Xen 3.0のすべて
内部実装詳解

VA Linux Systems Japan K.K.
山幡　為佐久<yamahata@valinux.co.jp>
Linux Kernel Conference 2005.11.11

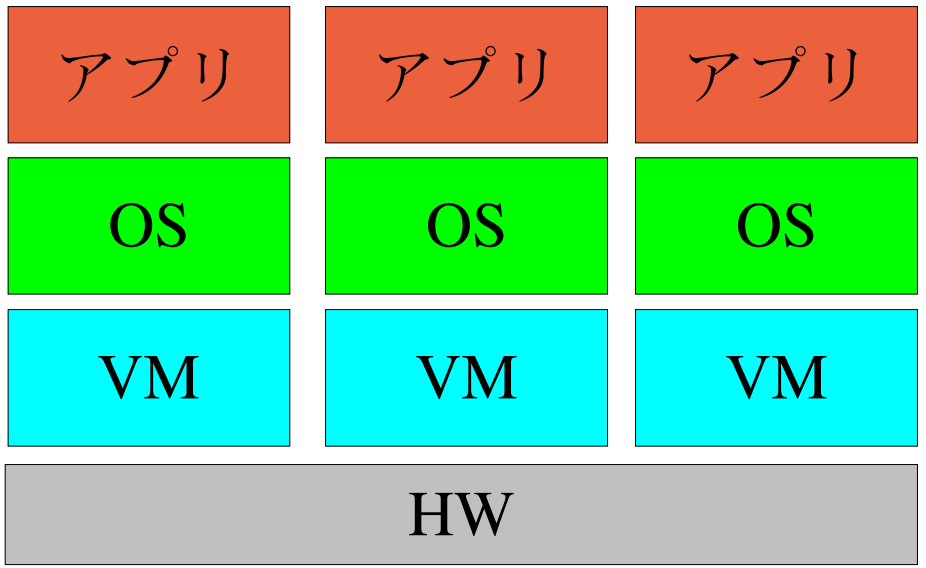
目次

- イントロダクション
- Xen概要
- ドメイン管理
- 時間管理とCPUスケジューラ
- 割り込み/例外処理の仮想化
- I/O デバイス仮想化
- MMU仮想化
- 完全仮想化
- 今後の展望

イントロダクション

仮想化とは

- マシン上で複数のVirtual Machineを動作させる技術。
  - VM上でOSを動作させる
- 古くは1960年代IBMメインフレームの頃に遡る



仮想化手法の分類

- 完全仮想化(full virtualization)
  - マシンを完全にエミュレートする
  - オーバーヘッド大
  - OSがそのまま動作
    - 実際はx86アーキテクチャ上完全な仮想化は難しくバイナリパッチをあてるなどする
  - 例) bochs, qemu, vmware, virtualpc
- 準仮想化(para virtualization)
  - 仮想化された環境を提供
  - ゲストOSに修正が必要
    - ゲストOSのソースコードが必要
    - アプリケーションは修正不要
  - 例) Xen, User Mode Linux(UML)

Xen3.0新機能

- SMP対応
  - ゲストOS SMP化
- 64bit対応(x86_64)
- 完全仮想化
  - VT-x(Virtualization Technology for IA32)
- 管理ツール、デバイスドライバ用フレームワーク
  - XenBus
  - XenStore

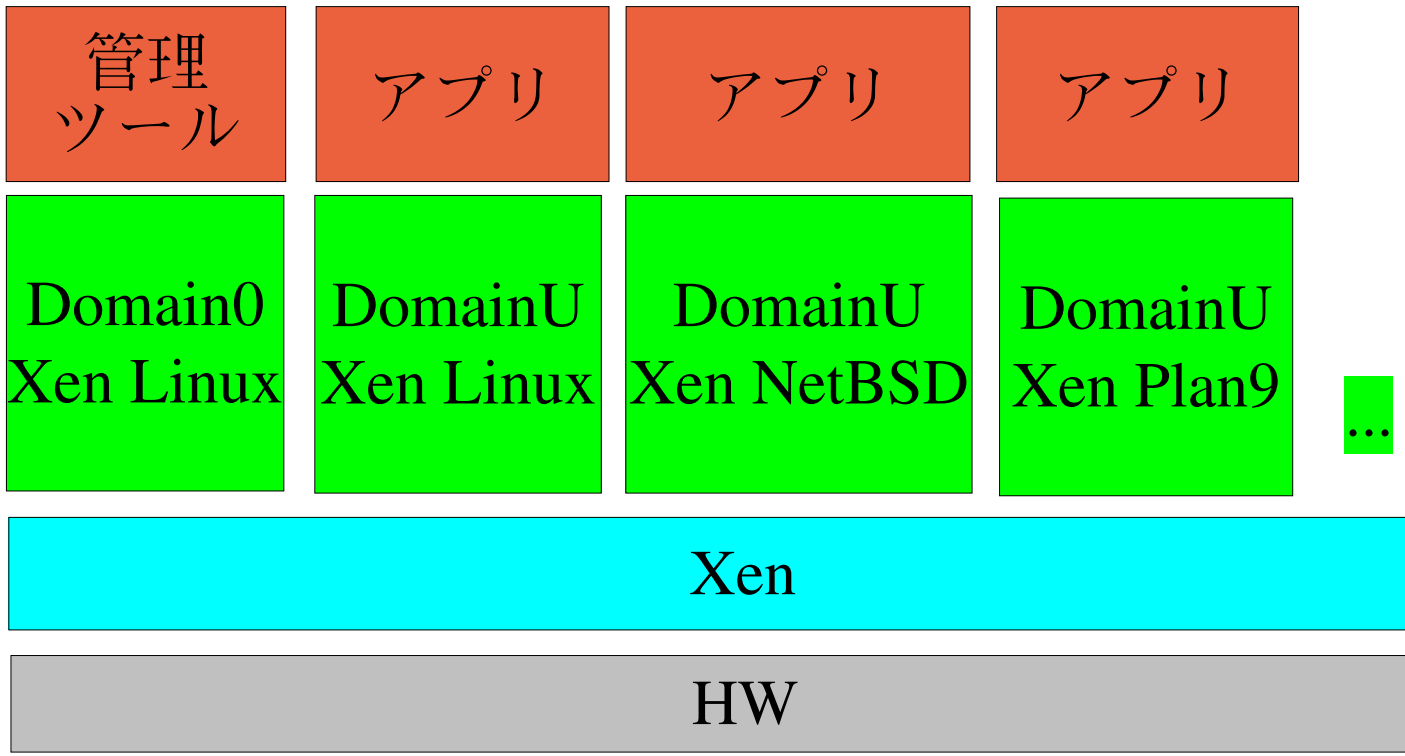
Xen概要

以降はIA32アーキテクチャに限定

ドメイン

- XenではVMをドメインと呼ぶ
  - 特権あり、特権なし
- Domain0
  - 特権あり
  - 起動時に必ず起動
  - 管理ツールが動作
  - 実デバイスの制御
- DomainU
  - ユーザが必要に応じて作成/破壊
  - Xen仮想デバイスを使用

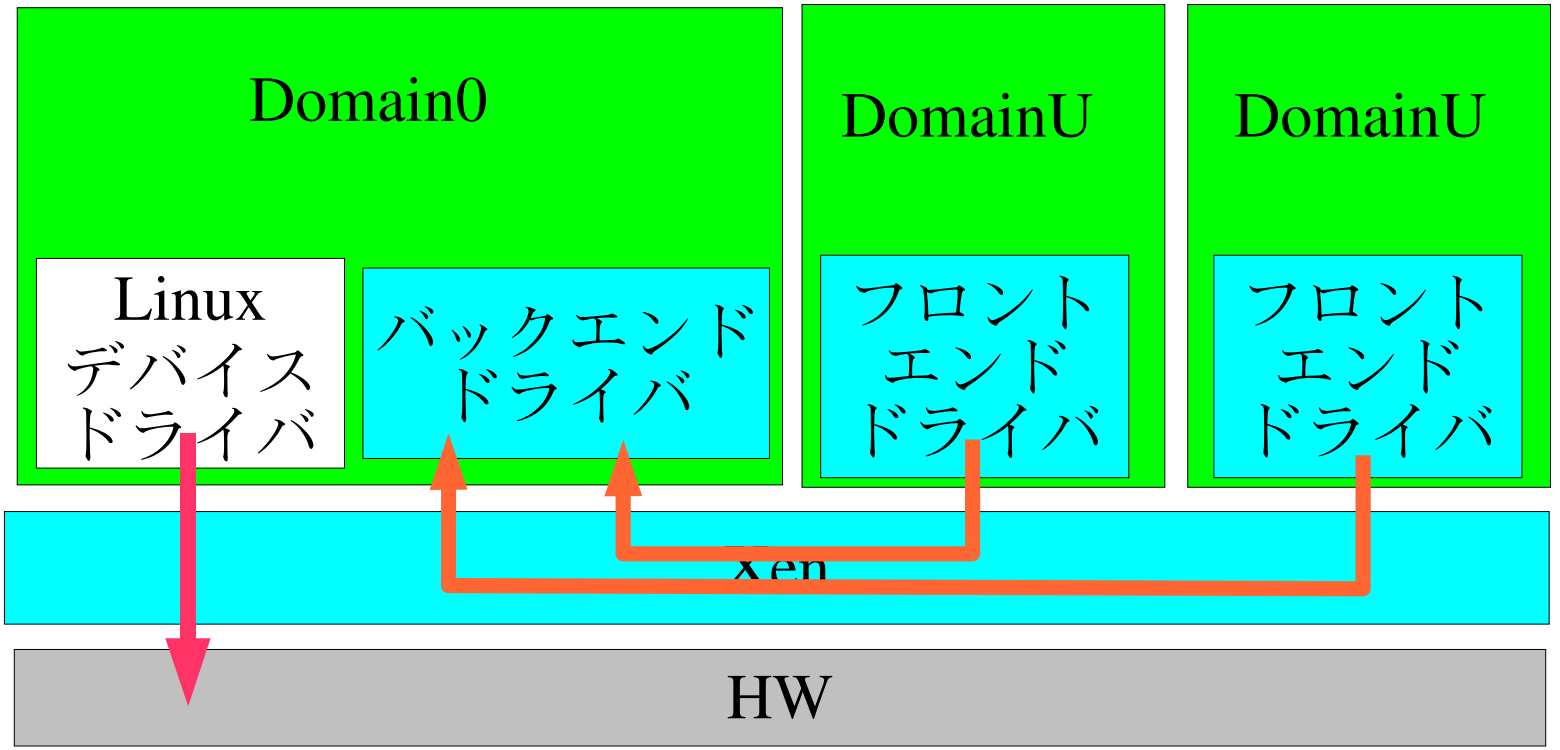
準仮想化(para-virtualization)



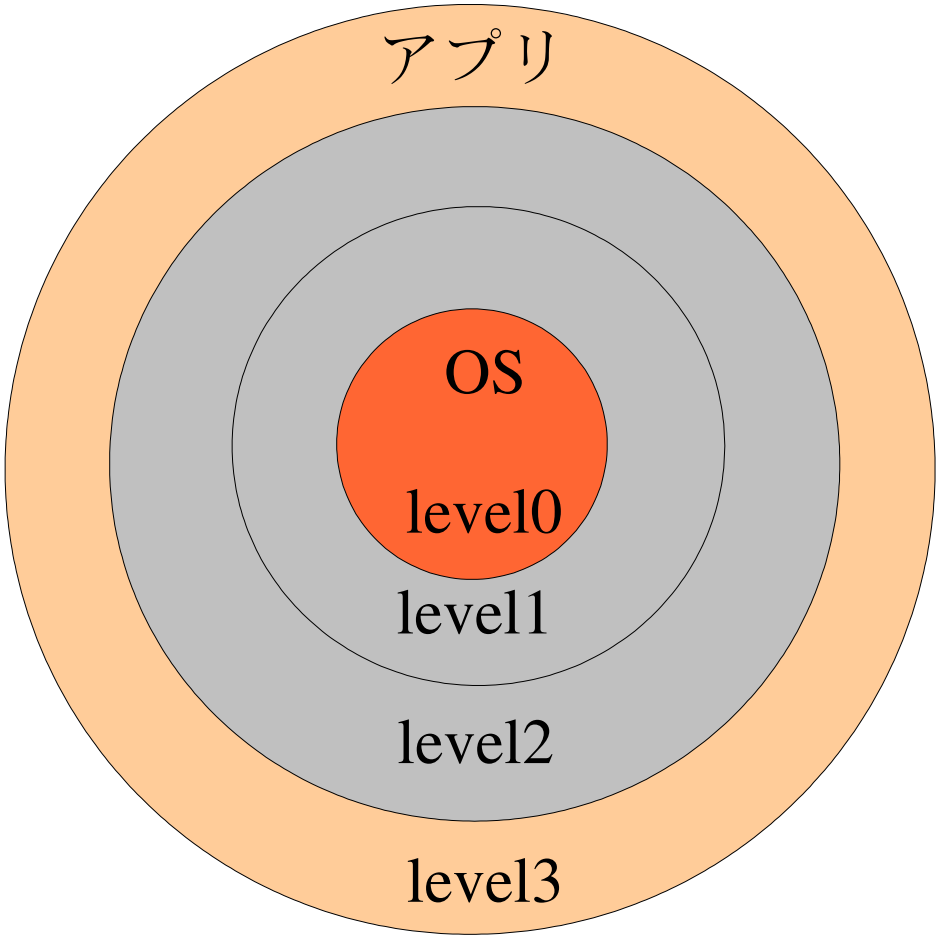
アプリケーションは
修正不要

Xen環境向けに
修正された
ゲストOSが動作する

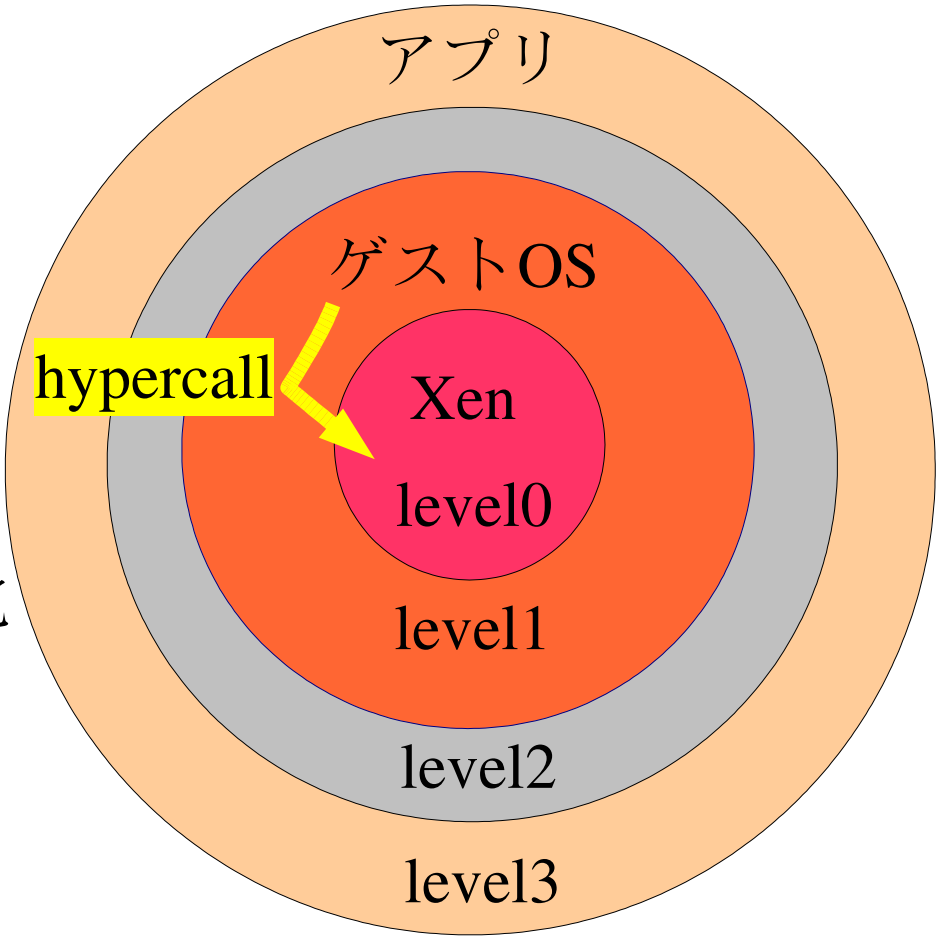
準仮想化(para-virtualization)(cont.)
デバイスドライバモデル



特権レベル



Xen化

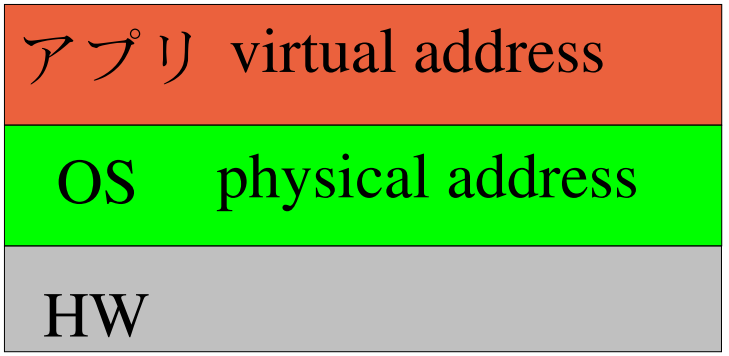


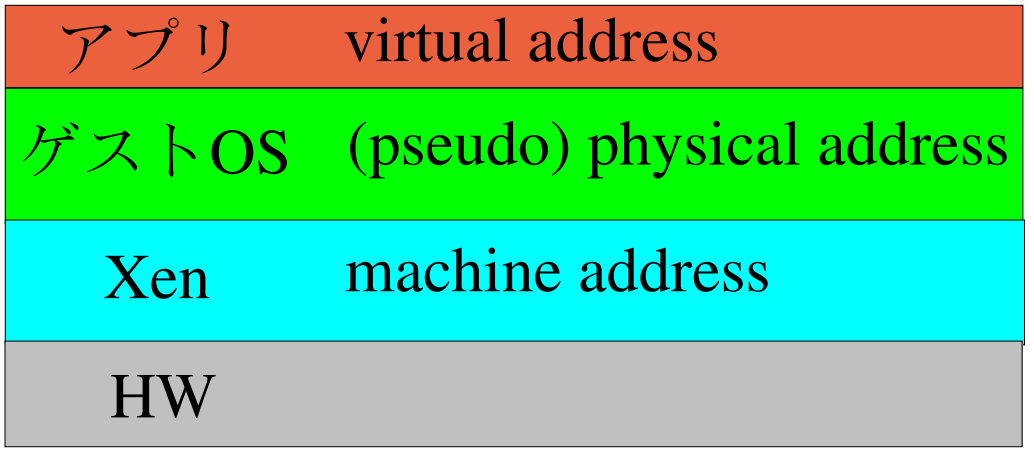
通常OSは特権レベル0で動作

Xenが特権レベル0で動作
OSは特権レベル1で動作
ゲストOSからXenを保護は
セグメント機能を使用

マシンアドレス

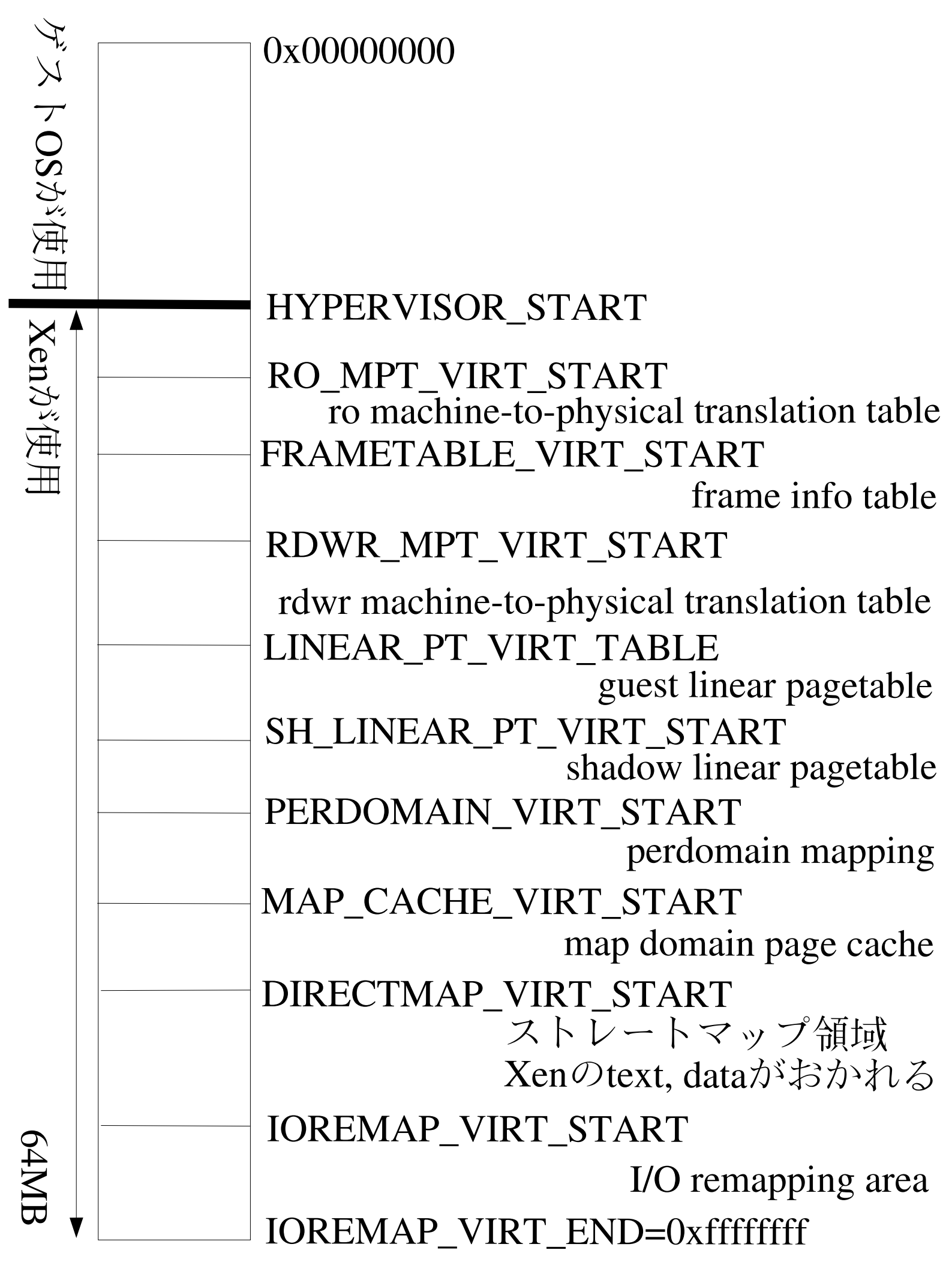
- ゲストOSはXenが管理する仮想アドレス空間上で動作する。
- XenはゲストOSが物理アドレス上で動作している幻想を与える
  - ゲストOSが実際に扱うのは疑似物理アドレス
- Xenが管理している本当の物理アドレスをマシンアドレスと呼び区別する。





