Technology Consulting Company
Research, Development &
Global Standard

BitVisor Summit 2012

Customizing BitVisor 1.3

松原　克弥／株式会社イーゲル

2012.12.4

はじめに：BitVisorとのかかわり

igel

2007
USB対応
(USBメモリ暗号化、libusb互換API for ICカード)

2008
ATAPI対応
(光学メディア暗号化)

2009
保護ドメイン
(VMM内Ring-3で動作)

2010
ATA piggyback
(VMMからディスクI/O)
サスペンド対応

2011
TPM対応
(Trusted Boot、乱数生成器)
バックグラウンドFDE
(OS動作中のFull Disk Encryption)

2012
USBドングル利用
(バイナリライブラリのリンク)
最新VT機能の対応
(性能向上)

BitVisor powered by IGEL

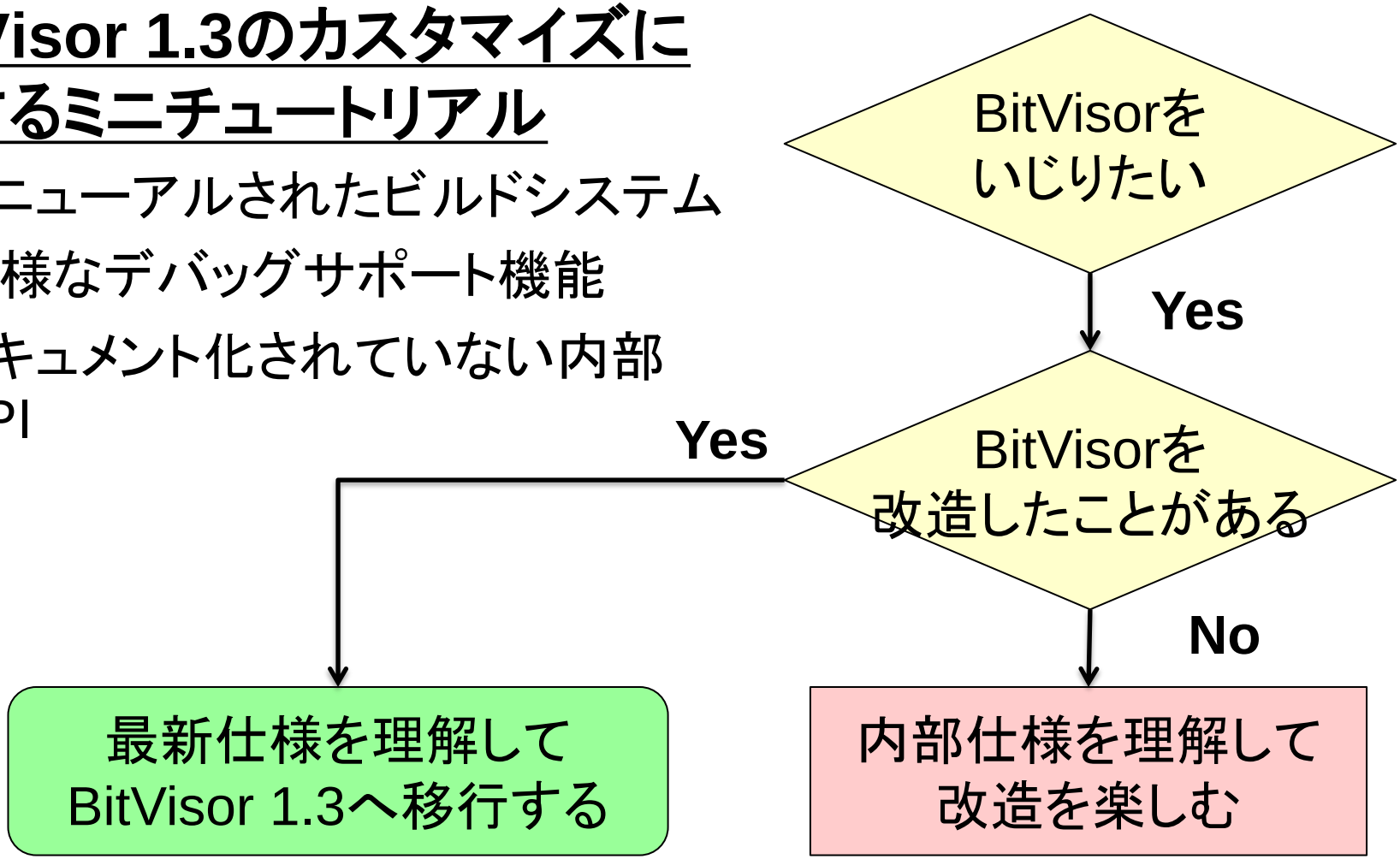
- 導入サポート
  - プラットフォーム選定、動作検証
  - インストール、セキュリティポリシー設定、リカバリメディア作成
- 受託カスタマイズ
  - 新たなデバイス対応、新機能実装
- ソリューション開拓
  - USBデバイス隠ぺい（偽装）
  - ICカード連携画面ロック
  - UEFI対応
  - “BitVisor for Mac”
- オープンソース・コミュニティ活動
  - インストールマニュアル作成
  - Realtek RTL8169ドライバ, ADvisorのマージ

