

**Московский авиационный институт  
(национальный исследовательский университет)**

**Факультет информационных технологий и прикладной  
математики**

**Кафедра вычислительной математики и программирования**

**Лабораторная работа №5 по курсу «Операционные системы»**

Студент: А. О. Дубинин  
Преподаватель: Е. С. Миронов  
Группа: М8О-206Б  
Дата:  
Оценка:  
Подпись:

**Москва, 2018**

## Лабораторная работа №5

**Цель работы:** Приобретение практических навыков в создании динамических библиотек, создании программ, которые используют функции динамических библиотек.

**Цель работы:** Требуется создать динамическую библиотеку, которая реализует определенный функционал. Далее использовать данную библиотеку 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)
2. Во время исполнения программы, подгрузив библиотеку в память с помощью системных вызовов

В конечном итоге, программа должна состоять из следующих частей: Динамическая библиотека, реализующая заданных вариантом интерфейс; Тестовая программа, которая использует библиотеку, используя знания полученные на этапе компиляции; Тестовая программа, которая использует библиотеку, используя только местоположение динамической библиотеки и ее интерфейс. Провести анализ между обоими типами использования библиотеки.

**Вариант 25 (4-4):**

4. Работа с бинарным деревом поиска
4. Md5 суммы

# 1 Описание

Использовалась библиотека для бинарного дерева поиска. В её интерфейс входит вставка в дерево печать дерева, удаление узла дерева, поиск элемента в дереве. Для линковки во время компиляции указываем путь до библиотеки и указываем название библиотеки, пользуемся функциями как обычно. В случае использования библиотеки во время исполнения нужно явно открыть библиотеку с помощью `dlopen()`, а затем присваивать указателям на функции результат `dlsym()`, который ищет по названию функции в библиотеке. Использованные системные вызовы:

`void *dlopen(const char *filename, int flag);` – открывает файл по пути `filename` со свойствами `flag`. Если библиотека имеет зависимости, то они также подключаются с теми же свойствами. В случае ошибки возвращает `NULL`. `flag` обязательно должен иметь либо `RTLD LAZY`, либо `RTLD NOW`, которые отвечают за загрузку библиотеки (ленивую, когда потребуется код в первый раз, либо целиком).

`char *dlerror(void);` – возвращает строку, которая описывает ошибку. Если ошибки не было, то возвращает `NULL`.

`void *dlsym(void *handle, const char *symbol);` – ищет в дереве подключенных через `dlopen()` библиотек строку `symbol`, если находит, то возвращает `void *` участок памяти, связанный с функцией. В случае ошибки возвращает `NULL` и устанавливает ошибку для `dlerror()`.

## 2 Исходный код

### Tree.h

```
1 //
2 // Tree.h
3 // 03.12.18
4 //
5
6 #ifndef LAB_5_TREE_H
7 #define LAB_5_TREE_H
8
9 typedef struct tree {
10     char* value;
11     struct tree *parent;
12     struct tree *left;
13     struct tree *right;
14 } tree;
15
16 tree *search(tree *parent, char* value);
17 tree *min(tree *parent);
18 tree *max(tree *parent);
19
20 void printTree(tree *root, int tabs);
21 void traverse(tree *parent);
22 void insert(tree **root, char* value, tree *parent);
23 tree * _delete(tree *t, char* x);
24
25 int get_int(char *prompt);
26 char* get_key(char *prompt);
27 void flush_stdin(void);
28
29 #endif //LAB_5_TREE_H
```

### mainDyn.c

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include "Tree.h"
4 #include <dlfcn.h>
5
6
7 int main(int argc, char *argv[]) {
8
9     tree* (*search)(tree *parent, char* value);
10     tree* (*min)(tree *parent);
11     tree* (*max)(tree *parent);
12     void (*printTree)(tree *root, int tabs);
13     void (*traverse)(tree *parent);
14     void (*insert)(tree **root, char* value, tree *parent);
```

```

15 tree* (*_delete)(tree *t, char* x);
16 int (*get_int)(char *prompt);
17 char* (*get_key)(char *prompt);
18 void (*flush_stdin)(void);
19
20 char *err;
21 void *libHandle;
22 libHandle = dlopen("libtree.so", RTLD_LAZY);
23 if (!libHandle) {
24     fprintf(stderr, "%s\n", dlerror());
25     exit(1);
26 }
27
28 search = dlsym(libHandle, "search");
29 min = dlsym(libHandle, "min");
30 max = dlsym(libHandle, "max");
31 printTree = dlsym(libHandle, "printTree");
32 traverse = dlsym(libHandle, "traverse");
33 insert = dlsym(libHandle, "insert");
34 _delete = dlsym(libHandle, "_delete");
35 get_int = dlsym(libHandle, "get_int");
36 flush_stdin = dlsym(libHandle, "flush_stdin");
37 get_key = dlsym(libHandle, "get_key");
38
39
40 int cmd = 0;
41 char* value;
42 tree *root = NULL;
43
44 do {
45     printf("Commands\n\n"
46           "1 - Insert\n"
47           "2 - Search\n"
48           "3 - Delete\n"
49           "4 - Print tree\n"
50           "0 - Quit\n\n");
51
52     cmd = get_int("Command");
53
54     switch (cmd) {
55     case 1:
56         value = get_key("value to insert");
57         insert(&root, value, NULL);
58         break;
59     case 2:
60         value = get_key("value to find");
61         if (search(root, value) == NULL) {
62             printf("%s is not found in tree!\n", value);
63         } else {

```

