

## Лабораторная работа №5

**Тема Лабораторной работы:** Многофайловый проект, условная компиляция, утилиты Make.

### Задание 1.1

**Постановка задачи:** Напишите программу из нескольких файлов (модулей), включая файл основной программы. Файлы должны содержать вынесенные отдельно функции для выделения памяти под динамические двумерные и одномерные массивы и функции для перемножения матриц. Собрать проект используя утилиту Make

### Список идентификаторов:

Имя	Тип	Смысл

### Код программы:

#### Makefile

```
CC = gcc
CFLAGS = -Wall -Wextra -Werror -std=c99
LDFLAGS =

SRCS = main.c RAM.c matrix.c
OBJS = $(SRCS:.c=.o)
TARGET = matrix_multiplication

.PHONY: all clean

all: $(TARGET)

$(TARGET): $(OBJS)
    $(CC) $(LDFLAGS) -o $@ $^

%.o: %.c
    $(CC) $(CFLAGS) -c -o $@ $<

clean:
    rm -f $(OBJS) $(TARGET)
```

## Main

```
1 #include <stdio.h>
2 #include "RAM.h"
3 #include "matrix.h"
4
5 int main() {
6     int **a, **b, **c;
7     int n = 3, m = 3, p = 3;
8
9     a = allocate_matrix(n, m);
10    b = allocate_matrix(m, p);
11    c = allocate_matrix(n, p);
12
13    for (int i = 0; i < n; i++) {
14        for (int j = 0; j < m; j++) {
15            a[i][j] = i + j;
16        }
17    }
18
19    for (int i = 0; i < m; i++) {
20        for (int j = 0; j < p; j++) {
21            b[i][j] = i * j;
22        }
23    }
24
25    multiply_matrices(a, b, c, n, m, p);
26
27    printf("Result:\n");
28    for (int i = 0; i < n; i++) {
29        for (int j = 0; j < p; j++) {
30            printf("%d ", c[i][j]);
31        }
32        printf("\n");
33    }
34
35    deallocate_matrix(a, n);
36    deallocate_matrix(b, m);
37    deallocate_matrix(c, n);
38
39    return 0;
40 }
```

## RAM

```
1 #include <stdlib.h>
2
3 int **allocate_matrix(int n, int m) {
4     int **matrix = (int **) malloc(n * sizeof(int *));
5     for (int i = 0; i < n; i++) {
6         matrix[i] = (int *) malloc(m * sizeof(int));
7     }
8     return matrix;
9 }
10
11 void deallocate_matrix(int **matrix, int n) {
12     for (int i = 0; i < n; i++) {
13         free(matrix[i]);
14     }
15     free(matrix);
16 }
```

## Matrix

```
1 #include "RAM.h"
2
3 void multiply_matrices(int **a, int **b, int **c, int n, int m, int p) {
4     for (int i = 0; i < n; i++) {
5         for (int j = 0; j < p; j++) {
6             c[i][j] = 0;
7             for (int k = 0; k < m; k++) {
8                 c[i][j] += a[i][k] * b[k][j];
9             }
10        }
11    }
12 }
```

## Результат выполненной работы:

```
C:\Windows\System32>cd C:\Users\gnevn\OneDrive\Рабочий стол\учеба\РГПУ\Программирование\ЛР5\Codes
C:\Users\gnevn\OneDrive\Рабочий стол\учеба\РГПУ\Программирование\ЛР5\Codes>make
"make" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\gnevn\OneDrive\Рабочий стол\учеба\РГПУ\Программирование\ЛР5\Codes>
```