

# 第1章 仮想実験環境による組込みシステム 教育演習の取り組み

清尾 克彦<sup>1</sup>

## 1. はじめに

コンピュータを組み込んで実現される電子機器や制御機器は総称して「組込みシステム」と呼ばれ、デジタル情報家電や通信機器、自動車制御機器など日本のものづくり産業を支える基幹製品として、その発展が期待されている。半導体技術の進歩とデジタル化の進展により、組込みシステムは適用分野の拡大と高機能化による規模の拡大が進んでおり、これを担う組込みシステム技術者の育成が求められている。組込みシステムは、コンピュータ機能を含んだ LSI（大規模半導体集積回路）で実現されるハードウェアと、それを動かす「組込みソフトウェア」で構成されている。製品の応用分野の特性に応じてハードウェアとソフトウェアで機能を最適に分担する必要がある、組込みシステム技術者はハードウェアとソフトウェアの両方の技術を理解することが求められている。

組込みシステム技術者の教育では、組込みシステムの講義だけではなく、ハードウェアとソフトウェアによるものづくりのための演習が必須である。従来は、専用の演習室で Linux が搭載されたデスクトップパソコンやマイコン基板等の実機を使った演習が行われてきた。このような演習環境では、固定した時間がとれない社会人や遠隔地にいる受講希望者が教育を受けることが難しい状況にあった。

このような状況を改善するために、最近急速に発展してきた仮想化技術および仮想化技術を利用したクラウドコンピューティング技術を活用して、いつでもどこでも組込みシステムの演習ができるような仮想実験環境を提案する<sup>1)</sup>。本稿では、サイバー大学 IT 総合学部のテクノロジーコース・アーキテクチャプログラムでの演習科目やゼミナール、卒業研究で取り組んできた内容と残された課題について述べる。

## 2. 組込みシステム教育演習における課題

組込みシステムの世界では、実際にコンピュータを外部の装置に接続して動作させることから、講義形式だけでは必要な技術を身につけることができず、ハードウェアとソフト

---

<sup>1</sup> サイバー大学 IT 総合学部・教授

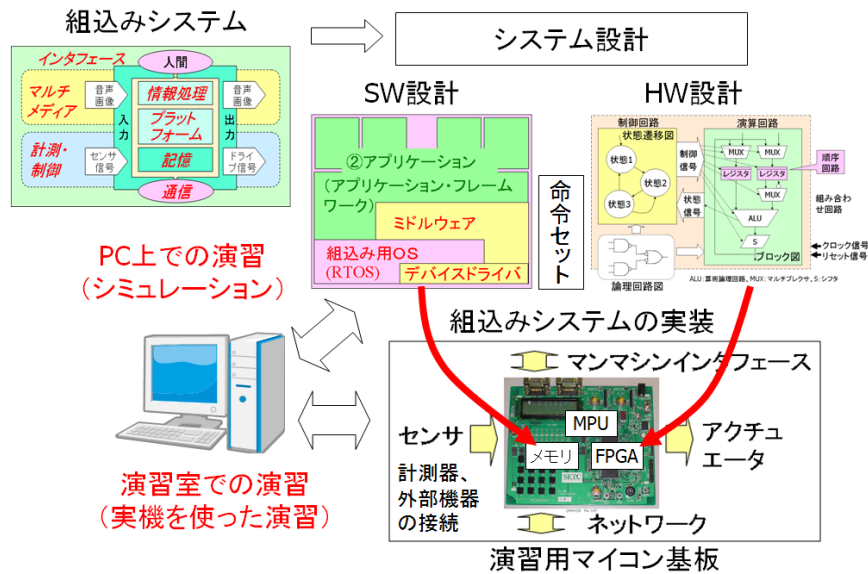


図1 組込みシステム教育演習のイメージ

ウェアを実際に製作し動作させる演習が必須である。組込みシステム教育演習のイメージを図1に示す。最初に、対象分野のニーズ（要求仕様）を満足するようにシステム設計を行う。このシステム設計では実現するアルゴリズムの評価を行い、ハードウェアとソフトウェアによる最適な機能分担を行う。この分担に基づいて、ハードウェアの設計とソフトウェアの設計が行われる。ハードウェアについては、システムレベル記述言語やハードウェア記述言語を使って記述し、コンピュータ上でシミュレーションにより動作確認（デバッグ）が行われる。ソフトウェアについては高位レベル言語（C, C++, JAVA 等）を使って記述し、同様にコンピュータ上でデバッグが行われる。ハードウェアの設計とソフトウェアの設計が完了すると演習用マイコン基板（プロセッサ, FPGA, フラッシュメモリ搭載）を使って組込みシステムの実機による確認（デバッグ）が行われる。ハードウェア構成情報を演習用マイコン基板に搭載されている FPGA (Field Programmable Gate Array) に、ソフトウェアのロードモジュールを演習用マイコン基板に搭載されているフラッシュメモリに、それぞれ USB 経由でダウンロードすることで実行可能となる。演習用マイコン基板に電源を投入することで、実機動作が開始される。リアルタイムで入力を与え、処理を行い、その出力結果に基づいて、所期の機能・性能が満足するかを確認することになる。

このような組込みシステムの典型的な設計の流れにおける教育演習の主な課題を図2に示す。  
 ①システム設計, ハードウェア設計, ソフトウェア設計で使われる開発環境が Windows や Linux などいろいろ混在している。特に Linux の場合は専門性が高く、以前は別の PC を準備する必要があった。図2では、システムレベル記述言語の SystemC<sup>2)</sup> が Linux 環境を必要とする。

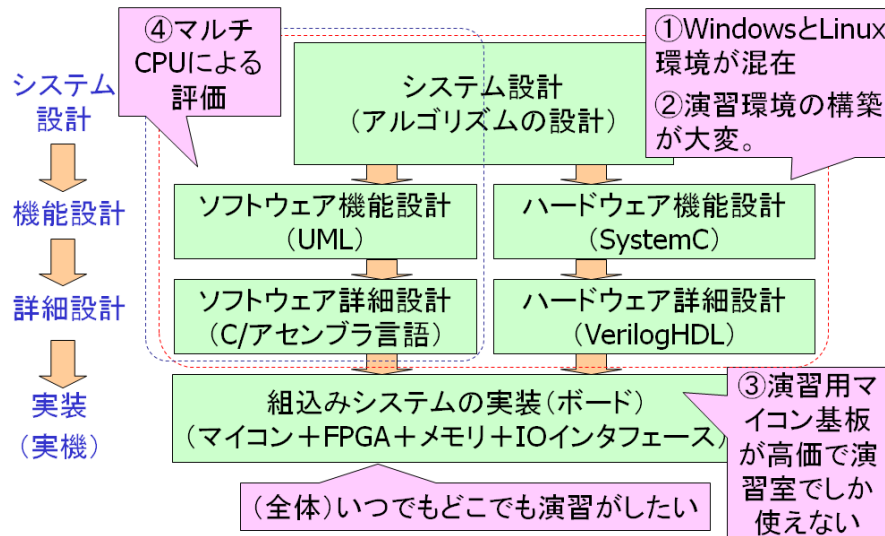


図2 組み込みシステム教育演習における課題

②個人のPCで演習を行う場合には、開発に必要なソフトウェアをダウンロードして所定の環境に設定する必要がある。使えるようにするためにかなりの手間がかかり、何らかの要因によりうまく動作しない場合もおきる。

③組み込みシステムを実装する演習用マイコン基板は比較的高価であることから、専用の演習室で使うのが一般的であり、個人で自由に利用することが難しい状況にある。最近では安価なマイコン基板が出現しているが、FPGAを搭載した高機能なものはまだまだ高価である。

④消費電力を抑えつつ性能を向上させる切り札としてマルチCPU化が進んでいる。組み込みシステムをマルチCPUで並列処理して高速化を検討する場合に、多くのCPUを搭載した高性能PCが必要となる。

このように組み込みシステム教育演習では、ハードウェアやソフトウェアの開発環境としてLinux環境を必要としたり、演習用マイコン基板のような設備を必要とすることから、専用の演習室で専用の時間帯で行われるのが一般的である。このため、遠隔地にいる受講希望者や社会人のように、限られた時間に特定の場所に拘束されることが難しい場合には、演習を受ける機会を持ってないことになる。

これらの組み込みシステム教育演習の課題を解決し、インターネットだけのeラーニング環境でいつでもどこでも演習を実現可能にするために、最近の仮想化技術および仮想化技術をベースとしたクラウドコンピューティング技術を活用した仮想実験環境について検討し、テクノロジーコース・アーキテクチャプログラムの一部の演習科目や卒業研究およびゼミナールで適用(予定を含む)した内容と残された課題について述べる。

### 3. 仮想実験環境の概要

#### 3.1. 全体の構成

今回取り組んだ仮想実験環境の全体像を図3に示す。受講生が演習で使うPCがクライアント側に、インターネットを介して演習環境のサービスを提供するのがサーバ(クラウド)側になる。4つの項目について検討を行ったが、最後の項目である遠隔実機演習環境の構築については、現状の調査までであり残された課題である。

今回の仮想実験環境の構築では、基本的にオープン系(オープンソースソフトウェア、オープンソースハードウェア、オープンクラウドサービス)のものや無償提供ツールを対象とした。また、受講生(クライアント側)のPCはWindows環境を基本としている。なお、Mac環境でも可能と思われるが検証をしていない。

##### 3.1.1. クライアント側での仮想マシン環境の構築(図3の(1)に対応)

組込みシステムの開発環境として大部分はWindows環境で対応できるが、一部はLinux環境を必要とする。Windows PC(ホストOS)上に仮想マシンを構築しゲストOSとしてLinux環境を立ち上げたものである。ハードウェア機能設計でのSystemCによる演習に適用した。

##### 3.1.2. 仮想デスクトップ環境の構築(図3の(2)に対応)

サーバ側のホストマシン上で仮想マシンのゲストOSとしてLinux(Windowsも可能)

