組合語言期末專題——

Base64 編碼轉換

第九組

U0924028 符絮棻

U0924043 陳皇任

U0924051 魏旭堂

大綱

壹、摘要

為了不讓一些特殊符號影響字串結構，因此我們改以 Base64，轉換資料內容的編碼方式，在本專案中，我們提供了編碼與解碼，讓使用者選擇將資料內容在兩種編碼(ASCII 與 Base64)中進行轉換。

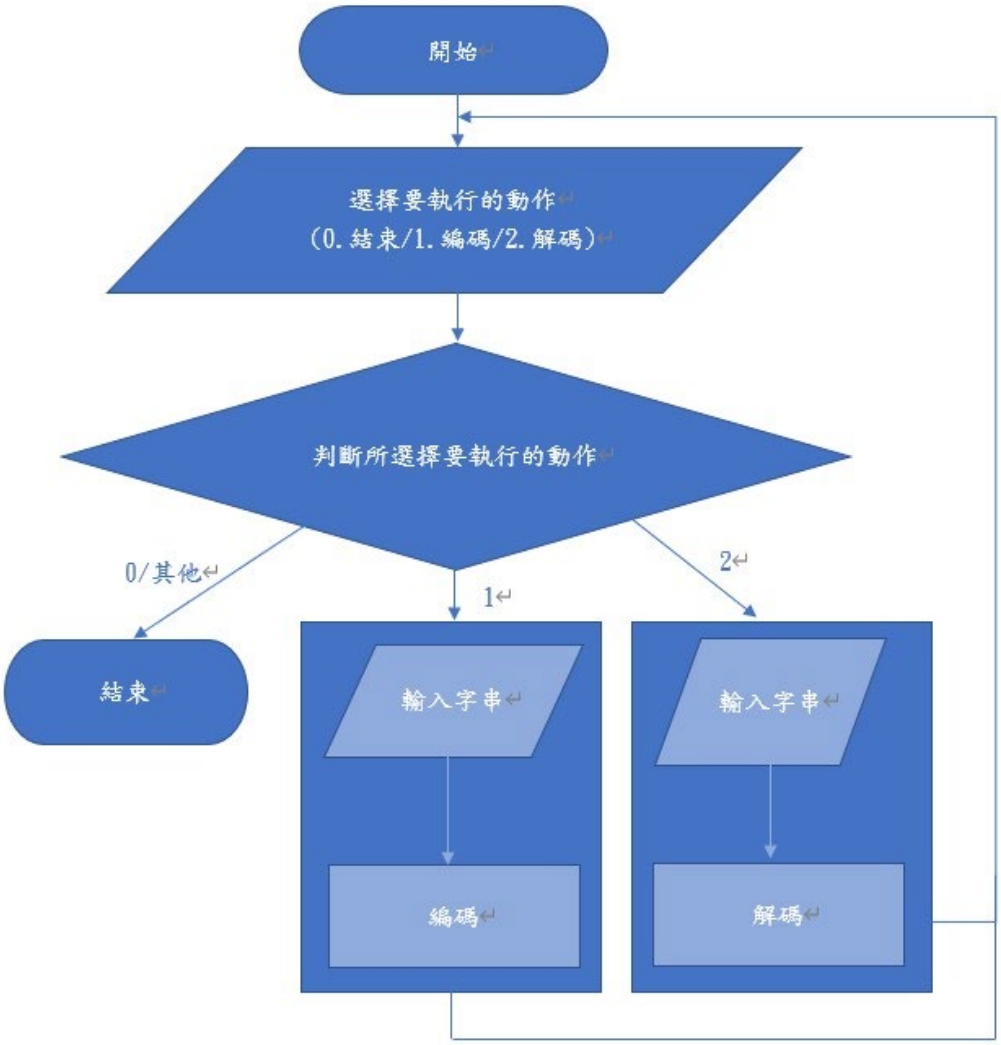
貳、動機與目的

若需將目標程式碼直接以字串的方式在載體程式碼中傳遞，容易因目標程式碼中，含有一些特殊符號，進而導致該字串結構被破壞影響載體程式碼之執行，甚至受到注入攻擊（Injection Attack)。因此為了不讓載體程式碼受到目標程式碼的破壞，我們想將目標程式碼之內容，轉換為一些更加安全的字符。

我們選擇將原始內容轉換為 Base64，將原始內容改以較安全之 64 個字符儲存，同時降低轉換時的複雜度，亦可使該轉換後之內容，可以較高效之位元空間、安全的字符傳遞。

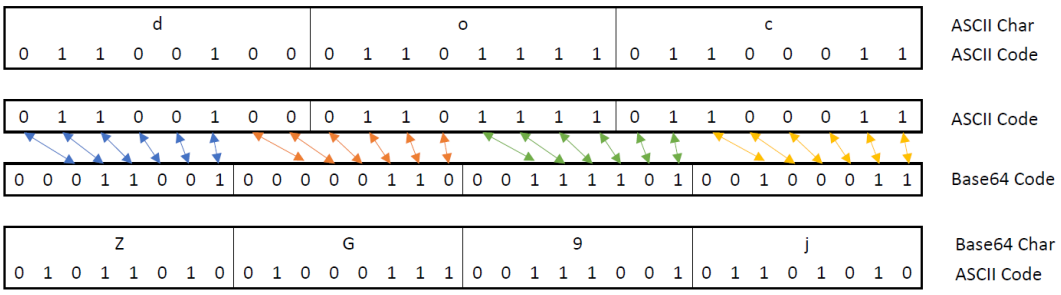
參、演算法

程式一開始會詢問使用者要執行的動作(結束、編碼或解碼)，當使用者輸入 1 時，程式將進行編碼，輸入 2 則將進行解碼，若輸入 0 或其他則將結束程式。進入編/解碼，都會需要使用者輸入字串，才會開始執行，當編/解碼結束後，將再次詢問使用者要執行的動作。流程圖如圖一所示。



