Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа**

**по курсу «Операционные системы»**

**III Семестр**

**Задание 5**

**Вариант 17**

|  |  |
| --- | --- |
| Студент: | Анисимов В.А. |
| Группа: | М80-208Б-18 |
| Преподаватель: | Миронов Е.С |
|  |  |
| Оценка: |  |
| Дата: |  |

Москва 2019

**1. Описание задания**

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 используют 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 17:

Работа с деком; тип данных – строки длинной от 0 до 255 символов.

**2. Код программы**

mainC.c

#include <stdlib.h>

#include <stdio.h>

#include "str\_deque.h"

extern str\_deque\* create();

extern void push\_back(str\_deque\* deque, char\* str);

extern void pop\_back(str\_deque\* deque);

extern void push\_front(str\_deque\* deque, char\* str);

extern void pop\_front(str\_deque\* deque);

extern void print(str\_deque\* deque);

extern void destroy(str\_deque\* deque);

int main(){

str\_deque\* deque = create();

pop\_back(deque);

print(deque);

push\_back(deque, "строка 1");

print(deque);

push\_back(deque, "строка 2");

push\_front(deque, "строка 3");

print(deque);

pop\_front(deque);

pop\_back(deque);

print(deque);

pop\_back(deque);

print(deque);

destroy(deque);

return 0;

}

mainR.c

#include <stdlib.h>

#include <stdio.h>

#include <dlfcn.h>

#include "str\_deque.h"

int main(){

void \*lib = dlopen("./str\_deque.so", RTLD\_LAZY);

if (!lib) {

printf("dlopen error\n");

return -1;

}

str\_deque\*(\*create)() = dlsym(lib, "create");

\_Bool(\*empty)(str\_deque\* deque) = dlsym(lib, "empty");

void(\*push\_back)(str\_deque\* deque, char\* str) = dlsym(lib, "push\_back");

void(\*pop\_back)(str\_deque\* deque) = dlsym(lib, "pop\_back");

void(\*push\_front)(str\_deque\* deque, char\* str) = dlsym(lib, "push\_front");

void(\*pop\_front)(str\_deque\* deque) = dlsym(lib, "pop\_front");

void(\*print)(str\_deque\* deque) = dlsym(lib, "print");

void(\*destroy)(str\_deque\* deque) = dlsym(lib, "destroy");

if(create == NULL || empty == NULL || push\_back == NULL || pop\_back == NULL ||

push\_front == NULL || pop\_front == NULL || print == NULL || destroy == NULL){

printf("dlsym error\n");

};

str\_deque\* deque = (create)();

(\*pop\_back)(deque);

(\*print)(deque);

(\*push\_back)(deque, "строка 1");

(\*print)(deque);

(\*push\_back)(deque, "строка 2");

(\*push\_front)(deque, "строка 3");

(\*print)(deque);

(\*pop\_front)(deque);

(\*pop\_back)(deque);

(\*print)(deque);

(\*pop\_back)(deque);

(\*print)(deque);

(\*destroy)(deque);

dlclose(lib);

return 0;

}

str\_deque.h

#ifndef STR\_DEQUE\_H

#define STR\_DEQUE\_H

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

typedef struct node{

struct node\* next;

struct node\* prev;

char\* data;

} node;

typedef struct str\_deque {

node\* Head;

node\* Tail;

size\_t size;

} str\_deque;

#endif

str\_deque.c

#include "str\_deque.h"

str\_deque\* create(){

str\_deque\* deque = (str\_deque\*)malloc(sizeof(str\_deque));

deque->Head = NULL;

deque->Tail = NULL;

deque->size = 0;

return deque;

}

\_Bool empty(str\_deque\* deque){

if(deque->size == 0)

return 1;

else

return 0;

}

void push\_back(str\_deque\* deque, char\* str){

node\* newnode = (node\*)malloc(sizeof(node));

newnode->next = NULL;

newnode->data = str;

if(empty(deque)){

deque->Head = newnode;

deque->Tail = newnode;

newnode->prev = NULL;

deque->size++;

return;

}

deque->Tail->next = newnode;

newnode->prev = deque->Tail;

deque->Tail = newnode;

deque->size++;

}

void pop\_back(str\_deque\* deque){

if (empty(deque)){

printf("Error. Deque is empty.\n");

return;

}

if(deque->size == 1){

free(deque->Tail);

deque->Head = NULL;

deque->Tail = NULL;

deque->size --;

return;

}

node\* tmp;

tmp = deque->Tail->prev;

deque->Tail = tmp;

free(tmp->next);

tmp->next = NULL;

deque->size--;

}

void push\_front(str\_deque\* deque, char\* str){

node\* newnode = (node\*)malloc(sizeof(node));

newnode->prev = NULL;

newnode->data = str;

if(empty(deque)){

deque->Head = newnode;

deque->Tail = newnode;

newnode->next = NULL;

deque->size++;

return;

}

deque->Head->prev = newnode;

newnode->next = deque->Head;

deque->Head = newnode;

deque->size++;

}

void pop\_front(str\_deque\* deque){

if (empty(deque)){

printf("Error. Deque is empty.\n");

return;

}

if(deque->size == 1) {

free(deque->Tail);

deque->Head = NULL;

deque->Tail = NULL;

deque->size--;

return;

}

node\* tmp;

tmp = deque->Head->next;

deque->Head = tmp;

free(tmp->prev);

tmp->prev = NULL;

deque->size--;

}

void print\_node(node\* node){

printf("%s\n", node->data);

if(node->next == NULL){

return;

}

print\_node(node->next);

}

void print(str\_deque\* deque){

if (empty(deque)){

printf("Error. Deque is empty.\n");

return;

}

printf("----------\n");

print\_node(deque->Head);

printf("++++++++++\n");

}

void destroy(str\_deque\* deque) {

while(!empty(deque)){

pop\_back(deque);

}

free(deque);

deque = NULL;

}

**3. Протокол работы программы**

walien@PC-name:~/2kurs/OS/lab5$ ./mainC

Error. Deque is empty.

Error. Deque is empty.

----------

строка 1

++++++++++

----------

строка 3

строка 1

строка 2

++++++++++

----------

строка 1

++++++++++

Error. Deque is empty.

walien@PC-name:~/2kurs/OS/lab5$ ./mainR

Error. Deque is empty.

Error. Deque is empty.

----------

строка 1

++++++++++

----------

строка 3

строка 1

строка 2

++++++++++

----------

строка 1

++++++++++

Error. Deque is empty.

**4. Общие сведения о программе**

В файле str\_deque.с осуществлена реализация коллекции, объявленной в str\_deque.h. В файлах mainС.c и mainR.c находятся реализации программ с разным типом использования динамической библиотеки: используя знания, полученные на этапе компиляции и используя только расположение динамической библиотеки и её интерфейс соответственно.

**Вывод**

Динамическая библиотека состоит из подпрограмм, подгружающихся в основную во время её выполнения. При компиляции программы такая библиотека не становится частью исполняемого файла, а остаётся отдельным модулем.

Использование динамических библиотек позволяет избавиться от дублирования кода и сократить размер программ, как на диске, так и в оперативной памяти, при условии, что несколько программ пользуются данной библиотекой. Так же, иногда выгодно вынести в динамическую библиотеку редко-используемые функции, чтобы они зря не «съедали» память.

Также, при использовании динамических библиотек, можно выделить возможность обновления такой библиотеки без перекомпиляции всех, использующих её файлов.

**Strace**

**mainC**

walien@PC-name:~/2kurs/OS/lab5$ strace ./mainC

execve("./mainC", ["./mainC"], 0x7ffd37191a80 /\* 63 vars \*/) = 0

brk(NULL) = 0x55e06ffda000

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, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=83168, ...}) = 0

mmap(NULL, 83168, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fc7da908000

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/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260\34\2\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2030544, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fc7da906000

mmap(NULL, 4131552, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fc7da305000

mprotect(0x7fc7da4ec000, 2097152, PROT\_NONE) = 0

mmap(0x7fc7da6ec000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fc7da6ec000

mmap(0x7fc7da6f2000, 15072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fc7da6f2000

close(3) = 0

arch\_prctl(ARCH\_SET\_FS, 0x7fc7da9074c0) = 0

mprotect(0x7fc7da6ec000, 16384, PROT\_READ) = 0

mprotect(0x55e06fae7000, 4096, PROT\_READ) = 0

mprotect(0x7fc7da91d000, 4096, PROT\_READ) = 0

munmap(0x7fc7da908000, 83168) = 0

brk(NULL) = 0x55e06ffda000

brk(0x55e06fffb000) = 0x55e06fffb000

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(136, 0), ...}) = 0

write(1, "Error. Deque is empty.\n", 23Error. Deque is empty.

