

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота 4

Архітектура комп'ютера

*“Дослідження роботи стеку МПС архітектури IA-32 (x86) у
Real Adress Mode”*

Виконали:
Студенти групи IT-01

Тимошенко Олексій
Тонкий Михайло

Перевірів:

Бердник Ю.М.

Київ 2021

Тема: Дослідження роботи стеку МПС архітектури IA-32 (x86) у Real Address Mode.

Мета: отримати навички роботи зі стеком.

Github: <https://github.com/OlexiiT/assembler/tree/main/L4>
<https://github.com/Mikuyoki/Asembler/tree/main/L4>

Хід роботи:

Під час виконання роботи було створено масив arr_stack 16x16, заповнений даними типу слово

```
arr_stack dw 0BE0Dh, 063F2h, 09568h, 01091h, 04072h, 0D3FBh, 00242h, 0BE89h, 0364Bh, 0B486h, 02D48h, 0DEF9h, 0E89Eh, 047E3h, 05C3Ch, 0C4DAh
dw 04CD7h, 0A437h, 009F6h, 0BEE4h, 0630Eh, 0364Bh, 033B8h, 05201h, 0E304h, 03E5Ch, 0E0FDh, 085CAh, 0DB12h, 066D5h, 00212h, 0A26Fh
dw 067D8h, 0DD41h, 06A32h, 087D8h, 0E66Ah, 038D9h, 046E8h, 055ECh, 02002h, 0D0AFh, 06482h, 046DBh, 049B5h, 08CB9h, 090DAh, 0FD67h
dw 0941Eh, 0A2D7h, 0752Ah, 0109Ch, 06DD2h, 04475h, 05CAAh, 0745Ah, 0CBFBh, 05B1Ah, 03AF5h, 0F94Ch, 084F8h, 047F3h, 01F43h, 0518Fh
dw 0BEC0h, 0C7E2h, 02A29h, 0F49Ch, 09965h, 05668h, 076E1h, 00120h, 024B1h, 02CE3h, 0943Ah, 05587h, 052AFh, 09A1Fh, 0F852h, 08FFBh
dw 04393h, 01C4Eh, 01CBFh, 00F80h, 04C69h, 04B58h, 04B4Fh, 07030h, 0983Ch, 06335h, 069FCh, 099F2h, 0DF40h, 0DB83h, 0A512h, 0BCFCh
dw 085A4h, 016BFh, 09CB0h, 0CF97h, 07F3Bh, 05169h, 05F88h, 0A29Bh, 08E84h, 06058h, 0FA49h, 0B2F5h, 05705h, 02191h, 06F24h, 09659h
dw 0DA28h, 0921Bh, 0C434h, 0A53Eh, 0637Dh, 045A1h, 040E4h, 05836h, 07B8Bh, 023B7h, 09B3Ch, 06515h, 08F9Eh, 05AC9h, 02C8Eh, 085B8h
```

Далі ми створили процедуру copy, що копіює масив:

```
PROC copy ;Процедура копіювання масиву
mov cx, 128
w1:
    mov ax, [ds:si] ;Отримуємо значення з масиву
    mov [ds:[si+256]], ax ;Та копіюємо його
    add si, 2 ;Збільшуємо лічильник на 2
    loop w1
ret
ENDP copy ;Кінець процедури
```

Та копіюємо його 7 разів, відповідно до варіанту. Далі необхідно було розрахувати розмір стеку та перенести до нього значення масиву. Оскільки масив складається з 128 слів, розміром 2 байти кожен, то загальний розмір стеку, має бути що найменше 256 байтів. Перенесення масиву виконується за допомогою циклу:

```
;-----Розміщення масиву у стеку-----
xor si, si ;Обнулення регістру
mov cx, 128 ;Задаємо кількість повторів циклу
w2:
    mov ax, [ds:si] ;Отримуємо значення з масиву
    push ax ;Та записуємо його у стек
    add si, 2 ;Збільшуємо лічильник на 2
    loop w2
```

Наступним завданням було заповнити сьомий рядок масиву у стеку числами дат народження (16.07.2003, 19.09.2003), користуючись базовою адресацією.

У нашій програмі стек має розмір у 2048 байтів, або ж 800h у шістнадцятковій системі. Розмір масиву - 256 байтів, або 100h. Отже масив починається у стеку за адресою 700h. Щоб отримати сьомий рядок необхідно додати ще шість рядків, або 60h, і ми отримуємо адресі початку рядка - 760h, і відповідно його кінець у 76Fh.

