

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
КАФЕДРА ТЕОРИТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ

Лабораторная работа №1
по дисциплине «Низкоуровневое программирование»

Факультет: ПМИ
Группа: ПМИ-41
Студент: Кислицын И. О.
Преподаватель: Лисицин Д. В.

Новосибирск
2016

1 Цель работы

Изучить и приобрести практические навыки работы с основными командами языка Ассемблера, функциями ввода-вывода, регистрами и символьными данными.

2 Задача

Разработать программу которая принимает на вход два числа в восьмеричной системе счисления, вычисляет их разность и выводит результат в шестнадцатеричной форме.

3 Текст программы

```
.386
.MODEL FLAT, STDCALL
    OPTION CASEMAP: NONE
    EXTERN CharToOemA@8: PROC
    EXTERN WriteConsoleA@20: PROC
    EXTERN ReadConsoleA@20: PROC
    EXTERN GetStdHandle@4: PROC
    EXTERN strlenA@4: PROC
    EXTERN ExitProcess@4: PROC

.DATA
    ; string data
    MSG1 DB Введите число: ,0
    MSG2 DB Введите ещё число: ,0
    ; descriptors
    DIN DD ?
    DOUT DD ?
    ; buffers
    BUF DB 200 dup (?)
    FLAG DB ?
    ; vars
    LENS DD ? ; len of string
    FIRST DD ? ; first operand
    SND DD ? ; second operand
    FIN DD ? ; result
    EIGHT DD 8 ; eight
    HEX DD 10h

.CODE
m_BufToIntOct MACRO buff, dest
    LOCAL CONVERT, FAIL, LEND, NEGATIVE, DONTINV
    MOV ESI, OFFSET buff
    XOR BX, BX
    XOR EAX, EAX
    ; for negative numbers
    MOV BL, [ESI]
    CMP BL, '-'
    JE NEGATIVE
    MOV FLAG, 0
    JMP CONVERT
NEGATIVE:
    MOV FLAG, 1
```

```

    INC ESI
CONVERT:
    MOV BL, [ESI]
    SUB BL, '0'
    CMP BL, 8
    JNB LEND
    MUL EIGHT
    ADD AX, BX
    INC ESI
    JMP CONVERT
LEND:
    CMP FLAG, 0
    JE DONTINV
    XOR EAX, 0FFFFFFFFh
    ADD EAX, 1
DONTINV:
    MOV dest, EAX
ENDM

m_NumToHexPrint MACRO num
    LOCAL CONV, LTEN, ADDCHAR, POSITIVE
    MOV EAX, num
    ; Let's discuss negative numbers
    CMP EAX, 0
    JNL POSITIVE
    XOR EAX, 0FFFFFFFFh
    ADD EAX, 1
    MOV num, EAX ; guardian
    MOV BUF, '-'
    PUSH 0
    PUSH OFFSET LENS
    PUSH 1
    PUSH OFFSET BUF
    PUSH DOUT
    CALL WriteConsoleA@20
    MOV EAX, num ; /guardian
POSITIVE:
    XOR EBX, EBX
CONV:
    CDQ
    DIV HEX
    CMP EDX, 10
    JL LTEN
    SUB EDX, 10
    ADD EDX, 'A'
    JMP ADDCHAR
LTEN:
    ADD EDX, '0'
ADDCHAR:
    PUSH EDX
    INC EBX
    CMP EAX, 0
    JG CONV
PRINT:
    POP EAX
    MOV BUF, AL
    PUSH 0

```

```

PUSH OFFSET LENS
PUSH 1
PUSH OFFSET BUF
PUSH DOUT
CALL WriteConsoleA@20
DEC EBX
CMP EBX, 0
JG PRINT
ENDM

MAIN PROC
; MSG1 & MSG2 to OEM
MOV EAX, OFFSET MSG1
PUSH EAX
PUSH EAX
CALL CharToOemA@8

MOV EAX, OFFSET MSG2
PUSH EAX
PUSH EAX
CALL CharToOemA@8

; get in-/output descs to EAX
PUSH -10
CALL GetStdHandle@4
MOV DIN, EAX
PUSH -11
CALL GetStdHandle@4
MOV DOUT, EAX

; 'please, put the first one'
PUSH OFFSET MSG1
CALL strlenA@4
PUSH 0
PUSH OFFSET LENS
PUSH EAX
PUSH OFFSET MSG1
PUSH DOUT
CALL WriteConsoleA@20

PUSH 0
PUSH OFFSET LENS
PUSH 200
PUSH OFFSET BUF
PUSH DIN
CALL ReadConsoleA@20
; put the num from BUF to int memcell
m_BufToIntOct BUF, FIRST

; 'please, put the second one'
PUSH OFFSET MSG2
CALL strlenA@4
PUSH 0
PUSH OFFSET LENS
PUSH EAX
PUSH OFFSET MSG2
PUSH DOUT

```

```

CALL WriteConsoleA@20

PUSH 0
PUSH OFFSET LENS
PUSH 200
PUSH OFFSET BUF
PUSH DIN
CALL ReadConsoleA@20
; put the num from BUF to int memcell
m_BufToIntOct BUF, SND

MOV EAX, FIRST
SUB EAX, SND
MOV FIN, EAX
; print result
m_NumToHexPrint FIN
EXIT:
PUSH 0
CALL ExitProcess@4
MAIN ENDP

END MAIN

```

4 Тесты

№	in	out
1	$300_8 \ 300_8$	0
2	$300_8 - 300_8$	180_{16}
3	$0_8 \ 100_8$	-40_{16}
4	$777_8 \ 0_8$	$1FF_{16}$