МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №7**

**з дисципліни «архітектура комп’ютера»**

**на тему: «Модульне програмування мовою Асемблера.»**

Виконав: студент гр. ПЗ181х

П І Б

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2019

Лабораторна робота 7

Мета.Модульне програмування мовою Асемблера.

Тема. Одержати навички використання процедур мовою Асемблера.

Постановка задачи

Загальне завдання: розробити програми, які виконують опрацювання масиву (завдання лабораторної роботи №6 частина 2, з використанням модулів.

Індивідуальне завдання:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Типи  виклику | Спосіб передачі параметрів | Рівень складності | | | | |
| Е  (достатньо) | D  (задовільно) | C  (добре) | B  (дуже добре) | A  (відмінно) |
| Ближній | Регістровий | + | + | + | + | + |
| Ближній | Через стек |  | + | + | + | + |
| Дальній | Регістровий |  |  | + | + | + |
| Дальній | Через стек |  |  |  | + | + |

**Лістинг**

Turbo Assembler Version 3.0 01/28/20 19:45:57 Page 1

lab7.asm

1 ;-------------------------------------------------------------------------------

2 ;Програма роботи з масивом чисел.

3 ;Елементи масиву – цілі числа зі знаком з діапазону слово.

4 ;Вхід: розмір масиву та його елементи вводяться з клавіатури, число с яким будемо порівнювати +

5 елементи масива.

6 ;Вихід: оновлений масив виводиться на консоль, кількість та сума елементів масиву які більше за+

7 задане число.

8 ;-------------------------------------------------------------------------------

9

10

11 ;сегмент стеку

12 0000 stk segment stack

13 0000 0100\*(??) db 256 dup (?)

14 0100 stk ends

15

16 ;сегмент даних

17 0000 data segment para public 'data'

18 0000 000A len dw 10 ;максимальний розмір масиву

19 0002 ???? counter dw ? ;розмір масиву

20 0004 05 division db 5

21 0005 0000 sum dw 0

22 0007 0000 quantity dw 0

23 0009 0A\*(????) A dw 10 dup (?) ;масив

24 001D ???? Comparison\_Number dw ? ;число с яким будемо порівнювати елементи масива(число должен вводить +

25 пользеватель)

26 ;буфер для введення числа в ASCII-формі (максимум 6 символів (-32768) + кінець рядка (0dh))

27 001F 07 ?? 07\*(??) buffer db 7, ?, 7 dup (?)

28 0028 07 ?? 07\*(??) buffer2 db 7, ?, 7 dup (?)

29 0031 4E 75 6D 62 65 72 20+ msgQuantity db 'Number of items greater than the number entered: $'

30 6F 66 20 69 74 65 6D+

31 73 20 67 72 65 61 74+

32 65 72 20 74 68 61 6E+

33 20 74 68 65 20 6E 75+

34 6D 62 65 72 20 65 6E+

35 74 65 72 65 64 3A 20+

36 24

37 0063 54 68 65 20 73 75 6D+ msgPrintsum db 'The sum of the elements that is greater than the number entered: $'

38 20 6F 66 20 74 68 65+

39 20 65 6C 65 6D 65 6E+

40 74 73 20 74 68 61 74+

41 20 69 73 20 67 72 65+

42 61 74 65 72 20 74 68+

43 61 6E 20 74 68 65 20+

44 6E 75 6D 62 65 72 20+

45 65 6E 74 65 72 65 64+

46 3A 20 24

47 00A5 45 6E 74 65 72 20 74+ msgCounter db 'Enter the number of elements in the array N (5 <= N <= 10): $'

48 68 65 20 6E 75 6D 62+

49 65 72 20 6F 66 20 65+

50 6C 65 6D 65 6E 74 73+

51 20 69 6E 20 74 68 65+

52 20 61 72 72 61 79 20+

53 4E 20 28 35 20 3C 3D+

54 20 4E 20 3C 3D 20 31+

55 30 29 3A 20 24

56 00E2 45 6E 74 65 72 20 61+ msgArray db 'Enter array elements:', 0ah, 0dh, '$'

57 72 72 61 79 20 65 6C+

58 65 6D 65 6E 74 73 3A+

59 0A 0D 24

60 00FA 0A 0D 45 52 52 4F 52+ msgError db 0ah, 0dh, 'ERROR!', 0ah, 0dh, '$'

61 21 0A 0D 24

62 0105 41 72 72 61 79 20 3A+ msgPrint db 'Array :', 0ah, 0dh, '$'

63 0A 0D 24

64 010F 50 72 65 73 73 20 61+ msgPause db 'Press any key ...', '$'

65 6E 79 20 6B 65 79 20+

66 2E 2E 2E 24

67 0121 0A 0D 24 CrLf db 0ah, 0dh, '$'

68 0124 data ends

69

70 extrn outString:near

71 extrn outPrompt:near

72 extrn convertStrToInt:near

73 extrn printInt:near

74 ;сегмент команд

75 0000 code segment para public 'code'

76 assume cs:code, ds:data, ss:stk

77 0000 main proc far

78 0000 B8 0000s mov ax, data

79 0003 8E D8 mov ds, ax

80

81 0005 mInCounter1:

82 0005 BA 0031r lea dx, msgQuantity ;виведення запрошення для введення числа

83 0008 E8 0000e call outString

84 000B BA 0028r lea dx, buffer2 ;DX = адреса буфера

85 000E B4 0A mov ah, 0ah ;введення рядка в буфер

86 0010 CD 21 int 21h

87 0012 80 3E 0029r 00 cmp buffer2 + 1, 0 ;перевірка довжини рядка

88 0017 E8 0000e call convertStrToInt ;перетворення рядка в слово (число зі знаком)

89 001A 72 0C jc mErrorCounter1 ;якщо помилка, то виведення повідомлення

90 001C A3 001Dr mov Comparison\_Number, ax

91 001F BA 0121r lea dx, CrLf

92 0022 E8 0000e call outString

93 0025 EB 09 90 jmp mEndLoop1

94 0028 mErrorCounter1:

95 0028 BA 00FAr lea dx, msgError

96 002B E8 0000e call outString ;виведення повідомлення 'ERROR!'

97 002E EB D5 jmp mInCounter1 ;перехід до початку циклу

98 0030 mEndLoop1:

99

100 ;введення кількості елементів масиву (цикл повторюється до введення "правильного" числа)

101 0030 mInCounter:

102 0030 BA 00A5r lea dx, msgCounter ;виведення запрошення для введення числа

103 0033 E8 0000e call outString

104 0036 BA 001Fr lea dx, buffer ;DX = адреса буфера

105 0039 B4 0A mov ah, 0ah ;введення рядка в буфер

106 003B CD 21 int 21h

107 003D 80 3E 0020r 00 cmp buffer + 1, 0 ;перевірка довжини рядка

108 0042 74 EC jz mInCounter ;якщо довжина = 0, перехід до початку циклу

109 0044 E8 0000e call convertStrToInt ;перетворення рядка в слово (число зі знаком)

110 0047 72 17 jc mErrorCounter ;якщо помилка, то виведення повідомлення

111 0049 3B 06 0000r cmp ax, len ;якщо кількість елементів > розміру масиву,

112 004D 7F 11 jg mErrorCounter ;то перехід на повідомлення про помилку

113 004F 3D 0004 cmp ax, 4 ;якщо кількість елементів <= 4,

114 0052 7E 0C jle mErrorCounter ;то перехід на повідомлення про помилку

115 0054 A3 0002r mov counter, ax ;збереження кількості елементів масиву

116 0057 BA 0121r lea dx, CrLf ;виведення управляючих символів

117 005A E8 0000e call outString

118 005D EB 09 90 jmp mEndLoop ;вихід з циклу введення кількості елементів

119 0060 mErrorCounter:

120 0060 BA 00FAr lea dx, msgError

121 0063 E8 0000e call outString ;виведення повідомлення 'ERROR!'

