

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
за IV семестр
по дисциплине: КСИС
Тема: Обработка строк.

Выполнил:
студент 2 курса
Корнаевич И. Д.

Проверил:
Савицкий Ю. В.

Брест 2021

Задание: Написать программу обработки текста на ассемблере. Если текст начинается и заканчивается на разные цифры, то перевернуть его, иначе заменить все буквы их кодами mod 10.

```
1 section .data
2 text db "1Some text2", '\0'
3
4 section .text
5 global _start
6
7 _start:
8     push text
9     call strlen
10    add rsp, 8
11
12    push rax
13    push text
14    call checkCondition
15
16    cmp rax, 1
17    je .first
18 .second:
19    call reverseText
20    jmp .done
21 .first:
22    call replaceByCode
23 .done:
24    sub rsp, 16
25
26    mov rax, 0x3c
27    mov rdi, 0x0
28    syscall
29
30 ;long checkCondition (char* string, long len)
31 ;1 - first condition
32 ;2 - second
33 checkCondition:
34     push rbp
35     mov rbp, rsp
36     mov rsi, [rbp + 16]
37     mov rax, [rbp + 24]
38
39     mov bl, [rsi] ;first symbol in text.
40     mov dl, [rsi + rax]
```

```
41
42     mov cl, dl
43     call .isNum
44     cmp rax, 0
45     je .second
46
47     mov cl, bl
48     call .isNum
49     cmp rax, 0
50     je .second
51
52     cmp dl, bl
53     jne .second
54
55 .first:
56     mov rax, 1
57     jmp .done
58
59 .second:
60     mov rax, 2
61
62 .done:
63     mov rsp, rbp
64     pop rbp
65     ret
66
67 ;.isNum(char cl)
68 .isNum:
69     mov rax, 1
70     cmp cl, '0'
71     jl .notNum
72     cmp cl, '9'
73     jg .notNum
74     ret
75 .notNum:
76     mov rax, 0
77     ret
78
79 ;reverseText(char* text)
80 reverseText:
81     push rbp
82     mov rbp, rsp
83     xor rax, rax
```

```

84
85     mov rsi, [rbp + 16]
86 .nextChar:
87     cmp [rsi], byte '\0'
88     je .reverce
89     lodsb
90     push rax
91     jmp .nextChar
92 .reverce:
93     mov rsi, [rbp + 16]
94 .nextRevercedChar:
95     cmp [rsi], byte '\0'
96     je .done
97     pop qword [rsi]
98     inc rsi
99     jmp .nextRevercedChar
100
101 .done:
102     mov rsp, rbp
103     pop rbp
104     ret
105
106
107 ;replaceByCode(char* text)
108 replaceByCode:
109     push rbp
110     mov rbp, rsp
111     mov rsi, [rbp + 16]
112 .loop:
113     lodsb
114     cmp al, '\0'
115     je .done
116
117     mov dl, 10
118     div dl
119     mov [rsi], dl
120     jmp .loop
121 .done:
122     mov rsp, rbp
123     pop rbp
124     ret
125
126 ;-----

```

```

127 ; long strlen(char* buffer)
128 strlen:
129     push rbp
130     mov rbp, rsp
131     xor rax, rax          ; rax <- 0
132     movzx esi, byte [rbp + 15] ; esi <- first byte
        before buffer
133 .loop:                    ; finds index of '\0' in buffer
134     inc esi                ; goto next symbol
135     inc rax
136     test esi, esi
137     je .loop              ; while esi != 0
138
139     pop rbp
140     ret

```

Вывод: Разработана программа обработки строк на ассемблере. Изучены команды **lodsb**, **stosb**, **movsb**.