```

64         printf("%s is found in tree!\n", value);
65     }
66     break;
67 case 3:
68     value = get_key("value to delete");
69     root = _delete(root, value);
70     break;
71 case 4:
72     printf("tree:\n");
73     printTree(root, 0);
74     break;
75 }
76
77 } while (cmd != 0);
78
79 return 0;
80 }

```

## Makefile

```

1 CC = gcc
2 FLAGS = -std=c99 -w -Werror -Wall -pedantic
3
4 all: run
5
6 run: libtree.so mainStat.o mainDyn.o
7     $(CC) $(FLAGS) -o stat mainStat.o -L. -ltree -Wl,-rpath,.
8     $(CC) $(FLAGS) -o dyn mainDyn.o -ldl -Wl,-rpath,.
9
10 mainStat.o: mainStat.c
11     $(CC) -c $(FLAGS) mainStat.c
12
13 mainDyn.o: mainDyn.c
14     $(CC) -c $(FLAGS) mainDyn.c
15
16 libtree.so: Tree.o
17     $(CC) $(FLAGS) -shared -o libtree.so Tree.o
18
19 Tree.o: Tree.c
20     $(CC) -c -fPIC $(FLAGS) Tree.c
21
22 clean:
23     rm -f *.o stat dyn *.so *.a

```

### 3 Тесты

#### Статическая

```
art@mars:~/workdir/OS/lab_5$ make
gcc -c -fPIC -std=c99 -w -Werror -Wall -pedantic Tree.c
gcc -std=c99 -w -Werror -Wall -pedantic -shared -o libtree.so Tree.o
gcc -c -std=c99 -w -Werror -Wall -pedantic mainStat.c
gcc -c -std=c99 -w -Werror -Wall -pedantic mainDyn.c
gcc -std=c99 -w -Werror -Wall -pedantic -o stat mainStat.o -L. -ltree -Wl,-rpath,.
gcc -std=c99 -w -Werror -Wall -pedantic -o dyn mainDyn.o -ldl -Wl,-rpath,.
art@mars:~/workdir/OS/lab_5$ ./stat
Commands
```

```
1 -Insert
2 -Search
3 -Delete
4 -Print tree
0 -Quit
```

```
Command:
1
value to insert:
5
Commands
```

```
1 -Insert
2 -Search
3 -Delete
4 -Print tree
0 -Quit
```

```
Command:
1
value to insert:
3
Commands
```

```
1 -Insert
2 -Search
3 -Delete
```

```
4 -Print tree
0 -Quit
```

Command:

```
4
```

tree:

```
5
```

```
3
```

Commands

```
1 -Insert
```

```
2 -Search
```

```
3 -Delete
```

```
4 -Print tree
```

```
0 -Quit
```

Command:

```
2
```

value to find:

```
3
```

3 is found in tree!

Commands

```
1 -Insert
```

```
2 -Search
```

```
3 -Delete
```

```
4 -Print tree
```

```
0 -Quit
```

Command:

```
3
```

value to delete:

```
3
```

Commands

```
1 -Insert
```

```
2 -Search
```

```
3 -Delete
```

```
4 -Print tree
```

```
0 -Quit
```



Command:

4

tree:

5

Commands

1 -Insert

2 -Search

3 -Delete

4 -Print tree

0 -Quit

Command:

0

art@mars:~/workdir/OS/lab\_5\$

### Динамическая

art@mars:~/workdir/OS/lab\_\$ ./dyn

Commands

1 -Insert

2 -Search

3 -Delete

4 -Print tree

0 -Quit

Command:

1

value to insert:

2

Commands

1 -Insert

2 -Search

3 -Delete

4 -Print tree

0 -Quit

Command:

1

value to insert:

5

Commands

1 -Insert

2 -Search

3 -Delete

4 -Print tree

0 -Quit

Command:

4

tree:

5

2

Commands

1 -Insert

2 -Search

3 -Delete

4 -Print tree

0 -Quit

Command:

1

value to insert:

0

Commands

1 -Insert

2 -Search

3 -Delete

4 -Print tree

0 -Quit

Command:

2

value to find:

7

7 is not found in tree!

Commands

