

# Emscripten と D3.js を用いた C 言語ポインタの Web ベース動的可視化システムの開発

## Development of a Dynamic Visualization System for C pointers that uses Emscripten and D3.js

西山 恭平<sup>\*1</sup>, 香川 考司<sup>\*2</sup>  
Kyohei NISHIYAMA<sup>\*1</sup>, Koji KAGAWA<sup>\*2</sup>

<sup>\*1</sup> 香川大学大学院工学研究科

<sup>\*1</sup>Graduate School of Engineering, Kagawa University

<sup>\*2</sup> 香川大学工学部

<sup>\*2</sup>Faculty of Engineering, Kagawa University

Email: s15g480@stu.kagawa-u.ac.jp

**あらまし** : C 言語学習者にとって、ポインタの概念はイメージしづらく、学習意欲を下げる原因となりうる。イメージを補助するためのツールや研究もあるが、それらは Web アプリケーションではなく導入作業などが必要となる。そこで、C 言語学習者が気軽に利用できる、新たなポインタのアニメーションによる可視化システムを Emscripten と D3.js を用いて実現する Web ベースシステムとして提案し、開発する。

**キーワード** : C 言語ポインタ, 可視化, Emscripten, D3.js, Web ベース

### 1. はじめに

C 言語プログラミングの学習において、ポインタは実際の動作をイメージすることが難しい構造である。特に、ポインタを用いたリストや木構造などのデータ構造はテキストや図だけでは理解が難しいことがある。ポインタの理解を支援する方法として、GDB や統合開発環境の Eclipse や Visual Studio のデバッグ機能を用いて変数のメモリ番地の割り当てを確認する方法が挙げられる。しかし、この方法では文字情報として得られるアドレスからつながりをイメージしなければならないため学習者の負担が大きい。また DDD などの GUI デバッガでポインタのつながりを可視化する方法もあるが、ブレークポイントの設定や、ステップ実行の仕組みなどを学習する必要があり、慣れていないと使い辛い。

また、ポインタの理解を支援する既存研究としてメモリ空間を可視化し、ポインタのつながりも図示する SuZMe<sup>(1)</sup>などのデバッガも開発されている。これはメモリ空間とポインタ変数の対応付けがイメージとして理解しやすいデバッガであるが、Web アプリケーションではない。さらに多くの学習者に気軽に利用してもらうためには Web アプリケーションにする必要があると考えられる。そこで、本研究では Web アプリケーションによるポインタ可視化システムの開発を行う。Web アプリケーションとして開発する上で、Emscripten<sup>(2)</sup>を基盤とし、D3.js<sup>(3)</sup>によるアニメーション表現により、学習者のポインタの動作イメージの補助と理解を図る。

### 2. システムの概要

本研究において開発するシステムは、ポインタを学び始めた段階の C 言語学習者を対象とする。また、学習者の導入と操作方法の学習の負担を軽減するた

め、ブラウザにコードを入力するだけで使える Web ベースシステムとする。ブラウザ上で入力されたコードをコンパイルし、結果を表示するために C 言語から JavaScript への変換を行う Emscripten を用い、Emscripten が生成する asm.js<sup>(4)</sup>形式の JavaScript を解析することで得られたデータを用い、データ駆動型のグラフィックライブラリである D3.js によるアニメーションを表示する。

### 3. 関連技術

#### 3.1 Emscripten

Emscripten は C/C++ で記述されたソースコードを中間コードに変換し、それを JavaScript に変換する。実際に C のコンパイラを用いて中間コードを生成するため、コードの解釈を間違えることがない。また、Emscripten で出力された JavaScript は asm.js という形式でありアセンブリに近い記述である。つまり、アドレスの参照先への書き込みや読み出しを直接記述しているため、それを読み取ることでポインタの動作を抽出することができる。

#### 3.2 D3.js

D3.js は JavaScript のデータ駆動型グラフィックライブラリである。配列や JSON といったデータに基づいて SVG 形式の画像を出力することが可能である。本研究では抽出したポインタの動作データを参照し、ブラウザ上に SVG 形式の画像を出力してアニメーションを行う。D3.js はデータからグラフを生成する仕組みや、グラフにアニメーション効果を付与する仕組みが存在するため、データからポインタの動作イメージをアニメーション化するのに適している。また、グラフやアニメーションの詳細は自由に記述することもできるため表現の自由度も高い。

#### 4. システムの構成

本システムは大きく二つの部分に分けられる．一つはブラウザからデータを受け取り，Emscripten の生成した JavaScript を解析してアニメーション用のデータを作成する Java Servlet によるサーバサイドのバックエンドである．もう一つはブラウザ上でアニメーションを表示する HTML5 と JavaScript (D3.js) によるフロントエンドである．このシステム構成を図 1 に示す．

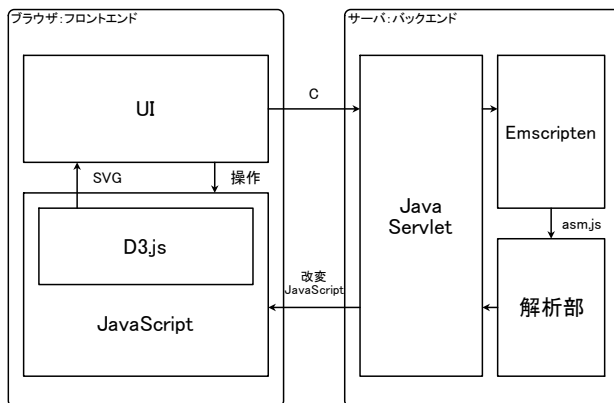


図 1 システムの構成図

本システムは学習者つまりユーザからみるとただの Web ページであり，C 言語のソースコードをブラウザのテキストエリアに記述するか，ファイルアップロードフォームより送信するだけで利用可能である．

ファイルを送信すると図 2 のようなアニメーション表示画面が表示される．アニメーションは画面上の再生ボタンを押すと開始され，プログラムの実行状態に合わせて一定間隔で進行する．アニメーションに使われているグラフは，メモリ上のアドレスの位置を意識するというよりは，直感的なポインタの接続イメージに近いものとなっている．

アニメーション制御の操作ボタンは動画プレイヤーを参考にした形状をしており，直感的な操作に近づける目的がある．アニメーション中は，画面上に表示された C 言語のソースコードの対応する行がハイライトされる．これにより学習者は今アニメーションで表示されているのがどのポインタ操作なのかを，ソースコードと対応付けて確認できる．また，アニメーションの間隔は学習者自身で調整することが可能である．繰り返しを用いたポインタ操作などを，間隔を短くして一気に飛ばしたり，間隔を長くして動作の詳細を把握したりといったことに使うことが想定される．表示されているノードはドラッグすることで固定することもできる．これで自動的なレイアウトが気に入らない学習者でも自分の好みのレイアウトを実現できる．また，この固定は，ドラッグするだけででき，ダブルクリックで自動レイアウトに戻るため，DDD などのレイアウト作業に比べると手間がかからない．

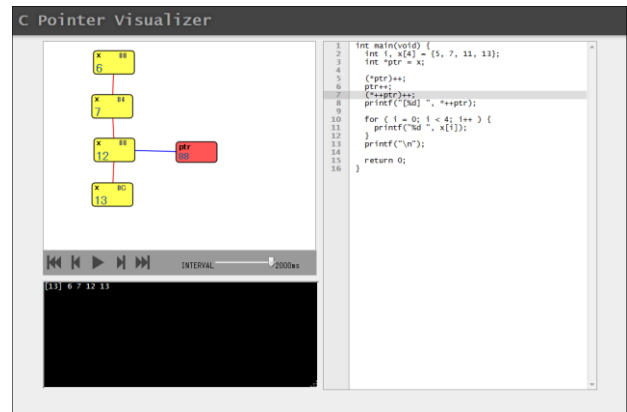


図 2 アニメーション表示画面

#### 5. まとめ・今後の課題

C 言語学習者を対象とした，Web ベースポインタ可視化システムの開発を行った．Emscripten と D3.js を用いることで，Web ベースでの可視化システムが実現でき，Web ベースのため導入などの手間がかからず簡単な操作で使うことができるようになった．また，ポインタの繋がりを強調した簡単なグラフによるアニメーションは，学習者のポインタのイメージを支援することにつながると考えられる．

また，本研究室に所属する 4 年次生の学生 2 名に本システムを試用してもらったところ，使いやすく，ポインタが直感的にイメージしやすいという意見が得られた．一方，アニメーションの動きに無駄がある所や，UI が直感に寄っているため言葉による説明を付加したほうが良いなどの改善意見も得られた．これらの意見に関しては今後対応していく必要がある．

本システムは，現時点ではローカルサーバ上でのみ動く試作段階であり，今後公開サーバ上に構築する必要がある．また，システムが利用可能になればプログラミングの講義などで実際にポインタを学び始めたばかりの学習者による評価を行う必要がある．アニメーションによる支援範囲も現時点では基礎的なポインタにのみの対応となっており，今後構造体や関数ポインタ，ポインタのポインタなどの複雑な構造にまで支援範囲を広げる必要がある．

#### 参考文献

- (1) 小池伸弥，郷健太郎，“プログラム実行時のメモリ空間可視化におけるインタラクションの提案”，情報処理学会インタラクション 2012, pp.665-670, 2012.
- (2) Alon Zakai, “Emscripten: An LLVM-to-JavaScript Compiler”, Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion, pp.301-312, 2011.
- (3) Mike Bostock, “D3.js - Data-Driven Documents”, <http://d3js.org/> (2015)
- (4) David Herman, Luke Wagner, Alon Zakai, “asm.js: Working Draft 17 March 2013”, Available online at <http://asmjs.org/spec/latest/>, Accessed June, 2013.