

実行履歴情報に基づく c 言語理解支援ツール

佐藤 雄二[†], 大坪 聖[†], 紫合 治[†]

東京電機大学 情報環境学部[†]

1. はじめに

近年, 初学者に向けたプログラミング教育システムの開発や研究は様々なものが行われている. しかしそれらの研究の多くはプログラム実行時のプログラムの処理における最新の状態のみを利用者に示しているものが多く見られる.

そこで, 変数の変遷の様子なども動的に動いている様子を見ることができるとシステムを構築することで初学者がプログラミングを学ぶ際により理解を深めることができると考え, 本システムを提案する.

2. 関連研究

初学者用のプログラミング教育支援システムにおいてプログラム可視化を目的としたシステムとして, 初学者用教育のためのプログラム可視化用 c 仮想マシン[1]や, プログラミング学習のための可視化対話環境[2]などが提案されている. これらのシステムによって表示される情報は実行時における最新の情報が主である. これらは実行時における最新の状態についてはすぐに理解をすることができるがそれより過去の履歴情報をすぐには見ることができないために必要な手順が増えてしまうことが考えられる.

3. システムの概要

本システムは c 言語の初学者へ向けた教育支援システムとしている.

本システムでは visual studio 等のエディタ等とは異なりプログラムを入力してから実行, という流れではなく既に書かれているプログラムを読み込み実行, という形となる. また visual studio 等とは違い本システムでは多様な機能を持つわけではなく出来るだけ簡素なシステム構造とした. これは visual studio 等が多様な機能を持っていてそれらを扱うことができればいいが, プログラミング初学者にとってそれらの機能すべてを扱う事は困難である. よって本システムにおいては扱う機能を最小限にす

ることで初学者であっても扱いやすくなるようなシステムとした.

本システムでは構築したシステム用に設計したコンパイラを使用しプログラムの実行処理を行う. このコンパイラでは読み込んだプログラムの単語の変数情報や関数情報を内部に記憶しておきそれらの情報を GUI 側に送る役割を担っている.

4. システムの構成

本システムのプログラム実行例の画面は図 1 に示すように, (1)プログラム表示部, (2)実行履歴表示部, (3)変数履歴表示部, (4)コンソール部で構成している.

図 1 においてはプログラム中におけるある箇所にブレークポイントを設定しプログラムを実行した際の画面となっている.

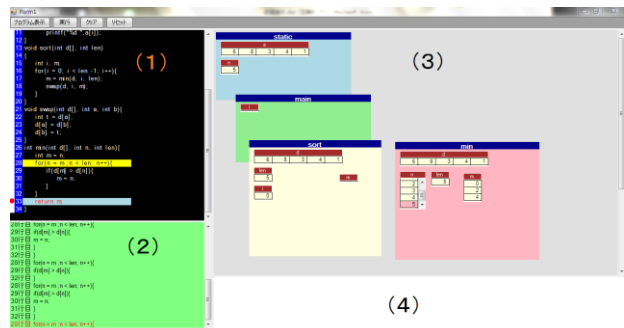


図 1 実行例による表示画面

また図 1 におけるそれぞれの機能は以下のようになっている.

(1) プログラム表示部

ここでは利用者が選択した任意の c ファイル及びテキストファイル等から実行したいプログラムを表示する. プログラム表示された後にプログラム表示部の白線部をクリックすることでブレークポイントの設定が可能としている. ブレークポイントが設定された行は色を変えることでブレークポイントの位置をわかりやすくなるような表示とした. そしてプログラム実行中には現在の実行行の位置も示す.

(2) 実行履歴表示部

実行履歴表示部ではプログラムが実行されて

いく際の経由した行と行番号を記録していく。図 2 に示すものがその際の表示である。図 2 は for 文のなかで a++を実行する際の実行行を表示したものである。その際実行の最新の行は色を変更することで実行行をわかりやすくした。

```

6行目 a = 0;
7行目 for(int i = 0; i < 5; i++)
8行目 {
9行目 a++;
10行目 }
7行目 for(int i = 0; i < 5; i++)
8行目 {
9行目 a++;
10行目 }
7行目 for(int i = 0; i < 5; i++)
8行目 {
9行目 a++;
10行目 }

```

図 2 実行行の表示

(3) 変数履歴表示部

プログラムの実行段階に応じて関数と関数内部の変数の変遷の状況が表示される。この際の変数の変遷は本システム用に構築したコンパイラからの情報に基づいたものから構成され、表示に反映されていく。

また実行時に最新の更新があった変数は他の部分と配色を変更することで複数の変数がある中でも最新の更新箇所をわかりやすくした。

図 3 にてその詳細を示す。図 3 はある配列の中身をソートするプログラムを実行した時の変数の変遷の様子でこの段階で swap の配列 d における変数 a での位置と変数 b の位置との入れ替えによって最小の値と先頭のものを入れ替えの一回目を終了した時点の履歴を表示しているものである。

パネル上部にある単語は関数名となっており static はグローバル関数を示し、どのプログラムを読み込んだ場合でも表示される。それ以外のパネルは読み込んだプログラム中に存在する関数によって異なる。

それぞれの関数のパネル内部にあるものは関数内部の変数となっており配列なら配列のサイズ分の幅を取得した表を生成し、int などの場合は単一の表を生成する。そしてプログラムの実行をしていく過程で変数の値が変化した場合は対応する変数の表が下方方向に追加されていく。

また、それぞれの変数の表は縦方向に伸縮することができるので、一つの表示で変数の変遷を最後まで見ることも可能である。また、現在

の表示してあるものよりも多くの変数履歴が表示される場合にはスクロールバーによってその履歴を確認することが可能になっている。

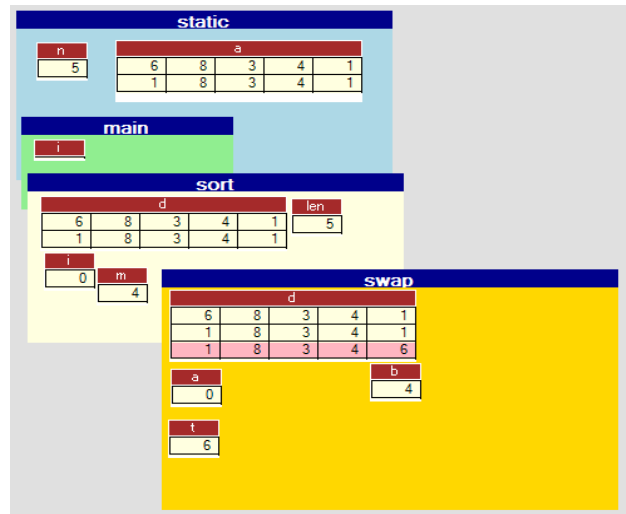


図 3 変数履歴表示部における実行例

(4) コンソール表示部

printf 文や scanf 文による入出力命令が実行された際に printf 文であれば出力を、scanf 文であれば入力した値や文字を出力し、それらを記録しておく。

5. おわりに

本システムは c 言語の教育支援システムとして履歴情報を基にしたプログラム実行状況を可視化するシステムを構築した。今後の課題としては

- (1) 対応できるプログラムの種類を増やす
 - (2) 実際の教育の現場でテストを行う
- などが挙げられる。

(1)においては本システムでは現在ポインタ等における処理については非対応となっているためであり、それらにも対応するようなシステムにする必要がある。

参考文献

[1] Aiguo HE: 初学者教育のためのプログラム可視化用 C 仮想マシン, 電子情報通信学会論文誌 D, Vol. J98-D, No. 10, pp. 1292-1300, 2015.
 [2] 川崎雄登, 平井佑樹, 金子敬一: プログラミング学習のための可視化対話環境, 情報教育シンポジウム 2014 論文集, 2014(2), 143-150