122 0066 EB C8 jmp mInCounter ;перехід до початку циклу

123 0068 mEndLoop:

124 ;введення елементів масиву

125 0068 BA 00E2r lea dx, msgArray ;виведення запрошення для елементів масиву

126 006B E8 0000e call outString

127 006E 8B 0E 0002r mov cx, counter ;лічильник циклу введення елементів масиву

128 0072 33 F6 xor si, si ;індекс елементів масиву

129 ;введення елементу масиву

130 0074 mInArray:

131 0074 E8 0000e call outPrompt ;виведення запрошення для елемента масиву

132 0077 BA 001Fr lea dx, buffer ;DX = адреса буфера

133 007A B4 0A mov ah, 0ah ;введення рядка в буфер

134 007C CD 21 int 21h

135 007E 80 3E 0020r 00 cmp buffer + 1, 0 ;перевірка довжини рядка

136 0083 74 EF jz mInArray ;якщо довжина = 0, перехід до початку циклу

137 0085 E8 0000e call convertStrToInt ;перетворення рядка в число зі знаком (слово)

138 0088 73 08 jnc mElementOk ;якщо немає помилки, то збереження елементу

139 008A BA 00FAr lea dx, msgError ;інакше виведення повідомлення 'ERROR!'

140 008D E8 0000e call outString

141 0090 EB E2 jmp mInArray ;перехід до початку циклу

142 0092 mElementOk:

143 0092 8B FE mov di, si ;DI = індекс поточного елементу масиву

144 0094 03 FF add di, di ;DI = зміщення поточного елементу

145 0096 89 85 0009r mov A[di], ax ;збереження поточного елементу масиву

146 009A 46 inc si ;перехід до наступного елементу масиву

147 009B BA 0121r lea dx, CrLf ;виведення управляючих символів

148 009E E8 0000e call outString

149 00A1 E2 D1 loop mInArray ;якщо не всі елементи введені - до початку циклу

150 ;виведення масиву на консоль

151 00A3 BA 0105r lea dx, msgPrint ;виведення повідомлення

152 00A6 E8 0000e call outString

153 00A9 8B 0E 0002r mov cx, counter ;лічильник циклу виведення елементів масиву

154 00AD 33 F6 xor si, si ;індекс поточного елементу масиву A

155 00AF mPrintArray:

156 00AF E8 0000e call outPrompt ;виведення "імені" елемента масиву

157 00B2 8B FE mov di, si ;DI = індекс поточного елементу масиву

158 00B4 03 FF add di, di ;DI = зміщення поточного елементу

159 00B6 8B 85 0009r mov ax, A[di] ;AX = поточний елемент

160 00BA E8 0000e call printInt ;виведення поточного елементу

161 00BD BA 0121r lea dx, CrLf ;виведення управляючих символів

162 00C0 E8 0000e call outString

163 00C3 46 inc si ;перехід до наступного елементу масиву

164 00C4 E2 E9 loop mPrintArray ;не всі елементи виведені - на початок циклу

165 00C6 33 FF xor di, di

166 00C8 8B 0E 0002r mov cx, counter

167 00CC BB 0000 mov bx, 0

168

169 00CF mCondition:

170 00CF 8B 85 0009r mov ax, A[di]

171 00D3 3B 06 001Dr cmp ax, Comparison\_Number

172 00D7 7C 08 jl mNext

173 00D9 03 9D 0009r add bx, A[di]

174 00DD FF 06 0007r inc quantity

175 00E1 mNext:

176 00E1 47 inc di

177 00E2 47 inc di

178 00E3 E2 EA loop mCondition

179

180 00E5 mPrintsum:

181 00E5 BA 0063r lea dx, msgPrintsum

182 00E8 E8 0000e call outString

183 00EB 8B C3 mov ax, bx

184 00ED E8 0000e call printInt

185 00F0 BA 0121r lea dx, CrLf ;виведення управляючих символів

186 00F3 E8 0000e call outString

187 00F6 BA 0031r lea dx, msgQuantity

188 00F9 E8 0000e call outString

189 00FC A1 0007r mov ax, quantity

190 00FF E8 0000e call printInt

191 0102 BA 0121r lea dx, CrLf ;виведення управляючих символів

192 0105 E8 0000e call outString

193 ;затримка виконання та вихід з програми

194 0108 BA 010Fr lea dx, msgPause

195 010B E8 0000e call outString ;виведення повідомлення 'Press any key...'