本講演の内容

igel

BitVisor 1.3のカスタマイズに関するミニチュートリアル

- リニューアルされたビルドシステム
- 多様なデバッグサポート機能
- ドキュメント化されていない内部API



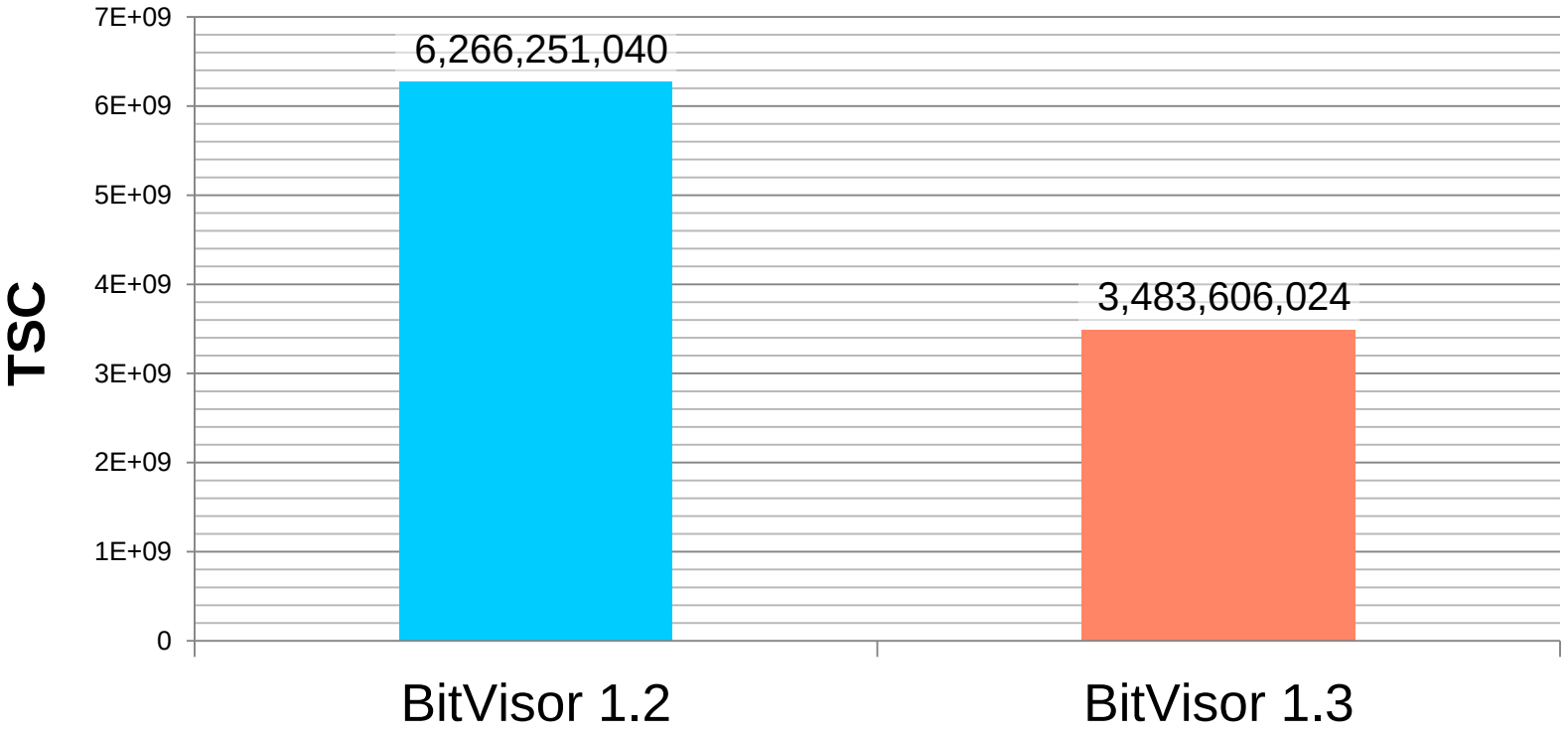
BitVisor 1.3 Changes

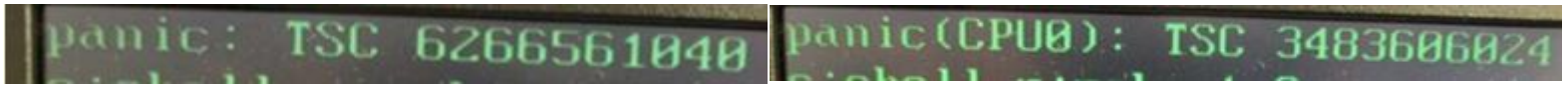
igel

- AMD-Vマルチコア対応
- 64ビットゲストOS対応
- バックグラウンド暗号化
- 起動時間短縮
- 高速化
- 画面への描画機能（ADVisor）
- ビルドシステム改善
- OpenSSL更新
- バグfix、その他機能改善

BitVisor 1.2/1.3の起動時間

BitVisor起動に要する時間をTSCでカウントした。BitVisor 1.3は、1.2と比較して44.4%起動時間が減少している。





測定環境: ThinkPad X200 (Core 2 Duo P8600 2.4GHz/メモリ2GB)

