Восстановление объектных структур данных при декомпиляции

Фокин А. П., студент гр. 527
Научный руководитель доц. к. ф.-м. н. Чернов А. В.

Декомпиляция

- Декомпиляция – восстановление программ на языке высокого уровня из программ на языке низкого уровня.
- Задача декомпиляции программ, написанных на языке Си, сравнительно хорошо изучена.
- При декомпиляции программ, написанных на языке Си++, необходимо также восстанавливать конструкции языка Си++.

Постановка задачи

- Разработать и реализовать метод восстановления объектных структур данных из низкоуровневого представления программ, написанных на языке Си++, для компиляторов GCC и MSVC для платформы x86.
- Метод должен восстанавливать:
  - Множество полиморфных классов программы;
  - Отношение наследования на этом множестве;
  - Соответствующие каждому классу виртуальные функции.
- Полученная иерархия классов должна быть эквивалентна иерархии классов исходной программы.

Информация о типах времени выполнения в Си++

- Если в программе на языке Си++ используются операторы **`dynamic_cast<>`** или **`typeid`**, то необходима компиляция с использованием *информации о типах времени выполнения* (*RTTI*).

- В случае, если в программе не используются операторы **`dynamic_cast<>`** или **`typeid`**, возможна компиляция без использования информации о типах времени выполнения.

Восстановление иерархий полиморфных классов с использованием информации о типах времени выполнения

- Задача поиска структур, содержащих информацию о типах времени выполнения, сводится к задаче поиска таблиц виртуальных функций.

Метод поиска таблиц виртуальных функций, основан на том, что:

- Каждая таблица виртуальных функций представляет собой массив из указателей на функции.
- Из сегмента кода существуют ссылки на первый элемент таблицы виртуальных функций.
- На остальные элементы таблицы виртуальных функций ссылок нет.

Восстановление иерархий полиморфных классов при отсутствии информации о типах времени выполнения

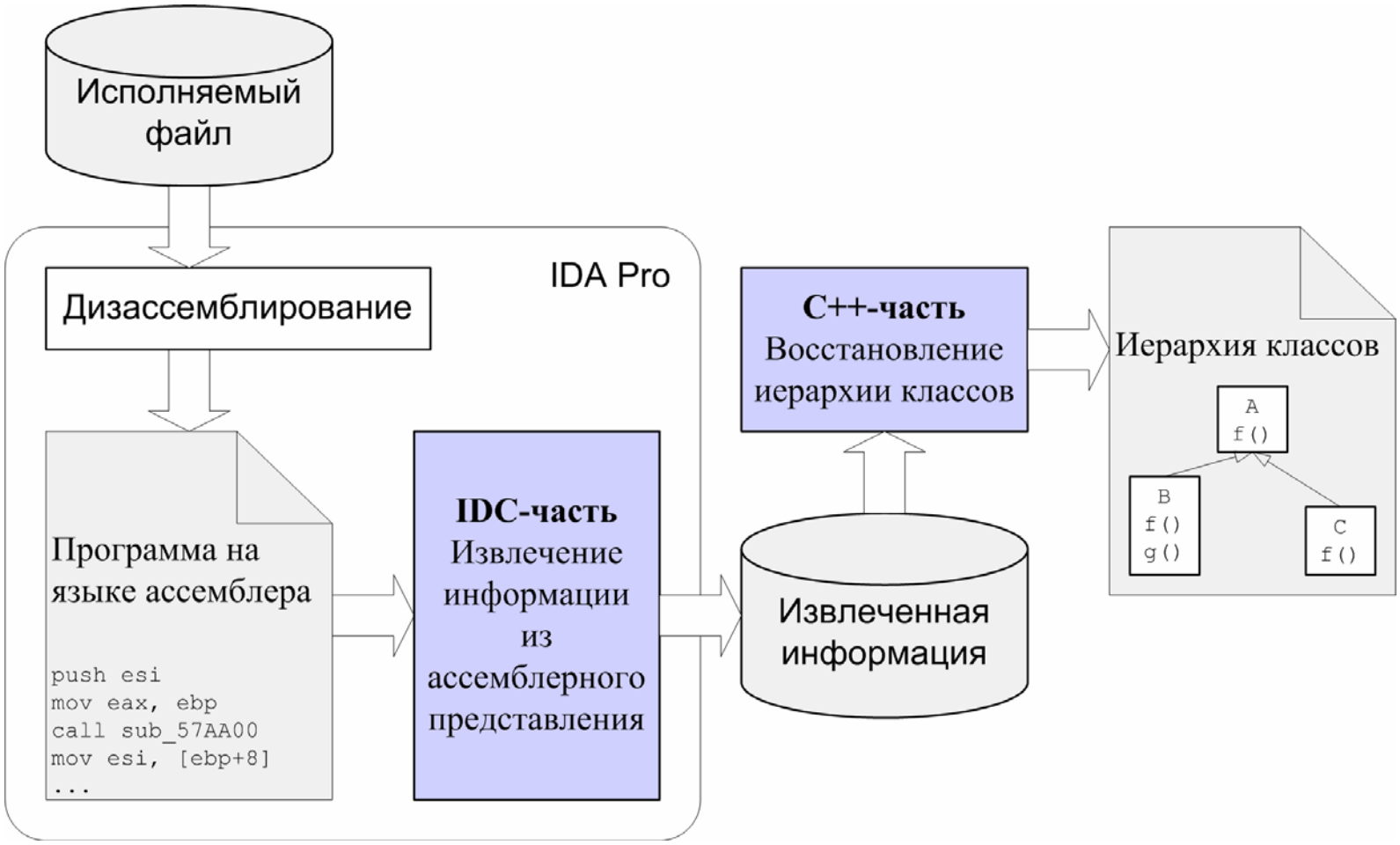
Метод основан на анализе:

