

Assignment No : 7 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 "Non Overlapped without string", 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],'1'  
JE NONOVERLAPPED  
;----Non overlapped copying without string instruction----
```

NONOVERLAPPED:

;---- Initialization of starting addresses

```
mov byte[cnt_a2],5H
mov rsi,array
mov rdi,array+20H
label2:
mov cl,00H
mov cl,byte[rsi]
mov byte[rdi],cl
inc rsi
inc rdi
dec byte[cnt_a2]
jnz label2
jmp OUTPUT
```

;-----OUTPUT of Non-Overlapped -----

OUTPUT:

```
scall 1,1,copy,copy_len
mov byte[cnt_a],05H
mov rsi,array+20H
jmp label3
;-----Printig ADDRESS:VALUE OF COPIED DATA -----
```

label3:

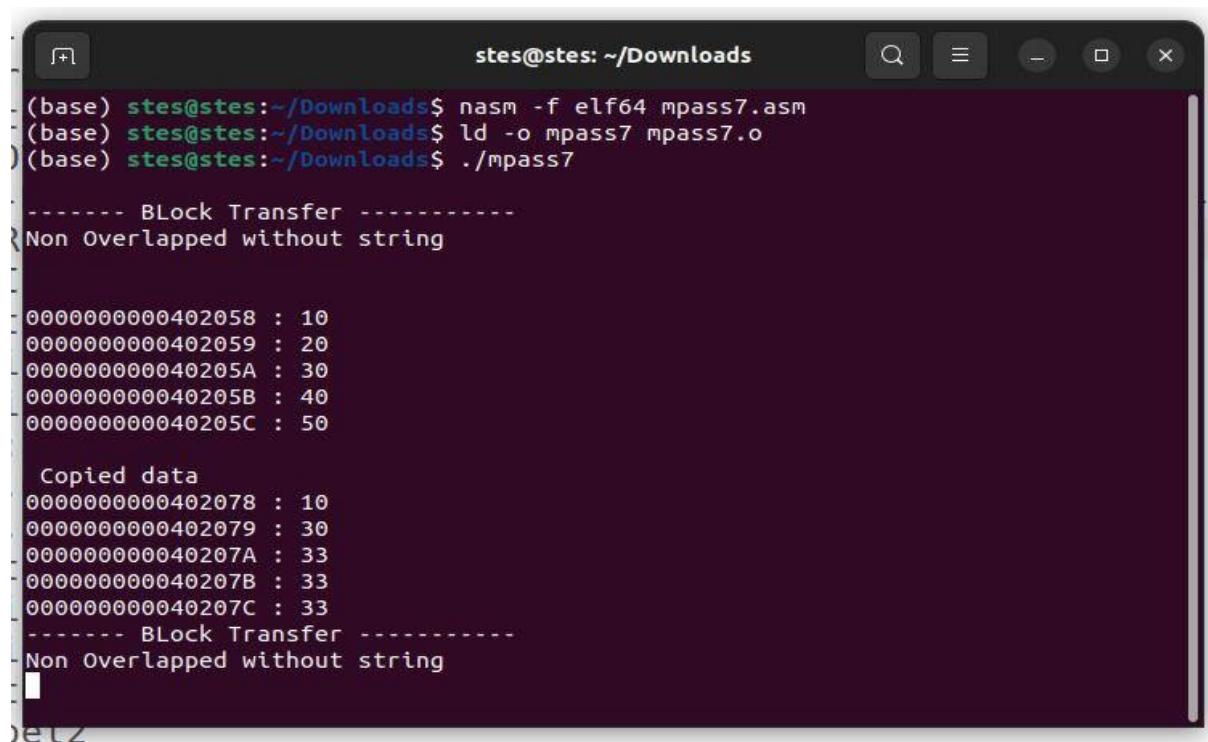
```
push rsi
mov rbx,rsi
mov rdi,address
call HtoA_address
scall 1,1,newline,1
scall 1,1,address,16
scall 1,1,colon,colon_len
pop rsi
mov bl,byte[rsi]
```

```
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:



The terminal window shows the following steps:

- (base) stes@stes:~/Downloads\$ nasm -f elf64 mpass7.asm
- (base) stes@stes:~/Downloads\$ ld -o mpass7 mpass7.o
- (base) stes@stes:~/Downloads\$./mpass7

Output from the program:

```
----- BLock Transfer -----
Non Overlapped without string

0000000000402058 : 10
0000000000402059 : 20
000000000040205A : 30
000000000040205B : 40
000000000040205C : 50

Copied data
0000000000402078 : 10
0000000000402079 : 30
000000000040207A : 33
000000000040207B : 33
000000000040207C : 33
----- BLock Transfer -----
Non Overlapped without string
```