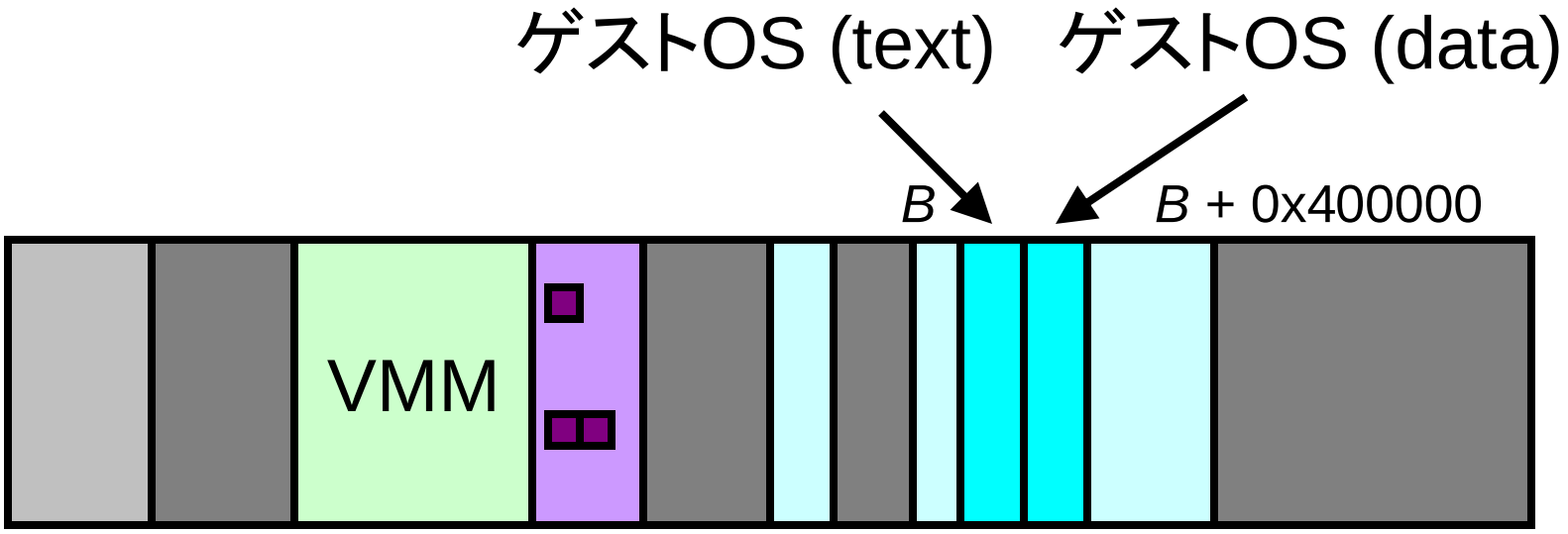
VMの物理メモリの確保

- VMの物理メモリとして使用する領域を実マシンの物理メモリ上に確保する
  - 前述のアロケーターを使用



ゲストOSのロード

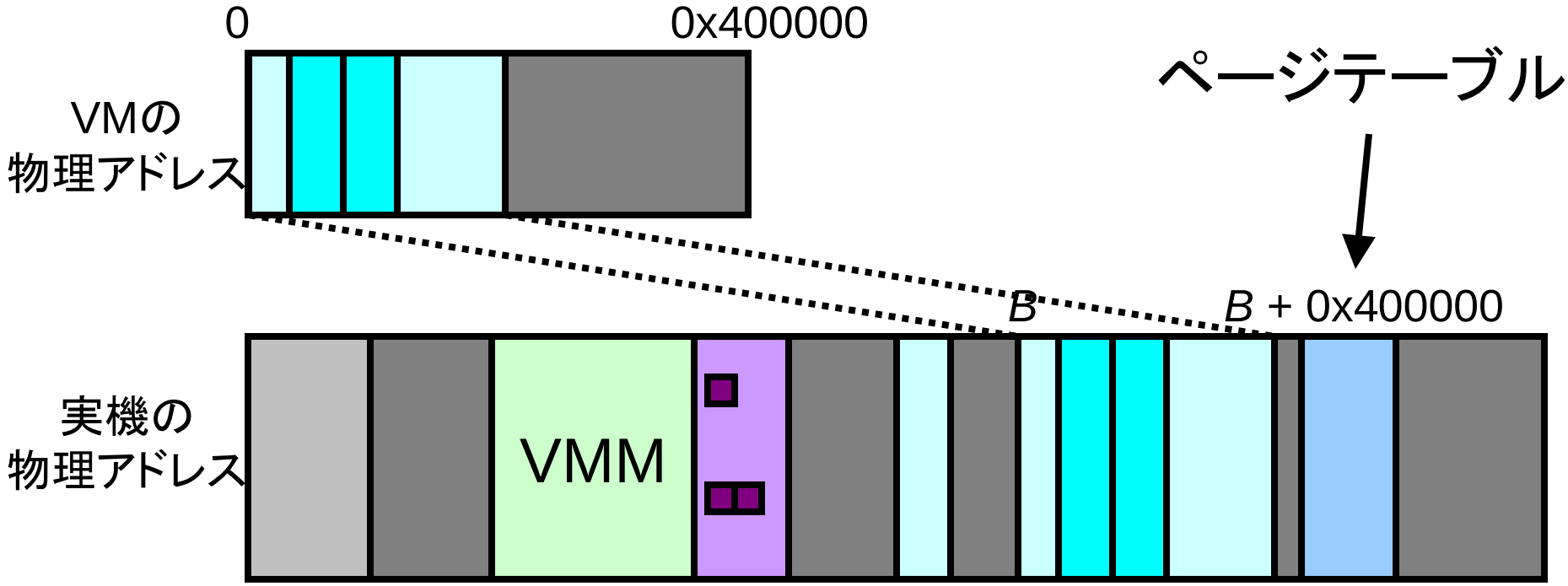
- ELF形式を解釈し、それに従って、
  カーネルをVMの物理メモリ上に配置



Nested Pagingの設定

- ページテーブルを作成する
  - VMの物理アドレス (x) ➔ 実機の物理アドレス ($B + x$)

    ※ 実際の実装では、VGAの関係上、少し複雑



レジスタを初期化

- ELFを解釈し、RIPを設定する
- …

3. ゲストOSの起動

- VMRUN命令を実行
- #VMEXITが発生したらエラーコードを表示

未実装なVMMの機能

- 割り込み
- I/Oデバイスへのアクセス
- 複数の仮想マシンの同時起動
- …

※ VM上でWindowsやLinuxを起動するには、
まだまだ多くの機能の実装が必要

関連研究

- Xen virtual machine monitor
  - 大規模すぎて、理解・改良が困難
    - 複数アーキテクチャへ対応させるため仕方がない面も
  - Nested pagingには未対応

- Research Hypervisor
  - http://www.research.ibm.com/hypervisor/

まとめ

- AMD64の仮想化技術を利用した（非常に単純な）仮想マシンモニタを実装した
  - http://web.yl.is.s.u-tokyo.ac.jp/~tvmmにて公開
  - 使ってみたい方は是非

その他

SimNow

- AMD64上で動作するAMD64のシミュレータ
  - 豊富なデバッグ機能を提供する
    例）レジスタ・メモリのダンプ、実行のトラップ
  - シミュレーション速度は非常に遅い

雑多な感想

- 数多くのアドレッシングモードがあり煩雑
  - Long mode (64-bit mode, Compatibility mode), Legacy mode (Physical-address extension, Page-size extension), …
  - モードを制御するフラグが、コントロールレジスタ・セグメントデスクリプタ・ページテーブルなどに点在
- Webから取得可能な情報がまだ少ない

参考文献 (1/2)

- X86-64 Linux
  - http://www.x86-64.org
- AMD64 Architecture Programmer's Manual
  - http://www.amd.com/us-en/Processors/TechnicalResources/0,,30_182_739_7044,00.html
  - Volume 3: General-Purpose and System Instructions, Section 15: Secure Virtual Machine

参考文献 (2/2)

- SimNow
  - http://developer.amd.com/login.aspx?msg=simnow
- Xen virtual machine monitor
  - http://www.cl.cam.ac.uk/Research/SRG/netos/xen
- Multiboot Specification Manual
  - http://www.gnu.org/software/grub/manual/multiboot