

Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»

Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

*Звіти до комп'ютерних практикумів з модуля
«Системне програмування»*

Прийняв
доцент кафедри ТК
Лісовиченко О.І.

Виконав
Студент групи ІТ-61
Тимченко О. Ю.

Комп'ютерний практикум №5

Тема: Макрозасоби мови асемблер

Завдання:

Скласти програму на нижче наведені завдання:

- 1) переписати Програму 2.1 з використанням макросів;
- 2) переписати Програму 3.1, з використанням макросів;
- 3) переписати Програму 4.1 з використанням макросів.

Ход роботи:

Оскільки у програмах 3.1 та 4.1 було використано стандартну бібліотеку C, а самі програми розбито на функції з C-style call, то основним використанням макросів стала перевірка результатів роботи scanf(вона повертає кількість успішно зчитаних параметрів). Макрос приймає число параметрів, яке повинне бути зчитане, і в разі невідповідності виводить повідомлення про помилку і завершує роботу програми.

Текст програми 4.1

```
%define SIZE 20
%macro
CHECK_INPUT_VALID_INTEGER 1
cmp eax,%1
call invalid_integer_input
%endmacro

extern printf,scanf
global print_array, read_array,
invalid_integer_input, array, array_size,
read_2d_array, array_2d_size

section .data
in_mes_size db "Enter array size in range 1
to 20", 10,0
size_error_mes db "Invalid array size",10,0
parse_error db "Invalid character found",
10, 0
in_mes_digit db "Enter next digit:",10,0
in_format db "%d",0
new_line db 10,0
out_format db "%d ",0
in_mes_2d db "Enter two array
sizes(rows,cols) in format 0 0",10,0
in_2d_format db "%d %d",0
section .bss
array resd SIZE
array_size resd 1
temp_var resd 1
array_2d_size resd 2
section .text

read_array:
push in_mes_size
call printf
add esp,4
push array_size
push in_format
call scanf
add esp,8
CHECK_INPUT_VALID_INTEGER 1

parse_array_no_inp_mes:
mov eax, [array_size]
cmp eax, 0
jle invalid_array_size
cmp eax, SIZE
jg invalid_array_size
mov esi, [array_size]
mov edi, array
push in_mes_digit

parse_next_int:
call printf
push temp_var
push in_format
call scanf
add esp,8
mov eax, [temp_var]
mov [edi], eax
add edi, 4
dec esi
jnz parse_next_int

finalize:
add esp,4
ret

print_array:
mov esi, [array_size]
mov edi, array
sub esp,4
push out_format
write_next_int:
mov eax, [edi]
```

```

mov [esp+4], eax
call printf
add edi,4
dec esi
jnz write_next_int
add esp,8
push new_line
call printf
add esp,4

ret

```

```

invalid_array_size:
push size_error_mes
call printf
add esp,4
mov eax,1
int 80h

```

```

invalid_integer_input:
push parse_error

```

Текст програми 3.1:

```

#define error jne log_error
section .data
    out_mes db "result = %.3f",10,0
    in_mes db "Enter integer to calculate
function:",10,0
    error_mes db "Input parsing error",10,0
    in_format db "%d",0
section .bss
    in_val resd 1
    temp_val resd 1
    out_val resd 2; res 4*2=8 byte for float
output
section .text
extern printf, scanf
global main
main:
    push in_mes
    call printf
    add esp, 4
    push in_val
    push in_format
    call scanf
    add esp,8; in_val now contains integer
    cmp eax, 1
    error
    mov eax, [in_val]
    cmp eax, 1
    jle first_case
    cmp eax, 20

```

```

call printf
add esp,4
mov eax,1
int 80h

```

```

read_2d_array:
push in_mes_2d
call printf
add esp,4
push array_2d_size + 4
push array_2d_size
push in_2d_format
call scanf
add esp,12
CHECK_INPUT_VALID_INTEGER 2
mov eax, [array_2d_size + 4]
mov ebx, [array_2d_size]

```

```

imul ebx
mov [array_size], eax
jmp parse_array_no_inp_mes

```

```

jle second_case
jmp third_case

```

```

first_case:
call calc_le_one
jmp fin
second_case:
call calc_le_twenty
jmp fin
third_case:
call calc_gt_twenty
jmp fin

```

```

fin:
push dword [out_val + 4]
push dword [out_val]
push out_mes
call printf
add esp,12
mov eax, 0x1
int 80h

```

```

calc_le_one:
fild dword [in_val]
fild dword [in_val]
fmulp

```

```

mov dword [temp_val], 75
fild dword [temp_val]

```

```

fmulp

fild dword [in_val]
mov dword [temp_val], 17
fild dword [temp_val]
fmulp

fsubp

fstp qword [out_val]
ret

calc_le_twenty:
fild dword [in_val]
fild dword [in_val]
fmulp

fild dword [in_val]
fld1
faddp

fdivp

mov dword [temp_val], 54
fild dword [temp_val]
faddp

fstp qword [out_val]
ret

calc_gt_twenty:
mov dword [temp_val], 85
fild dword [temp_val]
fild dword [in_val]
fmulp

fld1
fild dword [in_val]
faddp

fdivp

fstp qword [out_val]
ret

log_error:
push error_mes
call printf
mov eax, 0x1
mov ebx, 0x1
int 0x80

```

Вміст .lst файлу програми 4.1

```

#define SIZE 20
2          %macro CHECK_INPUT_VALID_INTEGER 1
3              cmp eax,%1
4              jne invalid_integer_input
5          %endmacro
6
7
8          extern printf,scanf
9          global print_array, read_array, invalid_integer_input, array,
array_size, read_2d_array, array_2d_size
10
11          section .data
12 00000000 456E74657220617272-    in_mes_size  db "Enter array size in range 1 to
20", 10,0
13 00000009 61792073697A652069-
14 00000012 6E2072616E67652031-
15 0000001B 20746F2032300A00
16 00000023 496E76616C69642061-    size_error_mes db "Invalid array size",10,0
17 0000002C 727261792073697A65-
18 00000035 0A00
19 00000037 496E76616C69642063-    parse_error  db "Invalid character found", 10, 0
20 00000040 686172616374657220-
21 00000049 666F756E640A00
22 00000050 456E746572206E6578-    in_mes_digit db "Enter next digit:",10,0
23 00000059 742064696769743A0A-
24 00000062 00
25 00000063 256400          in_format  db "%d",0

```

```

26 00000066 0A00          new_line    db 10,0
27 00000068 25642000      out_format  db "%d ",0
28 0000006C 456E7465722074776F- in_mes_2d  db "Enter two array
sizes(rows,cols) in format 0 0",10,0
29 00000075 206172726179207369-
30 0000007E 7A657328726F77732C-
31 00000087 636F6C732920696E20-
32 00000090 666F726D6174203020-
33 00000099 300A00
34 0000009C 256420256400      in_2d_format  db "%d %d",0
35
36 00000000 <res 00000050>      array      resd SIZE
37 00000050 <res 00000004>      array_size  resd 1
38 00000054 <res 00000004>      temp_var    resd 1
39 00000058 <res 00000008>      array_2d_size resd 2
40
41          section .text
42
43          read_array:
44 00000000 68[00000000]      push in_mes_size
45 00000005 E8(00000000)      call printf
46 0000000A 83C404          add esp,4
47 0000000D 68[50000000]      push array_size
48 00000012 68[63000000]      push in_format
49 00000017 E8(00000000)      call scanf
50 0000001C 83C408          add esp,8
51          CHECK_INPUT_VALID_INTEGER 1
52 0000001F 83F801          <1> cmp eax,%1
53 00000022 0F8590000000      <1> jne invalid_integer_input
54
55          parse_array_no_inp_mes:
56 00000028 A1[50000000]      mov eax, [array_size]
57 0000002D 83F800          cmp eax, 0
58 00000030 7E72          jle invalid_array_size
59 00000032 83F814          cmp eax, SIZE
60 00000035 7F6D          jg invalid_array_size
61 00000037 8B35[50000000]      mov esi, [array_size]
62 0000003D BF[00000000]      mov edi, array
63 00000042 68[50000000]      push in_mes_digit
64
65          parse_next_int:
66 00000047 E8(00000000)      call printf
67 0000004C 68[54000000]      push temp_var
68 00000051 68[63000000]      push in_format
69 00000056 E8(00000000)      call scanf
70 0000005B 83C408          add esp,8
71 0000005E A1[54000000]      mov eax, [temp_var]
72 00000063 8907          mov [edi], eax
73 00000065 83C704          add edi, 4
74 00000068 4E          dec esi
75 00000069 75DC          jnz parse_next_int
76
77          finalize:
78 0000006B 83C404          add esp,4
79 0000006E C3          ret

```

```

79
80             print_array:
81 0000006F 8B35[50000000]      mov esi, [array_size]
82 00000075 BF[00000000]      mov edi, array
83 0000007A 83EC04          sub esp,4
84 0000007D 68[68000000]      push out_format
85             write_next_int:
86 00000082 8B07          mov eax, [edi]
87 00000084 89442404      mov [esp+4], eax
88 00000088 E8(00000000)      call printf
89 0000008D 83C704          add edi,4
90 00000090 4E          dec esi
91 00000091 75EF          jnz write_next_int
92 00000093 83C408          add esp,8
93 00000096 68[66000000]      push new_line
94 0000009B E8(00000000)      call printf
95 000000A0 83C404          add esp,4
96
97 000000A3 C3          ret
98
99
100            invalid_array_size:
101 000000A4 68[23000000]      push size_error_mes
102 000000A9 E8(00000000)      call printf
103 000000AE 83C404          add esp,4
104 000000B1 B801000000      mov eax,1
105 000000B6 CD80          int 80h
106
107            invalid_integer_input:
108 000000B8 68[37000000]      push parse_error
109 000000BD E8(00000000)      call printf
110 000000C2 83C404          add esp,4
111 000000C5 B801000000      mov eax,1
112 000000CA CD80          int 80h
113
114            read_2d_array:
115 000000CC 68[6C000000]      push in_mes_2d
116 000000D1 E8(00000000)      call printf
117 000000D6 83C404          add esp,4
118 000000D9 68[5C000000]      push array_2d_size + 4
119 000000DE 68[58000000]      push array_2d_size
120 000000E3 68[9C000000]      push in_2d_format
121 000000E8 E8(00000000)      call scanf
122 000000ED 83C40C          add esp,12
123            CHECK_INPUT_VALID_INTEGER 2
124 000000F0 83F802      <1> cmp eax,%1
125 000000F3 75C3      <1> jne invalid_integer_input
126 000000F5 A1[5C000000]      mov eax, [array_2d_size + 4]
127 000000FA 8B1D[58000000]      mov ebx, [array_2d_size]
128
129 00000100 F7EB          imul ebx
130 00000102 A3[50000000]      mov [array_size], eax
131 00000107 E91CFFFFFF      jmp parse_array_no_inp_mes

```

Вміст .lst файлу програми 3.1

```
1          %define error jne log_error
2          ; for some reason commands without args is implemented like
fcommand TO st1 and then pushes st1 down
3          section .data
4 00000000 726573756C74203D20-      out_mes  db "result = %.3f",10,0
5 00000009 252E33660A00
6 0000000F 456E74657220696E74-      in_mes   db "Enter integer to calculate
function:",10,0
7 00000018 6567657220746F2063-
8 00000021 616C63756C61746520-
9 0000002A 66756E6374696F6E3A-
10 00000033 0A00
11 00000035 496E70757420706172-      error_mes db "Input parsing error",10,0
12 0000003E 73696E67206572726F-
13 00000047 720A00
14 0000004A 256400          in_format db "%d",0
15          section .bss
16 00000000 <res 00000004>      in_val   resd 1
17 00000004 <res 00000004>      temp_val resd 1
18 00000008 <res 00000008>      out_val  resd 2; res 4*2=8 byte for float output
19          section .text
20          extern printf, scanf
21          global main
22          main:
23 00000000 68[0F000000]      push in_mes
24 00000005 E8(00000000)      call printf
25 0000000A 83C404      add esp, 4
26 0000000D 68[00000000]      push in_val
27 00000012 68[4A000000]      push in_format
28 00000017 E8(00000000)      call scanf
29 0000001C 83C408      add esp,8; in_val now contains integer
30 0000001F 83F801      cmp eax, 1
31 00000022 0F85E5000000      error
32 00000028 A1[00000000]      mov eax, [in_val]
33 0000002D 83F801      cmp eax, 1
34 00000030 7E07      jle first_case
35 00000032 83F814      cmp eax, 20
36 00000035 7E09      jle second_case
37 00000037 EB0E      jmp third_case
38
39          first_case:
40 00000039 E830000000      call calc_le_one
41 0000003E EB0E      jmp fin
42          second_case:
43 00000040 E86A000000      call calc_le_twenty
44 00000045 EB07      jmp fin
45          third_case:
46 00000047 E896000000      call calc_gt_twenty
47 0000004C EB00      jmp fin
48
49          fin:
50 0000004E FF35[0C000000]      push dword [out_val + 4]
51 00000054 FF35[08000000]      push dword [out_val]
```

```

52 0000005A 68[00000000]      push out_mes
53 0000005F E8(00000000)      call printf
54 00000064 83C40C          add esp,12
55 00000067 B801000000      mov eax, 0x1
56 0000006C CD80          int 80h
57
58                      calc_le_one:
59 0000006E DB05[00000000]      fild dword [in_val]
60 00000074 DB05[00000000]      fild dword [in_val]
61 0000007A DEC9          fmulp;x^2
62
63 0000007C C705[04000000]4B00-   mov dword [temp_val], 75
64 00000084 0000
65 00000086 DB05[04000000]      fild dword [temp_val]
66 0000008C DEC9          fmulp ;75x^2
67
68 0000008E DB05[00000000]      fild dword [in_val]
69 00000094 C705[04000000]1100-   mov dword [temp_val], 17
70 0000009C 0000
71 0000009E DB05[04000000]      fild dword [temp_val]
72 000000A4 DEC9          fmulp;17x
73
74 000000A6 DEE9          fsubp; 54x^2 - 17x
75
76 000000A8 DD1D[08000000]      fstp qword [out_val]
77 000000AE C3          ret
78
79                      calc_le_twenty:
80 000000AF DB05[00000000]      fild dword [in_val]
81 000000B5 DB05[00000000]      fild dword [in_val]
82 000000BB DEC9          fmulp; x^2
83
84 000000BD DB05[00000000]      fild dword [in_val]
85 000000C3 D9E8          fld1
86 000000C5 DEC1          faddp; x+1
87
88 000000C7 DEF9          fdivp; x^2/(x+1)
89
90 000000C9 C705[04000000]3600-   mov dword [temp_val], 54
91 000000D1 0000
92 000000D3 DB05[04000000]      fild dword [temp_val]
93 000000D9 DEC1          faddp; 54+(x^2/(x+1))
94
95 000000DB DD1D[08000000]      fstp qword [out_val]
96 000000E1 C3          ret
97
98                      calc_gt_twenty:
99 000000E2 C705[04000000]5500-   mov dword [temp_val], 85
100 000000EA 0000
101 000000EC DB05[04000000]      fild dword [temp_val]
102 000000F2 DB05[00000000]      fild dword [in_val]
103 000000F8 DEC9          fmulp; 85x
104
105 000000FA D9E8          fld1

```



```

106 000000FC DB05[00000000]      fild dword [in_val]
107 00000102 DEC1                faddp; x+1
108
109 00000104 DEF9                fdivp; 85x/(1+x)
110
111 00000106 DD1D[08000000]      fstp qword [out_val]
112 0000010C C3                  ret
113
114                                log_error:
115 0000010D 68[35000000]         push error_mes
116 00000112 E8(00000000)        call printf
117 00000117 B801000000          mov eax, 0x1
118 0000011C BB01000000          mov ebx, 0x1
119 00000121 CD80                int 0x80

```

Висновок:

Були переписані програми 2.1, 3.1 та 4.1 з використанням макросів.