

MICROPROCESSOR LABORATORY

ASSIGNMENT NO. 8

NAME :- OJUS PRAVIN JAISWAL

ROLL NO. :- SACO19108

DIVISION :- A

Assignment No. 8

Program :

;Overlapped Block Transfer

%macro print 2

Mov rax,1

Mov rdi,1

Mov rsi,%1

Mov rdx,%2

syscall

%endmacro

%macro exit 0

mov rax,60

mov rdi,0

syscall

%endmacro

section .data

sblock db 10h,20h,30h,40h,50h

dblock times 5 db 0

msg1 db 10,13,"Before Overlapped Block Transfer :- ",10,13

msg1_len equ \$-msg1

```
msg2 db 10,13,"After Overlapped Block Transfer :- ",10,13
msg2_len equ $-msg2
```

```
new_line db 10,13
```

```
smsg db 10, "Source Block is : "
smsg_len equ $-smsg
```

```
dmsg db 10,"Destination Block is : "
dmsg_len equ $-dmsg
```

```
space db " "
```

```
section .bss
Char_ans resb 2
```

```
Section .text
global _start
_start:
```

```
print msg1,msg1_len
print smsg,smsg_len
mov rsi,sblock
call block_display
```

```
print dmsg,dmsg_len
mov rsi,dblock-2
call block_display
```

```
call block_transfer
```

```
print new_line,1
```

```
print msg2,msg2_len
```

```
print smsg,smsg_len
```

```
mov rsi,sblock
```

```
call block_display
```

```
print dmsg,dmsg_len
```

```
mov rsi,dblock-2
```

```
call block_display
```

```
print new_line,1
```

```
exit
```

```
block_transfer:
```

```
mov rsi,sblock+4
```

```
mov rdi,dblock+4
```

```
mov rcx,5
```

```
up:
```

```
mov al,[rsi]
```

```
mov [rdi],al
```

```
dec rsi
```

```
dec rdi
```

```
dec rcx
```

```
jnz up
```

```
ret
```

block_display:

mov rbp,5

next:

mov al,[rsi]

push rsi

call display

print space,1

pop rsi

inc rsi

dec rbp

jnz next

ret

display:

Mov rsi,Char_ans+1

mov rcx,2

mov rbx,16

up1:

xor rdx,rdx

div rbx

cmp dl,09

jbe add30

add dl,07h

add30:

add dl,30h

mov [rsi],dl

dec rsi

dec rcx


jnz up1

print Char_ans,2

ret

```
</> Code  Input  >_ Output  Run  Save
1 ;Overlapped Block Transfer
2
3 %macro print 2
4 Mov rax,1
5 Mov rdi,1
6 Mov rsi,%1
7 Mov rdx,%2
8 syscall
9 %endmacro
10
11 %macro exit 0
12 mov rax,60
13 mov rdi,0
14 syscall
15 %endmacro
16
17 section .data
18 sblock db 10h,20h,30h,40h,50h
19 dblock times 5 db 0
20
```


```
</> Code  Input  >_ Output  Run  Save
21 msg1 db 10,13,"Before Overlapped Block Transfer :- ",10,13
22 msg1_len equ $-msg1
23
24 msg2 db 10,13,"After Overlapped Block Transfer :- ",10,13
25 msg2_len equ $-msg2
26
27 new_line db 10,13
28
29 smsg db 10, "Source Block is : "
30 smsg_len equ $-smsg
31
32 dmsg db 10,"Destination Block is : "
33 dmsg_len equ $-dmsg
34
35 space db " "
36
37 section .bss
38 Char_ans resb 2
39
40 Section .text
```

 Code  Input  Output

 Run

 Save




```
41 global _start
42 _start:
43
44 print msg1,msg1_len
45 print smsg,smsg_len
46 mov rsi,sblock
47 call block_display
48
49 print dmsg,dmsg_len
50 mov rsi,dblock-2
51 call block_display
52
53 call block_transfer
54
55 print new_line,1
56 print msg2,msg2_len
57 print smsg,smsg_len
58 mov rsi,sblock
59 call block_display
60
```

 Code  Input  Output

 Run

 Save

```
61 print dmsg,dmsg_len
62 mov rsi,dblock-2
63 call block_display
64 print new_line,1
65
66 exit
67
68 block_transfer:
69 mov rsi,sblock+4
70 mov rdi,dblock+4
71 mov rcx,5
72 up:
73 mov al,[rsi]
74 mov [rdi],al
75 dec rsi
76 dec rdi
77 dec rcx
78 jnz up
79 ret
80
```

 Code  Input  Output

 Run

 Save

```
81 block_display:
82 mov rbp,5
83 next:
84 mov al,[rsi]
85 push rsi
86 call display
87 print space,1
88 pop rsi
89 inc rsi
90 dec rbp
91 jnz next
92 ret
93
94 display:
95 mov rsi,Char_ans+1
96 mov rcx,2
97 mov rbx,16
98 up1:
99 xor rdx,rdx
100 div rbx
```

```
</> Code  Input  Output  Run  Save
92  ret
93
94  display:
95  mov rsi,Char_ans+1
96  mov rcx,2
97  mov rbx,16
98  up1:
99  xor rdx,rdx
100 div rbx
101 cmp dl,09
102 jbe add30
103 add dl,07h
104 add30:
105 add dl,30h
106 mov [rsi],dl
107 dec rsi
108 dec rcx
109 jnz up1
110 print Char_ans,2
111 ret
```

Output :

```
</> Code  Input  Output  Run  Save
Before Overlapped Block Transfer :-

Source Block is : 10 20 30 40 50
Destination Block is : 40 50 00 00 00

After Overlapped Block Transfer :-

Source Block is : 10 20 30 40 50
Destination Block is : 40 50 10 20 30

[Program exited with exit code 0]
```