ASSIGNMENT - 4

SIMPLE SMART LOADER: An Updated SimpleLoader in C

Link to GitHub: https://github.com/shrutya22487/OS_Assignment_4.git

IMPLEMENTATION:
This ELF loader implementation loads and runs executable programs in the ELF format. It handles page faults within segments by allocating memory for individual pages. When a page fault occurs, it calculates the page number, allocates a single page at a time, and copies the data from the ELF file into the page. This approach ensures that virtual memory for intra-segment space is contiguous, while physical memory may or may not be contiguous. The code also tracks the number of page faults and internal fragmentation.

1. load_ehdr(size_t size_of_ehdr):

In this function, memory is allocated to store the ELF header, and the ELF header is read from the file into memory. The function also checks whether the opened file is a 32-bit ELF file. It's a critical step to load the necessary information about the ELF file's structure.

2. load_phdr(size_t size_of_phdr):

Similar to load_ehdr, this function allocates memory for program headers and reads them from the file into memory. Program headers contain essential information about various segments in the ELF file, which is needed to load and execute the program.

3. setup_signal_handler():

This function sets up a signal handler, specifically a segmentation fault (SIGSEGV) handler. The signal handler detects and handles page faults (e.g., missing segments) during the program's execution. It efficiently manages page allocation and fault handling.

4. segfault_handler(int signum, siginfo_t *sig, void *context):

The segmentation fault handler is the core of this program. When a page fault occurs, it checks which segment the fault belongs to, allocates memory for the missing segment, and continues the program's execution. It keeps track of the number of page faults and the allocation status of each page within a segment, which is crucial for memory efficiency.

CONTRIBUTIONS:

Swara Parekh (2022524):

- Defined the load_ehdr and load_phdr functions, which are responsible for loading the ehdr and phdr.
- Constructed the load_and_run_elf function, to identify the segment closest to the entry point and finding its address in the ELF.
- Contributed to handling segmentation faults (page faults) by updating the segfault_handler function. It calculates page faults, page allocations, and internal fragmentation.
- Performed advanced functionalities for the code.

Shrutya Chawla (2022487):

- Implemented the find_entry_pt function, which is part of the load_and_run_elf logic, and it iterates through program headers to identify the segment closest to the entry point.
- Contributed to handling segmentation faults (page faults) by updating the segfault_handler function to handle the faults in 4KB page sizes. Finding the address of the segment fault in the ELF and loading the memory.
- Managed error handling wherever necessary.
- Contributed to advanced functionalities for the code.

Sources:

How to use mmap:
https://man7.org/linux/man-pages/man2/mmap.2.html
How to create a static library:
https://makori-mildred.medium.com/how-to-create-static-library-in-c-and-how-to-use-it-b8b3e1fde999

For finding where the fault occurred
https://www.mkssoftware.com/docs/man5/siginfo_t.5.asp