ss:0700	BF	85	8E	2C	C9	5A	9E	8F	15	65	3C	9B	B7	23	8B	7B
ss:0710	36	58	E4	40	A1	45	7D	63	3E	A5	34	C4	1B	92	28	DA
ss:0720	59	96	24	6F	91	21	05	57	F5	B2	49	FA	58	60	84	8E
ss:0730	9B	A2	88	5F	69	51	3B	7F	97	CF	B0	9C	BF	16	A4	85
ss:0740	FC	BC	12	A5	83	DB	40	DF	F2	99	FC	69	35	63	3C	98
ss:0750	30	70	4F	4B	58	4B	69	4C	80	0F	BF	1C	4E	1C	93	43
ss:0760	16	07	20	03	19	09	20	03	16	07	20	03	19	09	20	03
ss:0770	20	01	E1	76	68	56	65	99	9C	F4	29	2A	E2	C7	C0	BE
ss:0780	8F	51	43	1F	F3	47	F8	84	4C	F9	F5	3A	1A	5B	FB	CB
ss:0790	5A	74	AA	5C	75	44	D2	6D	9C	10	2A	75	D7	A2	1E	94
ss:07A0	67	FD	DA	90	B9	8C	B5	49	DB	46	82	64	AF	D0	02	20
ss:07B0	EC	55	E8	46	D9	38	6A	E6	D8	87	32	6A	41	DD	D8	67
ss:07C0	6F	A2	12	02	D5	66	12	DB	CA	85	FD	E0	5C	3E	04	E3
ss:07D0	01	52	B8	33	4B	36	0E	63	E4	BE	F6	09	37	A4	D7	4C
ss:07E0	DA	C4	3C	5C	E3	47	9E	E8	F9	DE	48	2D	86	B4	4B	36
ss:07F0	89	BE	42	02	FB	D3	72	40	91	10	68	95	F2	63	0D	BE

Також було необхідно зарезервувати ділянку пам'яті розміром у 2 параграфи, занести до неї масив зі стеку зафіксувати це та звільнити її.

```
;-----Резервування нової ділянки пам'яті-----  
;-----Функція виділення пам'яті-----  
    mov ah, 048h                ;Ознака переривання  
    mov bx, 02h                ;Розмір нової ділянки  
    int 21h  
  
;-----Перенесення масиву до нової ділянки-----  
  
    mov es, ax  
    mov cx, 128  
    xor si, si  
w3:  
    pop dx  
    mov [es:si], dx  
    add si, 2  
    loop w3  
  
;-----Функція звільнення пам'яті-----  
  
    mov ah, 049h  
    int 21h
```

```
DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[ ]=Dump
es:0000 BF 85 8E 2C C9 5A 9E 8F 15 65 3C 9B B7 23 8B 7B 7 àÀ, ZRÀSe<Ç#i{
es:0010 36 58 E4 40 A1 45 7D 63 3E A5 34 C4 1B 92 28 DA 6XS0iE)c>N4-+ff(
es:0020 59 96 24 6F 91 21 05 57 F5 B2 49 FA 58 60 84 8E Yû$0æ!4WJ I-X`aA
es:0030 9B A2 88 5F 69 51 3B 7F 97 CF B0 9C BF 16 A4 85 Ç6ê_iQ:0ù= f7-ñà
es:0040 FC BC 12 A5 83 DB 40 DF F2 99 FC 69 35 63 3C 98 "J+ñA[e]z0"i5c<ij
es:0050 30 70 4F 4B 58 4B 69 4C 80 0F BF 1C 4E 1C 93 43 0p0KXKiLC*7-N-ôC
es:0060 16 07 20 03 19 09 20 03 16 07 20 03 19 09 20 03 -·♥lo♥=-·♥lo♥
es:0070 20 01 E1 76 68 56 65 99 9C F4 29 2A E2 C7 C0 BE @BvhUe0f()*)Γ||L
es:0080 8F 51 43 1F F3 47 F8 84 4C F9 F5 3A 1A 5B FB CB ÅQCv≤G°aL·J:→[J||
es:0090 5A 74 AA 5C 75 44 D2 6D 9C 10 2A 75 D7 A2 1E 94 Zt-\\uD7mf>*u||6Δ0
es:00A0 67 FD DA 90 B9 8C B5 49 DB 46 82 64 AF D0 02 20 g² rÉ||i-|Féd»||0
es:00B0 EC 55 E8 46 D9 38 6A E6 D8 87 32 6A 41 DD D8 67 0U0F-8jμ+g2jA|†y
es:00C0 6F A2 12 02 D5 66 12 DB CA 85 FD E0 5C 3E 04 E3 0ó+0f†||à²α\\>+π
es:00D0 01 52 B8 33 4B 36 0E 63 E4 BE F6 09 37 A4 D7 4C @R73K6/cΣJ÷o7ñ||L
es:00E0 DA C4 3C 5C E3 47 9E E8 F9 DE 48 2D 86 B4 4B 36 Γ<\\nGR0·H-ã|K6
es:00F0 89 BE 42 02 FB D3 72 40 91 10 68 95 F2 63 0D BE èJBeJ"reæ>hò≥cJd
es:0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
es:0110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
es:0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
es:0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
es:0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

Висновки:

У ході даної роботи ми навчилися використовувати механізми адресації у архітектурі IA-32 у real adress mode, та здобули навички роботи зі стеком.