Difference Between Static and Dynamic Loading in Operating System

Last Updated : 22 May, 2025

Static loading means loading the entire program and all necessary libraries into memory before starting execution. Dynamic loading loads only the parts of the program such as specific functions or libraries when they are needed during execution.

Static Loading in OS

In static loading, the entire program is compiled and linked, leaving no program or external module dependencies at compile time. Static Loading is very useful when the program isn't too big or complicated and when you plan to use the program multiple times.

Example: Static loading is only done in the case of structured programming languages like C.

Features of Static Loading

1. Static loading is used as a small program for memory usage.
2. The program is linked and compiled before being loaded and also all the modules and libraries are included in the executable.
3. Static loading is generally faster than another way of lading the program.
4. It has less processing power consumption.
5. Once a program is loaded, then execution depends on the memory unit.
6. Static loading is more secure because the entire program is loaded only once.

Advantages

- **Faster execution:** With static loading, the entire program is loaded into memory before execution, which means that there is no need to read the program from disk during runtime. This can lead to faster execution times.
- **Predictable behavior:** Since the entire program is loaded into memory, the behavior of the program is more predictable. This can be particularly useful for time-critical or safety-critical applications.
- **More efficient use of resources:** Static loading can be more efficient in terms of resource usage, since the program's memory space is allocated up front, rather than being dynamically allocated during runtime.

Disadvantages

- **Higher memory usage:** Since the entire program is loaded into memory, static loading can lead to higher memory usage. This can be particularly problematic for larger programs, or on systems with limited memory.

- **Longer startup time:** Static loading can lead to longer startup times, since the entire program must be loaded into memory before execution can begin. This can be particularly problematic for time-sensitive applications.
- **Inflexibility:** Static loading can be less flexible than dynamic loading, since it requires that the entire program be loaded into memory before execution. This can make it more difficult to implement certain programming techniques, such as dynamic linking.

Dynamic Loading in OS

Dynamic loading refers to the technique where a program loads parts of its code, such as functions or libraries, into memory only when needed during execution, rather than loading everything at the start. This process allows a program to be more efficient in terms of memory usage.

Example: Dynamic loading occurs in OOPs languages such as C++ and Java.

Features of Dynamic Loading

1. It reduces memory usage and improves performance.
2. In Dynamic loading, the program is not linked and compiled before being loaded in memory.
3. It has more processing power consumption.
4. It provides more efficiency and flexibility for the program.
5. It only loads what's needed of the user and makes the faster startup time.
6. All the necessary modules and libraries are pre-built when the system is needed.

Advantages

- **Lower memory usage:** Dynamic loading can reduce memory usage, since only the parts of the program that are needed at a given time are loaded into memory.
- **Faster startup time:** Dynamic loading can lead to faster startup times, since only the parts of the program that are needed at startup are loaded into memory. This can be particularly useful for time-sensitive applications.
- **More flexibility:** Dynamic loading can be more flexible than static loading, since it allows for dynamic linking and other programming techniques that require runtime loading of code.

Disadvantages

- **Overhead:** Dynamic loading can introduce overhead, as the system must perform additional operations to load code into memory during runtime. This can lead to slower execution times.
- **Unpredictable behavior:** Dynamic loading can lead to unpredictable behavior, since the behavior of the program can change as different parts of the program are loaded and unloaded during runtime.

- **Complexity:** Dynamic loading can be more complex than static loading, particularly when it comes to managing dependencies between different parts of a program.

Differences between Static and Dynamic Loading

Static Loading	Dynamic Loading
Static loading is used to load the entire program in the main memory before executing that program	It refers to the process of loading any program in the main memory according to the conditions.
It doesn't need any additional software to compile the entire program.	In Dynamic loading, all the necessary, modules and libraries are pre-loaded on the system.
It is more secure and more system resources are consumable.	It is less secure but system resources consume only when needed.
The processing speed of the static loading is faster when the files update during processing time.	The processing speed is slower compared to then.
Limited flexibility because all libraries are loaded at once.	More flexible because libraries can be loaded on demand.
Requires a recompilation and relinking for any program change.	Allows for dynamic updates without the need for recompilation.
After loading the code in the main memory it might be executed.	After loading the code in memory execution is required when it is.
For example: static loading happened in a structured programming language like C.	For example: the Dynamic loading in the OOPs language like Java, C++