```
1 -Insert
2 -Search
3 -Delete
4 -Print tree
0 -Quit
```

Command:

```
0
```

```
art@mars:~/workdir/OS/lab_5$
```

## 4 Диагностика

Dynamic:

```
art@mars:~/workdir/OS/lab_5$ strace ./dyn
execve("./dyn",["./dyn"],0x7ffe93271c40 /* 56 vars */) = 0
brk(NULL)                                = 0x55fa5be33000
access("/etc/ld.so.nohwcap",F_OK)        = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload",R_OK)        = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"./tls/haswell/x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD,"./tls/haswell/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./tls/x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./haswell/x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./haswell/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file
or directory)
openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC) = 3
fstat(3,st_mode=S_IFREG|0644,st_size=105825,...) = 0
mmap(NULL,105825,PROT_READ,MAP_PRIVATE,3,0) = 0x7fde666a5000
close(3)                                = 0
access("/etc/ld.so.nohwcap",F_OK)        = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libdl.so.2",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>P6"... ,832) = 832
fstat(3,st_mode=S_IFREG|0644,st_size=14560,...) = 0
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fde666a3000
mmap(NULL,2109712,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fde66294000
mprotect(0x7fde66297000,2093056,PROT_NONE) = 0
mmap(0x7fde66496000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7fde66496000
close(3)                                = 0
openat(AT_FDCWD,"./tls/haswell/x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
```

```

openat(AT_FDCWD,"./tls/haswell/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./haswell/x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./haswell/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file
or directory)
access("/etc/ld.so.nohwcap",F_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>604"... ,832) = 832
fstat(3,st_mode=S_IFREG|0755,st_size=2030544,...) = 0
mmap(NULL,4131552,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fde65ea3000
mprotect(0x7fde6608a000,2097152,PROT_NONE) = 0
mmap(0x7fde6628a000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7fde6628a000
mmap(0x7fde66290000,15072,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1,0)
= 0x7fde66290000
close(3) = 0
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fde666a0000
arch_prctl(ARCH_SET_FS,0x7fde666a0740) = 0
mprotect(0x7fde6628a000,16384,PROT_READ) = 0
mprotect(0x7fde66496000,4096,PROT_READ) = 0
mprotect(0x55fa5a289000,4096,PROT_READ) = 0
mprotect(0x7fde666bf000,4096,PROT_READ) = 0
munmap(0x7fde666a5000,105825) = 0
openat(AT_FDCWD,"./tls/haswell/x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD,"./tls/haswell/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./tls/x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./haswell/x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT

```

```

(No such file or directory)
openat(AT_FDCWD,"./haswell/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./libtree.so",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>60"... ,832) = 832
brk(NULL) = 0x55fa5be33000
brk(0x55fa5be54000) = 0x55fa5be54000
fstat(3,st_mode=S_IFREG|0755,st_size=12920,...) = 0
getcwd("/home/art/study/semester_3/OS/lab_5",128) = 36
mmap(NULL,2105528,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fde65ca0000
mprotect(0x7fde65ca2000,2093056,PROT_NONE) = 0
mmap(0x7fde65ea1000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7fde65ea1000
close(3) = 0
mprotect(0x7fde65ea1000,4096,PROT_READ) = 0
fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(136,0),...) = 0
write(1,"Commands",9Commands
) = 9
write(1,"",1
) = 1
write(1,"1 -Insert",111 -Insert
) = 11
write(1,"2 -Search",112 -Search
) = 11
write(1,"3 -Delete",113 -Delete
) = 11
write(1,"4 -Print tree",154 -Print tree
) = 15
write(1,"0 -Quit",90 -Quit
) = 9
write(1,"",1
) = 1
write(1,"Command: ",10Command:
) = 10
fstat(0,st_mode=S_IFCHR|0620,st_rdev=makedev(136,0),...) = 0
read(0,0
"0",1024) = 2
exit_group(0) = ?
+++ exited with 0 +++

```

## 5 Выводы

Выполнив лабораторную работу, я приобрёл практические навыки в создании динамических библиотек, создании программ, которые используют функции динамических библиотек. Статическая линковка при компиляции упаковывает библиотеку в исполняемый файл, таким образом гарантируется переносимость программы. Негативные стороны этого таковы: увеличение исполняемого файла в размерах и необходимость обновлять весь файл при апдейте библиотеки. При динамической линковке библиотека подключается во время исполнения. Различные копии программы и другие программы, будут использовать одну общую библиотеку, это позволяет экономить оперативную память, но данная библиотека уже должна быть установлена на используемой машине, это несколько затрудняет распространение данной библиотеки.