

Simple Loader for ELF in C

Contributors :

1→Nimit Bajaj , 2023340 (NAMIT-BAJAJ12)

2→Nihal , 2023345 (GoDRoger69)

Link for Github

repository:<https://github.com/GoDRoger69/Simple-Loader-for-ELF>

Overview:

This code implements a simple ELF (Executable and Linkable Format) loader in C, responsible for loading a 32-bit ELF executable into memory, validating it, setting up necessary memory mappings, and executing its entry point.

Implementation

- **Header Includes and Global Variables:**

The code includes the "loader.h" header and standard system headers for file operations and memory management. Global variables include ehdr (pointer to an Elf32_Ehdr structure for the ELF header), phdr (pointer to an Elf32_Phdr structure for program headers), fd (file descriptor for the ELF file), and virt_mem (pointer to the memory-mapped segment).

- **Loader Cleanup Function:**

loader_cleanup() is used to release resources. It frees the memory allocated for ehdr and phdr, unmaps the virtual memory using munmap() if it is mapped, and closes the file descriptor fd if it is open.

- **Verify ELF Function:**

verify_elf() checks if the file is a valid ELF file by reading its header and comparing its magic number with the expected ELF signature (0x7F 'E' 'L' 'F'). If the check fails, it reports an error and performs cleanup.

- **Load and Run ELF Function:**

1. This is the main function for loading and executing the ELF file. It:
2. Reads the ELF header into ehdr.
3. Iterates through the program headers to find segments of type PT_LOAD.
4. Uses mmap() to allocate virtual memory for the segment defined by p_memsz.
5. Computes the entry point address from the ELF header and maps it to the corresponding function pointer.
6. Calls the function at the entry point, _start, and prints the returned value.

Main Function:

The main function checks the command-line arguments for the ELF file, verifies its validity using verify_elf(), loads and executes the ELF file with load_and_run_elf(), and finally, calls loader_cleanup() to free resources.