المشكلة عندنا ان مش كل البرامج اللي بتطلع platform compatible يعني بتشتغل ع كل الاجهزه بنفس الكفاءة

C,C++ => Cross platform but not platform compatible

يعني ايوه بيشتغلوا ع كل الانظمة بس مش بنفس الكفاءة يعني ممكن يشتغل ف ويندوز احسن من لينكس و هكذا و ده ليه ؟   
لان كل اللغات دي بتطلع الاوتبوت بتاعهم ك كود عبارة عن 0و1 بس مش كل الانظمة بتتعامل مع الكود اللي خارج ده بنفس الآلية

ف ده هو اللي عمل المشكلة.

Any 0,1 code may run on all platforms but not in the same efficiency.

ميكروسوفت عملت C# عشان تحل الموضوع ده ف مبقاش الكود بيتحول ل 0و1 ، لا ده بقي حاجه اسمها

MSIL =>Microsoft intermediate language

وبعدين بيعدي ع حاجه اسمها JIT دي بتبقي محددة لكل نظام تشغيل و دي اللي بتحول ال IL لـ 0و1 ع حسب نظام التشغيل

لما بنكتب بقي اي كود ف C# احنا كده بنكتب نوعين من الكود ، اول واحد الكود اللي احنا كاتبينه شوية الاوامر بتاعتنا دي

لكن تاني واحد هو هو بيستدعي مكتبات و حاجات built-in ، الحاجات دي بقي اسمها BCL اللي هي زي console.writeline

طيب دي بتتكتب ازاي ف الIL ؟ بيجي الـBody بتاعها الأصلي في الكود

ف كده ال IL code النهائي بيبقي فيه بس الكود بتاعي اللي انا كتبته ب ايدي والـ Calling بتاع ال BCLs

المهم ان فالاخر الكود بقي بيتعمل كده

C# code => MSIL =>JIT => 0,1 code

وطبعا ال JIT دي بتختلف من نظام للتاني

C# code is a managed code

طيب يعني ايه اصلا كلمة managed ؟

زمان ف C,C++ انت كنت مسؤول عن حتة ال memory allocation ، كنت بتاخد المفك و تنزل الميموري تحجز المكان بتاعك :’’

و بوينترز بقي و كلام كبير و كمان كنت مسؤول عن انك تفضي المكان اللي انت حجزته ده والا هيبقي حاجز مساحة ع الفاضي وانت مبتستخدمهاش.

لكن فالـ C# بقي في حاجه اسمها Garbage collection مسؤولة عن الموضوع ده

Garbage collector handles the memory.

في حاجه كمان ف C# اسمها Type checker دي اللي بتمنعك انك تستخدم انواع المتغيرات ف بعض ، يعني مينفعش تحط int ف string و كده و كمان بتمنعك من استخدام متغير مش محطوط ليه قيمه.

في حاجه بقي اسمها CLR

CLR => Common Language Runtime

ده عنده حاجات كتير هي اللي بتخلينا بقينا نكتب managed code

و ده كمان اللي بيخلينا نقدر نعمل Jitting اصلا عشان الكود يتحول ل 0و1 يعني ال jit ده اصلا جزء من CLR