効果的な可視化手法を用いたアセンブリ言語教育支援システムの構築に関する研究

効果的な可視化手法を用いた アセンブリ言語教育支援システムの構築に関する研究

Study on educational system for assembly language
by using effective visualization method

小林晴紀
Haruki Kobayashi
指導教員　和田幸一

法政大学大学院理工学研究科応用情報工学専攻修士課程

In this paper, I propose an educational system for assembly language and computer architecture by using effective visualization method. The System allows students to study assembly program technique and internal structure and behavior of computer visually. Current educational system has a problem that visualization of control structures and data structures. The system supports it and necessary function such as editor, debugger, tracer, and profiler which reference current educational system.
***Key Words** : Assembly language, Computer Architecture, Educational System, CASL Ⅱ*

１．はじめに

情報系の学科において，アセンブリ言語教育はカリキュラムの柱の一つであり，多くの教科書が存在するが，一般的に教育は難しいとされている．抽象度の低い言語であるため，コード量が増えるにつれ，プログラムの振る舞いを把握することが難しくなることや，アセンブリ言語と関連性の高い分野である計算機アーキテクチャやOS，アセンブラの教育が難しいことが主な原因と言われている．特に計算機アーキテクチャ教育に関していえば，計算機の高機能化やブラックボックス化が進み．学習者が計算機内部の構造や動作の仕組みを理解することが難しくなっている現状がある．

実際の教育現場では計算機アーキテクチャを学習させた後，アセンブリ言語を学習させるケースが数多く存在する．つまり，計算機アーキテクチャとアセンブリ言語の双方の関連性が理解でき，それらの教育を支援できるシステムが求められている．

提案システムは，アセンブリ言語と計算機アーキテクチャの教育者と受講者を対象にアセンブリ言語プログラミング技法や計算機の基本構成要素であるプロセッサ，メモリ，入出力装置の機能，動作，及び要素間の相互関係の理解の支援を目的とする．講義の受講者は初学者が大半を占める．そのため，計算機は基本となるアーキテクチャを学べるものが望ましい．よって，COMETⅡCASLⅡシステムを採用する[1]．COMETⅡCASLⅡシステムは情報処理技術者試験のために仕様を策定されたものである．

ここでは，先行調査，システムの設計，実装，今後の展望について述べる．

２．先行調査

システムを開発するにあたって，まず必要な機能を明らかにする．そのために，既存の COMETⅡCASLⅡシステムと先行する教育支援システムを調査・比較した．調査したシステムの一覧表を次頁に示す．

主に 2 点に着目する．一つ目は計算機内部の構造や動作を可視化しながらシミュレーションする機能である．これを以下，可視化シミュレーション機能と呼ぶ．可視化シミュレーション機能は数々の研究によって学習効果が証明されている．[2][3][4]

本誌では，可視化シミュレーション機能をアセンブリ言語レベルと論理ブロックレベルに分類して，可視化方法や抽象度に関して調査する．指標に関しては付録[1]を参照．抽象度は計算機の動作レベルとプログラマが定義したデータの抽象度に分けて考える．

二つ目は，他言語の教育システムが中核機能としている機能である．先行調査では，教育支援システムを構築する上でも中核機能として期待できる機能について調査する．ここでは，エディタ，デバッガ，プロファイラ，トレーサの 4 項目に関して調査・比較をする.調査内容と結果に関しては[5]にまとめた.

表 1. 調査対象システム一覧

1	CaslBuilder
2	WCASL-Ⅱ
3	Casl2 Visible Inside
4	CASL2 シミュレータ
5	ドリームキャスル
6	中身が見える Caslxxxx
7	CASL2 cmd
8	CASL2 する?
9	キャッスルシミュレータ
10	CASL & COMET Simulator
11	CCaslⅡ
12	CASL2000 for Windows95,98/NT 4.0
13	EduCasl
14	Hello CASLⅡ
15	IPA 公式シミュレータ
16	Web 版キャッスルⅡシミュレータ
17	VisuSim
18	ECAS
19	MCSim
20	コンピュータの仕組みを直感的に理解できるプログラミング教材
21	内部動作を視覚化した教育用 MIPS プロセッサシミュレータシステム
22	eAVR
23	命令セット定義可能な汎用アセンブラ

既存システムは,計算機レベルでは充分な機能が確認できた．しかし，アセンブリ言語レベルにおいては制御構造の可視化やプログラマが定義したデータ構造の可視化機能が不充分だった．また，COMETⅡCASLⅡシステムのエディタ，デバッガ，トレーサ，プロファイラはどれも他のアセンブリ言語シミュレータの機能より劣っていた．そして既存システムの中には COMETⅡCASLⅡの仕様を忠実に再現できていないシミュレータが多く存在した．