196 010E B4 08 mov ah, 08h ;затримка виконання програми

197 0110 CD 21 int 21h

198 0112 B8 4C00 mov ax, 4C00h ;завершення програми

199 0115 CD 21 int 21h

200 0117 main endp

201 0117 code ends

202 end main

Symbol Name Type Value

??DATE Text "01/28/20"

??FILENAME Text "lab7 "

??TIME Text "19:45:57"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text LAB7

@WORDSIZE Text 2

A Word DATA:0009

BUFFER Byte DATA:001F

BUFFER2 Byte DATA:0028

COMPARISON\_NUMBER Word DATA:001D

CONVERTSTRTOINT Near ----:---- Extern

COUNTER Word DATA:0002

CRLF Byte DATA:0121

DIVISION Byte DATA:0004

LEN Word DATA:0000

MAIN Far CODE:0000

MCONDITION Near CODE:00CF

MELEMENTOK Near CODE:0092

MENDLOOP Near CODE:0068

MENDLOOP1 Near CODE:0030

MERRORCOUNTER Near CODE:0060

MERRORCOUNTER1 Near CODE:0028

MINARRAY Near CODE:0074

MINCOUNTER Near CODE:0030

MINCOUNTER1 Near CODE:0005

MNEXT Near CODE:00E1

MPRINTARRAY Near CODE:00AF

MPRINTSUM Near CODE:00E5

MSGARRAY Byte DATA:00E2

MSGCOUNTER Byte DATA:00A5

MSGERROR Byte DATA:00FA

MSGPAUSE Byte DATA:010F

MSGPRINT Byte DATA:0105

MSGPRINTSUM Byte DATA:0063

MSGQUANTITY Byte DATA:0031

OUTPROMPT Near ----:---- Extern

OUTSTRING Near ----:---- Extern

PRINTINT Near ----:---- Extern

QUANTITY Word DATA:0007

SUM Word DATA:0005

Groups & Segments Bit Size Align Combine Class

CODE 16 0117 Para Public CODE

DATA 16 0124 Para Public DATA

STK 16 0100 Para Stack

Turbo Assembler Version 3.0 01/28/20 19:50:49 Page 1

mod1.asm

1 ;сегмент стеку

2 0000 stk segment stack

3 0000 0100\*(??) db 256 dup (?)

4 0100 stk ends

5

6 ;сегмент даних

7 0000 data segment para public 'data'

8

9 ;

10 0000 41 20 5B 20 24 msgPrompt1 db 'A [ $'

11 0005 20 5D 20 3D 20 24 msgPrompt2 db ' ] = $'

12

13 000B data ends

14

15 public outString

16 public outPrompt

17 public convertStrToInt

18 public printInt

19 ;сегмент команд

20 0000 code segment para public 'code'

21 assume cs:code, ds:data, ss:stk

22

23

24 ;-------------------------------------------------------------------------------

25 ;процедура виведення рядка символів, обмежених +

26 символом '$' на консоль

27 ;вхід: DX - адреса рядка

28 ;-------------------------------------------------------------------------------

29 0000 outString proc near

30 0000 B4 09 mov ah, 09h

31 0002 CD 21 int 21h

32 0004 C3 ret

33 0005 outString endp

34 ;-------------------------------------------------------------------------------

35 ;процедура виведення на консоль імені масиву та +

36 індексу елемента

37 ;-------------------------------------------------------------------------------

38 0005 outPrompt proc near

39 0005 BA 0000r lea dx, msgPrompt1 ;виведення імені масиву

40 0008 E8 FFF5 call outString

41 000B 8B C6 mov ax, si ;AX = індекс елементу масиву

42 000D E8 0068 call printInt ;виведення індексу

43 0010 BA 0005r lea dx, msgPrompt2

44 0013 E8 FFEA call outString

45 0016 C3 ret

46 0017 outPrompt endp

47 ;-------------------------------------------------------------------------------

48 ;процедура перетворення рядка в число зі знаком в +

49 форматі слова

50 ;(діапазон значення числа від -32768 до 32767)

