

Morse Code Translator

姓名：李名智

學號：1102924

一、介紹

由組合語言來製作一個摩斯密碼轉換器，輸入規則為將輸入的文字轉換成大寫後再進行轉換，而因 x86 為 little-endian 故當儲存英文字母 A-Z 時需反過來存，這樣才會有正確的輸出。若輸出沒有定義的符號時則用 "@" 表示。而最後會問是否繼續轉換，若答 Y 則繼續輸入，否則結束程式。

二、程式說明

(1) DATA 介紹

```
.data
msg1 BYTE "Enter the your string: ",0
mystring BYTE BUFMAX+1 DUP(?) ;儲存輸入文字
myupper BYTE BUFMAX+1 DUP(?) ;儲存變大寫的文字
ans BYTE 2 DUP(?) ;最後回答Y/n的文字
msg2 BYTE "Morse Code: ",0
spas BYTE " ",0
msg3 BYTE "Would you like to proceed another translation (y/n)?",0
mymorse DWORD 30 DUP(?) ;暫存摩斯密碼的地方
strSize DWORD ? ;輸入文字的長度
i DWORD 000000000h ;用於位置計算的變數
```

存 A-Z, 0-9 的摩斯密碼

```
;A-Z Letter
letter_table DWORD "_.", "..._", "._-_", "..._", "._", "._.", ".___", "....",
;0-9 Digit
zero BYTE "____",0
one BYTE ".____",0
two BYTE "..____",0
three BYTE "...__",0
four BYTE "...._",0
five BYTE ".....",0
six BYTE ".....",0
seven BYTE "_....",0
eight BYTE "___..",0
nine BYTE "____.",0
```

儲存各種符號的摩斯密碼

```

;Punctuation
Period BYTE  ". _ . _ . _",0
Comma BYTE  " _ _ . _ _",0
QuestionMark BYTE  ". _ _ _ . _",0
ParenthesesL BYTE  " _ . _ _ .",0
ParenthesesR BYTE  " _ . _ _ .",0
Apostrophe  BYTE  ". _ _ _ _",0
Semicolon BYTE  " _ . _ . _",0
Colon BYTE  " _ _ _ . _",0
QuotationMarks BYTE  ". _ . _ .",0
Hyphen BYTE  " _ . . . _",0
FractionBar BYTE  " _ . . _ .",0
DollarSign BYTE  " . . . _ . _",0
Other BYTE  "@",0
Space BYTE  "/",0
;-----

```

(2) 主程式介紹

```

main PROC
beginwhile:
    mov edx,OFFSET msg1 ;輸出enter your string
    call WriteString
    call InputTheString ;輸入文字

    call lowertoCap      ;小寫變大寫的副程式

    call Crlf
    mov edx,OFFSET msg2 ;輸出我的Morse
    call WriteString
    call MorseTran       ;輸出Morse的副程式

    call Crlf
    mov edx,OFFSET msg3 ;是否再一次?
    call WriteString
    mov ecx,2
    mov edx,OFFSET ans
    call ReadString      ;輸入Y/n
    mov al,ans
    cmp al,'Y'           ;若為Y 再進一次迴圈，若否就跳出迴圈並結束程式
    je beginwhile
endwhile:

    exit
main ENDP

```

(3) 副程式介紹

副程式一：InputTheString(用於輸入文字)

```

;-----
InputTheString PROC
    pushad
    mov ecx,BUFMAX
    mov edx,OFFSET mystring ; 將輸入文字存到mystring
    call ReadString         ; 輸入
    mov strSize,eax         ; 儲存文字長度
    popad
    ret
InputTheString ENDP
;-----

