УНИВЕРСИТЕТ ИТМО

ПИиКТ

Языки Системного Программирования

ЛАБОРАТОРНАЯ РАБОТА №1

Группа: P3200

Студент: Базарова Анна Александровна

2018 год

Санкт-Петербург

Цель: дать определение функциям в файле lib.inc

Выполнение:

section .text

string\_length:

;rdi - указатель на строку

xor rax, rax

;rax - смещение

.loop: ;если в ячейке по адресу rdi+rax - 0x00 то

;выходим из цикла

cmp byte[rdi+rax], 0

je .good

inc rax

jmp .loop

.good:

ret

print\_string:

;rdi - указатель на нуль-терминированную строку

;(начало)

xor rax, rax

;получим длинну строки из пред. функции

call string\_length

;длина строки лежит в rax

mov rsi, rdi ;откуда брать данные

mov rdi, 1 ;stdout

mov rdx, rax ;сколько байт строка

mov rax, 1 ;номер write

syscall

ret

print\_char:

;rdi - символ, который надо вывести в stdout

;нужно вывести не его код, а сам символ

xor rax, rax

push rdi

;rsp указывает на ячейку с этим символом

mov rsi, rsp ;откуда брать данные

mov rdi, 1 ;stdout

mov rdx, 1 ;сколько байт строка

mov rax, 1 ;номер write

syscall

pop rdi

ret

print\_newline:

;положим в rdi 0x0A и print\_char нам в помощь

mov rdi, 0x0A

xor rax, rax

ret

print\_uint:

;rdi - 8-байтовое число в 10 с сч

xor rax, rax

mov r8, rsp ;запоминаем стек чтобы не попортить,

;и с него же будем выводить из стека

dec rsp ;push вручную

mov byte[rsp], 0;символ конца строки

mov r9, 10 ;делитель

mov rax, rdi ;делимое

.loop

xor rdx, rdx

div r9 ;ругается если просто число, надо регистр

add rdx, 0x30 ;добавив код нуля в ascii получим код цифры

dec rsp

mov [rsp], dl ;младшие 8 бит

test rax, rax

jnz .loop

mov rdi, rsp ;начало числа

call print\_string

mov rsp, r8 ;возврат указателя стека

ret

print\_int:

;rdi - 8-байтовое знаковое число в 10 с сч

xor rax, rax

test rdi, rdi

jns .good

neg rdi

push rdi

mov rdi,"-"

call print\_char

pop rdi

jmp .good

.good

call print\_uint

ret

string\_equals:

xor rcx, rcx ;счетчик цикла

mov r10, rdi

mov r9, rsi

.loop:

mov r11b, byte [r10+rcx] ;r11 = буква первой строки

mov r8b, byte [r9+rcx] ;r8 = буква второй строки

cmp r8b, r11b

jne .goodbye ;не равны - до свидания

inc rcx

cmp r11b, 0x0

je .good

jmp .loop

.good:

mov rax, 1 ;1 = строки равны

ret

.goodbye:

mov rax, 0 ;0 = строки не равны

ret

read\_char:

xor rax, rax

push 0

mov rax, 0 ;read

mov rdi, 0 ;stdin

mov rsi, rsp ;куда

mov rdx, 1 ;сколько

syscall

pop rax

ret

read\_word:

xor r8, r8

mov r10, rsi

mov r9, rdi

.loop:

call read\_char

cmp rax, 0x20 ;пробел?

je .loop

cmp rax, 0x09 ;таб?

je .loop

cmp rax, 0x0A ;перенос?

je .loop

jmp .good ;все хорошо, это буква

.ask:

cmp al, 0x20 ;пробел?

je .return

cmp rax, 0x09 ;таб?

je .return

cmp rax, 0x0A ;перенос?

je .return

.good:

cmp al, 0x09 ;таб

je .return

cmp al, 0x00 ;конец строки

je .return

mov byte [r9+r8], al

inc r8

call read\_char

cmp r8, r10 ;если строка стала длиннее буфера

je .outofrange

jmp .ask

.return:

mov rax, r9

mov byte [r9+r8], 0

mov rdx, r8 ;длина строки

ret

.outofrange:

mov rax, 0

xor rdx, rdx ; string length = 0

ret

; rdi points to a string

; returns rax: number, rdx : length

parse\_uint:

xor rax, rax

xor rcx, rcx

xor r8, r8

mov r9, 10

.loop:

cmp byte[rdi+r8], "0"

jb .good

cmp byte[rdi+r8], "9"

ja .good ;это не число, идем в конец

mul r9 ;цифра, методом

;x=a\*10^3+b\*10^2+c\*10+d

;запоминаем число в rax

mov cl, byte[rdi+r8]

sub cl, "0"

add rax, rcx

inc r8

jnz .loop

.good:

mov rdx,r8

ret

; rdi points to a string

; returns rax: number, rdx : length

parse\_int:

xor rax, rax

cmp byte [rdi], 0x2d

je .negative

call parse\_uint

ret

.negative:

inc rdi

call parse\_uint

cmp rdx, 0 ;если мы не считали ни одной цифры,

;то минус не нужен

je .return

neg rax

inc rdx

.return:

ret

string\_copy:

xor rcx, rcx ;счетчик цикла

.loop:

cmp rcx, rdx ;rdx - размер буфера

je .oversized

mov rax, [rdi+rcx] ;rdi - указатель на строку

mov [rsi+rcx], rax ;rsi - указатель на буфер

cmp byte [rdi+rcx], 0x0

je .return

inc rcx

jmp .loop

.return:

mov rax, rsi

ret

.oversized:

mov rax, 0

ret