圖一： 主程式流程圖

一般而言，輸入的文字是使用 ASCII 編碼。圖二為「以 ASCII 為例之 Base64 編解碼示意圖」，其中 Base64 Code 與 Base64 Char 之關係對照，如圖三「Base64 編碼與字元對照表」所示，若欲將 Base64 Char 呈現，則仍需以該字元之 ASCII Code 作為輸出的字元編碼。



圖二： 以 ASCII 為例之 Base64 編解碼示意圖

十進位	二進位	字元	十進位	二進位	字元	十進位	二進位	字元	十進位	二進位	字元
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/

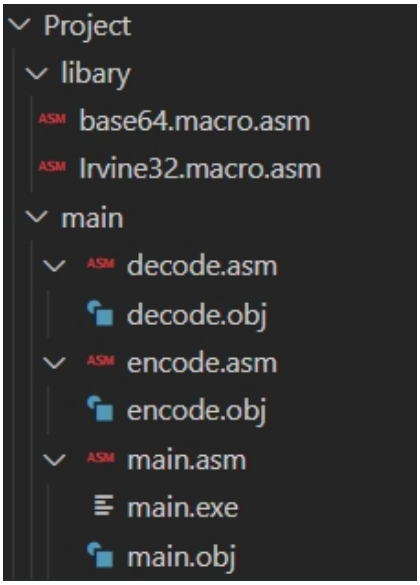
圖三： Base64 編碼與字元對照表

在編碼的時候，會先接收待編碼之字串，而其通常以 ASCII Code 來做儲存。接著將字串中的字元以 3 個為一組取出，每一個字元有 8 個位元，總計取得 24 個位元，再將該 24 個位元切割成四等分，每一份共 6 個位元，該 6 個位元所代表之值，即為 Base64 Code 之值。接著把各 Base64 Code 個別存至一個 Byte，並對照至 Base64 字元對照表，將該 Base64 Code 所對應字元之 ASCII Code 作為輸出的字元編碼。

在做解碼時，會先接收待編碼之字串，其通常以 ASCII Code 來做儲存 Base64 Char，需透過 Base64 字元對照表來取得 Base64 Code。接著將字串中的字元以 4 個為一組取出，每一個字元有 8 個位元，總計取得 32 個位元，再將該 32 個位元切割成四等分，每一份共 8 個位元，將各位元組之末 6 個位元取出並合併，並取得一組 24 位元，即為原資料之字元編碼。

肆、問題與解決

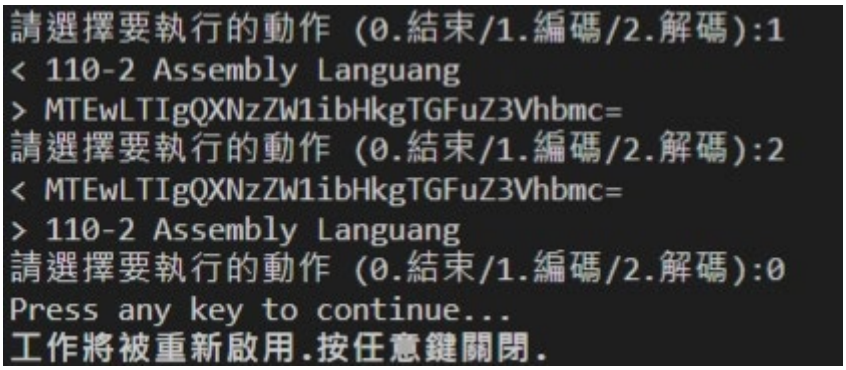
「專案分檔案的結構設計」：在設計時須考慮 macro 與 proc 兩者之間的差異，例如：macro 為置前處理器就會先處理的部分，所以，一個僅由 macro 組成的檔案，無法建立物件檔。若要將副程式建立為一獨立之檔案，亦須考慮將如何將檔案整合，是使用 include 將檔案載入後，再做編譯與連結，或者將檔案編譯為物件檔後，再將所有物件檔連結為執行檔，更甚至將副程式之物件檔連結為 library 檔，以便於使用 includelib 將檔案於連結時期連接使用。專案目錄結構如圖四。



圖四： 專案目錄結構

伍、成果與心得

詢問使用者要執行的動作，並依據使用者的選擇進行編/解碼，或結束本程式，執行結果如圖五。未來若有機會擴充本程式的話，我們希望 (1)在輸入文字的部分提供檔案進行編碼轉換，不侷限於單一字串，能夠讀取檔案內容進行編/解碼(2)提供命令列之帶參數之呼叫，不必開啟程式輸入需要轉換編碼的內容。



圖五： 執行結果

透過期末專題的方式，不僅讓我們實作出自己所訂定的專題內容，一些沒有理解得很透徹的東西，也能在製作的過程中一步步釐清，並將其應用在我們的程式碼中，同時更檢視了自己的一些不足之處。很高興有這個機會讓我們用不同的方式檢視自己這學期的所學。

陸、參考資料

[1] https://en.wikipedia.org/wiki/Base64
[2] https://zh.wikipedia.org/wiki/Base64
[3] https://www.base64encode.org/
[4] https://www.base64decode.org/

柒、分工表

組員	檔案
U0924028 符絮棻	main.asm、encode.asm、decode.asm
U0924043 陳皇任	Irvine32.macro.asm、base64.macro.asm
U0924051 魏旭堂	base64.macro.asm