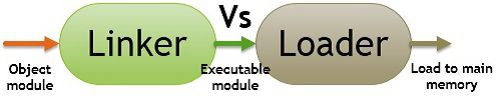
# Difference Between Linker and Loader

Linker and Loader are the utility programs that plays a major role in the execution of a program. The Source code of a program passes through compiler, assembler, linker, loader in the respective order, before execution. On the one hand, where the linker intakes the object codes generated by the assembler and combine them to generate the executable module. On the other hands, the loader loads this executable module to the main memory for execution. Let us discuss the difference between Linker and loader with the help of a comparison chart.

### Comparison Chart

| Basis for Comparison | Linker | Loader |
| --- | --- | --- |
| Basic | It generates the executable module of a source program. | It loads the executable module to the main memory. |
| Input | It takes as input, the object code generated by an assembler. | It takes executable module generated by a linker. |
| Function | It combines all the object modules of a source code to generate an executable module. | It allocates the addresses to an executable module in main memory for execution. |
| Type/Approach | Linkage Editor, Dynamic linker. | Absolute loading, Relocatable loading and Dynamic Run-time loading. |

### Definition of Linker

The Assembler generates the object code of a source program and hands it over to the linker. The linker takes this object code and generates the executable code for the program, and hand it over to the Loader.

The high-level language, programs have some built-in libraries and header files. The source program may contain some library functions whose definition are stored in the built-in libraries. The linker links these function to the built-in libraries. In case the built-in libraries are not found it informs to the compiler, and the compiler then generates the error.

Sometimes the large programs are divided into the subprograms which are called modules. Now when these modules are compiled and assembled, the object modules of the source program are generated. The linker has the responsibility of combining/linking all the object modules to generate a single executable file of the source program. We have two types of linkers.

Linkage Editor: It is a linker that generates the relocatable, executable module.

Dynamic Linker: It defers/postpones the linkage of some external modules until the load module/executable module is generated. Here, linking is done during load time or run time.

### Definition of Loader

As the program that has to be executed currently must reside in the main memory of the computer. It is the responsibility of the loader, a program in an operating system, to load the executable file/module of a program, generated by the linker, to the main memory for execution. It allocates the memory space to the executable module in main memory.

There are three kinds of loading approaches:

* Absolute loading
* Relocatable loading
* Dynamic run-time loading

Absolute loading: This approach loads the executable file of a program into a same main memory location each time. But it has some disadvantages like a programmer must be aware of the assignment strategy for loading the modules to main memory.  In case, the program is to be modified involving some insertion and deletion in the program, then all the addresses of the program have to be altered.

Relocatable loading: In this approach, the compiler or assembler does not produce actual main memory address. It produces the relative addresses.

Dynamic Run-Time loading: In this approach, the absolute address for a program is generated when an instruction of an executable module is actually executed. It is very flexible, the loadable module/executable module can be loaded into any region of main memory. The executing program can be interrupted in between and can be swapped out to the disk and back to main memory this time at a different main memory address.

## Key Differences Between  Linker and Loader

1. The key difference between linker and loader is that the linker generates the executable file of a program whereas, the loader loads the executable file obtained from the linker into main memory for execution.
2. The linker intakes the object module of a program generated by the assembler. However, the loader intakes the executable module generated by the linker.
3. The linker combines all object module of a program to generate executable modules it also links the library function in the object module to built-in libraries of the high-level programming language. On the other hands, loader allocates space to an executable module in main memory.
4. The linker can be classified as linkage editor, and dynamic linker whereas loader can be classified as absolute loader, relocatable loader and dynamic run-time loader.

### Conclusion:

The linker takes the object modules of a program from the assembler and links them together to generate an executable module of a program. The executable module is then loaded by the loader into the main memory for execution.