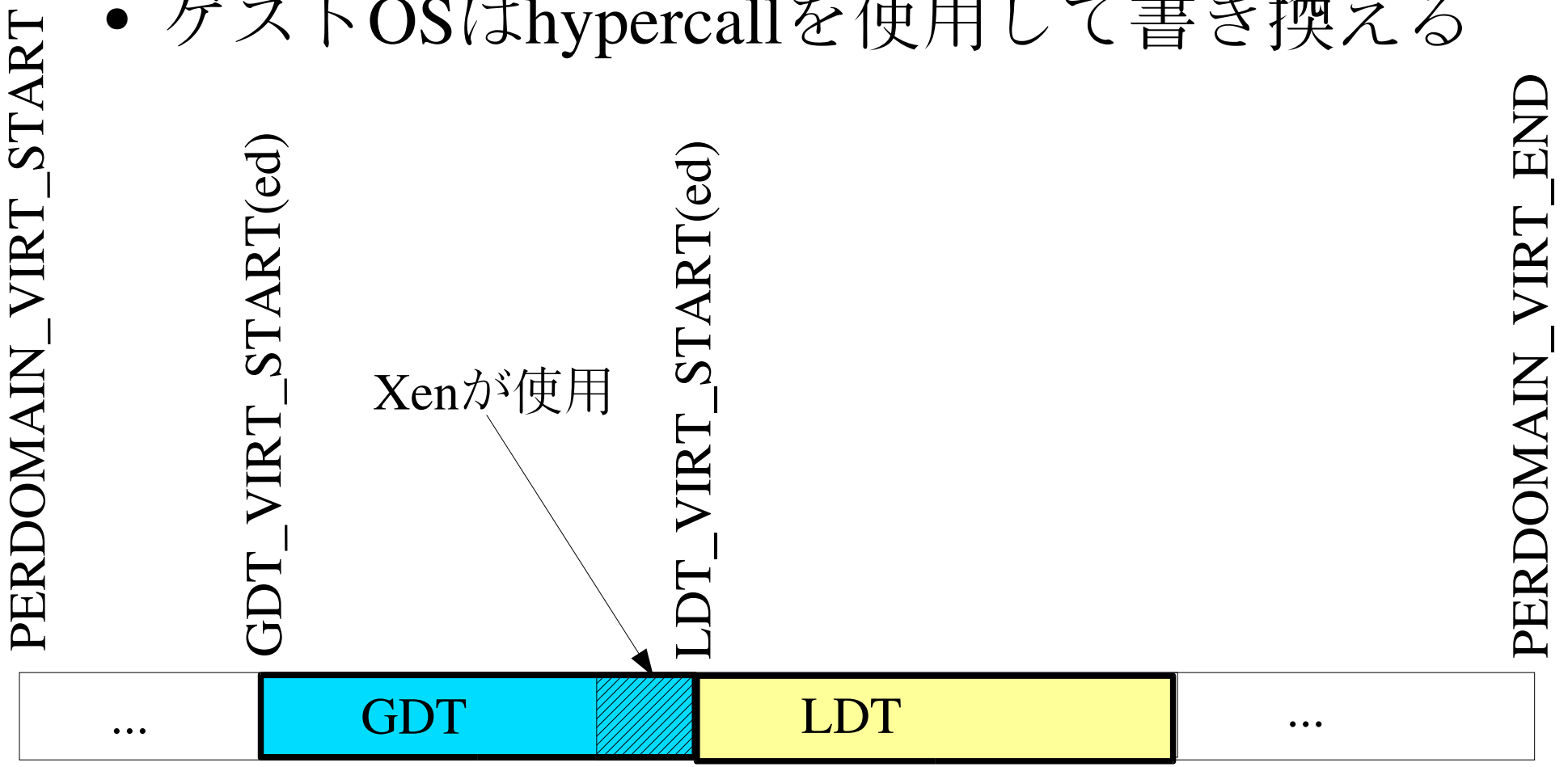
空間レイアウト

- Xenは仮想アドレスの高位64MBを使用する(32bitの場合)
  - ゲストOSは上位64MBを使用しないように修正される
- マシンアドレス->物理アドレス変換テーブルを持ちXenはread-writeでアクセスでき,ゲストOSはread-onlyでアクセスできるようマップする
  - machine-to-physical translation table
- 高位16MB – 4MB =12MBをストレートマップ
- 高位4MBはI/O用に予約

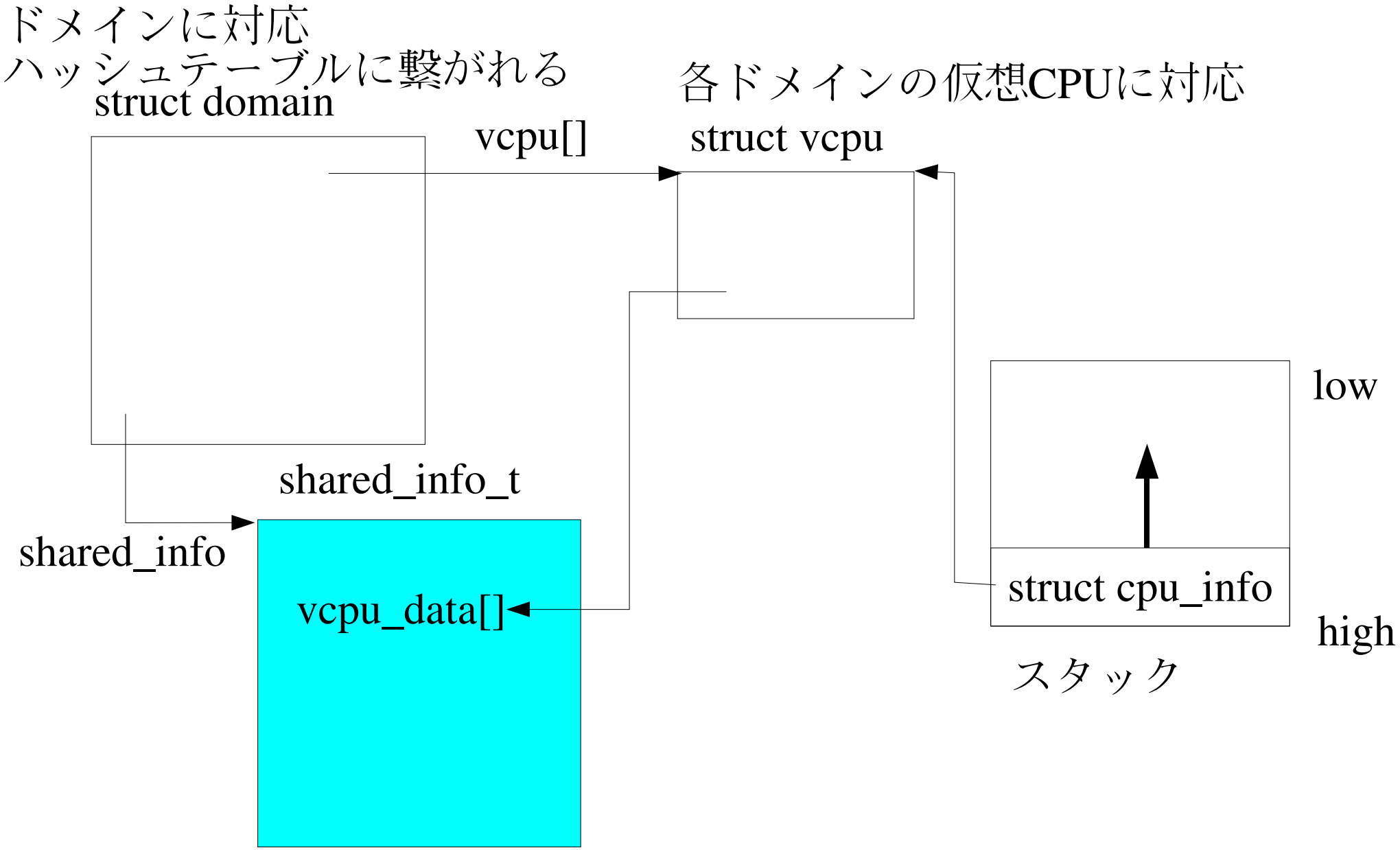

ゲストOSが使用
Xenが使用
64MB
0x00000000
HYPERVISOR_START
RO_MPT_VIRT_START
ro machine-to-physical translation table
FRAMETABLE_VIRT_START
frame info table
RDWR_MPT_VIRT_START
rdwr machine-to-physical translation table
LINEAR_PT_VIRT_TABLE
guest linear pagetable
SH_LINEAR_PT_VIRT_START
shadow linear pagetable
PERDOMAIN_VIRT_START
perdomain mapping
MAP_CACHE_VIRT_START
map domain page cache
DIRECTMAP_VIRT_START
ストレートマップ領域
Xenのtext, dataがおかれる
IOREMAP_VIRT_START
I/O remapping area
IOREMAP_VIRT_END=0xffffffff

GDT/LDTの仮想化
(セグメンテーションの仮想化)

- perdomain領域にvcpu毎GDT,LDT領域を確保
- GDTの後部をxenが使用
- ゲストOSはhypercallを使用して書き換える



ドメイン管理

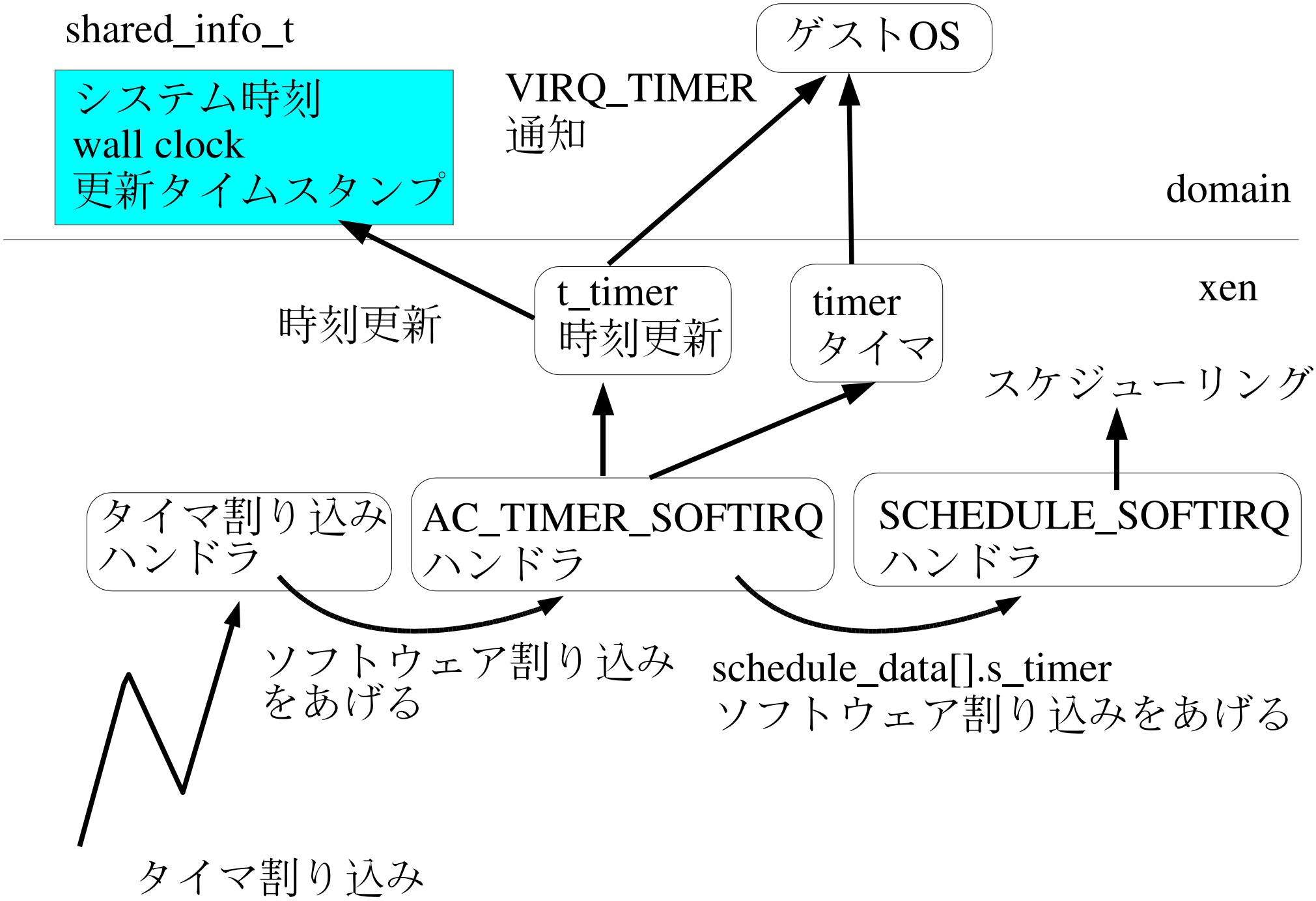


このページはゲストOSがアクセス可能なアドレスにmapする

時間管理とCPUスケジューラ

Xenのタイマ処理とソフトウェア割り込みと時間管理

- Xenのタイマ割り込み処理はAC_TIMER_SOFTIRQソフトウェア割り込みをあげるのみ
  - タイマソフトウェア割り込みで実際の処理
- ゲストOSは実タイマ割り込みを全て受け取れるわけではない
  - ゲストOSでのタイマティックのカウントが難しい
- ゲストOSがアクセスできるshared_info_tに時刻情報を含める。
  - ゲストOSはshared_info_tを読み取って時刻を知る

