

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки і комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

Звіт
про виконання лабораторної роботи № 9
“Застосування динамічних бібліотек в ОС Windows і Linux”

Виконав
студент групи ФЕІ-23
Дзюрах І.М .
Перевірив
ас. Сінькевич О. О.

Львів - 2019

Завдання

Реалізуйте застосування для Linux і Windows, що може бути розширене під час виконання. Інтерфейс модуля розширення функцій типу **void** без параметрів. Після запуску застосування видає на екран підказку й очікує введення команди з клавіатури. Можливі такі команди: **load ім'я_модуля** (завантаження модуля в пам'ять), **unload ім'я_модуля** (вилучення модуля з пам'яті) **call ім'я_функції** (виклик функції з модуля). Кожен модуль розширення повинен містити код, який виконується під час його завантаження в пам'ять і вилучення з пам'яті.

Реалізація

soul.c

```
#include <stdio.h>
#include <dlfcn.h>

int main(){
    int var;
    char input[100];
    void* library = NULL;
    double (*func)();
    char* mess;
    while(1){
        fprintf(stdout, "\n1. load\n");
        fprintf(stdout, "2. call\n");
        fprintf(stdout, "3. unload\n");
        fprintf(stdout, "4. exit\n");
        scanf("%i", &var);
        switch(var)
        {
            case 1:{
                fprintf(stdout, "Enter the path: ");
                scanf("%s", input);
                library = dlopen(input, RTLD_NOW);
                if(library)
                {
                    fprintf(stdout, "Done! The library is loaded!\n");
                }
                else
                {
                    fprintf(stdout, "%s\n", dlerror());
                }
                break;
            }
            case 2:{
                if(library != NULL)
                {
                    fprintf(stdout, "Enter a function name: ");
                    scanf("%s", input);
```

```

        func = dlsym(library, input);
        if ((mess = dlerror()) == NULL)
        {
            fprintf(stdout, "Function %s:\n", input);
            (*func)();
        }
        else
        {
            fprintf(stderr, "%s\n", mess);
        }
    }
    else{
        fprintf(stdout, "Library not loaded!\n");
    }
    break;
}
case 3:{
    if(library != NULL)
    {
        dlclose(library);
        fprintf(stdout, "Done! The library is unloaded!\n");
        library = NULL;
    }
    else
    {
        fprintf(stdout, "Library not loaded!\n");
    }
    break;
}
case 4:{
    return 0;
}
}
}
}

```

library1.c

```
#include <stdio.h>
```

```

void func(){
    fprintf(stdout, "This function belongs to the library1\n");
}

```

library2.c

```
#include <stdio.h>
```

```

void func(){
    fprintf(stdout, "This function belongs to the library2\n");
}

```

Makefile

```
all: library_1 library_2 soul clean run

library_1:
    gcc -fpic -c library1.c -o library1.o
    gcc -shared library1.o -o library1.so

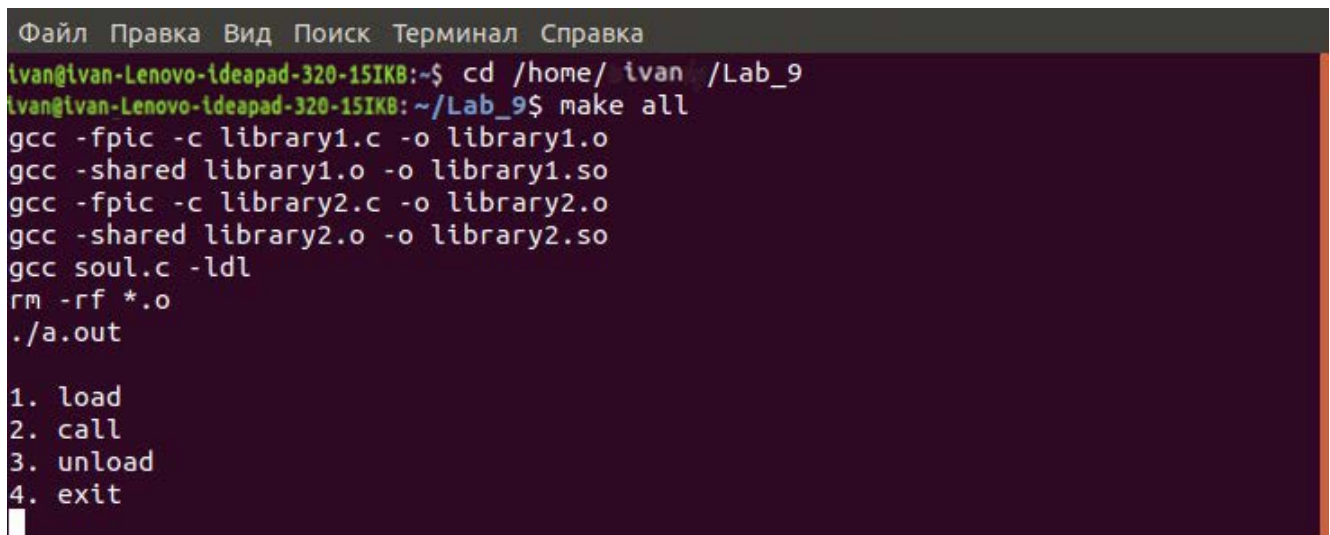
library_2:
    gcc -fpic -c library2.c -o library2.o
    gcc -shared library2.o -o library2.so

soul:
    gcc soul.c -ldl

clean:
    rm -rf *.o

run:
    ./a.out
```

Тестування



```
Файл  Правка  Вид  Поиск  Терминал  Справка
ivan@ivan-Lenovo-Ideapad-320-15IK8:~$ cd /home/ivan /Lab_9
ivan@ivan-Lenovo-Ideapad-320-15IK8:~/Lab_9$ make all
gcc -fpic -c library1.c -o library1.o
gcc -shared library1.o -o library1.so
gcc -fpic -c library2.c -o library2.o
gcc -shared library2.o -o library2.so
gcc soul.c -ldl
rm -rf *.o
./a.out

1. load
2. call
3. unload
4. exit
```

Файл Правка Вид Поиск Терминал Справка

1. load
2. call
3. unload
4. exit

1

Enter the path: /home/andrew/Lab_9/library1.so

Done! The library is loaded!

1. load
2. call
3. unload
4. exit

2

Enter a function name: func

Function func:

This function belongs to the library1

1. load
2. call
3. unload
4. exit

3

Done! The library is unloaded!

1. load
2. call
3. unload
4. exit

```
Файл Правка Вид Поиск Терминал Справка
1. load
2. call
3. unload
4. exit
1
Enter the path: /home/andrew/Lab_9/library2.so
Done! The library is loaded!

1. load
2. call
3. unload
4. exit
2
Enter a function name: func
Function func:
This function belongs to the library2

1. load
2. call
3. unload
4. exit
3
Done! The library is unloaded!

1. load
2. call
3. unload
4. exit
```

Висновок

Виконавши дану лабораторну роботу, я навчився створювати і застосовувати динамічні бібліотеки в ОС Windows і Linux. Реалізував застосування для Linux і Windows, що може бути розширене під час виконання.