«Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Боровцов Е.Г.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Отчёт

Лабораторной работе №2

**«**Программирование на языке ассемблера IBM PC и использованиесистемы прерываний»

по дисциплине «Машинно-зависимые языки программирования»

Студент группы ПИ 92 В.М. Шульпов

Преподаватель доцент, к.т. н. Боровцов Е.Г.

Барнаул 2021

Цель лабораторной работы:Закрепление знакомства с системой команд Intel80x86. Ознакомление с общей  
структурой программы, директивами ассемблера IBM PC, использованием системы  
программных прерываний.  
Задание к лабораторной работе:Реализовать программу, используя систему команд и язык ассемблера IBM PC

Задание:

1. 

.model tiny ; MASM's support for COM files

.dosseg ; Make sure you are using dos segment CODE, DATA + STACK

.data

A dw ?

B dw ?

C\_1 dw ?

D dw ?

Y dw ?

input\_buf db 06,00,5 dup(?)

in\_A db 0ah, 0dh, 'Input A :$'

in\_B db 0ah, 0dh, 'Input B :$'

in\_C db 0ah, 0dh, 'Input C :$'

in\_D db 0ah, 0dh, 'Input D :$'

answer db 7 dup(?), '$' ;ввод

messout db 0dh, 0ah, 'result : $' ;вывод

.code

.startup

beg: jmp start

include bin2str.asm

include str2bin.asm

;==================ввод=====================================

start: lea dx,in\_A ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov A,ax ; запомнить число в переменной A

;==================ввод=====================================

lea dx,in\_B ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov B,ax ; запомнить число в переменной B

;==================ввод=====================================

lea dx,in\_C ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov C\_1,ax ; запомнить число в переменной C

; ==================ввод=====================================

lea dx,in\_D ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov D,ax ; запомнить число в переменной D

;y=a/b-c/(d\*d)

mov ax,D ; в регистр ax заносится D

imul ax ; ax = D\*D

mov bx,C\_1 ; в регистр ax заносится C

xchg ax,bx ; ax=C, bx=D\*D

idiv bx ; ax = C/(D\*D)

mov bx,ax ; bx = C/(D\*D)

mov ax,A ; в регистр bx заносится A

mov cx,B ; в регистр cx заносится B

idiv cx ; ax = A/B

sub ax,bx ; ax = A/B-C/(D\*D)

mov Y,ax ;результат поместить в Y

lea bx,answer ; поместить в BX адрес буфера для символьного представления

call bin2str ; обратиться к подпрограмме преобразования

lea dx,messout ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

lea dx,answer+1

mov ah,09h

int 21h

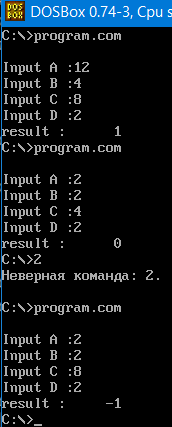
int 20h ; конец программы

end

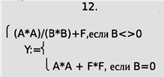
|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 2 |
| В | 2 |
| С | 4 |
| D | 2 |
| result | 0 |

|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 2 |
| В | 2 |
| С | 8 |
| D | 2 |
| result | -1 |

|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 12 |
| В | 3 |
| С | 8 |
| D | 2 |
| result | 1 |



2)



.model tiny ; MASM's support for COM files

.dosseg ; Make sure you are using dos segment CODE, DATA + STACK

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Our data section. Here we declare our strings for our console message

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.data

A dw ?

B dw ?

