|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-01 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | О |  | Естественнонаучный |
|  |  | шифр |  | наименование |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Компьютерный практикум | | |

Отчёт по практической работе

|  |
| --- |
| Создание динамических библиотек при помощи |
| набора компиляторов и утилит GCC |
| и их применение |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И507Б |
| Шмаков И.Н. | | | | | |
| Фамилия И.О. | | | | | |
| **Преподаватель** | | | | | |
| Васюков В.М. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2021г. |

САНКТ-ПЕТЕРБУРГ

2021 г.

**Вариант 4**

Подсчитать общее количество отрицательных элементов в массиве D (42) и матрице A (6х9).

**Задания к лабораторной работе**

1. Написать программу в соответствии с вариантом.
2. Массив и матрицу заполнять случайными числами от -50 до 50.
3. Функции для работы с массивами и матрицами поместить в две отдельные динамические библиотеки.
4. При запуске программы пользователь должен увидеть меню, в котором можно выбрать, с чем будет проходить работа: с матрицей или с массивом.
5. В зависимости от выбора пользователя загружается одна или другая динамическая библиотека.
6. Библиотеки должны быть скомпилированы с учетом возможного использования в ОС семейств Linux или Windows.

**Текст программы:**

**main.c**

#include <stdio.h>

#include "func\_prototype.h"

int main() {

func();

return 0;

}

**loadLib.h**

void LoadRun(const char \*);

**func\_prototype.h**

void func();

**loadMenu.c**

#include "loadLib.h"

#include <stdio.h>

#include <stdlib.h>

int main() {

system("chcp 65001");

int a = 0, b = 1;

puts("Выберите библиотеку:");

puts("1. Библиотека с матрицей");

puts("2. Библиотека с массивом");

puts("3. Выход");

while (b) {

scanf("%d", &a);

if (a == 1)

LoadRun("funcMatrix.dll");

if (a == 2)

LoadRun("funcArray.dll");

if (a == 3)

b = 0;

else puts("Введите число в диапазоне от 1 до 3.");

}

return 0;

}

**loadLib.c**

#include "loadLib.h"

#include "func\_prototype.h"

#include <stdio.h>

#include <windows.h>

void LoadRun(const char \*fileName) {

void \*pLibrary;

void (\*pFunction)();

pLibrary = LoadLibrary(fileName); //загрузка библиотеки в память;

if (!pLibrary) {

printf("can not open library '%fileName'\n", fileName);

return;

}

pFunction = (void (\*)()) GetProcAddress((HINSTANCE) pLibrary, "func"); //получение указателя на функцию из библиотеки;

if (pFunction == NULL) {

printf("can not load function func\n");

} else {

pFunction();

}

FreeLibrary((HINSTANCE) pLibrary); //выгрузка библиотеки;

}

**funcMatrix.c**

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include "func\_prototype.h"

void outMatrix(int \*\*, int, int);

void func() {

system("chcp 65001");

srand(time(NULL));

int matrixRow = 6, matrixColumn = 9, count = 0;

int \*\*ptrMatrix = (int \*\*) malloc(matrixRow \* sizeof(int \*));

for (int i = 0; i < matrixRow; ++i) {

ptrMatrix[i] = (int \*) malloc(matrixColumn \* sizeof(int));

}

for (int i = 0; i < matrixRow; ++i) {

for (int j = 0; j < matrixColumn; ++j) {

ptrMatrix[i][j] = rand() % 101 - 50;

}

}

puts("\nЗаполнение матрицы случайными числами:\n");

outMatrix(ptrMatrix, matrixRow, matrixColumn);

for (int i = 0; i < matrixRow; ++i) {

for (int j = 0; j < matrixColumn; ++j) {

if (ptrMatrix[i][j] > 0) count += 1;

}

}

puts("Выполнение условия:\n");

outMatrix(ptrMatrix, matrixRow, matrixColumn);

printf("Общее число отрицательных элементов в матрице: %d", count);

puts("\n");

system("pause");

}

void outMatrix(int \*\*matrix, int matrix\_row, int matrix\_column) {

for (int i = 0; i < matrix\_row; ++i) {

for (int j = 0; j < matrix\_column; ++j) {

printf("%d\t", matrix[i][j]);

if (j == 5) puts("\n");

}

}

}

**funcArray.c**

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include "func\_prototype.h"

void outArray(int \*, int);

void func() {

system("chcp 65001");

srand(time(NULL));

int arraySize = 42, count = 0;

int \*matrix = (int \*) malloc(arraySize \* sizeof(int));

for (int i = 0; i < arraySize; ++i) {

matrix[i] = rand() % 101 - 50;

}

puts("Заполнение массива случайными числами:");

outArray(matrix, arraySize);

for (int i = 0; i < arraySize; ++i) {

if (matrix[i] > 0) count += 1;

}

puts("\n\nВыполнение условия:");

outArray(matrix, arraySize);

printf("Общее число отрицательных элементов в массиве: %d", count);

puts("\n");

system("pause");

}

void outArray(int \*ptrArray, int array\_size) {

for (int i = 0; i < array\_size; ++i) printf("%d\t", ptrArray[i]);

}

**Процесс компиляции показан на рисунке 1:**

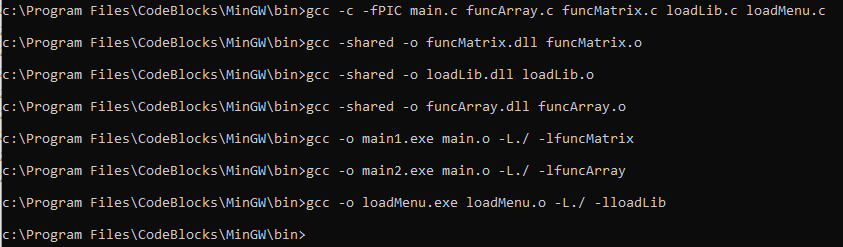
****

Рисунок 1

Описание команд:

gcc -c -fPIC создаются объектные файлы. Параметр -fPIC требует от компилятора создание позиционно-независимого кода при создании объектных файлов.

gcc -shared -o – создаются динамические библиотеки.

gcc -o – в резльтате выполнения команды будет создан исполняемй файл с подключенной динамической библиотекой.

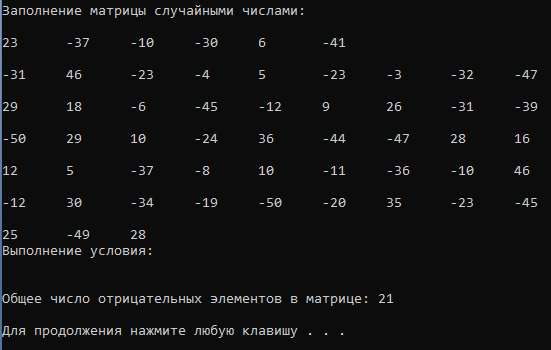
**Результат работы программы для матрицы показан на рисунке 2:**

Рисунок 2

**Результат работы программы для массива показан на рисунке 3:**

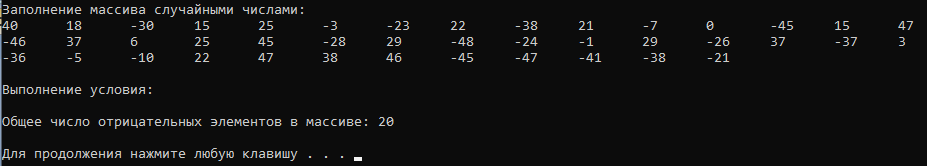
****

Рисунок 3