

Android 端末を活用した 機械語プログラミング演習環境の開発

古庄 裕貴^{1,a)} 中西 恒夫^{1,b)}

概要: 機械語プログラミングを通して、ノイマン型計算機の動作原理、さらにはハードウェアとソフトウェアの知識をつなぎ立体的に理解させることを目的として、基本情報処理技術者試験で用いられる仮想の計算機、COMET II の Android 端末向けエミュレータを実現する。単にコンピュータの原理を理解させるのみならず、学生の動機付けを図る「面白い」課題を演習において与えられるよう、当該エミュレータは楽音再生や図形描画等の入出力機能を設けている。学部2年生を対象に当該エミュレータを用いた演習を実施した結果、これらの入出力機能を用いた応用課題を達成した学生からは良好な反応を観察することができた一方、演習の現場において教員や学生に必要な支援を与える仕組みが望まれることも明らかになった。

キーワード: 機械語プログラミング, エミュレータ, COMET II

Implementation of Android Based Exercise Environment for Machine Language Programming

HIROKI FURUSHO^{1,a)} TSUNEO NAKANISHI^{1,b)}

Abstract: An Android based emulator for COMET II, a virtual computer used in the fundamental information technology engineer examination, is implemented to make students understand the principle of the von Neumann machine and integrate knowledge on hardware and software through machine language programming. The emulator equips input and output functions such as playing musical tones, drawing pictures etc. not only to make students understand the principle of the computer but also to motivate them with interesting exercises. The authors conducted exercises for sophomores with the emulator and observed positive responses from them especially for applied exercises with those input and output functions, while it needs a mechanism helping instructors and students in the exercise to be implemented.

Keywords: Machine language programming, Emulator, COMET II

1. はじめに

コンピュータを観念的にしか理解していない学生に対して、機械語、あるいはアセンブリ言語に関する演習を実施する重要性は大きい。機械語やアセンブリ言語に関する演習によって、学生がそれまで学んできた、あるいは今後学ぶコンピュータアーキテクチャや論理回路といったハードウェアに関する知識と、C言語などの高級言語によるプログラミング、オペレーティングシステム、コンパイラといったソフトウェアに関する知識をつなぎ、抽象度の異なるこれらの事柄を立体的に理解させる効果が期待でき

る。筆者らの所属する福岡大学工学部電子情報工学科では、そうした効果を期待して、以前は市販のスタンドアロン型マイコン学習教材を用いた機械語によるコーディングの演習を1年生の基礎演習で実施していたが、カリキュラム改正に伴い、2016年度からは2年生を対象とする「電子情報工学実験」で実施することとなり、この機にAndroidタブレット端末ベースの機械語／アセンブリ言語学習システム「COMET II エミュレータ」を開発し実験に投入することとした。本稿で述べるCOMET II エミュレータは、基礎情報処理技術者試験で用いられている仮想の計算機、COMET II のエミュレータであり、Androidアプリケーションとして実現されている。機械語プログラムをステップ実行でき、レジスタやメモリの状態を常時、参照および変更できるようにしており、学生は機械語プログラムの

¹ 福岡大学 電子情報工学科
Fukuoka University, Johnan, Fukuoka 819-0180, Japan

^{a)} furusho@fukuoka-u.ac.jp

^{b)} tun@fukuoka-u.ac.jp

個々の命令に対するプロセッサの挙動を目視できるようになっている。これによって学生にノイマン型計算機概念と動作原理を理解せしめることを期待している。さらに、COMET II のスーパーバイザ命令で図形描画や楽音再生ができるようにしており、それによって学生の関心をひき、ゲームの実装といった自由課題を通して創意工夫を発揮できる余地を残している。

COMET II エミュレータにはクラウドと連携する機能も持たせている。近年、学生の質保証のための成績評価の厳格化が強く求められ、そのため成績評価のプロセスを定めエビデンスを残すことが必要となっているが、それらの確実な実施には一定の負担を要する。エビデンスとなる学生の成果物や学習履歴をクラウドに自動的に吸い上げることで、これら教育に係る「間接業務」の負担を下げ、演習内容の充実にリソースを割けるようにする。また、演習の間、限られた教員ですべての学生をくまなく指導することは困難を伴う。クラウドに吸い上げた学生の成果物や学習履歴を分析することで、演習についてこれしていない早期に学生を見つけ必要な支援を提供したり、学生がよく陥っているミスを精査し指導内容の改善を図ったりすることが可能となる。

以下、本稿では第2章では開発した COMET II エミュレータについて述べ、第3章では当該エミュレータを用いた福岡大学における演習について報告する。最後に第4章で結論と今後の課題について述べる。

