

## Assignment No :8 Program :

%macro scall 4 ;macro to take input and output

mov rax,%1

mov rdi,%2

mov rsi,%3

mov rdx,%4

syscall

%endmacro

Section .data

title: db 0x0A,"----- BLock Transfer -----",0x0A

db " Overlapped with String Instruction",0x0A

t\_len: equ \$-title

copy: db 0x0A,0x0A," Copied data",

copy\_len: equ \$-copy

newline: db 0x0A

colon:db " : "

colon\_len: equ \$-colon

cnt\_a: db 05H

cnt\_a2:db 05H

cnt :db 05H

cnt2:db 05H

array: db 10H,20H,30H,40H,50H;data to be transferred

;----- BSS Section -----

Section .bss

address: resb 16

val: resb 2

copied: resb 5

choice: resb 2

;----- MAIN CODE Section -----

Section .text

global \_start

\_start:

```

scall 1,1,title,t_len
scall 0,0,choice,2 ;read choice
cmp byte[choice],'5' ;if choice==5 then exit
je EXIT

;----- Print Source Array ADDRESS: VALUE -----
mov byte[cnt_a],05h
mov rsi,array
label1:
push rsi
mov rbx,rsi
mov rdi,address
call HtoA_address
scall 1,1,newline,1
scall 1,1,address,16
scall 1,1,colon,colon_len
pop rsi
mov bl,byte[rsi]
push rsi
mov rdi,val
call HtoA_value
scall 1,1,val,2
pop rsi
inc rsi
dec byte[cnt_a]
jnz label1

;----- CHOOSE OPTION -----
;compare choice here
cmp byte[choice],'4'
je OVERLAPPED_STR
;-----overlapped with string instruction-----
OVERLAPPED_STR:
mov byte[cnt_a2],05H

```

```

mov rsi,array+04H
mov rdi,array+07H
STD
label6:
MOVSb
dec byte[cnt_a2]
jnz label6
jmp OUTPUT1
;-----OUTPUT of Overlapped -----
OUTPUT1:
mov cl,byte[array+4H]
mov byte[array+7H],cl
scall 1,1,copy,copy_len
mov byte[cnt_a],08H
mov rsi,array
;-----Printig ADDRESS:VALUE OF COPIED DATA -----
label3:
push rsi
mov rbx,rsi
mov rdi,address
call HtoA_address
scall 1,1,newline,1
scall 1,1,address,16
scall 1,1,colon,colon_len
pop rsi
mov bl,byte[rsi]
push rsi
mov rdi,val
call HtoA_value
scall 1,1,val,2
pop rsi
inc rsi

```

```

dec byte[cnt_a]
jnz label3
;jmp to start of program
jmp _start
EXIT:
mov rax,60
mov rdi,0
syscall

;-----HEX TO ASCII CONVERSION METHOD FOR ADDRESS -----
HtoA_address: ;hex_no to be converted is in ebx //result is stored in rdi/user defined variable
mov byte[cnt2],10H
aup:
rol rbx,04
mov cl,bl
and cl,0FH
cmp cl,09H
jbe ANEXT
ADD cl,07H
ANEXT:
add cl, 30H
mov byte[rdi],cl
INC rdi
dec byte[cnt2]
JNZ aup
ret

-----HEX TO ASCII CONVERSION METHOD FOR VALUE(2 DIGIT) -----
HtoA_value: ;hex_no to be converted is in ebx //result is stored in rdi/user defined variable
mov byte[cnt2],02H
aup1:
rol bl,04
mov cl,bl
and cl,0FH

```

```
CMP CL,09H
jbe ANEXT1
ADD cl,07H
ANEXT1:
add cl, 30H
mov byte[rdi],cl
INC rdi
dec byte[cnt2]
JNZ aup1
ret
```

#### OUTPUT :

```
----- BLock Transfer -----
Overlapped with String Instruction

000000000040205E : 10
000000000040205F : 20
0000000000402060 : 30
0000000000402061 : 40
0000000000402062 : 50

Copied data
000000000040205E : 10
000000000040205F : 20
0000000000402060 : 30
0000000000402061 : 10
0000000000402062 : 20
0000000000402063 : 30
0000000000402064 : 30
0000000000402065 : 30
----- BLock Transfer -----
Overlapped with String Instruction
```