MICROPROCESSOR LABORATORY

Assignment No.5
X86/64 ALP to perform non-overlapped without string specific instructions. Block containing
data can be defined in the data segment.

Program:

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

section .data
   menumsg db 10,10,"*****Menu for Non-overlapped Block Transfer*****",10
         db 10,"1.Block Transfer without using String Instructions"
         db 10,"1.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

section .text
global _start
   _start:
      wfun 1,1,blk_afrmsg,blk_bfrmsg_len
      call dispsrc_blk_proc
```

```
call dispdest_blk_proc

menu:
   wfun 1,1,menumsg,menumsg_len
   wfun 0,0,optionbuff,02
   cmp byte[optionbuff],31h
   je wos
   cmp byte[optionbuff],32h
   je ws

exit:
   mov rax,60
   mov rdi,00
   syscall

dispsrc_blk_proc:
   wfun 1,1,srcmsg,srcmsg_len
   mov rsi,srcblk
   mov rcx,05h

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

   call disp8_proc

   wfun 1,1,spacechar,spchlength
   pop rsi
   inc rsi
   pop rcx
   loop up1
   ret

dispdest_blk_proc:
   wfun 1,1,dstmsg,dstmsg_len
   mov rdi,dstblk
   mov rcx,05h

up2:
   push rcx
   mov bl,[rdi]
   push rdi

   call disp8_proc

   wfun 1,1,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

    wfun 1,1,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

    wfun 1,1,blk_afrmsg,blk_afrmsg_len
    call dispsrc_blk_proc
    call dispdest_blk_proc
    jmp menu

disp8_proc:
    mov rsi,dispbuff
    mov rcx,02

dup1:
    rol bl,4
    mov dl,bl
    and dl,0Fh
    cmp dl,09h
    jbe dskip
    add dl,07h

dskip:
    add dl,30h
    mov [rsi],dl
    inc rsi
    loop dup1
```

```
    wfun 1,1,dispbuff,02
    ret
```

Output:

```
guest-yi4DwY@student-OptiPlex-380:~$ nasm -f elf64 -o A5_String.asm A5_String.o
guest-yi4DwY@student-OptiPlex-380:~$ ld -o A5_String A5_String.o
guest-yi4DwY@student-OptiPlex-380:~$  ./ A5_String

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
1.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-overlapped Block Transfer*****

1.Block Transfer without using String Instructions
1.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-overlapped Block Transfer*****

1.Block Transfer without using String Instructions
1.Block Transfer with using String Instructions
3.Exit
3
guest-yi4DwY@student-OptiPlex-380:~$
```