

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

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



Лекция 1. Знакомство с .NET

.NET Framework

- Платформа .NET стоит в одном ряду с самыми значительными достижениями корпорации Microsoft
- Разработка .NET началась в 1998 году. Она сменила несколько названий:
 - Project 42
 - COM Object Runtime (COR)
 - Lightning
 - COM+ 2.0
 - Next Generation Web Services (NGWS)
 - **.NET Framework**
- Спецификация основной части платформы .NET стандартизована ассоциацией ECMA

1.1. Чему посвящен наш курс

Узнаете:

- Архитектура .NET
- Язык CIL
- Принципы динамической генерации кода
- Параллельное программирование в .NET

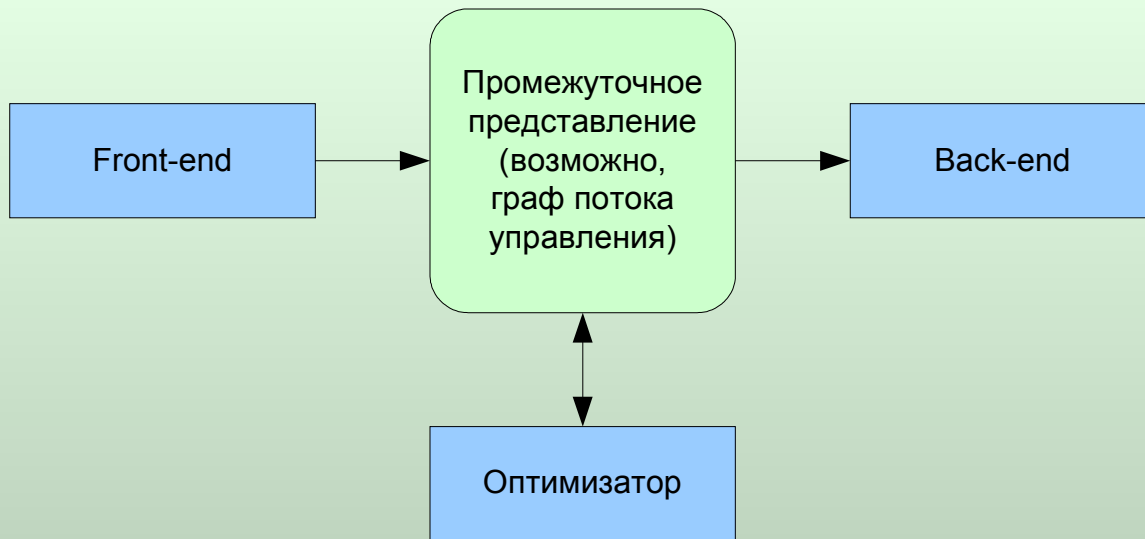
Не узнаете:

- ASP .NET
- ADO .NET
- Разработка графического пользовательского интерфейса
- C# и Visual Basic .NET

1.1.1. Разработка метаинструментов

- Мы будем называть метаинструментами программы, для которых другие программы выступают в роли данных
- Примеры метаинструментов:
 - Компиляторы
 - Средства быстрой разработки приложений (RAD)
 - Оптимизаторы
 - Отладчики
 - Верификаторы
 - Профайлеры
 - и т.п.

1.1.2. Конструирование компиляторов



- В нашем курсе мы будем рассматривать представление кода в виде графа потока управления, узлы которого соответствуют инструкциям языка, а ребра обозначают передачу управления между ними. Такое представление можно использовать в качестве промежуточного представления кода в компиляторе

1.2. Предшественники платформы .NET

- Многие идеи, которые легли в основу платформы .NET, были разработаны задолго до ее появления. В этом разделе мы совершим краткий экскурс в историю и рассмотрим несколько программных систем, которые по праву можно считать предшественниками .NET:
 - UCSD p-System
 - Технология ANDF
 - Платформа Java

