

# プログラミング教育の実践例の紹介と Raspberry Pi ハッカソン

小出 洋<sup>1,a)</sup>

**概要:** 夏のプログラミング・シンポジウムのテーマのひとつ、プログラミング教育の実践例として、この夏に行った演習付きの2回の特別講義（セキュリティ・キャンプ九州 in 福岡, enPiT におけるクラウド分野の Web アプリケーションのサーバサイド）の概要と成果を報告した。さらに、夏のプログラミング・シンポジウムで同時に行われたハッカソンのテーマのひとつに Raspberry Pi 2 を用いた作品作りがある。ちょうどセキュリティ・キャンプ九州 in 福岡では、Raspberry Pi を利用したプログラミング演習を行ったので、その内容から Raspberry Pi 2 を用いた作品作りハッカソンの遂行に役に立ちそうなノウハウを紹介した。

**キーワード:** プログラミング教育, enPiT, Raspberry Pi, ハッカソン

## Introduce Practical Examples of Programming Education and a Hackathon for Raspberry Pi

**Abstract:** We introduce the results of practical examples of programming education, Security Camp Kyushu in Fukuoka and special lecture about server side programming of web applications in enPiT Cloud Field as one of themes of programming symposium 2015 in summer. These lectures includes practical exercises. Moreover, there is works of programming for Raspberry Pi in themes of hackathon in programming symposium 2015 in summer. We introduce some technical tips to help the hackathon because the lecture of Security Camp in Kyushu includes programming for Raspberry Pi.

**Keywords:** Programming Education, enPiT, Raspberry Pi, Hackathon

### 1. はじめに

2015 年度の夏の受講生のタイプも目的も異なるプログラミング演習付きの特別講義を2回行った。2015 年度夏のプログラミング・シンポジウム

のテーマには、「プログラミング教育の実践例」が含まれていたため、この2回の特別講義の概要と成果を紹介した。特別講演のひとつは、「セキュリティ・キャンプ九州 in 福岡」であり、もうひとつは「分野・地域を超えた実践的情報教育協調ネットワーク (enPiT) におけるクラウド分野の Web アプリケーションのサーバサイドに関する特別講

<sup>1</sup> 九州工業大学大学院情報工学研究院

<sup>a)</sup> koide@ai.kyutech.ac.jp

義」である。前者の概要を2章で、後者の概要を2.2章で紹介する。

さらに、夏のプログラミング・シンポジウムで同時に行われたハッカソンのテーマのひとつに「Raspberry Pi 2を用いた作品作り」がある。ちょうどセキュリティ・キャンプ九州 in 福岡では、Raspberry Pi を利用したプログラミング演習を行っていたので、その内容から Raspberry Pi 2を用いた作品作りハッカソンの遂行に役に立ちそうなノウハウを紹介した(3章)。

## 2. セキュリティ・キャンプ九州 in 福岡

「いじって壊して遊んでハッカーになろう」という題目で、IPA およびセキュリティ・キャンプ九州実施委員会が2015年8月28日(金)から8月30日(日)にかけて実施した講義のうちのひとつである。30日9時から12時までと13時から15時半にわたり5時間半かけて行った。受講者を公募し、書類による選抜が行われた。約2倍程度の倍率であった。最終的に採択された受講生20名は、大学生もしくは専門学校生であった。

この講義の目的は「情報技術を自分で楽しみながら更に高めるため、本物のマシンでプログラムを書いたり、オープンソースのコードをいじったり、壊したり、遊んだりできるようになること」である。このため講義内容は、自分で楽しみながら、いろいろな技術を試すことで、高度な情報技術に精通でき、成長できる構成にした。

### 2.1 導入

講義題目に「ハッカー」という言葉が含まれていることから、まずハッカーの定義について考察を行った。「ハッカー」は一般につきの2通りの意味を持つ。

- (1) コンピュータや電子回路について普通のひとより深い技術的知識を持ち、技術的課題をクリアできるひと
  - (2) 公開されていないシステムの内側に自分の技術ではいりこんで、破壊行動を行うひと
- ここでいう技術は「情報セキュリティ」に限っておらず、OS、ネットワーク、コンパイラ、電子

回路等の計算機科学に関するさまざまな技術が含まれていることを説明し、講義の目的についても説明した。受講者は技術に精通したいという希望を持って参加している。

#### 2.1.1 プラットホームとしての Raspberry Pi

計算機科学に関するさまざまな技術を試すためのプラットフォームとしては、現在は Raspberry Pi 2 Model B が最適である。サーバとして利用可能な十分な使用(表1)を持ち、古い技術から最新技術までのさまざまな実験、演習が可能である。クラウド上の仮想マシンではなく実機であり、35ドル程度で安価に販売されている。Raspberry Pi を利用するメリットについて説明し、Raspberry Pi の他の応用事例についても紹介した。

表1 Raspberry Pi 2 Model B (仕様)

プロセッサ	900MHz/ARM Cortex-A7 4 コア
GPU	24GFLOPS
主記憶	1GB LPDDR2 SDRAM
USB	2.0 ポート x4
二次記憶	microSD カード
GPIO	40 ピン
OS	Debian, Fedora, Arch Linux,... Windows 10 IoTCore
価格	35US ドル

#### 2.1.2 演習環境の構築

Raspberry Pi を利用するには、HDMI 接続のディスプレイ、USB 接続のキーボード、マウスが必要となる。これらを用意する代わりにノート PC を1台用意すれば、Raspberry Pi をネットワークに接続して、ssh でログインできる他、リモートデスクトップからも利用できる方法を紹介した(図1)。さらに今回のような演習や夏のプログラミング・シンポジウムにおけるハッカソンのように軽快にプログラミングを行いたいという場合に有効な方法もある(図2)。

#### 2.1.3 ARM アセンブラ

Raspberry Pi では、通常利用されているほとんどのプログラミング言語を利用できる。逆に Raspberry Pi で Native に開発可能である ARM プロセッサの機械語は、PC では行えない。ARM

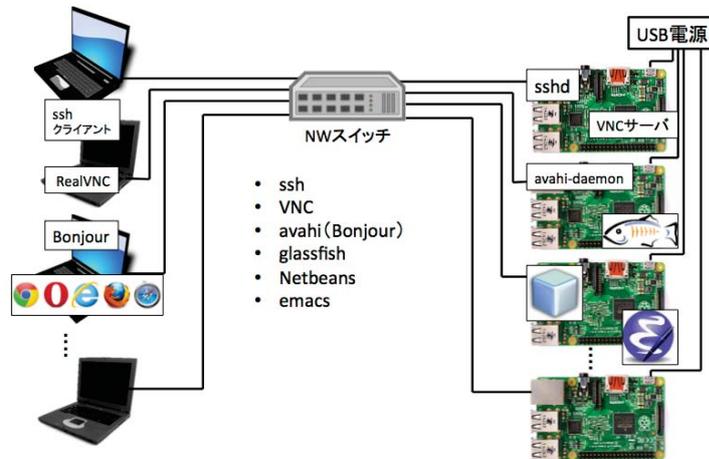


図 1 用意した演習環境.

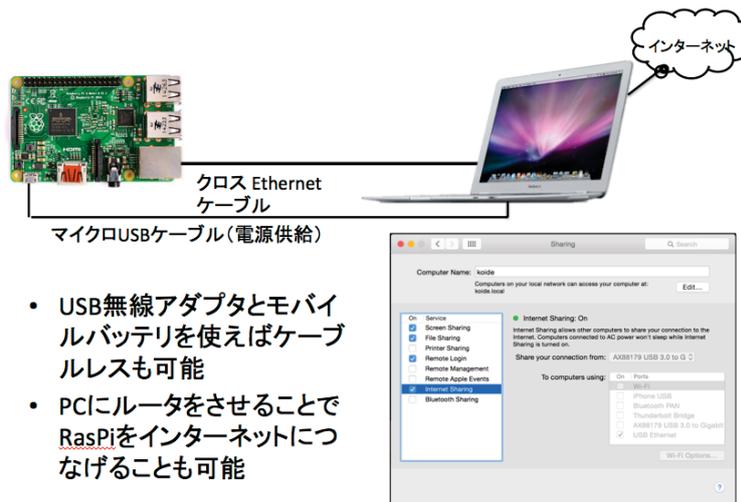


図 2 モバイル開発環境.

