

Отчет по лабораторной работе №5

По курсу «Операционные системы».

Работу выполнила
студентка группы 8О-208Б
Понагайбо А.О.
Сдано: _____

Тема: Динамические библиотеки.

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

Задание: 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

Провести анализ между обоими типами использования библиотеки.

Вариант: 24. Структура данных, с которой должна обеспечивать работу библиотека: Работа с деком. Тип данных, используемый структурой: Md5 суммы.

Текст программы:

```
deq.h
#ifndef _DEQ_H_
#define _DEQ_H_

typedef struct Tmd5 {
    char hash[33];
} Tmd5;

struct node {
    Tmd5* data;
    struct node* next;
};
typedef struct node TNode;

TNode* PushFront(TNode* dq, Tmd5 data);
TNode* PushBack(TNode* dq, Tmd5 data);
Tmd5* TopFront(TNode* dq);
Tmd5* TopBack(TNode* dq);
void PrintDeque(TNode* dq);
int DequeSize(TNode* dq);
TNode* PopFront(TNode* dq);
TNode* PopBack(TNode* dq);
```

```

TNode* DeleteDeque(TNode* dq);
int CheckMD5(Tmd5* num);
#endif

```

deq.c

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "deq.h"

TNode* PushFront(TNode* dq, Tmd5 data) {
    TNode* tmp = (TNode*)malloc(sizeof(TNode));
    tmp->data = (Tmd5*)malloc(sizeof(Tmd5));
    strcpy(tmp->data->hash, data.hash);
    tmp->next = dq;
    return tmp;
}

TNode* PushBack(TNode* dq, Tmd5 data) {
    TNode* tmp = (TNode*)malloc(sizeof(TNode));
    tmp->data = (Tmd5*)malloc(sizeof(Tmd5));
    strcpy(tmp->data->hash, data.hash);
    TNode* last = dq;
    while (last->next != NULL) last = last->next;
    last->next = tmp;
    return dq;
}

Tmd5* TopFront(TNode* dq) {
    if (DequeSize(dq) == 0) {
        printf("Deque is empty!\n");
        return NULL;
    }
    return dq->data;
}

Tmd5* TopBack(TNode* dq) {
    if (DequeSize(dq) == 0) {
        printf("Deque is empty!\n");
        return NULL;
    }
    TNode* last = dq;
    while (last->next != NULL) last = last->next;
    return last->data;
}

void PrintDeque(TNode* dq) {
    if (DequeSize(dq) == 0) {
        printf("Deque is empty!\n");
        return;
    }
    TNode* tmp = dq;
    while (tmp != NULL) {
        printf("%s\n", tmp->data->hash);
        tmp = tmp->next;
    }
}

int DequeSize(TNode* dq) {
    TNode* tmp = dq;
    int sz = 0;
    while (tmp != NULL) {
        sz++;
        tmp = tmp->next;
    }
    return sz;
}

TNode* PopFront(TNode* dq) {

```

```

    if (DequeSize(dq) == 0) {
        printf("Deque is empty!\n");
        return NULL;
    }
    TNode* tmp = dq->next;
    free(dq);
    return tmp;
}

TNode* PopBack(TNode* dq) {
    if (DequeSize(dq) == 0) {
        printf("Deque is empty!\n");
        return NULL;
    }
    if (DequeSize(dq) == 1) {
        free(dq);
        return NULL;
    }
    TNode* last = dq;
    while (last->next->next != NULL) last = last->next;
    free(last->next);
    last->next = NULL;
    return dq;
}

TNode* DeleteDeque(TNode* dq) {
    TNode* tmp;
    while (dq != NULL) {
        tmp = dq;
        dq = dq->next;
        free(dq);
    }
    return NULL;
}

int CheckMD5(Tmd5* num) {
    int sz;
    int len = strlen(num->hash);
    char c;
    for (sz = 0; sz < len; sz++) {
        c = num->hash[sz];
        if (c >= 48 && c <= 57 || c >= 65 && c <= 70 || c >= 97 && c <= 102) {
            continue;
        } else {
            return 0;
        }
    }
    return 1;
}

main_rt.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <dlfcn.h>
#include "deq.h"

int main() {
    printf("It is a programm for loading a shared library at runtime.\n");
    void* handle = dlopen("./libdeq.so", RTLD_LAZY);
    if (handle == NULL) {
        printf("Can not open library.\n");
        return 1;
    }
    void (*PrintDeque)(TNode*) = dlsym(handle, "PrintDeque");
    TNode* myDeque = NULL;
    Tmd5* myMd = malloc(sizeof(Tmd5));
    TNode* (*PushF)(TNode*, Tmd5) = dlsym(handle, "PushFront");
    printf("\n1. Push front \"123abc456def7890123abc456def7890\", print deq:\n");

```

