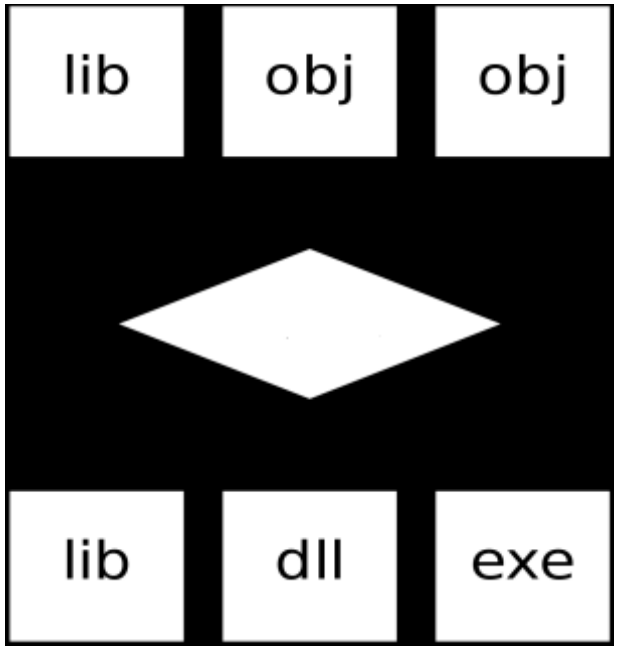
SNS COLLEGE OF ENGINEERING

(Autonomous)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

EC9791 – Embedded and Real Time Systems

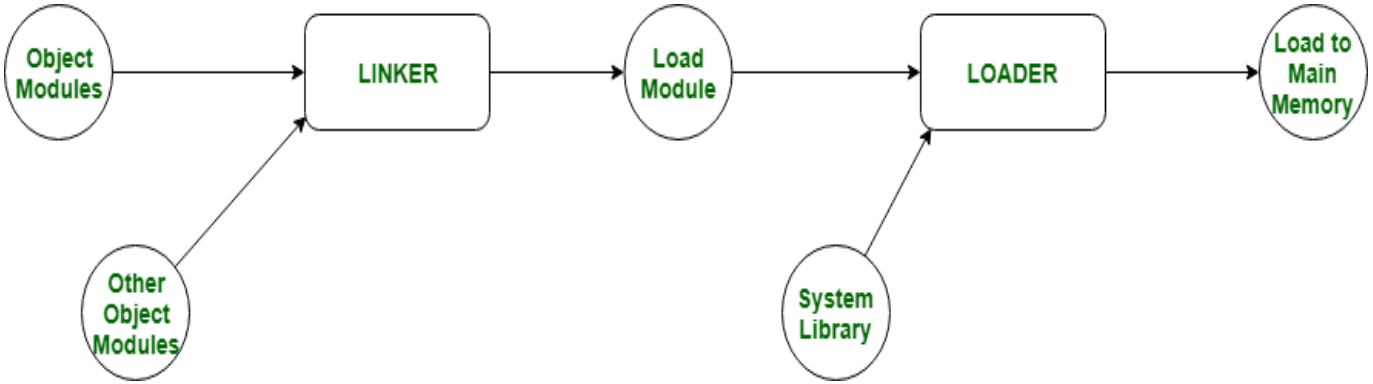


Guess Today's Topic????

Loading and Linking

Loading: Bringing the program from secondary memory to main memory is called Loading.

Linking: Establishing the linking between all the modules or all the functions of the program in order to continue the program execution is called linking.