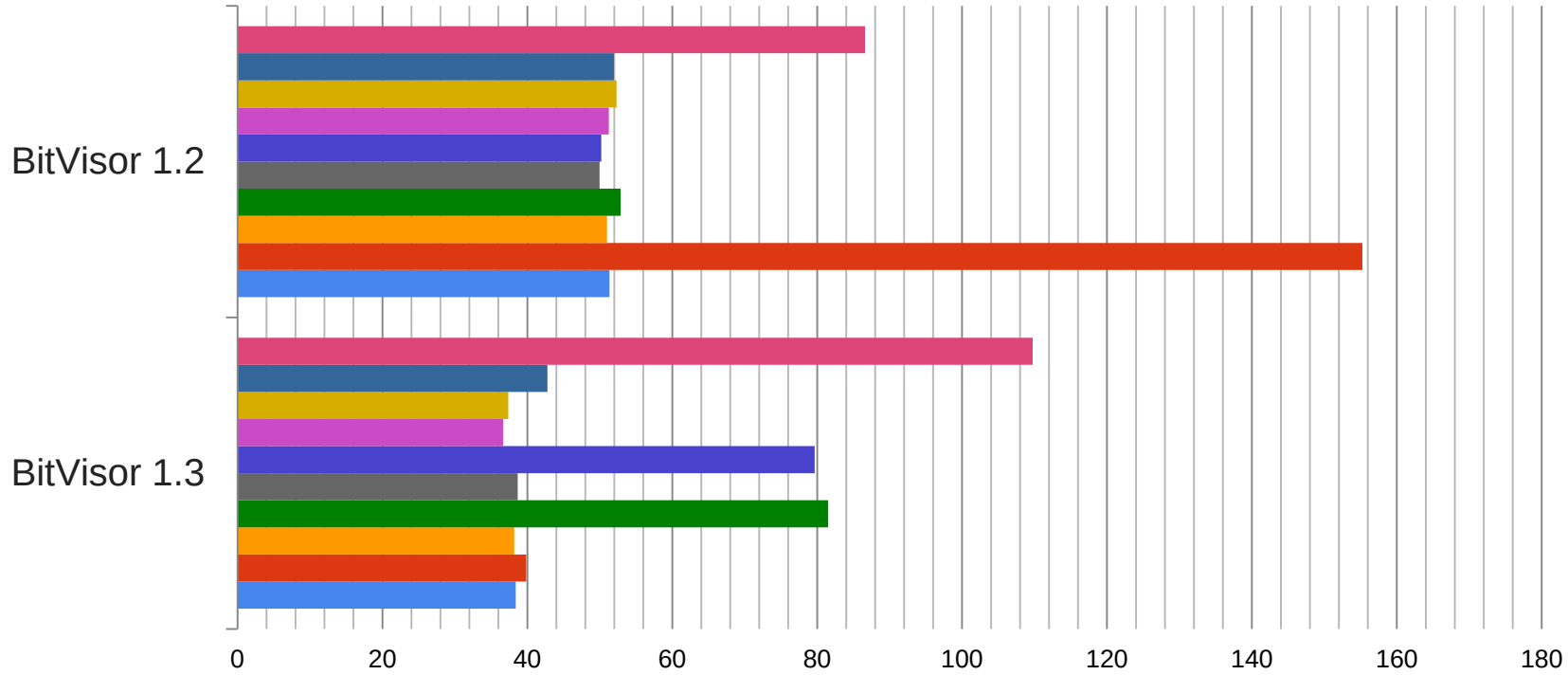
BitVisor 1.3 Changes

igel

- AMD-Vマルチコア対応
- 64ビットゲストOS対応
- バックグラウンド暗号化
- 起動時間短縮
- 高速化
- 画面への描画機能（ADVisor）
- ビルドシステム改善
- OpenSSL更新
- バグfix、その他機能改善

暗号化ディスクからのOS起動時間

