




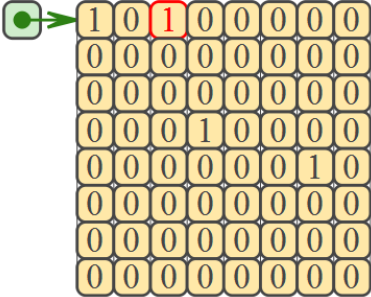

Визуализатор памяти отлаживаемых программ на языке C/C++

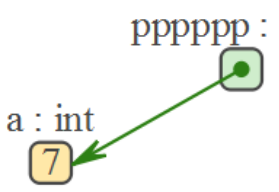
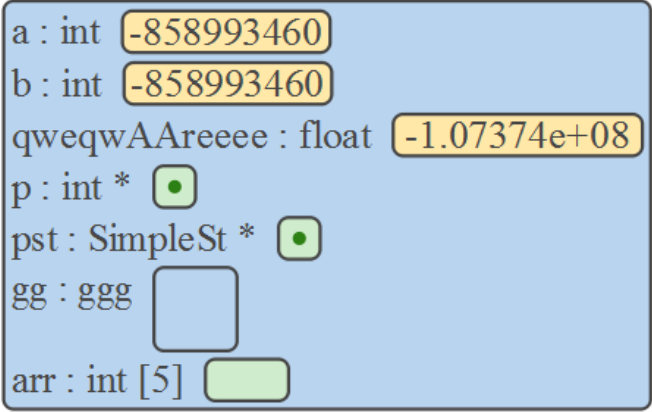

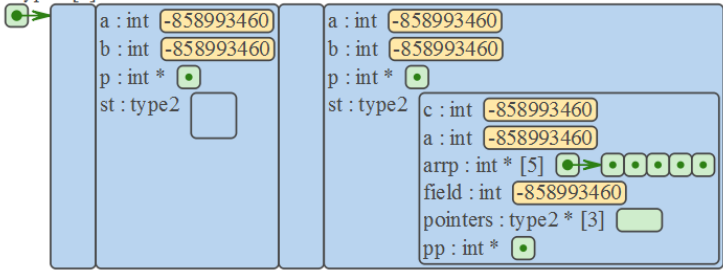
Общее.

Визуализатор предназначен для наглядного отображения структуры данных (локальных переменных текущей функции и выделенных блоков динамической памяти) отлаживаемой программы в виде графа, где дуги проведены от указателей к переменным. Он подключается к отладчику MS Visual Studio и отображает структуру данных программы на каждом шаге трассировки.

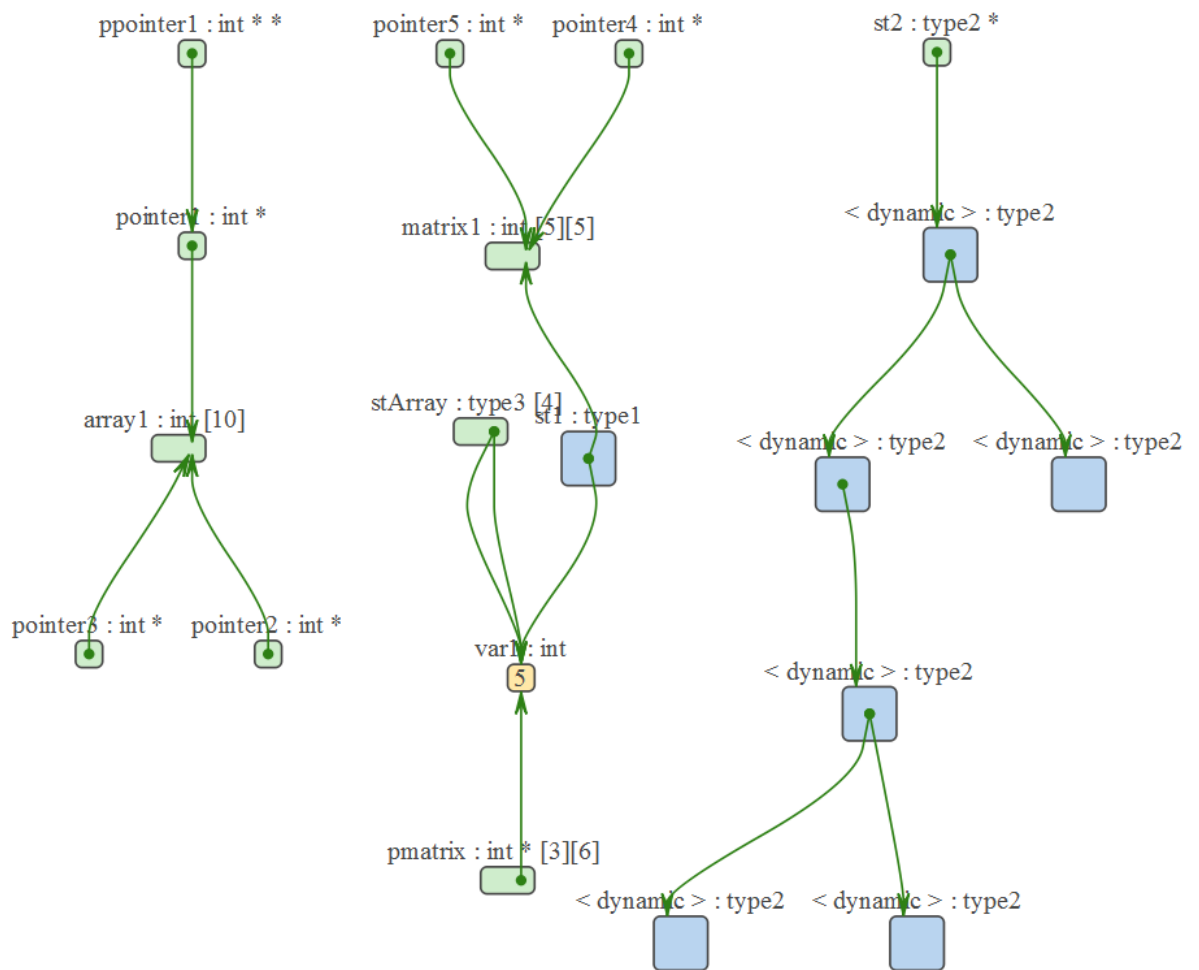
Граф, представляющий структуру данных.

Каждой переменной программы соответствует определённая вершина графа. Основные типы вершин:

| Описание | Графическое представление |
|--------------------------------------|--|
| Переменная простого типа | <pre>param : int</pre>  |
| Массив | <pre>arr : double [10]</pre>  |
| Свёрнутый массив | <pre>array1 : int [10]</pre>  |
| Матрица | <pre>field : int [8][8]</pre>  |
| Указатель (ни на что не указывающий) | <pre>array2 : int *</pre>  |

| | |
|---|--|
| <p>Указатель, указывающий на переменную</p> | <p>pppppp : int *</p>  <p>a : int 7</p> |
| <p>Структура (объект класса)</p> | <p>s : SimpleSt</p>  <p>a : int -858993460 b : int -858993460 qweqwAAreeee : float -1.07374e+08 p : int * pst : SimpleSt * gg : ggg arr : int [5]</p> |
| <p>Свёрнутая структура</p> | <p>s : SimpleSt</p>  |
| <p>Массив структур</p> | <p>stArray : type3 [4]</p>  <p>a : int -858993460 b : int -858993460 p : int * st : type2</p> <p>a : int -858993460 b : int -858993460 p : int * st : type2 c : int -858993460 a : int -858993460 arr : int * [5] field : int -858993460 pointers : type2 * [3] pp : int *</p> |

Граф укладывается поуровневому методу, что позволяет уменьшить количество пересечений дуг. Пример уложенного графа:



Основные действия в программе.

1. Подключение к отладчику.

Запустить отладку программы в MS Visual Studio, остановить её на точке останова.

В визуализаторе нажать кнопку «Connect». При необходимости, предварительно настроить, к какой версии MS Visual Studio подключается визуализатор.

Если подключение прошло успешно, отобразится граф текущей структуры данных программы.

При укладке графа после подключения сворачиваются все вершины, находящиеся глубже первого уровня вложенности (для более компактного изображения).

2. Трассировка программы.

Выполнение программы до следующей точки останова (или до конца) – кнопка «Go to next breakpoint».

Шаг с заходом в подпрограмму, с выходом и без захода в подпрограмму - кнопки «Step Into», «Step out», «Step over» соответственно. При переходе отлаживаемой программы из одной функции в другую – граф укладывается автоматически. В пределах одной функции в процессе трассировки новые дуги добавляются прямыми, новые вершины добавляются рядом с инцидентными им, а положение имеющихся вершин не меняется.

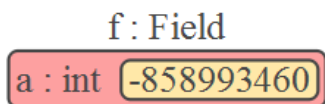
Эти действия дублируют аналогичные кнопки MSVS, однако при нажатии на них сразу происходит обновление изображения.

В случае, если после подключения продолжалась трассировка программы из MSVS, повторное нажатие кнопки «Connect» всегда актуализирует изображение и уложит граф автоматически.

Изменённые после очередного шага переменные подсвечиваются красным.

3. Автоматическая укладка графа - кнопка «Layout graph». В результате граф укладывается стандартным поуровневым методом. Свёрнутость/развёрнутость вершин при этом сохраняется.

