МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ МОЛДОВЫ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ МОЛДОВЫ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ, ИНФОРМАТИКИ И МИКРОЭЛЕКТРОНИКИ

КАФЕДРА АВТОМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЕТ

По лабораторной работе №4 по дисциплине

Архитектура компьютеров

По теме:

Процедуры и макросы

Выполнил: студент группы TI – 155 Буянов Евгений

Проверил: Колесник Виктор

Кишинев 2017

**Тема:** Процедуры и макросы.

**Цель работы:** Работа рассматривает процедуры и макрокоманды,команды сдвига, логические команды и несколько примеров их применения.

**Варианты:** 3, 7

**Программа 4.6.1**

**Условие**

Следующая программа выполняет последовательный поиск первого пробела в ASCII последовательности **pos.** Если пробел не найден в EAX записывается -1, а если найден запишется номер позиции пробела в последовательности. Последовательность **pos** имеет **l** символов.

**Код**

INCLUDE Irvine32.inc

.data

pos DB 'This\_is\_a\_sequence!'

l EQU sizeof pos ; эквивалент размеру смещения pos

.code

main proc

mov **ecx**,l

mov **esi**,0

mov **al**,' '

sld:

inc **esi**

cmp **al**,pos[**esi**]

loopne sld

jne ne\_naiden ;переход к метке ne\_naiden если при сравнении отриц результат

mov **eax**,l

sub **eax**,**ecx**

jmp vihod

ne\_naiden: mov **eax**,-1

vihod: nop ; ничего не делает, оператор 3х тактов

exit

main ENDP

END main

**Листинг**

INCLUDE Irvine32.inc

C ; Include file for Irvine32.lib (Irvine32.inc)

C

C ;OPTION CASEMAP:NONE ; optional: make identifiers case-sensitive

C

C INCLUDE SmallWin.inc ; MS-Windows prototypes, structures, and constants

C .NOLIST

C .LIST

C

C INCLUDE VirtualKeys.inc

C ; VirtualKeys.inc

C .NOLIST

C .LIST

C

C

C .NOLIST

C .LIST

C

00000000 .data

00000000 54 68 69 73 5F pos DB 'This\_is\_a\_sequence!'

69 73 5F 61 5F

73 65 71 75 65

6E 63 65 21

= 00000013 l EQU sizeof pos

00000000 .code

00000000 main proc

00000000 B9 00000013 mov ecx,l

00000005 BE 00000000 mov esi,0

0000000A B0 20 mov al,' '

0000000C sld:

0000000C 46 inc esi

0000000D 3A 86 00000000 R cmp al,pos[esi]

00000013 E0 F7 loopne sld

00000015 75 09 jne ne\_naiden

00000017 B8 00000013 mov eax,l

0000001C 2B C1 sub eax,ecx

0000001E EB 05 jmp vihod

00000020 B8 FFFFFFFF ne\_naiden: mov eax,-1

00000025 90 vihod: nop

exit

00000026 6A 00 \* push +000000000h

00000028 E8 00000000 E \* call ExitProcess

0000002D main ENDP

END main

**Программа 4.6.2**

**Условие**

Следующая программа выполняет сложение двух чисел в упакованном BCD формате. Размерность чисел **n** байт.

**Код**

INCLUDE Irvine32.inc

.DATA

bcd1 DB 22h,41h,75h,32h ; 32754122

bcd2 DB 31h,27h,53h,62h ; 62532731

n EQU lengthof bcd2 ; возвращает число байт последовательности

.CODE

main PROC

lea **esi**,bcd1 ; получение смещения источника

lea **edi**,bcd2

mov **ecx**,n

clc ; Carry Flag = 0

cycle:

mov **al**,[**esi**]

adc **al**,[**edi**]

daa ;установка carry Flag в зависимости от результата

mov [**edi**],**al**

inc **esi**

inc **edi**

loop cycle

exit

main ENDP

END main

**Листинг**

INCLUDE Irvine32.inc

C ; Include file for Irvine32.lib (Irvine32.inc)