既存の COMETⅡCASLⅡシステムは，CASLⅡプログラムがリロケータブルであることを理解させるような可視化が行われていないと判断した．リロケータブルなプログラムは，主記憶装置に配置するときにロードモジュールの相対アドレス部が実アドレスに変換される．これを理解させるには，配置位置に依存する部分と依存しない部分を効果的に見せることが重要となる．既存システムの中には実アドレスに変換された部分に対してラベル名を注釈したものが存在した．しかし，この可視化手法は，対応位置は掴めるが，依存する部分の値が配置位置によって変わることを理解するには不充分だと考える．

また，教育の観点から見ると，COMETⅡCASLⅡの言語仕様には問題点が 2 つあると考える．一つはプログラムレジスタやスタックポインタをプログラマが操作できないことだ．これを用いたアドレッシングモードが学習できない．二つ目はマクロ命令を定義できないことだ．これではマクロ命令の本来の用途を学習できない．以上の不充分な面の教育を支援するためには，言語仕様を拡張する必要があると考える．

３． システムの設計

前章で挙げた教育支援システムは全て教育現場での実績があるシステムである．従って，先行する教育支援システムにおいて教育効果が実証されている機能は参考にしつつ，不充分な面の教育を支援できる機能を提供する．可視化シミュレーションにおいては,制御構造とデータ構造を効果的に可視化する．エディタ,デバッガ，トレーサ，プロファイラに関しては，他言語の開発環境や教育支援システムを参考にして必要な機能を提供する．アセンブリ言語と機械語の対応の掴みにくさに対しては,主記憶に配置する位置に依存しないプログラムを可視化することで対応する．CASLⅡを採用に伴う学習範囲の制限に対しては，後述する言語仕様の拡張を行うことで対応する．また，計算機アーキテクチャにおける学習を考慮した機械語プログラミング機能を提供する．

以上の設計方針に従って，システムを実現するために 3 つモードを用意する．一つ目はアセンブリ言語プログラミングモードである．アセンブリ言語プログラミングモードは CASLⅡプログラムを編集するモードである．拡張した言語仕様に基づき作成されたプログラムも同様に編集できる．そのために 2 種類のコンパイラを用意する．言語の仕様を忠実に実現したプログラムを対象としたものと，前述の拡張言語仕様に対応したものである． 二つ目は機械語プログラミングモードである．機械語プログラミングモードは COMETⅡに直接ビットを投入するモードである．三つめはデバッグモードである． デバッグモードはプログラムをシミュレーションするモードである．デバッグモードはアセンブリ言語レベルのシミュレーションを行う状態とマシンサイクルレベルのシミュレーションを行う状態を持つ．双方で可視化シミュレーション機能及びデバッガ，トレーサ，プロファイラの機能を提供する．

以下にシステムの構成を示す．

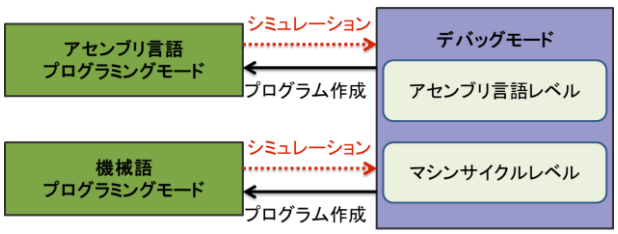

図 1. システムの構成

システムは移植性や開発期間,拡張性を考慮して，Java と javascipt で作成する．使用するテクノロジーを以下に示す.

表 2. 使用するテクノロジー一覧

テクノロジー	概要
Spring Boot	アプリケーションフレームワーク
JavaFX 8	Java ベースの UI プラットフォーム
ControlsFX	JavaFX の外部ライブラリ
JFoeniX	JavaFX の外部ライブラリ
Ace Editor	javascript 製のテキストエディタ
Nashorn	JavaVM 上で動作する javascript エンジン

４． システムの開発[6]

提案システムは Java と Javascript で記述され,Java(1.8)がインストール済みで JavaVM あるいは JRE が利用可能な Windows,Mac,Linux 上で動作するデスクトップ型アプリケーションである．以下，前章で述べた設計方針に基づいて開発した提案システムの特徴を示す.

