Loader_implementation:

● We initiated the process by utilizing the "open" system call to obtain the file descriptor corresponding to the input binary.

Leveraging the acquired file descriptor, we utilized the "read" system call to extract the binary's content.

● With the binary content in hand, we proceeded to allocate heap memory of appropriate size using the "malloc" function.

This dynamically allocated memory was intended to accommodate the binary content that we were in the process of copying.

● As part of our analysis, we systematically iterated through the Program Header (PHDR) table.

Our objective was to identify a section within the table with a "p_type" corresponding to PT_LOAD.

This section was significant as it held the memory address of the entrypoint method present in the "fib.c" file.

● Having determined the target segment, we proceeded to allocate memory using the "mmap" function.

The allocated memory space was sized according to the "p_memsz" value of the identified segment.

● The final step involved replicating the content of the segment we identified earlier.

We meticulously copied this content into the memory space allocated through the "mmap" function.

We proceeded to the next phase by directing our attention to the entrypoint address denoted as "e_entrypoint."

This address corresponded to the segment we previously loaded into memory.

It's important to note that the entrypoint address might not coincide with the starting address specified by "p_vaddr" within that segment.

- As a result, we embarked on a process to traverse through the segment, gradually advancing to the virtual address indicated by "e_entrypoint."
- Our objective was to move through the segment while keeping track of our progress until we arrived at the virtual address identified by "e_entrypoint."
- Specifically, we typecasted the address to that of a function pointer which corresponded to the "matching _start method" present in the "fib.c" file.

With the function pointer in hand, we proceeded to call the "_start" method.