

Group Members

1 . Keshav Chhabra 2022247

2. Kartik Prasad 2022240

Implementation of Code

LOADER.C

Open the file using the open system call and store the file descriptor in fd .

Allocate heap memory pointed by ehdr and phdr .

Read the file using the read system call and store the file descriptor in fileread.

Find the segment containing the entrypoint virtual address by calling the function traversingphdr and loading it . the pointer pointing to the first byte of the mapped segment is returned to virtual_mem .

Typecasting the entrypoint address .

MAKEFILE INSIDE THE LAUNCHER DIRECTORY

Compiles using gcc and generates a Position Independent Code according to a 32 bit architecture . It uses a relative path to link the shared library simplerloader.so .

MAKEFILE INSIDE THE LOADER DIRECTORY

Compiles using gcc and generates a Position Independent Code according to a 32 bit architecture. The "shared" flag indicates the the output file should be a shared library.

THE TOPLEVEL MAKEFILE

invokes the make command in all the specified subdirectories.

Makes the bin directory using mkdir command , it got accidentally deleted during implementation so I had to create it again .

moves launch in launcher to the bin

moves simplerloader.so in loader to bin

Cleans the specified directories and finally removes the bin directory using the rm command .

Contribution

1. By Keshav :

* Implemented the make files of the loader, launcher , and the Top level make file .

* Collaborated in the logic to find the entry point address segment and wrote the documentation .

2 . By Kartik :

* Handled the file handling and mapped the segment containing the entry point address into the memory.

* Implemented Error handling, and pointer arithmetic, and the typecasting of entry point address.

Github repository link

<https://github.com/kartikprasad12345/OSAssignmentrepo.git>

Note:We have also done the bonus question.