（１）プログラミングアシスト機能

Ace Editor が提供可能なテキストエディタとしての基本機能に加えて，オートコンプリート，命令セットの情報表示，スペルチェッカー，クロスリファレンス，シンタックスハイライト機能を提供する．基本機能を除く各機能はユーザが設定を変更できる．変更を可能にしたことで，各々のユーザに特化したアシストが可能になっている．ここではその例として，オートコンプリート機能を挙げる．オートコンプリート機能は，候補となる命令及び例題コードを表示し,命令を学習者に選択させる．この機能は学習者のコード記述負担を軽減するというメリットがある反面，命令セットの学習面では悪影響となる機能である．そのため，この機能を使用しないようにすることが可能である(Ctrl + Space により切り替え可能)．シンタックスハイライトに関しては命令系，リテラル，コメント，レジスタ，特殊文字によって，ハイライト色を場合わけして施すことによりプログラム構造を理解しやすいものとしている．以上の機能はアセンブリ言語プログラミングモードで利用可能である．以下にアセンブリ言語プログラミングモードにおけるプログラミング編集画面を示す.

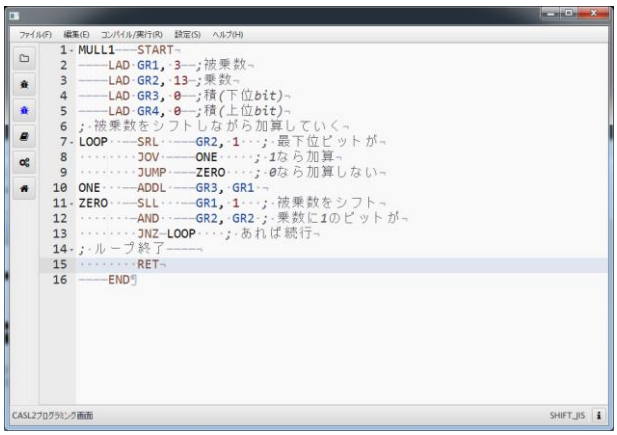

図 2. プログラム編集画面

（２）デバッグ機能

C 言語の開発環境 Visual Studio 2015 を参考にして必要な機能を提供する．尚，これらの機能は提案システムのデバッグモードで利用可能である．以下にデバッグモードの通常画面を示す.

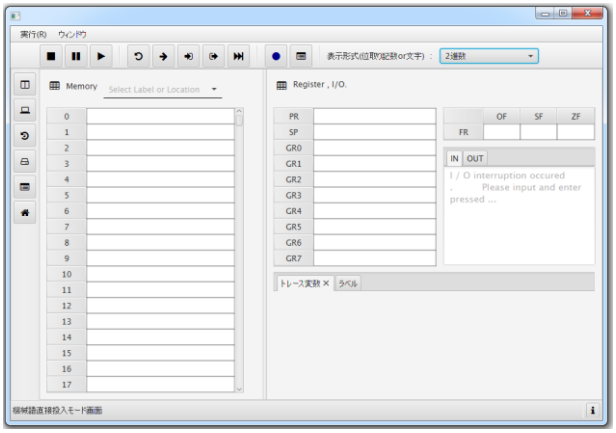

図 3 デバッグモードの通常画面

a）実行

プログラムを 1 ステップ単位で実行する．ユーザはサブルーチンやマクロ命令を一ステップで実行するか選択できる． 以下で述べるブレークポイントやレジスタ・メモリ内容変更機能と組み合わせることにより，ユーザ-システム間のインタラクティブなやり取りが可能である.

b）ブレークポイント

ユーザが任意の行で実行中のプログラムを一時停止できる機能である．指定した命令に差し掛かった際に停止する命令ブレークポイントに加え，ユーザが指定した条件を満たした際に動作を停止する，条件ブレークポイントを提供する．他に，メモリへの書き込みがあった際に停止するもの，分岐命令及びサブルーチンコールに差し掛かった際に停止するもの，変数の値が変更された際に停止するものを提供する.

c）変数のトレース機能

ユーザが指定した変数をトレースする．対応している変数は PR, FR, 汎用レジスタ, SP・主記憶装置・ユーザ定

義のデータ構造である．

d）実行プロファイル機能

メモリ使用量や命令の呼び出し回数を表示する

（３）言語仕様の拡張

プログラムレジスタ PR とスタックポインタ SP をオペランドとして使用できる．また，プログラマがマクロ命令を定義できる．以下にマクロ命令のシンタックスを拡張 BNF 記法で示す．

シンタックスに関しては ARM という既存のアセンブラを参考にした．

MACRO
[$*label*] *MINST* {$*param*{,$*param*}} (1)
MEND

$label はマクロが呼び出された時に指定されたラベルが代入される引数を指定する．MINST にはマクロの名前を指定する．予約語及びプログラムですでに使用されている名前は指定できない．$param はマクロ呼び出された時に値が代入される引数を指定する

