

CLR-ը (.NET-ի սիրտն է):

Common Language Runtime-ը (CLR) Microsoft .NET Framework-ի հիմնական գործառնական միջավայրն է: Այն ապահովում է ծրագրի կատարման համար անհրաժեշտ բոլոր low-level ծառայությունները՝ հիշողության կառավարում, անվտանգության ստուգումներ, threading, exception handling և այլն: Կարելի է ասել՝ CLR-ը .NET-ի “virtual machine”-ն է, ինչպես Java-ի JVM-ը:

Հիմնական ֆունկցիաները

1. Հիշողության կառավարում (Memory Management)

- Կիրառում է *Garbage Collector*՝ ավտոմատ մաքրելու համար չօգտագործվող օբյեկտները:
- Բացառում է memory leak-երի և manual allocation/free սխալների վտանգը:

2. Անվտանգություն (Security)

- Աջակցում է Code Access Security (CAS) և type safety:
- Ստուգում է, որ կոդը չխախտի թույլատրությունների սահմանները:

3. Կատարման կառավարում (Execution Management)

- *Just-In-Time (JIT) Compiler*-ը MSIL (Intermediate Language) կոդը թարգմանում է native machine code:
- Տարբեր JIT տեսակներ կան՝ Normal JIT, Pre-JIT (NGen), Econo JIT:

4. Լեզվական ինտեգրում (Language Integration)

- CLR-ը ապահովում է, որ տարբեր .NET լեզուներ (C#, VB.NET, F#, ...) կարողանան աշխատել միասին:
- Սա հնարավոր է, քանի որ բոլոր լեզուները կոմպիլացվում են Intermediate Language (IL) կոդի, որն identical է բոլորի համար:

5. Exception Handling

- Տրամադրում է միասնական մեխանիզմ՝ errors/exceptions-ի համար:
- Օրինակ, C# և VB.NET ծրագրերը կարող են catch անել միմյանց thrown exceptions:

6. Thread Management

- CLR-ը կառավարում է բազմաթելայինությունը (multithreading):
- Տրամադրում է synchronization primitives (Monitor, Mutex, Semaphore):

CLR-ի Կազմը

- **Class Loader** – բեռնում է assemblies և classes:
- **MSIL to Native Compiler (JIT)** – թարգմանում է intermediate code-ը:
- **Garbage Collector (GC)** – կառավարում է հիշողությունը:
- **Security Engine** – ստուգում է թույլատրությունները:
- **Exception Manager** – աշխատեցնում է error handling-ը:
- **Thread Support** – կառավարում է thread-երը:

CLR-ի աշխատանքային ցիկլը

1. Ծրագրի կոդը (օր.՝ C#) կոմպիլացվում է **IL** (Intermediate Language):
2. IL-ը պահվում է **Assembly** (.exe կամ .dll) ֆայլում:
3. Երբ ծրագիրը աշխատում է, **CLR-ը բեռնում է Assembly-ն**:
4. **JIT Compiler-ը IL-ը թարգմանում է native code՝** կոնկրետ օպերացիոն համակարգի ու արոցետորի համար:
5. **CLR-ը կառավարում է կատարման ընթացքը՝** հիշողություն, անվտանգություն, exceptions, threading:

Օրինակ (C#)

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        Console.WriteLine("CLR Demo: Hello from .NET!");
```

```
    }
```

```
}
```

Այս կոդը compile անելուց հետո կստանա IL կոդ (.exe), իսկ CLR-ը run-time-ում JIT-ով թարգմանում է դա native instructions-ի: