

C 言語の初学者を対象としたプログラム実行状況の可視化ツールの評価

Evaluation of a Visualization Tool of C Program Execution for Beginners

水谷 泰治^{*1}, 中本 翔太^{*1(a)}, 安田 耕記^{*1(b)}, 西口 敏司^{*1}, 橋本 渉^{*1}

Yasuharu MIZUTANI^{*1}, Shota NAKAMOTO^{*1}, Koki YASUDA^{*1}, Satoshi NISHIGUCHI^{*1}, Wataru HASHIMOTO^{*1}

^{*1}大阪工業大学 情報科学部

^{*1}Faculty of Information Science and Technology, Osaka Institute of Technology

Email: yasuharu.mizutani@oit.ac.jp

あらまし： 本研究では著者らが開発している C プログラムの実行状況の可視化ツールを評価した. 本ツールは C プログラムをステップ実行しながら変数の内容と関数呼び出しの状況をブラウザ上で表示するツールであり, プログラミングの初学者に対して実行状況の理解を支援することを目的としている. 本発表では, 大学 1 年生を対象としたプログラミング演習の受講生に本ツールを使用した際のアンケート結果について報告する.

キーワード： プログラミング, C 言語, 可視化, 学習支援

1. はじめに

大学等のプログラミング演習において初めてプログラミングを学ぶ学生にとって繰り返し処理や関数定義を含むプログラムの流れの理解が難しいという事例が見受けられる. また, 演習においてよく用いられるプログラミング言語として C 言語があるが, C 言語は配列の添字が有効であることの検査を行わないため, 範囲外の添字の要素を参照するプログラムを記述しても初学者は誤りに気が付きにくい.

このような初学者を支援することを目的に, 著者らは C プログラムをステップ実行しながら, 配列の添字の検査, およびプログラム中で宣言された変数の内容や関数の呼び出し状況を図示するツールを開発した⁽¹⁾. 本稿では, このツールを大学 1 年生のプログラミング演習で使用し, 受講者に対して行ったアンケート調査の結果について報告する.

2. C プログラムの実行状況可視化ツール

本節では著者らが提案した C プログラムの実行状況可視化ツール(以降本ツール)について説明する.

2.1 全体像

本ツールは大学等におけるプログラミングの初学者を対象としており, Web ブラウザ上で動作する. ブラウザ上で動作する C 言語の実行環境⁽²⁾⁽³⁾および学習支援の研究⁽⁴⁾⁽⁵⁾は多く存在する. 本ツールは実装において, Web サーバへの依存を減らしたいことから, ブラウザ上のみで C プログラムを JavaScript に変換して実行するアプローチを採っている.

図 1 に本ツールの外観を示す. 図 1 の左から順に, C プログラムを記述するエディタ部, 実行時の標準出力, 可視化領域を表す. エディタ部の下部のテキストフィールドは scanf 関数に対する入力を表す.

エディタ部に C プログラムを記述した状態で左下の「コンパイル」を押したとき, 誤りが無ければ実行開始状態となる. 「コンパイル」を押したとき, ツール内部で C プログラムを JavaScript のコードに変換する. このとき, 可視化, ステップ実行, および

(a) 現在, (株)アクティス

(b) 現在, (株)トランスコスモス・テクノロジー

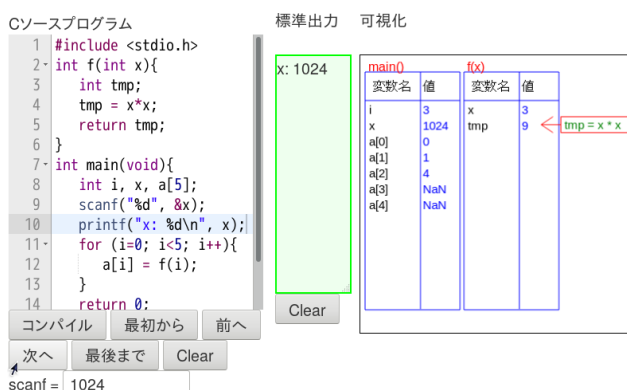


図 1. 本ツールの外観

```
function* f() {
  文1; yield;
  文2; yield;
  文3; }

```

(a) 定義

```
var a = f();
a.next(); //文1を実行して中断
a.next(); //文2を実行して中断
a.next(); //文3を実行して終了

```

(b) 使用

図 2. JavaScript におけるジェネレータ関数

配列の添字検査を行うコードを JavaScript のコードに挿入する. このように生成した JavaScript のコードをブラウザ上で実行することで, C プログラムの実行状況の表示とステップ実行を実現している. コンパイル後, 「次へ」を押す度にプログラムを 1 文ずつ進め, 「前へ」を押す度に 1 文実行前の状態に戻る.

2.2 可視化

本ツールで可視化する情報は, 関数呼び出しの状況と変数の内容である. 図 1 の実行状況は, 図 1 の左部に記述した C プログラムの for 文において, i が 3 のときに関数 f を呼び出している状況である.

2.3 ステップ実行

文献(1)では変換した JavaScript のコードを文単位に分割して 1 つずつ実行することでステップ実行を実現していた. この手法は実行時のオーバーヘッドが大きいため, 今回用いたツールではステップ実行の実装を変更した. 具体的には, JavaScript のジェネレータ関数という機能を用いてステップ実行を実現した. 図 2 にジェネレータ関数の定義と使用方法を示

す。図 2(a)のように `function*` で定義した関数はジェネレータ関数となる。ジェネレータ関数は図 2(b)のように関数呼び出しによってインスタンスを返却し、そのインスタンスの `next` 関数を呼ぶことで処理を進める。`next` 関数を呼び出すと関数の実行を開始し、`yield` の部分で関数の処理を中断して呼び出し元に戻る。その後、再度 `next` 関数を呼ぶことで中断した場所から処理を再開し、次の `yield` まで処理を進める。

本ツールでは「次へ」を押したときに `next` 関数を呼び出すことでステップ実行を実現している。また、プログラムの実行開始から `next` 関数を呼び出した回数 `n` を計測しておき、「前へ」を押したときにプログラムを最初から実行し直し、`n-1` 回 `next` 関数を呼び出すことで 1 つ前の状態に戻る機能を実現している。

2.4 制限

本ツールは C から JavaScript への変換を容易にするために、大学等における最初のプログラミング演習で扱うプログラムを表現できる程度に文法を簡略化している。具体的には、ポインタ、`break` 文、`continue` 文の除外、および宣言の簡略化等を行っている。また、大域変数の宣言、配列の添字に関数呼び出しを含む配列参照は、プログラムの記述と実行は可能であるが可視化の対象外としている。

3. アンケート

大阪工業大学情報科学部の 1 年生を対象としたプログラミング演習の受講生に本ツールを使用してもらい、アンケートを実施した。演習の第 9 週にツールを公開し、その 4 週間後にアンケートを実施した。

表 1 に本ツールの使用状況のアンケート結果を示す。受講者 503 名のうち 139 名から回答が得られた。少しでも使用した受講生は回答者の約 47% (65 名) に留まった。ツールの周知方法が口頭および web によるツールの存在のみであり使用を強いることはしなかったため、ツールに関心を示さなかったり、存在に気付かない受講生が多数いたと考えられる。

表 2 にツールの使用者 65 名に対する質問とその回答を示す。質問 2 では、「とても・少し役に立った」の合計が 54.7% であり、演習の課題を解く上で効果があったと考えられる。役に立つと回答した理由として、繰り返し処理を視覚的に追っていくことができる、2 重 `for` 文の誤りを発見するのが容易になったといったものが挙げられており、可視化とステップ実行による補助の効果があったと考えられる。一方、役に立たなかった理由としては、使い方がわからなかった、可視化領域が狭いため要素数の大きい配列が表示されないといったものがあつた。

質問 3 は「とても・少し思う」の合計が 46.9% となった。これらの理由として、プログラムがどのように実行されているかを知ることができた等があげられた。一方、「あまり・全く思わない」の理由として、正しいプログラムを書いてもエラーが出る、可視化されてもプログラム中のどの部分を表しているかわかりにくい等があげられた。

表 1. 本ツールの使用状況

| 質問1. 本ツールを合計どれぐらい使用しましたか? [単位:人] | | | | | |
|----------------------------------|-------|---------|--------|-------|-----|
| 2時間以上 | 1-2時間 | 30分-1時間 | 10-30分 | 10分未満 | 未使用 |
| 6 | 6 | 11 | 19 | 23 | 74 |

表 2. 本ツールの使用者に対するアンケート結果

| 質問2. 本ツールは演習の課題を解く上で役に立ちましたか? | | | | |
|-------------------------------|---------|-----------|-------------|------------|
| とても役に立った | 少し役に立った | どちらとも言えない | あまり役に立たなかった | 全く役に立たなかった |
| 21.9% | 32.8% | 34.4% | 6.3% | 4.7% |

質問3. 本ツールを用いることでC言語の配列や関数の理解度が向上したと思いますか?

| とても思う | 少し思う | どちらとも言えない | あまり思わない | 全く思わない |
|-------|-------|-----------|---------|--------|
| 17.2% | 29.7% | 39.1% | 10.9% | 3.1% |

質問4. 本ツールを用いることでプログラムのデバッグ作業が容易になったと思いますか?

| とても思う | 少し思う | どちらとも言えない | あまり思わない | 全く思わない |
|-------|-------|-----------|---------|--------|
| 15.6% | 32.8% | 35.9% | 12.5% | 3.2% |

質問 4 については、「とても・少し思う」の合計が 48.4% であった。回答の理由として、配列の範囲外を参照したときのエラーに気づきやすかった、この関数が間違っているのか調べるのに役立った等があげられた。特に、前者の配列要素の参照についての回答理由は多かった。

以上より、ツールの利用者数は少なかったものの、質問 2 から 4 のいずれにおいても「全く・あまり役に立たなかった・思わない」よりも「とても・少し役に立った・思う」が上回る結果となり、本ツールはプログラミング初学者を支援するツールとして一定の効果があると考えられる。

4. まとめ

本研究では、初学者を対象としたプログラミング演習において C プログラムの実行状況を可視化するツールのアンケート調査を行った。今後の課題として、今回の調査で指摘された使いにくい点の改善を行った上で、より多くの受講生からのアンケート調査を行い、詳細な分析を行うことがあげられる。また、ポインタや構造体にも対応し、プログラミング演習への適用範囲を拡大することがあげられる。

謝辞: 本研究は JSPS 科研費 JP15K21511 の助成を受けたものである。

参考文献

- (1) 吉本拓斗, 三村明, 辻賢紀, 古本顕吾, 水谷泰治. "C 言語の初学者を対象としたプログラム実行状況の理解支援ツール", 電子情報通信学会 2016 総合大会講演論文集, D-15-38, (2016).
- (2) "codepad", <http://codepad.org/>
- (3) "ideone", <https://ideone.com/>
- (4) 長江明彦, 香川考司. "JavaScript 上で動作する C 言語用視覚的デバッグの開発", 教育システム情報学会第 37 回全国大会論文集, pp.394-395, (2012)
- (5) 畝川みなみ, 宇野健. "プログラミング演習支援システムの定量的学習データを用いた成績不振兆候者の早期発見に関する研究", 情報処理学会第 77 回全国大会, 4ZF-04, (2015)