この塊を１ブロックとする．START 命令の直前であれば 50 までのブロック数を定義できる．ユーザがプログラムを拡張コンパイルした時に検査結果を提示する．以下にプログラム例を示す．赤字がマクロ定義部分及び使用箇所である．

```
MACRO
$LABEL          MUL     $1,$2,$3
$LABEL.LOOP     SRL     $2,=1
                JOV     $LABEL.ONE
                JUMP    $LABEL.ZERO
$LABEL.ONE      ADDL    $3,$1
$LABEL.ZERO     SLL     $1,1
                AND     $2,$2
                JNZ     $LABEL.LOOP
MEND
EX3             START
                LAD     GR1,3
                LAD     GR2,4
                XOR     GR3,GR3
TEST            MUL     GR1,GR2,GR3
                RET
                END
```

図 4 マクロ定義及び使用例

主記憶の提示画面では，ユーザが定義したマクロ命令に対応する箇所に色付けする．マクロ命令はデバッグモードのステップ実行で１命令として抽象化して実行できる．

（４）可視化機能

ここではユーザ定義の変数ウォッチ機能と制御構造の可視化について述べる．

a）レジスタ，主記憶

汎用レジスタと主記憶及びスタックは 2 進数や 10 進数，16 進数の中からからユーザが選択した位取り記数法で提示する．表現形式に関しては，符号付表現，符号無表現，嵩上げ表現，浮動小数点表現，補数表現をサポートする．学習者の理解を助けるため，1 命令実行毎に次に参照された変数を黄色に，更新された変数を赤色に変化させ，ユーザの注意を喚起する．

図 4B のテーブルに機械語がロードされている主記憶の領域を提示する．二語で構成される命令語(第二語がアセンブラ命令(DC DS)やマクロ命令(IN，OUT，RPUSH，RPOP)，ラベル，リテラルに割り当てられた番地となる命令)に対しては，二語目の番地の内容を注釈する．スタック領域と定数領域，リテラル領域，プログラム領域は，ユーザがみて区別がつくように提示する． このように可能な限り内部のビットパターンについて説明を加えることで，アセンブリプログラムと機械語プログラムの対応を掴みやすくした．しかし，プログラマが独自のデータ構造をプログラム中で定義した場合，図 4B のような１語をテーブルの１行で表現形式ではデータ構造が把握しづらいという欠点がある．そのための解決策として後述する変数ウォッチ機能を提供する．この機能を使うことでユーザは可視化の形式を設定することが可能になる．

b）変数ウォッチ機能

ユーザが定義した変数の可視化に用いる．変数テーブル画面を以下に示す．変数テーブルは Excel ライクに編集が可能である．

c）プログラムの制御構造

アセンブリ言語プログラムレベルと機械語レベルの双方において，プログラムの呼び出し元と呼び出し先を画面上に提示して，制御の移行を可視化した．制御の移行を見せることで制御構造をわかりやすくしている．ここでは，呼び出し元と呼び出し先を OS，メインプログラム，サブルーチンという分類で分け，制御の移行を矢印で表現している．

d）主記憶に配置する位置に依存しないプログラム

主記憶に配置する位置に依存しないプログラムの可視化画面を以下に示す．この部分を可視化することで，CASL Ⅱ プログラムがリロケータブルであることの理解や，アセンブリ言語と機械語の対応を掴みやすくしている．ここでは，相対番地に相当する箇所を****(+相対値)で表現している．ここで提示される内容は，アセンブリプログラムと機械語の対応が掴みやすくするためにも，ラベル名やリテラル記述により自動生成された箇所に関する注釈を加える．

####	1030
####	####(DATA1)
####	1010
####	####(DATA2)
####	1043
####	####(DATA3)
####	8100
####(DATA1)	0001
####(DATA2)	0002
####(DATA3)	0003
####	0004

番地	主記憶
1000	1030
1001	1006(DATA1)
1002	1010
1003	1007(DATA2)
1004	1043
1005	1008(DATA3)
1005	8100
1006(DATA1)	0001
1007(DATA2)	0002
1008(DATA3)	0003
1009	0004

図 5.　主記憶に配置する位置に依存しないプログラムの可視化

e）仮想計算機

仮想計算機 COMETⅡの内部構造を可視化する．仕様に明記されていない箇所は独自に定義する．独自に定義したものは，アドレスバス，データバス，制御バス,内部バス，メモリアドレスレジスタ，メモリデータレジスタ，ALU，コントローラー，クロックである．計算機アーキテクチャレベルの可視化を行うことで計算機が機械語プログラムを制御する仕組みを効果的に理解させることが可能となる．

