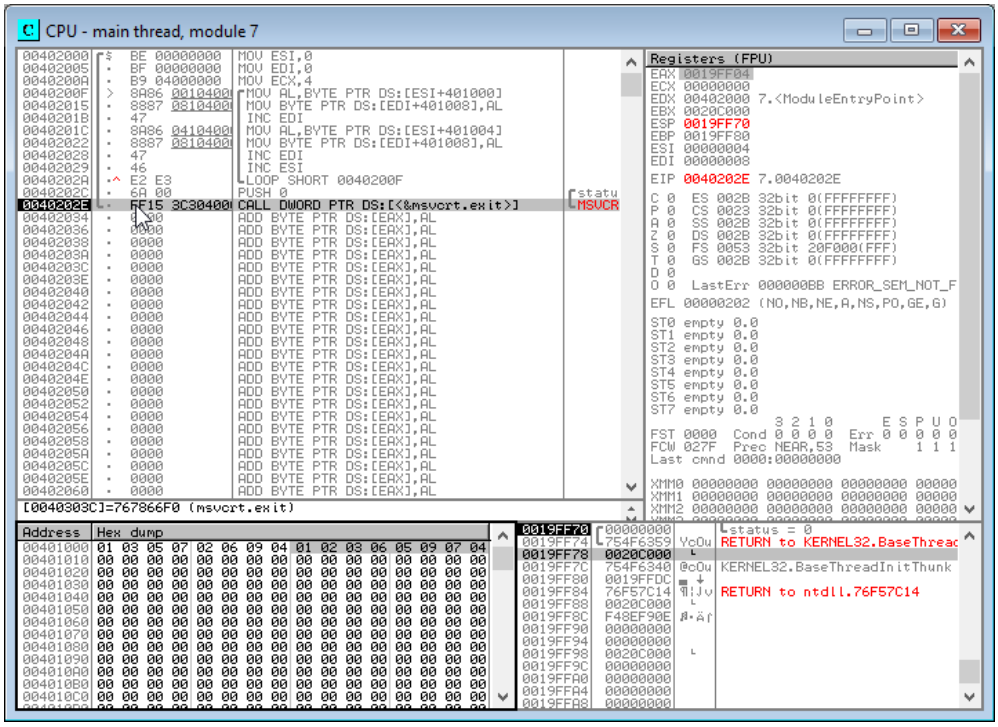
Neacșu-Miclea Liviu-Ștefan
Grupa 215

7. Se dau două șiruri de octeți *S1* și *S2* de aceeași lungime. Să se obțină șirul *D* prin intercalarea elementelor celor două șiruri.

Data	Code	Iter.1	Iter.2	Iter.3	Iter.4

`S1 DB 1, 3, 5, 7 ; source 1`
`S2 DB 2, 6, 9, 4 ; source 2`
`S_Len EQU $-S2`
`; S is the same for S1 and S2`
 	
`; destination`
`D RESB 2 * S_Len`
	`xor ESI, ESI ; ESI = 0`
	`xor EDI, EDI ; EDI = 0`
	`mov ECX, S_Len ; ECX = 4, prepare loop`
	`.iterate:`				

	`mov AL, [S1+ESI] ; AL = S1[ESI]`	AL=1	AL=3	AL=5	AL=7
	`mov [D+EDI], AL ; D[EDI] = AL`				
	`inc EDI ; EDI++`	EDI=1	EDI=3	EDI=5	EDI=7
	`mov AL, [S2+ESI] ; AL = S2[ESI]`	AL=2	AL=6	AL=9	AL=4
	`mov [D+EDI], AL ; D[EDI] = AL`				
	`inc EDI ; EDI ++`	EDI=2	EDI=4	EDI=6	EDI=8
	`inc ESI ; ESI ++`	ESI=1	ESI=2	ESI=3	ESI=4
	`loop .iterate ; loop till ECX=0`	ECX=3	ECX=2	ECX=1	ECX=0



Alternativă cu șiruri

```
cld
mov ESI, S1
mov EBX, S_Len-1
mov EDI, D
mov ECX, S_Len
.iterate:
    movsb
    add ESI, EBX
    movsb
    stc
    sbb ESI, EBX
loop .iterate
```

Neacșu-Miclea Liviu-Ștefan
Grupa 215

8. Se dă un șir de caractere *S*. Să se construiască șirul *D* care să conțină toate literele mari din șirul *S*.

Data	Code	I1	I2	I3	I4	I5	I6	I7	I8	Reg
S DB "aAbB2%xM" *D* RESB ($-*S*)	xor ESI, ESI ; ESI = 0 xor EDI, EDI ; EDI = 0 mov ECX, (*D*-*S*) ; ECX = len(S); prepare loop .iterate:									
	mov AL, [*S*+ESI] ; AL = S[ESI]	61h	41h	62h	42h	32h	25h	78h	4Dh	*AL*
	inc ESI ; ESI++	1	2	3	4	5	6	7	8	*ESI*
	cmp AL, 'A' ; compare S[ESI] to 'A'	>	=	>	>	<	<	>	>	
	jb .next ; if S[ESI] < 'A', skip	No	**No**	No	**No**	Yes	Yes	No	**No**	
	cmp AL, 'Z' ; compare S[ESI] to 'Z'	>	<	>	<	<	<	>	<	
	ja .next ; if S[ESI] > 'Z', skip	Yes	**No**	Yes	**No**	No	No	Yes	**No**	
	; if we are here, S[ESI] is letter ; so append it to D(estination)	×	↓	×	↓	×	×	×	↓	
	mov [*D*+EDI], AL		*'A'*		*'B'*				*'M'*	
	inc EDI ; prepare next dest. index	0	1	1	2	2	2	2	3	*EDI*
	.next:									
	loop .iterate ; loop until ECX == 0	7	6	5	4	3	2	1	0	*ECX*
	; end the new string with '\0' character									
	mov [*D*+EDI], BYTE 0