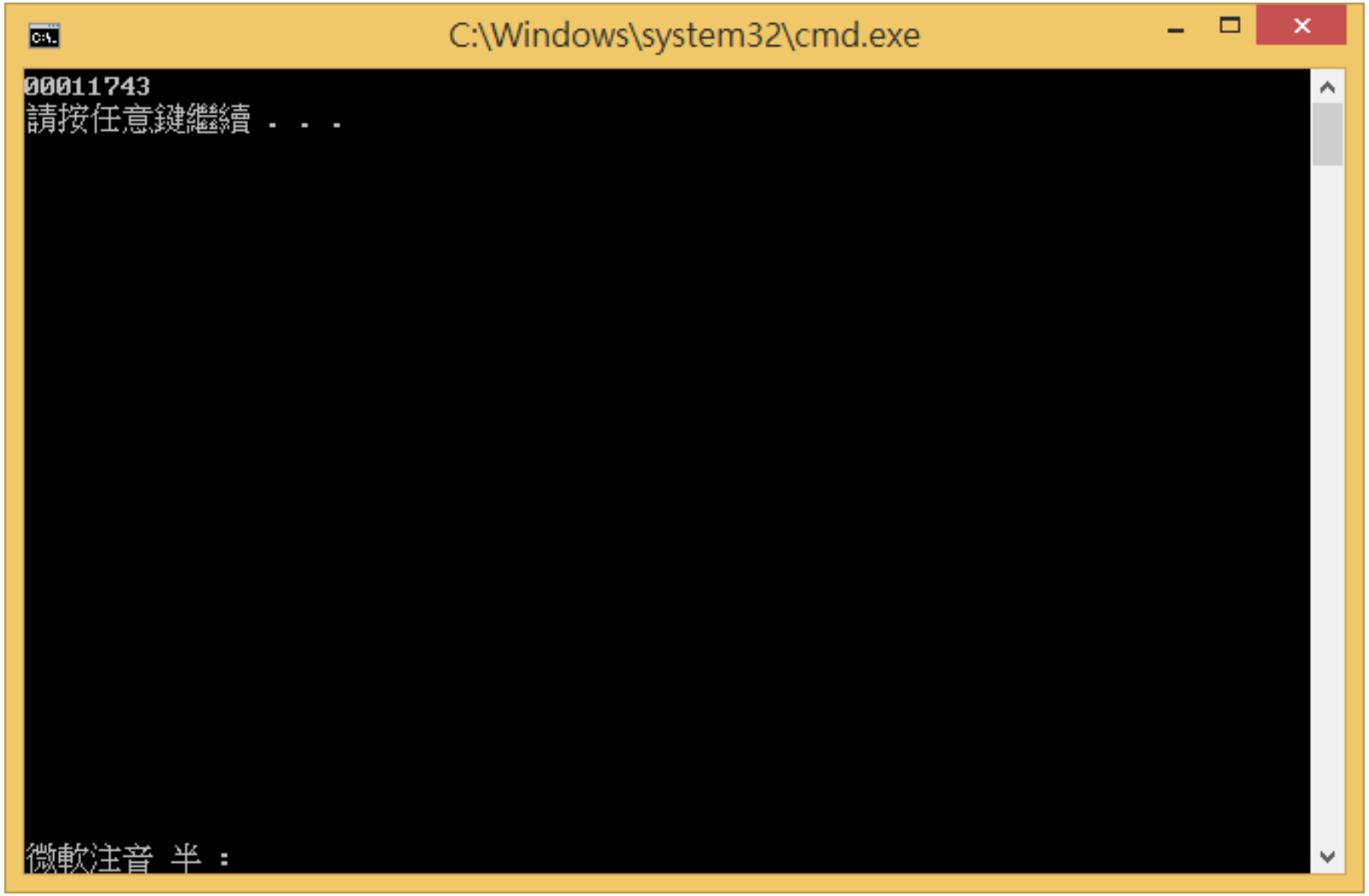
x86 Intel Assembly Language Programming
Some applications using Chapter 7 concepts

Some sample programs for Chapter 7

Dept. of CSIE,
Fu Jen Catholic University.
周賜福

```
TITLE Packed Decimal Examples          (AddPacked.asm)
; This program adds two decimal integers.
INCLUDE Irvine32.inc

.data
packed_1 WORD 4536h
packed_2 WORD 7207h
sum DWORD ?

.code
main PROC
; Initialize sum and index.
mov sum,0
mov esi,0

; Add low bytes.
mov  al,BYTE PTR packed_1[esi]
Add  al,BYTE PTR packed_2[esi]
daa
mov   BYTE PTR sum[esi],al

; Add high bytes, include carry.
inc esi
mov al,BYTE PTR packed_1[esi]
adc al,BYTE PTR packed_2[esi]
daa
movBYTE PTR sum[esi],al

; Add final carry, if any.
inc  esi
mov  al,0
adc  al,0
mov BYTE PTR sum[esi],al

; Display the sum in hexadecimal.
mov eax,sum
call  WriteHex
call  Crlf
exit
main ENDP
END main
```

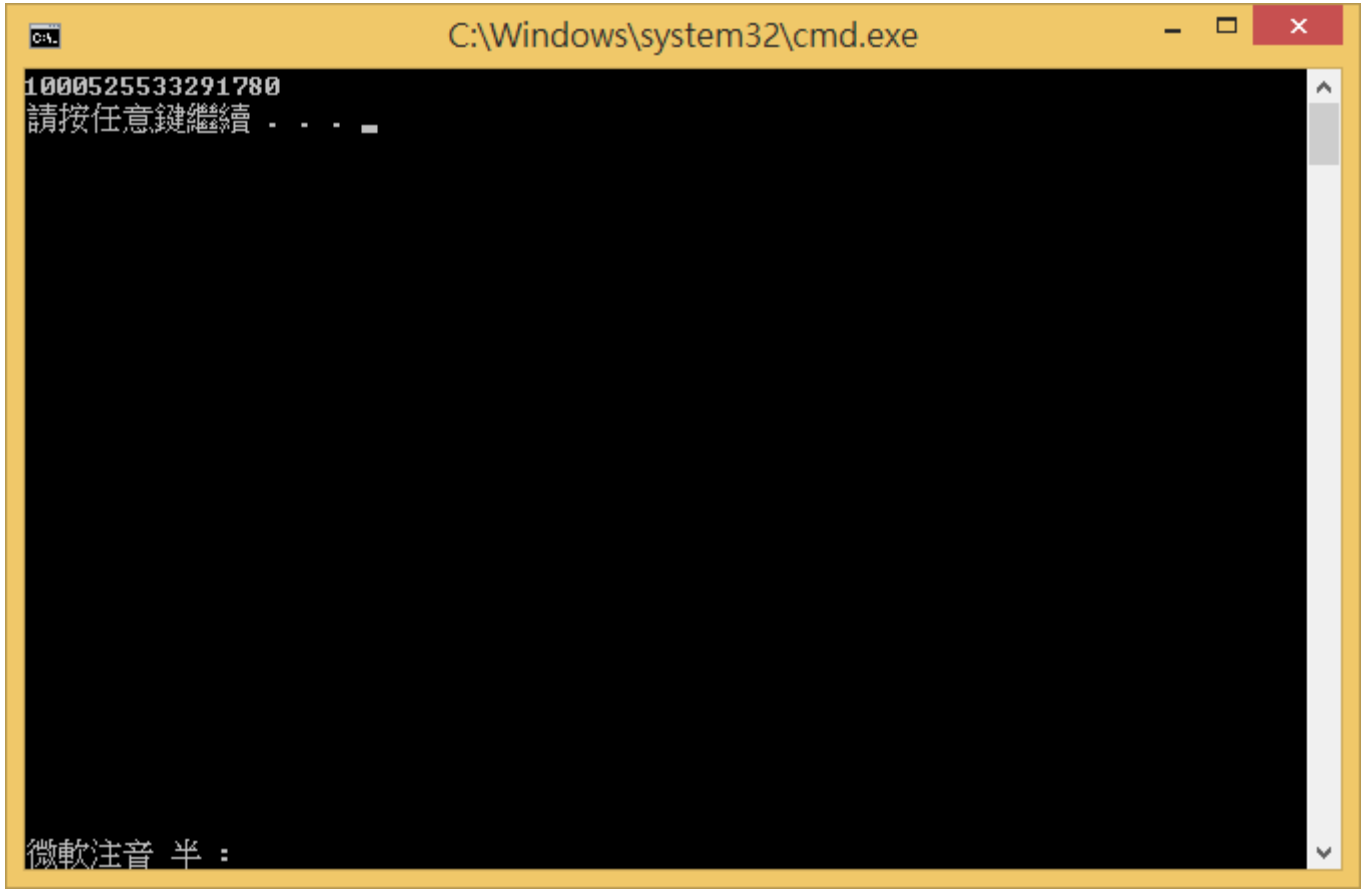
Result of running the program



```
TITLE ASCII Addition                    (ASCII_add.asm)

; This program performs ASCII arithmetic on digit strings having
; implied fixed decimal points.

INCLUDE Irvine32.inc

DECIMAL_OFFSET = 5                                          ; offset from right of string
.data
decimal_one BYTE "100123456789765"                          ; 1001234567.89765
decimal_two BYTE "900402076502015"                          ; 9004020765.02015
sum BYTE (SIZEOF decimal_one + 1) DUP(0),0

.code
main PROC

; Start at the last digit position.
        mov     esi,SIZEOF decimal_one - 1
        mov     edi,SIZEOF decimal_one
        mov     ecx,SIZEOF decimal_one
        mov     bh,0                                        ; set carry value to zero
```

```
L1:     mov     ah,0                    ; clear AH before addition
        mov     al,decimal_one[esi]     ; get the first digit
        add     al,bh                   ; add the previous carry
        aaa                             ; adjust the sum (AH = carry)
        mov     bh,ah                   ; save the carry in carry1
        or      bh,30h                  ; convert it to ASCII
        add     al,decimal_two[esi]     ; add the second digit
        aaa                             ; adjust the sum (AH = carry)
        or      bh,ah                   ; OR the carry with carry1
        or      bh,30h                  ; convert it to ASCII
        or      al,30h                  ; convert AL back to ASCII
        mov     sum[edi],al             ; save it in the sum
        dec     esi                     ; back up one digit
        dec     edi
        loop    L1
        mov     sum[edi],bh             ; save last carry digit
; Display the sum as a string.
        mov     edx,OFFSET sum
        call    WriteString
        call    Crlf
        exit
main ENDP
END main
```

