**Overlapping.**

section .data

sblock db 10h,20h,30h,40h,59h

dblock db 00h,00h,00h,00h,00h

msg1 db 10,"Before block transfer",10

msg1\_len equ $-msg1 ;data section to declare and initialize required variables.

msg2 db 10,"After block transfer",10

msg2\_len equ $-msg2

smsg db 10,"Source Block is",10

smsg\_len equ $-smsg

dmsg db 10,"Destination Block is",10

dmsg\_len equ $-dmsg

space db " "

spaces equ $-space

%macro print 2 ;print macro to print messages.

mov rax,1

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall ;system call for print.

%endmacro

section .bss

char\_ans resb 16 ;.bss section to initialize runtime variables

section .text ;.text section to write business logic

global \_start

\_start:

print msg1,msg1\_len

print smsg,smsg\_len ;print source block before overlap

mov rsi,sblock

call disp\_block

print dmsg,dmsg\_len ;print destination block before overlap

mov rsi,dblock-2

call disp\_block

call block\_transfer ;goto block\_transfer label

print msg2,msg2\_len

print smsg,smsg\_len ;print destination block after overlap

mov rsi,sblock

call disp\_block

print dmsg,dmsg\_len ;print destination block after overlap

mov rsi,dblock-2

call disp\_block

mov rax,60

mov rdx,00 ;program termination

syscall

display: ;display method

mov rbx,16

mov rcx,5

mov rsi,char\_ans+1

cnt:

mov rdx,0

div rbx

cmp dl,09h ;compare dl with 09

jbe add30 ;goto add30 if it is below or equal

add dl,07h

add30:

add dl,30h

mov [rsi],dl

dec rsi

dec rcx

jnz cnt

print char\_ans,2

print space,spaces

ret

disp\_block: ;display block elements

mov rbp,5

next\_num:

mov al,[rsi]

push rsi

call display

pop rsi

inc rsi

dec rbp

jnz next\_num

ret

block\_transfer:

mov rsi,sblock+4 ;points to source block

mov rdi,dblock+2 ;points to destination block

mov rcx,5 ;set counter to 5

back:

mov al,[rsi]

mov [rdi],al

dec rsi

dec rdi

dec rcx

jnz back

ret