C

C ;OPTION CASEMAP:NONE ; optional: make identifiers case-sensitive

C

C INCLUDE SmallWin.inc ; MS-Windows prototypes, structures, and constants

C .NOLIST

C .LIST

C

C INCLUDE VirtualKeys.inc

C ; VirtualKeys.inc

C .NOLIST

C .LIST

C

C

C .NOLIST

C .LIST

C

00000000 .DATA

00000000 22 41 75 32 bcd1 DB 22h,41h,75h,32h ; 32754122

00000004 31 27 53 62 bcd2 DB 31h,27h,53h,62h ; 62532731

= 00000004 n EQU lengthof bcd2

00000000 .CODE

00000000 main PROC

00000000 8D 35 00000000 R lea esi,bcd1

00000006 8D 3D 00000004 R lea edi,bcd2

0000000C B9 00000004 mov ecx,n

00000011 F8 clc

00000012 cycle:

00000012 8A 06 mov al,[esi]

00000014 12 07 adc al,[edi]

00000016 27 daa

00000017 88 07 mov [edi],al

00000019 46 inc esi

0000001A 47 inc edi

0000001B E2 F5 loop cycle

exit

0000001D 6A 00 \* push +000000000h

0000001F E8 00000000 E \* call ExitProcess

00000024 main ENDP

END main

**Программа 4.6.3**

**Условие**

Следующая программа выполняет сложение двух чисел представленных в виде двух последовательностей ASCII символов.

**Код**

INCLUDE Irvine32.inc

.DATA

asc1 DB '7','5','2','1','3'; 31257

asc2 DB '6','1','9','9','5'; 59916

l EQU sizeof asc1 ; эквивалент размеру массива asc1

.CODE

main PROC

lea **esi**,asc1

lea **edi**,asc2

mov **ecx**,l

clc

cycle:

mov **al**,[**esi**]

adc **al**,[**edi**]

aaa ;asci коррекция после сложения

mov [**edi**],**al ;пересылка содержимого регистра**

inc **esi**

inc **edi**

loop cycle

exit

main ENDP

END main

**Листинг**

INCLUDE Irvine32.inc

C ; Include file for Irvine32.lib (Irvine32.inc)

C

C ;OPTION CASEMAP:NONE ; optional: make identifiers case-sensitive

C

C INCLUDE SmallWin.inc ; MS-Windows prototypes, structures, and constants

C .NOLIST

C .LIST

C

C INCLUDE VirtualKeys.inc

C ; VirtualKeys.inc

C .NOLIST

C .LIST

C

C

C .NOLIST

C .LIST

C

00000000 .DATA

00000000 37 35 32 31 33 asc1 DB '7','5','2','1','3'; 31257

00000005 36 31 39 39 35 asc2 DB '6','1','9','9','5'; 59916

= 00000005 l EQU sizeof asc1

00000000 .CODE

00000000 main PROC

00000000 8D 35 00000000 R lea esi,asc1

00000006 8D 3D 00000005 R lea edi,asc2

0000000C B9 00000005 mov ecx,l

00000011 F8 clc

00000012 cycle:

00000012 8A 06 mov al,[esi]

00000014 12 07 adc al,[edi]

00000016 37 aaa

00000017 88 07 mov [edi],al

00000019 46 inc esi

0000001A 47 inc edi

0000001B E2 F5 loop cycle

exit

0000001D 6A 00 \* push +000000000h

0000001F E8 00000000 E \* call ExitProcess

00000024 main ENDP

END main

**Исходные коды .asm**

**Задание 1**

Напишите программу, которая переводит некоторое число из 16-ти цифр в упакованном BCD формате расположенного по адресу **packed** в число в неупакованном BCD формате по адресу **unpacked**. Последовательности вывести на экран в двоичном коде.

**Код**

INCLUDE Irvine32.inc

pack MACRO

mov **ecx**, 8

mov **edx**,0

mov **ebx**,0

loop1:

mov **al**,packed[**ebx**]

shr **al**,4

mov byte ptr unpacked[**edx**],**al**

mov **al**,packed[**ebx**]

shl **al**,4

shr **al**,4

mov byte ptr unpacked[**edx**+1],**al**

add **edx**,2

inc **ebx**

loop loop1

ENDM

.data

packed db 12h, 23h, 34h, 45h, 56h, 67h, 78h, 89h

unpacked dw 8 DUP(0)

.code

main PROC

pack

mov **ecx**,8

mov **edx**,0

mov **ebx**,0

loop2:

movzx **eax**,packed[**ebx**]

call WriteHex

mov **al**,45

call WriteChar

call WriteChar

movzx **eax**, unpacked[**edx**]

call WriteBin

call Crlf

add **edx**,2

inc **ebx**

loop loop2

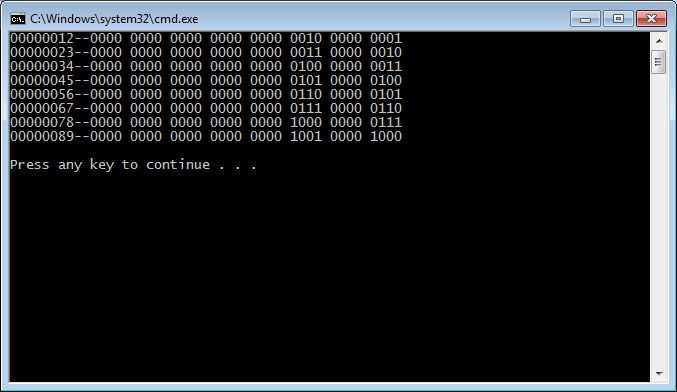
call Crlf

exit

main ENDP

END main

**Результат работы программы**



**Задание 2**

С клавиатуры вводится последовательность из 10-ти символов, напишите программу, которая выводит на экран 20 строк по 75 символов в строке, содержащие введенную последовательность.

**Код**

INCLUDE Irvine32.inc

print MACRO

mov **ecx**,20

loopOuter:

push **ecx**

mov **ecx**,75

mov **edx**,0

loopInner:

mov **al**,string[**edx**]

call WriteChar

inc **edx**

cmp **edx**,10

jnz ifend

mov **edx**,0

ifend:

loop loopInner

call Crlf

pop **ecx**

loop loopOuter;

ENDM

.data

string db 10 DUP(?)

msg db "Symbol ",0

colon db ':'

.code

main PROC

mov **ecx**,10

loop1:

mov **edx**, OFFSET msg

call WriteString

mov **eax**,11

sub **eax**,**ecx**

call WriteDec

mov **al**,colon

call WriteChar

call ReadChar

call WriteChar

mov **edx**,10

sub **edx**,**ecx**

mov string[**edx**],**al**

call Crlf

loop loop1;

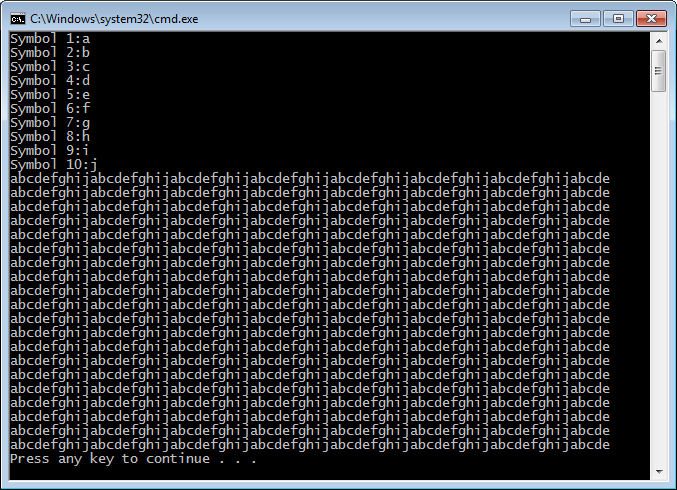
print

exit

main ENDP

END main

**Результат работы программы**



**Выводы**

В данной лабораторной работе были укреплены и дополнительно изучены некоторые аспекты языка ассемблер. В процессе выполнения работы были изучены методы и принципы использования процедур и макропроцедур в языке ассемблер. В процессе выполнения лабораторной работы были написаны 2 программы:

* Программа, которая переводит некоторое число из 16-ти цифр в упакованном BCD формате в число в неупакованном BCD формате и выводящая полученные последовательности на экран в двоичном коде.
* Программа, для которой с клавиатуры вводится последовательность из 10-ти символов, и выводящая на экран 20 строк по 75 символов в строке, содержащие введенную последовательность.

Написанный код показал, насколько сильно языки высокого уровня упрощают процесс разработки приложений, хотя языки высокого уровня и компилируются в машинный код и выполняют одну и ту же работу, хотя процедуры и макропроцедуры позволяют сократить код.