GRUBメニューよりWindowsを選択した時間から、スタートアップ設定したアプリケーションが起動してWindowsデスクトップに表示されるまでの時間をストップウォッチで10回測定した。（Windowsログインパスワードは削除状態）

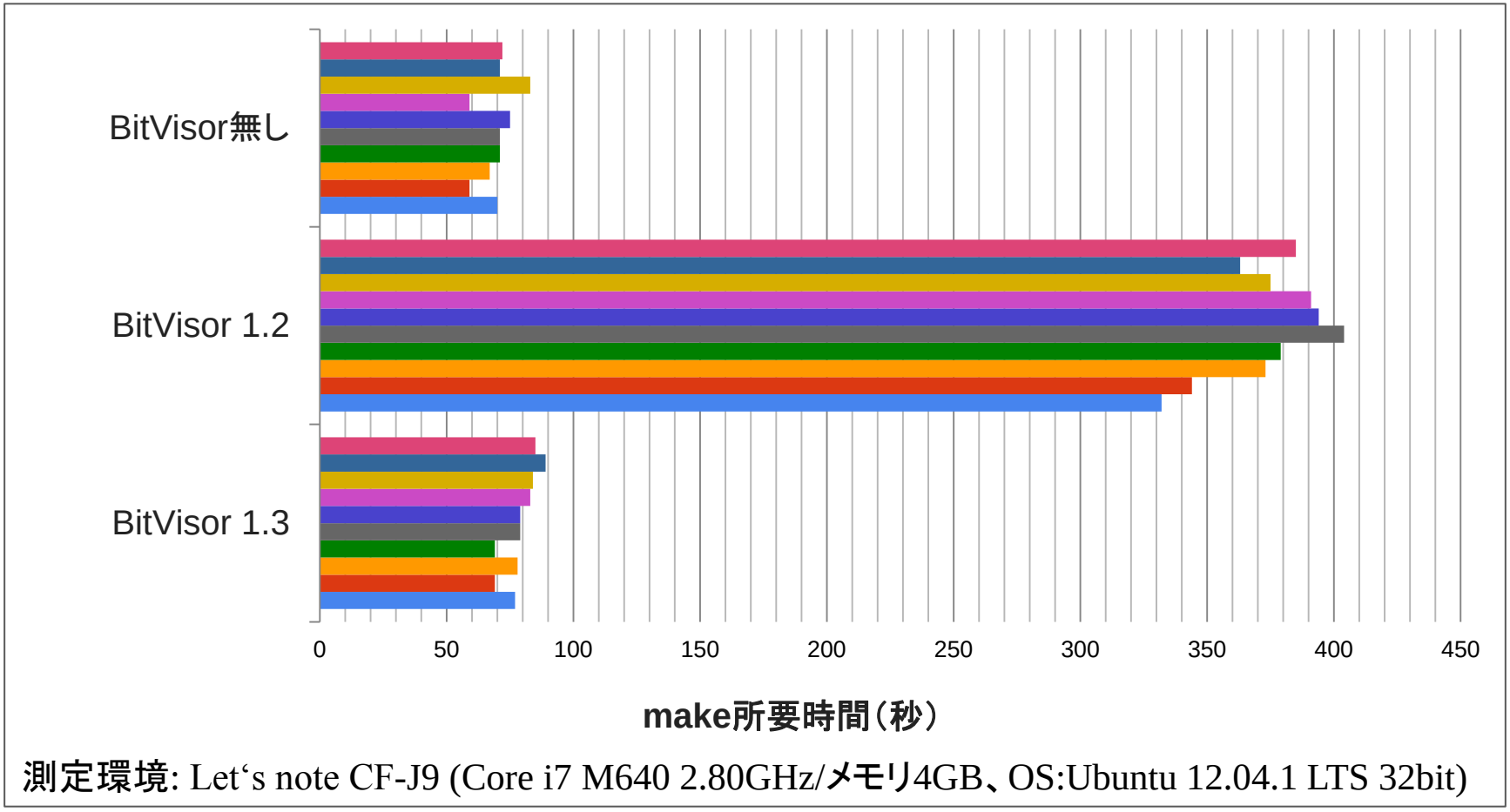


測定環境: Let‘s note CF-J9 (Core i7 M640 2.80GHz/メモリ4GB、OS:Windows7 32bit)