Result of running the program ASCII add is shown below.



```
TITLE Binary to ASCII                (BinToAsc.asm)

; This program converts a 32-bit binary integer to ASCII.

INCLUDE Irvine32.inc

.data
binVal      DWORD 1234ABCDh                     ; sample binary value
buffer      BYTE 32 dup(0),0

.code
main PROC
            mov     eax,binVal                          ; EAX = binary integer
            mov     esi,OFFSET buffer           ; point to the buffer
            call    BinToAsc                            ; do the conversion

            mov     edx,OFFSET buffer           ; display the buffer
            call WriteString                    ; output: 0001 0010 0011 0100 1010 1011 1100 1101

            call Crlf
            exit
main ENDP
```

```
;-------------------------------------------------------
; BinToAsc PROC
; Converts 32-bit binary integer to ASCII binary.
; Receives: EAX = binary integer, ESI points to buffer
; Returns: buffer filled with ASCII binary digits
;-------------------------------------------------------
BinToAsc PROC
        push    ecx
        push    esi
        mov     ecx,32                          ; number of bits in EAX

L1:     shl     eax,1                           ; shift high bit into Carry flag
        mov     BYTE PTR [esi],'0'              ; choose 0 as default digit
        jnc     L2                              ; if no Carry, jump to L2
        mov     BYTE PTR [esi],'1'              ; else move 1 to buffer

L2:     inc     esi                             ; next buffer position
        loop    L1                              ; shift another bit to left
        pop     esi
        pop     ecx
        ret
BinToAsc ENDP
END main
```

Program to translate simple arithmetic into Assembly language

```
TITLE Unsigned Arithmetic Expressions       (Express.asm)

; This program shows how to translate simple
; arithmetic expressions into assembly language.

INCLUDE Irvine32.inc

.data
msg1 BYTE "Unsigned overflow!",0dh,0ah,0
var1 DWORD 3
var2 DWORD 6
var3 DWORD 4
var4 DWORD ?

.code
main PROC

;Divide Overflow example:
        mov ax,1000h
        mov bl,0
        div bl
        jmp quit
```

```
;Example 1: var4 = (var1 + var2) * var3;
Example1:
        mov eax,var1
        add eax,var2
        mul var3                ; EAX * var3
        jc  tooBig          ; overflow?
        mov var4,eax
        jmp Example2

Example2:                                   ; var4 = (var1 * 5) / (var2 - 3);

        mov eax,var1
        mov ebx,5
        mul ebx                 ; EDX:EAX = product
        mov ebx,var2
        sub ebx,3
        div ebx
        mov var4,eax
```

```
Example3:                       ; var4 = (var1 * -5) / (-var2 % var3);
        mov eax,var2            ; begin right side
        neg eax
        cdq                     ; sign-extend dividend
        idiv var3               ; EDX = remainder
        mov ebx,edx             ; EBX = right side

        mov eax,-5              ; begin left side
        imul var1               ; EDX:EAX = left side
        idiv ebx                ; final division
        mov var4,eax            ; quotient

tooBig:
        mov edx,OFFSET msg1
        call WriteString

quit:
        exit
main ENDP
END main
```

Take a look at the rest of the program under the directory Ch7