2. COMET II エミュレータ

2.1 概要

著者らの開発した COMET II エミュレータは、独立行政法人情報処理推進機構 (IPA: Information-technology Promotion Agency) の実施する基本情報処理技術者試験で使用される仮想の計算機、COMET II の命令の挙動をその仕様 [1] 通りに再現するエミュレータである。我が国の学部のコンピュータアーキテクチャ教育では、Hennessy の教科書が広く使われていることもあり、MIPS の命令セットがよく教授され、MIPS のエミュレータ SPIM も広く利用されている。しかし、福岡大学工学部電子情報工学科のディプロマポリシーでは、学部卒業生の要件として上述の基本情報処理技術者試験に (特に追加の勉強をしなくとも) 合格する程度の専門知識を有することが定められており、そのため COMET II を演習で学ぶ命令セットとして採用している。

COMET II エミュレータは Android 4.4 以上で動作するアプリケーションでありタッチスクリーンを備える端末上で使用できる。PC ベースのアプリケーションではないため、端末が充電されてさえいれば、どのような教室でも COMET II エミュレータを用いた演習を実施することができる。現在、市場に流通しているスマートフォンやタブ

レット端末で使用でき、意欲のある学生は自身のスマートフォン上で自学することも可能である。また、Android 端末の低価格化が進んでおり、学科で安価な端末を人数分買い揃えてもさほどの経済的負担とはならない。実際、本学での演習は1台あたりわずか17,000円で購入した Amazon Kindle Fire HD7 を用いて実施している。

COMET II エミュレータはコンテキスト編集画面 (図1) と実行画面 (図3) とで構成されている。コンテキスト編集画面は、汎用レジスタ、プログラムレジスタ (=プログラムカウンタ)、スタックポインタ、フラグレジスタ、メモリといった COMET II が備える構成要素の状態を確認ならびに編集できる画面である。実行画面は機械語プログラムによって出力される文字や図形を表示する画面である。

COMET II エミュレータは以下の機能を備えている。

- **レジスタ表示/編集機能:** COMET II が備える各種レジスタの値を表示する。また、レジスタの表示領域をタップすることでそのレジスタに保存されている値を変更することができる。
- **メモリ表示/編集機能:** メモリの状態を表示する。また、メモリの表示領域をタップすることで、レジスタと同様に、選択したアドレスに格納されている値を変更することができる。
- **ステップ実行機能:** プログラムレジスタが示すメモリアドレスに格納されている命令を一つ実行する。
- **連続実行機能:** 現在プログラムレジスタが示しているメモリアドレスから命令を連続的に実行する。命令の実行間隔は高い精度は保証されていないが、1ミリ秒単位で指定できる。実行の中断も可能である。
- **コンテキスト保存/復元機能:** COMET II エミュレーション環境のコンテキスト、すなわちレジスタとメモリに格納されている全てのデータを、任意の名前をつけて端末内にファイルとして保存できる。また、保存したファイルからエミュレーション環境のコンテキストを復元することも可能である。
- **成果物提出機能:** 保存したデータを成果物としてサーバにアップロードする。

2.2 スーパーバイザコール命令による機能拡張

COMET II の仕様には、転送、加減算、論理演算、シフト演算、分岐、スタックに係る命令が定められているのみであり、入出力に関してはマクロ命令として文字列の入出力が定められている程度である。これらの命令や入出力のみでは学生を動機づけるような「面白い」課題を実現するのは困難である。

そこで COMET II エミュレータでは、COMET II のスーパーバイザコール命令 (SVC 命令) を用いて、COMET II の命令セットには提供されていない演算、ならびに入出力の機能を追加実装している。COMET II の SVC 命令

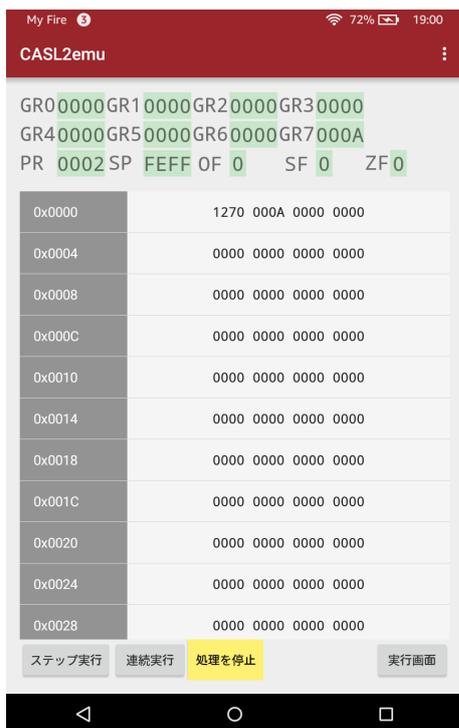


図 1 COMET II エミュレータのコンテキスト編集画面

は、オペランドに指定された実効アドレスをサービス番号と見做し、(仕様には与えられていないが) システムで規定されるサービスを呼び出す命令である。以下は SVC 命令によって学生の作成する機械語プログラムから利用できる COMET II エミュレータの機能である。

