ELF (Executable Linkable Format)

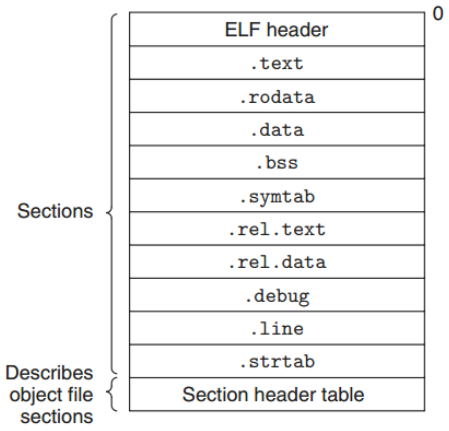
- Is the modern UNIX/LINUX format used to represent an object file (a sequence of bytes stored in the hard disk).

The ELF header begins with 160-byte sequence that describes

- Word-size
- Byte ordering of a system that generated the file.
- Size of the header
- Size of section headers
- Number of section headers
- Start of section header
- Type of object file

The location and size of various sections are described by the section header table and in between this two ELF header and section headers table there exist a section

ELF header example



ELF supports:

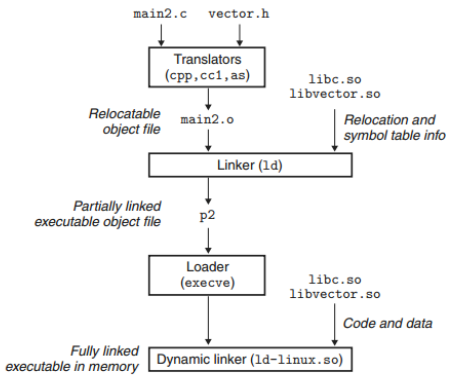
- Different processors
- Different data encoding
- Different classes of machines

Dynamics Linking

-refers much of the linking process until a program starts running. It performs the lining process "on the fly" as programs execute in the system. During dynamic linking the name of the shared library is placed in the final executable file while the actual linking takes place at turn time when bot executable file and library.

The files linked:-

I) Libraries object files
II) Shared object files
III) Relocatable object files



Example of dynamic link usage

Advantage of dynamic linking

- Distributing software
- Building high performance web servers

Static Libraries: - have an extension of .a (hello.a) and they are linked with object files at compile time.

Static libraries, like all software, need to be maintained and updated periodically. If application programmers want to use the most recent version of a library, they must somehow become aware that the library has changed, and then explicitly relink their programs against the updated library.

Dynamics Shared Libraries: - have an extension of .so and they are linked at a load time.

It makes executable smaller in size. In addition, they are modern innovations that address the disadvantage of static libraries.

1. Dynamically linked at run time. The libraries must be available during compile/link phase. The shared object are not included into the executable component but are tied to the execution
2. Dynamically loaded/unloaded and linked during execution (i.e. browser plugin) using the dynamic linking loader system functions.

Loading

-is the operating systems function that calls the system to move the code and the data of the executable object file into memory