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. 150502: Снитко Д.А.

Проверил: Туровец Н.О.

Минск 2022

Теоретические сведения:

1. Управление памятью.

2. Особенности загрузки и выполнения программ.

3. Создание оверлейных модулей.

4. Особенности загрузки и вызова оверлейных процедур.

Задание:

Заменить в строке слова, содержащие заданный набор букв, на другое заданное слово.

Код:

.model tiny

.stack 100h

.data

invalidArgumentsMessage db "Invalid arguments! 1st arg - number of child program executions, next - child program arguments.$"

loadErrorMessage db "Program load error!$"

path db "source.com",0

argumentsSize equ 127

n dw 0

ten dw 000Ah

nBuffSize equ 4

nBuff db nBuffSize dup(?)

arguments db argumentsSize dup(?)

epb dw 0

cmd\_off dw ?

cmd\_seg dw ?

fcb1 dd ?

fcb2 dd ?

.code

jmp start

outputString proc

mov ah, 9

int 21h

ret

outputString endp

readN proc

mov di, 81h

lea si, nBuff

xor ax, ax

xor bx, bx

skipSpaces:

cmp es:[di], 0Dh

jz readError

cmp es:[di], ' '

jnz getN

inc di

jmp skipSpaces

getN:

mov cx, nBuffSize

dec cx

getNCycle:

mov al, es:[di]

cmp al, ' '

jz parseN

cmp al, 0Dh

jz parseN

cmp al, '0'

jl readError

cmp al, '9'

jg readError

mov [si], al

inc di

inc si

loop getNCycle

cmp es:[di], ' '

jz parseN

cmp es:[di], 0Dh

jnz readError

parseN:

lea si, nBuff

cmp [si], 0

jz readError

xor ax, ax

xor dx, dx

xor bx, bx

parseNCycle:

mul ten

mov bl, [si]

sub bl, '0'

add ax, bx

inc si

cmp [si], 0

jne parseNCycle

parseNCycleEnd:

cmp ax, 255

jg readError

cmp ax, 0

jbe readError

mov n, ax

ret

readError:

lea dx, invalidArgumentsMessage

call outputString

jmp errorExit

ret

readN endp

loadError:

lea dx, loadErrorMessage

call outputString

jmp errorExit

start:

mov ax, @data

mov ds, ax

call readN

lea si, arguments

mov ax, di

sub ax, 80h

mov [si], al

saveArguments:

mov al, es:[di]

mov [si], al

cmp al, 0Dh

jz shrinkMemory

inc di

inc si

jmp saveArguments

shrinkMemory:

mov ax, @data

mov es, ax

mov bx, last

shr bx, 4

add bx, 17

mov ah, 4Ah

int 21h

mov ax, bx

shl ax, 4

dec ax

mov sp, ax

mov cx, n

lea bx, arguments

mov cmd\_off, bx

mov cmd\_seg, ds

lea dx, path

lea bx, epb

loadProgramsCycle:

pusha

mov ax, 4B00h

int 21h

popa

jb loadError

loop loadProgramsCycle

mov ax, 4C00h

int 21h

errorExit:

mov ax, 4C01h

int 21h

last: db 0

end start