4. Выделение объекта – объект выделяется левым кликом мыши, при этом он подсвечивается красным цветом. Выделить можно невложенную вершину или дугу. Возможно выделение группы вершин – при обведении их с зажатой правой кнопкой.



5. Перемещение объекта – достаточно зажать левой кнопкой прямоугольник и перетащить. Перетаскивать можно только невложенные вершины.

6. Сворачивание/разворачивание вершин. Чтобы свернуть/развернуть вершину, надо кликнуть по ней правой кнопкой мыши и выбрать соответствующий пункт в контекстном меню. При этом свернётся или развернётся только текущая вершина. Примеры свёрнутых вершин приведены ранее.

7. Работа с чёрным списком. Добавленная в чёрный список вершина не появляется при дальнейших шагах трассировки (в пределах текущей функции).

7.1. Чтобы добавить объект в чёрный список необходимо щёлкнуть по нему правой кнопкой и выбрать «add to black list». Добавлять в чёрный список можно только невложенные объекты.

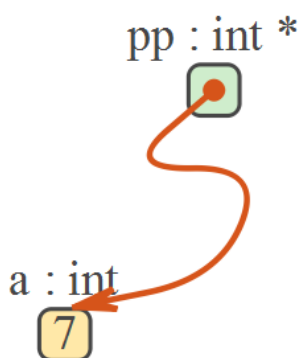
7.2. Для просмотра объектов, находящихся в чёрном списке необходимо нажать кнопку «Black list».

7.3. Для восстановления объекта из чёрного списка: в режиме просмотра списка нужно нажать кнопку «Recover», выбрав в списке нужный объект. В результате объект появится рядом с инцидентными вершинами, а если их нет – в левом верхнем углу экрана.

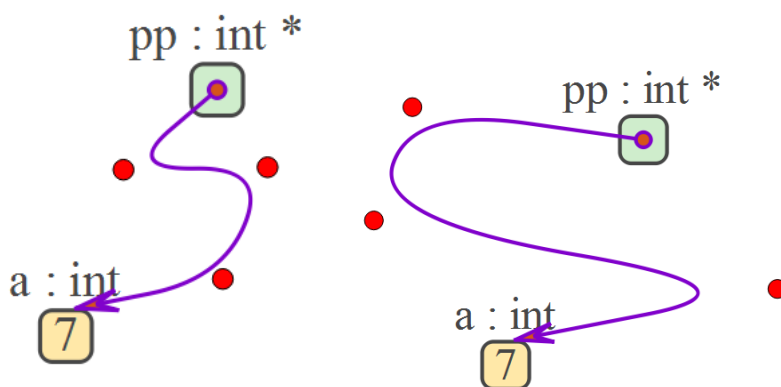
8. Действие с группой объектов. Все выделенные вершины можно свернуть/развернуть, или добавить в чёрный список, выбрав соответствующий пункт в контекстном меню по правому клику по одному из выделенных элементов.

9. Редактирование дуг. (принцип рисования, редактирования).

9.0 Для выделения дуги нужно щёлкнуть по ней левой кнопкой. При этом она подсветится красным цветом.



9.1 Для перехода в режим редактирования дуги необходимо сделать двойной щелчок по дуге. При этом станут видимыми её точки изгиба и дуга подсветится другим цветом.



9.2 Добавление точки изгиба: в режиме редактирования дуги необходимо сделать левый клик по некоторому участку дуги, зажав кнопку alt.

9.3 Удаление точки изгиба: в режиме редактирования дуги необходимо сделать левый клик по точке изгиба, зажав кнопку alt.

10. Сохранение изображения в файл – кнопка меню “Save current image”.

11. Просмотр динамически выделенных объектов.

Динамические данные отображаются только когда на них есть указатель в текущих переменных функции (или на них указывают другие динамические объекты, на которые есть указатель), либо указатель на них присутствовал в текущей функции, был перенаправлен в процессе трассировки в визуализаторе, и при этом не было переходов в другие функции.

Возможно отображение не только динамических объектов, но и массивов. Для работы этой функции необходимо в отлаживаемый проект подключить специальный заголовочный файл SeePPDynamicArrays.h, имеющийся в папке SeePP.

12. Масштабирование изображения. Производится колёсиком мыши.

Ограничения

1. Поддерживаемые типы данных:

- все скалярные типы;
- указатели;
- одномерные и двумерные массивы;
- структуры (классы);
- динамически выделенные объекты, если на них есть указатель.

2. Не поддерживаются:

- указатели void*;
- указатели в структуре на поля этой же структуры, указатель в структуре на себя;
- глобальные переменные (отображаются локальные переменные функции);
- матрицы в структурах;
- матрицы структур;
- типы union и enum;

Олифер Н.