А.В. Макаров, С.Ю. Скоробогатов, А.М. Чеповский

Учебный курс “CIL и системное программирование в Microsoft .NET”

Лекция 11.

Язык CIL: обработка исключений

Введение

- Существует два основных способа перехвата ошибок, возникающих в процессе работы программы:
  - Обработка кодов возврата
  - Обработка исключений
- Обработка исключений частично закодирована в телах методов (в виде специальных инструкций), а частично – в заголовках методов

11.1. Предложения обработки исключений в заголовках методов

- Будем называть областью непрерывную последовательность инструкций в коде метода
- Область будет определяться своими координатами, а именно парой чисел (offset, length), где offset – это смещение первой инструкции области относительно начала тела метода, а length – длина области. Как смещение, так и длину будем измерять в байтах
- Заголовок каждого метода содержит специальный массив, элементы которого называются предложениями обработки исключений (exception handling clause)

Области в предложении обработки исключений

- Каждое предложение обработки исключений представляет собой структуру, состоящую из нескольких полей
- В этих полях записаны координаты двух или трех областей, а именно: в любом предложении присутствуют координаты защищенной области (protected block) и области обработчика (exception handler), а в некоторых предложениях дополнительно описана область фильтра (filter block)

Поля предложения обработки исключений в случае короткого формата

Смещение	Размер	Поле	Описание
0	2	**Flags**	Флаги
2	2	**TryOffset**	Координаты защищенной области
4	1	**TryLength**	
5	2	**HandlerOffset**	Координаты области обработчика
7	1	**HandlerLength**	
8	4	**ClassToken**	Токен метаданных
8	4	**FilterOffset**	Смещение области фильтра

Поля предложения обработки исключений в случае длинного формата

Смещение	Размер	Поле	Описание
0	4	**Flags**	Флаги
4	4	**TryOffset**	Координаты защищенной области
8	4	**TryLength**	
12	4	**HandlerOffset**	Координаты области обработчика
16	4	**HandlerLength**	
20	4	**ClassToken**	Токен метаданных
20	4	**FilterOffset**	Смещение области фильтра

Допустимые значения поля Flags предложения обработки исключений

Значение	Описание
0	Обработчик исключений с фильтрацией по типу (catch)
1	Обработчик исключений с пользовательской фильтрацией
2	Обработчик finally
4	Обработчик fault

- Первые два типа обработчиков мы будем относить к категории обработчиков с фильтрацией, а последние два типа – к категории обработчиков без фильтрации

Типы обработчиков исключений

- Обработчик с фильтрацией по типу
  - Получает управление, если тип исключения совместим по присваиванию с типом, указанным в поле ClassToken предложения обработки исключений
- Обработчик с пользовательской фильтрацией
  - Решение о том, получит или не получит управление обработчик, принимает код, содержащийся в области фильтра
- Обработчик finally
  - Вызывается при выходе из защищенной области, независимо от того, было или не было сгенерировано исключение
- Обработчик fault
  - Вызывается, если внутри защищенной области было сгенерировано любое исключение

11.2. Инструкции CIL для обработки исключений

- В CIL предусмотрено несколько инструкций, отвечающих за порождение исключений и передачу управления из обработчиков исключений

11.2.1. Инструкция throw

- Инструкция **throw** генерирует исключение, включая тем самым механизм обработки исключений

Код	Инструкция	Встроенный операнд	Описание
0x7A	**`throw`**	–	Генерирует исключение

- Диаграмма стека для инструкции **throw**:

  … , **`obj ->`** …

Инструкция rethrow

- Инструкция **rethrow** разрешена только внутри обработчика исключений с фильтрацией по типу и предназначена для генерации того же самого исключения, которое было поймано обработчиком

Код	Инструкция	Встроенный операнд	Описание
0xFE 0x1A	**`rethrow`**	–	Генерирует то же самое исключение, что было поймано обработчиком

- Диаграмма стека для инструкции **rethrow**:

  … **`->`** …

11.2.2. Инструкции leave

- Инструкции **leave** являются аналогами инструкций безусловного перехода и используются для выхода из защищенных областей и областей обработчиков с фильтрацией

Код	Инструкция	Встроенный операнд	Описание
0xDD	**`leave`**	int32	Выход из области
0xDE	**`leave.`**`s`	int8	Выход из области (короткий переход)

- Диаграмма стека для инструкции **leave**:

  … **->**

Инструкция endfinally

- Инструкция **endfinally** используется для выхода из областей обработчиков без фильтрации. У нее есть псевдоним – **endfault**

Код	Инструкция	Встроенный операнд	Описание
0xDC	**`endfinally (endfault)`**	–	Выход из finally-блока (fault-блока)

- Диаграмма стека для инструкции **endfinally**:

  … **`->`** …

Инструкция endfilter

- Инструкция **endfilter** завершает область фильтра
  - Ее основная задача состоит в том, чтобы вернуть целое число (0 или 1). Значение 0 означает, что данное исключение не может быть обработано, и нужно поискать другой обработчик. Значение 1 говорит о том, что нужно передать управление на обработчик

Код	Инструкция	Встроенный операнд	Описание
0xFE 0x11	**`endfilter`**	–	Завершение области фильтра

- Диаграмма стека для инструкции **endfilter**:

  …, **`value ->`** …

11.3. Правила размещения блоков

- Итак, в общем случае предложение обработки исключений определяет три области в коде метода:
  - защищенную область
  - область фильтра
  - область обработчика
- Эти области должны быть расположены в соответствии с определенными правилами

Правила 1 и 2

- Области, определяемые в предложении обработки исключений, не могут перекрываться
- Область фильтра всегда расположена непосредственно перед областью обработчика и завершается инструкцией **endfilter**

Правило 3

- Для любой пары предложений обработки исключений A и B должно быть справедливо следующее:
  - если защищенная область предложения A находится внутри защищенной области предложения B, то области фильтра и обработчика предложения A также должны располагаться внутри защищенной области предложения B
  - если защищенная область предложения A не пересекается с защищенной областью предложения B, то области фильтров и обработчиков этих предложений тоже не должны пересекаться
  - если защищенная область предложения A совпадает с защищенной областью предложения B, то области фильтров и обработчиков этих предложений не должны пересекаться

11.4. Ограничения на передачу управления

- Передача управления внутрь защищенных областей, из них и между ними и их обработчиками регламентирована следующими правилами:
  1. Передача управления на обработчики осуществляется только через механизм обработки исключений
  2. Существует только два способа передать управление извне на защищенную область:
     - передача управления на первую инструкцию защищенной области
     - использование инструкции **leave** из области обработчика с фильтрацией, связанной с данной защищенной областью
  3. Перед входом в защищенную область стек вычислений должен быть пустым

(продолжение на следующем слайде)

Ограничения на передачу управления (продолжение)

- Для выхода из защищенной области, области фильтра или из области обработчика существуют только следующие возможности :
  - порождение исключения инструкцией **throw**
  - использование инструкции **leave** из защищенной области или области с фильтрацией
  - использование инструкции **endfilter** из области фильтра
  - использование инструкции **endfinally** из области без фильтрации
  - использование инструкции **rethrow** из области с фильтрацией

11.5. Семантика обработки исключений

- Система выполнения обрабатывает это исключение в два этапа:
  - Задача первого этапа – поиск подходящего для этого исключения обработчика с фильтрацией
  - Задача второго этапа – выполнение нужных обработчиков без фильтрации и передача управления найденному во время первого этапа обработчику с фильтрацией

Первый этап обработки исключения

- Выполнение первого этапа начинается с просмотра массива предложений обработки исключений, принадлежащего методу, где произошло исключение. В этом массиве осуществляется поиск такого предложения, что:
  - оно описывает обработчик с фильтрацией;
  - адрес инструкции, породившей исключение, попадает в диапазон адресов защищенной области этого предложения;
  - исключение удовлетворяет фильтру обработчика
- Таким образом, на первом этапе finally и fault-блоки пропускаются. Кроме того, происходит последовательный вызов фильтров для блоков с пользовательской фильтрацией

Первый этап обработки исключения (продолжение)

- Если в методе, внутри которого было сгенерировано исключение, не оказалось подходящего предложения обработки исключений, то система выполнения переходит на следующий метод в стеке вызовов
- Первый этап может завершиться либо нахождением подходящего предложения, либо обнаружением того факта, что подходящее предложение не существует на всей последовательности методов в стеке вызовов. В первом случае система переходит ко второму этапу, а во втором выполнение программы аварийно завершается

Второй этап обработки исключения

- На втором этапе система выполнения второй раз просматривает массивы предложений, вызывая все обработчики без фильтрации
- Она останавливается, когда доходит до предложения, найденного на первом этапе, после чего вызывает обработчик, описываемый этим предложением