

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Организация ЭВМ и системы»
Тема: Написание собственного прерывания

Студента гр. 1383

Панов М.Ю.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить прерывания на языке Ассемблера.

Задание.

Вариант 3А

3- 23h - прерывание, генерируемое при нажатии клавиш Control + C

А - Выполнить вывод сообщения на экран заданное число раз, после чего вставить фиксированную задержку и вывести сообщение о завершении обработки.

Выполнение работы.

Определяется процедура пользовательского прерывания WRITE. С помощью output выводится строка, это повторяется по заданному значению в cx с помощью loop. Метка delay реализует задержку перед выводом завершающей строки на экран с помощью прерывания 1Ah, в таком случае время задержки задается через регистр bx, которое определяется в тиках процессора.

В процедуре MAIN задается количество раз которое повторяется строка и время задержки, так же с помощью функции 35h мы получаем оригинальный вектор прерываний и заменяем прерывание по смещению 23h на необходимое нам. При нажатии Control+C происходит вызов написанного прерывания, после чего восстанавливается оригинальный вектор прерываний.

Выводы.

В ходе работы были изучены методы работы с прерываниями и разработано собственное прерывание.

Исходный код

ASSUME CS:CODE, DS:DATA, SS:AStack

DATA SEGMENT

KEEP_CS DW 0 ; для хранения сегмента

KEEP_IP DW 0 ; и смещения прерывания

message db 'hello \$'

end_message db 'end!\$'

DATA ENDS

Astack SEGMENT STACK

DW 512 DUP('0')

Astack ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:Astack

WRITE PROC FAR

push ax

push bx

push cx

push dx

mov dx, offset message

mov ah, 09h

output:

int 21h

loop output

mov ah, 0

int 1Ah

add bx, dx

delay:

mov ah, 0

int 1Ah

cmp bx, dx

jg delay

mov dx, offset end_message

mov ah, 09h

int 21h

pop ax

pop bx

pop cx

pop dx

mov al, 20h

out 20h, al

iret

WRITE ENDP

MAIN PROC FAR

push DS

sub AX, AX

push AX

mov AX, DATA

mov DS, AX

mov AX, 3523h

int 21h

mov KEEP_IP, BX

mov KEEP_CS, ES

push DS

mov bx, 30h

mov cx, 05h

mov DX, OFFSET WRITE

mov AX, SEG WRITE

mov DS, AX

mov AH, 25h

mov AL, 23h

int 21H

pop DS

read:

mov AH, 0

int 16h

cmp AL, 3

je do

jmp read

do:

int 23h

cli

push DS

mov DX, KEEP_IP

mov AX, KEEP_CS

mov DS, AX

mov AX, 2523h

int 21h

pop DS

sti

ret FAR

MAIN ENDP

CODE ENDS

END MAIN