Министерство цифрового развития, связи
и массовых коммуникаций Российской Федерации

Сибирский государственный университет
телекоммуникаций и информатики

Кафедра прикладной математики и кибернетики

ЛАБОРАТОРНАЯ РАБОТА №6

По дисциплине: «Операционные системы»

Выполнили:

Студенты 3 курса группы ИП-111
Корнилов А.А.,
Попов М.И.,
Толкач А.А.

Проверил:

Профессор кафедры ПМиК
Малков Е.А.

Новосибирск, 2023

Задание: получите список имён экспортируемых функций библиотеки совместного доступа, разработанной в лабораторной 5. Дополните программу лекции 6 алгоритмом поиска имён разделов.

Цель: знакомство со структурой ELF файлов.

Выполнение работы:

Для написания elf библиотеки сначала надо ознакомится с заголовками разделов файла общего доступа написанного в 5 лабораторной работе liblab05.so, для этого воспользуемся флагом -S

```
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B ВУЗ/OS/6$ readelf -S liblab05.so
Имеется 30 заголовков раздела, начиная со смещения 0x3958:

Заголовки разделов:
  [Нм] Имя              Тип              Адрес             Смещение
       Размер           Разм.Ent         Флаги  Ссылк Инфо  Выравн
  [ 0]                  NULL             0000000000000000  00000000
       0000000000000000 0000000000000000           0     0     0
  [ 1] .note.gnu.pr[...] NOTE            00000000000002a8  000002a8
       0000000000000020 0000000000000000   A       0     0     8
  [ 2] .note.gnu.bu[...] NOTE            00000000000002c8  000002c8
       0000000000000024 0000000000000000   A       0     0     4
  [ 3] .gnu.hash        GNU_HASH         00000000000002f0  000002f0
       0000000000000048 0000000000000000   A       4     0     8
  [ 4] .dynsym          DYNSYM           0000000000000338  00000338
       0000000000000258 0000000000000018   A       5     1     8
  [ 5] .dynstr          STRTAB           0000000000000590  00000590
       000000000000015f 0000000000000000   A       0     0     1
  [ 6] .gnu.version     VERSYM           00000000000006f0  000006f0
       0000000000000032 0000000000000002   A       4     0     2
  [ 7] .gnu.version_r   VERNEED          0000000000000728  00000728
       0000000000000040 0000000000000000   A       5     1     8
  [ 8] .rela.dyn        RELA             0000000000000768  00000768
       00000000000000d8 0000000000000018   A       4     0     8
  [ 9] .rela.plt        RELA             0000000000000840  00000840
       0000000000000108 0000000000000018  AI       4    23     8
  [10] .init            PROGBITS         0000000000001000  00001000
       000000000000001b 0000000000000000  AX       0     0     4
  [11] .plt             PROGBITS         0000000000001020  00001020
       00000000000000c0 0000000000000010  AX       0     0     16
  [12] .plt.got         PROGBITS         00000000000010e0  000010e0
       0000000000000010 0000000000000010  AX       0     0     16
```

```
  [13] .plt.sec          PROGBITS         00000000000010f0  000010f0
       00000000000000b0  0000000000000010  AX       0     0     16
  [14] .text             PROGBITS         00000000000011a0  000011a0
       0000000000000aca  0000000000000000  AX       0     0     16
  [15] .fini             PROGBITS         0000000000001c6c  00001c6c
       000000000000000d  0000000000000000  AX       0     0     4
  [16] .rodata           PROGBITS         0000000000002000  00002000
       0000000000000610  0000000000000000   A       0     0     8
  [17] .eh_frame_hdr     PROGBITS         0000000000002610  00002610
       000000000000005c  0000000000000000   A       0     0     4
  [18] .eh_frame         PROGBITS         0000000000002670  00002670
       0000000000000164  0000000000000000   A       0     0     8
  [19] .init_array       INIT_ARRAY       0000000000003e00  00002e00
       0000000000000008  0000000000000008  WA       0     0     8
  [20] .fini_array       FINI_ARRAY       0000000000003e08  00002e08
       0000000000000008  0000000000000008  WA       0     0     8
  [21] .dynamic          DYNAMIC          0000000000003e10  00002e10
       00000000000001c0  0000000000000010  WA       5     0     8
  [22] .got              PROGBITS         0000000000003fd0  00002fd0
       0000000000000030  0000000000000008  WA       0     0     8
  [23] .got.plt          PROGBITS         0000000000004000  00003000
       0000000000000070  0000000000000008  WA       0     0     8
  [24] .data             PROGBITS         0000000000004070  00003070
       0000000000000008  0000000000000000  WA       0     0     8
  [25] .bss              NOBITS           0000000000004080  00003078
       0000000000003d18  0000000000000000  WA       0     0     32
  [26] .comment          PROGBITS         0000000000000000  00003078
       000000000000002b  0000000000000001  MS       0     0     1
  [27] .symtab           SYMTAB           0000000000000000  000030a8
       00000000000004b0  0000000000000018          28    26     8
  [28] .strtab           STRTAB           0000000000000000  00003558
       00000000000002f2  0000000000000000           0     0     1
  [29] .shstrtab         STRTAB           0000000000000000  0000384a
       000000000000010d  0000000000000000           0     0     1
```

Необходимо вывести раздел .shstrtab, который содержит имена основных разделов. Используя структуру Elf64_Ehdr ищем в ней раздел с индексом e_shstrndx и считываем содержимое раздела в Elf64_Shdr. Добавляем вывод только наших функций библиотеки, делая проверку по полю st_shndx (лис.1).

```c
#include <elf.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main(){
   const char* elfFile = "liblab05.so";
```

```
    int i,j;
    char a, sname[32];
  Elf64_Ehdr header;
  Elf64_Shdr sheader, symtab, strtab, shstrtab;
  Elf64_Sym sym;

  FILE* file = fopen(elfFile, "rb");
  fread(&header, sizeof(header), 1, file);
  fseek(file, header.e_shoff, SEEK_SET);
  for(i=0; i < header.e_shnum; i++){
     fseek(file, header.e_shoff + header.e_shentsize * i, SEEK_SET);
     fread(&sheader, sizeof(sheader), 1, file);
     if(i == 4)symtab = (Elf64_Shdr)sheader;
     if(i == 5)strtab = (Elf64_Shdr)sheader;
     if (i == header.e_shstrndx) shstrtab = (Elf64_Shdr)sheader;
  }
    fprintf(stdout, "%s\t%s\t%s\t%s\t%s\t%s\n",""№ ",
"st_size","ST_TYPE","ST_BIND","st_shndx","sname");
  for(i=0; i < symtab.sh_size / symtab.sh_entsize; i++){
     fseek(file, symtab.sh_offset + symtab.sh_entsize * i, SEEK_SET);
     fread(&sym, sizeof(Elf64_Sym), 1, file);
     fseek(file, strtab.sh_offset + sym.st_name, SEEK_SET);
     fread(sname, 1,32, file);
        if (sym.st_shndx != 0) fprintf(stdout, "%d\t%ld\t%u\t%u\t%hd\t%s\n", i,
sym.st_size, ELF64_ST_TYPE(sym.st_info), ELF64_ST_BIND(sym.st_info),
                sym.st_shndx, sname);
  }
  j = 0;
  for(i=0; i < shstrtab.sh_size; i++){
     fseek(file, shstrtab.sh_offset + i, SEEK_SET);
     fread(&a, sizeof(char), 1, file);
     if (a == '\0'){
        fprintf(stdout, "\n #%d - ", j);
        j++;
     }
     else fprintf(stdout, "%c", a);
  }
  return 0;
}
```

Листинг 1 – lab06_3.c

Команда для компиляции и запуск программы:

```
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B ВУ3/OS/6$ gcc lab06_3.c -o lab06_3
miron@DESKTOP-UMC1Q46:/mnt/u/Documents/B ВУ3/OS/6$ ./lab06_3
№       st_size ST_TYPE ST_BIND st_shndx        sname
16      240     2       1       14      exportDatabaseToFile
17      464     2       1       14      deleteRecord
18      388     2       1       14      searchRecord
19      466     2       1       14      editRecord
20      4       1       1       25      databaseSize
21      499     2       1       14      addRecord
22      269     2       1       14      importDatabaseFromFile
23      251     2       1       14      viewAllRecords
24      15600   1       1       25      database

 #0 - .symtab
```

```
#1 - .strtab
#2 - .shstrtab
#3 - .note.gnu.property
#4 - .note.gnu.build-id
#5 - .gnu.hash
#6 - .dynsym
#7 - .dynstr
#8 - .gnu.version
#9 - .gnu.version_r
#10 - .rela.dyn
#11 - .rela.plt
#12 - .init
#13 - .plt.got
#14 - .plt.sec
#15 - .text
#16 - .fini
#17 - .rodata
#18 - .eh_frame_hdr
#19 - .eh_frame
#20 - .init_array
#21 - .fini_array
#22 - .dynamic
#23 - .got.plt
#24 - .data
#25 - .bss
#26 - .comment
```