MID-WAY IMPLEMENTATION DETAILS - AYUSH JAIN [jain207@purdue.edu]

Specific questions asked in lab7 for implementation

Q1. What have you been able to implement and get working up until now?

1) Successfully implemented parsing of ELF as per the ELF format details in Chapter 4 of System V specification link. Only 32-bit compatible struct have been declared as Xinu runs on x86 architecture.

The code is present in **elf_parser.c** using header **elf.h**. It is done using parse method taking file descriptor of opened ELF file as argument. This parse method is called in main function.
The steps include :

a) Parsing of ELF Header and also checking the first 4 bytes to see if it's a valid ELF format file. (read_elf_header function)
b) Reading section header table containing information about all the sections like symbol table, text section etc.
c) Printing all the symbols in the symbol table to see if symbols like main, kprintf are present for the helloworld object file.
d) Separating a section like text section using the information from the section header table.

2) Adding programs semdump.c and ls.c in programs directory (pseudocode) as they will be called using load_program system call in shell. This system call is called when the entered command in the shell don't match the predefined commands in shell.c

Q2. What do you plan on implementing before the final due date?

Everything mentioned in the lab7 requirement including loading and unloading the library in Xinu.

Q3. What problems have you run into while performing your implementation? What did you do to solve those problems?

Initially, I was copying the whole elf file in Xinu memory and then trying to parse it in a sequence using type casting explicitly. Since, the read function for rfs has a limit of 1024 bytes at one time, it became highly inconvenient that way.

Solution

Read the information about a particular module by seeking to a particular location and reading only specific amount of bytes. This was possible as the offset information is already present for each struct and offset is 0 for ELF header section. The type cast was directly specified in the read system call along with number of bytes which were to be read.

Upcoming Work

Resolving relocation entries for symbols and successfully loading the object file in Xinu.