OS上でのビルド・ベンチマーク

igel

BitVisor無し、BitVisor 1.2、BitVisor 1.3の各条件にて、BitVisor1.3をmakeし、終了するまでの 時間をtimeコマンドで10回測定した。（BitVisorによるディスク暗号化無 ）



測定環境: Let‘s note CF-J9 (Core i7 M640 2.80GHz/メモリ4GB、OS:Ubuntu 12.04.1 LTS 32bit)

BitVisor 1.3 Changes

igel

- AMD-Vマルチコア対応
- 64ビットゲストOS対応
- バックグラウンド暗号化
- 起動時間短縮
- 高速化
- 画面への描画機能（ADVisor）
- ビルドシステム改善
- OpenSSL更新
- バグfix、その他機能改善

BitVisor 1.3をカスタマイズする

igel

題材

1. VMM内で動作するタスクを追加
2. VMMから画像を表示する
3. 保護ドメインを使ってタスクを隔離する
4. ディスクからデータを読み込む

以降、このマークの場所では、
実例を動作させて説明します。

ディレクトリ構成 in 1.3

igel

- boot – ブートローダ、mini OS
- core – VMMコアコード
- crypto - OpenSSL
- drivers – PCI, ATA, USB, ネットワーク等ドライバ
- idman – ICカード管理
- include – ヘッダファイル
- process – 保護ドメイン動作のタスク・ライブラリ
- storage – ストレージ暗号化
- tools – 各種ツール
- vpn - VPN

Makefileの変更（1）

igel

コンパイル対象を最小限に

1.3でビルドシステム（Makefile）が大幅に改善

並列コンパイル可

- core/にファイルを追加
  - 自動的にビルド対象（Makefileの変更は必要なし）
- 新たなディレクトリを追加
  - 親ディレクトリのMakefileに以下の行を追加

```
subdirs-1 += newdir
```

  - ディレクトリ内Makefileに以下を記述

```
objs-1 += newfile.o
```

- process/にファイルを追加 ⇒ 後述

1. VMM内動作するタスクを追加

igel

一定周期(5秒毎)にログ出力を行うheartbeatタスク

```
static void heartbeat_thread (void *arg)
{
    ...
    for (;;) {
       schedule();
       cur = get_time();
       ...
    }
    thread_exit();
}

static void heartbeat_kernel_init (void)
{
    thread_new (heartbeat_thread, NULL, VMM_STACKSIZE);
}

INITFUNC ("config1", heartbeat_kernel_init);
```

INITFUNCマクロ

igel

VMMから呼び出される初期化関数を登録

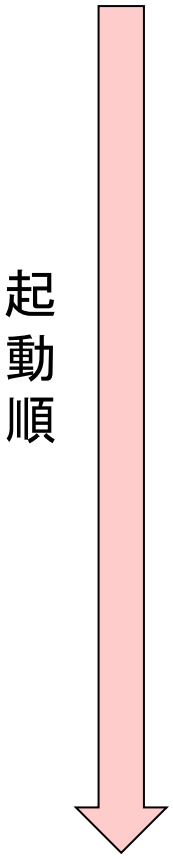
INITFUNC(呼び出しタイミング・ラベル, 関数)

呼び出しタイミング・ラベル

＝「プレフィックス＋数字」の文字列

- VMMコアからcall_initfunc(プレフィックス)で呼び出し
- 同じプレフィックス間はascii順にソートされて呼び出し
- ラベルは8文字以内

INITFUNC: ラベル・プレフィックス一例



プレフィックス	呼び出しタイミング
global	VMM起動直後
pcpu	（各）プロセッサコア開始直後
msg	メッセージI/F初期化後
vmmcall	ゲストOS I/F利用可能後
config0	mini OS 起動後
driver	ドライバの初期化時
config1	mini OS 起動＋ドライバ初期化済
suspend	サスペンド直前
wakeup	（各）プロセッサ再開直後
resume	レジューム直後

