



Отчёт по лабораторной работе № 25-26 по курсу 1

студента группы М8О-108-19 Горохов Михаил Антонович, № по списку 7

Адреса www, e-mail, jabber, skype ma.gorokhoov@gmail.com

Работа выполнена: "26" марта 2020г.

Преподаватель: Поповкин Александр Викторович каф.806

Входной контроль знаний с оценкой _____

Отчёт сдан "26" марта 2020г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Автоматизации сборки программы на Си с использованием утилиты make. Абстрактные типы данных. Рекурсия. Стек
2. **Цель работы:** Освоить основные структуры данных, такие как стек, дек, очередь, линейный список и другие. Научиться собирать многофайловые проекты с помощью утилиты make и makefile (Make файла)
3. **Задание (вариант № 7):** Реализовать стек. Реализовать конкатенацию двух стеков, быструю сортировку Хоара. Собрать проект с помощью утилиты make
4. **Оборудование (лабораторное):**
ЭВМ ---, процессор Intel(R) Pentium(R) CPU G2140 @ 3.30GHz, имя узла сети alice5 с ОП 8192 МБ
НМД 350 ГБ. Терминал --- адрес 192.168.2.105. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Ryzen 3 2200U 2.5 GHz, ОП 4096 МБ, НМД 128 ГБ. Монитор ---
Другие устройства ---

5. **Программное обеспечение (лабораторное):**
Операционная система семейства Linux, наименование Ubuntu версия 18.04.3 LTS
Интерпретатор команд Bash версия 4.4.19(1)-release
Система программирования GCC, cc, make версия ---
Редактор текстов GNU Emacs версия 25.2.2
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождения и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства Linux, наименование Arch Linux версия 5.5.10
Интерпретатор команд Bash версия 5.0.9(1)-release
Система программирования GCC, cc, make версия ---
Редактор текстов vim версия 8.1
Утилиты операционной системы _____

Прикладные системы и программы _____
Местонахождения и имена файлов программ и данных _____

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

Структура проекта:

Файл библиотека с реализацией классического авторасширяемого стека с сортировкой Хоара и функцией слияния стека (без реверса): *laba2526_mag.h*

Файл, использующий библиотеку (для тестирования работы функций): *laba2526_mag.c*

Make-файл, собирающий весь проект воедино: *makefile*

Весь алгоритм записан на языке Си в 7 пункте.

7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты, либо соображения по тестированию].

Итоговый код *laba2526_mag.h*:

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

#define EXTRA 100

typedef long T;
typedef struct {
    int size;
    long index;
    T* data;
} Stack;
void create(Stack* s, int size) {
    s->index = -1;
    s->size = size;
    s->data = malloc(sizeof(T)*size);
}
int is_empty(Stack* s) {
    if(s->index < 0) {
        return 1;
    }
    else {
        return 0;
    }
}
void push(Stack* s, T val) {
    if(s->index >= s->size-1) {
        T* data = malloc(sizeof(T)*(s->size + EXTRA));
        for(int i=0; i < s->size; i++) {
            data[i] = s->data[i];
        }
        free(s->data);
        s->size += EXTRA;
        s->data = data;
    }
    s->index += 1;
    s->data[s->index] = val;
}
T pop(Stack* s) {
    if(is_empty(s) != 0) {
        printf("ERROR! Stack is Empty!");
        return -1;
    }
    s->index -= 1;
    return s->data[s->index+1];
}
```

```

void print(Stack* s) {
    if (is_empty(s) != 0) {
        printf("Stack is Empty!");
    }
    else {
        for(int i=0; i <= s->index; i++) {
            printf("%ld ", s->data[i]);
        }
        printf("\n");
    }
}

int get_size(Stack* s) {
    return s->index+1;
}

void destroy(Stack* s) {
    s->index = -1;
    s->size = 0;
    free(s->data);
}

void merge(Stack* s1, Stack* s2) {
    while(is_empty(s2) != 0) {
        push(s1, pop(s2));
    }
    return;
}

void data_sort(T data[], int left, int right) {
    if (left >= right) {
        return;
    }

    int ind = left;
    for(int i=left+1; i <= right; i++) {
        if(data[ind] >= data[i]) {
            T temp = data[ind];
            data[ind] = data[i];
            data[i] = temp;

            ind = i;
        }
    }
    data_sort(data, left, ind-1);
    data_sort(data, ind+1, right);
}

void sort(Stack* s) {
    if(is_empty(s)) {
        return;
    }
    data_sort(s->data, 0, s->index);
}

```

Итоговый код *laba2526_mag.c*:

```
#include "laba2526_mag.h"
```

```

int main() {
    Stack st;
    create(&st, 10);
    print(&st);
    push(&st, 150);
    print(&st);
    for(int i=0; i < 12; i++) {
        push(&st, i*i);
    }
}

```