) = 23

write(1, "Error. Deque is empty.\n", 23Error. Deque is empty.

) = 23

write(1, "----------\n", 11----------

) = 11

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 1\n", 15строка 1

) = 15

write(1, "++++++++++\n", 11++++++++++

) = 11

write(1, "----------\n", 11----------

) = 11

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 3\n", 15строка 3

) = 15

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 1\n", 15строка 1

) = 15

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 2\n", 15строка 2

) = 15

write(1, "++++++++++\n", 11++++++++++

) = 11

write(1, "----------\n", 11----------

) = 11

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 1\n", 15строка 1

) = 15

write(1, "++++++++++\n", 11++++++++++

) = 11

write(1, "Error. Deque is empty.\n", 23Error. Deque is empty.

) = 23

exit\_group(0) = ?

+++ exited with 0 +++

**mainR**

walien@PC-name:~/2kurs/OS/lab5$ strace ./mainR

execve("./mainR", ["./mainR"], 0x7ffceba9e870 /\* 63 vars \*/) = 0

brk(NULL) = 0x555be9870000

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, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=83168, ...}) = 0

mmap(NULL, 83168, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fe53e55a000

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, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\16\0\0\0\0\0\0"..., 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) = 0x7fe53e558000

mmap(NULL, 2109712, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe53e144000

mprotect(0x7fe53e147000, 2093056, PROT\_NONE) = 0

mmap(0x7fe53e346000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fe53e346000

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/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260\34\2\0\0\0\0\0"..., 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) = 0x7fe53dd53000

mprotect(0x7fe53df3a000, 2097152, PROT\_NONE) = 0

mmap(0x7fe53e13a000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fe53e13a000

mmap(0x7fe53e140000, 15072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe53e140000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe53e555000

arch\_prctl(ARCH\_SET\_FS, 0x7fe53e555740) = 0

mprotect(0x7fe53e13a000, 16384, PROT\_READ) = 0

mprotect(0x7fe53e346000, 4096, PROT\_READ) = 0

mprotect(0x555be7ea5000, 4096, PROT\_READ) = 0

mprotect(0x7fe53e56f000, 4096, PROT\_READ) = 0

munmap(0x7fe53e55a000, 83168) = 0

brk(NULL) = 0x555be9870000

brk(0x555be9891000) = 0x555be9891000

openat(AT\_FDCWD, "./str\_deque.so", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200\7\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=12392, ...}) = 0

getcwd("/home/walien/2kurs/OS/lab5", 128) = 27

mmap(NULL, 2105432, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe53db50000

mprotect(0x7fe53db51000, 2097152, PROT\_NONE) = 0

mmap(0x7fe53dd51000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7fe53dd51000

close(3) = 0

mprotect(0x7fe53dd51000, 4096, PROT\_READ) = 0

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(136, 0), ...}) = 0

write(1, "Error. Deque is empty.\n", 23Error. Deque is empty.

) = 23

write(1, "Error. Deque is empty.\n", 23Error. Deque is empty.

) = 23

write(1, "----------\n", 11----------

) = 11

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 1\n", 15строка 1

) = 15

write(1, "++++++++++\n", 11++++++++++

) = 11

write(1, "----------\n", 11----------

) = 11

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 3\n", 15строка 3

) = 15

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 1\n", 15строка 1

) = 15

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 2\n", 15строка 2

) = 15

write(1, "++++++++++\n", 11++++++++++

) = 11

write(1, "----------\n", 11----------

) = 11

write(1, "\321\201\321\202\321\200\320\276\320\272\320\260 1\n", 15строка 1

) = 15

write(1, "++++++++++\n", 11++++++++++

) = 11

write(1, "Error. Deque is empty.\n", 23Error. Deque is empty.

) = 23

munmap(0x7fe53db50000, 2105432) = 0

exit\_group(0) = ?

+++ exited with 0 +++