プロセッサは、AV 機器、カーナビ、携帯電話等の多くのデバイスに内蔵されており、事実上の業界標準である。夏のプログラミング・シンポジウムでも、ARM プロセッサの特徴的な命令体系について話題になっていた。また x86 系以外のアーキテクチャを学ぶことができる数少ないプラットフォームである。Netbeans IDE を用いた開発手法、および gdb を使ったデバッグについて紹介した。スライドの一部について、図3を紹介する。

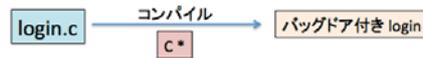
### 2.1.4 バックドア作り演習

システムプログラムや OS を理解することを目的として、自由に改変可能なオープンソースのシステムプログラムのソースコードを読んで書き換え、コンパイルして正規のものと入れ替える演習を行った。例題としては telnet サーバのソースコードを書き換えて、自分だけがいつでもログインできる裏口 (バックドア) を作るものとした。具体的には、telnetd から呼び出される telnetlogin に特定のユーザ名とパスワードをいれると、必ずロ



## login コマンドにおける hack

1. Cコンパイラがloginをコンパイルするときに特定のパスワードでログインできるようにするバックドアのコードを生成 (C\*)



2. Cコンパイラが自分自身をコンパイルするときにはいつも上記のコードを生成するようしておく (C\*\*)



3. ソースコードの形跡を消してから、Cコンパイラが自分自身をコンパイルしてもバックドアが残り続ける

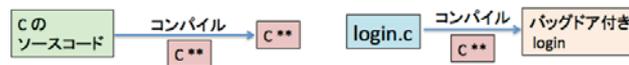


図 4 Ken Thompson のハッキング。

を記述して、簡単な Web アプリケーションを作成する演習としたが、この時点で時間が不足してきたため、本演習は割愛せざるを得なかった。

### 2.1.6 GPIO 演習

Raspberry Pi でしかできない演習のひとつに、いろいろな実世界の機器をインターネットに接続するという応用が考えられる。Raspberry Pi には、電氣的に用意に外部機器を接続できることを目的とする汎用ポート GPIO (General Purpose Input/Output) が 26 あり、ソフトウェアで自由に入出力を切り替えて利用することができ、センサやアクチュエータ等の外部の電子機器を接続できる。このとき間違った配線を行うと物理的に計算機を破壊することがある。この種の演習を行うには、万一間違った配線を行い破壊することがあっても被害が少なく済む Raspberry Pi のような安価な計算機が最適である。

その基本的なやり方を紹介するため、出力デバイスとしては LED、入力デバイスとしてタクトスイッチブレッドボードとジャンプワイヤを用意して Raspberry Pi に接続し、GPIO の制御、Python プログラムから利用する演習を行った (図 5、図 6)。

### 2.1.7 デバイスドライバ演習

Raspberry Pi は、オープンソースの OS である Linux で動作しており、GPIO に接続されている自作機器のデバイスドライバを記述して組み込むこ

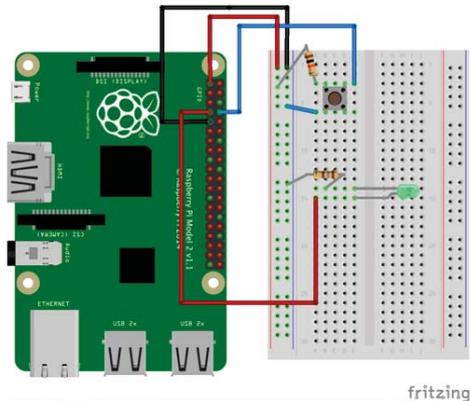
ともできる。デバイスドライバは謝った記述を行う OS の動作に影響を及ぼすことがある。そのため容易には試すことができず、デバイスドライバを作成するには難しいこととされていたため、いままではなかなか実験ができなかった。この種の演習を行うには、万一謝った記述を行ってもマイクロ SD メモリに OS イメージを入れ直せば済む Raspberry Pi が最適である。GPIO 演習で行った LED を点滅させる回路のためのデバイスドライバを Linux のカーネルモジュールとして作成し、OS に組み込む演習を実施した。

