

Міністерство освіти і науки України
Національний авіаційний університет
Навчально-науковий інститут комп'ютерних інформаційних технологій
Кафедра комп'ютеризованих систем управління

Лабораторна робота №1
з дисципліни «Системне програмування»
на тему «Введення та виведення тексту»
Варіант №4

Виконав:
студент ННІКІТ СП-225
Клокун Владислав
Перевірив:
Артамонов Є. Б.

Київ 2018

1 Мета роботи

Ознайомитись з синтаксисом та структурою програми на мові асемблера. Навчитися використовувати функції операційної системи для введення та виведення тексту. Дослідити різницю між файлами .COM та .EXE.

2 Хід роботи

Основним завданням роботи є розробка програми, до задач якої входять:

1. Введення рядка з клавіатури.
2. Вивід тексту на екран.

В результаті виконання роботи була розроблена програма (ліст. 2.1).

Лістинг 2.1: Основна програма

```
1      global      _main
2      extern      _scanf
3      extern      _printf
4
5      section .text
6  _main:
7      push      rbx ; align the stack
8
9      lea      rdi, [rel fmtoutstr] ; load format string
10     lea      rsi, [rel inputprompt] ; load input prompt text
11     call     _printf
12
13     lea      rdi, [rel fmtinstr] ; load input format string
14     lea      rsi, [rel inputstr] ; load pointer to stored data
15     call     _scanf
16
17     lea      rdi, [rel fmtoutstr] ; load format string
18     lea      rsi, [rel inputstr] ; load the string we read
19     call     _printf
20
21     pop      rbx ; clean up the stack
22     ret
23
24     section .data
25     fmtoutstr:      db "%s", 10, 0 ; equivalent to "%s\n"
26     inputprompt:    db "Enter a string:", 0
27     fmtinstr:       db "%[^", 10, "]"s", 0 ; equiv to "%[^\\n]s"
28
```

```

29      section .bss
30  inputstr:      resb      256

```

Додатковим завданням роботи було написання програми, яка виведе з введеного рядку на екран тільки парні за порядком літери. В результаті виконання роботи була розроблена відповідна програма (ліст. 2.2).

Лістинг 2.2: Додаткова програма

```

1      .section      __TEXT,__text,regular,pure_instructions
2      .macosx_version_min 10, 11
3      .intel_syntax noprefix
4      .globl      _main
5      .align      4, 0x90
6  _main:                                ## @main
7  ## BB#0:
8      push      rbp
9      mov      rbp, rsp
10     sub      rsp, 320
11     lea      rdi, [rip + L_.str]
12     mov      rax, qword ptr [rip + ___stack_chk_guard@GOTPCREL]
13     mov      rax, qword ptr [rax]
14     mov      qword ptr [rbp - 8], rax
15     mov      dword ptr [rbp - 276], 0
16     mov      qword ptr [rbp - 288], 0
17     mov      al, 0
18     call     _printf
19     lea      rdi, [rip + L_.str.1]
20     lea      rsi, [rbp - 272]
21     mov      dword ptr [rbp - 300], eax ## 4-byte Spill
22     mov      al, 0
23     call     _scanf
24     lea      rdi, [rip + L_.str.2]
25     lea      rsi, [rbp - 272]
26     mov      dword ptr [rbp - 304], eax ## 4-byte Spill
27     mov      al, 0
28     call     _printf
29     lea      rdi, [rbp - 272]
30     mov      dword ptr [rbp - 308], eax ## 4-byte Spill
31     call     _strlen
32     mov      qword ptr [rbp - 288], rax
33     mov      qword ptr [rbp - 296], 0
34  LBB0_1:
35     mov      rax, qword ptr [rbp - 296]
36     cmp      rax, qword ptr [rbp - 288]

```

```

37     jae     LBB0_6
38     ## BB#2:
39     mov     rax, qword ptr [rbp - 296]
40     add     rax, 1
41     and     rax, 1
42     cmp     rax, 0
43     jne     LBB0_4
44     ## BB#3:
45     lea     rdi, [rip + L_.str.3]
46     mov     rax, qword ptr [rbp - 296]
47     movsx   esi, byte ptr [rbp + rax - 272]
48     mov     al, 0
49     call    _printf
50     mov     dword ptr [rbp - 312], eax ## 4-byte Spill
51 LBB0_4:
52     jmp     LBB0_5
53 LBB0_5:
54     mov     rax, qword ptr [rbp - 296]
55     add     rax, 1
56     mov     qword ptr [rbp - 296], rax
57     jmp     LBB0_1
58 LBB0_6:
59     mov     rax, qword ptr [rip + ___stack_chk_guard@GOTPCREL]
60     mov     rax, qword ptr [rax]
61     cmp     rax, qword ptr [rbp - 8]
62     jne     LBB0_8
63     ## BB#7:
64     xor     eax, eax
65     add     rsp, 320
66     pop     rbp
67     ret
68 LBB0_8:
69     call    ___stack_chk_fail
70
71     .section  __TEXT,__cstring,cstring_literals
72 L_.str:
73     .asciz   "Enter a string:\n" ## @.str
74
75 L_.str.1:
76     .asciz   "%[^\\n]s" ## @.str.1
77
78 L_.str.2:
79     .asciz   "You entered: %s\\n" ## @.str.2
80
81 L_.str.3:
82     ## @.str.3

```

```
82     .asciz    "%c\n"  
83  
84  
85 .subsections_via_symbols
```

3 Висновок

Під час виконання даної лабораторної роботи ми ознайомились з синтаксисом та структурою програми на мові асемблера; навчились використовувати функції операційної системи для введення та виведення тексту, а також дослідили різницю між файлами .COM та .EXE.