OPERATING SYSTEM

IMPLEMENTING LOADER

GROUP MEMBERS: WASIF ALI(2022583) | VIPUL VERMA(2022577)

GITHUB :- https://github.com/A-WASIF/OS

https://github.com/vipul22577/OS_Repositry/blob/main/without-bonus/loader.c

---

Implementation

We have implemented a SimpleLoader for loading an ELF 32-bit executable in plain-C without using any library APIs available for manipulating ELF files.First of all with the use the “open” system call to get the file descriptor for the input binary and “read” system call to read the content of the binary and then checks for the error after that create a heap allocated memory of appropriate size with the use of malloc function.

After all ,Iterate through the PHDR table and check the section of PT_Load type that contains the address of the entrypoint method in fib.c

Allocate memory using mmap function and then copy the segment content.
Navigate to the entrypoint address into the segment loaded in the memory.The entrypoint address may not be the starting address in that segment.You have to walk that segment to reach the virtual address.

And the final step is we have reached that location, simply typecast the address to that of the function pointer matching “_start” method in fib.c and call the “_start” method and print the value returned from the “_start”.

Contribution

***WASIF ALI*:-**Contribution of ,conversion of file into binary and after memory allocation iterate through the program header table and then check type of load ,and last point is memory allocation by mmap function,formatting source code.

VIPUL VERMA:- Contribution of ,navigating to the entrypoint address into the segment loaded in the memory, typecast the address to that of function pointer matching “_start” method in fib.c,Call the “_start” method and print the value returned from the “_start”.

Error checking.