

## 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)

宇 野 健・畝 川 みなみ

### Development of the Programming Environment on Web for C Language Study Support (2)

Takeshi UNO and Minami UNEKAWA

#### 要 約

初学者のプログラミング教育において、演習の自学習は不可欠である。そのため、本研究グループでは、学習環境の構築が不要とする、WebベースのC言語学習支援システムの開発を行ってきた。しかし、これまでのシステムでは、ソースプログラムの実行は、サーバ上でコンパイルした実行ファイルを利用者がダウンロードして実行する形式であった。そのため、サーバOSに依存すること、実行手順が複雑になること、実行時エラーの取得ができないなどの問題があった。そこで今回は、演習の自学習の促進と学習者個別の進捗の把握を目的とし、Web上でC言語プログラムの疑似実行が可能なシステムを開発した。そして、開発したシステムを用いて実際の授業での運用実験を行い、利用状況とアンケート調査をから、システムの有効性を検証した。

#### 1. はじめに

プログラミングの学習では、学習者の自学習が重要である。特に、演習は学習内容の理解と、プログラミングスキルの向上には不可欠であり、授業内のみならず、授業時間外の自学習が理解とスキル獲得に大きな影響を与える。しかし、初学者にとってプログラミング学習環境を自力で構築することは困難であり、自学習の大きな障壁となっている[1]。

このような問題を解決するべく、本研究グループでは、WebベースのC言語学習支援システムを開発し、これを実際の授業において運用してきた[2]。このシステムの開発により、授業時間外、特に自宅等での利用率の向上が確認された。

しかしこのシステムでは、サーバのオペレーションシステム (OS) でコンパイルした実行ファイルを、学習者のPCでダウンロードする必要がある。そのため、学習者側のPCのOSがサーバと同じOSにしか対応しないこと、特定のブラウザ依存すること、学習者側の操作手順が増えるなどの問題点があった。また、個別の進捗把握や躰き箇所の把握が困難であるコンピュータ演習においては、学習者の最終的な成果物だけでなく、そこに至るまでのプロセスを把握することが重

要とされている[3]。しかし教員側での実行時エラーが取得できず、より詳細なプロセスや進捗の把握が困難であった。

そこで本研究では、Webブラウザ上でC言語の実行ファイルを疑似的に実行する、C言語ソースプログラムの疑似実行システムを開発し、C言語学習支援システムに実装する。これにより、完全にOSやブラウザに依存しないシステムを開発し、それを利用させることによって学習者の自学習の機会を増加させる。また、これまで不可能であった実行時エラーの取得等も行い、学習者の個人ごとの学習のプロセスおよび進捗の分析に活用する。最後にこれらのシステムの運用実験を、実際の授業において行い、ログデータの分析と学習者へのアンケート分析をもとに評価を行う。

## 2. C言語ソースプログラムの疑似実行システムの開発

### 2.1 開発背景

我々の研究室ではプログラミング教育における、以下のような問題点を認識し、これらをWebアプリケーションの開発、運用によって解決することを目的としている。

- (1) プログラミング学習機会を増やすための環境構築が困難であること
- (2) 学習者個人の理解度・進捗の把握が困難であること

(1)の問題を解決するために、C言語のソースプログラムのコンパイル、実行を行うシステム「Webコンパイルシステム」を開発し、C言語のプログラミング演習をWebブラウザだけで使用できるようにした (Fig.1)。これにより、インターネットに接続されたWindows PCがあれば、どこでもC言語プログラミングの演習が可能となり、自宅学習の利用率が大幅に向上した[2]。

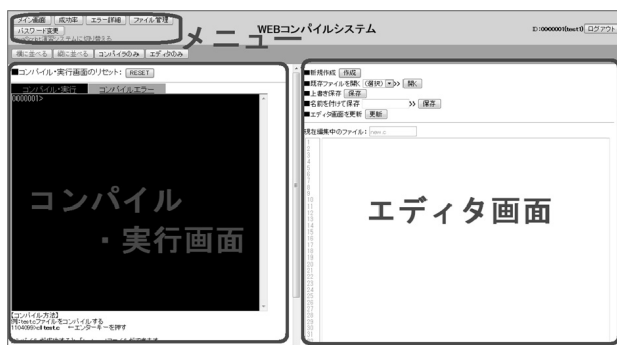


Fig.1 Webコンパイルシステムのインターフェース

(2)の問題に対しては、学習者の学習記録の蓄積が必要となる。まず、システムへのログインや、ソースプログラムの入力記録、コンパイル、実行等の詳細なログを学習記録としてデータベースに蓄積した。これらの膨大なデータをリアルタイムに分析し、学習者には個人ごとのエラー傾向を推定して通知するようにした。また、教員向けには、学習者個人の学習状況と、全体の学習状況やエラー傾向などを把握可能とし、Webブラウザ上で提示することを可能とした。

