Lab 4 Reading Material

You should read the task description as part of the reading material.

What is ELF?

ELF stands for "Executable and Linkable Format", used by the Linux operating system. An ELF file contains binary data specifying the program code in machine language, as well as instructions to the OS on where and how to load the file into memory, where the actual code starts, etc.

Technical Description

1. Read the ELF white paper: (See ELF format manual in Mandatory lecture), for lab4 you should read section 1: Object files (up-to and including symbol table)
2. Read the manual of the *readelf* utility (*man readelf*).
3. See ELF_format_file.png in Lab 4 Files, which illustrates the ELF object file format.
4. Examine the *elf.h* header file provided with linux installations (can be found at /usr/include/elf.h), and locate the definitions of structures for the ELF header, section headers, and symbol table (In future labs you will also include this header in your own code, and need to access these structures).
5. You should be able to answer the following questions regarding any ELF executable/object (similar questions will appear in the final exam):
    - Where in the file is the entry point specified, and what is its value?
    - How many sections are there in this file?
    - What is the size of .text section?
    - Does the symbol "_end" occur in the file? If so, what is its type and bind, where is it mapped to in virtual memory?
    - Does the symbol ".rodata" occur in the file? If so, what is its type and bind, where it is mapped to in virtual memory?
    - Where in the file does the code of function "main" start?
6. To get the file offset of a function, you first need to find out in which section it resides. Afterwards, the offset can be calculated from the following formula: section_file_offset + function_virtual_address - section_virtual_address. (WHY?? think about it and be prepared to explain.)

Additional material for Lab 4

In this lab you will be operating on files a lot. You will need to re-read on the system calls: open( ), read( ), write( ), lseek( ), and close( ). Note that lseek ( ) (see man 2 lseek) can be used to determine the size of the file using SEEK_END.