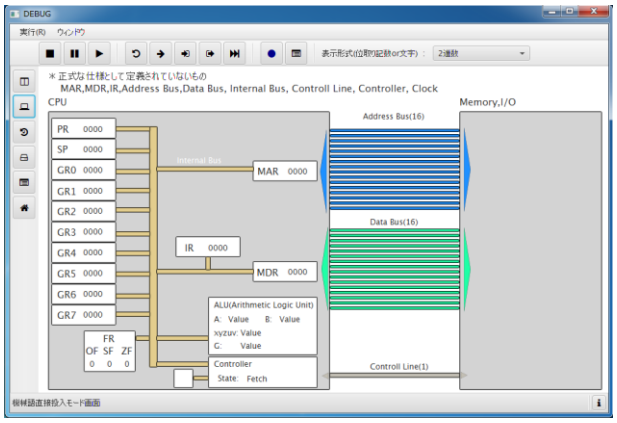

図 6.　計算機内部の可視化画面

５．現状

現時点で実装が完了した個所は，アセンブリプログラミングモード及び機械語プログラミングモードの全機能及び，デバッグモードのユーザインタフェース，実行機能，ブレークポイント，トレース機能と主記憶に配置する位置に依存しないプログラムの可視化である．

６．まとめ/今後の展望

本稿ではアセンブリ言語と計算機アーキテクチャの教育を支援するシステムについて述べた．提案シミュレータは学習者 - シミュレータ間のインタラクティブなやり取りを可能にし，かつ COMETⅡのレジスタやメモリを効率的に理解させる可視化や学習者のプログラミング能力を向上させる機能を実現している．今後の研究の課題は未実装の計算機レベルのシミュレーション機能を実装することである．

そして 提案システムを実際の教育現場で運用し評価を行う．また，関連研究[4]のように様々な論理ブロックレベルでのシミュレーション機能を組み込むことも課題として考えられる．

謝辞:

本研究を進めるに当たって，懇切なご指導，ご討論をいただき，本研究の意義，内容，今後の展開に関して，ご多忙の中，貴重なご教示をいただいた，法政大学理工学部応用情報工学科 和田幸一教授，大阪大学名誉教授 都倉信樹氏に深く感謝の意を表します．また，機能の検討に当たり，アドバイスをいただきました大阪工業大学 中西通雄教授に感謝の意を表します．

関連発表論文

1) 小林晴紀，上村宗嗣，和田幸一，“アセンブリ言語教育のための CASLⅡシミュレータの調査と比較”，第 9 回情報科学ワークショップ(WTCS)予稿集，pp31-34, (2013-09)
2) 小林晴紀，和田幸一，“アセンブリ言語教育支援システム SimAI の設計と実装”，第 10 回情報科学ワークショップ(WTCS)予稿集, pp351-358(2014-09)
3) 小林晴紀，和田幸一，“アセンブリ言語教育支援システム SimAI の設計と実装”，情報処理学会，研究報告コンピュータと教育（CE） 2015-CE-129(1), 1-7, (2015-03)

参考文献

1) 情報処理技術者試験出題範囲 pp.5-12(2011 - 07) http://www.jitec.ipa.go.jp/1_13download/shiken_yougo_ver2_0.pdf
2) 西牧悠二，北道敦司，宮崎敏明，“内部動作を視覚化した教育用 MIPS プロセッサシミュレータシステムの開発” 電子情報通信学会論文 Vol.J96-D No.10 (2013-10)
3) 今井慈郎，金子敬一，中川正樹，“計算機アーキテクチャ教育支援システムの開発と強調学習への適用” 電子情報通信学会論文誌 D Vol. J91- D No.2 pp.188-199(2008-02)
4) 堀匠吾，井本直希，那須聖史，中西通雄，“計算機シミュレータ ECAS の学習機能およびユーザ定義部品作成機能の実装”，JSiSE 学生研究発表会（2012-3）
5) 小林晴紀，上村宗嗣，和田幸一 “アセンブリ言語教育のための CASLⅡシミュレータ調査と比較” - シミュレータ比較詳細，法政大学理工学部応用情報工学科計算機科学研究室 テクニカルレポート TR15 - 9, 1 - 43, (2015 - 09) .
6) 小林晴紀，林毅，和田幸一，“アセンブリ教育支援システムにおける学習用 CASLⅡシミュレータの提案 - CASLⅡ&COMETⅡシミュレータ仕様書,” 法政大学理工学部応用情報工学科計算機科学研究室 テクニカルレポート TR14 - 6, 1 - 16, (2014 - 06)