51 ;вхід: DX - адреса буфера

52 ;вихід: AX - число в форматі слова (в разі помилки AX = 0)

53 ; CF = 1 - помилка

54 ;-------------------------------------------------------------------------------

55 0017 convertStrToInt proc near

56 ;збереження регістрів

57 0017 53 push bx

58 0018 51 push cx

59 0019 52 push dx

60 001A 56 push si

61 001B 57 push di

62 001C 55 push bp

63 001D 8B F2 mov si, dx ;SI - адреса буфера

64 001F 33 C9 xor cx, cx

65 0021 8A 4C 01 mov cl, [si + 1] ;CL = довжина введеного рядка

66 ;перевірка знаку числа

67 0024 83 C6 02 add si, 2 ;SI - адреса рядка

68 0027 8A 1C mov bl, [si] ;BL = перший символ рядка

69 0029 80 FB 2D cmp bl, '-' ;порівняння першого +

70 символу з '-'

71 002C 75 03 jne mNoSign ;якщо не '-', то +

72 перетворюємо рядок як число без знаку

73 002E 46 inc si ;інкремент адреси +

74 рядка

75 002F FE C9 dec cl ;декремент довжини +

76 рядка

77 ;перетворення рядка у слово без знаку

78 0031 mNoSign:

79 0031 8B EB mov bp, bx ;зберігаємо регістр

80 0033 33 DB xor bx, bx

81 0035 BF 000A mov di, 10 ;DI = множник 10 (база +

82 системи числення)

83 0038 33 C0 xor ax, ax ;AX = 0, для обчислення +

84 числа, яке введене з клавіатури

85 003A mScanString:

86 003A 8A 1C mov bl, [si] ;завантаження в BL +

87 чергового символу рядка

88 003C 46 inc si ;інкремент адреси

89 003D 80 FB 30 cmp bl, '0' ;якщо код символу менше +

90 коду '0'

91 0040 72 2C jb mErrorValue ;повертаємо помилку

92 0042 80 FB 39 cmp bl, '9' ;якщо код символу більше +

93 коду '9'

94 0045 77 27 ja mErrorValue ;повертаємо помилку

95 0047 80 EB 30 sub bl, '0' ;перетворення +

96 символу-цифри в число

97 004A F7 E7 mul di ;AX = AX \* 10

98 004C 72 20 jc mErrorValue ;якщо результат більше 16 +

99 бітів - помилка

100 004E 03 C3 add ax, bx ;додаємо цифру

101 0050 72 1C jc mErrorValue ;якщо переповнення - +

102 помилка

103 0052 E2 E6 loop mScanString ;продовжуємо сканувати рядок

104 ;обробка знаку числа та перевірка діапазону для +

105 від'ємного числа

106 0054 8B DD mov bx, bp ;відновлюємо регістр

107 0056 80 FB 2D cmp bl, '-' ;знову перевіряємо знак

108 0059 75 0A jne mPlus ;якщо перший символ не '-', +

109 то число додатне

110 005B 3D 8000 cmp ax, 32768 ;модуль від'ємного числа +

111 повинен бути не більш 32768

112 005E 77 0E ja mErrorValue ;якщо більше (без знаку), +

113 повертаємо помилку

114 0060 F7 D8 neg ax ;інвертуємо число +

115 (число - від'ємне)

116 0062 EB 06 90 jmp mOk ;переходимо до +

117 нормального завершення процедури

118 ;перевірка діапазону для додатного числа

119 0065 mPlus:

120 0065 3D 7FFF cmp ax, 32767 ;додатне число повинно +

121 бути не більше 32767

122 0068 77 04 ja mErrorValue ;якщо більше, повертаємо +

123 помилку

124 ;успішне перетворення рядка у число в форматі слова

125 006A mOk:

126 006A F8 clc ;CF = 0 ознака успішного +

127 перетворення рядка у число

128 006B EB 04 90 jmp mExit ;переходимо до виходу з +

129 процедури

130 ;помилка перетворення рядка у число в форматі слова

131 006E mErrorValue:

132 006E 33 C0 xor ax, ax ;AX = 0

133 0070 F9 stc ;CF = 1 повертаємо +

134 помилку

135 0071 mExit:

136 ;відновлення регістрів

137 0071 5D pop bp

138 0072 5F pop di

139 0073 5E pop si

140 0074 5A pop dx

141 0075 59 pop cx

142 0076 5B pop bx

143 0077 C3 ret

144 0078 convertStrToInt endp

145 ;-------------------------------------------------------------------------------

146 ;процедура виведення цілого числа у форматі слова +

147 на консоль

148 ;вхід: AX - ціле число

149 ;-------------------------------------------------------------------------------

150 0078 printInt proc near

151 ;збереження регістрів

152 0078 50 push ax

153 0079 53 push bx

154 007A 51 push cx

155 007B 52 push dx

156 ;обробка знаку числа AX

157 007C 3D 0000 cmp ax, 0 ;якщо число додатне

158 007F 7D 0A jge mPositive ;перехід до його обробки

159 0081 F7 D8 neg ax ;інакше - перетворення +

160 від'ємного числа на додатне

161 0083 50 push ax

162 0084 B2 2D mov dl, '-' ;виведення на консоль

163 0086 B4 02 mov ah, 02h ;знаку '-'

164 0088 CD 21 int 21h

165 008A 58 pop ax

166 ;обробка додатного числа

167 008B mPositive:

168 008B BB 000A mov bx, 10

169 008E 33 C9 xor cx, cx ;лічильник цифр числа

170 ;отримуємо цифри числа AX

171 0090 mDivisionNumber:

Turbo Assembler Version 3.0 01/28/20 19:50:49 Page 4

mod1.asm

172 0090 99 cwd

173 0091 F7 F3 div bx ;ділимо з остачею

174 0093 52 push dx ;зберігаємо остачу в стеку

175 0094 41 inc cx ;збільшуємо лічильник +

176 цифр числа

177 0095 0B C0 or ax, ax ;якщо число ще не 0,

178 0097 75 F7 jnz mDivisionNumber ;продовжуємо цикл

179 ;виведення на консоль цифр числа AX

180 0099 mPrintChar:

181 0099 5A pop dx ;DX = цифра числа

182 009A 80 C2 30 add dl, 30h ;DL = символ цифри числа

183 009D B4 02 mov ah, 02h ;виведення символу-цифри на +

184 консоль

185 009F CD 21 int 21h

186 00A1 E2 F6 loop mPrintChar ;лічильник цифр числа не 0 - +

187 продовжуємо цикл

188 ;відновлення регістрів

189 00A3 5A pop dx

190 00A4 59 pop cx

191 00A5 5B pop bx

192 00A6 58 pop ax

193 00A7 C3 ret

194 00A8 printInt endp

195 ;-------------------------------------------------------------------------------

196 00A8 code ends

197 end

Symbol Name Type Value

??DATE Text "01/28/20"

??FILENAME Text "mod1 "

??TIME Text "19:50:49"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text MOD1

@WORDSIZE Text 2

CONVERTSTRTOINT Near CODE:0017

MDIVISIONNUMBER Near CODE:0090

MERRORVALUE Near CODE:006E

MEXIT Near CODE:0071

MNOSIGN Near CODE:0031

MOK Near CODE:006A

MPLUS Near CODE:0065

MPOSITIVE Near CODE:008B

MPRINTCHAR Near CODE:0099

MSCANSTRING Near CODE:003A

MSGPROMPT1 Byte DATA:0000

MSGPROMPT2 Byte DATA:0005

OUTPROMPT Near CODE:0005

OUTSTRING Near CODE:0000

PRINTINT Near CODE:0078

Groups & Segments Bit Size Align Combine Class

CODE 16 00A8 Para Public CODE

DATA 16 000B Para Public DATA

STK 16 0100 Para Stack

Висновок: Параметри процедури – передані ззовні дані, з якими буде працювати процедура.

Існують такі варіанти передачі параметрів в процедуру:

-через регістри;

-через спільну область пам'яті;

-через стек;

-за допомогою директив EXTRN і PUBLIC.