

# IoT デバイス上の軽量実行環境における入出力制御の試作

2015SE035 加藤将太郎

指導教員：宮澤元

## 1 はじめに

近年、モノをインターネットに接続する Internet-of-Things(IoT) 技術が普及している。IoT 技術によりインターネットに接続された機器を IoT デバイスと呼ぶ。

IoT デバイスには、センサやアクチュエータなどが接続されているものが多く、インターネットを介して IoT デバイスを制御したりデータを得ることができる。しかし、IoT デバイスの計算リソースは限られているものが多く、セキュリティ面では脆弱性を突かれる可能性がある。

計算リソースの限られた IoT デバイスで効率的に IoT 用のアプリケーションを動作させるために、Unikernels と呼ばれる技術を IoT デバイスに導入するという試みが提案されている [2]。Unikernels とは、ライブラリ OS の技術を用いて、単一のアプリケーションをハイパーバイザ上で直接動作する仮想マシンイメージとして提供するシステムである [1]。実行に不必要な機能を排除しているので、実行イメージサイズが非常に小さく、起動が高速である。そこで、クラウドコンピューティングやエッジコンピューティングと Unikernels を組み合わせた技術の実用化に向けた研究が進められている。

しかし、Unikernels はもともと標準的な入出力装置だけを持つ仮想環境で利用されることを想定している。IoT デバイスが持つセンサやアクチュエータのような様々な入出力装置を利用するためには、これらの入出力装置を利用できるような入出力制御機能を Unikernels 内のライブラリ OS やハイパーバイザに実装する必要がある。

本稿では、IoT デバイス上で動作する Unikernels 環境で試作した入出力制御機能について述べる。Raspberry Pi 3 を IoT デバイスに見立て、GPIO に接続した LED を点灯・消灯するインタフェースを hvt [4]<sup>\*1</sup> と solo5 [3] を利用して実現する。

## 2 研究の背景

IoT デバイスには計算リソースが少ないものが多く、そのため十分なセキュリティ対策が取れないこともある。一方クラウドコンピューティングでは、軽量でセキュリティも考慮された Unikernels という技術が提案されている。そこで、IoT デバイスで Unikernels を用いることによって、計算リソースやセキュリティの問題点を解決しようというアプローチの研究が行われている [2]。本節では、Unikernels と、Unikernels をホスト型仮想化システムで動作させる hvt/solo5 について述べる。

### 2.1 Unikernels

Unikernels とは、ライブラリ OS の技術を用いて、単一のアプリケーションをハイパーバイザ上で直接動作する仮想マシンイメージとして提供するシステムである [1]。単一アプリケーションのコードと、ライブラリ OS 以外の機能を徹底的に排除しており、イメージファイルのサイズを削減することができる。Unikernels は、使用メモリが少なく軽量なので、起動時間が 1 秒以内と他の仮想マシンと比べて高速である。アプリケーションの実行に最適化して必要最低限の機能だけを含める形で実行イメージが作成されるので、セキュリティの脆弱性を突かれるリスクが非常に低い。

Unikernels は Xen[5] などのように、ハイパーバイザ上で高速な起動・実行が可能となる。従来の汎用 OS に存在するオーバーヘッドが削減でき、メモリフットプリントの割合を小さくすることができる。Unikernels は研究途中にあるため一般的にはまだ使われていないが、実用化に向けて様々な実験をしている研究者が多い。

図 1 は、Unikernels のイメージ図である。アプリケーションと OS が一体化することで、ハイパーバイザ上でアプリケーションが直接実行することができる。

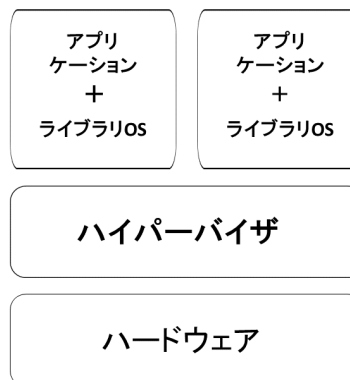


図 1 Unikernels の図

### 2.2 hvt/solo5

hvt は、64bit の Linux 環境にて動作する仮想化システムを前提とした軽量実行環境である。hvt はホスト OS 上で直接動作し、Unikernels とのインタフェースの役割を担う後述の solo5 と連携して、Unikernels の実行環境を提供する。

solo5 は、KVM を利用して Unikernels を動作させるために作られた仮想化ソフトウェアである。hvt と Unikernels とのインタフェースの役割をしており、hvt と Unikernels との橋渡し役を担っている。

<sup>\*1</sup> もともと ukvm の名称で開発されていたが、2018 年 9 月 13 日に hvt に改称された。

## 2.3 IoT デバイスへの Unikernels 導入による問題点

Unikernels は一定のハードウェア環境を仮想的に実現できる仮想マシンを前提としているので、IoT デバイスで利用するためには、仮想環境では通常提供されないような IoT デバイス特有の入出力装置に対応する必要がある。また、IoT デバイスの計算能力や計算リソースによっては、ハイパーバイザなどの仮想化環境を動作させられないこともありうる。

## 3 研究の内容

IoT デバイスに導入した軽量実行環境を用いて、IoT デバイスの持つ入出力装置の制御を行うプログラムを試作する。そのためには Unikernels から呼び出して入出力制御を行うハイパーコールを定義し、実装する必要がある。本研究では、Raspberry Pi 3 上の hvt/solo5 に入出力制御ハイパーコールを実装して、Unikernels から汎用入出力 (GPIO) に接続されている LED の点灯・消灯を行うことを目的とする。

## 4 入出力制御機能の試作

Raspberry Pi 3 に hvt をインストールし、Raspberry Pi 3 の入出力制御を行う hvt 用のプログラムを実装する。hvt は 64bit の Linux 環境にて動作するので、OS には Linux ディストリビューションである openSUSE Tumbleweed LXQt をインストールした。

### 4.1 LED 制御の仕組み

図 2 は Raspberry Pi 3 の GPIO の配線図である。今回は LED を GPIO 25 番に接続した。

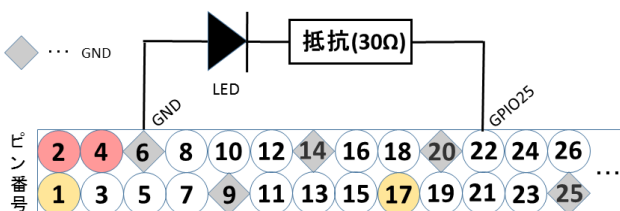


図 2 GPIO の配線図

Linux が提供する sysfs の機能を用いて GPIO の制御を行うことができる。今回利用したシステムでは GPIO 番号のオフセットは 458 であり、GPIO 25 番を制御するためには `/sys/class/gpio/gpio483/` を操作する。以下に GPIO 操作をするにあたって必要な、参照先と変更する値の手順の一例を示した。順番に値を変更することで LED が点灯・消灯する。

1. `/sys/class/gpio/export` を “483” に変更する。
2. `/sys/class/gpio/gpio483/direction` を “out” に変更する。
3. `/sys/class/gpio/gpio483/value` を “1” に変更する。
4. `/sys/class/gpio/gpio483/value` を “0” に変更する。

5. `/sys/class/gpio/unexport` を “483” に変更する。

## 4.2 実装内容

Unikernels からホスト OS の Linux の機能を直接利用することはできないので、ハイパーバイザである solo5 に Unikernels から呼び出して LED を制御するためのハイパーコールを定義する。solo5 は、ハイパーコールを hvt に中継し、実際の入出力制御は hvt 内で前節の sysfs を用いて行う。これにより、Unikernels からハイパーコールを用いて LED 制御を行うことができるようになる。

新たにファイルを作成したものは以下の通りである。以下に示すファイルパスは、カレントディレクトリである solo5 直下とする。

- tests/test\_led/test\_led.c
- bindings/hvt/led.c
- tenders/hvt/hvt\_module\_led.c

ファイルの内容を変更するものは以下の通りである。

- include/solo5/hvt\_abi.h
- include/solo5/solo5.h
- tenders/hvt/hvt\_core.c
- tenders/hvt/hvt.h
- tests/tests.bats
- tests/test\_led/GNUMakefile

モジュールである tenders/hvt/hvt\_module\_led.c にハイパーコールの定義をした。bindings/hvt/led.c にハイパーコールを実行するプログラムを書き、それを tests/test\_led/test\_led.c が読み込むことでハイパーコールを使用することができる。

## 5 おわりに

IoT デバイス上の軽量実行環境から入出力装置を制御する機能を hvt/solo5 を用いて試作した。IoT デバイスに見立てた Raspberry Pi 3 で LED を点灯・消灯させることができた。ハイパーコールをモジュールに定義することにより、LED 以外の入出力装置の制御を行うことも可能だと考えられる。今後は IoT デバイス上の Unikernels から入出力装置を制御するための一般的な枠組みを実現するための研究を行っていきたい。

## 参考文献

- [1] Anil Madhavapeddy and David J. Scott: “Unikernels: Rise of the Virtual Library Operating System”, Distributed Computing Vol. 11 No. 11, Jan., 2014.
- [2] Bob Duncan, et. al: “Enterprise IoT security and scalability: How unikernels can improve the status Quo”, In Proceedings of ACM UCC '16, pp.292-297 Dec., 2016.
- [3] Dan Williams: The Solo5 Unikernel, <https://github.com/djwillia/solo5/>.
- [4] Dan Williams and Ricardo Koller: “Unikernel Monitors: Extending Minimalism Outside of the Box”, In 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16) Jun., 2016.
- [5] Xen project: A Linux Foundation Collaborative Project, <https://www.xenproject.org/>.