MICROPROCESSOR ASSIGNMENTS

NAME : SHRIRANG. R. MHALGI

CLASS : S.E.

DIVISION : B

ROLL NO 222006

PROBLEM STATEMENT :

Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method. (use of 64-bit registers is expected).

CODE :

%macro scall 4

mov rax,%1

mov rdi,%2

mov rsi,%3

mov rdx,%4

syscall

%endmacro

section .data

n1 db "Enter the first 8 digit number : "

n1\_l equ $-n1

n2 db 10,"Enter the second 8 digit number : "

n2\_l equ $-n2

res db 10,"Result is : "

res\_l equ $-res

endl db " ",10

endl\_l equ $-endl

menu db 10,"1: Succesive addition "

db 10,"2: Add and Shift "

db 10,"3: exit",10

menu\_l equ $-menu

section .bss

no1 resd 3 ;same as resb 9

no2 resd 3

result resd 3

choice resb 2

section .txt

global \_start

\_start:

scall 1,1,n1,n1\_l

scall 0,0,no1,9 ;9 because of ascii value of enter

mov rsi,no1

call accept

mov dword[no1],ebx

scall 1,1,n2,n2\_l

scall 0,0,no2,9

mov rsi,no2

call accept

mov dword[no2],ebx

scall 1,1,menu,menu\_l

scall 0,0,choice,2

mov al,[choice]

cmp al,31h

je p1

cmp al,32h

je p2

p1 : call sadd

call exit\_p

p2 : call ash

call exit\_p

;----------------- succesive addition--------------------

sadd :

mov ecx,[no1]

mov ebx,[no2]

mov eax,0

loop : add eax,ebx

dec ecx

jnz loop

call display

ret

;-------------------- add and shift ---------------------

ash:

mov ecx,32

mov edx,0

mov eax,[no1]

mov ebx,[no2]

b : shl edx,1

shl ebx,1

jnc a

add edx,eax

a: dec ecx

jnz b

mov eax,edx

call display

ret

;----------------- accept -------------------------------

accept:

mov rcx,8

mov ebx,0

mov eax,0

back : rol ebx,4

mov al,[rsi]

cmp al,39h

jbe next

sub al,07h

next : sub al,30h

add ebx,eax

inc rsi

dec rcx

jnz back

ret

;--------------- display ---------------------------------

display:

mov rcx,8

mov qword[result],0

mov rbx,result

back1 : rol eax,4

mov dl,al

and dl,0Fh

cmp dl,09h

jbe next1

add dl,07h

next1 : add dl,30h

mov [rbx],dl

inc rbx

dec rcx

jnz back1

scall 1,1,res,res\_l

scall 1,1,result,8

scall 1,1,endl,endl\_l

ret

;---------------- exit -------------------------------

exit\_p:

mov rax,60

mov rdi,0

syscall

OUTPUT :

