

OS Assignment-1:

Goal of this assignment: Creating a simple loader which reads and opens ELF elf, loads segments into memory and prints the required value from it through entry point navigation

Steps of execution:

1. First we create global variables(ehdr,phdr,fd) and free up space to store memory in them respectively
2. Then several function blocks are created such as (validating_elf_magic: which checks if they match elf file syntax; check_lseek_validity: which positions fd to give offset)
3. Error handling for such functions are also created to help error handling catching
4. Then we open targeting elf file, read the ELF header and PHDR header and allocate bytes accordingly
5. Then PDHR is iterated and segments are loaded accordingly by checking type of PT_LOAD and making them virtually using mmap function
6. Then from given hardcoded file, we obtain entry point which helps us point to required segment and prints the final required value

Launcher execution:

1. The goal of the loader is to provide an interface system and show in a structured way to call the loader
2. The launcher expects at max only 1 argument- if more than 2 is implemented, then it exists the program
3. Once it detects one argument being passed, it calls for the loader main function with argv[1]
4. After the ELF program is run and executed, it makes sure the loader cleanup is also executed and everything is released properly

Contribution of each part:

Manojna:

- Debugging of code
- Final Launcher of code working simultaneously with loader
- Implementing the loader file

Shanthini:

- Debugging of code
- Implementing the loader file
- Commenting and explanations of codes

Final repository link of assignment: <https://github.com/shanthini11/OS-assignment1-section-a.gits>