**Практическая работа № 4**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил | Студент гр. МФ-11 | Лазаренко А. В. | 13.03.2021 |
| Проверил | Ст. преподаватель | Меняйлов Е. С. |  |

**Цель работы:**

1) Ознакомиться и освоить работу с командами битовых операций на языке Ассемблер.

2) Освоить включение ассемблерных команд в программу на языке высокого уровня.

3) Выполнить задание согласно варианту. Результаты вычислений вывести на экран в двоичной системе счисления.

**Задание:**

1) Ознакомиться и освоить работу с командами битовых операций на языке Ассемблер.

2) Освоить включение ассемблерных команд в программу на языке высокого уровня.

3) Выполнить задание согласно варианту. Результаты вычислений вывести на экран в изначальной и двоичной системах счисления.

**Вариант 11:**

**Задано число в регистре AX. Установить биты 1,2,3 в значение 1.**

**Результат записать в регистр BX. Результат ывести на экран.**

/\* Algorithm:

\* 1) declaring variables "number" (which will store the number from 0 to 255) and "result" (which will store the result of the logical operation);

\* 2) declaring the variable "mask" and initializing it with 7 (1, 2, 3 bites will be 1);

\* 3) in the assembler part:

\* 3.1) putting the value of the variable "number" in the AX register;

\* 3.2) putting the value of the variable "mask" in the BX register;

\* 3.3) performing a bitwise operation on two numbers stored in the BX and AX registers respectively; result overwrites the BX register value;

\* 3.4) taking the value stored in the BX register and putting it as a value of the variable "result";

\* 4) declaring char-type arrays for storing the binary code of the variables "number", "mask" and "result";

\* 5) outputting the initial values of the variables "number", "mask" and "result" and first 8 elements of the specialized char-type arrays.

\*/

#include <iostream>

constexpr auto BYTE\_MAX\_VALUE = 256;

int main(void)

{

int number = 0; // starting from 0 and go on until 255 to get through all the 256 possible options

int mask = 14; // 2\*\*3 + 2\*\*2 + 2\*\*1 = 8 + 4 + 2 = 12 + 2 = 14 (00001110)

int result = 0; // a variable which stores the result value of "or" operation in assembler part

while (number != 255)

{

\_\_asm

{

mov eax, number; // putting the value of the variable "number" in the AX register

mov ebx, mask; // putting the value of the variable "mask" in the BX register

or ebx, eax; // performing a bitwise operation on two numbers stored in the BX and AX registers respectively; result overwrites the BX register value

mov result, ebx; // taking the value stored in the BX register and putting it as a value of the variable "result"

}

char Number[BYTE\_MAX\_VALUE] = "0"; // creating an array for storing the binary code of a value of the variable "number"

char Mask[BYTE\_MAX\_VALUE] = "0"; // creating an array for storing the binary code of a value of the variable "mask"

char Result[BYTE\_MAX\_VALUE] = "0"; // creating an array for storing the binary code of a value of the variable "result"

\_itoa\_s(number, Number, 2); // converting integer to a string, storing its binary code in specialized char array

printf("The number =\t%2d\t%08s\n", number, Number); // outputting the value of the variable "number" and fist 8 symbols of its char array

\_itoa\_s(mask, Mask, 2); // converting integer to a string, storing its binary code in specialized char array

printf("The mask =\t%2d\t%08s\n", mask, Mask); // outputting the value of the variable "mask" and fist 8 symbols of its char array

\_itoa\_s(result, Result, 2); // converting integer to a string, storing its binary code in specialized char array

printf("The result =\t%2d\t%08s\n", result, Result); // outputting the value of the variable "result" and fist 8 symbols of its char array

system("pause");

system("cls"); // clearing the screen every iteration

number++;

}

return 0;

}

**Результаты:**

1) 

2) 

3) 

**Задание\*:**

#include <iostream>

using namespace std;

int main(void)

{

char Number[255] = "";

char Result[255] = "";

unsigned int number;

unsigned int result;

cin >> number;

result = number;

result |= (1 << 1) | (1 << 2) | (1 << 3);

\_itoa\_s(number, Number, 2);

printf("The initial number : %u.\tIn binary : % 016s\n", number, Number);

\_itoa\_s(result, Result, 2);

printf("Operation result : % 4u.\tIn binary : % 016s\n", result, Result);

return 0;

}

**Результат выполнения:**



**Вывод:**

Я ознакомился и освоил работу с командами битовых операций на языке Ассемблер, освоил процесс включения ассемблерных команд в программу на языке высокого уровня.