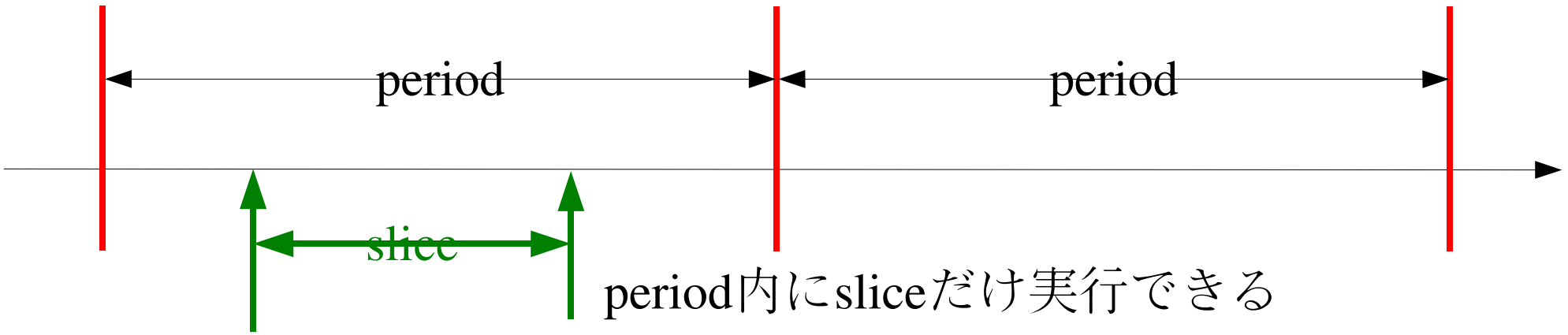

shared_info_t
システム時刻
wall clock
更新タイムスタンプ
ゲストOS
VIRQ_TIMER
通知
domain
xen
時刻更新
t_timer
時刻更新
timer
タイマ
スケジューリング
タイマ割り込み
ハンドラ
AC_TIMER_SOFTIRQ
ハンドラ
SCHEDULE_SOFTIRQ
ハンドラ
ソフトウェア割り込み
をあげる
schedule_data[].s_timer
ソフトウェア割り込みをあげる
タイマ割り込み

CPUスケジューラ

- SEDFとBVTの二つのスケジューラが提供されている。
  - Xen起動時に選択可能
  - SEDFがデフォルト
  - スケジューリングする単位はvcpu構造体
- SEDF: Simple Early Deadline First scheduler
- BVT: Borrowed Virtual Timer scheduler
- どちらも良く知られたリアルタイムスケジューリングを実装したもの
  - ゲストOSがCPUを手放すのはゲストOSのidle時のみと考えられる

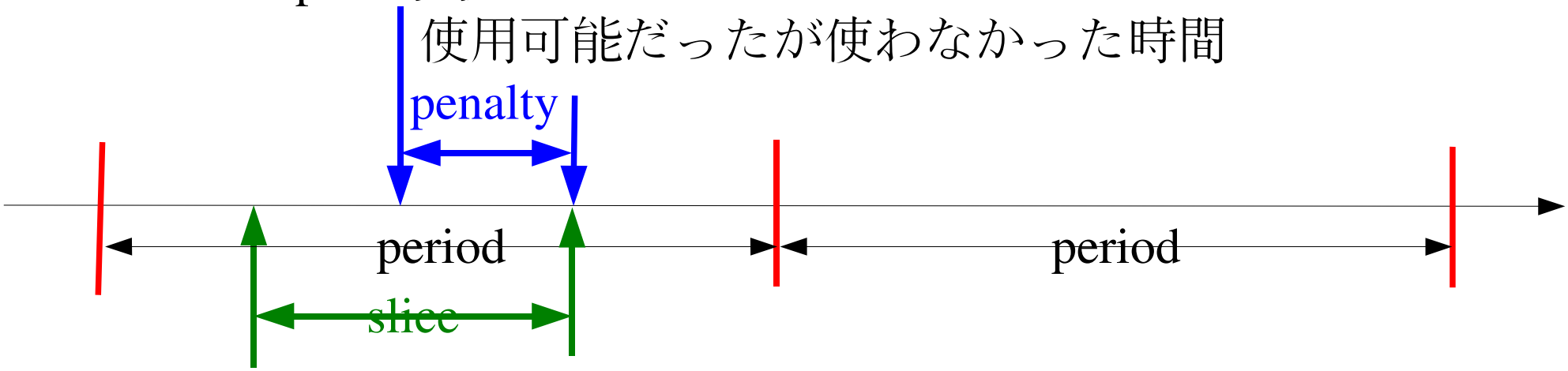
SEDFスケジューラ

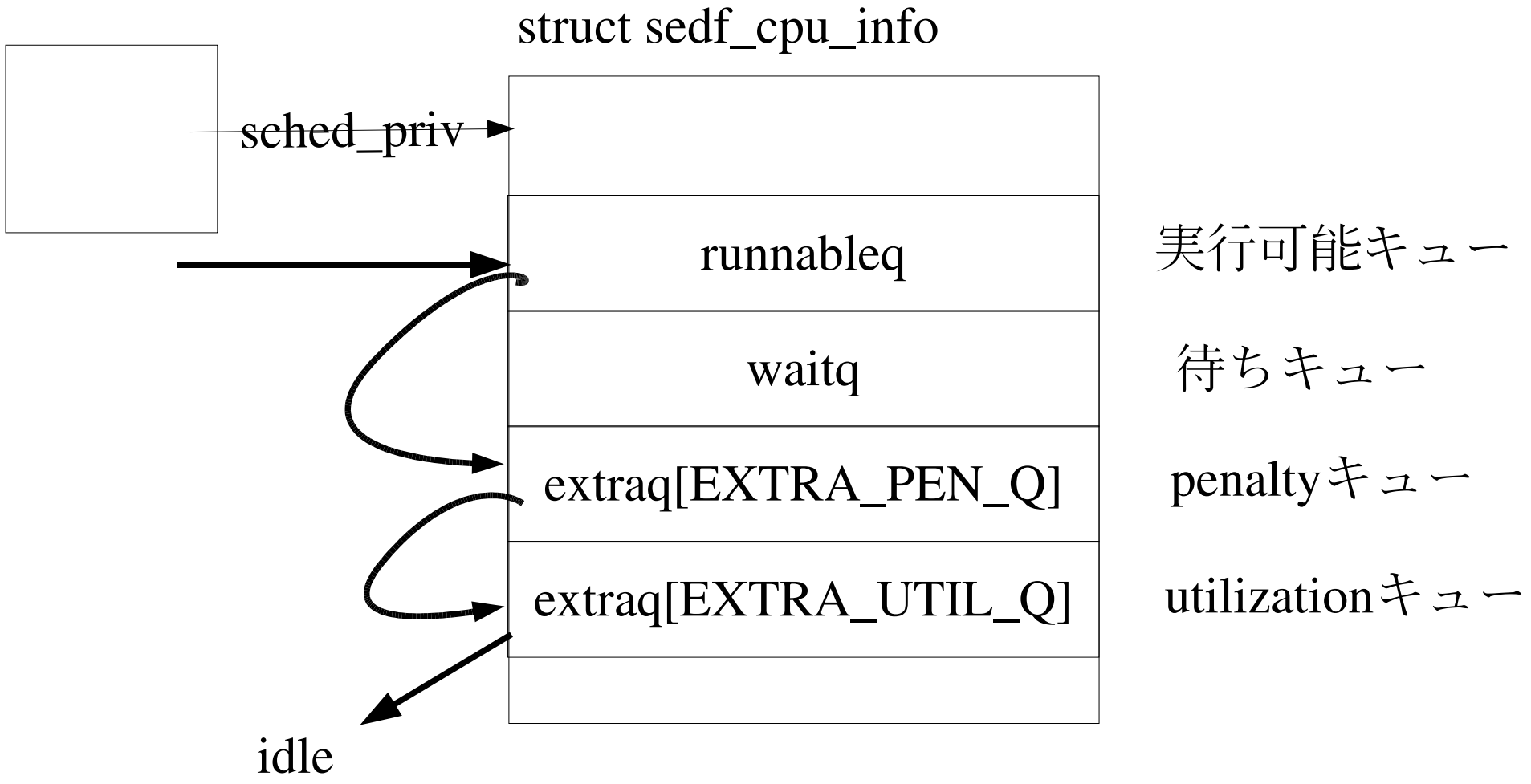
- それぞれのvcpuに一定期間(period)とその期間内に使用可能なcpu時間(slice)が与えられる。
  - 各vcpuへの重み付け(weight)に比例してsliceを決定
- デッドラインが一番短いものにcpuが与えられる



SEDFスケジューラ(cont.)

- extraキュー
  - idle時に実行するvcpuを優先度順に繋ぐ
- penalty: 使用しなかったCPU時間
  - CPU時間を使いきらずにCPUを手放した時の残り時間
- utilization: extraweight(ユーザが設定)で決定
  - extraweightが低いものを優先する

