Assembly Language Report (Week 7)

Group 7

104502517戴逸任

104502518劉冠聲

Program CODE:

.data

myID DWORD 104502518d

size\_ID = 9

space BYTE 7 DUP(0)

myID\_result BYTE "104502518",0

.code

convert MACRO ID,Siz,Result

IFB<ID>

EXIT

ENDIF

IFB<Siz>

EXIT

ENDIF

IFB<Result>

EXIT

ENDIF

mov ecx,Siz

mov eax,ID

mov ebx,10

L:

mov edx,0

div ebx

test edx,1

jnz L2 ;odd

L1: ;even

mov edx,65

PUSH edx

jmp L3

L2: ;odd

mov edx,66

PUSH edx

L3:

loop L

mov ecx,Siz

mov edx,OFFSET Result

LL:

POP [edx]

inc edx

loop LL

ENDM

start@0 PROC

mov eax,myID

call WriteDec

call Crlf

convert myID, size\_ID, myID\_result

mov edx, OFFSET myID\_result

call WriteString

call Crlf

call WaitMsg

exit

start@0 ENDP

END start@0

Program Step & Register state:

mov ecx,Siz ;ecx=9

mov eax,ID ;eax=104502518

mov ebx,10 ;ebx=10

L:

mov edx,0 ;edx=0

div ebx ;edx=8🡪1🡪……🡪0🡪1

test edx,1

jnz L2

L1:

mov edx,65 ;edx=65

PUSH edx

jmp L3

L2:

mov edx,66 ;edx=66

PUSH edx

L3:

loop L

mov ecx,Siz ;ecx=9

mov edx,OFFSET Result ;0x0040500

LL:

POP [edx]

