MICROPROCESSOR LABORATORY

Assignment No.6
X86/64 ALP to perform overlapped block transfer with string specific instructions. Block containing data can be defined in the data segment.

Program:

```
section .data
        menumsg db 10,'##### Menu for overlapped Block Transfer #####',10
                db 10,'1.Block Overlap without using string instructions'
                db 10,'2.Block Overlap with using string instructions'
                db 10,'3.Exit',10,10
        menumsg_len equ $-menumsg

   blk_bfrmsg db 10,10,'Block contents before Overlap'
        blk_bfrmsg_len equ $-blk_bfrmsg

        blk_afrmsg db 10,'Block contents after Overlap',10
        blk_afrmsg_len equ $-blk_afrmsg

        srcmsg db 10,'Source block contents::'
        srcmsg_len equ $-srcmsg

        posmsg db 10,10,10,'Enter position to overlap::'
        posmsg_len equ $-posmsg

        spacechar db 20h
        spchlength equ $-spacechar

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

section .bss
        optionbuff resb 02
        dispbuff resb 02
        numascii resb 03
        pos resb 00

%macro wrfun 4
        mov rax,%1
        mov rdi,%2
        mov rsi,%3
        mov rdx,%4
        syscall
%endmacro


section .text
global _start
   _start:
```

```
        wrfun 1,1,blk_bfrmsg,blk_bfrmsg_len
        call disp_src_blk_proc
        wrfun 1,1,posmsg,posmsg_len

        wrfun 0,0,numascii,3

        call packnum_proc

menu:
    wrfun 1,1,menumsg,menumsg_len

          wrfun 0,0,optionbuff,02

        cmp byte [optionbuff],31H
    je wos

          cmp byte [optionbuff],32H
          je ws


exit:
          mov rax,60            ;Exit
          mov rbx,00
          syscall

disp_src_blk_proc:
                wrfun 1,1,srcmsg,srcmsg_len
                mov rsi,srcblk
                mov rcx,05h
                add cl,[pos]

        up1:
          push rcx
                mov bl,[rsi]
                push rsi

                call disp8_proc

                wrfun 1,1,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

blkup1:
        mov al,[rsi]
        mov [rdi],al
        dec rsi
        dec rdi
        loop blkup1

        wrfun 1,1,blk_afrmsg,blk_afrmsg_len
        call disp_src_blk_proc

        jmp exit

ws:
        mov esi,srcblk+4
        mov edi,esi
        add edi,[pos]

        mov ecx,05

        std
        rep movsb

        wrfun 1,1,blk_afrmsg,blk_afrmsg_len
        call disp_src_blk_proc

   jmp exit

disp8_proc:
        mov ecx,2
        mov edi,dispbuff
dup1:
        rol bl,4
        mov al,bl
        and al,0fh
        cmp al,09
        jbe dskip
        add al,07h

dskip:
   add al,30h
        mov [edi],al
        inc edi
        loop dup1

        wrfun 1,1,dispbuff,2
        ret
```

```
packnum_proc:
        mov bx,0
        mov ecx,2
        mov esi,numascii

up2:
   rol bl,4
        mov al,[esi]
        sub al,30h
        cmp al,09h
        jbe skip5
        sub al,07h

skip5:
        add bl,al
        inc esi
        loop up2
   mov [pos],bl
        ret
```

Output:

```
guest-yi4DwY@student-OptiPlex-380:~$ nasm -f elf64 -o A6_String.asm A6_String.o
guest-yi4DwY@student-OptiPlex-380:~$ ld -o A6_String A6_String.o
guest-yi4DwY@student-OptiPlex-380:~$  ./ A6_String

Block contents before Overlap
Source block contents::01 02 03 04 05


Enter position to overlap:: 02
##### Menu for overlapped Block Transfer #####

1.Block Overlap without using string instructions
2.Block Overlap with using string instructions
3.Exit
1

Block contents after Overlap

Source block contents::01 02 01 02 03 04 05

Block contents before Overlap
Source block contents::01 02 03 04 05


Enter position to overlap::03
```

Menu for overlapped Block Transfer

1.Block Overlap without using string instructions
2.Block Overlap with using string instructions
3.Exit
2

Block contents after Overlap

Source block contents::01 02 03 01 02 03 04 05

Menu for overlapped Block Transfer

1.Block Overlap without using string instructions
2.Block Overlap with using string instructions
3.Exit
3

guest-yi4DwY@student-OptiPlex-380:~$