```

副程式二：lowertoCap(用於讓輸入文字變成大寫)

```

;-----
lowertoCap PROC
    pushad
    mov esi, OFFSET mystring
    mov edi, OFFSET myupper
    mov ecx, strSize          ; 長度

L1: mov al, [esi]
    cmp al, 0                ; 檢查是否到達結尾
    je do                    ; 如果是，結束循環

    cmp al, 'a'              ; 檢查是否為小寫字母
    jb A1                    ; 如果小於 'a'，跳過轉換
    cmp al, 'z'
    ja A1
    sub al, 32                ; 若a<=[esi]<=z 將小寫字母轉換為大寫字母

A1: mov [edi], al
    inc esi
    inc edi
    loop L1
do: mov BYTE PTR [edi], 0    ;確保輸出以 NULL 結尾
    popad
    ret
lowertoCap ENDP
;-----

```

副程式三：MorseTran(用於輸出 A-Z Morse)

```
MorseTran PROC
    pushad
    mov esi,OFFSET myupper ;用來讀取upper的字串
    mov ecx,strSize
L1: mov al,[esi] ;若[esi]範圍落在A~Z，去找對應的Morse
    cmp al,'A' ;若沒落在此範圍內代表不是letter，就去看是否為數字(Digit)
    jae ckZ ;[esi]>=A
    jnb not_letter ;[esi]<A (不是字母)

ckZ:cmp al,'Z'
    jbe Find ;[esi]<=Z
    jmp not_letter ;[esi]>Z (不是字母)
Find:
    sub al,41h ;看是第幾個字母
    add BYTE PTR i,al
    mov eax,i
    mov edx,4
    mul edx
    mov ebx,letter_table[ebx] ;eax為該字母對應的Morse位置
    mov mymorse,ebx
    mov edx,OFFSET mymorse ;輸出
    call WriteString
    jmp qt

not_letter: ;若不為字母，檢查是否為數字
    cmp al,'0'
    jae ck9
    jmp not_digit
ck9:cmp al,'9'
    jbe is_Digit
    jmp not_digit
is_Digit: ;若是數字，去DIGIT
    call DIGIT
    jmp qt
not_digit: ;若不為數字，去Punctuation(符號)
    call Punctuation

qt: mov i,0
    mov edx,OFFSET spas
    call WriteString
    mov mymorse,0
    inc esi
    loop L1
    popad
    ret
MorseTran ENDP
```

副程式四：DIGIT(用於輸出 0-9 Morse)

```
DIGIT PROC
    pushad
    mov ecx,10          ;迴圈10 0-9
    mov i,30h           ;i=00000030
    mov al,[esi]
L1: cmp al,BYTE PTR i ;找為0-9中的哪個
    je Find0
    add i,1
    loop L1
Find0:
    mov eax,i
    sub eax,30h         ;eax-30=該數字
    cmp eax,0           ;比較0
    je out0              ;若符號跳至out0 (輸出該數字的Morse)
    ja Find1             ;否則繼續往下找
Find1:
    cmp eax,1
    je out1
    ja Find2
```

```
...
out0:
    mov edx,OFFSET zero
    jmp qt
out1:
    mov edx,OFFSET one
    jmp qt
```

```
...
qt: call WriteString
    popad
    ret
DIGIT ENDP
```

副程式五：Punctuation(用於輸出各種符號的 Morse)

```
Punctuation PROC
    mov al,[esi]
    pushad
Find0:
    cmp eax,' '
    je out0
Find1:
    cmp eax,'.'
    je out1
Find2:
    cmp eax','
    je out2
Find3:
    cmp eax,'?'
    je out3
```

```
...
out0:
    mov edx,OFFSET Space
    jmp qt
out1:
    mov edx,OFFSET Period
    jmp qt
out2:
    mov edx,OFFSET Comma
    jmp qt
out3:
    mov edx,OFFSET QuestionMark
    jmp qt
```

...

```
out12:
    mov edx,OFFSET DollarSign
    jmp qt
other1:
    mov edx,OFFSET Other
    jmp qt
qt: call WriteString
    popad
    ret
Punctuation ENDP
```

三、結果呈現

```
Enter the your string: what's your name? 1<2*3.
```

Would you like to proceed another translation (y/n)?n