デバッグ:ログ出力

igel

- logコマンド in dbgsh
  - CONFIG_DBGSH=1 (vmm.dbgsh=1)
  - tools/dbgshにゲストOS上で動作するツール
- シリアル出力
  - CONFIG_TTY_SERIAL=1
  - 115200bps@ttyS0(COM1)で接続
  - core/serial.cを変更することで、他の設定も可
- ネットワーク出力
  - CONFIG_TTY_PRO1000=1 or CONFIG_TTY_RTL8169=1 (vmm.tty_pro1000=1, vmm.tty_pro1000_macに受信側MACアドレス)
  - ログ受け取り側は、core/tty.c:L126-129にperlスクリプトあり

デバッグ:ログ出力 (contd.)

igel

- 画面出力
  - CONFIG_VGA_INTEL_DRIVER=1, CONFIG_TTY_VGA=1
  - 画面がスクロールするので、動画で撮影するのがおススメ
- linux syslog出力
  - CONFIG_LOG_TO_GUEST
  - tools/logにあるカーネルモジュールをロード
  - OS上のログとVMMのログが時系列に並ぶので、OS挙動とのVMM処理のマッピング把握に便利

Tips:Panicログの読み方

SEGV等のエラーが発生したら‥

```
Error: An exception in VMM!
Interrupt number: 0x0E Page-Fault Exception (#PF)
CR0: 0x8000003B  CR2: 0x00F20119  CR3: 0x71772000
...
CPU0  RSP on interrupt: 0x42887F90  Stack information:
+00 Errcode 0000000000000000  +20 RSP 0000000042887FC8
+40 00000000
+08 RIP     00000000401566D9 +28 ...
```

addr2lineコマンドでソースコード該当行を見つける

```
~/bitvisor$ addr2line -e bitvisor.elf 0x401566D9
/home/matsu/bitvisor/heartbeat/kernel.c:16
```

