Балаковский инженерно-технологический институт - филиал

федерального государственного автономного образовательного учреждения

высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий

Кафедра

КОНТРОЛЬНАЯ РАБОТА

по дисциплине

\_\_\_\_ Инструментальные средства информационных систем\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил: студент группы \_ ИФСТ 2з\_\_\_ \_\_\_\_\_\_\_\_\_\_Семенов М.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

|  |  |
| --- | --- |
|  | Проверил \_\_\_\_\_\_\_\_\_\_\_  должность  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г. |

Балаково 2022

1. Дизассе́мблер

Дизассе́мблер (от англ. disassembler [ˌdɪsəˈsɛmblə(r)]) — транслятор, преобразующий машинный код, объектный файл или библиотечные модули в текст программы на языке ассемблера.

По режиму работы с пользователем делятся на автоматические и интерактивные. Такие дизассемблеры генерируют готовый листинг, который можно затем править в текстовом редакторе. Пример интерактивного — IDA. Он позволяет изменять правила дизассемблирования и является весьма удобным инструментом для исследования программ.

Дизассемблеры бывают однопроходные и многопроходные. Основная трудность при работе дизассемблера — отличить данные от машинного кода, поэтому на первых проходах автоматически или интерактивно собирается информация о границах процедур и функций, а на последнем проходе формируется итоговый листинг. Интерактивность позволяет улучшить этот процесс, так как просматривая дамп дизассемблируемой области памяти, программист может сразу выделить строковые константы, дать содержательные имена известным точкам входа, прокомментировать разобранные им фрагменты программы. Обычно однопроходный дизассемблер (как и построчный ассемблер) является составной частью отладчика.

Чаще всего дизассемблер используют для анализа программы (или её части), исходный текст которой неизвестен — с целью модификации, копирования или взлома. Реже — для поиска ошибок (багов) в программах и компиляторах, а также для анализа и оптимизации создаваемого компилятором машинного кода.

При работе с исполняемым кодом или байт-кодом, созданным на некоторых языках высокого уровня (например, java) имеется возможность восстановить не только текст на языке ассемблера, но даже и структуру классов программы, а если при компиляции исполняемого файла не была отключена отладочная информация — то и исходный текст программы. Для исключения таких возможностей используется обфускация.

Примеры программ-дизассемблеров:

* IDA
* Sourcer
* Hiew
* Beye
* HT editor
* Hacker Disassembler Engine
* CADt
* Objdump
* Radare2
* Ghidra

1. Декомпиля́тор

Декомпиля́тор — это программа, транслирующая исполняемый модуль (полученный на выходе компилятора) в эквивалентный исходный код на языке программирования высокого уровня.

Декомпиля́ция — процесс воссоздания исходного кода декомпилятором.

Декомпиляция, в частности, используется при обратной разработке программ.

Удачность декомпиляции зависит от объема информации, представленной в декомпилируемом коде. Байт-код, используемый большинством виртуальных машин (таких как Java Virtual Machine или .NET Framework Common Language Runtime) часто содержит обширные метаданные, делающие декомпиляцию вполне выполнимой, в то время как машинный код более скуден и сложен в декомпиляции. В частности трудночитаемыми представляются вызовы подпрограмм или функций с косвенной адресацией вызовов (в терминах языков программирования высокого уровня — вызовы через указатели на функции/процедуры).

Если известно на каком языке была написана декомпилируемая программа, то в первую очередь дизассемблируются и анализируются библиотеки времени исполнения (RTL — runtime library) компилятора с этого языка, так как в основном компиляция программы сводится к вызовам с различными параметрами процедур из этих библиотек. Кроме того, многие компиляторы позволяют увидеть, в какой ассемблерный код превращаются операторы программы после компиляции. Эти конструкции становятся шаблонами для декомпилятора, поэтому процесс декомпиляции в некотором смысле похож на распознавание в машинном коде конечного набора подобных шаблонов. Легче всего распознаются вызовы процедур и возвраты из них. Они служат границами для восстановления операторов процедуры.

Некоторые компиляторы и инструменты, используемые после компиляции, подвергают программный код обфускации с целью затруднить декомпиляцию.

Классификация по режиму работы с пользователем делятся на автоматические и интерактивные. Примером автоматических дизассемблеров может служить Sourcer. Такие дизассемблеры генерируют готовый листинг, который можно затем править в текстовом редакторе. Пример интерактивного — IDA. Он позволяет изменять правила дизассемблирования и является весьма удобным инструментом для исследования программ.

Классификация Дизассемблеры бывают однопроходные и многопроходные. Основная трудность при работе дизассемблера — отличить данные от машинного кода, поэтому на первых проходах автоматически или интерактивно собирается информация о границах процедур и функций, а на последнем проходе формируется итоговый листинг. Интерактивность позволяет улучшить этот процесс, так как просматривая дамп дизассемблируемой области памяти, программист может сразу выделить строковые константы, дать содержательные имена известным точкам входа, прокомментировать разобранные им фрагменты программы. Обычно однопроходный дизассемблер (как и построчный ассемблер) является составной частью отладчика.

Примеры декомпиляторов:

* FernFlower
* .NET Reflector, dotPeek — для декомпиляции сборок .NET
* ILSpy
* Delphi Decompiler
* JAD — JAva Decompiler

1. Дизассемблеры и декомпиляторы.

Предназначены для получения исходного кода на языке программирования из исполняемого модуля.

Характеристики:

* поддерживаемые языки (компиляторы);
* возможность использования символьной информации о файле (отладочной и др.);
* возможность интерактивной работы с листингом (замены имен переменных и функций, отслеживания вызовов, модификация кода).

Представители: Interactive DisAssembler (Data Resource), Sourcer, Decafe Pro, DeDe.

Программы отслеживания активности системы и изменений, происходящих в системе.

Позволяют отслеживать действия программ по изменению реестра, файловой системы, вызовов системных сервисов и т.д. Следят за загруженностью системы в целом.

Характеристики:

* тип отслеживаемых изменений/активности;
* возможность протоколирования (логирования);
* возможность фильтрации получаемой информации;
* возможность уведомления.

Представители:

Microsoft: Spy++, Process Viewer, ApiMon, SysMon (для Win2000/XP – ActiveX компонентдля mmc).

Winternals Systems (Mark E. Russinovich): RegMon, FileMon, HandleEx.

Программы-вериферы и контейнеры.

Создают виртуальную среду для отдельных классов программ, в которой можно исследовать поведение программы).

Представители: Driver Verifier, ActiveX Control Test Container (Microsoft Corp.)

Подготовка к анализу загрузочного файла – тема отдельного разговора. Отладчики – это программные средства, предназначенные для контроля выполнения программ. Отладчики позволяют приостановить выполнение программы в некоторой точке, изменить значение переменных и даже, в некоторых случаях, внести изменения в машинный код программы на лету в процессе ее выполнения. К сожалению, возможность выполнения отладчиком подобных действий зависит от включения в выполнимый код отладочной информации, прежде всего таблицы соответствия символов (для большинства загрузочных программ это не выполняется). Если отладочной информации в выполнимом коде нет, то отладчик может выполнить некоторые функции, хотя большую часть работы по отладке программ приходится выполнять вручную, например, при указании точек прерывания вместо имен приходится задавать адреса памяти.

Декомпилятор (или дизассемблер) – программа, которая преобразует двоичный код программ в исходный текст, написанный на одном из языков программирования, чаще всего – ассемблере. Некоторые дизассемблеры могут представить исходный текст на простом языке C. В процессе трансляции большая часть информации об исходном тексте программы теряется, например, имена переменных, поэтому декомпилятор пытается восстановить исходный текст программы настолько, насколько это возможно. Если при декомпиляции таблица соответствия имен была не найдена, то зачастую декомпилятор присваивает переменным имена, составленные из плохо воспринимаемой последовательности цифр и букв.

Проблема несколько упрощается, если исследователь в состоянии разобраться с ассемблерным кодом, генерируемым декомпилятором. В этом случае декомпилятор особенно полезен. Рассмотрим пример результатов работы декомпилятора.

Среди коммерческих декомпиляторов для Windows хорошая репутация у IDA Pro компании DataRescue.