F dw ?

input\_buf db 06,00,5 dup(?)

in\_A db 0ah, 0dh, 'Input A :$'

in\_B db 0ah, 0dh, 'Input B :$'

in\_F db 0ah, 0dh, 'Input F :$'

answer db 7 dup(?), '$' ;ввод

messout db 0dh, 0ah, 'result : $' ;вывод

.code

.startup

beg: jmp start

include bin2str.asm

include str2bin.asm

;==================ввод=====================================

start: lea dx,in\_A ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov A,ax ; запомнить число в переменной A

;==================ввод=====================================

lea dx,in\_B ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov B,ax ; запомнить число в переменной B

;==================ввод=====================================

lea dx,in\_F ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov F,ax ; запомнить число в переменной F

; (A\*A)/(B\*B)+F если B<>0; A\*A+F\*F, если B=0.

mov ax,B ;

cmp ax,0 ; сравнить B с нулём

jz func2 ; переход по нулю на функцию с алгоритмом: Y=A\*A+F\*F

mov ax,B ;

imul ax ; ax= B\*B

mov bx,A ; bx=A

xchg ax,bx ; ax=A, bx=B\*B

imul ax ; ax=A\*A

idiv bx ; ax=(A\*A)/(B\*B)

mov bx,F ;

add ax,bx ; ax= (A\*A)/(B\*B)+F

jmp output ; безусловный переход на вывод

func2: mov ax,A ;

imul ax ; ax= A\*A

mov bx,F ;

xchg ax,bx ; ax=F, bx=A\*A

imul ax ; ax= F\*F

add ax,bx ; ax=A\*A+F\*F

output: lea bx,answer ; поместить в BX адрес буфера для символьного представления

call bin2str ; обратиться к подпрограмме преобразования

lea dx,messout ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

lea dx,answer+1

mov ah,09h

int 21h

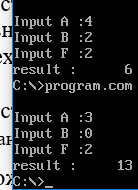
;Y=A-(B+C)^2/D

int 20h ; завершить работу

end

|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 4 |
| В | 2 |
| F | 2 |
| result | 6 |

|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 3 |
| В | 0 |
| F | 2 |
| result | 13 |



**3)**



.model tiny ; MASM's support for COM files

.dosseg ; Make sure you are using dos segment CODE, DATA + STACK

.data

A dw ?

N dw ?

input\_buf db 06,00,5 dup(?)

in\_A db 0ah, 0dh, 'Input A :$'

in\_N db 0ah, 0dh, 'Input N :$'

answer db 7 dup(?), '$' ;ввод

messout db 0dh, 0ah, 'result : $' ;вывод1

messout2 db 0dh, 0ah, 'error N<2$' ; вывод 2

.code

.startup

beg: jmp start

include bin2str.asm

include str2bin.asm

;==================ввод=====================================

start: lea dx,in\_A ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov A,ax ; запомнить число в переменной A

;==================ввод=====================================

lea dx,in\_N ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov N,ax ; запомнить число в переменной N

;Y=...

mov cx,N ;

cmp cx,2 ; сравнить N с 2 (при N<2 функция не имеет смысла)

jb err ; переход на вывод ошибки, если N<2

sub cx,1 ; cx=N-1

mov bx,1 ; занести 1 в bx для накопления произведения

loop\_: push cx ; сохранить счетчик цикла в стеке

mov ax,cx ; положить I в AX (подсчет с конца)

add ax,A ; ax = I + A

imul bx ; на 1 кр ax = (A+N), на 2 - ax = (A+N)(A+N-1)

xchg ax,bx ; ax=1,bx=(A+N)(A+N-1)\*...

pop cx ; вытянуть счетчик цикла из стека

loop loop\_ ;

mov ax,bx ; результат в ax

lea bx,answer ; поместить в BX адрес буфера для символьного представления

call bin2str ; обратиться к подпрограмме преобразования

lea dx,messout ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

lea dx,answer+1

mov ah,09h

int 21h

jmp end\_ ; безусловный переход на конец программы

err:lea dx,messout2 ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

end\_:int 20h ; завершить работу

end

|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 2 |
| N | 4 |
| result | 60 |

|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 2 |
| N | 2 |
| result | 3 |

|  |  |
| --- | --- |
| значение | 10 с.с. |
| А | 2 |
| N | 1 |
| result | error |

