;-------------------------------------------------------------------------------

;Write X86/64 ALP to perform non-overlapped and overlapped block transfer ;(with and without string specific instructions). Block containing data can ;be defined in the data

;-------------------------------------------------------------------------------

segment.%macro display 2

mov rax,1

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endm

%macro accept 2

mov rax,0

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endm

global \_start:

section .data

msg : db ' ##################MENU################## ',10,13,

' (1) Non-overlapping with string ',10,13,

' (2) Non-overlapping without string ' ,10,13,

' (3) Overlapping with string',10,13,

' (4) Overlapping without string ',10,13,

' Enter your choice : '

len : equ $-msg

msg1: db ' Initial Block : '

len1: equ $-msg1

msg2: db 10,13,' Updated Block : '

len2: equ $-msg2

msg3: db 10,13,' Enter the number of blocks to be overlaped : '

len3: equ $-msg3

msg4: db 10,13,' Invalid Choice ',10,13,10,13

len4: equ $-msg4

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

section .bss

arr1 resb 05

initial resb 14

updated resb 29

choice resb 02

overlap resb 02

section .text

\_start:

display msg,len

accept choice,02

cmp byte[choice],31h

jne dw1

mov rsi,arr

mov rdi,initial

mov rdx,5

call ascii

mov rsi,arr

add rsi,4

mov rdi,rsi

add rdi,5

mov rcx,5

std

repz movsb

mov rsi,arr

mov rdi,updated

mov rdx,10

call ascii

display msg1,len1

syscall

display initial,14

display msg2,len2

syscall

display updated,29

jmp ter

dw1: cmp byte[choice],32h

jne dw2

mov rsi,arr

mov rdi,initial

mov rdx,5

call ascii

mov rsi,arr

add rsi,4

mov rdi,rsi

add rdi,5

mov rcx,5

up: mov al,[rsi]

mov [rdi],al

dec rsi

dec rdi

dec rcx

jnz up

mov rsi,arr

mov rdi,updated

mov rdx,10

call ascii

display msg1,len1

syscall

display initial,14

display msg2,len2

syscall

display updated,29

jmp ter

dw2: cmp byte[choice],33h

jne dw3

mov rsi,arr

mov rdi,initial

mov rdx,5

call ascii

display msg3,len3

accept overlap,02

mov al,05h

mov bl,byte[overlap]

sub bl,30h

sub al,bl

mov bl,al

mov rsi,arr

add rsi,4

mov rdi,rsi

add rdi,rbx

mov rcx,5

std

mov rcx,5

repz movsb

mov rsi,arr

mov rdi,updated

mov rdx,10

call ascii

display msg1,len1

syscall

display initial,14

display msg2,len2

syscall

display updated,29

jmp ter

dw3: cmp byte[choice],34h

jne dw4

mov rsi,arr

mov rdi,initial

mov rdx,5

call ascii

display msg3,len3

accept overlap,02

mov al,05h

mov bl,byte[overlap]

sub bl,30h

sub al,bl

mov bl,al

mov rsi,arr

add rsi,4

mov rdi,rsi

add rdi,rbx

mov rcx,5

up1: mov al,[rsi]

mov [rdi],al

dec rsi

dec rdi

dec rcx

jnz up1

mov rsi,arr

mov rdi,updated

mov rdx,10

call ascii

display msg1,len1

syscall

display initial,14

display msg2,len2

syscall

display updated,29

jmp ter

dw4: display msg4,len4

jmp ter

ter: mov rax,60

mov rdi,0

syscall

ascii: mov al,[rsi]

mov cl,2

u1: rol al,4

mov bl,al

and al,0Fh

cmp al,09h

ja d2

add al,30h

jmp d1

d2: add al,37h

d1: mov [rdi],al

mov al,bl

inc rdi

dec cl

jnz u1

mov byte[rdi],20h

inc rdi

inc rsi

dec rdx

jnz ascii

ret

**OUTPUT :**

manu@ubuntu:~/ML$ nasm -felf64 A2.nasm

manu@ubuntu:~/ML$ ld -o A2 A2.o

manu@ubuntu:~/ML$ ./A2

##################MENU##################

(1) Non-overlapping with string

(2) Non-overlapping without string

(3) Overlapping with string

(4) Overlapping without string

Enter your choice : 1

Initial Block : 10 20 30 40 50

Updated Block : 10 20 30 40 50 10 20 30 40 50

manu@ubuntu:~/ML$ ./A2

##################MENU##################

(1) Non-overlapping with string

(2) Non-overlapping without string

(3) Overlapping with string

(4) Overlapping without string

Enter your choice : 2

Initial Block : 10 20 30 40 50

Updated Block : 10 20 30 40 50 10 20 30 40 50

manu@ubuntu:~/ML$ ./A2

##################MENU##################

(1) Non-overlapping with string

(2) Non-overlapping without string

(3) Overlapping with string

(4) Overlapping without string

Enter your choice : 3

Enter the number of blocks to be overlaped : 3

Initial Block : 10 20 30 40 50

Updated Block : 10 20 10 20 30 40 50 00 00 00

manu@ubuntu:~/ML$ ./A2

##################MENU##################

(1) Non-overlapping with string

(2) Non-overlapping without string

(3) Overlapping with string

(4) Overlapping without string

Enter your choice : 4

Enter the number of blocks to be overlaped : 2

Initial Block : 10 20 30 40 50

Updated Block : 10 20 30 10 20 30 40 50 00 00

manu@ubuntu:~/ML$