```

    print(&st);
    printf("Poped: %ld\n", pop(&st));
    print(&st);

    Stack st2;
    create(&st2, 2);
    for(int i=0; i < 4; i++) {
        push(&st2, i*i + 1);
    }
    print(&st2);
    Stack* st3;
    merge(&st, &st2);
    print(&st);
    sort(&st);
    print(&st);
    destroy(&st);
    print(&st);
    print(&st2);
    destroy(&st2);
    print(&st2);
    return 0;
}

```

Make-Файл *makefile*:

```

laba2526_mag: laba2526_mag.o
    cc -o laba2526_mag laba2526_mag.o
laba2526_mag.o: laba2526_mag.c laba2526_mag.h
    cc -c laba2526_mag.c

```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

- 8. Распечатка протокола** (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)

```

[revammark@Raft laba25-26]$ date
Чт 26 мар 2020 15:38:28 MSK
[revammark@Raft laba25-26]$ ls
l25-2012.djvu laba2526_mag laba2526_mag.h laba_2526_mag.odt
l26-2012.djvu laba2526_mag.c laba2526_mag.o makefile
[revammark@Raft laba25-26]$ make
cc -c laba2526_mag.c
cc -o laba2526_mag laba2526_mag.o
[revammark@Raft laba25-26]$ make
make: 'laba2526_mag' is up to date.
[revammark@Raft laba25-26]$ ls
l25-2012.djvu laba2526_mag laba2526_mag.h laba_2526_mag.odt
l26-2012.djvu laba2526_mag.c laba2526_mag.o makefile
[revammark@Raft laba25-26]$ ./laba2526_mag
Stack is Empty!
150
150 0 1 4 9 16 25 36 49 64 81 100 121
Poped: 121
150 0 1 4 9 16 25 36 49 64 81 100
Stack is Empty!
[revammark@Raft laba25-26]$

[revammark@Raft laba25-26]$
[revammark@Raft laba25-26]$ make
cc -c laba2526_mag.c
In file included from laba2526_mag.c:1:

```

```

laba2526_mag.h: In function 'merge':
laba2526_mag.h:74:12: warning: function returns address of local variable [-Wreturn-local-addr]
 74 |     return &new_stack;
    |           ^~~~~~
laba2526_mag.h: At top level:
laba2526_mag.h:76:6: error: conflicting types for 'qsort'
 76 | void qsort(T data[], int left, int right) {
    |      ^~~~~
In file included from laba2526_mag.h:1,
                  from laba2526_mag.c:1:
/usr/include/stdlib.h:830:13: note: previous declaration of 'qsort' was here
 830 | extern void qsort (void *__base, size_t __nmem, size_t __size,
    |           ^~~~~
make: *** [makefile:4: laba2526_mag.o] Error 1
[revammark@Raft laba25-26]$
[revammark@Raft laba25-26]$
[revammark@Raft laba25-26]$ date
Thu 26 Mar 2020 16:30:00 MSK
[revammark@Raft laba25-26]$ make
cc -c laba2526_mag.c
In file included from laba2526_mag.c:1:
laba2526_mag.h: In function 'merge':
laba2526_mag.h:74:12: warning: function returns address of local variable [-Wreturn-local-addr]
 74 |     return &new_stack;
    |           ^~~~~~
cc -o laba2526_mag laba2526_mag.o
[revammark@Raft laba25-26]$ ./laba2526_mag
Stack is Empty!
150
150 0 1 4 9 16 25 36 49 64 81 100 121
Popped: 121
150 0 1 4 9 16 25 36 49 64 81 100
1 2 5 10
Segmentation fault (core dumped)

[revammark@Raft laba25-26]$ make
cc -c laba2526_mag.c
cc -o laba2526_mag laba2526_mag.o
[revammark@Raft laba25-26]$ date
Thu 26 Mar 2020 17:08:15 MSK
[revammark@Raft laba25-26]$ make
make: 'laba2526_mag' is up to date.
[revammark@Raft laba25-26]$ ls
l25-2012.djvu laba2526_mag laba2526_mag.h laba_2526_mag.odt
l26-2012.djvu laba2526_mag.c laba2526_mag.o makefile
[revammark@Raft laba25-26]$ ./laba2526_mag
Stack is Empty!
150
150 0 1 4 9 16 25 36 49 64 81 100 121
Popped: 121
150 0 1 4 9 16 25 36 49 64 81 100
1 2 5 10
150 0 1 4 9 16 25 36 49 64 81 100
0 1 4 9 16 25 36 49 64 81 100 150
Stack is Empty!
1 2 5 10
Stack is Empty!
[revammark@Raft laba25-26]$ make
make: 'laba2526_mag' is up to date.
[revammark@Raft laba25-26]$ rm laba2526_mag.o
[revammark@Raft laba25-26]$ make
cc -c laba2526_mag.c
cc -o laba2526_mag laba2526_mag.o
[revammark@Raft laba25-26]$ ./laba2526_mag

```

