

Assignment 2

Name: Prajwal Khobragade

Roll no: SYCOB138

section .data

```

nline      db      10,10
nline_len  equ     $-nline

space      db      " "

ano         db      10,"  Assignment no  :2-D",
              db      10,"-----",
              db      10,"  Block Transfer-Overlapped with String instruction.",
              db      10,"-----",10
ano_len     equ     $-ano

bmsg       db      10,"Before Transfer::"
bmsg_len   equ     $-bmsg

amsg       db      10,"After Transfer::"
amsg_len   equ     $-amsg

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

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

sblock     db      11h,22h,33h,44h,55h
dblock     times 5   db      0

```

section .bss

```
char_ans resB 2
```

```
%macro      Print    2
    MOV     RAX,1
    MOV     RDI,1
    MOV     RSI,%1
    MOV     RDX,%2
    syscall
%endmacro
```

```
%macro      Read     2
    MOV     RAX,0
    MOV     RDI,0
    MOV     RSI,%1
    MOV     RDX,%2
    syscall
%endmacro
```

```
%macro Exit 0
    Print   nline,nline_len
    MOV     RAX,60
    MOV     RDI,0
    syscall
%endmacro
```

```
;------
```

```
section .text
    global _start
```

```
_start:
    Print   ano,ano_len

    Print   bmsg,bmsg_len

    Print   smsg,smsg_len
```

```
mov    rsi,sblock
call   disp_block
```

```
Print  dmsg,dmsg_len
mov    rsi,dblock-2
call   disp_block
```

```
call   BT_OS
```

```
Print  amsg,amsg_len
```

```
Print  smsg,smsg_len
mov    rsi,sblock
call   disp_block
```

```
Print  dmsg,dmsg_len
mov    rsi,dblock-2
call   disp_block
```

Exit

;-----

BT_OS:

; Block transfer function with overlapping blocks

mov rsi, sblock+4 ; Source pointer at the end of sblock (0+4=4)

mov rdi, dblock+2 ; Destination pointer at the end of dblock (-2+4=2)

mov rcx, 5 ; Set loop counter to 5

STD ; Set direction flag to decrement

REP MOVSB ; Repeat move byte from source to destination (decrementing both rsi and rdi), 5 times

RET

;-----

disp_block:

; Display block procedure

```
mov rbp, 5      ; Set loop counter to 5
```

```
next_num:
```

```
mov al, [rsi]   ; Load byte from source
```

```
push rsi       ; Save current value of rsi
```

```
; Call Disp_8 to display byte in hexadecimal
```

```
call Disp_8
```

```
Print space, 1 ; Print space character
```

```
pop rsi        ; Restore value of rsi
```

```
inc rsi        ; Increment source pointer
```

```
dec rbp        ; Decrement loop counter
```

```
jnz next_num   ; Loop until rbp is zero
```

```
RET
```

```
;-----
```

```
Disp_8:
```

```
; Display hexadecimal number
```

```
mov rsi, char_ans+1 ; Set rsi to point to the second byte of char_ans
```

```
mov rcx, 2         ; Set loop counter to 2
```

```
mov rbx, 16        ; Set base to 16 (hexadecimal)
```

```
next_digit:
```

```
xor rdx, rdx      ; Clear rdx for division
```

```
div rbx
```

```
; Convert remainder to ASCII character
```

```
cmp dl, 9         ; Check if remainder is less than 10
```

```
jbe add30        ; Jump if less than or equal to 9
```

```
add dl, 7         ; Adjust remainder for letters A-F
```

```
add30:
```

```
add dl, 30h      ; Convert remainder to ASCII character
mov [rsi], dl    ; Store character in char_ans buffer
dec rsi         ; Decrement rsi to point to the next byte
dec rcx         ; Decrement loop counter
```

```
; Loop until rcx is zero
```

```
jnz next_digit
```

```
; Print the content of char_ans buffer
```

```
Print char_ans, 2
```

```
RET
```

```
;-----
```

Output

Assignment no :2-D

Block Transfer-Overlapped with String instruction.

Before Transfer::

Source Block	:11 22 33 44 55
Destination Block	:44 55 00 00 00

After Transfer::

Source Block	:11 22 33 11 22
Destination Block	:11 22 33 44 55

[Execution complete with exit code 0]