МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №8**

**з дисципліни «Архітектура комп'ютера»**

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

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2020

Тема. Модульне програмування мовою Асемблера.

Мета. Одержати навички використання процедур мовою Асемблера.

Варіант 5

Розробити програму для опрацювання масиву з використанням модулів. Для кожного завдання розробити:

1. модуль інтерфейсу, який включає:
2. оголошення змінних вхідних і вихідних даних програми;
3. введення даних;
4. підготовка до виклику і виклик першої процедури;
5. відображення результатів опрацювання;
6. підготовка до виклику і виклик другої процедури;
7. відображення результатів опрацювання.
8. два модуля логіки, які виконують обробку масиву відповідно до типу виклику та способу передачі параметрів та повертають результати в модуль інтерфейсу.

Перша процедура:

1. тип: ближня;
2. передача параметрів: через регістри;
3. вхідні дані: адреса рядка, довжина рядка, додаткові дані по завданню – символ(и);
4. вихідні дані: результат обробки рядка.

Друга процедура:

1. тип: дальня;
2. передача параметрів: через стек;
3. вхідні дані: адреса рядка, довжина рядка, додаткові дані по завданню – символ(и), адреса результату обробки рядка.

В модулях логіки реалізувати завдання лабораторної роботи №6.

**1. Лістинг**

Модуль з ближньою процедурою

Turbo Assembler Version 3.0 12/25/20 01:40:28 Page 1

ak8.asm

1 ;Модуль інтерфейсу

2 ;Програма рахуэ кылькысть слыв, довжина яких менша +

3 ныж 3 символи

4 ;Вхідні дані: рядок символів

5 ;Вихідні дані: кількість слів, символів у яких менше +

6 4-х

7 ;=======================================================================================

8 ;сегмент стека

9 0000 stk segment stack

10 0000 80\*(??) db 128 dup(?)

11 0080 stk ends

12

13 ;сегмент даних

14 0000 data segment para public 'data'

15 0000 50 buffer db 80

16 0001 ?? text db ?

17 0002 50\*(20) string db 80 dup (' ')

18 0052 45 6E 74 65 72 20 73+ msg\_prompt db 'Enter string : $'

19 74 72 69 6E 67 20 3A+

20 20 24

21 0062 52 65 73 75 6C 74 20+ msg\_result db 'Result : $'

22 3A 20 24

23 006C 00 slovo db 0

24 006D 00 slovo2 db 0

25 006E 0A 0D 24 crlf db 0ah, 0dh, '$'

26 0071 data ends

27

28 extrn modak8:far

29

30 ;сегмент команд

31 0000 code segment para public 'code'

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

33 0000 begin:

34 0000 B8 0000s mov ax, data

35 0003 8E D8 mov ds, ax

36 ;=============Ввод рядка=================

37 0005 in\_text:

38 0005 BA 0052r lea dx, msg\_prompt ;виведення запрошення

39 0008 E8 008B call printSting

40 000B BA 0000r lea dx, buffer ;запис рядка в буфер

41 000E B4 0A mov ah, 0ah

42 0010 CD 21 int 21h

43 0012 F6 06 0001r FF test text, 0FFh ;якщо введений рядок порожній,

44 0017 74 EC jz in\_text ;то перехід на повторне +

45 введення рядка

46 0019 BA 006Er lea dx, crlf

47 001C E8 0077 call printSting

48 ;=============Підготовка виклику calculating==============

49 001F 8A 0E 0001r mov cl, text

50 0023 BE 0002r lea si, string

51 0026 BB 006Cr lea bx, slovo

52 ;============Виклик calculating======================

53 0029 E8 0043 call calculating

54 002C BA 0062r lea dx, msg\_result ;вивід результату на екран

55 002F E8 0064 call printSting

56 0032 A0 006Cr mov al, slovo

57 0035 05 0030 add ax, 30h

\_Turbo Assembler Version 3.0 12/25/20 01:40:28 Page 2

ak8.asm

58 0038 8A D0 mov dl, al

59 003A B4 02 mov ah, 02h

60 003C CD 21 int 21h

61 003E BA 006Er lea dx, crlf

62 0041 E8 0052 call printSting

63

64 ;============Підготовка виклику modak8================

65 0044 33 C9 xor cx, cx

66 0046 8A 0E 0001r mov cl, text

67 004A 51 push cx

68 004B BE 0002r lea si, string

69 004E 56 push si

70

71 ;===========Виклик modak8===================

72 004F 9A 00000000se call modak8

73 0054 88 16 006Dr mov slovo2, dl

74 0058 BA 0062r lea dx, msg\_result ;вивід результату на екран

75 005B E8 0038 call printSting

76 005E A0 006Dr mov al, slovo2

77 0061 05 0030 add ax, 30h

78 0064 8A D0 mov dl, al

79 0066 B4 02 mov ah, 02h

80 0068 CD 21 int 21h

81 ;===========Заверщення програми================

82 006A B8 4C00 mov ax, 4C00h

83 006D CD 21 int 21h

84 ;===============================================

85 ;Процедура підрахунку слів довжиною <3 символів

86 ;Тип ближня

87 ;вхід:

88 ;cx - довжина масиву

89 ;si - індекс

90 ;вихід

91 ;slovo - лічильник слів

92 ;===============================================

93 006F calculating proc near

94 006F 33 F6 xor si, si ;індекс

95 0071 33 DB xor bx, bx ;bx = counter

96 0073 while\_begin:

97 0073 80 BC 0002r 20 cmp string[si], 20h ;порівння поточного символу з +

98 пробілом

99 0078 75 0E jne not\_space ;якщо символ не пробіл, +

100 перейти до мітки not\_space

101 007A 83 FB 03 cmp bx, 03h ;інакше порівняти counter та 3

102 007D 7F 04 jg next ;якщо counter > 3, перейти до +

103 мітки next

104 007F FE 06 006Cr inc slovo ;збільшити лічильник слів

105 0083 next:

106 0083 33 DB xor bx, bx ;counter = 0

107 0085 46 inc si ;збільшити індекс

108 0086 E2 EB loop while\_begin ;якщо не кінець рядка, перейти +

109 на початок циклу

110 0088 not\_space:

111 0088 43 inc bx ;збыльшити counter на 1

112 0089 46 inc si ;збільшити індекс на 1

113 008A E2 E7 loop while\_begin

114 008C 83 FB 03 cmp bx, 03h ;порівняти counter та 3

\_Turbo Assembler Version 3.0 12/25/20 01:40:28 Page 3

ak8.asm

115 008F 7F 04 jg next2 ;якщо counter > 3, перейти до +

116 мітки next2

117 0091 FE 06 006Cr inc slovo

118 0095 next2:

119 0095 C3 ret

120 0096 calculating endp

121

122 ;Процедура виводу рядка на екран

123 0096 printSting proc near

124 0096 B4 09 mov ah, 09h

125 0098 CD 21 int 21h

126 009A C3 ret

127 009B printSting endp

128 009B code ends

129 end begin

\_Turbo Assembler Version 3.0 12/25/20 01:40:28 Page 4

Symbol Table

Symbol Name Type Value

??DATE Text "12/25/20"

??FILENAME Text "ak8 "

??TIME Text "01:40:28"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text AK8

@WORDSIZE Text 2

BEGIN Near CODE:0000

BUFFER Byte DATA:0000

CALCULATING Near CODE:006F

CRLF Byte DATA:006E

IN\_TEXT Near CODE:0005

MODAK8 Far ----:---- Extern

MSG\_PROMPT Byte DATA:0052

MSG\_RESULT Byte DATA:0062

NEXT Near CODE:0083

NEXT2 Near CODE:0095

NOT\_SPACE Near CODE:0088

PRINTSTING Near CODE:0096

SLOVO Byte DATA:006C

SLOVO2 Byte DATA:006D

STRING Byte DATA:0002

TEXT Byte DATA:0001

WHILE\_BEGIN Near CODE:0073

Groups & Segments Bit Size Align Combine Class

CODE 16 009B Para Public CODE

DATA 16 0071 Para Public DATA

STK 16 0080 Para Stack

\_

Модуль з дальньою процедурою

Turbo Assembler Version 3.0 12/25/20 01:52:46 Page 1

modak8.asm

1 0000 stk segment stack stack

2 0000 80\*(??) db 128 dup(?)

3 0080 stk ends

4

5 public modak8

6

7 0000 code segment para public 'code'

8 assume cs:code, ss:stk

9

10 ;===============================================

11 ;Процедура підрахунку слів довжиною <3 символів

12 ;Тип дальня

13 ;вхід:

14 ;cx - довжина масиву

15 ;si - індекс

16 ;вихід

17 ;dx - лічильник слів

18 ;===============================================

19 0000 modak8 proc far

20 0000 55 push bp

21 0001 8B EC mov bp, sp

22 0003 8B 76 06 mov si, [bp+6]

23 0006 8B 4E 08 mov cx, [bp+8]

24 0009 33 C0 xor ax, ax

25 000B 33 DB xor bx, bx

26 000D 33 D2 xor dx, dx

27 000F while\_begin:

28 000F 8A 04 mov al, [si]

29 0011 3C 20 cmp al, 20h ;порівння поточного символу з +

30 пробілом

31 0013 75 0B jne not\_space ;якщо символ не пробіл, перейти до +

32 мітки not\_space

33 0015 83 FB 03 cmp bx, 03h ;інакше порівняти counter та 3

34 0018 7F 01 jg next ;якщо counter > 3, перейти до мітки +

35 next

36 001A 42 inc dx ;збільшити лічильник слів

37 001B next:

38 001B 33 DB xor bx, bx ;counter = 0

39 001D 46 inc si ;збільшити індекс

40 001E E2 EF loop while\_begin ;якщо не кінець рядка, перейти +

41 на початок циклу

42 0020 not\_space:

43 0020 43 inc bx ;збыльшити counter на 1

44 0021 46 inc si ;збільшити індекс на 1

45 0022 E2 EB loop while\_begin

46 0024 83 FB 03 cmp bx, 03h ;порівняти counter та 3

47 0027 7F 01 jg next2 ;якщо counter > 3, перейти до +

48 мітки next2

49 0029 42 inc dx

50 002A next2:

51 002A 8B E5 mov sp, bp

52 002C 5D pop bp

53 002D CB ret

54 002E modak8 endp

55 002E code ends

56 end

\_Turbo Assembler Version 3.0 12/25/20 01:52:46 Page 2

Symbol Table

Symbol Name Type Value

??DATE Text "12/25/20"

??FILENAME Text "modak8 "

??TIME Text "01:52:46"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text MODAK8

@WORDSIZE Text 2

MODAK8 Far CODE:0000

NEXT Near CODE:001B

NEXT2 Near CODE:002A

NOT\_SPACE Near CODE:0020

WHILE\_BEGIN Near CODE:000F

Groups & Segments Bit Size Align Combine Class

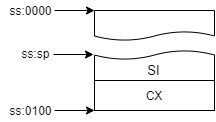
CODE 16 002E Para Public CODE

STK 16 0080 Para Stack

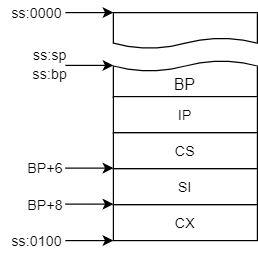
\_

**2. Стан стеку**

1. Стек перед викликом процедури

****

2. Стек після виклику процедури



**3. Висновок**

Під час виконання лабораторної роботи працював з процедурами. Процедура може бути дальнього та ближнього типу виклику. Процедуру ближнього типу виклику можна оголосити в кінці сегмента коду. Для процедури дальнього типу виклику у головному модулі необхідно прописати директиву extrn та директиву public в іншому модулі з процедурою. Виклик процедур виконується за допомогою команди CALL, під час виконання якої, у стек записується адрес значення регістру СS:IP наступної команди після команди CALL. Для передачі параметрів у процедуру використовують регістри або стек.