```

strcpy(myMd->hash, "123abc456def7890123abc456def7890");
myDeque = (*PushF)(myDeque, *myMd);
PrintDeque(myDeque);
TNode* (*PushB)(TNode*, Tmd5) = dlsym(handle, "PushBack");
printf("\n2. Push back  \"987def654abc3210987def654abc3210\\", print deq:\n");
strcpy(myMd->hash, "987def654abc3210987def654abc3210");
myDeque = (*PushB)(myDeque, *myMd);
PrintDeque(myDeque);
printf("\n3. Push front  \"a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d\\", print deq:\n");
strcpy(myMd->hash, "a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d");
myDeque = (*PushF)(myDeque, *myMd);
PrintDeque(myDeque);
TNode* (*PopB)(TNode*) = dlsym(handle, "PopBack");
printf("\n4. Pop back, print deq:\n");
myDeque = (*PopB)(myDeque);
PrintDeque(myDeque);
Tmd5* (*TopB)(TNode*) = dlsym(handle, "TopBack");
printf("\n5. Top back, print element:\n");
myMd = (*TopB)(myDeque);
if (myMd != NULL) printf("%s\n", myMd->hash);
TNode* (*PopF)(TNode*) = dlsym(handle, "PopFront");
printf("\n6. Pop front, print deq:\n");
myDeque = (*PopF)(myDeque);
PrintDeque(myDeque);
printf("\n7. Pop back, print deq:\n");
myDeque = (*PopB)(myDeque);
PrintDeque(myDeque);
Tmd5* (*TopF)(TNode*) = dlsym(handle, "TopFront");
printf("\n8. Top front, print element:\n");
myMd = (*TopF)(myDeque);
if (myMd != NULL) printf("%s\n", myMd->hash);
TNode* (*Del)(TNode*) = dlsym(handle, "DeleteDeque");
(*Del)(myDeque);
dlclose(handle);
return 0;
}

```

main_link.c

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "deq.h"

int main() {
    printf("It is a programm for linking with a shared library.\n");
    TNode* myDeque = NULL;
    Tmd5* myMd = malloc(sizeof(Tmd5));
    printf("\n1. Push front  \"123abc456def7890123abc456def7890\\", print deq:\n");
    strcpy(myMd->hash, "123abc456def7890123abc456def7890");
    myDeque = PushFront(myDeque, *myMd);
    PrintDeque(myDeque);
    printf("\n2. Push back  \"987def654abc3210987def654abc3210\\", print deq:\n");
    strcpy(myMd->hash, "987def654abc3210987def654abc3210");
    myDeque = PushBack(myDeque, *myMd);
    PrintDeque(myDeque);
    printf("\n3. Push front  \"a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d\\", print deq:\n");
    strcpy(myMd->hash, "a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d");
    myDeque = PushFront(myDeque, *myMd);
    PrintDeque(myDeque);
    printf("\n4. Pop back, print deq:\n");
    myDeque = PopBack(myDeque);
    PrintDeque(myDeque);
    printf("\n5. Top back, print element:\n");
    myMd = TopFront(myDeque);
    if (myMd != NULL) printf("%s\n", myMd->hash);
    printf("\n6. Pop front, print deq:\n");
    myDeque = PopFront(myDeque);
    PrintDeque(myDeque);
    printf("\n7. Pop back, print deq:\n");
}

```

```

    myDeque = PopBack(myDeque);
    PrintDeque(myDeque);
    printf("\n8. Top front, print element:\n");
    myMd = TopFront(myDeque);
    if (myMd != NULL) printf("%s\n", myMd->hash);
    DeleteDeque(myDeque);
    return 0;
}

```

makefile

```

deq: main_link.o libdeq.so
    gcc -o deq main_link.o -L. -ldeq -Wl,-rpath,.
main_link.o: main_link.c
    gcc -c main_link.c
libdeq.so: deq.o
    gcc -shared -o libdeq.so deq.o
deq.o: deq.c
    gcc -c -fPIC deq.c
clean:
    rm -f *.o deq

```

Тесты:

```

OC/lab5$ make
gcc -c main_link.c
gcc -c -fPIC deq.c
gcc -shared -o libdeq.so deq.o
gcc -o deq main_link.o -L. -ldeq -Wl,-rpath,.
nastya@nastya-GP70-2PE:~/Рабочий стол/OC/lab5$ ./deq
It is a programm for linking with a shared library.

```

1. Push front "123abc456def7890123abc456def7890", print deq:
123abc456def7890123abc456def7890

2. Push back "987def654abc3210987def654abc3210", print deq:
123abc456def7890123abc456def7890
987def654abc3210987def654abc3210

3. Push front "a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d", print deq:
a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d
123abc456def7890123abc456def7890
987def654abc3210987def654abc3210

4. Pop back, print deq:
a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d
123abc456def7890123abc456def7890

5. Top back, print element:
a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d

6. Pop front, print deq:
123abc456def7890123abc456def7890

7. Pop back, print deq:
Deque is empty!

8. Top front, print element:
Deque is empty!
OC/lab5\$ make clean
rm -f *.o deq
OC/lab5\$ gcc main_rt.c -o rt -ldl
OC/lab5\$./rt

It is a programm for loading a shared library at runtime.

1. Push front "123abc456def7890123abc456def7890", print deq:
123abc456def7890123abc456def7890

2. Push back "987def654abc3210987def654abc3210", print deq:

```
123abc456def7890123abc456def7890
987def654abc3210987def654abc3210
```

```
3. Push front "a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d", print deq:
a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d
123abc456def7890123abc456def7890
987def654abc3210987def654abc3210
```

```
4. Pop back, print deq:
a1b3c5d7e9f2a4b6c8d0e2f3a4b5c6d
123abc456def7890123abc456def7890
```

```
5. Top back, print element:
123abc456def7890123abc456def7890
```

```
6. Pop front, print deq:
123abc456def7890123abc456def7890
```

```
7. Pop back, print deq:
Deque is empty!
```

```
8. Top front, print element:
Deque is empty!
```

Выводы: Динамические библиотеки удобно использовать в том случае, когда доступ к данной библиотеке нужен сразу нескольким программам. С их помощью можно обойтись одним экземпляром библиотеки для всех программ.

Для подключения библиотеки во время компиляции необходимо указывать точное расположение этой библиотеки, добавляя ее адрес к переменной окружения `LD_LIBRARY_PATH` либо явно через `rpath` указывая, где ее искать. Если же в программе требуется использовать только несколько функций из динамической библиотеки, то удобнее ее подключать во время выполнения программы, подгрузив библиотеку в память и импортировав нужные функции.