Отчёт по лабораторной работе № 25-26 по курсу Практикум на ЭВМ

студента группы М8О-108Б Жерлыгина Максима Андреевича, № по списку 8

Адреса www, e-mail, jabber, skype mmaxim2710@gmail.com

Работа выполнена: "15" мая 2019г.

Преподаватель:_____ каф.806 _____

Входной контроль знаний с оценкой _____

Отчёт сдан "___" _____ 20____г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема**: Автоматизация сборки программ модульной структуры на языке Си с использованием утилиты make. Абстрактные типы данных. Рекурсия. Модульное программирование на языке Си.

2. **Цель работы**: Получить навыки работы по сборке программ модульной структуры.

3. **Задание** (*вариант №* 8): Поиск в очереди двух элементов, идущих подряд, первый из которых больше второго. Если такие элементы найдены, их перестановка. Сортировка методом пузырька.

4. **Оборудование** (*лабораторное*):
ЭВМ компьютер, процессор Intel Core2 Duo CPU E8500 @ 3.163GHz, имя узла сети cameron с ОП 16029 МБ
НМД 2 ГБ. Терминал gnome адрес 172.16.80.213. Принтер Лазерный с технологией pulling
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:
Процессор Intel Core i5-7200U @ 4x 2.712GHz, ОП 8073 МБ, НМД 464 ГБ. Монитор _____
Другие устройства _____

5. **Программное обеспечение** (*лабораторное*):
Операционная система семейства Unix, наименование Ubuntu версия 16.04
Интерпретатор команд bash версия 4.3.48
Система программирования _____ версия 8.0
Редактор текстов VIM версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождения и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства Unix, наименование Ubuntu версия 18.04
Интерпретатор команд bash версия 4.4.19
Система программирования _____ версия 8.0
Редактор текстов VIM версия _____
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.

    1. Изучить обучающие материалы по утилите make.
    2. Изучить материалы по теме «сортировка пузырьком»
    3. Написать программы, соответсвенно варианту, собрать модули программы с помощью утилиты make.

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

    1. Написать функцию, осуществляющую пузырьковую сортировку.
    2. Написать «main» программу.
    3. Написать файл «functions.h»
    4. Подготовить Makefile для сборки модульной программы.
    5. Осуществить модульную сборку программы.

*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)

Сборка программы

```
mmaxim2710@DESKTOP-RDPBU3D:~/2sem/mkfile$ make -f Makefile
gcc -c main.c
gcc -g main.o bubble.o -o sort
mmaxim2710@DESKTOP-RDPBU3D:~/2sem/mkfile$ ls
Makefile  bubble.c  bubble.o  functions.h  main.c  main.o  sort
```

Makefile

```
all: sort

sort: main.o bubble.o
	gcc -g main.o bubble.o -o sort

main.o: main.c
	gcc -c main.c

bubble.o: bubble.c
	gcc -c bubble.c
```

main.c

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

int main() {
	int size;
	printf("Введите количество элементов:");
	scanf("%d", &size);
	int array[size];
	putchar('\n');
```

```
	printf("Введите элементы через Enter:\n");
	for (int j = 0; j < size; j++) {
		scanf("%d", &array[j]);
	}

	bubble_sort(array, size);

	for (int j = 0; j < size; j++) {
		printf("%d", array[j]);
	}
	putchar('\n');
	return 0;
}
```

bubble.c

```
#include <stdio.h>

int bubble_sort(int *arr, int array_size) {
	int i = 0;
	int buff;
	char swap = 0;

	if (array_size = 0) return 0;

	while (i < array_size) {
		if (i + 1 != array_size && arr[i] > arr[i + 1]) {
			buff = arr[i];
			arr[i] = arr[i + 1];
			arr[i + 1] = buff;
			swap = 1;
		}
		i++;
		if (i == array_size && swap == 1) {
		i = 0;
		swap = 0;
		}
	}
}
```

functions.c

```
int bubble_sort(int *array, int array_size);
```

Результат работы программы

```
mmaxim2710@DESKTOP-RDPBU3D:~/2sem/mkfile$ ./sort
Введите количество элементов:5

Введите элементы через Enter:
3
2
1
4
```

5
12345

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	Примечание

10. Замечание автора по существу работы _____

11. Выводы Я получил навыки работы с утилитой make. Сделал вывод, что данная утилита помогает автоматизировать процесс сборки модульных программ. Если они объемные, то данная программа помогает со сборкой отдельной части программы, избегая полной компиляции всего проекта, что при объемности последнего может занять очень долгое время. Данная утилита может автоматизировать процесс сборки путём написания универсального шаблона, который можно подстраивать под нужный проект вместе с удалением ненужных файлов.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом ____
_____
_____
_____
_____
_____
_____

Подпись студента _____