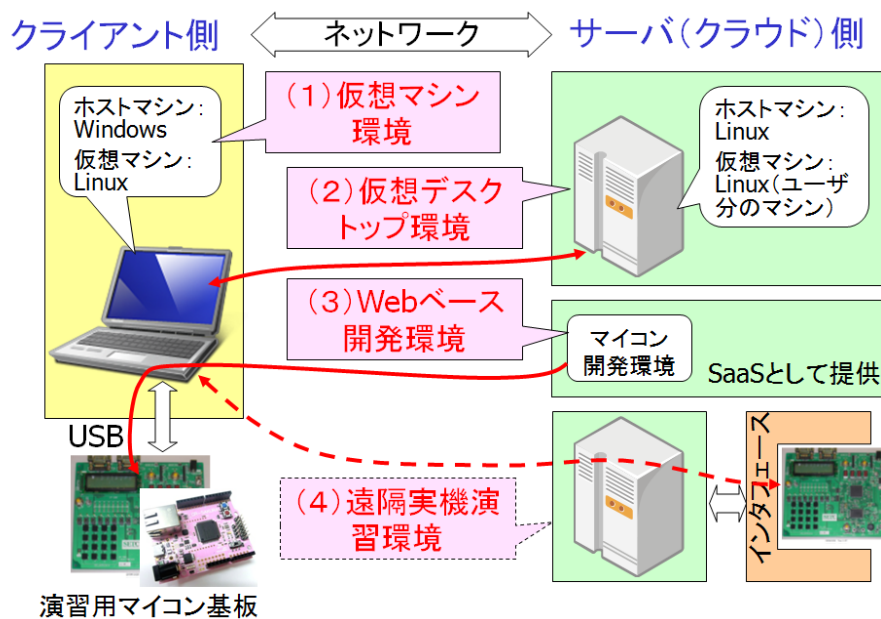


図3 仮想実験環境の全体像 ((4)遠隔実機演習環境は検討のみ)

## 第1章 仮想実験環境による組込みシステム教育演習の取り組み

を動かし、その上でアプリケーションプログラム（開発環境など）を動作させ、クライアント側の PC 上で仮想マシンのディスプレイの画面を表示できるようにしたものである。卒業研究でのマルチ CPU による静止画像圧縮伸長（JPEG）アルゴリズムの性能評価に適用した。

### 3.1.3. Web ベース開発環境の利用（図3の(3)に対応）

自分の PC 上に開発環境をインストールするのではなく、サーバ側の Web アプリケーションとして提供された Web ベース開発環境を利用するものである。最近のオープンソースハードウェアとして提供されるマイコン基板の一部で提供されるようになってきている。ゼミナールや研究プロジェクトでの演習で活用することを想定している。

### 3.1.4. 遠隔実機演習環境の検討（図3の(4)に対応）

組込みシステムで使用する演習用マイコン基板（FPGA 搭載版）は比較的高価であることから、個人で購入して演習を行うには負担が大きい。サーバ側の仮想マシンにこれらの設備を接続することにより、クライアント側の PC から遠隔でアクセスすることにより、あたかも自分が占有して使っているようなイメージで実機演習をサポートする仕組みである。すでに高等教育機関での事例が報告されており、今回は現状の報告をまとめた。

## 3.2. 仮想マシンの構築法

今回、クライアント側とサーバ側で仮想マシンを立ち上げた。仮想マシンの実現方法には、図4に示すようにホスト OS 型とハイパーバイザ型がある。ハイパーバイザ型には、完全仮想化型と準仮想化型および Linux KVM 型が存在する。

### 3.2.1. ホスト OS 型

ホスト OS 型は、ホスト OS（Windows, Linux など）上にアプリケーションとして仮想化ソフトウェアをインストールして利用するもので、ホストアプリケーションと並列に、仮想ハードウェア（ハードウェア機能を持つ仮想化ソフトウェア）の上でゲスト OS（Windows, Linux など）を動作させる形態である。仮想ハードウェアがホスト OS のアプリケーションとして動作し、ハードウェア動作をエミュレーションするので、ゲスト OS 上のアプリケーションの動作はそれほど高速ではない。ホスト OS 型の仮想化ソフトウェアとして、VMware 社の VMware Server や Microsoft 社の VirtualPC が有名であるが、2008 年当時に評価の高かった Sun Microsystems 社（現 Oracle 社）の VirtualBox<sup>3)</sup>を採用した。

### 3.2.2. ハイパーバイザ型

ホスト OS 型の課題である性能改善などを図るために、ハードウェアのすぐ上に仮想化ソフトウェア（ハイパーバイザと呼ばれる）を配置して、その上でゲスト OS を実行する

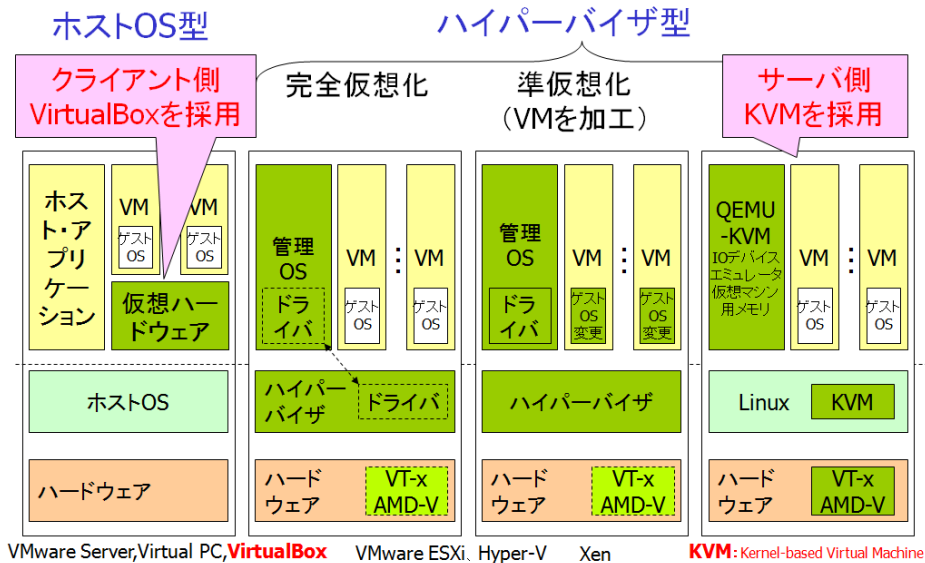


図4 仮想マシンの実現方法

形態である。ホスト OS 層がないので、ゲスト OS 上のアプリケーションの性能はホスト OS 型に比べて高くなる。実現方法には以下の 3 通りがある。

①完全仮想化型

完全仮想化では、実在するハードウェアを仮想化環境上でソフトウェア的にエミュレーションする形態で、組み込まれたデバイス管理機能（デバイスドライバ）を使ってデバイスの処理を行う。最近ではこれらのドライバをハイパーバイザにインストールするのではなく、ドライバを一元管理する管理 OS を別に用意して、その中のドライバを使う方式が採用されている。従来の OS のドライバをそのまま活用できるメリットはあるが、仮想環境上で何重ものソフトウェアが動作するため性能が低下するという欠点もある。

②準仮想化型

完全仮想化型の性能改善を図るために、仮想化ソフトウェア（ハイパーバイザ）上に仮想環境にとって都合のよい仮想ハードウェアを用意して、仮想マシン上のゲスト OS に仮想ハードウェアに対応したドライバを組み込んで利用する形態である。ゲスト OS のカーネルに変更を加える必要があるが、ハードウェアのエミュレーションのオーバーヘッドが減ることで性能は向上する。ゲスト OS からの IO 処理要求に関しては完全仮想化型と同じように管理 OS 上のドライバを使って処理が行われる。

③Linux KVM 型

ハイパーバイザとして Linux 自体を仮想マシンの実行基盤として機能させる形態である。

KVM (Kernel-based Virtual Machine) <sup>4)</sup> はオープンソースソフトウェアとして開発が行われており、現在は Linux カーネルのソースツリーにマージされている。ハードウェアによる仮想化支援機構 (Intel の VT もしくは AMD-V) が必須要件となっている。x86 系システムを構成する IO デバイスをエミュレーションする QEMU-KVM がホストアプリケーションとして動作する仕組みになっている。今回のサーバ側の仮想化ソフトウェアとして KVM を採用した。

#### 4. 仮想実験環境構築の取り組みと適用事例

##### 4.1. クライアント側での仮想マシン環境の構築と適用事例

クライアント側のホスト OS である Windows 上に仮想化ソフトウェアの VirtualBox をインストールし、VirtualBox の上でゲスト OS として Linux (今回は Fedora) をインストールした。VirtualBox による動作イメージを図 5 に示す。ディスプレイ画面にはホスト OS の Windows 画面が表示されており、VirtualBox を起動すると VirtualBox マネージャ画面が表示される。あらかじめインストールして設定したゲスト OS の Linux を起動するとゲスト OS の画面が表示される。ホスト OS 画面とゲスト OS 画面の間は、マウスで自由に制御を移すことができる。ホスト OS とゲスト OS 間でのデータのやり取りは共有フォルダを介して行うことができる。USB 接続機器についてもホスト OS への接続とゲスト OS への接続を自由に切り替えることができる。また、ゲスト OS から直接インターネットへアクセスすることもできる。このようにホスト OS である Windows 上で、ゲスト OS として Linux を簡単に扱うことができる。

テクノロジーコース・アーキテクチャプログラムの演習科目において、システムレベル記述言語である SystemC を使ったハードウェアの機能設計・検証に適用した。SystemC は IEEE で標準化されたシステムレベル記述言語で、プログラム言語である C++ のクラスライブラリとして無償で提供されている。ライブラリにはハードウェア記述の為の機能で

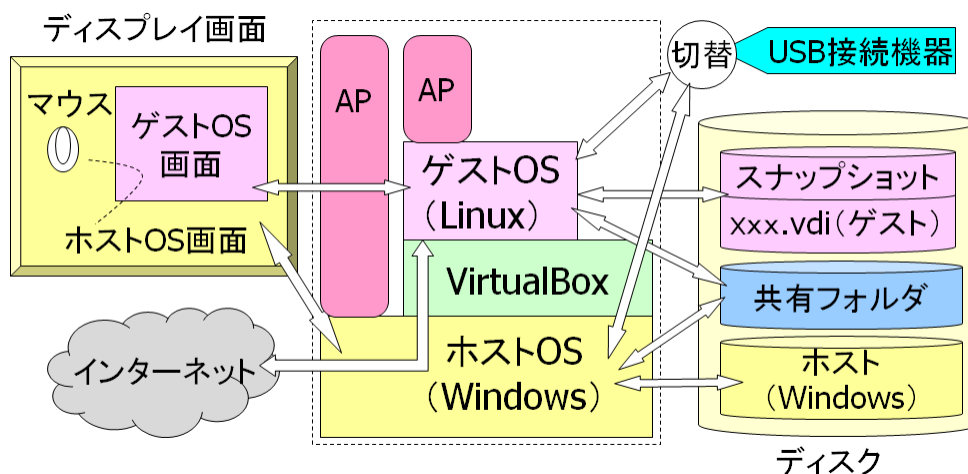


図 5 VirtualBox の動作イメージ

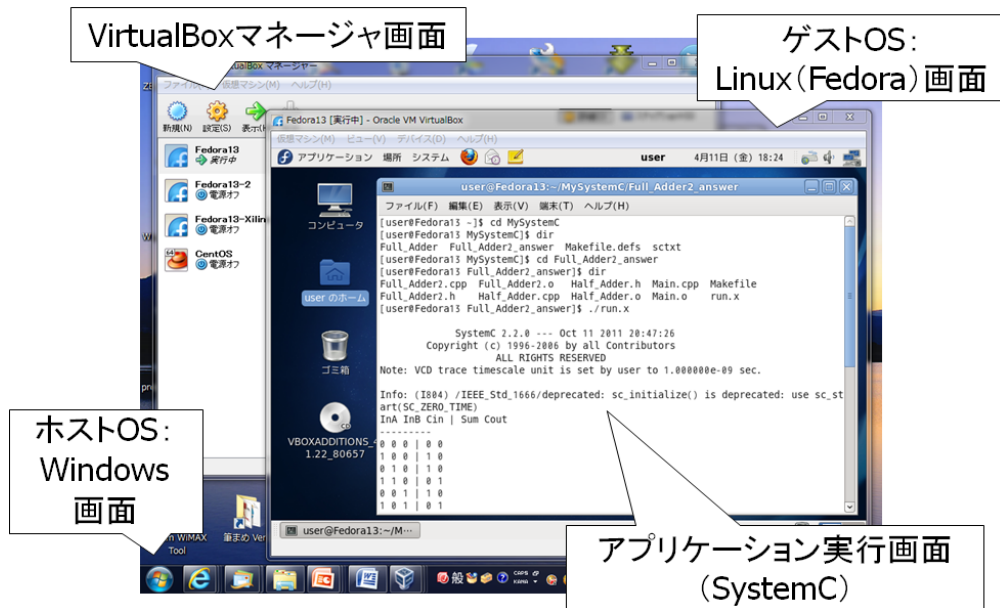


図 6 VirtualBox による実行例 (Fedora 上で SystemC を実行)

ある並列実行の概念やデータ型，それを扱う各種関数が定義されている。これらを使って書かれたプログラムは通常の C++コンパイラでコンパイルすることができ，生成されたオブジェクトはハードウェアのシミュレータとして動作する。

この SystemC の開発環境は，当初 Linux 版であったことから，今回は Windows 上のゲスト OS である Linux 上で動作させることにした。なお，最新版では Windows でも Mac OS X でもサポートされている。また，Linux は Fedora 以外でも問題ない。VirtualBox での実行例を図 6 に示す。

#### 4.2. 仮想デスクトップ環境の構築と適用事例

サーバ (クラウド) 側のホストマシン上に構築された仮想マシン上で演習を行い，その画面をクライアント側のディスプレイ画面に表示させながら処理を進めていく実行環境である。仮想デスクトップ環境の構築例を図 7 に示す。

サーバ側の実現方法として，パブリッククラウドの IaaS (Infrastructure as a Service, たとえば AWS : Amazon Web Service) の上で仮想マシンを立ち上げることができるが，今回はプライベートな研究室のサーバ環境において，3.2 章で説明したオープンソースソフトウェアである KVM により仮想化環境を構築した。サーバ側のホスト OS として Linux (今回は Fedora) を採用し，仮想化ソフトウェアとして KVM をインストールした。KVM のもとでゲスト OS として同じ Linux (今回は Fedora) を導入した。

仮想デスクトップ環境を実現する遠隔操作プロトコルとして，Microsoft の RDP (Remote Desktop Protocol) や Citrix 社の ICA (Independent Computing Architecture)



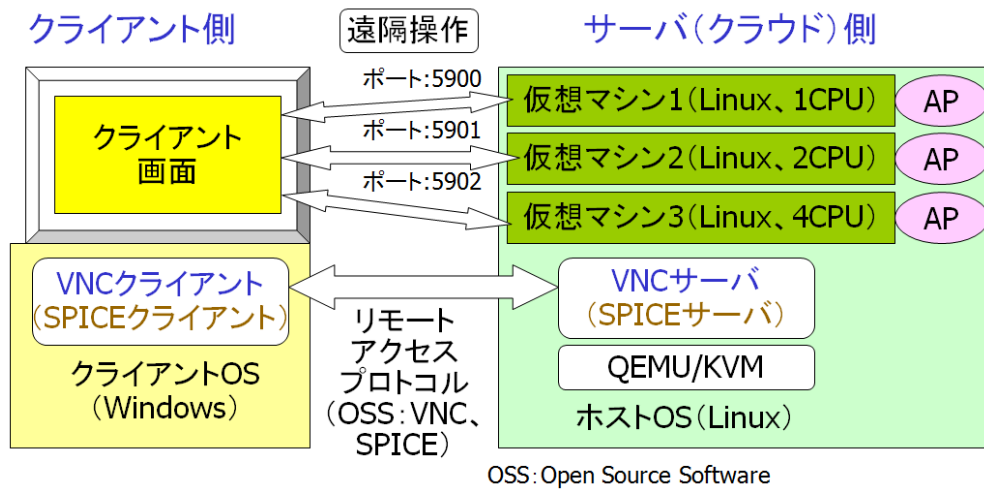


図7 仮想デスクトップ環境

が有名であるが、今回はオープンソースソフトウェアで良く使われている VNC (Virtual Network Computing)<sup>5)</sup> を採用した。VNC は性能的に遅いところがあり、RedHat 社が開発し映像・音声伝送等を得意とするオープンソースソフトウェアの SPICE<sup>6)</sup> を評価したが、評価時点の 2010 年では動作が安定しなかったので使用を断念した。クライアント側に VNC (または SPICE) クライアントをインストールし、サーバ側に VNC (または SPICE) サーバをインストールすることにより、クライアント側から遠隔操作することが可能になる。