- Таблиц виртуальных функций;
- Параметров виртуальных функций;
- Вызовов виртуальных функций;
- Виртуальных деструкторов.

Каждый из пунктов накладывает ограничения на возможную структуру иерархии классов. После сбора всех ограничений применяется алгоритм построения иерархии классов, удовлетворяющей соответствующему множеству ограничений.

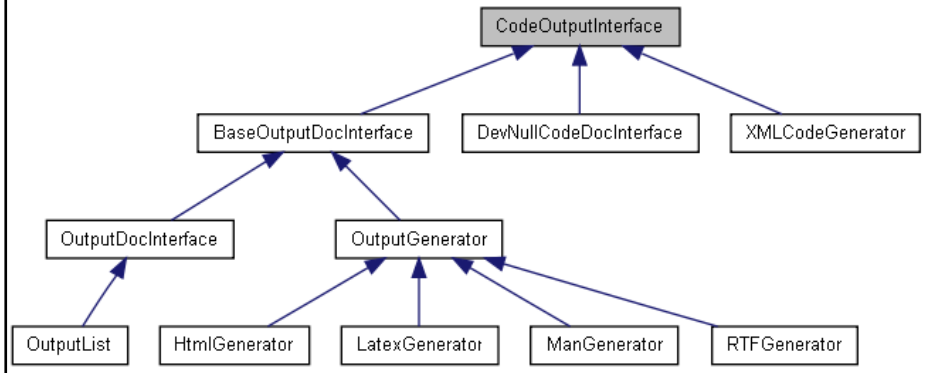
Реализация

Предложенные методы восстановления объектных структур данных реализованы в приложении crec.