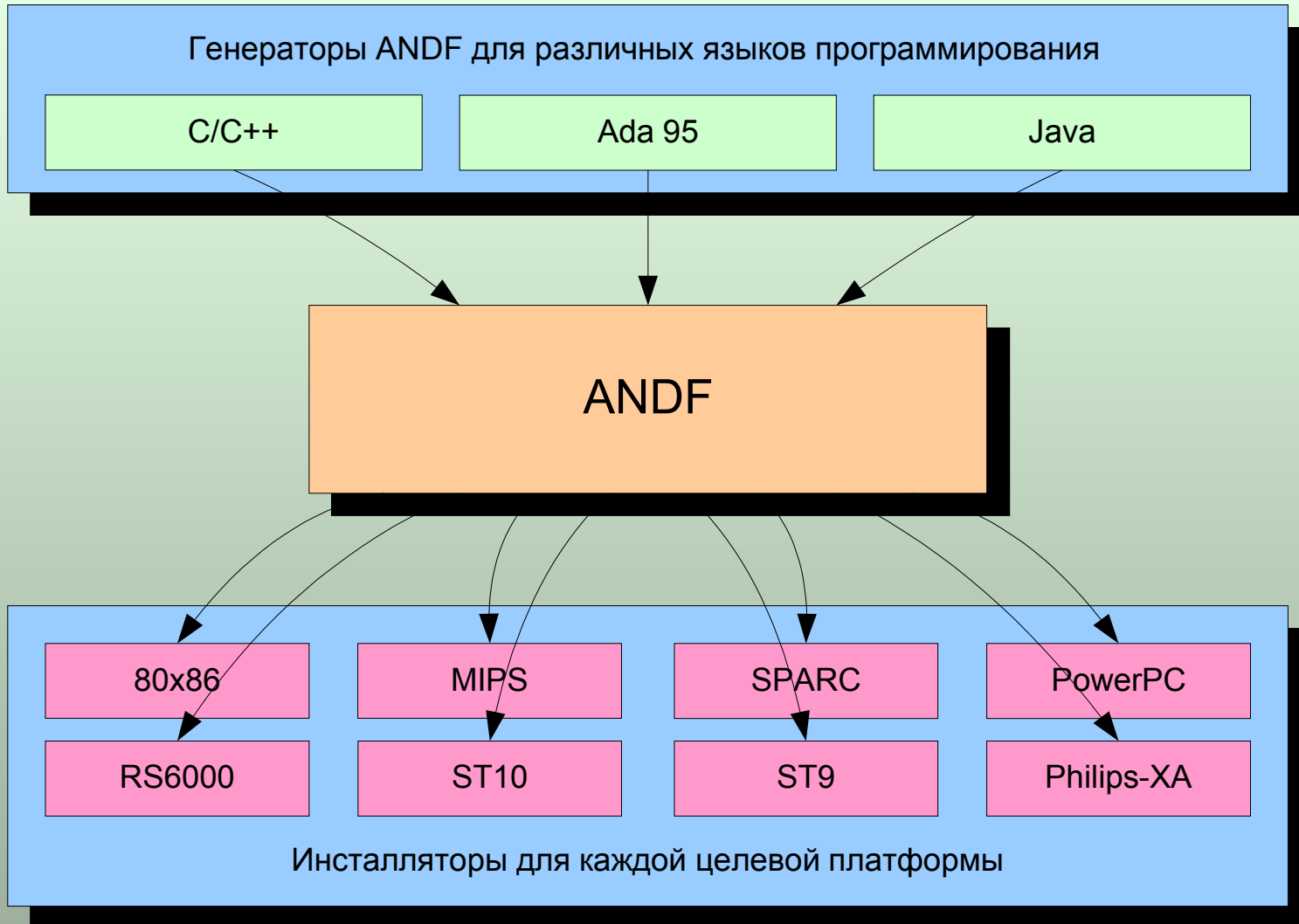
1.2.1. UCSD p-System

- Операционная система UCSD p-System была разработана в 1978 году в Калифорнийском университете для учебных целей
- Независимость операционной системы от аппаратной платформы достигалась путем введения понятия виртуальной р-машины (p-Machine), обладавшей собственным набором инструкций, который назывался р-кодом (p-code)
- Платформа .NET использует похожую схему обеспечения независимости программ от аппаратной платформы. Все программы, работающие на платформе .NET, закодированы на языке CIL, который представляет собой набор инструкций некой абстрактной стековой машины

1.2.2. Технология ANDF

- Технология ANDF (Architectural Neutral Distribution Format) была разработана в первой половине 90-х годов в OSF (Open Software Foundation) для увеличения переносимости программного обеспечения
- Процесса компиляции разделяется на две разнесенные во времени и пространстве фазы:
 - перевод программы в формат ANDF
 - трансляция программы, представленной в формате ANDF, в исполняемый файл при установке программы на компьютер пользователя
- Технология ANDF имеет много общего с принципами распространения программного обеспечения, используемыми на платформе .NET

Схема использования технологии ANDF



1.2.3. Платформа Java

- Платформа Java была разработана в середине 90-х годов в Sun Microsystems для подключаемых к компьютерным сетям бытовых приборов. Затем произошло стремительное развитие Internet-технологий, которое способствовало широкому распространению Java. В настоящее время Java является основным конкурентом платформы .NET
- Краеугольным камнем платформы Java является виртуальная машина, набор инструкций которой называется Java Byte-Code

1.3. Обзор архитектуры .NET

- Платформа .NET состоит из двух основных компонентов:

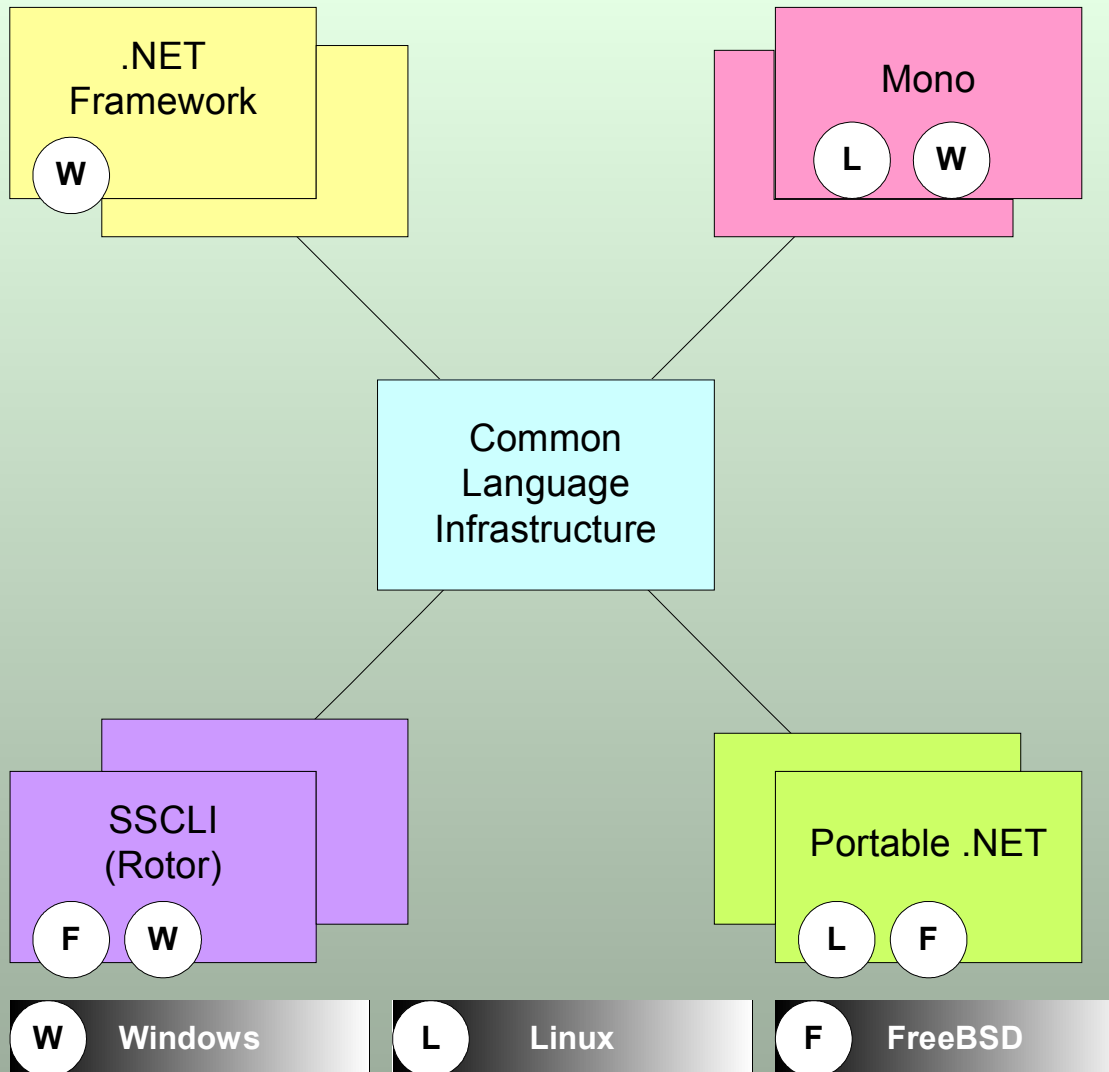
- Common Language Runtime (CLR)