- **乗算と除算:** COMET II には実装されていない乗算と除算を行う。
- **文字列入出力:** COMET II の仕様に記載されている半角英数字と記号の入出力を行う。文字列入力は、SVC 実行時に出現する入力フォームに対して行う。文字列出力は、エミュレータが備える出力画面に対して行われる。
- **乱数生成:** 16 ビットの乱数を生成する。
- **楽音再生:** 1 オクターブ分の楽音を再生する。
- **図形描画:** 点、直線、長方形、円を出力画面に描画する。

例として SVC 命令による図形描画機能の呼出例を図 2 に示す。(例では COMET II のアセンブリ言語 CASL II で書かれているが、COMET II エミュレータは相当する機械語コードを入力する必要があることに注意されたい。) 図形描画機能を使用するときは、図形の種類を表すコード(上の例では円)、図形の描画情報(円の場合は中心座標、半径、色コード)をメモリ上に準備し、その先頭アドレスを汎用レジスタ GR7 に設定し、サービス番号として解釈される実効アドレス FF30H を指定して SVC 命令を実行する。他の機能も、このようにサービスに係るパラメータをメモリ上に配置してその先頭アドレスを指定したり、ある

いは所定のレジスタにパラメータを設定し、サービス番号として解釈される実効アドレスを指定して SVC 命令を呼ぶことで呼び出すようになっている。

LAD	GR7, DRWCMD
SVC	FF30H
...	
DRWCMD	DC 3 ; 描画する図形に円を指定
DC	100 ; 円の中心 X 座標
DC	100 ; 円の中心 Y 座標
DC	10 ; 円の半径
DC	1 ; 円の色 (赤色)

図 2 SVC 命令による図形描画

2.3 クラウド連携とラーニングアナリティクス

受講生に起きた事象に関する記録、収集、分析することで学習促進の方策を探る、ラーニングアナリティクスという研究分野が注目されている。その趨勢にあわせ、将来的には、COMET II エミュレータをはじめとする、各種電子教材をクラウドと連携させ、学生の成果物や学習履歴を蓄積し、それらを分析し、講義や演習の現場での実時間で教員ならびに学生のコンピュータ支援、講義や演習のプロダクトおよびプロセスの改善に活用することを構想している。その布石として、COMET II エミュレータには、現時点では学生の成果物のクラウドストレージへのアップロード機能のみが実現されている。また、別途、提出された成果物が課題で与えた仕様を満足しているかテストする採点あるいは採点支援ソフトウェアを Java で開発し、採点に要する教員の負担を大きく軽減している。

3. 演習での運用

筆者らは、2016 年度後学期に、筆者の所属する福岡大学工学部電子情報工学科において、2 年生を対象とする「電子情報工学実験」において、COMET II エミュレータを用いて機械語プログラミングの演習を実施した。この科目は、電子通信工学および情報工学の両方の基礎的な実技を学ぶ科目となっており、オシロスコープ (1 日)、ラジオ回路 (4 日)、論理回路 (4 日)、そして機械語プログラミング (4 日) に関する実習を実施している。COMET II エミュレータは機械語プログラミングの実習で用いている。1 日の実験時間は 180 分である。

学生は演習を受けるまでに、1 年次において論理回路、データ表現、データ構造とアルゴリズム、C 言語プログラミングについてすでに学んでいる。また、この演習と同時期にコンピュータアーキテクチャやコンパイラに関する講義が開講されており、本演習を通してこれら科目の多面的な理解が期待される。

