

遠隔教育のためのプログラミング言語教育環境の構築

Development of programming language educational environment for distance education

那須 靖弘

Yasuhiro NASU

大阪滋慶学園

Osaka Jikei College

Email: y-nasu@wojc.jikei.com

あらまし：近年，MOOCs などインターネットを利用した遠隔教育が注目されている．遠隔教育では，学習者が使用するコンピュータ環境はさまざまであり，プログラミング教育のための，開発環境をどのように準備させるのが問題となる．

そこで，Web ブラウザ上で実行できるプログラミング開発環境を構築し，それを用いたプログラミング学習教材を開発した．本教材は，SCORM に準拠しており，プログラムの入力から実行までをブラウザ上で行うことができ，キー入力や画面表示を行う対話型プログラムの実行が可能となっている．

キーワード：プログラミング教育，C 言語，遠隔教育，SCORM

1. はじめに

近年，MOOCs 等のインターネットを利用したオンデマンド型の遠隔教育が注目されている．これらの講義動画の配信による授業は，いわゆる講義科目に適した方式であり，学生がさまざまな作業を行いながら理解を深める授業では，それぞれの授業科目の特性に応じて，教員が遠隔教育に適した教材の開発を行う必要がある．

プログラミング教育は，学習者が実際にプログラムを触ることによって学習意欲を高めることができるだけでなく，試行錯誤しながら，さまざまなエラーを修正しプログラムを動作させるという作業を通して理解を深めることができるものであり，学習者自身がプログラムを記述し，そのプログラムを動作させてみるのが重要である．しかし，遠隔教育においては，学習者の利用するコンピュータ環境はさまざまであり，インターネットカフェ等における学習など，開発環境の導入が難しいケースも想定される．このため，Web ブラウザ上で動作するプログラム開発環境があれば大変都合がよい．

現在，Web 上で各種のプログラミング言語を簡単に実行できるサイトは多数存在する⁽¹⁾．また，高等教育用にも同様の機能を持った学習システムが開発されている．これらのサイトは，サーバ側でプログラムを実行して結果を表示するもの⁽²⁾と，クライアント側でプログラムを実行している⁽³⁾もの，それらのハイブリッド⁽⁴⁾に分類できる．特定の言語のサポートだけを行うのであれば，クライアント側でプログラムを実行させる方式は，サーバを準備する必要が無く，悪意のあるプログラムの実行等によってシステム障害を引き起こす可能性がないなど，運営側から見てメリットが大きい．

そこで，クライアント側でプログラムを翻訳・実行する方式によるプログラミング教育システムを構

築した．本稿は，開発した遠隔教育用プログラム開発環境について述べるものである．

2. システムの概要

本システムが想定している対象学生はプログラミングの初学者であり，プログラミングを通してコンピュータの仕組みやアルゴリズムに関する理解を深めることを目的とする科目での利用を想定している．また，本システムは SCORM に準拠しており，moodle などの学習管理システム上に登録して利用するものである．教材は複数ページで構成され，課題の解説ページと実行環境を含むページが配置され，学習者はページをめくりながら学習を進める形態となっている．



図 1 演習画面

3. 演習実行環境

クライアント側で，学習者が作成したプログラムを実行するためには，Javascript 上で動作するプログラム実行環境が必要となる．そこで，Javascript 上で動作するプログラミング言語として，C 言語のサブ

セットとなるプログラミング言語（以下、TinyC と呼ぶ）を作成した。

TinyC とは C 言語の仕様から授業で扱うものだけを実装したもので、具体的には、goto 文、ポインタ、共用体、typedef 等は除外している。一方、string 型、ラベル付き break、デフォルト引数などの本来の C 言語にはない機能を一部追加した独自仕様となっている。

また、TinyC はバイトコードまでコンパイルし、バイトコードをインタプリタで実行する方式となっている。このバイトコードインタプリタは、実行速度を優先させ、Javascript の演算機能をそのまま利用している。Javascript では数値型はすべて実数型（IEEE 754 形式）であるため、TinyC では int 型の演算も内部的には実数型で行っている。

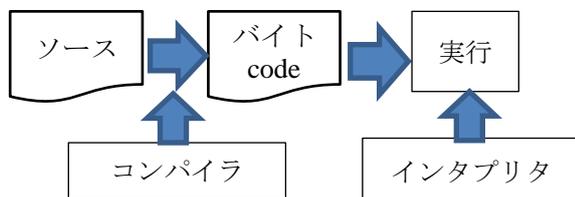


図2 プログラム実行の流れ

ブラウザ上で動作する Javascript はシングルスレッド動作が基本であり、Javascript 実行中は Web ブラウザの画面更新が行われない。このため、大きな数の素数判定を行うプログラムなど、実行に時間を要するプログラムを動かすと、ブラウザがフリーズしたような状態になる。一方、HTML5 から導入された Web worker を用いると、ブラウザの画面はプログラム実行時にも更新されるが、プログラム実行中にキー入力を受け付けることができない。画面表示やキーボードからの入力などのインタラクティブな機能は学習者がプログラムの動作を簡単に確認する手段であり、教育用のプログラム実行環境として必要な機能である。

本システムはプログラムをバイトコードに変換した後に、バイトコードインタプリタによってプログラムを実行しているが、バイトコードインタプリタは、任意の時点で実行中断ならびに再開が可能である。そこで、本システムでは、一定の間隔で実行を停止して、ブラウザが画面更新を行うことができるようにしており、対話型プログラムによる演習が可能となっている。

4. プログラムの保存

本システムではプログラム入力のためのエディタとして、Web ブラウザ上で動作するオープンソースのエディタである、CodeMirror を利用している。このエディタは非常に高機能であり、利用するプログラミング言語の構文あわせのハイライト表示等が可能であり、一般的なエディタと同様の使い方ができる。

通常、Web ブラウザ上で入力した情報はページを移動すると消えてしまう。これは、学習者が入力したプログラムも例外ではないが、本システムでは、エディタに入力されているソースプログラムを、ページを移動するタイミングで学習管理システム上に自動的に保存している。

このため、課題プログラムの作成途中で、以前に学習した内容の確認のためにページを移動することや、作業を中断して別のパソコンで続きの作業を行うことなどが可能となっている。

5. まとめ

プログラミング教育はプログラミングに関する理解、特定のプログラミング言語に関する学習、コンピュータシステムの理解、情報科学分野への導入などの情報学に関する目的だけでなく、問題解決能力の養成、モデル化や抽象化能力の獲得など幅広い目的での実施が可能であり、大学においても幅広い専門の学生が学ぶべき内容である。

情報を専門としない学生、あるいは初学者が学ぶことを考えた場合、コンピュータがより多くの手助けをする方が望ましい。その点、C 言語等の静的型付き言語は、人間に代わってコンパイラがプログラムのミスを検査する機能を有するため、プログラミングの学習に本質的には適している。

ただし、業務用のコンパイラは、プログラミングに習熟した者が使用するものであり、エラーメッセージも初学者にとってわかりやすいものではない。一方、独自コンパイラでは、エラーメッセージのカスタマイズや、エラーに対応する教材の該当ページへのリンクを張るということも可能である。

また、ブラウザ側でプログラムを実行する方式は、ブラウザと連携することが容易であり、学習者が興味を持つグラフィック等の機能が簡単に実現できる。

ところで、TinyC は実装されていない機能も少なくない。C 言語自体の学習が授業目的であるならば、さらに多くの機能を実装していくことが必要となるが、ある程度まで学習が進んだ時点で、サーバ側でプログラムを実行させる方式を併用するといったことも検討すべきであろう。

参考文献

- (1) アサインナビ: “ブラウザ上で簡単にプログラミングができる”, <http://room.assign-navi.jp/column/1519/>, (参照 2016.5.30)
- (2) 平田克己, AhmadZulkamal: “Web ブラウザ上で動作する C プログラミングシステムの開発”, 小山工業高等専門学校研究紀要, No.42, pp.123-128 (2010)
- (3) 那須野薫, 上野山勝也, 松尾豊: “次世代プログラミング学習サイト構築の試み”, 2013 年度人工知能学会全国大会 (第 27 回) 富山 (2013)
- (4) 宇野 健, 畝傍みなみ: “C 言語学習支援のための Web 上でのプログラミング環境の開発(2)”, 県立広島大学経営情報学部論集, 第 6 号, pp.35-41 (2014)