図7の例ではサーバ側に高性能な 4CPU 構成の PC を採用し、3 種類の仮想マシンを設定した。ゲスト OS はすべて Linux (今回は Fedora) を採用し、CPU 構成を 1CPU, 2CPU, 4CPU の 3 種類として、マルチ CPU による並列処理による性能評価を行った。それぞれの仮想マシンに異なるポート番号を割りつけて、VNC クライアントからポート番号を指定してアクセスすることで、該当する仮想マシンを遠隔から操作することができる。

静止画像の圧縮伸張を行う JPEG を題材に OpenMP<sup>7)</sup> を使って並列処理を適用した例を図8に示す。OpenMP はプログラムの中に「プラグマ」と呼ばれる並列化を指示するディレクティブを入れることで並列化を行うオープンソースソフトウェアである。図8は、CPU コア 1 個、2 個、4 個の環境で実行した時の各 CPU の使用率のグラフを示したものである。CPU の個数が増えるのに従い性能は向上するが、並列化のためのオーバーヘッドにより、性能向上の比率が低くなるのが解る。

#### 4.3. Web ベース開発環境の利用と適用事例

4.1 章の場合も含めて、受講生が自分の PC (クライアント側) に演習環境をインストールする場合には、それ自体が手間のかかる作業であり、またそれぞれの環境 (OS の種類・バージョン、CPU 性能、メモリ容量、ディスク容量など) が異なるために予期せぬトラブル

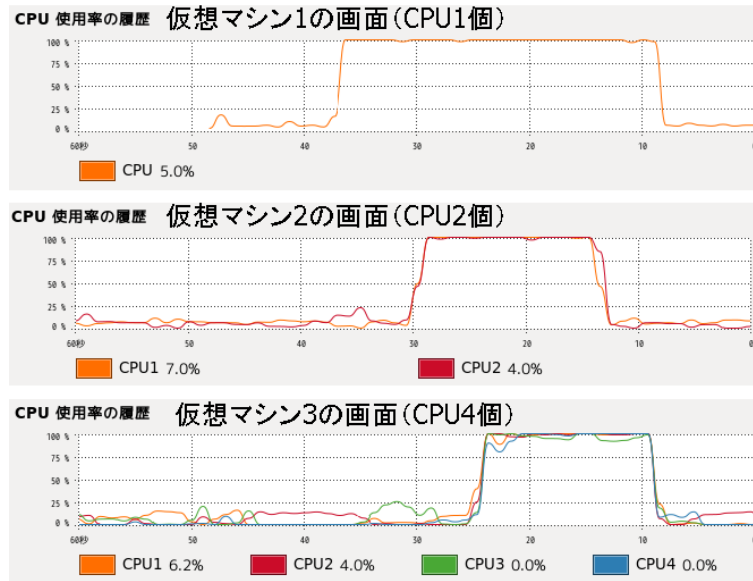


図 8 JPEG のマルチ CPU 環境での実行例

ルが発生することがある。また、オープンソースソフトウェアの開発環境も時々刻々バージョンが変わるのでその対応も必要である。このような課題を解決するために、4.2 章のようにサーバ側に開発環境を準備して使わせる方法を紹介したが、最近では Web アプリケーションとして Web ベースの開発環境を無償で提供するものが出現するようになってきた。

最近注目されているフィジカルコンピューティング<sup>8)9)</sup>の世界で使われるオープンソースハードウェアに準拠したマイコン基板が登場してきており、コンピュータに不慣れな人でも手軽にソフトウェアを作って動かせる環境が整ってきた。特に有名なイタリア発の Arduino<sup>10)</sup>の世界では、プログラムのことをスケッチと呼び、ベースとなるマイコン搭載基板にオプションとなるハードウェアモジュール(シールドと呼ぶ)を積み重ねることで、容易に組み込みアプリケーションを実現できる。

Web ベースのマイコン基板の開発環境の概要を図 9 に示す。あらかじめユーザ登録することにより無償で使うことができる。マイコン基板 (この例ではルネサスエレクトロニクス社の GR-SAKURA<sup>11)</sup>:Arduino 互換)を USB 経由クライアント側の PC に接続すると、サーバ側の開発環境 (この例では GR-SAKURA 開発環境)につながり、開発環境の画面 (この例では Web コンパイラ)が表示される。ユーザはこの画面の中でソースプログラムを入力し、コンパイルすることができる。コンパイルエラー無くなると、生成されたバイナリコードをサーバ側から USB に接続されたマイコン基板にダウンロードする。PC から見ると USB に接続されたマイコン基板は USB メモリのように扱われている。

このようにマイコン基板を USB に接続するだけで、すぐにプログラムを作成して動かしてみることが容易にできる。Web コンパイラ画面での実行例を図 10 に示す。GR-SAKURA 以外にも TI 社が提供する BeagleBone<sup>12)</sup>では、Linux ベースの JavaScript

## 第1章 仮想実験環境による組込みシステム教育演習の取り組み

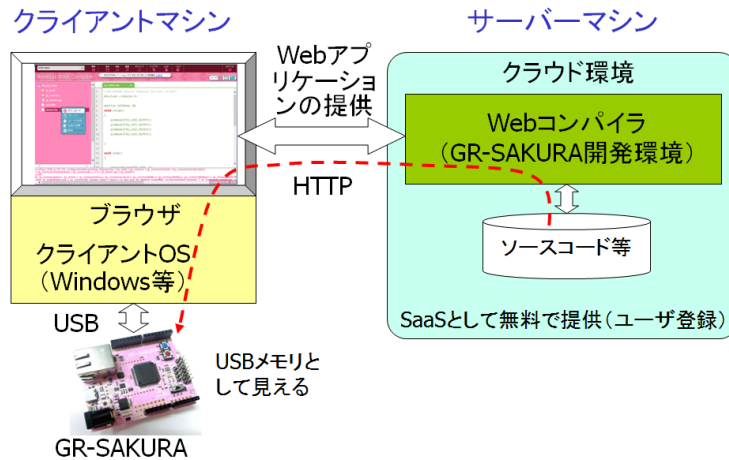


図9 Web ベース開発環境の概要

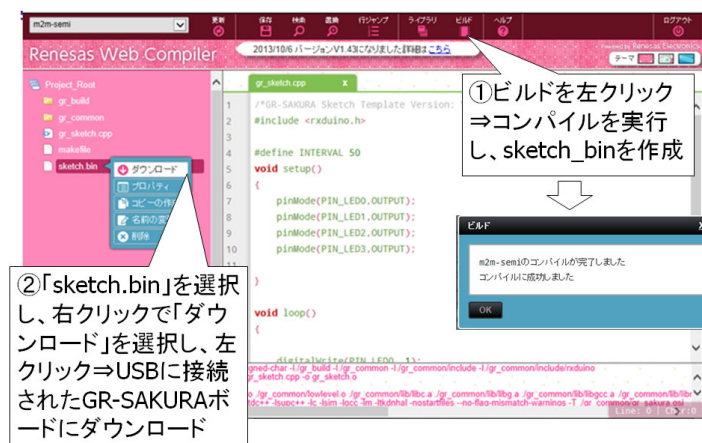


図10 Web ベース開発環境 (Web コンパイラ) の実行例

開発環境を Web ベースで提供している。ゼミナールの M2M (Machine to Machine) /IoT (Internet of Things) に関する演習での活用を予定している。

### 4.4. 遠隔実機演習環境の検討

組込みシステムのハードウェア設計演習では、FPGA 搭載基板が使われるが、比較的高価であることから、専用の演習室に設置された設備を使って行うケースが大部分である。通学式の高等教育機関では一般に専用の演習室で専用の時間帯で演習を行うことになるが、決められた時間内に終わらなかった場合や自宅で復習したい場合などに対応することができなかった。また、社会人向けの講習会では、さらに制約が多く、いつでもどこからでも

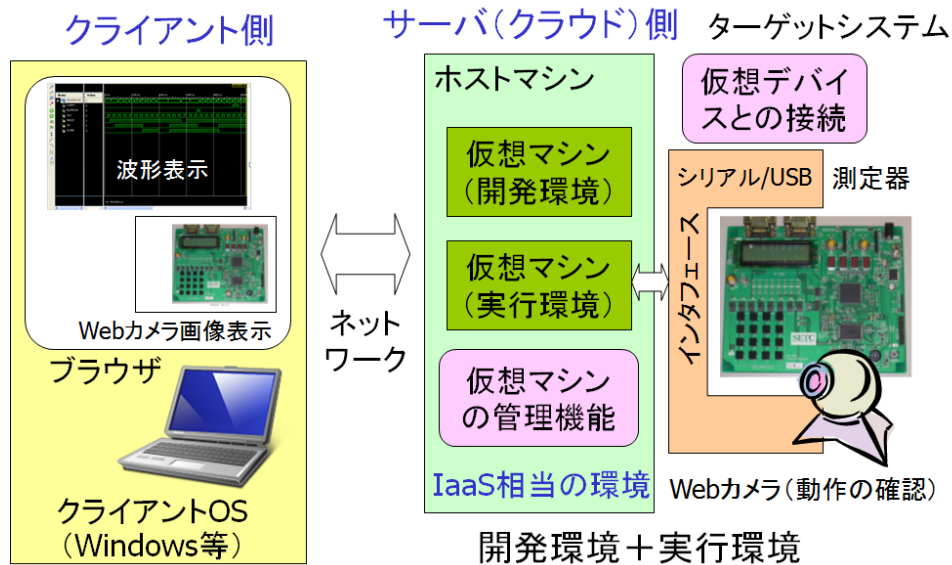


図 11 遠隔演習環境の構築イメージ

使いたいという要望に対応できなかった。このような課題を改善する目的で、組込みシステム技術者向けの遠隔実機演習環境について、既にいろいろな取り組み<sup>13)14)15)16)17)</sup>行われている。入力の与え方、出力（特にディスプレイ画面）の表示の仕方、Webカメラによる実験環境の映像表示、演習用マイコン基板の排他的使用管理、セキュリティ等について、それぞれ工夫がされているが、まだいろいろな課題が残っている。

サイバー大学のようにすべてインターネットを使ったeラーニング環境で、本格的な組込みシステム教育演習を行うには、いつでもどこでも使える遠隔実機演習環境の構築が必要になる。従来の先行事例を参考に、最近の仮想化技術を取り込んだ遠隔実機演習環境の構築イメージを図11に示す。

サーバ側の仮想マシンの管理（ユーザ管理を含む）や仮想マシンと仮想デバイスであるマイコン（FPGA搭載）基板との接続など、具体的な実現方法について検討を進めていく必要がある。今回は構想の検討までで、実現は今後の課題である。

## 5. まとめ

仮想化技術および仮想化技術をベースとしたクラウドコンピューティング技術を活用して、いつでもどこからでも組込みシステムの演習ができる仮想実験環境について、いろいろな角度から取り組んだ。

