**Практическая работа № 5**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил | Студент гр. МФ-11 | Лазаренко А. В. | 20.03.2021 |
| Проверил | Ст. преподаватель | Меняйлов Е. С. |  |

**Цель работы:**

1) Ознакомиться с принципами обработки битов в регистрах (поиск, инверсия, сброс бита).

**Задание:**

1) Проанализировать задание, составить алгоритм решения поставленной задачи, с учетом применения команд ассемблера, определить необходимые входные и выходные данные.

2) Написать на языке ассемблер программу, выполняющую поставленную задачу.

3) Написать программу на С++, которая вводит значения входных

параметров и осуществляет проверку на аномалии, при необходимости

выдавая предупреждающее сообщение, после чего проводит обработку

данных на ассемблере и выводит результат на экран.

4) Протестировать программу на корректных и аномальных входных

данных.

**Вариант 11**

**Номер 1:**

**В регистре ECX число задано случайным образом. Определить номер первого встретившегося бита равного 1. Число положительное. Результат вывести на экран.**

Не аномальные значения, при которых программа выполнится правильно: диапазон [−2 147 483 648, 2 147 483 647].

/\* Algorithm:

\* 1) declaring necessary variables "temp", "number" and "result";

\* 2) declaring the necessary char array "Number" (for storing the binary code of the number);

\* 3) asking a user to assign the value to a variable "temp";

\* 4) while a user keeps entering value, which is higher than int max or lower than int min, he is asked to do it again with special preventive message;

\* 5) if the entered value is "correct", the program keeps going;

\* 6) clearing the screen every cycle's iteration;

\* 7) putting the value of a variable "number" in a ECX register;

\* 8) searching for the first equal to '1' bit in a ECX register, and overwrite data stored in ECX register with the number of that bit;

\* 9) putting the value stored in a CX register in a variable "result";

\* 10) outputting the result of the program; initial number, its binary code, the number of the first equal to '1' bit.

\*/

#include <iostream>

using namespace std;

int main(void)

{

long long temp;

int number;

int result = 0;

char Number[255] = "";

srand(time(0));

std::cout << "Enter the number: ";

std::cin >> temp;

while ((temp > INT32\_MAX) || (temp < INT32\_MIN))

{

std:cout << "You entered inadmissible value, it should be lesser than " << INT32\_MAX << " or greater than " << INT32\_MIN << "" << std::endl;

std::cout << "Please reenter the number: ";

std::cin >> temp;

}

number = temp;

for (int i = 0; i != 4; i++)

{

system("cls");

//number = rand() % 2000000;

\_\_asm

{

bsf ECX, number;

mov result, ECX;

}

if (number == 0)

{

printf("There are no bites that equal one in this number.\n");

system("pause");

return 0;

}

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

printf("The initial number: %d\tIn binary: %08s\n", number, Number);

printf("operation result : %d.\n", result);

system("pause");

}

return 0;

}

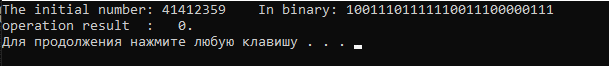
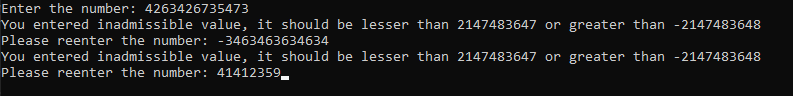
**Результаты программы:**

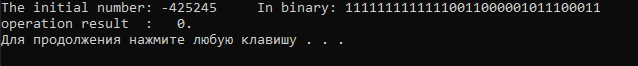
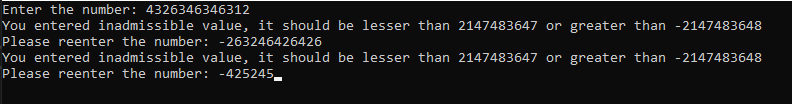
Нормальные данные:

1) 

2) 

Аномальные данные:

3) 

4) 

**Номер 2:**

**В регистре CX задано число случайным образом. Заменить 10-й бит**

**на 0.**

Не аномальные значения, при которых программа выполнится правильно: диапазон [−32 768,32 767].

/\* Algorithm:

\* 1) declaring necessary variables "temp", "number" and "result";

\* 2) declaring necessary char arrays "Number" and "Result" (for storing the binary code of the number);

\* 3) asking a user to assign the value to a variable "temp";

\* 4) while a user keeps entering value, which is higher than short max or lower than short min, he is asked to do it again with special preventive message;

\* 5) if the entered value is "correct", the program keeps going;

\* 6) clearing the screen every cycle's iteration;

\* 7) putting the value of a variable "number" in a CX register;

\* 8) searching for the 10-th bit in a CX register and resetting it to 0;

\* 9) putting the value stored in a CX register in a variable "result";

\* 10) outputting the resukt of the program; initial number, its binary code, changed number, and its binary code.

\*/

#include <iostream>

using namespace std;

int main(void)

{

int temp;

short number;

short result = 0;

char Number[255] = "";

char Result[255] = "";

srand(time(0));

std::cout << "Enter the number: ";

std::cin >> temp;

while ((temp > SHRT\_MAX) || (temp < SHRT\_MIN))

{

std:cout << "You entered inadmissible value, it should be lesser than " << SHRT\_MAX << " or greater than " << SHRT\_MIN << "" << std::endl;

std::cout << "Please reenter the number: ";

std::cin >> temp;

}

number = temp;

for (int i = 0; i != 4; i++)

{

system("cls");

// number = rand() % 32700;

\_\_asm

{

mov CX, number;

btr CX, 10;

mov result, CX;

}

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

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

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

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

system("pause");

}

return 0;

}

Нормальные данные:

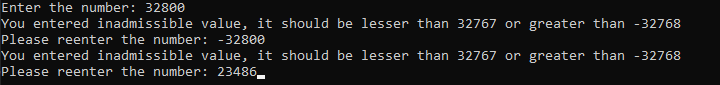
1) 

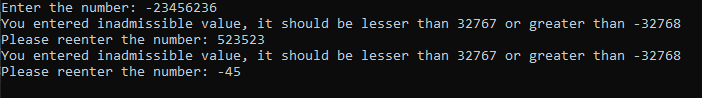
2) 

3) 

4) 

Аномальные данные:

5) 

6) 



З**адание\*:**

**Номер 1**

#include <iostream>

using namespace std;

int main(void)

{

char Number[255] = "";

unsigned int number;

cin >> number;

unsigned int bit\_counter = 0;

unsigned int temp = number;

while ((temp & 1) != 1)

{

temp >>= 1;

bit\_counter++;

}

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

printf("Initial number : %u \tIn binary : \t % 032s; \n", number, Number);

printf("The number of the first bit equal to \'1\' is: %d.\n", bit\_counter);

return 0;

}

**Результат выполнения:**



**Номер 2**

#include <iostream>

using namespace std;

int main(void)

{

char Number[255] = "";

char Result[255] = "";

unsigned int number;

unsigned int result;

cin >> number;

result = number & (~(1 << 10));

\_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 с принципами обработки битов в регистрах (поиск, инверсия, сброс бита), используя команды языка ассемблер.