Contributions

1. Rishabh Jakhar (2023435): Implemented Launch.c, Makefile commands and error handling
2. Uday pandita (2023563): Implemented Loader.c and error handling

Simple Loader Implementation

Implenting the loader required the same steps mentioned in the starter pdf, that is:

a) Use the “open” system call to get the file descriptor for the input binary and “read” system call to read the content of the binary into a heap allocated memory of appropriate size. Use malloc to allocate space for copying content of the binary.

b) 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

c) Allocate memory of the size “p_memsz” using mmap function as shown below and then copy the segment content.

d) Navigate to the entrypoint address (e_entrypoint) into the segment loaded in the memory in above step. The entrypoint address may not be the starting address in that segment indicated by “p_vaddr”. You have to walk that segment to reach the virtual address referred by “e_entrypoint”.

e) Once you reach that location, simply typecast the address to that of function pointer matching “_start” method in fib.c.

f) Call the “_start” method and print the value returned from the “_start”.

Then the shared library was created dynamically.

- Loader.c was used to create the shared library “lib_simpleloader.c”
- Launch.c used the library to execute fib.c and display the output.

For running the code, we enter the following commands in terminal:

```
make
cd bin
./launch ../test/fib
```

We get our desired output and then run the clean command

```
make clean
```

Github repository Link:

https://github.com/Rishabh4Jakhar/SimpleLoader