inc edx ;0x0040501🡪0x0040502……

loop LL

ENDM

start@0 PROC

mov eax,myID ;eax=104502518

call WriteDec

call Crlf

convert myID, size\_ID, myID\_result

mov edx, OFFSET myID\_result ;0x0040500

call WriteString

call Crlf

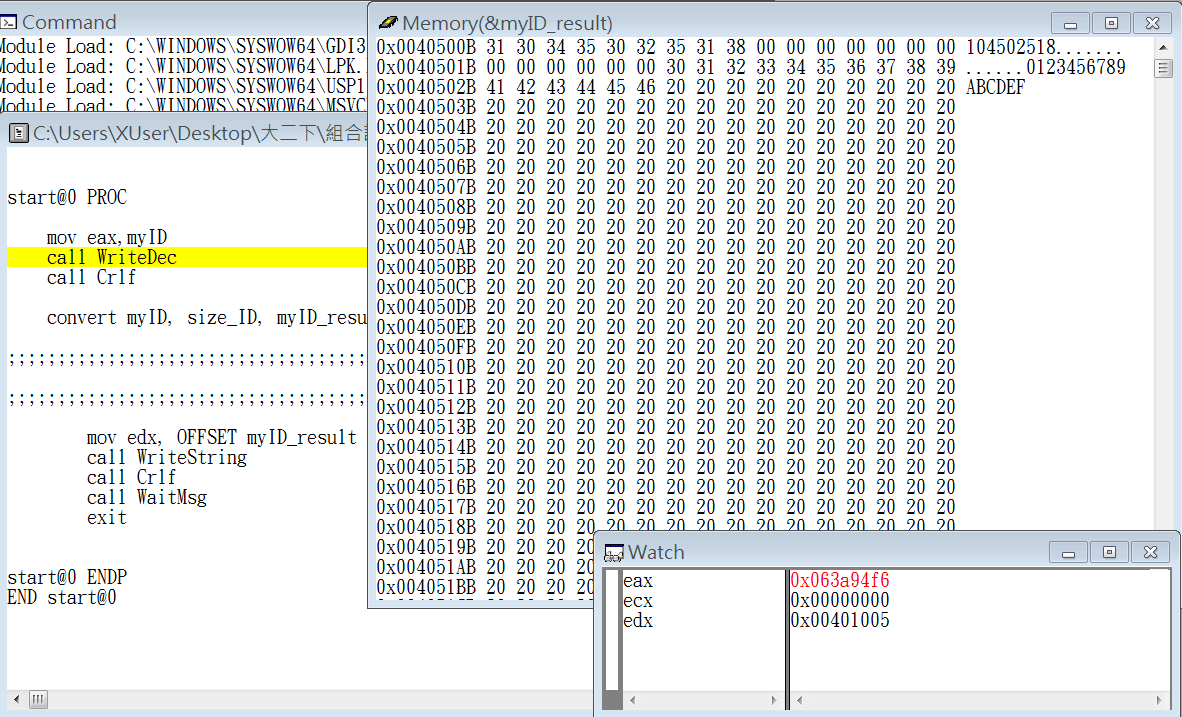
call WaitMsg

exit

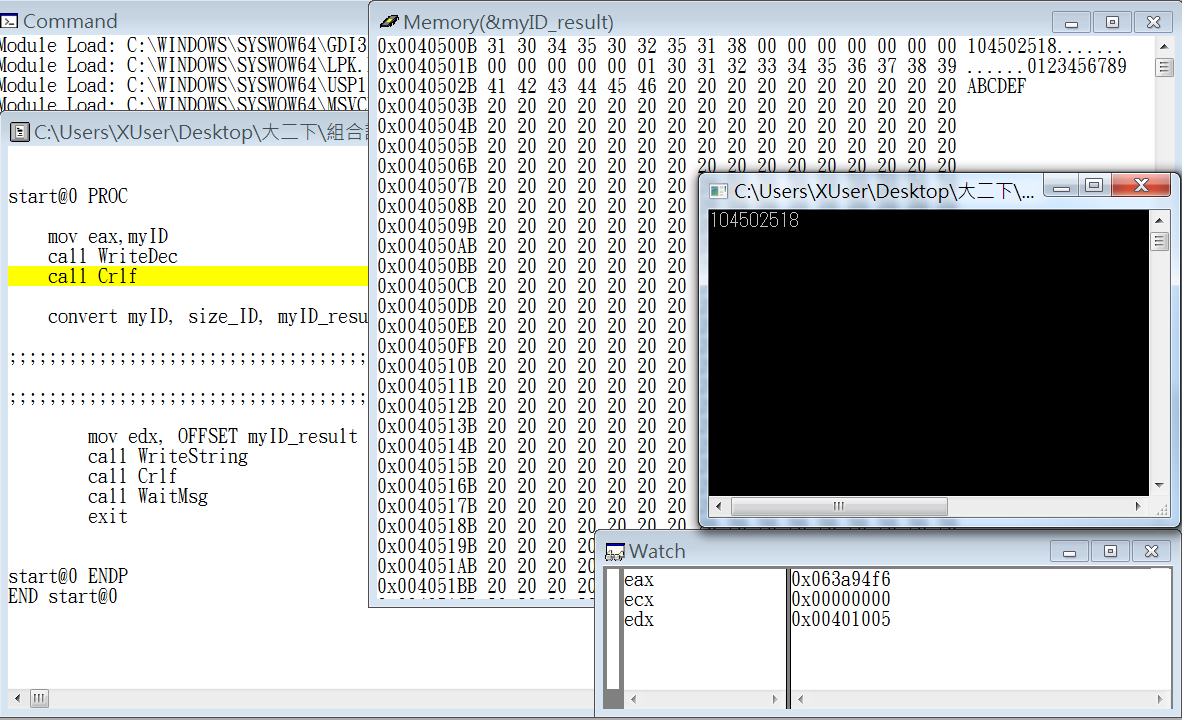
start@0 ENDP

END start@0

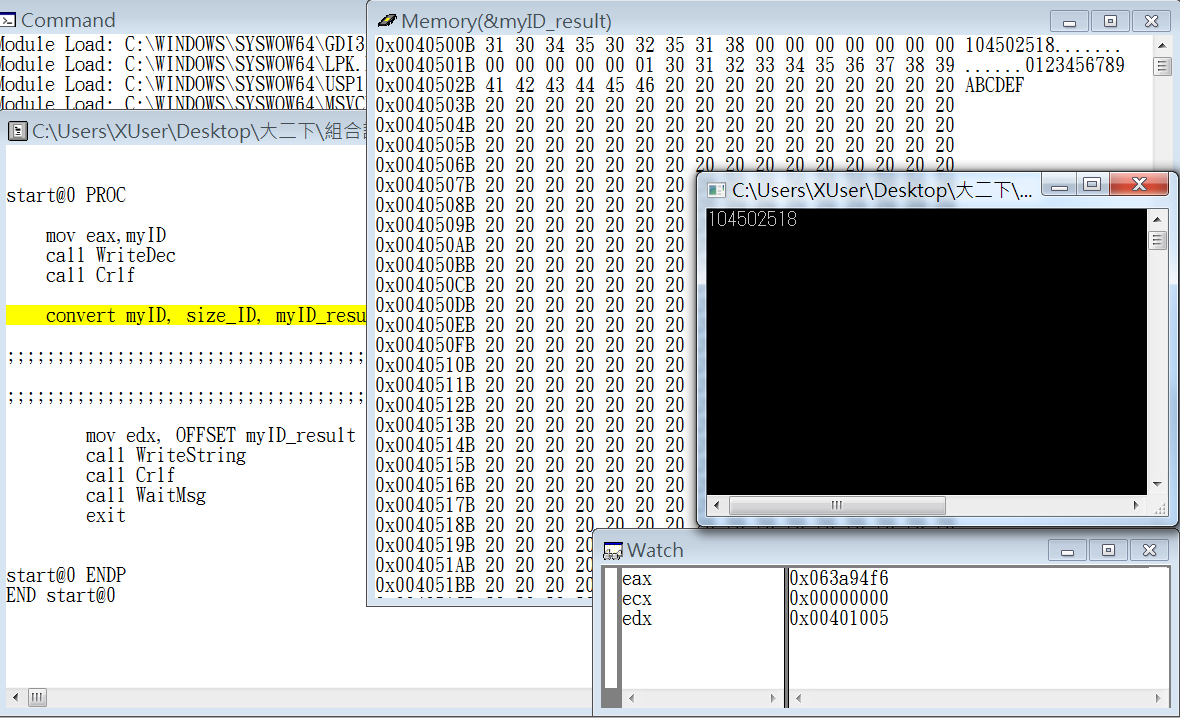
Picture & Discription:



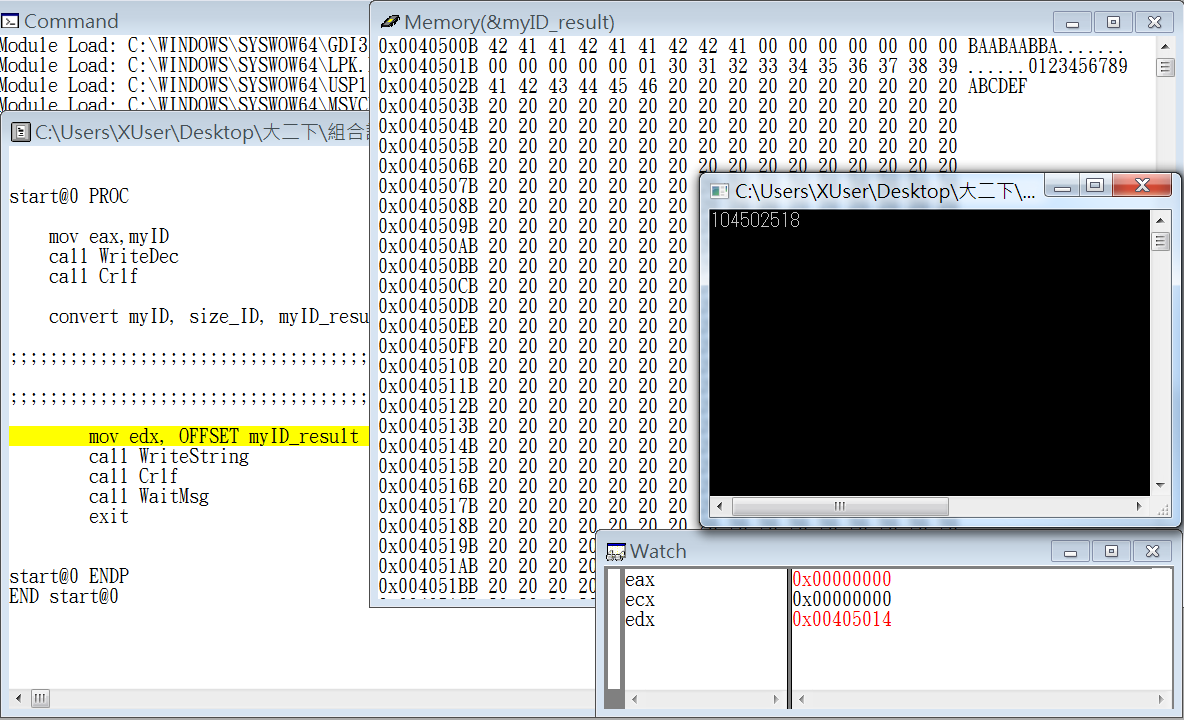
Step1: move myID into register eax



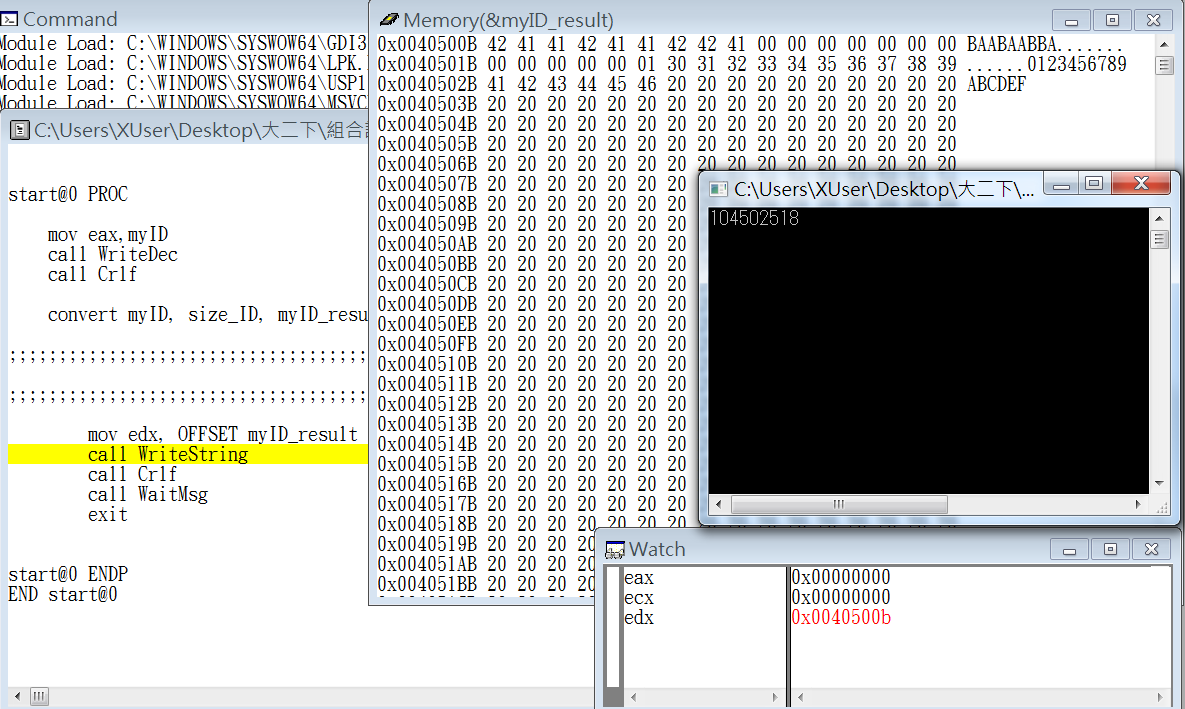
Step2: show myID in decimal on terminal



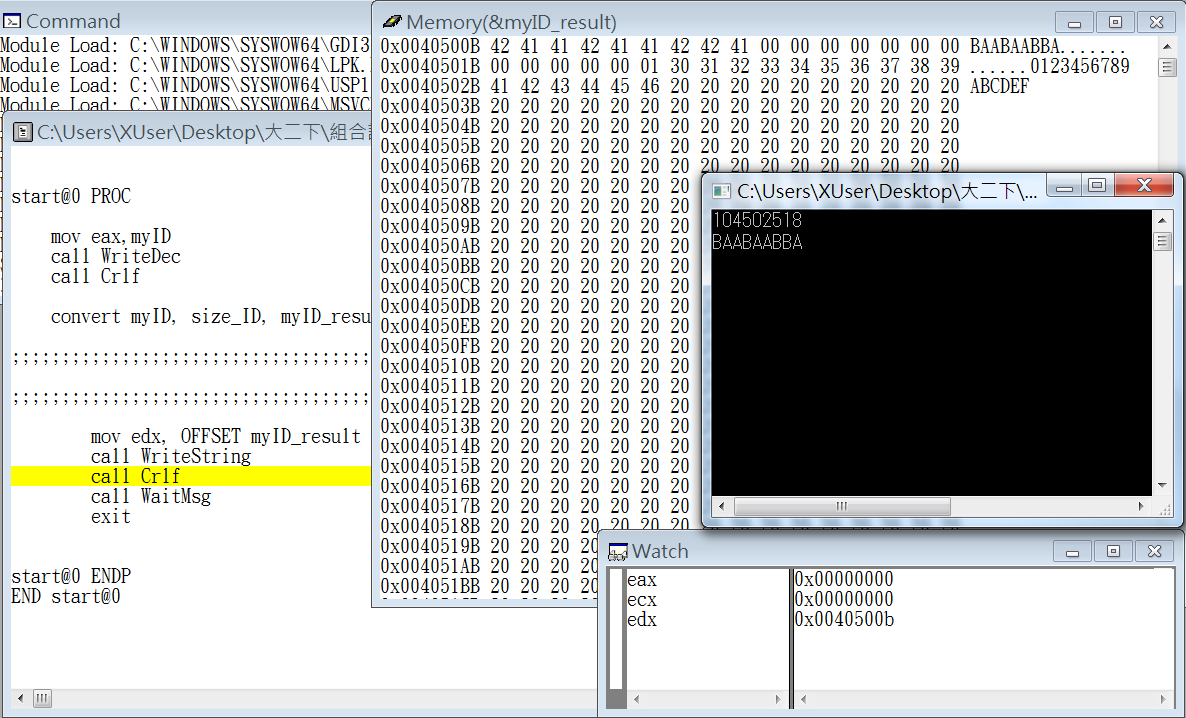
Step3: line feed in terminal



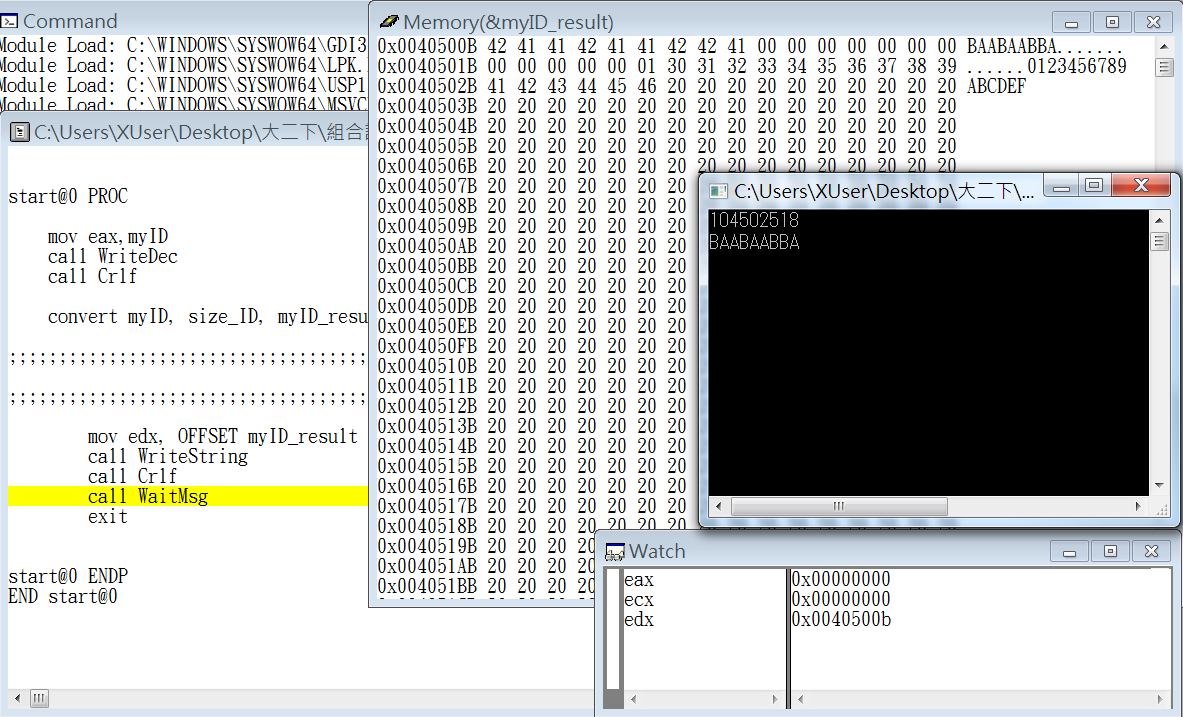
Step4: call convert function



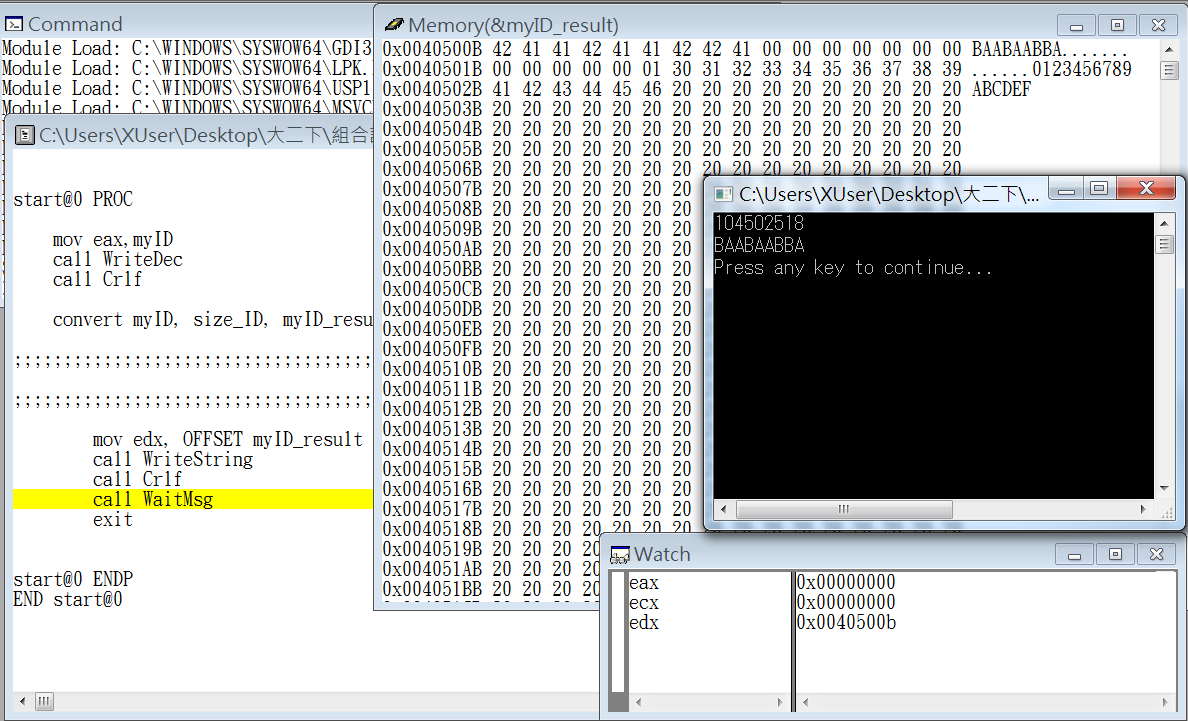
Step5: move the address of myID\_result into register edx



Step6: show the convterted string on terminal



Step7: line feed in terminal



Step8: end the program

Review:

The exercise this week is the last practice before the midterm exam, and the exercise this time is about use “MACRO” to help us finish the problem. In the first, we were not clearly know how use MACRO, after running error for many times, we finally successfully finished the problem. Until we run the Windbg step by step, we found that using MACRO is really faster because MACRO would run the program directly, and we don’t have to use a lot of time running the loop in the program. Now we just have to call MACRO for one time, and we can easily got the consequence.