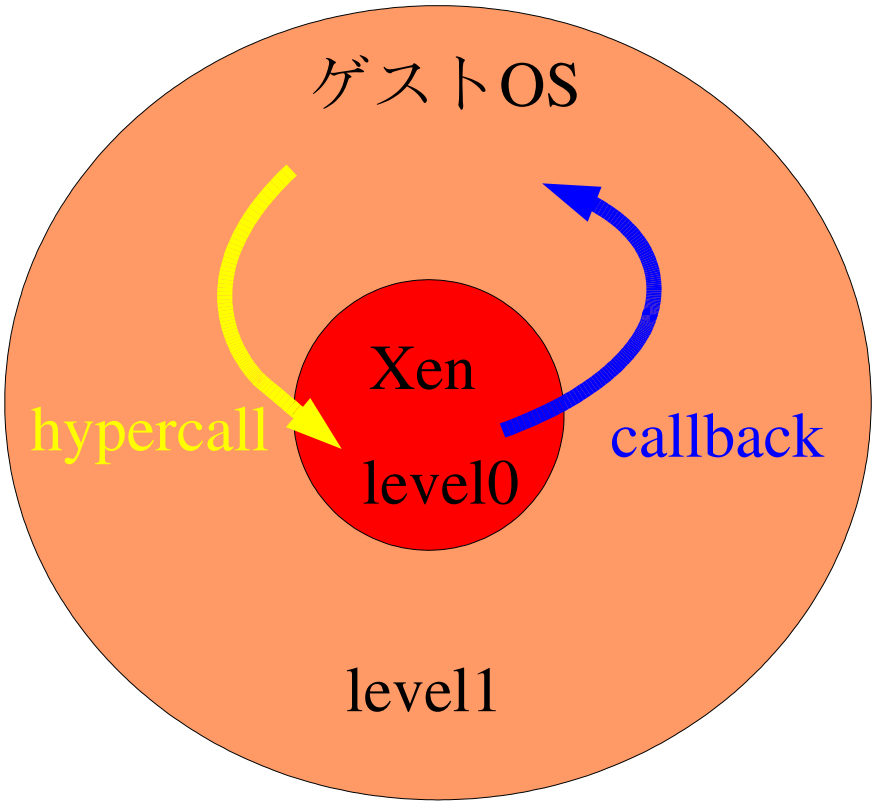




実行可能キューにvcpuが無ければextraqにあるものを実行する

割り込み/例外処理の仮想化

callback

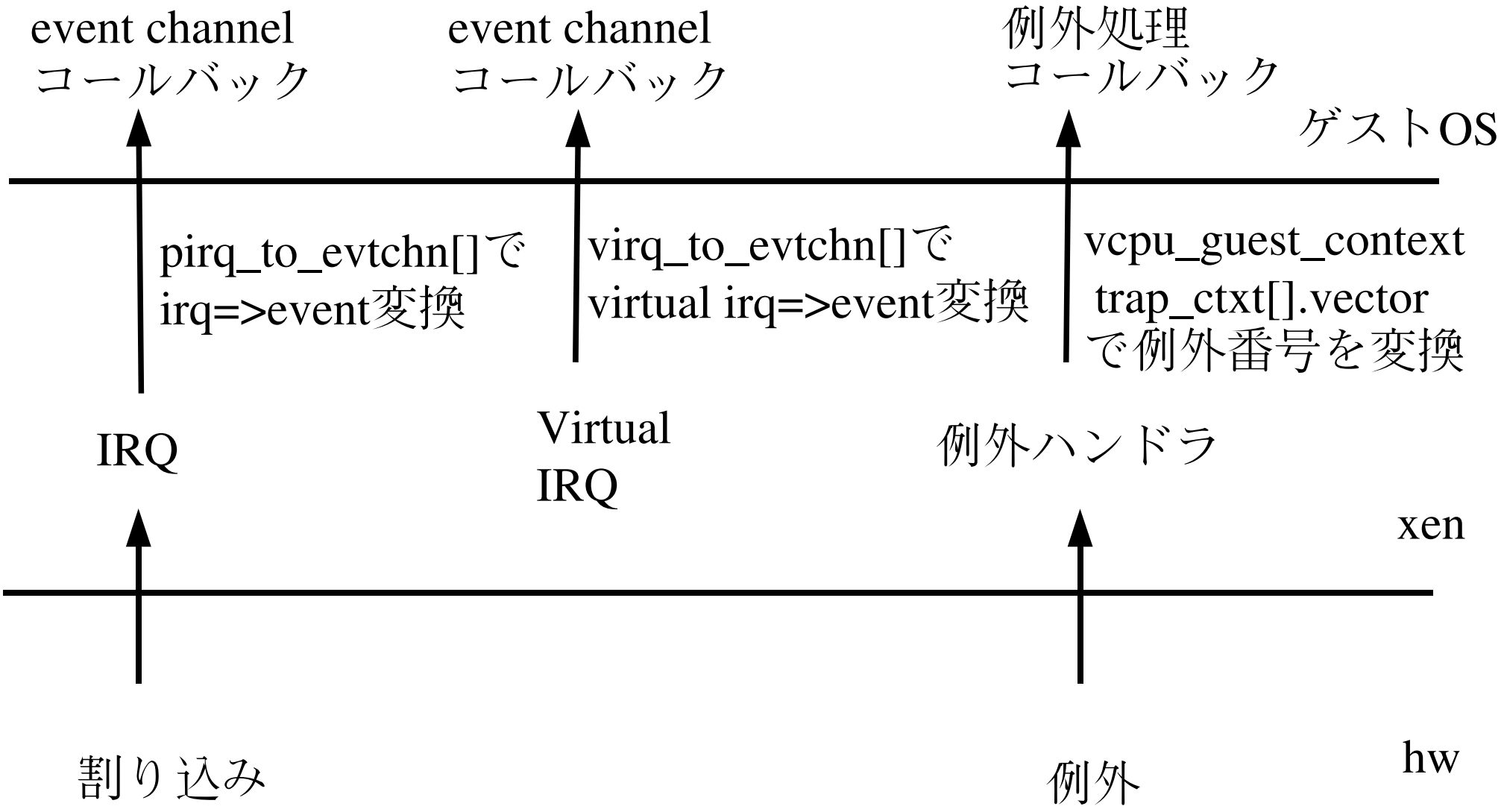


- XenからゲストOSへの呼び出し
- ゲストOS上のスタックへ待避レジスタを積み、実行ポインタを事前に登録されたアドレスにしてゲストOSへ制御を移す
- ゲストOSが処理すべき割り込み／例外処理とevent channel(後述)配送に使用

割り込み/例外の仮想化

- 必要があればXenはゲストOSに割り込み/例外を通知する
- 実割り込み/仮想割り込み
  - event channelを通じてゲストOSに通知
- 例外処理
  - ゲストOSが登録したハンドラをcallbackする
- システムコール
  - Xenが介入せずにゲストOSが直接呼び出されるようにIDTを設定する。

割り込み/例外の仮想化(cont.)

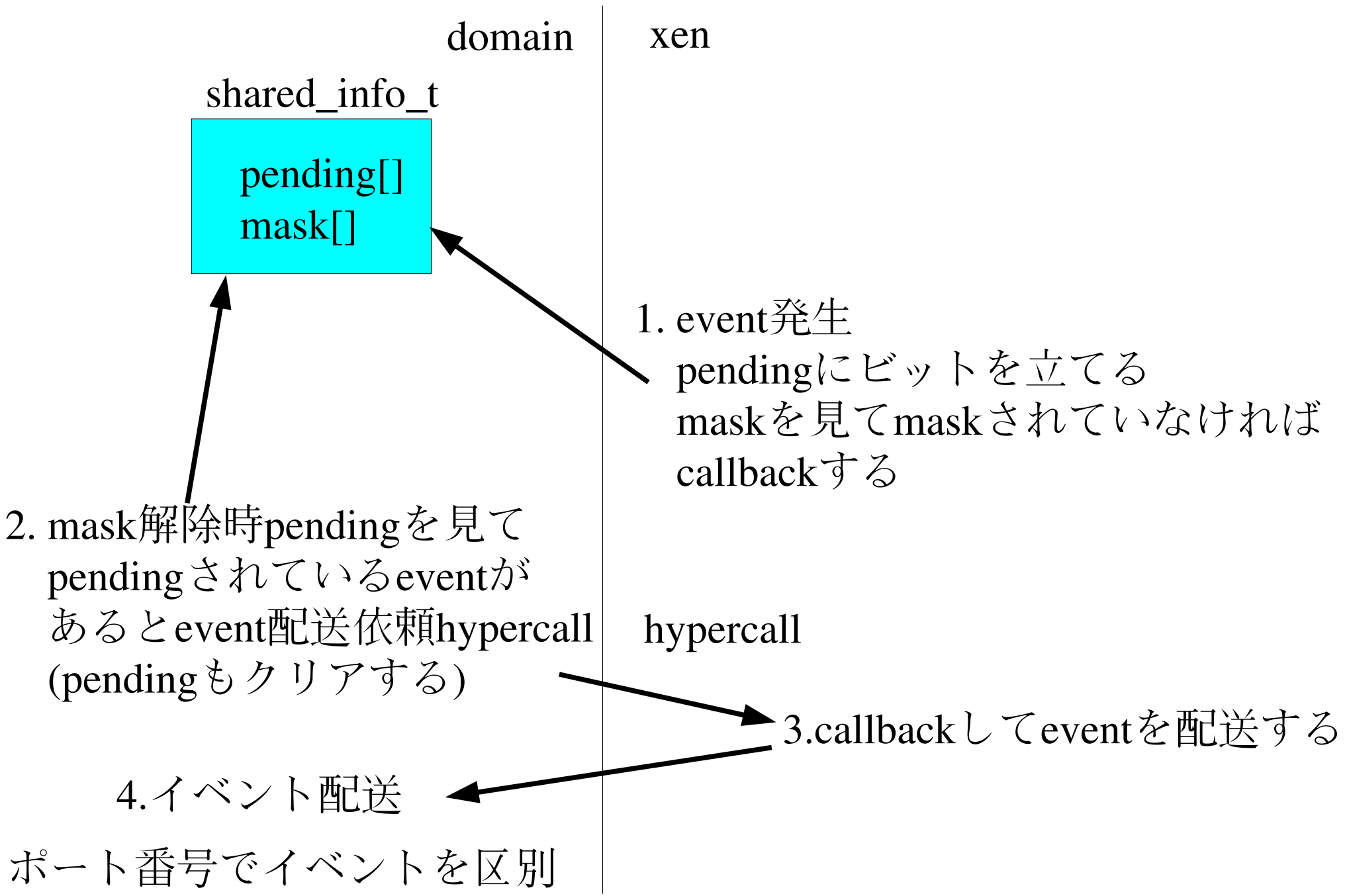


I/O デバイス仮想化

event channel

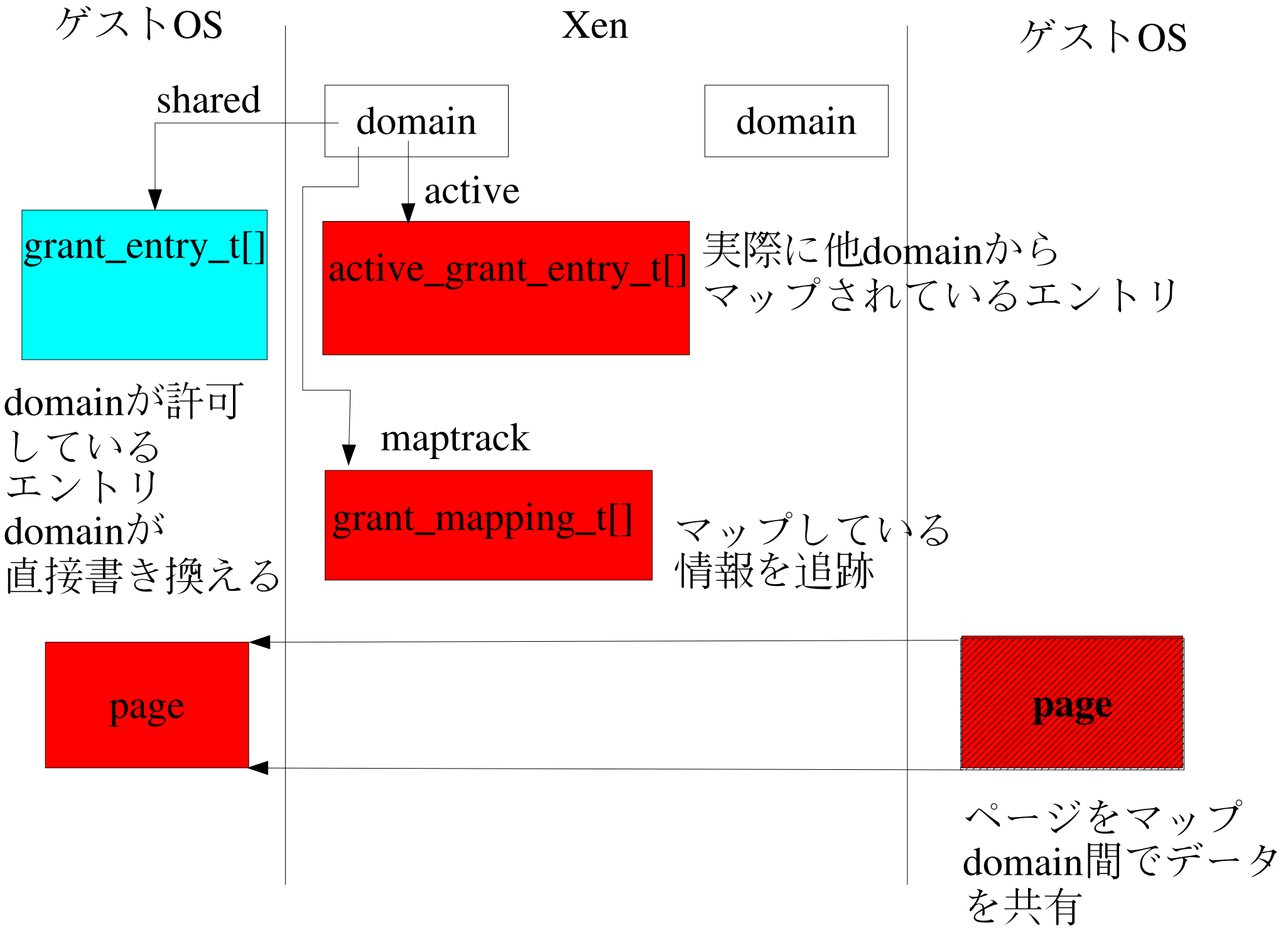
- Xen-domain/domain-domain間のevent通知機構
- ゲストOSはポート番号でeventを区別
- event発生源からポート番号の対応付けにより状態が分かれる
- shared_info_t構造体pending/maskメンバ
  - イベントのpending/mask状況
  - pending: xenからゲストOSへeventがあることを知らせる
  - mask: ゲストOSがイベントの配送をmaskすることをxenに知らせる

event channel



grant table

- 非特権ドメインが別のドメインのページにアクセスすることを許す機構
  - 特権domainはどのドメインのページにもアクセスできる
- アクセスさせるだけでなく、ページを渡すこともできる(ページ所有権の移動)
- 主にデバイスドライバ間でデータをやり取りする為に使用


ゲストOS
Xen
ゲストOS
shared
domain
domain
active
grant_entry_t[]
active_grant_entry_t[]
実際に他domainから
マップされているエントリ
domainが許可
している
エントリ
domainが
直接書き換える
maptrack
grant_mapping_t[]
マップしている
情報を追跡
page
page
ページをマップ
domain間でデータ
を共有

Device Channel (I/O ring)

- Device Channel(I/O ring)
  - ゲストOSのフロントエンドドライバ/バックエンドドライバでのデータをやり取りする機構
  - domain間でページを共有
    - フロントエンド/バックエンド ドライバ間のインターフェース
    - xenからは共有したページの使用方法には関知しない
  - ドライバはリング上で情報をやり取りする
- memory reservation
  - ドメインがxenにページを返したり、ページを確保したりする
  - balloon driverが使用
  - ドメインの使用メモリ量を調整

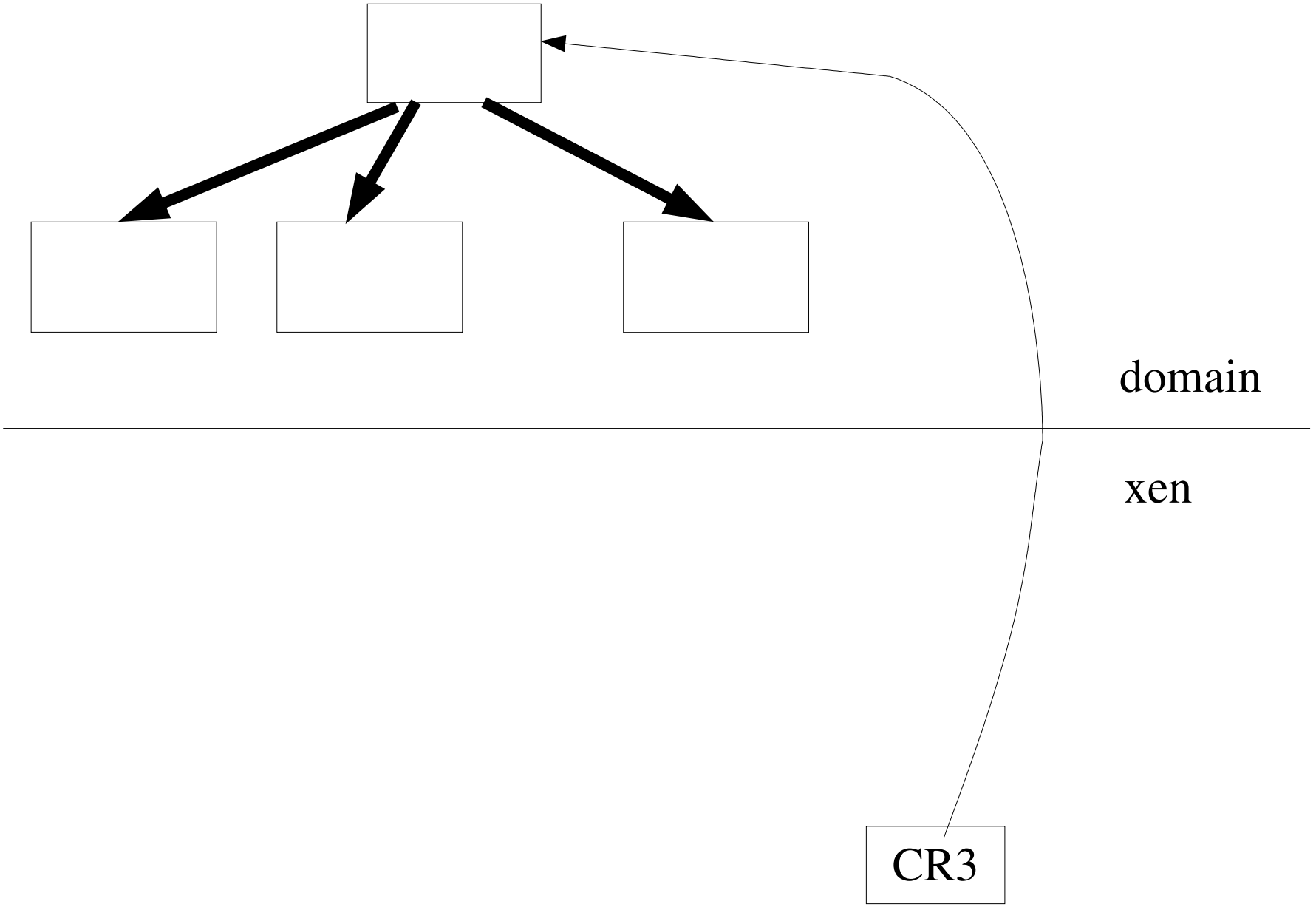
MMU仮想化

MMU仮想化

- 保護の為にゲストOSのMMUの設定が妥当なものか調べる必要がある
- Xenは多様なMMU仮想化をサポート
  - direct mode, writable page table mode, shadow mode
- Xenはページをゲスト OSの使用目的別にタイプに分けて追跡する
  - page table(PML4, PDP, PD, PT), gdt, ldt, writable(通常のページ), その他
  - タイプは排他的
- writable以外はドメインは原則としてread only でしかアクセスできない

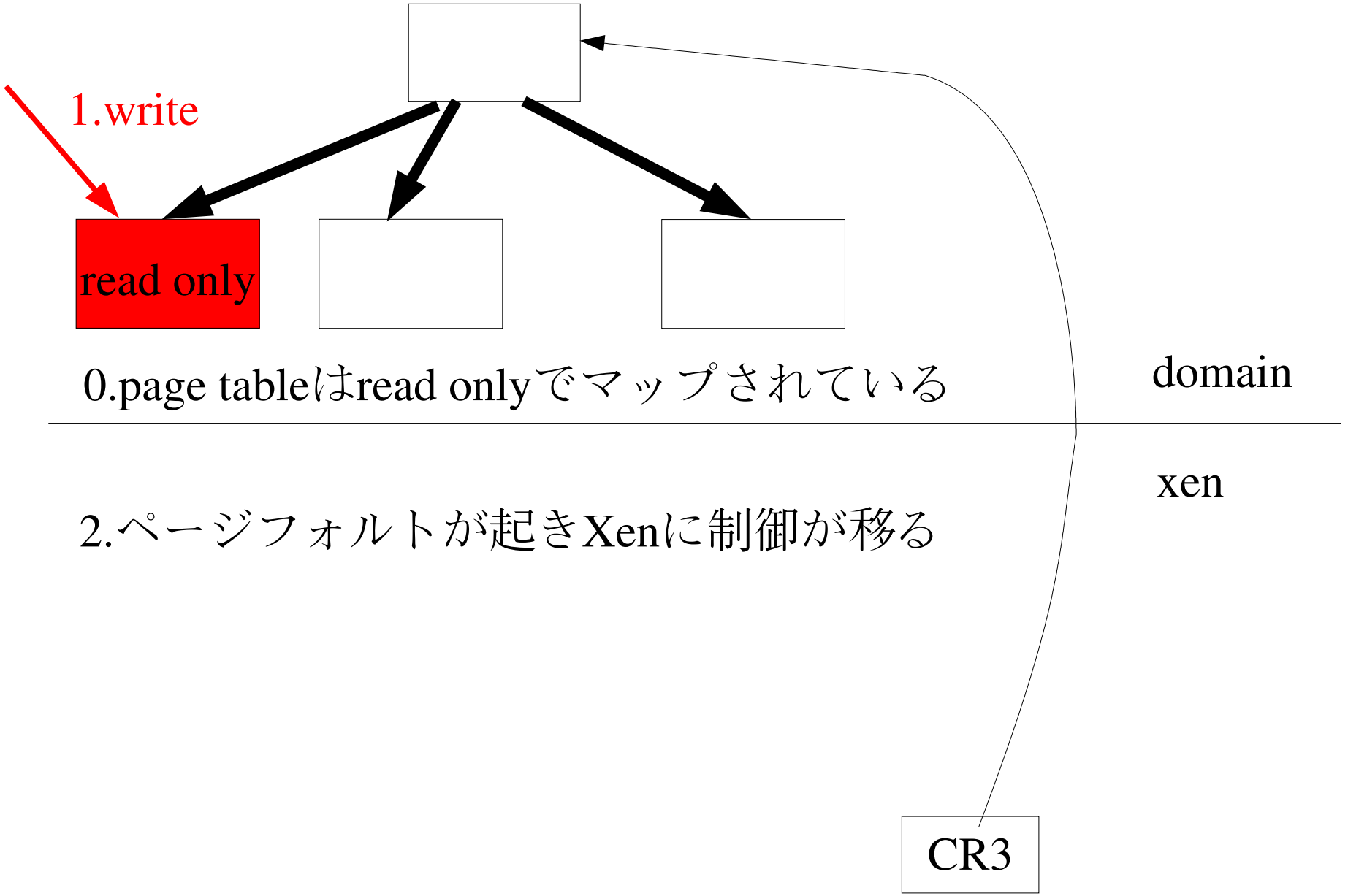
direct mode

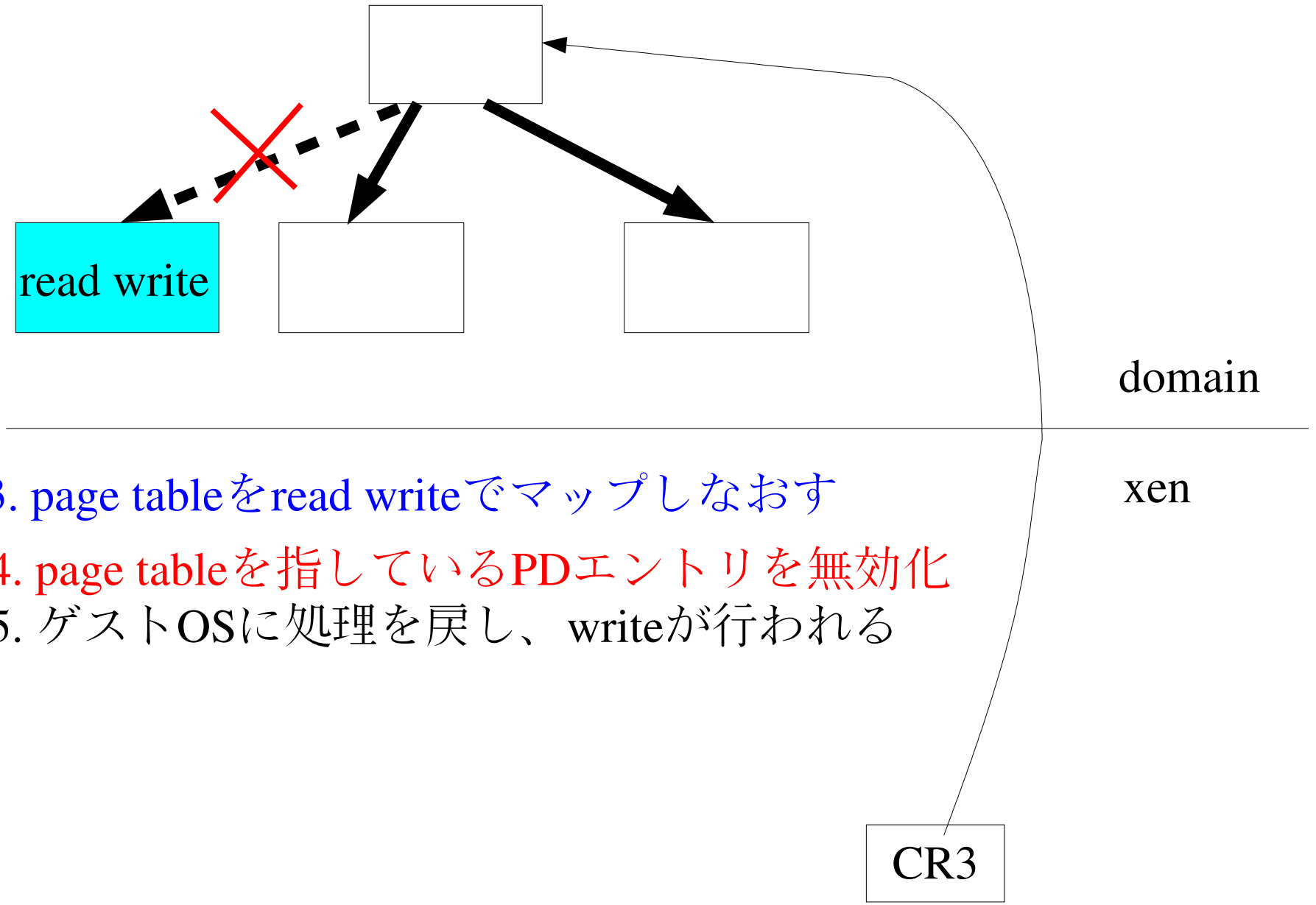
- ドメイン内のページテーブルを直接使用
- ページテーブルはread onlyでマップしゲストOSは直接書き換えられない
  - hypercallを利用して変更する
- ゲストOSは(pseudo)物理アドレスとマシンアドレスの変換をする
  - ページテーブルにはマシンアドレスが入る


domain
xen
CR3

writable page table mode

- direct modeを拡張したもの
- ゲストOSは(pseudo)物理アドレスとマシンアドレスを意識する
- ゲストOSがページテーブル(PTのみ)へ書き込むことを許可する


1.write
read only
0.page tableはread onlyでマップされている
domain
xen
2.ページフォルトが起きXenに制御が移る
CR3

