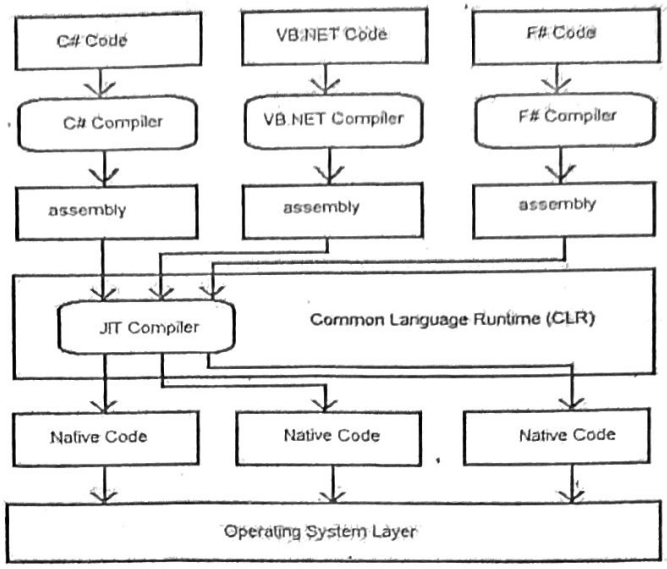
Figures Shows Compile-Time and Runtime Process



Metadata and Assemblies

Metadata :- Metadata is the binary information describing the program, which is either stored in a portable executable file (PE) or in the memory

- Metadata is the complete way of describing what is in a .NET assembly.
- Digging into the metadata yields the types available in that assembly, viz. classes, interfaces, enums, structs, etc., and their containing namespaces, the name of each type, its visibility/scope, its base class, the interfaces it implemented, its methods and their scope, and each method's parameters, type's properties, and so on.
- The assembly metadata is generated by the high-level compilers automatically from the source files.
- The compiler embeds the metadata in the target output file, a dll, an .exe or a .netmodule in the case of multi-module assembly.
- In the case of a multi module assembly ever module that contains IL must have the metadata embedded in it to describe the types in that module.
- Every compiler targeted for the .NET CLR is required to generate and embed the metadata in the output file, and that metadata must be in a standard format. .NET Reflection extensively uses the metadata information to know the type information dynamically.
- Metadata is additional information in a managed assembly describing things like types, type names, method names, etc (basically, the information that you can retrieve from the Reflection services).

Assemblies:-Assembly is a logical unit consisting of the assembly manifest, type metadata, IL code, and a set of resources like image files.

Assembly Manifest: