Name: Prabodh Wankhede

Roll no. 57

Class: SE(B)

Batch – B3

Assignment No. 2

Problem Statement: 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.

**Code:**

**Program: Non-overlapped block transfer**

section .data  
    menumsg db 10,10,'\*\* Menu for Non Overlapped block Transfer++'  
    db 10,'1.Block Transfer Without using string instructions'  
    db 10,'2.Block Transfer With using string instructions'  
    db 10,'3.Exit',10  
    menumsg\_len equ $-menumsg  
      
    blk\_bfrmsg db 10,'Block Contents Before Transfer'  
    blk\_bfrmsg\_len equ $-blk\_bfrmsg  
      
    blk\_afrmsg db 10,'Block Contents After Transfer'  
    blk\_afrmsg\_len equ $-blk\_afrmsg  
      
    srcmsg db 10,'Source Block Contents::'  
    srcmsg\_len equ $-srcmsg  
      
    dstmsg db 10,'Destination Block Contents::'  
    dstmsg\_len equ $-dstmsg  
      
    srcblk db 01h,02h,03h,04h,05h  
    dstblk db 00,00,00,00,00  
    spacechar db 20h  
    spchlength equ $-spacechar  
      
section .bss  
    optionbuff resb 02  
    dispbuff resb 02  
      
%macro display 2  
    mov rax,01  
    mov rdi,01  
    mov rsi,%1  
    mov rdx,%2  
    syscall   
%endmacro  
      
%macro accept 2  
    mov rax,00  
    mov rdi,00  
    mov rsi,%1  
    mov rdx,%2  
    syscall   
%endmacro  
  
section .text  
global \_start  
\_start:  
    display blk\_bfrmsg,blk\_bfrmsg\_len  
    call dispsrc\_blk\_proc  
    call dispdest\_blk\_proc  
      
menu:   display menumsg,menumsg\_len  
    accept optionbuff,02  
    cmp byte [optionbuff],31h  
    je wos  
    cmp byte [optionbuff],32h  
    je ws  
      
exit:   mov rax,60  
    mov rbx,00  
    syscall  
      
dispsrc\_blk\_proc:  
    display srcmsg,srcmsg\_len  
    mov rsi,srcblk  
    mov rcx,05h  
      
up1:    push rcx  
    mov bl,[rsi]  
    push rsi  
    call disp8\_proc  
    display spacechar,spchlength  
    pop rsi  
    inc rsi  
    pop rcx  
    loop up1  
    ret  
  
dispdest\_blk\_proc:  
    display dstmsg,dstmsg\_len  
    mov rdi,dstblk  
    mov rcx,05h  
  
up2:   push rcx  
    mov bl,[rdi]  
    push rdi  
    call disp8\_proc  
    display spacechar,spchlength  
    pop rdi  
    inc rdi  
    pop rcx  
    loop up2  
    ret  
  
wos:  
    mov rsi,srcblk  
    mov rdi,dstblk  
    mov rcx,05  
      
again:  mov bl,[rsi]  
    mov [rdi],bl  
    inc rsi  
    inc rdi  
    loop again  
      
    display blk\_afrmsg,blk\_afrmsg\_len  
    call dispsrc\_blk\_proc  
    call dispdest\_blk\_proc  
    jmp menu  
      
ws:  
    mov rsi,srcblk  
    mov rdi,dstblk  
    mov rcx,05  
    cld  
    rep movsb  
    display blk\_afrmsg,blk\_afrmsg\_len  
    call dispsrc\_blk\_proc  
    call dispdest\_blk\_proc  
    jmp menu      
      
disp8\_proc:  
        mov ecx,2  
        mov edi,dispbuff  
back1:  
        rol bl,4  
        mov dl,bl  
        and dl,0FH  
        cmp dl,09H  
        jbe skip1  
        add dl,07H  
skip1:  
        add dl,30H  
     
        mov [edi],dl  
        inc edi  
        loop back1  
        display dispbuff, 2  
        ret

**OUTPUT:-**  
[HL@localhost ~]$ nasm -f elf64 blktr.asm  
[HL@localhost ~]$ ld -o blktr blktr.o  
[HL@localhost ~]$ ./blktr  
  
Block Contents Before Transfer  
Source Block Contents::01 02 03 04 05   
Destination Block Contents::00 00 00 00 00   
  
\*\* Menu for Non Overlapped block Transfer++  
1.Block Transfer Without using string instructions  
2.Block Transfer With using string instructions  
3.Exit  
1  
  
Block Contents After Transfer  
Source Block Contents::01 02 03 04 05   
Destination Block Contents::01 02 03 04 05   
  
\*\* Menu for Non Overlaped block Transfer++  
1.Block Transfer Without using string instructions  
2.Block Transfer With using string instructions  
3.Exit  
2  
  
