Лабораторна робота №17

З системного програмування

Виконав – **Нестерук Юрій, гр. ІО-12**

**Варіант**:

|  |  |  |
| --- | --- | --- |
| Варіант | № варіанта mod 10 | Функція |
| 220 | 0 | sin(x) |

**Код програми:**

EXTRN \_printf:near, \_scanf:near, \_sin:near, \_\_astart:near ;определение ссылок

;на точку входа библиотеки и библиотечные функции

EXTRN \_\_fltused:word

.8087

DGROUP GROUP \_DATA

\_DATA SEGMENT WORD PUBLIC 'DATA'

inputmsg db "Enter x",0Dh, 0Ah, 0

arg dd ?

two dd 2

tan dd ?

formatIn db "%f", 0 ;строка формата для ввода аргумента с помощью \_scanf

formatOut db "%f", 0Dh, 0Ah, 0 ;строка формата для вывода результата с помощью \_printf

\_DATA ENDS ;конец сегмента данных

\_TEXT SEGMENT WORD PUBLIC 'CODE'

ASSUME CS:\_TEXT, DS:DGROUP

**print** macro msg ;макрос вывода строки без форматирования

MOV DX, offset DGROUP:msg ;сохранение адреса строки

PUSH DX ;запись адреса строки в стек

CALL \_printf ;вызов функции вывода \_printf

ADD SP, 2 ;восстановление стека

ENDM ;конец макроса

**printform** macro ;макрос вывода данных с форматированием

;(данные должны быть записаны в стек до вызова макроса)

MOV DX, OFFSET DGROUP:formatOut ;сохранение адреса строки форматирования в стек

PUSH DX ;запись адреса строки форматирования в стек

CALL \_printf ;вызов функции вывода \_printf

ADD SP,10 ;восстановление стека (8 байтов на данные и 2 на адрес)

ENDM

lab: JMP \_\_astart ;переход на точку входа библиотеки

\_main proc ;обьявление выполняющейся процедуры

public \_main

print inputmsg ;вывод приветственной строки

MOV DX, OFFSET DGROUP:arg ;сохранение адреса области данных для аргумента

PUSH DX ;запись адреса в стек

MOV DX, OFFSET DGROUP:formatIn ;сохранение адреса строки форматирования для ввода

PUSH DX ;запись адреса в стек

CALL \_scanf ;ввод аргумента

ADD SP, 4 ;восстановление стека (по 2 байта на оба адреса)

FLD arg ;загрузка аргумента в стек математического сопроцессора

SUB SP,8 ;подготовка места в стеке главного процессора

MOV BP, SP ;загрузка указателя стека в базовый регистр косвенной адресации

FSTP qword ptr[BP] ;выгрузка аргумента из стека математического сопроцессора в стек

;главного процессора, с освобождением вершины в сопроцессоре

printform ;вывод аргумента

;2\*tg(x/2)/(1 + tg(x/2)\*tg(x/2))

fld1

fld arg ;x

fidiv two ;x/2

fptan ;1 | tg(x/2)

fxch st(1) ;tg(x/2) | 1

fst tan ;tan := tg(x/2)

fmul tan ;tg(x/2)\*tan | 1

fadd st, st(1) ;tg(x/2)\*tg(x/2) + 1

fld1 ;tg(x/2)\*tg(x/2) + 1 | 1

fdiv st, st(1) ;1/(tg(x/2)\*tg(x/2) + 1)

fmul tan ;tg(x/2)/(tg(x/2)\*tg(x/2) + 1)

fimul two ;2\*tg(x/2)/(1 + tg(x/2)\*tg(x/2))

SUB SP,8 ;подготовка места в стеке главного процессора

MOV BP, SP ;загрузка указателя стека в базовый регистр косвенной адресации

FSTP qword ptr[BP] ;выгрузка аргумента из стека математического сопроцессора в стек

printform ;вывод аргумента

FLD arg ;загрузка аргумента в стек математического сопроцессора

SUB SP, 8 ;подготовка места в стеке главного процессора

MOV BP, SP ;загрузка указателя стека в базовый регистр косвенной адресации

FSTP qword ptr[BP] ;выгрузка аргумента из стека математического сопроцессора в стек

;главного процессора, с освобождением вершины в сопроцессоре

CALL \_sin ;вызов библиотечной функции

MOV BP, AX ;сохранение адреса результата

PUSH [BP+6] ;запись результата по адресу в стек. Результат занимает 8 байт

PUSH [BP+4] ;вместо использования математического сопроцессора

PUSH [BP+2]

PUSH [BP]

printform ;вывод аргумента

MOV AH, 1 ;окончание процедуры по нажатию клавишы

INT 21h

MOV AX, 4C00h

INT 21h

ret ;возвращение из процедуры

\_main endp ;конец выполняющей процедуры

\_TEXT ENDS ;конец сегмента кода

END lab