2. 画面に画像を描画する

igel

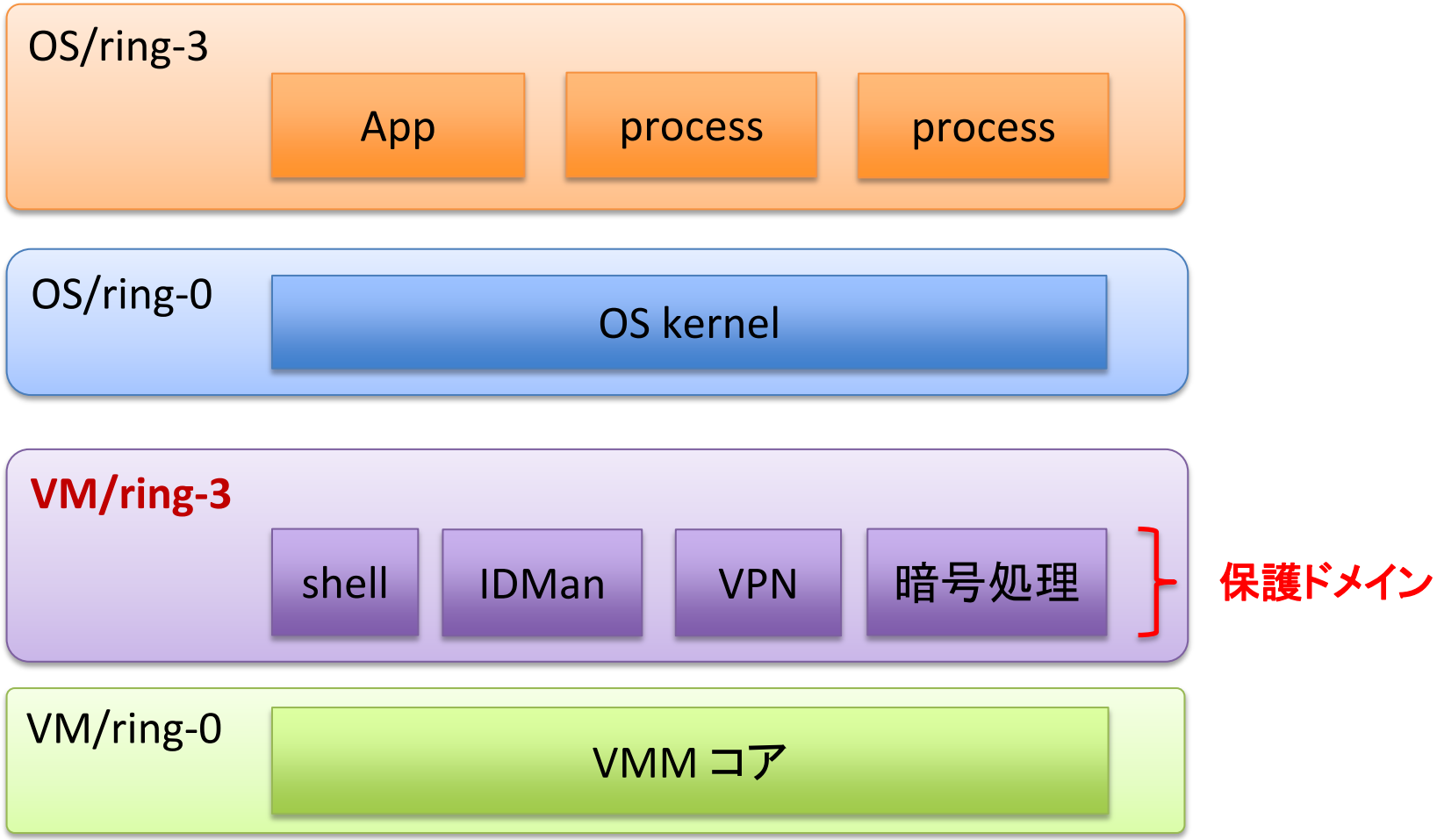
画面描画API in include/core/vga.h

- vga_is_ready ()
  画面描画が可能な状態かを調べる。グラフィックH/Wが初期化されるまでに描画してしまうことを防ぐ。
- vga_transfer_image ()
  画像データをフレームバッファへ転送する。
- vga_fill_rect ()
  画面に矩形を描画する。

3. 保護ドメインでタスクを隔離

igel

Ring-3特権レベルで実行されるVMM内プロセス



保護ドメインの特徴

igel

- メモリ空間が分離独立していて、他の保護ドメインやVMMコアのメモリ領域へ直接アクセス不可
  - **VMM各機能のコンパートメント化が可能**
- 保護ドメインのタスクが異常終了してもVMMは継続動作
  - **ロバストなVMMを実現**

保護ドメイン化

igel

1. processの下にコードを移動
   - Makefileには、以下の行を追加

```
bins-1 += myprocess
myprocess-objs = myprocess.c
myprocess-libs = ...
```

2. _start () 関数を作成
   - 終了時はexitprocess ()を実行すること
3. ヘッダファイル、ライブラリは process/libのものを使用
4. VMMコアとのやりとりはメッセージI/Fで

