Introduction:

This is a technical manual of ELF object file editor, this manual will explain the operational processes.

Overview:

1. Read ELF header and print out information.
2. Read the corresponding section select by user.
3. Print section or search, edit, add, delete the corresponding byte/bytes.
4. Write back to object file or abort the result.

In <file_editor.c>:

short file_format_checker(unsigned char): Check if file is DOS or ELF

int main(int,char*){}:

In elf.h:

unsigned char elf_is_32_64(unsigned char*): check ELF type(32-bit or 64-bit)

In elf_32.h:

elf_32_header elf_32_header_reader(unsigned char *): Read elf header

elf_32_session_header elf_32_session_header_reader(char*,int): Read section header

short elf_32_e_type_adapter(elf_32_header): Get e_type from elf_32_header

void elf_32_e_type_reader(elf_32_header): Read e_type

short elf_32_e_machine_adapter(elf_32_header): Get e_machine from elf_32_header

void elf_32_e_machine_reader(elf_32_header): Read e_machine

char* elf_32_section_header_string_table_reader(elf_32_header, char): Read string_table

void opcode_position(FILE, unsigned char):
Read opcode from <FILE> (which are <one-byte_opcodes.txt> and <two-byte_opcodes.txt>).

char opcode_finder(unsigned char): Find corresponding 1-byte opcode from list.

char two_bytes_opcode_finder(unsigned char):Find corresponding 2-byte opcode from list.

void elf_32_session_editer(unsigned char*, elf_32_session_header, char, elf_32_header, int, char*):
Apply print, information, find, long find, edit, add, delete, add, disassembly(Z), save function.

In elf_share.h:

void elf_e_type_reader(short): Analyze e_type value.

void elf_e_machine_reader(short): Analyze e_machine value.

void is_little_endian(char): Check ELF type(32-bit or 64-bit)