Апробация

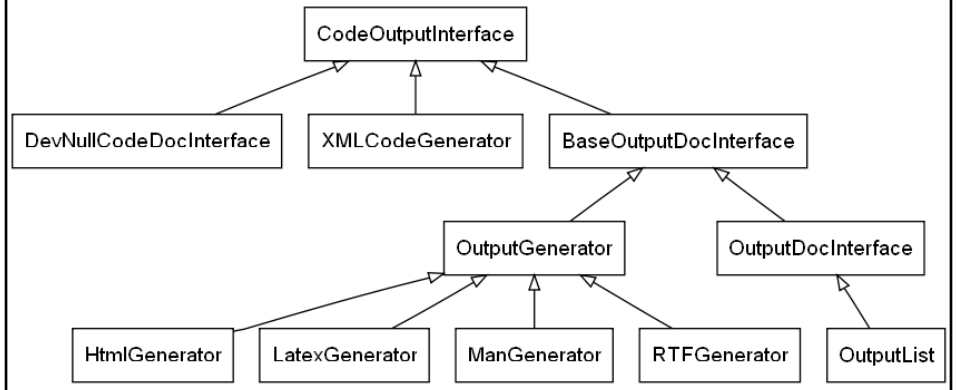
Восстановление иерархии классов при наличии информации о типах времени выполнения



Фрагмент иерархии классов doxygen 1.5.8, полученный путем анализа исходного кода.

Инструмент анализа: doxygen 1.5.8.

Инструмент визуализации: Graphviz 2.16 dot.



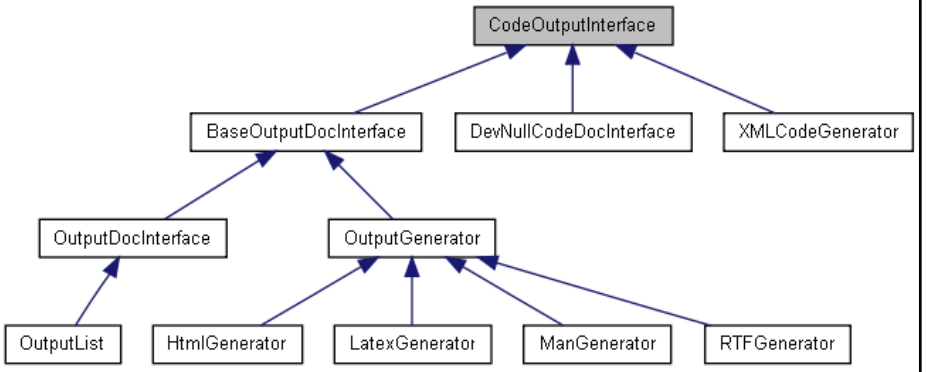
Фрагмент иерархии классов doxygen 1.5.8, полученный путем анализа исполнимого файла.

Инструмент анализа: скрипт для IDA Pro, анализирующий RTTI структуры.

Инструмент визуализации: Graphviz 2.16 dot.

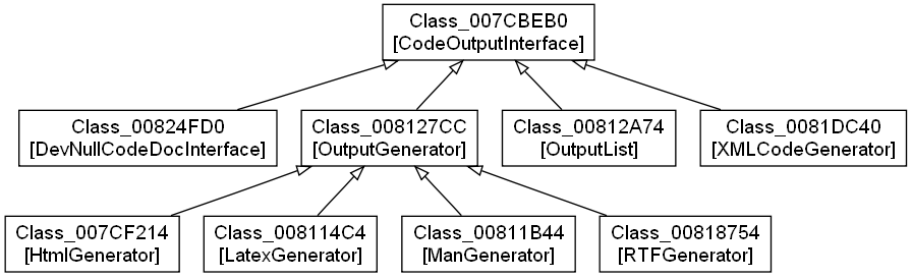
Апробация

Восстановление иерархии классов при отсутствии информации о типах времени выполнения



Фрагмент иерархии классов doxygen 1.5.8, полученный путем анализа исходного кода.

Инструмент визуализации: Graphviz 2.16 dot.



Фрагмент иерархии классов doxygen 1.5.8, полученный путем анализа исполнимого файла.

Инструмент визуализации: Graphviz 2.16 dot.

Заключение

В работе предложены методы автоматического восстановления объектных структур данных из низкоуровневого представления программ на языке Си++.

Выполнена прототипная реализация предложенных методов в рамках интерактивного дизассемблера IDA Pro.

Апробация на приложениях с открытым исходным кодом показала состоятельность предложенных методов.

Спасибо за внимание