**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №6  
дисциплина: Архитектура вычислительных систем  
тема: «Логические команды и команды сдвига»

Выполнил: ст. группы ВТ-31  
Ковалёв И. Д.  
Проверил: Осипов О. В.

**Белгород 2019**

**Цель работы:** изучение команд поразрядной обработки данных.

**Задание к работе:**

1. Вывести все 16-разрядные числа, в двоичном представлении которых есть две или одна единицы, остальные – ноль.
2. 40 байт, умножение без знака.

Задание №1:

486

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

INCLUDE C:\MASM32\INCLUDE\WINDOWS.INC

INCLUDE C:\MASM32\INCLUDE\KERNEL32.INC

INCLUDE C:\MASM32\INCLUDE\USER32.INC

INCLUDE C:\masm32\include\msvcrt.inc

INCLUDELIB C:\MASM32\LIB\USER32.LIB

INCLUDELIB C:\MASM32\LIB\KERNEL32.LIB

INCLUDELIB C:\masm32\lib\msvcrt.lib

.DATA

number\_format db "%d: ", 0

new\_line\_format db 13, 10, 0

count dd 0

.CODE

print\_bin proc

PUSH EAX

PUSH EBX

PUSH ECX

PUSH EDX

PUSH EBP

XOR EBX, EBX

MOV EAX, [ESP + 4 \* 6] ; Взять из стека аргумент

MOV ECX, 16 ;Счетчик цикла

j1:

ROL AX, 1 ; Сдвиг влево

MOV BX, AX

AND BX, 0000000000000001b;

ADD BX, '0'

PUSH EAX

PUSH ECX

PUSH EBX

CALL crt\_\_putch

ADD ESP, 4

POP ECX

POP EAX

LOOP j1

PUSH EAX

PUSH ECX

PUSH offset new\_line\_format

CALL crt\_printf

ADD ESP, 4

POP ECX

POP EAX

POP EBP

POP EDX

POP ECX

POP EBX

POP EAX

RET 8

print\_bin endp

;Вывести все 16-разрядные числа, в двоичном представлении

;которых есть две или одна единицы, остальные – ноль.

;1: 0000000000000001

;2: 0000000000000011

;3: 0000000000000101

START:

XOR ECX, ECX

XOR EBX, EBX

MOV ECX, 16

MOV EDX, 0000000000000001b

j1:

MOV EBX, ECX

MOV EAX, EDX

j2:

PUSH EAX

PUSH ECX

PUSH EDX

ADD count, 1

MOV EBP, count

PUSH EBP

PUSH OFFSET number\_format

CALL crt\_printf

ADD ESP, 8

POP EDX

POP ECX

POP EAX

INC ECX

ROL EAX, 1

ADD EAX, EDX

PUSH AX

CALL print\_bin

SUB EAX, EDX

DEC ECX

LOOP j2

ROL EDX, 1

MOV ECX, EBX

LOOP j1

CALL crt\_\_getch ; Задержка ввода с клавиатуры

PUSH 0

CALL ExitProcess ; Выход из программы

END START

Задание №2:

.486

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

INCLUDE C:\MASM32\INCLUDE\WINDOWS.INC

INCLUDE C:\MASM32\INCLUDE\KERNEL32.INC

INCLUDE C:\MASM32\INCLUDE\USER32.INC

INCLUDE C:\masm32\include\msvcrt.inc

INCLUDELIB C:\MASM32\LIB\USER32.LIB

INCLUDELIB C:\MASM32\LIB\KERNEL32.LIB

INCLUDELIB C:\masm32\lib\msvcrt.lib

.DATA

number\_format db "%d: ", 0

number dd 0d, 0d, 0d, 0d, 0d, 0d, 0d, 0d, 0d, 0d,0d, 0d, 0d, 0d, 0d,0d, 0d, 0d, 0d, 0d,0d, 0d, 0d, 0d, 0d,0d, 0d, 0d, 0d, 0d,0d, 0d, 0d, 0d, 0d,0d, 0d, 1d, 0d, 1d

.CODE

multiplication proc

PUSH EAX

PUSH EBX

PUSH ECX

PUSH EDX

XOR EDX, EDX

MOV ECX, [esp + 4 \* 5]

j1:

PUSH ECX

MOV ECX, 0

CLC; сброс флага переноса

j2:

jl j2

POP ECX

LOOP j1

POP edx

POP ecx

POP ebx

POP eax

RET 8

multiplication endp

START:

PUSH OFFSET number

PUSH 2

CALL multiplication

PUSH 0

CALL ExitProcess

END START

**Результаты работы программы:**



