**README**

**Team members:**

Aakash Chityala (2023005)

Parsh Jain (2023368)

**Summary:**

We first start out by opening the ELF file and storing the ELF header into ehdr and program header into pdhr. Then we set up the signal handler which intercepts segmentation faults (SIGSEGV) which occurs when we try to access unallocated segments. This segmentation fault is treated as a page fault because page allocation hasn’t happened yet.

setup\_signal\_handler() calls handle\_fault() which calls load\_segment\_on\_demand().

Here we begin dynamically allocating pages as needed using mmap to only allocate memory of PAGE\_SIZE (4096 bytes). This ensures that only the required pages are allocated as needed, avoiding unnecessary memory consumption. Additionally, the function calculates internal fragmentation by determining the difference between the page size and the actual size of the allocated segment as specified in the program header.

We then perform cleanup of fd, ehdr, phdr, and and unmapping any dynamically allocated memory pages in mapped\_pages

**Contributions:**

Aakash – Implementation of code, Debugging

Parsh – ReadMe and Debugging