- CLR занимается управлением памятью, компиляцией и выполнением кода, работой с потоками управления, обеспечением безопасности и т.п.

- .NET Framework Class Library

- Единая библиотека для всех языков, реализованных на платформе .NET

1.3.1. Спецификация CLI



- .NET – одна из возможных реализаций так называемой общей инфраструктуры языков (Common Language Infrastructure - CLI), спецификация которой разработана корпорацией Microsoft

Составные части спецификации CLI

- Общая система типов (Common Type System - CTS)
- Виртуальная системы исполнения (Virtual Execution System - VES)
- Система метаданных (Metadata System)
- Общий промежуточный язык (Common Intermediate Language - CIL)
- Общая спецификация языков (Common Language Specification - CLS)

1.3.2. JIT-компиляция

- Программы для платформы .NET распространяются в виде так называемых сборок (assemblies). Каждая сборка представляет собой совокупность метаданных, описывающих типы, и CIL-кода
- JIT-компиляция заключается в том, что CIL-код, находящийся в запускаемой сборке, тут же компилируется в машинный код, на который затем передается управление
- В .NET реализованы два JIT-компилятора: один компилирует сборку непосредственно перед ее выполнением, а другой позволяет откомпилировать ее заранее и поместить в так называемый кэш откомпилированных сборок

1.3.3. Сборка мусора

- Типичные ошибки при управлении памятью:
 - Преждевременное освобождение памяти (premature free)
 - Двойное освобождение (double free)
 - Утечки памяти (memory leaks)
 - Фрагментация адресного пространства (external fragmentation)
- Программы, работающие в среде .NET, используют реализованное в CLR автоматическое управление памятью, а именно – сборщик мусора.
- Данные, с которыми работает программа, находятся под полным контролем среды выполнения и называются, соответственно, управляемыми данными (managed data)

1.3.4. Верификация кода

- Классификация CIL-кода с точки зрения обеспечения безопасности:

- Недопустимый код (illegal code)
- Допустимый код (legal code)
- Безопасный код (safe code)
- Верифицируемый код (verifiable code)

Верифицированный код – это код, безопасность которого может быть строго доказана алгоритмом верификации, встроенным в CLR

- Верифицируемый код всегда является безопасным, а обратное в общем случае неверно