演習は講義と実験を並行するかたちで行った。講義の最初に、ノイマン型コンピュータや 2 進数によるデータ表現

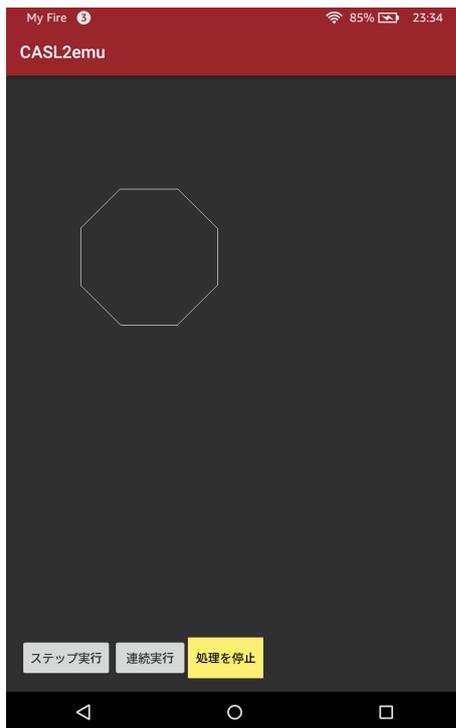


図 3 出力画面への図形描画

などの基礎事項を復習し、COMET II のアセンブリ言語である CASL II の命令を説明しては、簡単なプログラムを作成してその動作を確認する実験を行った。実験では、プログラミング用紙を学生に配布し、i) CASL II アセンブリ言語でプログラムを作成させ、ii) コード表を与えてハンドアセンブルさせ、iii) COMET II エミュレータのメモリをコンテキスト編集画面上で編集して機械語コードを入力させ、iv) 入力した機械語プログラムを実行させて動作を確認させた。アセンブラを使わず、ハンドアセンブルさせる理由は、コンピュータが直接理解できるのは2進数の機械語コードのみであること、アセンブリ言語は人間が機械語プログラムを容易に書くためのソフトウェア的手段であることを知らしめるためである。(コンピュータを観念的にしか理解していない学生は、しばしばコンピュータのプロセッサが高級言語やアセンブリ言語を直接理解し実行しているものと誤解している。)

4日の演習の期間中、6問の課題を与えた。このうちの3問は単純に転送命令、算術論理演算命令、スタック命令を理解するための小実験に過ぎないものである。残りの3問は、前述の SVC 命令による入出力機能を用いるもので、楽音再生機能を用いて小さな楽曲を演奏する課題、文字列入出力機能を用いて数当てゲームを作成する課題、図形描画機能を用いて正八角形を描く課題(図3)である。これらの課題では分岐命令を用いて条件分岐やループを実現することも要求される。ゲーム等を作らせる自由課題も計画していたが、演習時間が足りず実現には至らなかった。

今回は COMET II エミュレータを含む教材開発をする

のみで手一杯であり、残念ながら、教育効果を定量的に評価するには至らなかった。学生の反応は良好な印象である。命令の動作を確認する小実験の反応は薄いものだったが、楽曲演奏等の応用課題については成功したときの達成感があるようで、できたときには歓声をあげたり、他の学生と成果を見せ合うような反応が観察された。その一方で課題も残った。短期間で課題をやり遂げる学生が1割ほどいった一方で、全ての課題をこなせていない学生が2割ほどおり、それらの学生を演習の間に見つけ出して必要な支援することはできなかった。演習の間にオンラインで教員ならびに学生を支援する仕組みを早期に実現したい。また、プログラミング演習でありがちな「コピペ問題」はこの演習でも起こり、課題の答えを写真に撮って仲間内で LINE で共有する不正が見られ、演習時間中にスマートフォンに触れることを禁じる措置が必要となった。演習課題にバリエーションを導入する仕組み、バリエーションに対応した自動採点の仕組みを構築する必要性を感じられた。

4. おわりに

本稿では、学生に機械語プログラミングを実体験させ、コンピュータの動作原理、ならびに座学で学んだハードウェアならびにソフトウェアの知識をつなぎ、立体的に理解させることを目的として、筆者らが開発した COMET II エミュレータとそれを活用した演習の紹介を行った。教育効果の定量的な評価には至っていないが、学生の反応は良好であり、特に楽音再生や図形描画を伴う応用課題については達成したときにはポジティブな反応が観察された。また、採点自動化による演習における教員の負担軽減についても一定の成果が得られた。一方、演習についてこれしていない学生を見つけ、演習の現場において教員と学生を支援する仕組みについてその必要性が認められた。

今後は提出された成果物や演習中の学生の学習履歴をもとに分析を行い、教員や学生を演習の現場でオンラインで支援する仕組み、さらには講義内容や教材の改善を目指すラーニングアナリティクスを実施していくことが挙げられる。また、学生間での協業を促し、アクティブラーニングを実現する仕組みも構築していきたい。

参考文献

- [1] 情報処理推進機構：試験で使用する情報技術に関するプログラミング言語など、入手先 (https://www.jitec.ipa.go.jp/1_13download/shiken_yougo_ver2_0.pdf) (参照 2017-06-01)。