Университет ИТМО

Кафедра ВТ

**Языки системного программирования**

Лабораторная работа №2

Группа P3210

Нгу Фыонг Ань

ПРОВЕРИЛ:

2019 год

**colon.inc**

%define pointer 0

%macro colon 2

%2:

dq pointer

db %1, 0

%define pointer %2

%endmacro

**dict.asm**

global find\_word

extern string\_equals

section .text

; rdi is the address of the word

; in rsi the address of the last word

; returns result in rax (address / 0)

find\_word:

.loop:

test rsi, rsi

jz .end

push rsi

add rsi, 8

call string\_equals

pop rsi

test rax, rax

jnz .found

mov rsi, [rsi]

jmp .loop

.found:

mov rax, rsi

.end:

ret

**lib.asm**

section .text

global string\_length

global print\_newline

global print\_string

global read\_word

global string\_equals

; takes a pointer to a string, returns its length. rcx – counter

string\_length:

xor rcx, rcx

mov rax, rdi

.loop:

mov rax, [rdi+rcx]

inc rcx

test al, 0xFF

jnz .loop

dec rcx

mov rax, rcx

ret

; transition to a new line

; 0x0A - „\ n“

print\_newline:

push rdi

mov rdi, 0x0A

call print\_char

pop rdi

ret

; the output of the string, the first argument is a pointer to it

; set rsi, rdx, rdi and rax parameters, before calling sys\_write

; rsi - start address, rdx - number of characters

; rdi - explosive flow, rax - system command number

print\_string:

mov rsi, rdi

call string\_length

mov rdx, rax

mov rdi, 1

mov rax, 1

syscall

ret

section .data

word\_buffer times 256 db 0

section .text

; read the following word from the input stream, skipping before the whitespace characters

; returns: in rax - word, in rdx - word length

read\_word:

xor r8, r8

mov r8, word\_buffer

.skip:

call read\_char

test rax, rax

jz .finish

cmp rax, 32

jle .skip

.read\_next:

mov [r8], rax

inc r8

call read\_char

cmp rax, 32

jg .read\_next

.finish:

mov rdi, word\_buffer

call string\_length

mov rdx, rax

mov rax, word\_buffer

ret

; takes two pointers to strings (rax, rdx), compares them character-by-character.

; if they are equal, then in rax returns 1, otherwise 0

string\_equals:

xor rcx, rcx

.loop:

mov al, byte[rdi+rcx]

mov dl, byte[rsi+rcx]

cmp al, dl

jne .not\_equal

inc rcx

cmp al, 0

je .equal

jmp .loop

.not\_equal:

mov rax, 0

ret

.equal:

mov rax, 1

ret

**main.asm**

extern read\_word

extern print\_newline

extern print\_string

extern string\_length

extern find\_word

global \_start

section .data

not\_found\_exception: db "Sorry, we can't find your word", 0

section .text

%include "colon.inc"

%include "words.inc"

\_start:

call read\_word

mov rsi, pointer

call find\_word

test rax, rax

jz .not\_found

add rax, 8

mov rdi, rax

call string\_length

add rdi, rax

inc rdi

call print\_string

jmp .fin

.not\_found:

mov rdi, not\_found\_exception

call string\_length

mov rsi, rdi

mov rdx, rax

mov rax, 1

mov rdi, 2

syscall

.fin:

mov rax, 60

mov rdi, 0

syscall

**words.inc**

colon "first", first

db "first word explanation", 0

colon "second", second

db "second word explanation", 0

colon "third", third

db "third word explanation", 0

2. Result:

