**OPERATING SYSTEM**

**MAKING SIMPLE LOADER using C**

**Detailed Implementation:**

The provided C code implements a simple ELF loader, responsible for loading and executing ELF executable files. The code demonstrates key steps involved in loading an ELF file into memory and invoking its entry point.

Firstly, we use an system call “*open()*” which returns a file descriptor(fd) upon successful file opening. We get an integer output from fd. Then we use memory allocation for ehdr and phdr using “*malloc*” of their appropriate size and we use “*readFile()*” , “*movFilePointer()*” to read the file and iterate through the file.

After this, we Iterate through PHDR table to find the section of PT\_LOAD (p\_type) type that contains all the address of the entry-point method in fib.c.

After finding that PT\_LOAD we allocate the memory using mmap function and copy the segment content using “*readFile()*” , “*movFilePointer()*”.

After this, we navigate to the entrypoint address into the segment loaded in the memory and we simply typecasted the addresses of the function pointer matching “\_start” method in fib.c and called the “\_start” method and print the value returned by the “\_start”.

**Contribution:**

Prince → PHDR allocation and error handling, readFile() , movFilePointer() making and error handling i.e readFile uses read() system call to read the file and movFilePointer uses lseek() system call to iterate pointer. Lastly typcasting to get the desired output, other rest things also.

Prajil → ElF header allocation, virtual mem allocation, error handling and deallocation after used.

Breifing about how simpleloader was coded:

Github link: <https://github.com/Prince22378/OS-Assignments-2023.git>