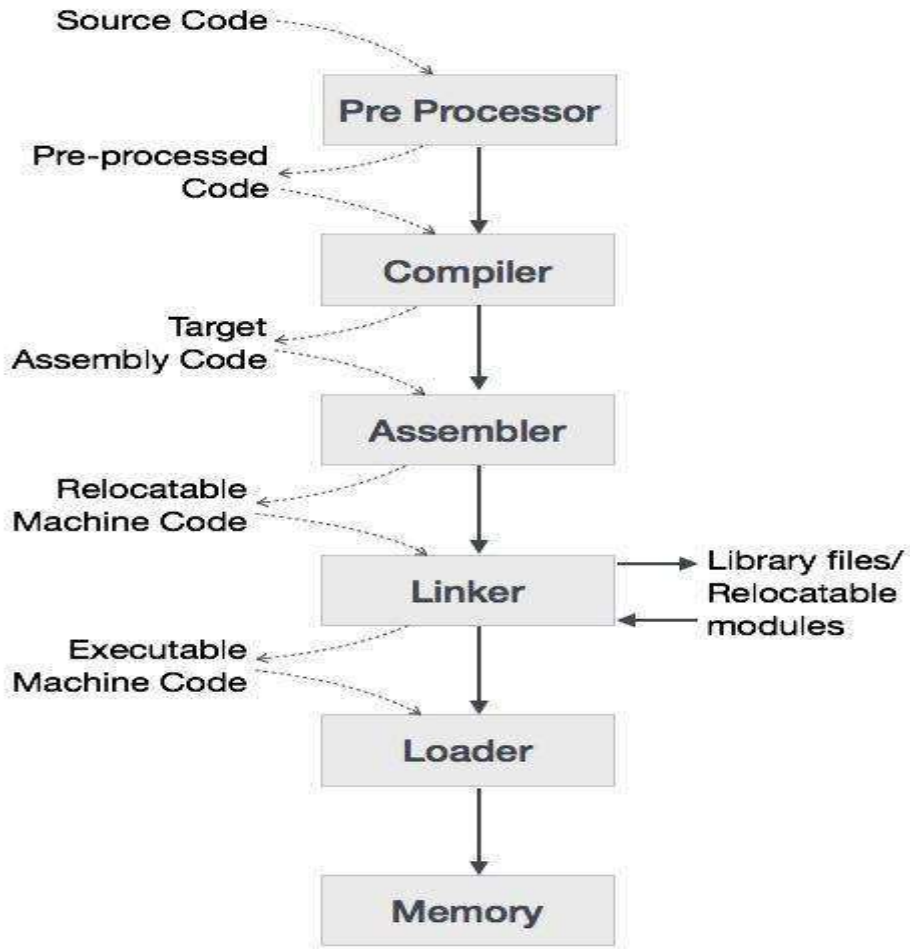


Loading and Linking

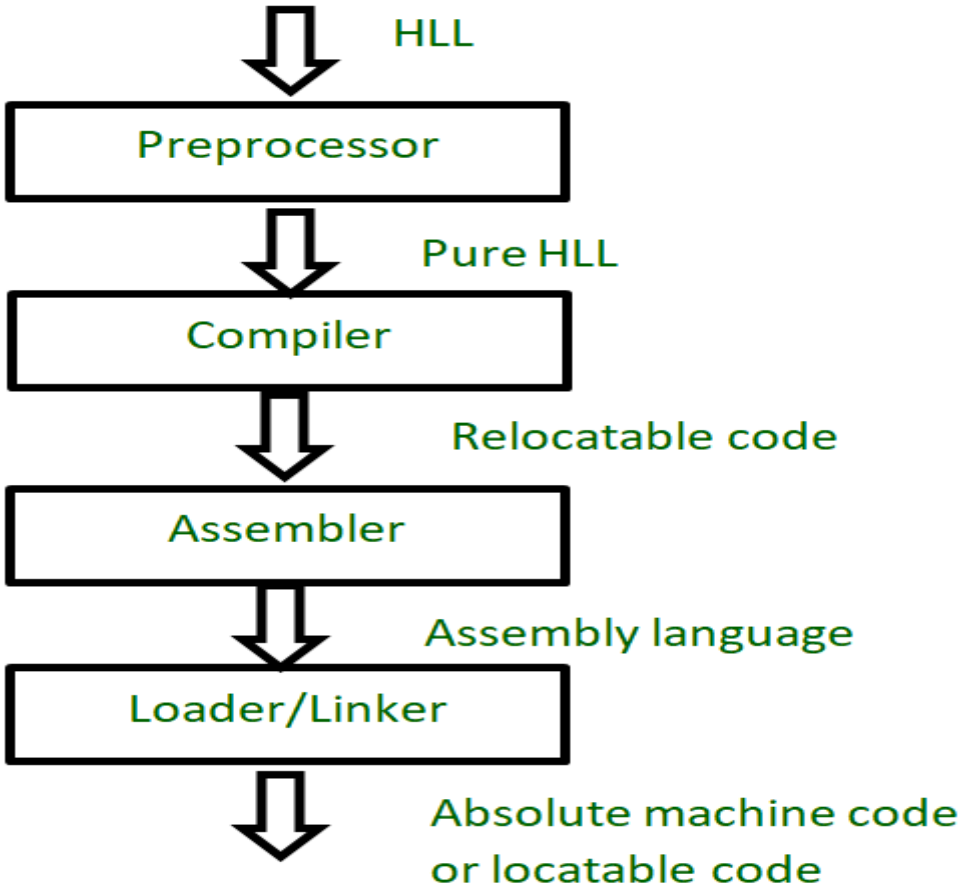
- **Linking is the process of taking several units of compiled code that would otherwise not be executable alone, and combining them into one unit that can be run.**

- **Loading is part of linking - it's how to load object files in memory and analyze them as units of data that can eventually be combined into an executable.**

Flow Chart


Source Code
Pre Processor
Pre-processed Code
Compiler
Target Assembly Code
Assembler
Relocatable Machine Code
Linker
Library files/ Relocatable modules
Executable Machine Code
Loader
Memory

Design Structure



File Status

- **Executable files -**An executable file contains all program data and instructions and associated information required for producing a memory image to execute a program.
- **Object files -**Object files are motivated by the desire to support *separate compilation*: the capability of breaking up complex programs into smaller pieces that can be compiled or assembled independently.

Compilation

- **Separate Compilation**

The ability to break up source code for a program into smaller code units — *separate compilation*

- **Relocation** When a compiler allocates memory locations for a source code file it starts with addresses just above the addresses set aside for the operating system.

- **External references** To be useful, separately compiled files must have references to each other. For example, one file will call subprograms in another file.

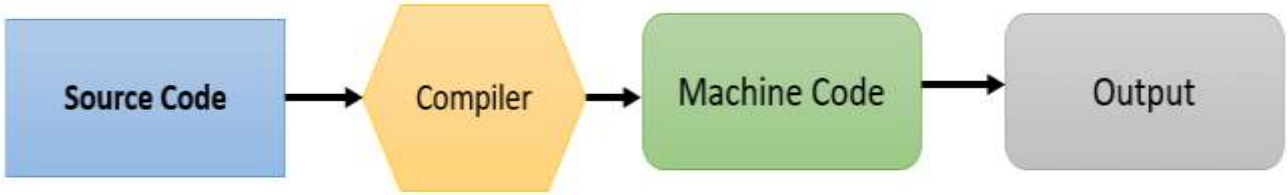
Advantages

Dynamic linking and loading has three important benefits:

- **Software always uses latest versions of shared libraries.**
- **Executable files are smaller. They do not include the shared libraries.**
- **The total memory footprint for multiple processes is reduced. With virtual memory, different programs using the same library function only need a single copy in physical memory.**

Justify the solution

How Compiler Works



How Interpreter Works


Thank
You