今回この演習は時間が足りず実施することができなかったが、受験者のひとりには実施後にテキストを参考に自習してくれ、Write Up として個人の blog に公開してくれている。それによると、彼はテキストの記述不足でうまく行かなかった点を参考情報を参照して、自己解決している。

## 2.2 enPiT サーバサイド特別講義

RESTful (Representation State Transfer) Web Service を用いる Web アプリケーションのサーバサイドを Jersey および EclipseLink, Mongo DB を用いて構築する演習付きの講義である。後期に開講している分散 PBL で学生らが自らアプリケーションサーバを構築できるようにすることを目的としている。enPiT のクラウド分野で連携している西日本の各大学 (大阪大, 神戸大, 和歌山大, 奈

## 配線してみよう



- shutdownしていったん電源を切ってから配線してね
- 配線したら隣の人と相互チェックをしてね



図 6 受講生が配線をしている様子。

良先端大、立命館大、京都工芸繊維大、九工大、九州産業大等)の大学院修士課程の1年生46名を対象に講義を行った(表2)。授業アンケートの結果は5点満点中4点程度であった。

- 表 2 RESTful Web Service Application の講義内容。
1. 講義の位置づけ
  2. RESTful Web サービス
  3. RESTful Web サービスが使われているところ
  4. HTTP
  5. 「演習 1」 RESTful Web アプリケーションを作る
  6. 「演習 2」 いろいろなりソースメソッドを試す
  7. 「演習 3」 パラメタを自作クラスで受け取ってみる
  8. 「演習 4」 XML による通信を試みる
  9. 「演習 5」 JSON による通信を試みる
  10. 「演習 6」 JPA による Mongo DB のアクセスを試す

### 3. 夏のプログラミング・シンポジウムにおけるハッカソン

2015年度の夏のプログラミング・シンポジウムでは、複数のテーマで参加者によるハッカソンが行われた。そのテーマのひとつに Raspberry Pi を用いたプログラミングがあった。2章で述べた「セキュリティ・キャンプ九州 in 福岡」の講義内容のうち、Raspberry Pi ハッカソンを実施するのに有用な情報が含まれているものを紹介した(2.1.1, 2.1.2, 2.1.3, 2.1.6の内容など)。特に図2に示したモバイル開発環境は非常に有用であり、ハッカソンでも同様の方法で開発している人が多数いた。

著者は以前に作成したプログラミング言語

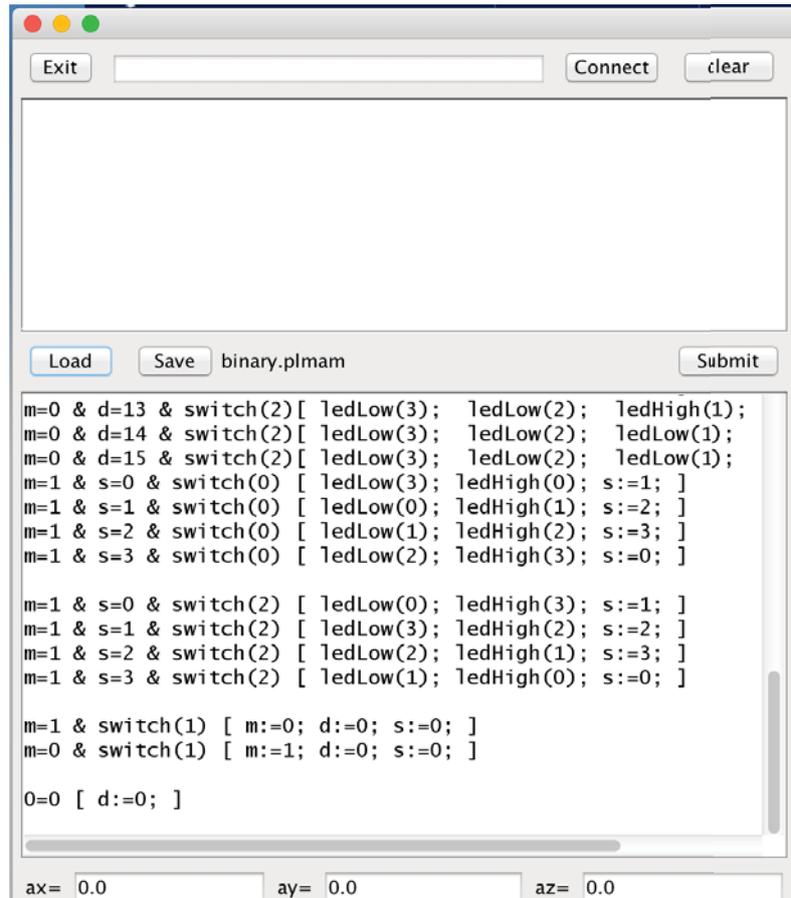


図 7 PLMAM1 の開発用 GUI.

PLMAM1[2] を Raspberry Pi 2 に移植した (図 7). その上で GPIO を操作するための命令を追加し LED やタクトスイッチを扱えるようにするとところまでハッカソンで開発を行った.

タクトスイッチでモードを変化させながら, LED を順次点点滅させるプログラム例を図 8 に示す.

#### 参考文献

- [1] Ken Thompson: **Reflections on Trusting Trust**, Communication of the ACM, Vol. 27, No. 8, pp. 761-763 August 1984.
- [2] 国立研究開発法人科学技術振興機構, 第 2 回科学の甲子園全国大会実技競技③実技マニュアル, [http://koushien.jst.go.jp/koushien/tournament/2012/2013/jitsugi03\\_jitsugi\\_manual.pdf](http://koushien.jst.go.jp/koushien/tournament/2012/2013/jitsugi03_jitsugi_manual.pdf) 2013.

#### 質疑・応答

原田 図 8 で, ledLow(3) ではなくて led(3, OFF) の方が良い.

小出 今回の Hackathon では複数の引数をとるようにはできなかったが, 今後そのようにしたい.

原田 図 8 のように繰り返しが多くなるので, 簡単にこの種のプログラムが書ける仕組みがあると嬉しいよね.

小出 取り組んでみたい面白い研究課題だと思います.