ハードウェア機能設計では、Linux版のSystemCの開発環境を必要としたことから、クライアント（受講生）側のWindows PCに仮想化ソフトウェアとしてVirtualBoxを導入し、その上でゲストOSであるLinux（Fedora）を立ち上げ、その上でSystemCを実

## 第1章 仮想実験環境による組込みシステム教育演習の取り組み

行した。受講生にとって、このような開発環境の立ち上げは負担が大きかった面があるが、立ち上げ後の演習において性能的な問題や大きなトラブルもなく運営することができた。また、Windows PC 上で手軽に仮想マシンを立ち上げてゲスト OS を動作させる手法を習得できたので、いろいろな面で応用することが期待できる。

卒業研究のテーマによっては、受講生の手持ちの PC のスペックでは評価できないケースがあり、サーバ側の高性能な PC 上に仮想デスクトップ環境を構築して対応した。サーバ側には仮想化ソフトウェアとしてオープンソースの KVM を導入し、その上に CPU の構成の異なる複数個の仮想マシンを立ち上げて、並列処理の評価に適用した。テーマが決まってから、急遽仮想マシン環境を構築したが、オープンソースにより比較的タイムリーに対応することができ、卒業研究の成果に反映することができた。今回は研究室のプライベートなサーバ環境で実現したが、今後はパブリックな IaaS 環境での適用も検討する必要がある。

Web ベースの開発環境 (SaaS としての提供) については、クラウドコンピューティングのさらなる発展に対応して、今後いろいろな分野で増加していくと考えられる。今回は、ゼミナールや研究プロジェクトでの活用を想定して、フィジカルコンピューティングで有名な Arduino と互換性のある GR-SAKURA マイコン搭載基板に対応した Web ベースの開発環境 (Web コンパイラ) を評価した。マイコン搭載基板を PC の USB に接続するだけで、Web ベースの開発環境にアクセスでき、作成したバイナリコードをダウンロードしてすぐに動作させることが可能になることがわかった。

最後に、遠隔実機演習環境について、先行事例を参考に検討を行った。進歩の著しい仮想化技術と仮想化技術をベースとしてクラウドコンピューティング技術を活用して構築するためには、いくつかの課題を解決する必要がある。最近では、半導体の急速な進歩で、ARM のような高性能な CPU が安く提供されるようになり、個人で購入できる時代を迎えようとしているが、FPGA を含めるとまだ高価であり、オープンソースで構築可能な遠隔実機演習環境の実現が望まれる。

テクノロジーコース・アーキテクチャプログラムの演習科目や卒業研究で仮想実験環境を実際に利用し、安定運用に協力いただいた受講生各位に感謝する次第である。

### 注および参考文献

- 1) 清尾克彦:「組込みシステム教育における仮想実験環境の検討」, 平成 23 年度電気学会 電子・情報システム部門大会講演論文集, 電気学会 電子・情報システム部門, p.760-761 (2011)
- 2) SystemC の HP : <http://www.accelera.org/downloads/standards/systemc> (2014. 4. 13)
- 3) VirtualBox の HP : <https://www.virtualbox.org/> (2014. 4. 13)
- 4) KVM の HP : [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page) (2014. 4. 13)
- 5) VNC の HP : <http://www.cl.cam.ac.uk/research/dtg/attarchive/vnc/> (2014. 4. 13)
- 6) SPICE の HP : <http://www.spice-space.org/home.html> (2014. 4. 13)

- 7) OpenMP の HP : <http://openmp.org/wp/> (2014. 4. 13)
- 8) Tom I. and Dan O. : 「Physical Computing: Sensing and Controlling the Physical World with Computers」, Thomson (May, 2004)
- 9) フィジカルコンピューティング : ニューヨーク大学のダン・オウサリバン教授が提案した思想で, 「コンピュータの機能上の制約を超え, 日常生活環境に沿った人 (身体) と端末との新しいインタラクションを探ることで, 身の回りの世界に新しい広がり生まれるのではないか」という考えに基づいている。  
(<http://itpro.nikkeibp.co.jp/article/COLUMN/20130812/497865/?ST=smartphone&P=2> より引用)
- 10) Arduino の HP : <http://www.arduino.cc/> (2014. 4. 13)
- 11) GR-SAKURA の HP : <http://sakuraboard.net/gr-sakura.html> (2014. 4. 13)
- 12) BeagleBone の HP : <http://beagleboard.org/Products/BeagleBone> (2014. 4. 13)
- 13) 中村立人他 : 「組込みシステム開発演習における授業時間外学習を支援する遠隔実習システム」, 信学技報 IEICE Technical Report ET2009-80 (2009-12), pp. 163-167
- 14) 金在成他 : 「e ラーニングと遠隔 FPGA の連携による異分野共同研究環境の開発」, 信学技報 IEICE Technical Report VLD2009-74, CPSY2009-56, RECONF2009-59 (2010-01)
- 15) 千葉慎二 : 「組込みシステム設計教育のための遠隔実習システムの開発と運用評価」, 平成 20 年度情報教育研究集会講演論文集, pp. 339-342 (2008)
- 16) 山脇彰, 芹川聖一 : 「VGA による動作可視化機構を備えたプロセッサ教育支援システム」, FIT2009 第3分冊, pp. 77-82 (2009)
- 17) 齋藤正和他 : 「WWW を利用したデジタル回路遠隔実験」, 信学技報 IEICE Technical Report ET2003-108 (2004-3), pp. 59-64