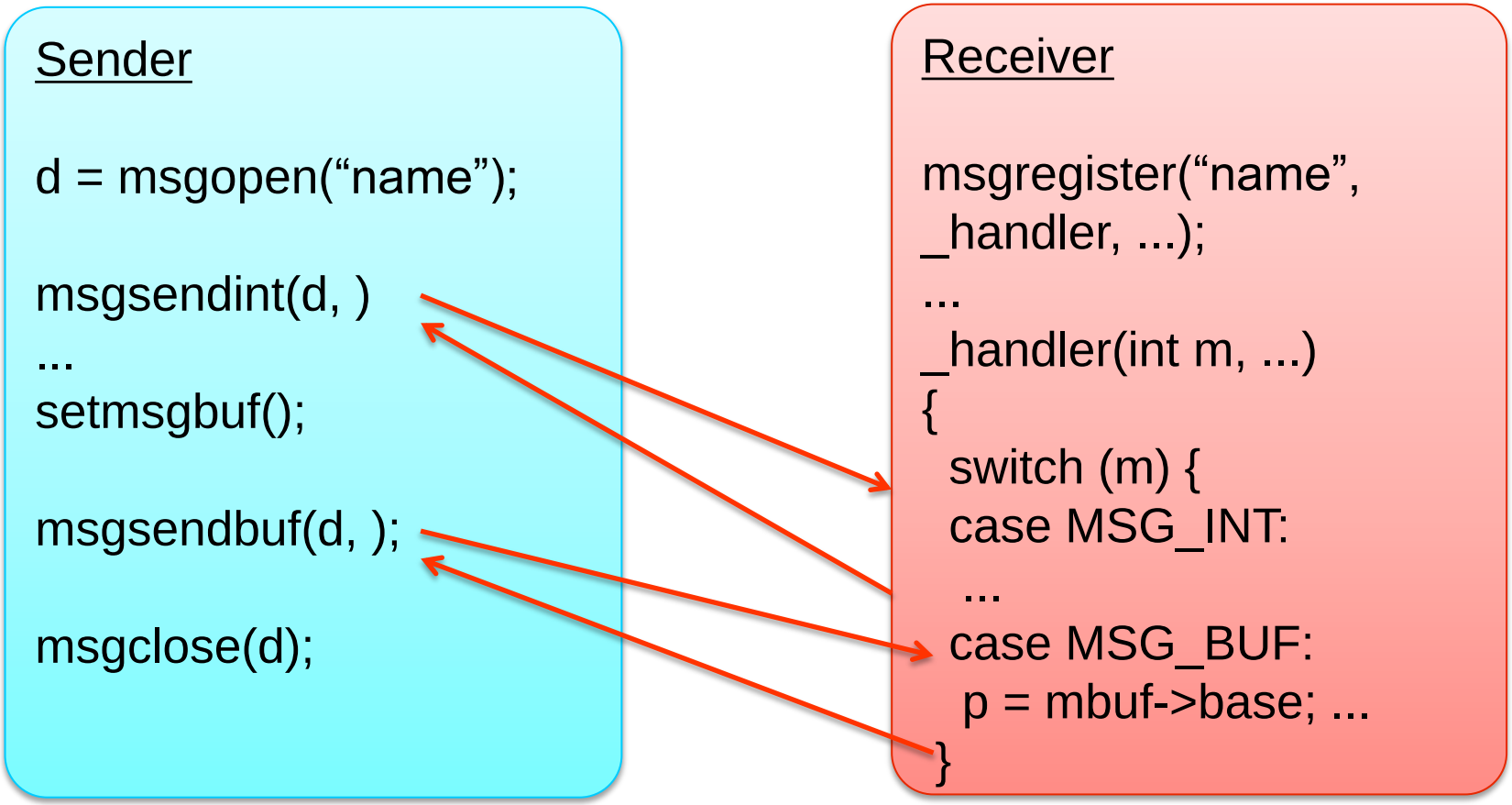
メッセージI/F

igel

- msgregister, msgunregister：メッセージハンドラの登録
- msgopen, msgclose：メッセージ経路の確率
- msgsendint：メッセージ（integer）の送信
- msgsenddesc：メッセージ（ディスクリプタ）の送信
- msgsendbuf：メッセージ（バッファ）の送信
- setmsgbuf：メッセージの作成

メッセージI/F (contd.)

igel



保護ドメインを適用したタスク起動

igel

- VMMからnewprocess()でインスタンス生成、msgsendint()で起動

```
p = newprocess(“process_name”);
d0 = msgopen (“ttyin”);
d1 = msgopen (“ttyout”);
msgsenddesc (p, d0);
msgsenddesc (p, d1);
msgsendint (p, 0);
...
```

- dbgshから起動

4. ディスクからデータを読み込む

igel

保護ドメイン・ディスクI/O in process/lib/lib_storage_io.h

- storage_io_init ()
  - 開始処理
- storage_io_get_num_devices ()
  - デバイス数をカウント、I/O前に必須な処理
- storage_io_deinit ()
  - 終了処理
- storage_io_aread ()
  - 非同期ファイルread
- storage_io_awrite ()
  - 非同期ファイルwrite

他の内部API

igel

- USB
  - libusb互換API
  - USB通信フック
- ネットワーク
  - UDPパケット送信
- PCI
  - PCI configuration spaceへのアクセスフック
- TPM
  - TCG BIOS API
- vmmcall

おわりに

igel

- 最新BitVisor 1.3のススメ
  - サポートプラットフォームの拡充
  - 性能向上
  - 画面描画やバックグラウンド暗号化等の新機能
- BitVisor改造テクニックを紹介
  - Makefileの仕組み
  - ログ出力
- “Undocumented” 内部APIの（一部を）解説
  - 画面描画機能
  - 保護ドメインとメッセージI/F
  - ディスクI/O

おわりに‥言いたいこと

igel

- 最新BitVisor 1.3のススメ
  - サポートプラットフォームの拡充
  - 性能向上
  - 画面描画やバックグラウンド暗号化等の新機能
- BitVisor改造テクニックを紹介
  - Makefileの仕組み
  - ログ出力
- “Undocumented” 内部APIの（一部を）
  - 画面描画機能
  - 保護ドメインとメッセージI/F
  - ディスクI/O

BitVisorの
エコシステムを
作りましょう！

実装したコードはオープンソースへ