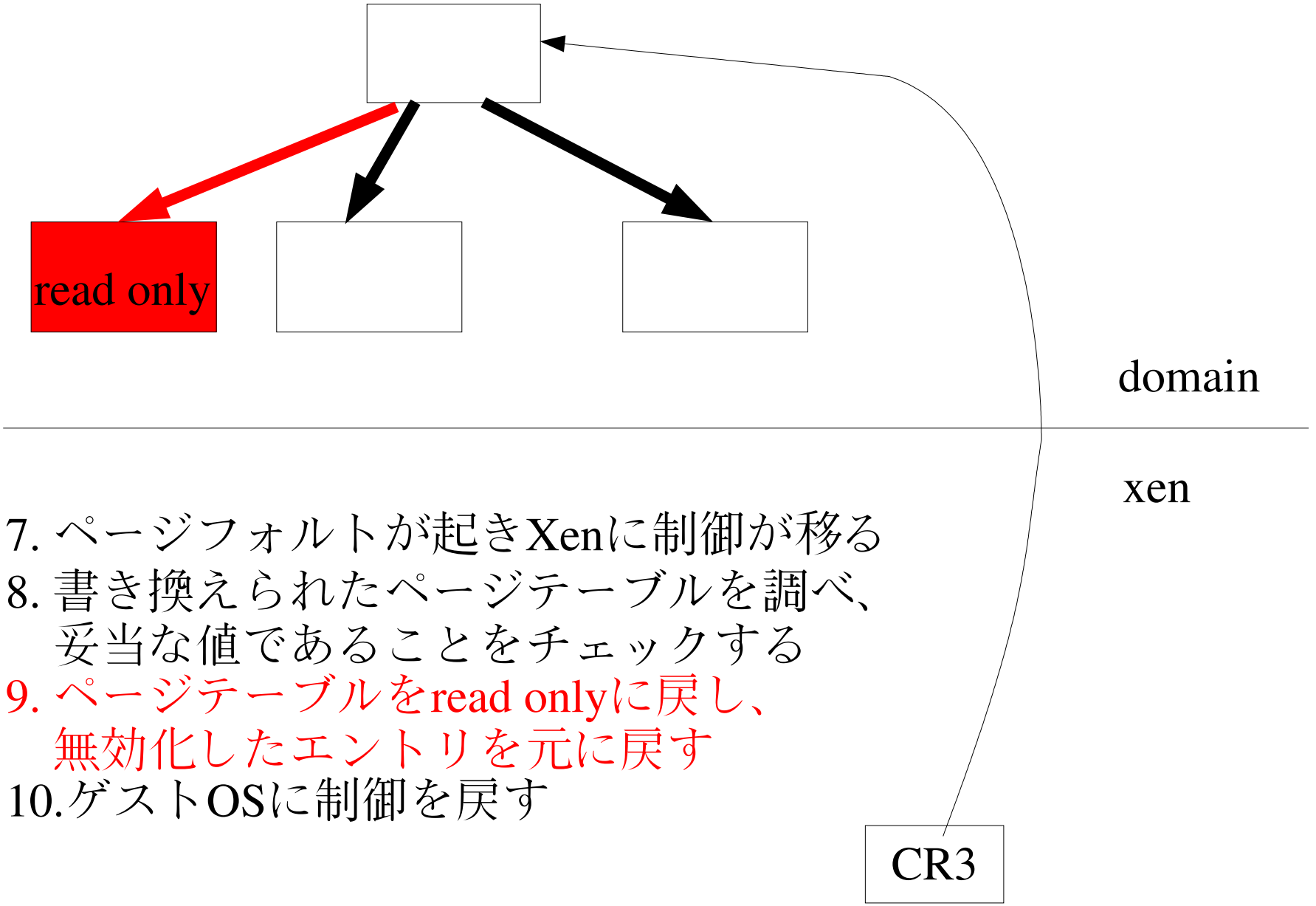


3. page tableをread writeでマップしなおす

4. page tableを指しているPDエントリを無効化

5. ゲストOSに処理を戻し、writeが行われる

6.page tableがマップしている(とゲストOSが思っている)
アドレスにアクセス

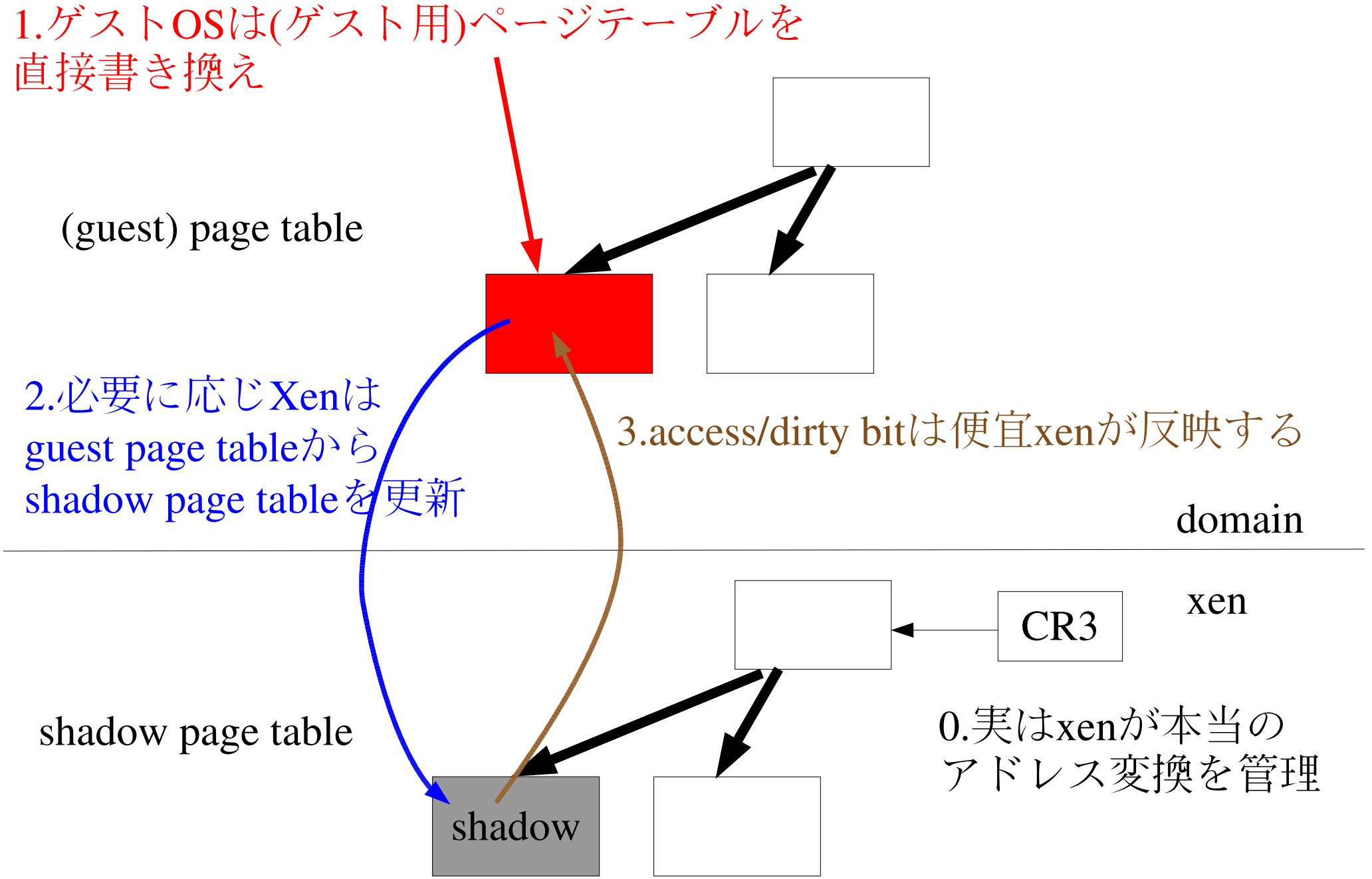


7. ページフォルトが起きXenに制御が移る
8. 書き換えられたページテーブルを調べ、
   妥当な値であることをチェックする
9. ページテーブルをread onlyに戻し、
   無効化したエントリを元に戻す
10.ゲストOSに制御を戻す

shadow mode

- ページテーブルはXenのもの(shadow page table)を使用する
  - ゲストOSは(virtual) page tableを直接書き変える
- XenはゲストOSが設定したエントリを必要に応じshadow page tableを設定する
- マップしているpageをnon-present or read-only にマップすることによりaccess bit, dirty bitを立てる

shadow mode (cont.)

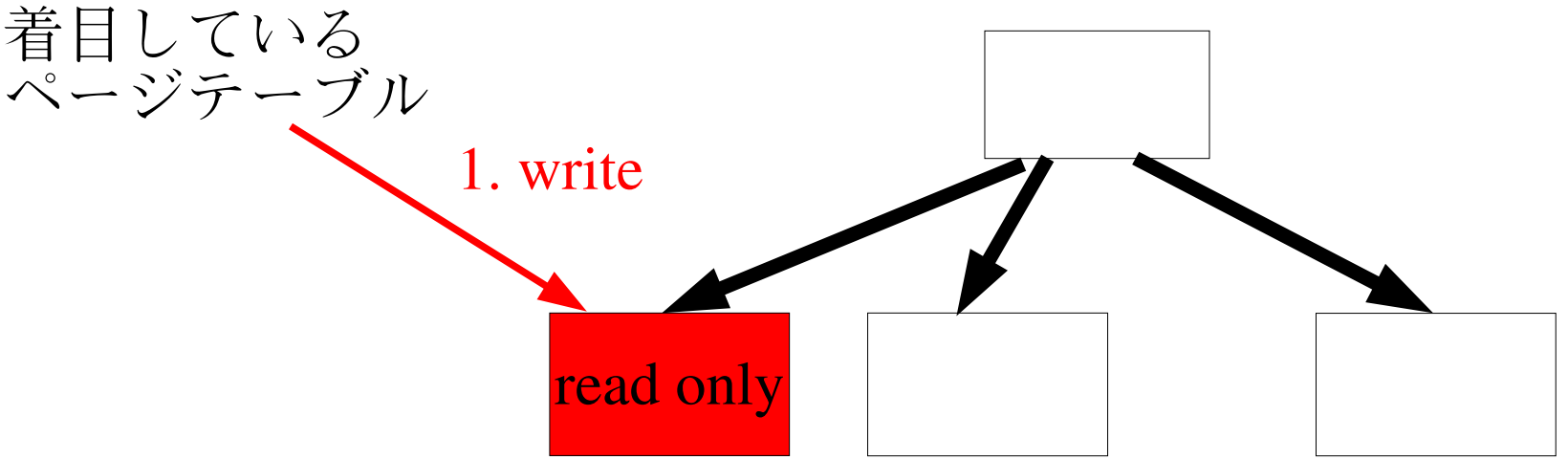


shadow mode(cont.)

- shadow modeにもいろいろモードがある
- refcounts: ページを参照カウンタで追跡
- write all: 全てのゲストページテーブルへのwriteを許す(pteの許可属性に関わらず)
- log dirty: ダーティーページを記録
  - live migrationに使用
- translate: xenが(pseudo)物理アドレス->マシンアドレスの変換を行う
- external: 完全仮想化で使用

shadow mode(cont.)

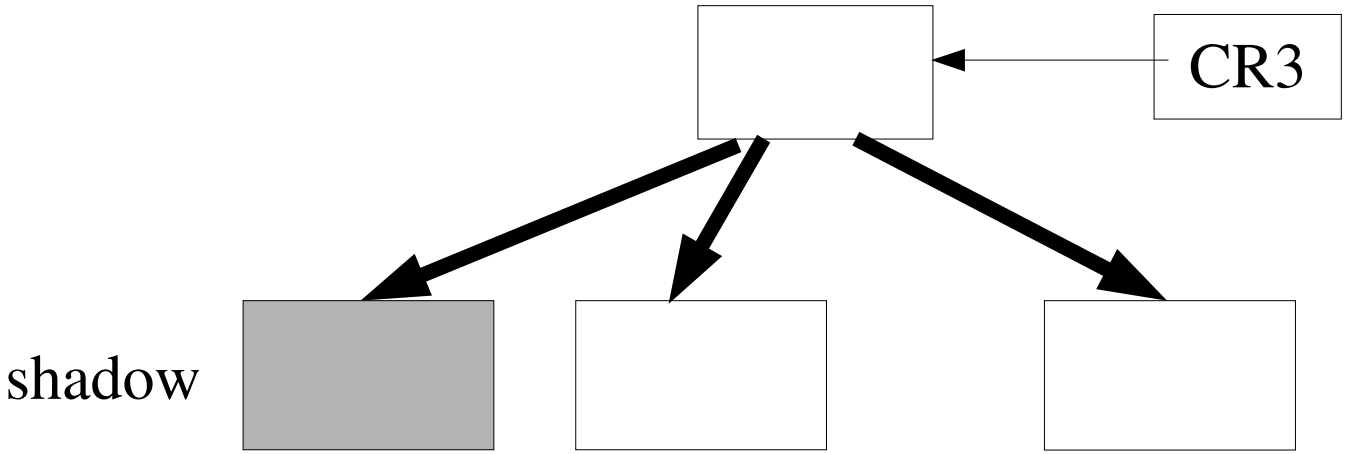
- promote
  - writable(通常ページ) -> page tableへの変換
  - ゲストOSがあるページを新たに(ゲスト)ページテーブルにする時に全てのシャドウページテーブルを調べてread-writeでマップしているものがあればread-onlyにマップしなおす
- demote
  - promoteの逆
  - ゲストページテーブルの追跡に使用した資源を解放する