## 夏のプログラミング・シンポジウム「プログラム詠み会」2015.9.4-6

```

m=0 & d=0 & switch(0) [ ledLow(3); ledLow(2); ledLow(1); ledLow(0); d:=1; ]
m=0 & d=1 & switch(0) [ ledLow(3); ledLow(2); ledLow(1); ledHigh(0); d:=2; ]
m=0 & d=2 & switch(0) [ ledLow(3); ledLow(2); ledHigh(1); ledLow(0); d:=3; ]
m=0 & d=3 & switch(0) [ ledLow(3); ledLow(2); ledHigh(1); ledHigh(0); d:=4; ]
m=0 & d=4 & switch(0) [ ledLow(3); ledHigh(2); ledLow(1); ledLow(0); d:=5; ]
m=0 & d=5 & switch(0) [ ledLow(3); ledHigh(2); ledLow(1); ledHigh(0); d:=6; ]
m=0 & d=6 & switch(0) [ ledLow(3); ledHigh(2); ledHigh(1); ledLow(0); d:=7; ]
m=0 & d=7 & switch(0) [ ledLow(3); ledHigh(2); ledHigh(1); ledHigh(0); d:=8; ]
m=0 & d=8 & switch(0) [ ledHigh(3); ledLow(2); ledLow(1); ledLow(0); d:=9; ]
m=0 & d=9 & switch(0) [ ledHigh(3); ledLow(2); ledLow(1); ledHigh(0); d:=10; ]
m=0 & d=10 & switch(0) [ ledHigh(3); ledLow(2); ledHigh(1); ledLow(0); d:=11; ]
m=0 & d=11 & switch(0) [ ledHigh(3); ledLow(2); ledHigh(1); ledHigh(0); d:=12; ]
m=0 & d=12 & switch(0) [ ledHigh(3); ledHigh(2); ledLow(1); ledLow(0); d:=13; ]
m=0 & d=13 & switch(0) [ ledHigh(3); ledHigh(2); ledLow(1); ledHigh(0); d:=14; ]
m=0 & d=14 & switch(0) [ ledHigh(3); ledHigh(2); ledHigh(1); ledLow(0); d:=15; ]
m=0 & d=15 & switch(0) [ ledHigh(3); ledHigh(2); ledHigh(1); ledHigh(0); d:=0; ]

m=0 & d=0 & switch(2) [ ledHigh(3); ledHigh(2); ledHigh(1); ledHigh(0); d:=1; ]
m=0 & d=1 & switch(2) [ ledHigh(3); ledHigh(2); ledHigh(1); ledLow(0); d:=2; ]
m=0 & d=2 & switch(2) [ ledHigh(3); ledHigh(2); ledLow(1); ledHigh(0); d:=3; ]
m=0 & d=3 & switch(2) [ ledHigh(3); ledHigh(2); ledLow(1); ledLow(0); d:=4; ]
m=0 & d=4 & switch(2) [ ledHigh(3); ledLow(2); ledHigh(1); ledHigh(0); d:=5; ]
m=0 & d=5 & switch(2) [ ledHigh(3); ledLow(2); ledHigh(1); ledLow(0); d:=6; ]
m=0 & d=6 & switch(2) [ ledHigh(3); ledLow(2); ledLow(1); ledHigh(0); d:=7; ]
m=0 & d=7 & switch(2) [ ledHigh(3); ledLow(2); ledLow(1); ledLow(0); d:=8; ]
m=0 & d=8 & switch(2) [ ledLow(3); ledHigh(2); ledHigh(1); ledHigh(0); d:=9; ]
m=0 & d=9 & switch(2) [ ledLow(3); ledHigh(2); ledHigh(1); ledLow(0); d:=10; ]
m=0 & d=10 & switch(2) [ ledLow(3); ledHigh(2); ledLow(1); ledHigh(0); d:=11; ]
m=0 & d=11 & switch(2) [ ledLow(3); ledHigh(2); ledLow(1); ledLow(0); d:=12; ]
m=0 & d=12 & switch(2) [ ledLow(3); ledLow(2); ledHigh(1); ledHigh(0); d:=13; ]
m=0 & d=13 & switch(2) [ ledLow(3); ledLow(2); ledHigh(1); ledLow(0); d:=14; ]
m=0 & d=14 & switch(2) [ ledLow(3); ledLow(2); ledLow(1); ledHigh(0); d:=15; ]
m=0 & d=15 & switch(2) [ ledLow(3); ledLow(2); ledLow(1); ledLow(0); d:=0; ]

m=1 & s=0 & switch(0) [ ledLow(3); ledHigh(0); s:=1; ]
m=1 & s=1 & switch(0) [ ledLow(0); ledHigh(1); s:=2; ]
m=1 & s=2 & switch(0) [ ledLow(1); ledHigh(2); s:=3; ]
m=1 & s=3 & switch(0) [ ledLow(2); ledHigh(3); s:=0; ]

m=1 & s=0 & switch(2) [ ledLow(0); ledHigh(3); s:=1; ]
m=1 & s=1 & switch(2) [ ledLow(3); ledHigh(2); s:=2; ]
m=1 & s=2 & switch(2) [ ledLow(2); ledHigh(1); s:=3; ]
m=1 & s=3 & switch(2) [ ledLow(1); ledHigh(0); s:=0; ]

m=1 & switch(1) [ m:=0; d:=0; s:=0; ]
m=0 & switch(1) [ m:=1; d:=0; s:=0; ]

0=0 [ d:=0; ]

```

図 8 改良した PLMAM1 のプログラム例.