Name: Vinayak Madan Shete

Roll No.: SYCOC303 Course Name: Microprocessor Architecture Lab

Div: C Batch: C4 Course Code: BCE4302

==========

Problem Statement:

Write X86/64 ALP to perform non-overlapped block transfer without string specific instructions. Blocks containing data can be defined in the data segment.

==========

Input:

```
;Assignment No.06


;=====================================================================
=============
section .data
      nline db
10,"============================================================",10
      nline_len equ $-nline


      space db " "


      ano   db
10,"============================================================"
            db 10,"==========Assignment No.06=========="
            db 10,"Block Transfer==> Non-overlapped without string
instruction"
            db
10,"============================================================"
      ano_len equ $-ano


      bmsg db 10,"Before Tranfer==>"
      bmsg_len equ $-bmsg
```

```
	amsg db 10,"After Tranfer==>"
	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,12H,13H,14H,15H

	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 display_block

	print dmsg,dmsg_len
	mov rsi,dblock
	call display_block

	call BT_NO

	print amsg,amsg_len

	print smsg,smsg_len
	mov rsi,sblock
```

```
	call display_block

	print dmsg,dmsg_len
	mov rsi,dblock
	call display_block

	exit

;================================================================================
=============
;actual tranfer==>

BT_NO:
	mov rsi,sblock
	mov rdi,dblock
	mov rcx,5

	back:
		mov al,[rsi]
		mov [rdi],al
	inc rsi
	inc rdi

	dec rcx
	jnz back

	ret
;================================================================================
=============

display_block:
	mov rbp,5

	next_num:
		mov al,[rsi]
```

```
            push rsi

            call display_8
            print space,1

            pop rsi
            inc rsi

            dec rbp
            jnz next_num
      ret
;================================================================================
=============

display_8:
      mov rsi,char_ans+1
      mov rcx,2
      mov rbx,16

      next_digit:
            xor rdx,rdx
            div rbx

            cmp dl,9
            jbe add30
            add dl,07H

      add30:
            add dl,30H
            mov [rsi],dl

            dec rsi
            dec rcx
            jnz next_digit
```

```
    print char_ans,2


    ret
;=============================================================================
=============
```

Output:

```
pccoe@pccoe:~/Desktop/MAL$ nasm -f elf64 Assignment6.asm
pccoe@pccoe:~/Desktop/MAL$ ld Assignment6.o -o Assignment6
pccoe@pccoe:~/Desktop/MAL$ ./Assignment6

============================================================
==========Assignment No.06==========
Block Transfer==> Non-overlapped without string instruction
============================================================
Before Tranfer==>
Source Block==>11 12 13 14 15
Destination Block==>00 00 00 00 00
After Tranfer==>
Source Block==>11 12 13 14 15
Destination Block==>11 12 13 14 15
============================================================
pccoe@pccoe:~/Desktop/MAL$
```

==========