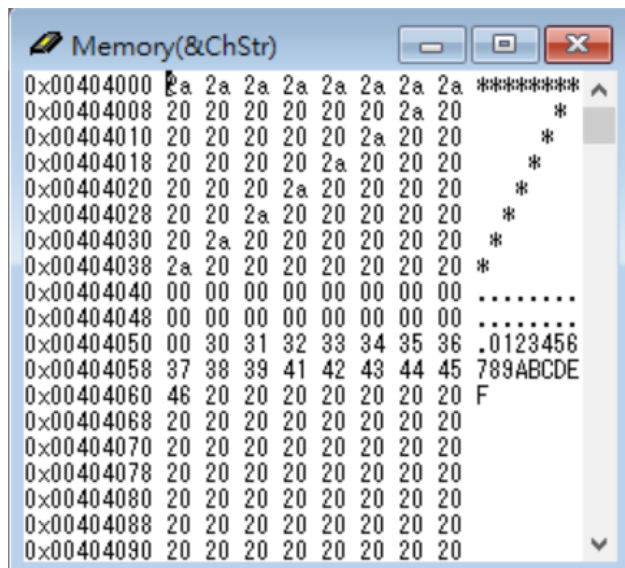


● 程式碼與執行流程與記憶體狀態截圖及說明



➤ 一開始以 2a 表示*，20 表示“ ”。

```
.data
ChStr BYTE 2Ah,2Ah,2Ah,2Ah,2Ah,2Ah,2Ah,2Ah
        BYTE 20h,20h,20h,20h,20h,20h,2Ah,20h
        BYTE 20h,20h,20h,20h,20h,2Ah,20h,20h
        BYTE 20h,20h,20h,20h,2Ah,20h,20h,20h
        BYTE 20h,20h,20h,2Ah,20h,20h,20h,20h
        BYTE 20h,20h,2Ah,20h,20h,20h,20h,20h
        BYTE 20h,2Ah,20h,20h,20h,20h,20h,20h
        BYTE 2Ah,20h,20h,20h,20h,20h,20h,20h
BitStr BYTE 8 DUP(?)

.code
change PROC
    mov bl,10000000b
    mov dl,0
    mov edi,0
L:
    mov al,ChStr[esi]
    cmp al,20h
    jz L1
    add al,7h
    add dl,bl
    shr bl,1
    jmp L2
L1:
    add al,10h
    shr bl,1
L2:
    inc esi
    cmp bl,0
    jnz L4
    mov BitStr[edi],dl
    mov bl,10000000b
    mov dl,0
    inc edi
L4:
    Loop L
    ret
change ENDP
```

- 接著我們進行 call change 環節，裡面有三個 L，依序做的內容如下：

L：

將 ChStr[esi]輸入進 al 裡，

比較 al 的 ASCII 碼跟 20h 的大小

- 如果 al 比較大的話就會先 al=al+07h，將 bl 加到 dl 裡，bl 的 1 右移 1 位再進入 L2
- 如果 al 跟 20h 相等的話，就會直接跳到 L1

L1:

al=al+10h

bl 的 1 右移 1 位

L2:

al 會 move 到 ChStr[esi]裡

接著 esi=esi+1 移到 ChStr 下一個位置。

比較 bl 與 0 的大小

- 如果 bl 不是 0 的話，就會跳到 L4，而 L4 的 Loop L 就會回到原本的 L
- 如果 bl 是 0 的話
 - ◆ 繼續執行下方，將 bl 跟 dl 恢復原本初始值
 - ◆ 將 dl 移至 BitStr[edi]
 - ◆ 接著再回到 L 繼續執行。

main PROC

```
mov esi,0  
mov ecx,64
```

```
CALL change
```

```
mov ecx,8  
mov esi,0
```

L3:

```
movzx eax,BitStr[esi]  
mov ebx,type byte  
call writebinb  
call crlf  
inc esi  
Loop L3  
exit
```

```
main ENDP  
END main
```

1111	1111
0000	0010
0000	0100
0000	1000
0001	0000
0010	0000
0100	0000
1000	0000

- 因為已經事先將值移入 BitStr 了，最後我們再使用 writebinb 跟輸出 type 為 byte 的形式數字，用 loop 迴圈 L3 在 cmd 上做 8 次輸出印出結果即可。

● 完整程式畫面截圖

```
.data
ChStr BYTE 2Ah,2Ah,2Ah,2Ah,2Ah,2Ah,2Ah,2Ah
      BYTE 20h,20h,20h,20h,20h,20h,2Ah,20h
      BYTE 20h,20h,20h,20h,20h,2Ah,20h,20h
      BYTE 20h,20h,20h,20h,2Ah,20h,20h,20h
      BYTE 20h,20h,20h,2Ah,20h,20h,20h,20h
      BYTE 20h,20h,2Ah,20h,20h,20h,20h,20h
      BYTE 20h,2Ah,20h,20h,20h,20h,20h,20h
      BYTE 2Ah,20h,20h,20h,20h,20h,20h,20h
BitStr BYTE 8 DUP(?)
.code
change PROC
    mov bl,10000000b
    mov dl,0
    mov edi,0
L:
    mov al,ChStr[esi]
    cmp al,20h
    jz L1
    add al,7h
    add dl,bl
    shr bl,1
    jmp L2
L1:
    add al,10h
    shr bl,1
L2:
    inc esi
    cmp bl,0
    jnz L4
    mov BitStr[edi],dl
    mov bl,10000000b
    mov dl,0
    inc edi
L4:
    Loop L
    ret
change ENDP
```

```
main PROC
    mov esi,0
    mov ecx,64

    CALL change

    mov ecx,8
    mov esi,0
L3:
    movzx eax,BitStr[esi]
    mov ebx,type byte
    call writebinb
    call crlf
    inc esi
    Loop L3
    exit
main ENDP
END main
```

● 心得

關於作業 2 的在 cmd 印出部分，希望助教能夠給再多一點的提示，不然我們很多都不知道怎麼打。雖然說自己摸索是個重要的一環，但是我們現在組合語言還在學習的階段，第一次的摸索不應該是全部讓我們自己探索。