これらの機能を中心として、C言語学習支援システムを構築し、実際の授業において運用してきた。

## 2.2 プログラミング環境の問題点

既存のプログラミング演習システムでは、サーバ上にあるコンパイラ (gcc) でコンパイルし、生成した実行ファイルを、学習者側のPCにダウンロードし、実行させていた (Fig.2)。

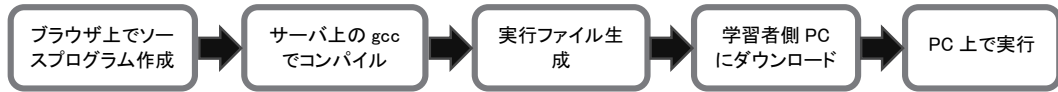


Fig.2 従来システムでのC言語プログラムの実行過程

この場合、scanfなどを用いた入力待ちを含むプログラムの場合、HTTP通信の特性から、ブラウザ上で実行することができなかつたためである。これにより、次のような問題が生じていた。

- (1) 特定のOSやブラウザでしか動作しない
- (2) 通常のコンパイラと比べて実行時の操作手順が増える
- (3) 入力ミスや実行時エラーのログの取得ができない

特に(1)の問題では、Windows PCにおいても、ブラウザごとに実行ファイルのダウンロード・実行のセキュリティの設定が異なるということがあった。様々な設定の変更が必要となる問題があった。このため、ブラウザも一部の製品、バージョンに限定して利用しなければならないという大きな問題があった。上記の問題点を解決するため、「C言語ソースプログラムの疑似実行システム (以下、疑似実行システム)」を開発した。

## 2.3 開発環境

疑似実行システムは、既存のC言語学習支援システムを構成する機能の一部として開発、実装した。開発環境、動作環境をTable1に示す。学習者側のインターフェースも既存システムのデザインを踏襲した。また、演習の処理はPHPとJavaScriptで行い、ブラウザ上のメッセージやデータ入出力において画面遷移を発生させないように、Ajax (Asynchronous JavaScript plus XML) による非同期通信を用いた。

Tabel.1 開発環境

・クライアント OS	: Windows7 Professional
・サーバ OS	: Windows Server2008
・Web サーバ	: Apache HTTP Server 2.2.15
・DBMS	: MySQL 5.1.41
・サーバサイド言語	: PHP 5.2.13
・クライアントサイド言語	: JavaScript

## 2.4 疑似実行システムの開発

### 2.4.1 システムの概要

本システムは、Windowsのコマンドプロンプトをモチーフとした、コンソールウィンドウ内でのCUI (Character User Interface) ベースのC言語プログラムの実行を、Web上で疑似的にエミュレートするものである。従来システムのように、サーバ上で生成した実行ファイルのダウンロード

が不要となるため、学習者側のOSやブラウザへの依存を解消することを可能とした (Fig.3)。これにより、フルブラウザが利用できるスマートフォンなどの携帯情報端末での利用も可能となり、移動中の電車内などでも利用可能とした。

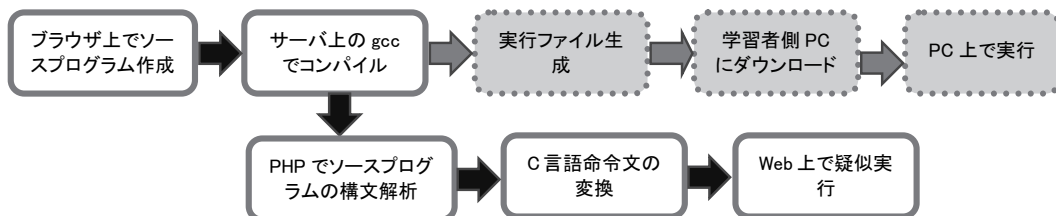


Fig.3 疑似実行システムでのC言語プログラムの実行過程 (実線部)

なお、C言語のすべての機能をエミュレートするのは困難であるため、このシステムでは、初学者向けの授業で利用する範囲のみを対象としている。具体的には、変数、各種演算子、条件分岐 (if, for), 繰り返し (while, switch), 配列 (2次元配列まで対応), 関数 (引数, 戻り値), ポインタ、構造体をサポートしている。

このシステムを開発するためには、以下の処理をサーバ側で行うことが必要である。

- (a) ソースプログラムの構文解析
- (b) C言語の命令文からPHPの命令文への変換
- (c) 文法エラーへの対応
- (d) 入力待ちを含むプログラムへの対応
- (e) 実行時エラー, 入力ミス情報の取得
- (f) コンソールのエミュレート

(a)については、PHPがC言語の文法をベースとしているため、文字列処理等のみで特殊な処理は不要であった。(c)については、C言語コンパイラであるgccで正常にコンパイルできたソースファイルのみを実行の対象としているため、文法エラーに対する処理は不要とした。また、(f)については、Ajaxを用いた非同期通信によって実現した。以下に、新たに開発した(b), (d), (e)について述べる。

## 2. 4. 2 C言語の命令文からPHPの命令文への変換と、入力待ちプログラムへの対応

### (1) 変数の表現

プログラムを疑似的に実行するためには、C言語で記述された命令や制御文の、PHPにおける変換方法を明確に定義しなければならない。入力待ちを含むプログラムの場合、サーバとの通信を断続的に行う必要があるため、スーパーグローバル変数であるセッション配列を用いてC言語の変数を表現した。

Tabel.2 C言語変数の表現の一例

C言語での表現	疑似実行システム (PHP) での表現
・変数型	<code>\$_SESSION['vartyp']['main']['x'] = 'int'</code>
・x(初期値 0)の データ型, 値	<code>\$_SESSION['varval']['main']['x'] = 0</code>
・アドレス	<code>\$_SESSION['varadd']['main']['x'] = 12345</code>
・ポインタ	<code>\$_SESSION['varpt'][12345] = "\$_SESSION['varval']['main']['x']"</code>

## (2) 標準ライブラリ関数への対応

C言語には多くの標準ライブラリ関数があり、これをPHPで動作する形式に変換する処理が必要である。対象のプログラミングの授業で扱うC言語の標準ライブラリ関数は、PHPにおいても同様の動作をするライブラリ関数が存在する。これを利用し、C言語で記述された命令文をPHPの形式に変換する処理を行うようにした。また、scanfなどの入力待ち関数の処理は、プログラムの実行状態をセッション配列に保持したまま非同期で断続的に通信する。これにより、疑似的にコンソール画面での入力待ち状態を表現することが可能となった。

### 2. 4. 3 実行時エラー・入力ミス情報の取得

本システムでは、プログラムの実行そのものをPHPで再現することにより、従来のシステムでは不可能であった入力ミスや実行時エラーの取得を可能とした。入力待ちを含むプログラムにおいては、入力値の型と個数がソース中のフォーマットと異なる場合、入力ミスと判定する (Fig.4 左)。直前の出力内容を再出力し、学習者に再入力を促すと同時に、データベースへの登録をおこなう。

また、疑似実行中に以下のエラーを検出した場合、実行時エラーとみなして疑似実行を強制的に終了し、エラーメッセージを出力するようにした (Fig.4右)。

- (a) 入力待ち関数 (scanf, gets) の記述ミス
- (b) ファイル操作関数 (fopen) でのファイル指定ミス
- (c) 無限ループ (ループ上限回数超過)

無限ループの判定に関しては、ループの上限回数を設定することで対応した。サーバへの負荷や、学習用であることなどから、一つのループにおける繰り返し回数の上限を2000回に設定した。

```

コンパイル・実行   コンパイルエラー
0000001>ex04_1.exe
a1とa2とa3の値を入力してください>5 4
##### 入力が不適です #####
【 再入力してください。 】
a1とa2とa3の値を入力してください>5 4 6
5+4-6の値は,3
0000001>

コンパイル・実行   コンパイルエラー
0000001>cl task13.c
command: gcc account%0000001%task13.c -o account%0000001
%task13.exe
task13.cファイルをコンパイルします
【コンパイル完了】 task13.exe
0000001>task13.exe
##### 実行時エラー #####
【 ! ファイルが存在しません! 】
{ fopen で指定されているファイルが存在しません }
{ ファイル名が間違っていないか、または、
{ ファイルが存在するか確認してください。 }
0000001>

```

Fig.4 値の入力ミス検出時の表示 (左) と、実行時エラー検出時の表示 (右)

### 2. 4. 4 JavaScript演習システムへの応用

本システムを応用し、Web言語のひとつであるJavaScriptの演習にも対応させた。システムの大部分はWEBコンパイルシステムと同一のものを流用し、JavaScriptに対応する部分を別途モジュールとして開発、実装した。インターフェースや操作性は、WEBコンパイルシステムとほぼ同様である。なお、JavaScriptはブラウザ上で動作するため、直接動作させることによってコンパイル作業や、実行にともなう構文解析等は不要であった。

### 3. 運用実験

#### 3.1 運用実験概要

開発した疑似実行システムを含む、C言語学習支援システムの運用実験を、実際のプログラミング授業において実施した。実験は、2012年度前期（5月～7月下旬の計12回の講義）、県立広島大学の経営情報学科2年生46名を対象に行った。講義中の演習や課題作成、試験勉強などに本システムを利用させ、運用期間後にアンケート調査を実施した。

#### 3.2 運用実験期間中のログ集計

期間中の運用実験でデータベースに蓄積されたログデータは、約33万件となった。システムへの総ログイン回数は1602回となり、そのうち学内からは1149回であった。内訳を見ると授業中が535回、授業外が614回と、学内においても授業外のログイン回数の方が多結果となった。また、学外からのログインは453回であり、学内の授業時間外のログイン回数と合わせて、自学習におけるログイン回数は計1228回に上った。

#### 3.3 アンケート結果

運用期間終了後、対象者全員に実施したアンケートの結果、演習室以外でも本システムを利用すると回答した学生は、昨年度の同じ質問と比較して大幅に増加した（Fig.5）。特に、自宅でのシステム利用については、昨年度が21人であったのに対し、今年度はその約2倍、ほぼ全員の41人となった。

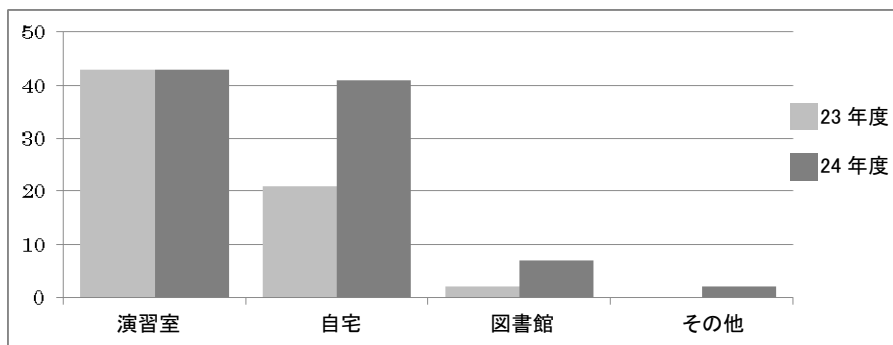


Fig.5 本システムの利用場所（複数回答可）

### 4. まとめ

本研究では、プログラミング学習における、自学習の促進と個別の進捗状況を把握するため、C言語プログラムの疑似実行システムの開発を行った。また、開発したシステムを用いて、実際のプログラミングの授業において運用実験を行った。

運用実験の結果から、利用頻度が昨年度の運用実験と比較して大幅に改善された。特に自宅での利用者は、昨年度の21名から41名（利用率95.3%）へと大幅に増加しており、本研究の目的の

一つである、自学習のさらなる促進が達成されたと考える。また、システムの操作性においても「使いやすい」と回答した学生が多く、高く評価された。

学生個別の操作分析を行ったところ、個人ごとに進捗の差が大きかったことが分かった。ほとんど差がないと考えていた授業時間中の進捗にも大きな違いが見られた。

今後の課題としては、個人ごとの進捗の差を埋めるような手法の開発があげられる。これに関しては、収集したログデータをもとに学習状況をフィードバックするシステムの改善を行い、結果だけでなく、演習の過程を評価するシステムの開発を行う予定である。また、授業の進行について来られなくなる、いわゆるドロップアウト兆候者を早期発見するための機能の開発・実装も今後の課題としたい。

### 参考文献

- [1] 倉澤邦美,鈴木恵介,飯島正也,横山節雄,宮寺廉造,プログラミング演習における一斉指導のための学習状況把握システムの開発,電子情報通信学会技術研究報告,ET,教育工学104, pp.19 - 24,2005
- [2] 宇野健,二階堂恵, 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, 県立広島大学経営情報学部論集, Vol.5, pp.77 - 84, 2013
- [3] 知見邦彦,樋山淳雄,宮寺庸造, 失敗知識を利用したプログラミング学習環境の構築, 電子情報通信学会論文誌,D-I情報・システム, No.1, pp. 66 - 75, 2005