



Common Intermediate Language (CIL, pronunciado «sil» o «kil») (anteriormente llamado Microsoft Intermediate Language o MSIL) es el lenguaje de programación legible por humanos de más bajo nivel en el Common Language Infrastructure y en el .NET Framework. Los lenguajes del .NET Framework compilan a CIL. CIL es un lenguaje ensamblador orientado a objetos, y está basado en pilas. Es ejecutado por una máquina virtual. Los lenguajes .NET principales son C#, Visual Basic .NET, C++/CLI, y J#.

CIL fue conocido originalmente como Microsoft Intermediate Language (MSIL) durante las versiones de prueba de los lenguajes .NET. Debido a esto, CIL es frecuentemente llamado MSIL. Es posible ejecutar este lenguaje en plataformas GNU/Linux gracias al Proyecto Mono, que implementa una máquina virtual similar a la de .NET pero de software libre.

Durante la compilación de los lenguajes de programación .NET, el código fuente es convertido a código CIL en lugar de código objeto específico del procesador o de la plataforma. CIL es un conjunto de instrucciones independientes de la plataforma y que puede ser ejecutado en cualquier CPU

Instrucciones CIL

El código intermedio de CIL incluye un conjunto de instrucciones para las siguientes grupos de tareas:

- Carga y almacenamiento
- Aritméticas
- Conversión de tipos
- Creación y manipulación de objetos
- Operadores de pila (push / pop)
- Transferencia de control (saltos)
- Invocación y retorno de métodos
- Manejo de excepciones
- Concurrencia basada en monitores



Compilación en tiempo de ejecución

La compilación en tiempo de ejecución (JIT, just-in-time) involucra convertir el código intermedio en código inmediatamente ejecutable por la CPU. La conversión es realizada gradualmente durante la ejecución del programa. La compilación JIT proporciona optimización específica para el ambiente, seguridad de tipos en tiempo de ejecución, y verificación de ensamblados. Para cumplir con esto, el compilador JIT examina los metadatos del ensamblado en busca de accesos ilegales y maneja las violaciones apropiadamente.

Generador de imágenes nativas

El generador de imágenes nativas (NGEN) produce una imagen binaria nativa para el entorno actual. El bytecode es omitido completamente o convertido en instrucciones nativas de CPU antes de la ejecución. Esto elimina la sobrecarga del JIT a costo de la portabilidad; cuando una imagen generada por NGEN es ejecutada en un entorno no compatible, el .NET framework automáticamente utilizará el compilador JIT.

Se pretende que NGEN haga la ejecución de ensamblados más rápida al eliminar el proceso de compilación JIT en tiempo de ejecución, pero en algunos casos, esto tiene efectos adversos debido a la optimización hecha por el optimizador JIT. Por este hecho, se sugiere usar NGEN solo después de medir el rendimiento de la aplicación antes y después de usarlo.

Metadato

Artículo principal: Metadatos .NET.

Los metadatos son información acerca de las clases compiladas. Al igual que los tipos de biblioteca en el Component Object Model, permite a las aplicaciones proporcionar y descubrir interfaces, clases, tipos, métodos y campos en el ensamblado. El proceso de lectura de metadatos es llamado reflexión.



Ejecución de CIL

- El código fuente es convertido a Common Intermediate Language y un ensamblado .NET es creado.
- Hasta la ejecución de un ensamblado .NET, su CIL es pasado a través del compilador JIT del Common Language Runtime para generar código nativo. (La compilación NGEN elimina este paso en tiempo de ejecución.)
- El código nativo es ejecutado por el procesador de la computadora.