Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 7

по дисциплине ПнаЯА

Вариант 1

Выполнил студент гр. 250502: Бекетова М.А.

Проверил: Туровец Н.О.

Минск 2023

1 ТЕОРИТИЧЕСКИЕ СВЕДИНИЯ

Для выполнения работы требуется рассмотреть следующие элементы языка ассемблера и операционной системы:

1. Управление памятью

-- Функция DOS 4Ah (INT 21h) – изменить размер блока памяти

-- Функция DOS 48h (INT 21h) – выделить блок памяти

-- Функция DOS 49h (INT 21h) – освободить блок памяти

2. Загрузка и выполнение программ.

Для загрузки и выполнения программ требуется использовать функцию DOS 4Bh (INT 21h) – загрузить и выполнить программу

1. Оверлейные модули.

Оверлей – это часть исполняемой программы (обычно процедура, хотя

это может быть полностью самостоятельная программа со своими сегментами данных и стека), которая по мере необходимости загружается в определенную область памяти.

2 КОД

.model tiny

.code

org 80h

cmd\_length db ?

cmd\_line db ?

org 100h

start:

cld ; clear directon flag

mov bp,sp

mov cl,cmd\_length

cmp cl,1 ; length<=1

jle exit

mov cx,-1

mov di,offset cmd\_line

find\_param:

mov al,' '

rep scasb

dec di

push di

mov si,di

scan\_params:

lodsb ; mov al, si;inc si

cmp al,0Dh ; caret

je params\_ended

cmp al,20h

jne scan\_params ;while != ' '

dec si

mov byte ptr [si],0

mov di,si

inc di

jmp short next\_param

params\_ended:

dec si

mov byte ptr [si],0

next\_param:

mov al,' '

repe scasb ;al, di

dec di

mov si, di

mov di, offset number

scan\_param:

cmp [si],0Dh ; caret

je param\_ended

cmp [si],20h ; space

je param\_ended

movsb ; mov di, si; inc si; inc di

jmp scan\_param

param\_ended:

mov byte ptr [si],0

mov si, offset number

string\_to\_num:

xor dx,dx

loop\_:

xor ax,ax

lodsb ; mov al,si; inc si

test al,al

jz ex

;cmp al,'9'

;jnbe loop\_

;cmp al,'0'

;jb loop\_

sub ax,'0'

push ax

mov ax, dx

mov dx, 10

mul dx

mov dx, ax

pop ax

add dx, ax

jmp loop\_

ex:

mov ax,dx

mov num, ax

mov sp, program\_length+100H+200H

mov ah, 4Ah ;set stack size

stack\_offset = program\_length+ 100h + 200h

mov bx, stack\_offset shr 4 + 1

int 21h

mov cx, num

cycle:

mov ax,4B00h

mov dx, offset cmd\_line+1

mov bx, 0

int 21h

jnc next

mov ah,9

lea dx, error

int 21h

next:

loop cycle

exit:

int 20h

error db "error",10,13,'$'

commandline db 125

db " /?"

command\_text db 122 dup(?)

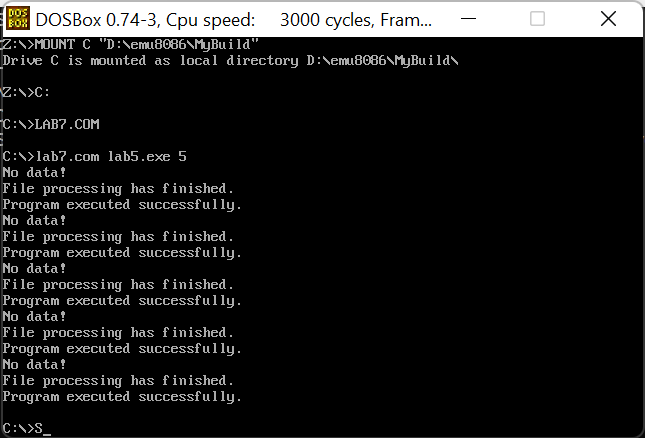
number db 80 dup(0)

num dw 0

program\_length equ $-start

end start

3 ДЕМОНСТРАЦИЯ РАБОТЫ



4 ВЫВОДЫ

Таким образом была реализована программа, запускающая саму себя N раз (N – число в диапазоне [1, 255]). При запуске и окончании программы выдается номер текущей копии. В результате удалось ознакомиться с загрузкой и выполнением программ, рассмотреть работу с памятью.