```

Stack is Empty!
150
150 0 1 4 9 16 25 36 49 64 81 100 121
Popped: 121
150 0 1 4 9 16 25 36 49 64 81 100
1 2 5 10
150 0 1 4 9 16 25 36 49 64 81 100
0 1 4 9 16 25 36 49 64 81 100 150
Stack is Empty!
1 2 5 10
Stack is Empty!
[revammark@Raft laba25-26]$ make
make: 'laba2526_mag' is up to date.
[revammark@Raft laba25-26]$ echo // Comment > laba
laba2526_mag    laba2526_mag.h    laba_2526_mag.odt
laba2526_mag.c    laba2526_mag.o
[revammark@Raft laba25-26]$ echo // Comment > laba2526_mag.h
[revammark@Raft laba25-26]$ make
cc -c laba2526_mag.c
laba2526_mag.c: In function 'main':
laba2526_mag.c:5:5: error: unknown type name 'Stack'
    5 |     Stack st;
      |     ^~~~~~
laba2526_mag.c:6:5: warning: implicit declaration of function 'create' [-Wimplicit-function-declaration]
    6 |     create(&st, 10);
      |     ^~~~~~
laba2526_mag.c:7:5: warning: implicit declaration of function 'print' [-Wimplicit-function-declaration]
    7 |     print(&st);
      |     ^~~~~~
laba2526_mag.c:8:5: warning: implicit declaration of function 'push' [-Wimplicit-function-declaration]
    8 |     push(&st, 150);
      |     ^~~~~~
laba2526_mag.c:15:5: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
   15 |     printf("Popped: %ld\n", pop(&st));
      |     ^~~~~~
laba2526_mag.c:15:5: warning: incompatible implicit declaration of built-in function 'printf'
laba2526_mag.c:2:1: note: include '<stdio.h>' or provide a declaration of 'printf'
    1 | #include "laba2526_mag.h"
      | ^
+++ |+#include <stdio.h>
    2 |
laba2526_mag.c:15:28: warning: implicit declaration of function 'pop' [-Wimplicit-function-declaration]
   15 |     printf("Popped: %ld\n", pop(&st));
      |                             ^~~~~~
laba2526_mag.c:18:5: error: unknown type name 'Stack'
   18 |     Stack st2;
      |     ^~~~~~
laba2526_mag.c:24:5: error: unknown type name 'Stack'
   24 |     Stack* st3;
      |     ^~~~~~
laba2526_mag.c:25:5: warning: implicit declaration of function 'merge' [-Wimplicit-function-declaration]
   25 |     merge(&st, &st2);
      |     ^~~~~~
laba2526_mag.c:27:5: warning: implicit declaration of function 'sort' [-Wimplicit-function-declaration]
   27 |     sort(&st);
      |     ^~~~~~
laba2526_mag.c:29:5: warning: implicit declaration of function 'destroy' [-Wimplicit-function-declaration]
   29 |     destroy(&st);
      |     ^~~~~~
make: *** [makefile:4: laba2526_mag.o] Error 1
[revammark@Raft laba25-26]$
[revammark@Raft laba25-26]$
[revammark@Raft laba25-26]$
[revammark@Raft laba25-26]$ make
cc -c laba2526_mag.c

```

```

cc -o laba2526_mag laba2526_mag.o
[revammark@Raft laba25-26]$ make
make: 'laba2526_mag' is up to date.
[revammark@Raft laba25-26]$ ./laba2526_mag
Stack is Empty!
150
150 0 1 4 9 16 25 36 49 64 81 100 121
Poped: 121
150 0 1 4 9 16 25 36 49 64 81 100
1 2 5 10
150 0 1 4 9 16 25 36 49 64 81 100
0 1 4 9 16 25 36 49 64 81 100 150
Stack is Empty!
1 2 5 10
Stack is Empty!
[revammark@Raft laba25-26]$

```

9. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	При меча ние

10. Замечание автора по существу работы _____ Очень интересная лабораторная работа! Мне понравилось реализовывать одну из самых лучших сортировок, сортировку Хоара. Интересно реализовывать основные структуры данных.
11. Выводы _____ Я научился собирать многофайловые проекты на Си с помощью утилиты make. Я научился реализовывать основные типы данных на примере стека, научился реализовывать алгоритм сортировки на примере quicksort (Быстрая сортировка Хоара). Сбока с обновлением файлов приложена в протоколе.
- Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента _____