CSE 231: Operating System **Assignment-4 (Section-A)**

SimpleSmartLoader : An Upgraded SimpleLoader in C

Group Details

Group Member 1 :

Name : Himanshu
Roll no: 2023241
Gmail ID : himanshu23241@iiitd.ac.in

Group Member 2 :

Name : Harsh Gupta
Roll no : 2023229
Gmail ID : harsh23229@iiitd.ac.in

Contribution

Himanshu :

1) Error checking during file Accessibility in the main function .
2) Error checking for the suitable permissions required for the file.
3) Writing code for the clean up in the loader cleanup function.
4) Determining size of binary file.
5) Reading and allocating memory for the binary file.
6) Reading and allocating memory for the Elf header.
7) Error checking during memory allocation for Elf header.
8) Setting the pointers to suitable locations using lseek.
9) Editing in load and run elf function a for details of page fault

Harsh Gupta :

1) Checking for errors during binary file opening.
2) Allocating memory and reading program header and setting the offset to start of program header.
3) Iterating the program header .

4) Error checking while iterating the program header in case entry point is not found.
5) Allocating memory using the mmap function for segment of type PT_Load.
6) Navigating to the entry point address in the segment.
7) Type Casting and execute the entry point.
8) Segment fault handler function.

Code Documentation

=> MainFunctions

Using the access function for checking the accessibility of the binary file. The access function takes two arguments, the location of the file and the permission type like R_OK for read permission ,F_OK for the existence of the file.Otherwise the integer will be -1 and the program will exit with an error message. Calling load and run elf with “argv” argument which includes the ELF file name Calling loader cleanup function which will free the memory allocated during load and run elf function call.

=> Load_and_run_elf function

Opening elf file and checking for errors during opening the elf file Using lseek to set the offset at the start of the elf file Reading and allocating memory in heap for the binary file Setting the offset to the start of the binary file Reading and allocating space in heap for the elf header Calculating size of elf header and checking for errors during reading the elf header Setting the offset to the start of the elf header Allocating memory for program header in heap Reading the program header and setting the offset to the start of the program header using the ehdr->e_phoff Iterating through the program header for a total of ehdr->e_phnum time which denotes the total number of elements in the program header In the while loop ,we are checking for errors during loading of each segment and looking for the program header with the type PT_LOAD and if encountered the exiting the loop Checking for errors that if the segment is found or not Allocating memory of the size p_memsz for the segment using mmap function Reading the segment and moving offset to the start of the segment Calculating address of the entry point in the segment Casting the entry point address to the function pointer Calling the start function and printing the result.

Loads the ELF binary and executes its _start function while tracking memory allocation. Manages the full process of loading, executing, and handling segmentation faults to dynamically allocate memory as needed. Handles segmentation faults by allocating memory for fault-causing segments and reports results including the output from _start, the number of page faults, page allocations, and internal fragmentation.

=> loader_cleanup function

Freeing memory allocated during loading and running using the free function.

=> Handling segmentation fault function

This function is a signal handler for segmentation faults that occur when the program tries to access memory that hasn't been allocated yet (a page fault). When triggered, it identifies the address causing the fault and allocates memory for it by mapping pages based on the ELF program's headers. It then loads the needed content into the newly allocated memory. Essentially, this function manages page faults by setting up memory for the requested segment and loading its content while the ELF binary runs.

Github private repository link:

https://github.com/himanshu23241/Operating_System-CSE231