Лекция 5  
Технологии программирования.

Механизмы исключений, обработка исключительных ситуаций

Исключением называется некоторое событие, которое является неожиданным для системы или прерывает нормальный процесс выполнения программы. При работе программы выделяют 2 основных типа исключений:

* Синхронное
* Асинхронное

Синхронное исключение возникают в заранее известных точках программы. К ним могут относится ошибки чтения файла, коммутации и т.д.

Асинхронные возникают в любой момент времени и не зависят от того какую конкретную инструкцию выполняет программа. Примером исключений может аварийный сбой питания, новые данные и т.д.

Некоторые типы исключений могут быть отнесены как к синхронный, так и к асинхронным. На ряде платформ операция деления на 0 решается на аппаратном уровне. В отсутствии собственного механизма обработки исключений для программ наиболее общей реакцией является прекращение выполнение программы с выдачей пользователю сообщения о характере возникшей ошибки. Можно сказать, что в данном случае единственным обработчиком исключений является ОС. Теоретически возможно игнорирование исключений и продолжение работы программы, однако такая работа может привести к непредсказуемым процессам, связанным с получением неправильного результата. Обработка исключительных ситуаций программой заключается в том, что при возникновении исключения управление передается некоторому заранее подготовленному блоку кода, выполняющему необходимые действия, такой блок принято называть обработчиком исключений.

Существуют 2 принципиально разных механизма функционирования обработчиков исключений

* Обработка с возвратом. Данный механизм предполагает, что обработчик ликвидирует возникшую проблему и приводит программу в состояние, когда она может работать дальше по основному алгоритму. В этом случае после того как выполнится код обработчика управление передается в точку программы, где возникла исключительная ситуация. Обработка с возвратом типична для асинхронных исключений, для синхронных она мала пригодна.
* Обработка без возврата. Заключается в том, что после выполнения кода обработчика исключений, управление передается в некоторое заранее заданное место программы и с него продолжается алгоритм действия. Внутри обработчика может иметься возврат в точку возникновения исключение, но это делается руками

Существует 2 варианта подключения обработчика исключительных ситуаций в программа:

* Структурная обработка. Требует обязательной поддержки со стороны ЯП, так как обеспечивается специальными языковыми конструкции. Такая конструкция имеет контролируемый блок и блок обработки. При возникновении исключений при структурной обработки находится первый обработчик в данном блоке. Код обработчика выполняется, после блок завершается. Структурные исключения используются для исключений, которые могут быть явно сгенерированных с помощью языковых конструкций внутри пользовательской программы. При использовании структурного подхода не накладывается ограничение на число вложенностей обработчиков исключений. Если возникла ситуация, которая не может быть обработана, то она поднимется на уровень выше. В ряде случае бывает неудобно завершать обработку исключений в текущем блоке, тогда происходит ретрансляция исключений, т. Е. явная передача вышестоящим процедурам.
* Неструктурная обработка. Реализуется в виде механизма регистрации функции или команд обработчиков для каждого возможного типа исключений. Современные ЯП предоставляют набор библиотек, в которых имеются 2 стандартные процедуры:
  + Регистрация. Производится привязка типа структуры к исключения
  + Разрегистрация. Отменяет привязку

В случае неструктурной обработчик, если в программе происходит исключение, то начинается выполнение обработчика. Независимо от места возникновения на исключения будет реагировать последний зарегистрированный по данному типу обработчик. В ряде ЯП зарегистрированный обработчик сохраняет свою силу только на блок кода, тогда разрегистрация не требуется. Неструктурная обработка – единственная возможность обработать асинхронные исключения.

Стандарт языка С предусматривает 2 встроенных механизма обработки исключений:

* SEH. Является исключением стандарта С++, при котором исключения возбуждаются. Возбужденные исключения могут обрабатываться как программно, как и аппаратно. Когда возбуждается исключительная ситуация, задается ее код, который рассматривается как тип исключения. Специальный фильтр может обрабатывать код, чтобы организовать различные варианты обработки исключения. SEH предоставляет разработчика программы 2 механизма обработки исключений:
  + Обработка завершения. Try Finally
  + Обработка исключения. Try Except

В выражении фильтре возможно использовать функции GetExceptionCode и GetExceptrionFormance(). \_\_leave передает управление в конец защищаемого блока.

Плюсы:

* + Возможность отловить широкий набор исключений, связанных с арифметическими операциями и операциями со стеками
  + Обработка исключений идет на уровне ядра ОС.
  + Позволяет уменьшить общий объём программы, без включения в сборку стандартных исключений C++, так как в данном случае будут загружены функции ядра

Минусы:

* + Плохая совместимость с C++
  + Автоматический не происходит вызов деструктора пользовательских классов.
  + Нельзя одновременно использовать SEH и CRTL
* CRTL. Структурная обработка исключений, включенная в стандарт языка не зависящая от ядра, так же, как и SEH предполагает наличие 2 блоков. Защищаемый блок начинается с Try, а защищенный с Catch. Параметром Catch является тип и имя возникшего исключения. В отличие от SEH CRTL предполагает выбрасывание исключения. Выбрасывать исключение может только программа, а возбудить все что угодно. THROW ретранслирует исключения на вышестоящий уровень. Для обработки исключения используется информация о переменных на момент генерации исключения, тогда может известен тип исключения и имя. Если ничего не указано в параметре, то Catch отлавливает все исключения

Система просматривает предложение catch последовательно и выбирает для работы первый подходящий фильтр. Все эти чудеса валяются в заголовочном файле exception.h

Генерация исключений

В ряде случае необходимо выбрасывать исключительные ситуации как реакцию на нарушение программной логики. Выбрасываем при помощи Throw, причем параметром будет тип класса исключения.

Непредвиденное исключение

В рамках работы программы могут возникнуть непредвиденные исключения. Для обработки неспицифицированных исключений должна вызываться функция unexpected. Эта функция вызывает текущий обработчик неспицифицированных исключений, которым по умолчанию вызывается системный обработчик terminate, который валит программу с кодом возникшей ошибки. Программист может создать свой обработчик неспицифицированных событий, путем перегрузки функции.