Block Contents After Transfer  
Source Block Contents::01 02 03 04 05   
Destination Block Contents::01 02 03 04 05   
  
\*\* Menu for Non Overlaped block Transfer++  
1.Block Transfer Without using string instructions  
2.Block Transfer With using string instructions  
3.Exit  
3  
[HL@localhost ~]$

**Program: Overlapped Block Transfer**

section .data

menumsg db 10,10,'\*\* Menu for Overlaped block Transfer++'

db 10,'1.Block Transfer Without using string instructions'

db 10,'2.Block Transfer With using string instructions'

db 10,'3.Exit',10

menumsg\_len equ $-menumsg

blk\_bfrmsg db 10,'Block Contents Before Transfer'

blk\_bfrmsg\_len equ $-blk\_bfrmsg

blk\_afrmsg db 10,'Block Contents After Transfer'

blk\_afrmsg\_len equ $-blk\_afrmsg

srcmsg db 10,'Source Block Contents::'

srcmsg\_len equ $-srcmsg

posmsg db 10,10,'Enter Position to Overlap::'

posmsg\_len equ $-posmsg

srcblk db 01h,02h,03h,04h,05h,00h,00h,00h,00h,00h

spacechar db 20h

spchlength equ $-spacechar

section .bss

optionbuff resb 02

dispbuff resb 02

numascii resb 03

pos resb 00

%macro display 2

mov rax,01

mov rdi,01

mov rsi,%1

mov rdx,%2

syscall

%endmacro

%macro accept 2

mov rax,00

mov rdi,00

mov rsi,%1

mov rdx,%2

syscall

%endmacro

section .text

global \_start

\_start:

display blk\_bfrmsg,blk\_bfrmsg\_len

call dispsrc\_blk\_proc

display posmsg,posmsg\_len

accept numascii, 03

call packnum

menu: display menumsg,menumsg\_len

accept optionbuff,02

cmp byte [optionbuff],31h

je wos

cmp byte [optionbuff],32h

je ws

exit: mov rax,60

mov rbx,00

syscall

dispsrc\_blk\_proc:

display srcmsg,srcmsg\_len

mov rsi,srcblk

mov rcx,05h

add rcx,[pos]

up1: push rcx

mov bl,[rsi]

push rsi

call disp8\_proc

display spacechar,spchlength

pop rsi

inc rsi

pop rcx

loop up1

ret

wos:

mov rsi,srcblk +4

mov rdi,rsi

add rdi,[pos]

mov rcx,05

again: mov al,[rsi]

mov [rdi],al

dec rsi

dec rdi

loop again

display blk\_afrmsg,blk\_afrmsg\_len

call dispsrc\_blk\_proc

jmp exit

ws:

mov rsi,srcblk +4

mov rdi,rsi

add rdi,[pos]

mov rcx,05

std

rep movsb

display blk\_afrmsg,blk\_afrmsg\_len

call dispsrc\_blk\_proc

jmp exit

disp8\_proc:

mov ecx,2

mov edi,dispbuff

back1:

rol bl,4

mov dl,bl

and dl,0FH

cmp dl,09H

jbe skip1

add dl,07H

skip1:

add dl,30H

mov [edi],dl

inc edi

loop back1

display dispbuff, 2

ret

packnum:

mov bl,00

mov ecx,02

mov esi,numascii

UP1: rol bl,04

mov al,[esi]

sub al,30h

cmp al,09h

jbe skip2

sub al,07h

skip2: add bl,al

inc esi

loop UP1

mov [pos],bl

ret

**OUTPUT:-**

[HL@proj9-PC ~]$ nasm -f elf64 ass2.asm

[HL@proj9-PC ~]$ ld -o ass2 ass2.o

[HL@proj9-PC ~]$ ./ass2

Block Contents Before Transfer

Source Block Contents::01 02 03 04 05

Enter Position to Overlap::02

\*\* Menu for Overlaped block Transfer++

1.Block Transfer Without using string instructions

2.Block Transfer With using string instructions

3.Exit

1

Block Contents After Transfer

Source Block Contents::01 02 01 02 03 04 05 [HL@proj9-PC ~]$ nasm -f elf64 ass2.asm

[HL@proj9-PC ~]$ ld -o ass2 ass2.o

[HL@proj9-PC ~]$ ./ass2

Block Contents Before Transfer

Source Block Contents::01 02 03 04 05

Enter Position to Overlap::03

\*\* Menu for Overlaped block Transfer++

1.Block Transfer Without using string instructions

2.Block Transfer With using string instructions

3.Exit

2

Block Contents After Transfer

Source Block Contents::01 02 03 01 02 03 04 05 [HL@proj9-PC ~]$ ^C

[HL@proj9-PC ~]$