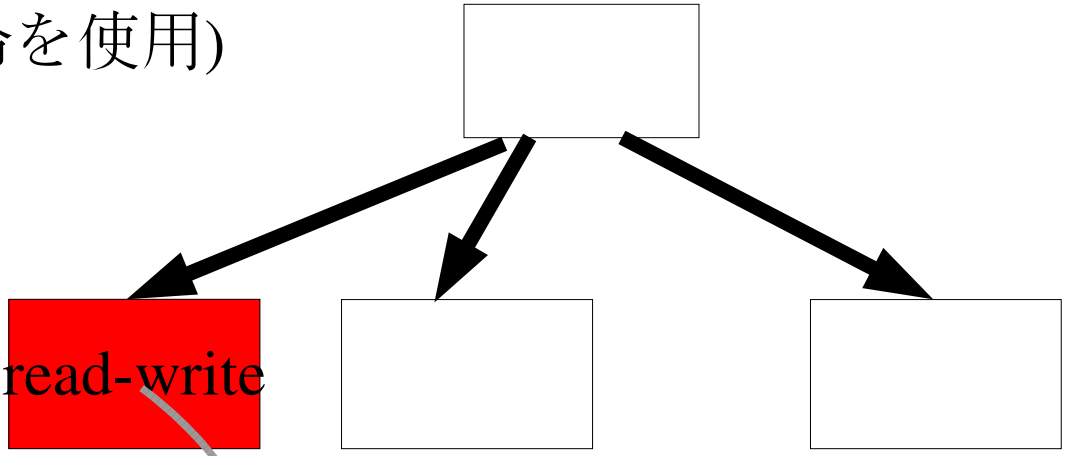
0. (ゲスト)ページテーブルはread-onlyでマップされる　domain

---

2. ページフォルトし制御がXenに移る　xen
3. (ゲスト)ページテーブルをread/writeでマップしなおす
4. 制御をゲストOSへ戻し、
   writeされる



5.ゲストOSはTLBをフラッシュする
(hypercall or 特権命令を使用)

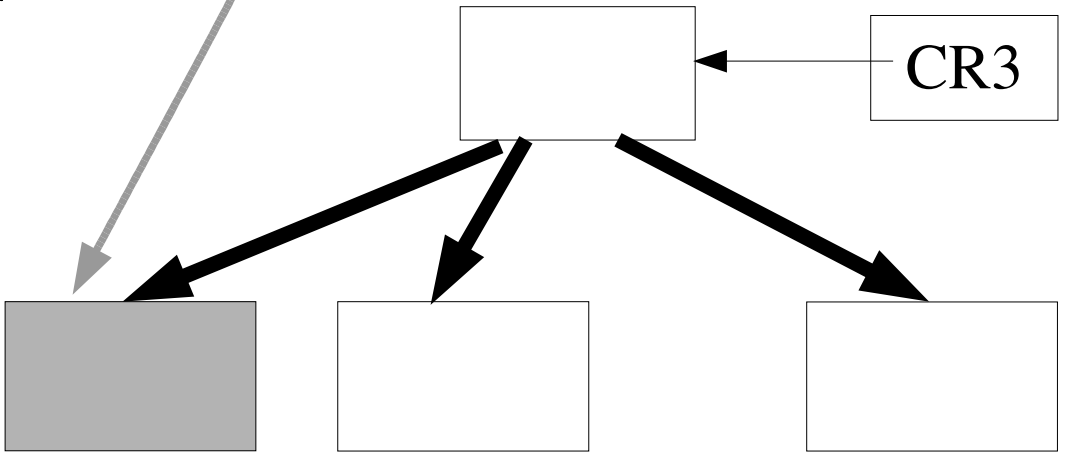


domain

6. hypercallあるいは特権違反で
制御がXenに移る

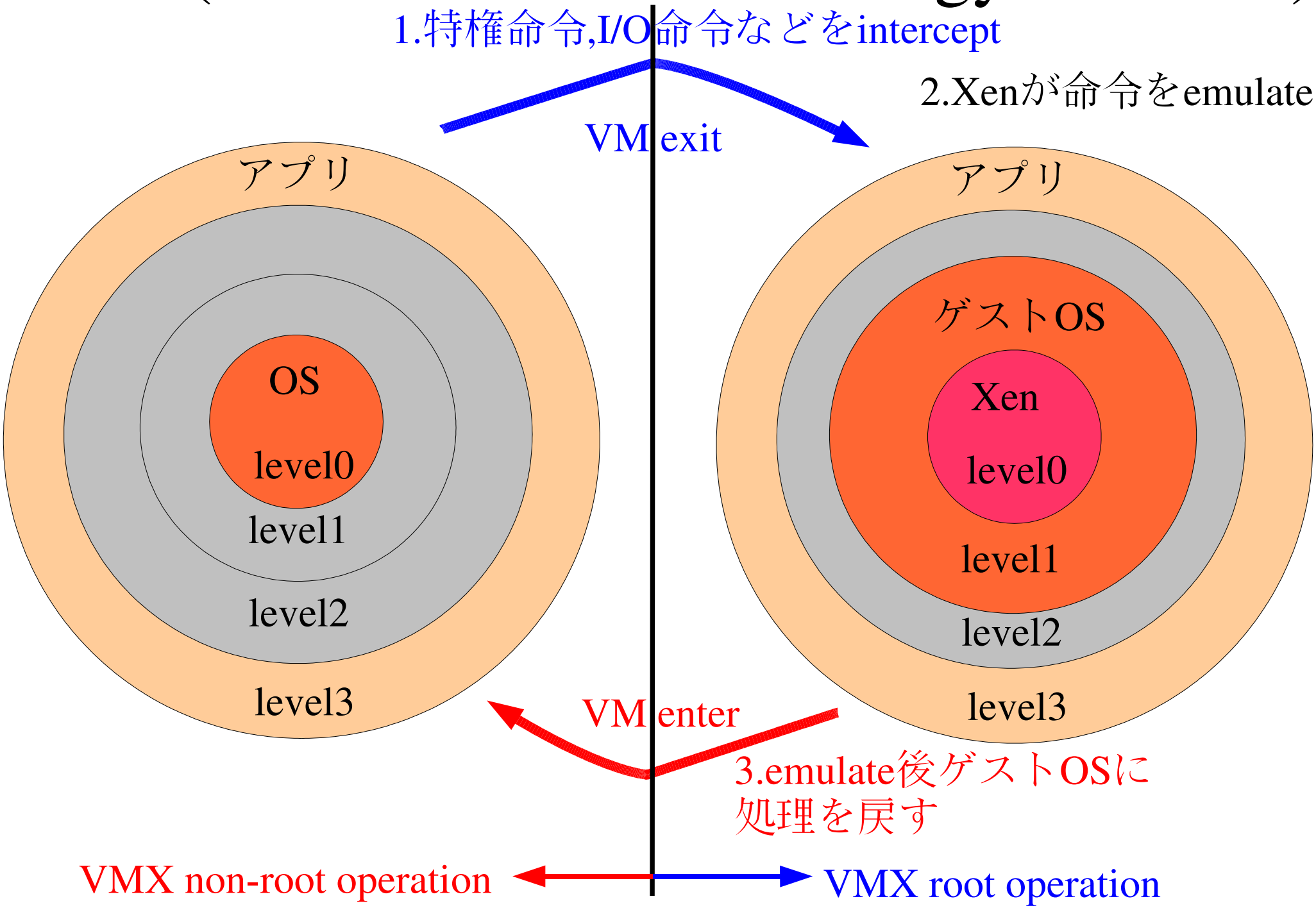
7. 書き込み可能にしていた　　xen
(ゲスト)ページテーブルを調べ
shadowページテーブルを更新する

8. 制御をゲストOSへ戻す

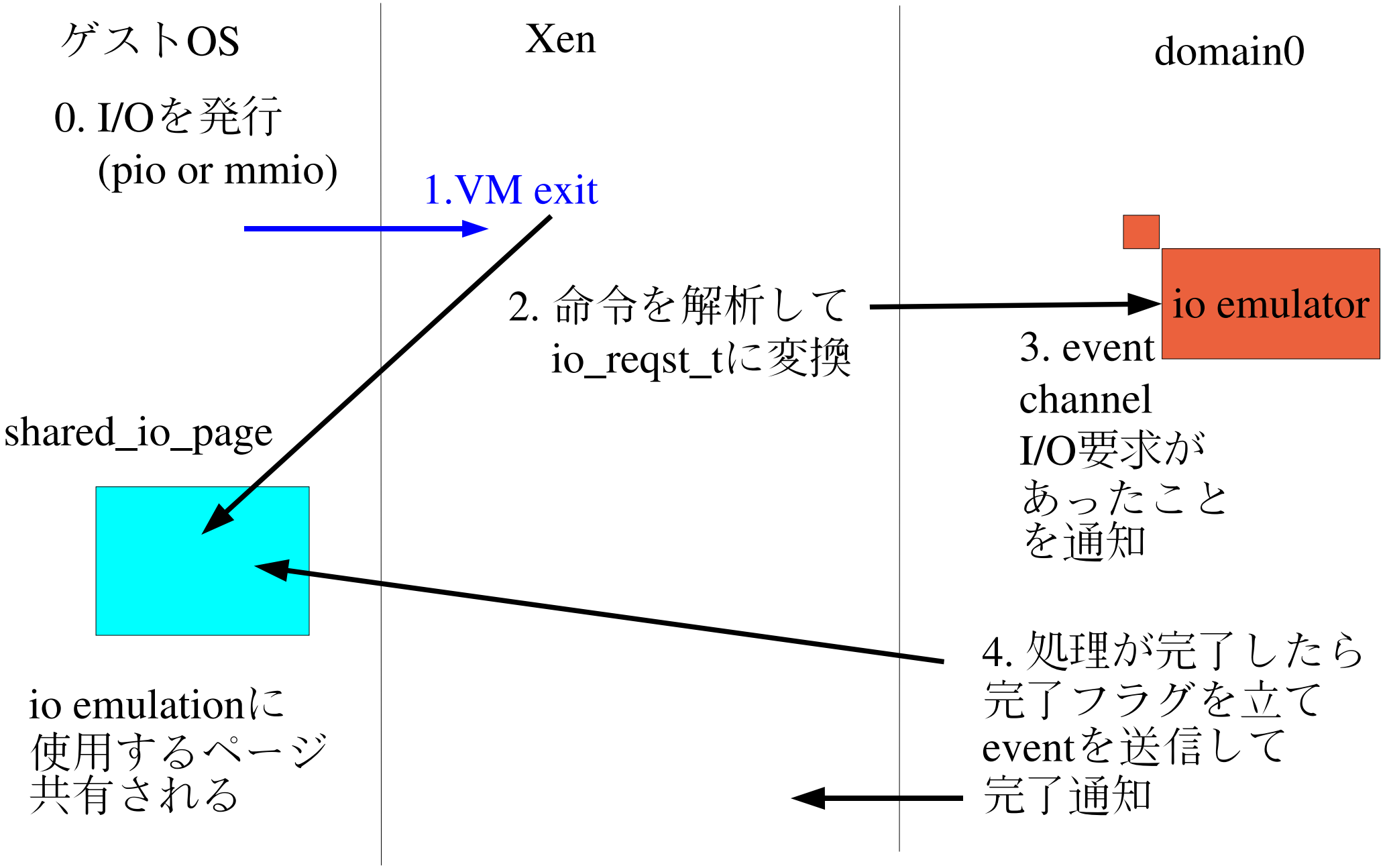


完全仮想化

VT-x(Virtualization Technology for IA32)



I/O emulation



今後の展望

- 一層の安定化
  - デバッグ支援ツール
  - 性能測定ツール
  - テストツール
- 他アーキテクチャへの対応
  - IA64
  - PPC
- 完全仮想化
  - Pacifica対応
  - デバイスエミュレーション
- xenfs: ファイルシステムの仮想化

今後の展望(cont.)

- リソースコントロール
- NUMA
- ホットプラグ
- 仮想デバイス
  - 使いやすい仮想デバイス

ご静聴ありがとうございました

オープンソースマガジン
Xen実装解説連載予定
乞うご期待