

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра Информатики
Дисциплина «Конструирование программ»

ОТЧЕТ
к лабораторной работе №6
на тему:
«ЗАГРУЗКА И ВЫПОЛНЕНИЕ ПРОГРАММ. РАБОТА С ПАМЯТЬЮ»
БГУИР 6-05-0612-02 01

Выполнил студент группы 353503
АБДУЛОВ Александр Алексеевич

(дата, подпись студента)

Проверил ассистент каф. Информатики
РОМАНЮК Максим Валерьевич

(дата, подпись преподавателя)

Минск 2024

1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Задание 1. Вариант 1. Написать программу, запускающую другую программу N раз (N – число в диапазоне [1, 255]). Имя запускаемой программы задается константой.

2 ВЫПОЛНЕНИЕ РАБОТЫ

Для написания данной лабораторной работы используется эмулятор ети8086, а также DOSBox для запуска программы. Ниже представлен листинг кода программы.

```
.model small
.stack 100h
.data

    N db 3 dup(0)
    buff db 3 dup(0)
    err db "illegal value of N", 24h
    emp db "Cmd is empty", 0Ah, 0Dh, 24h
    exe db "Hello.exe", 0

    EPB dw 0
    dw offset commandline, 0
    dw 005Ch, 0, 006Ch, 0
    commandline db 125
    db " /?"

    dsize dw $-N

.code

print macro str
    lea dx, str
    mov ah, 09h
    int 21h
endm

main:

    mov ax, @data
    mov ds, ax

    call readCmd
    call CmdToInt ;res dx

    mov cx, dx
    mov dx, 1

    mov sp, csize+100h+200h
    mov ah, 4ah
    mov bx, (csize/16)+100/16+(dsize/16)+25
    int 21h
```

```

        mov ax, cs

        mov word ptr EPB+02h, ax
        mov word ptr EPB+06h, ax
        mov word ptr EPB+0Ah, ax

cycle:

        mov ax, 4B00h
        mov dx, offset exe
        mov bx, EPB
        int 21h
        jc Error

loop cycle

ENDPROG:
        mov ah, 4Ch
        int 21h

Error:
        print err
        jmp ENDPROG

Empty:
        print emp
        jmp ENDPROG

;=====;

proc readCmd near
        xor cx, cx
        xor ax, ax

        mov cl, es:[80h] ; count of symbols in cmd

        mov si, 81h ; adress of arguments
        mov di, offset N

        cmp cx, 0
        je Empty

        inc si
        dec cx

        cmp cx, 3
        jg Error

copy:
        push ds
        mov ax, es
        mov ds, ax

        mov al, [si]

        pop ds

        mov [di], al

        dec cl

```

```

        inc si
        inc di

        cmp cl,0
jne copy

        ret
readCmd endp

;=====;

proc CmdToInt near
    xor dx, dx
    xor cx, cx

    mov cl, es:[80h]
    dec cx

    mov si, offset N
convert:
    xor ax, ax
    mov al, [si]

    cmp al, 30h
    jl Error

    cmp al, 39h
    jg Error

    sub al, 30h

    push dx
    push cx

    dec cx

    cmp cx, 0
    je skip

multiply:
    mov dx, 10
    mul dx

    dec cx
    cmp cx, 0
jne multiply

skip:

    pop cx
    pop dx

    add dx, ax

    inc si
loop convert

    cmp dx,255
    jg Error

```

```

        cmp dx,0
        je Error

        ret
CmdToInt endp

;=====;

        csize dw $-main
end main

```

Выполняется обработка командной строки, преобразование аргументов в число, выделение памяти и многократный запуск внешнего приложения. В сегменте данных хранятся переменные: N для числовых значений, строки сообщений об ошибках err, emp, имя исполняемого файла exe, и зарезервированная область для EPB. Основной алгоритм включает чтение и проверку аргументов командной строки readCmd, их преобразование в число CmdToInt, управление памятью через прерывание 21h и циклический запуск программы "Hello.exe". Встроенный макрос print упрощает вывод сообщений. Ошибки, такие как пустая команда или недопустимое значение, обрабатываются переходом в соответствующие обработчики с завершением программы.

На рисунке 1 представлен результат выполнения программы.

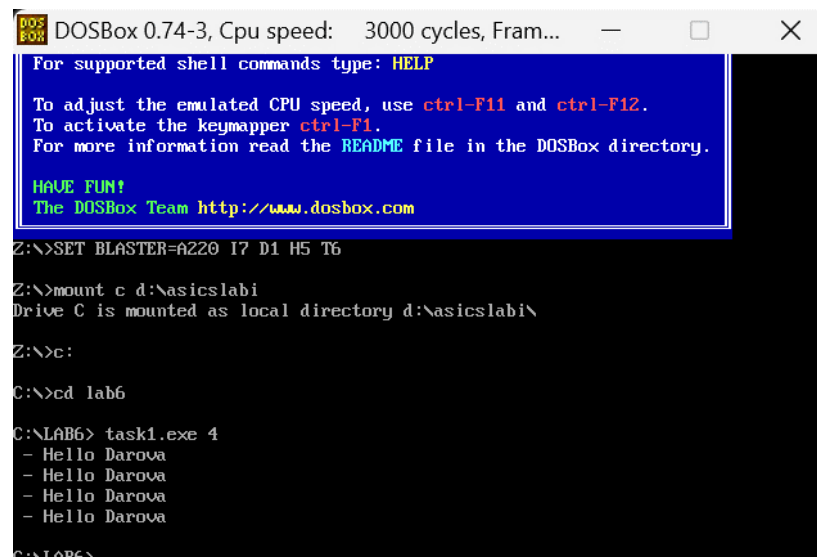


Рисунок 1 – Результат выполнения программы

Ниже представлен листинг кода файла Hello.asm.

```

.model small
.stack 100h
.data
    text db " - Hello Darova", 0Ah, 0Dh, 24h
.code

main:

```

```
mov ax, @data
mov ds, ax

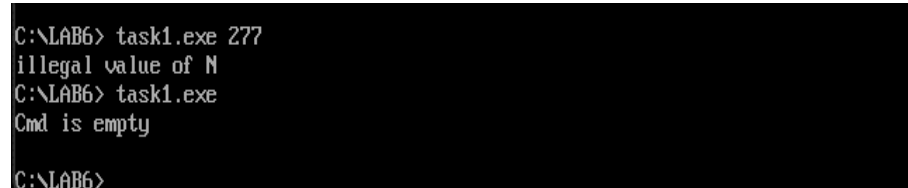
lea dx, buffer
mov ah, 09h
int 21h

lea dx, text
mov ah, 09h
int 21h

mov ah, 4Ch
int 21h

end main
```

На рисунке 3 представлен результат выполнения обработки ошибок.



```
C:\LAB6> task1.exe 277
illegal value of N
C:\LAB6> task1.exe
Cmd is empty
C:\LAB6>
```

Рисунок 3 – Результат выполнения обработки ошибок

ВЫВОД

В ходе выполнения лабораторной работы было достигнуто ознакомление с процессом загрузки и выполнения программ в среде DOS. Рассмотрены методы работы с памятью, включая выделение и управление памятью для различных задач.