OS Assignment - 1 Report

Submitted by Group ID Number - **62**

1. **Akshit Gupta 2022058**
2. **Manit Kaushik 2022277**

Contributions -

Manit Kaushik	Akshit Gupta
• Figuring out how to iterate through the program header table and method of identifying the section of PT_LOAD (p_type) type that contains the entry point address. • Handle the memory mapping process using the mmap function, and ensure that memory is allocated based on the size "p_memsz" and that the segment content is copied accurately. • Provided comments and remarks in the code explaining the functionality and working of the program.	• Handled the "open" system call to obtain the file descriptor of the input binary and implemented necessary checks for input ELF File. • Managed the memory allocation for copying the binary content, the ELF Header and Program Header. This includes dynamically allocating memory using malloc and ensuring the memory space is appropriately sized to hold the data needed. • Writing SimpleLoader implementation in the assignment report.

SimpleLoader Implementation -

Initially declaring the necessary global Variables which will be used throughout the code namely **ehdr** (ELF Header), **phdr** (Program Header) and **fd** (file description). In the **main** function, initially adding checks if the input ELF file exists or not and whether it's readable or not. Then, passing it to the loader for carrying out the loading/execution. Then in the function **load_and_run_elf**, firstly opening the ELF file using the **fd** global variable and putting a checking method for the same. Continuing further, we loaded the size of the binary file in a variable so that we can allocate memory for storing the binary content and we do the same for the ELF Header table and Program Header table. Additionally, we move the Setting pointer to the start of the program header table and read the program header available. We use the condition whether if after reading the size of a particular

program header section is the appropriate size to continue the while loop, which is used to find the iterate through the **PHDR** table and find the section of **PT_LOAD (p_type)** type that contains the address of the entrypoint method in **fib.c.** In the while loop, we use the if-else loop to check if the section is **PT_LOAD** type or not and if the entrypoint address is in **phdr->p_vaddr.** When this condition is met we allocate memory of the size "**p_memsz**" using the mmap function. And if this condition is not met then using **lseek** we move to the next program header entry. Once we reach that location we store it using variables and then typecast the address to a function pointer named **my_function** which matches the "**_start**" method in fib.c and then we use the same method and print the value returned from the **_start**. Finally, we invoke the cleanup routine inside the loader using the **loader_cleanup** function.

Private Github Repository Link - https://github.com/ManitK/CSE231-Assignments