Arhitectura sistemelor de calcul

Curs 7 - aritmetica de pointeri

Spre deosebire de limbajele de nivel înalt care accesează memoria prin numele variabilelor, în asamblare memoria poate fi accesată prin intermediul formulei de calcul al offsetului unui operand și pe baza aritmeticii de pointeri.

În cadrul sistemului de adresare se efectuează operații cu adrese, adică expresiile aritmetice o să aibă ca operanzi adrese, aceste operații utilizând aritmetica de pointeri.

Operațiile aritmetice permise cu pointeri sunt:

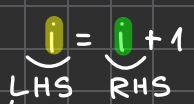
Scăderi de adrese (util pentru size of array)

Adunări și scăderi de constante la o/ dintr-o adresă (necesare și utile pentru accesarea elementelor dintr-un tablou)

Adunarea și înmulțirea adreselor sau înmulțirea unei adrese cu constante nu sunt permise deoarece în maj. cazurilor rezultatul nu se află în memorie.

Exerciții:

expresie_calcul_de_adresă = expresie



LHS (i) = Left hand side i = adresa lui i

RHS (i) = Right hand side i = conținutul de la adresa lui i

offset = [bază] + [index scală] + [constantă]
(SIB) (+ deplasament ± imediat)

Numele unei variabile este offsetul său raportat la segmentul în care apare definiția sa. Offseturile variabilelor sunt constante.

```
mov eax, [v]                          ✓  [constantă]
mov eax, [ebx + ecx·2 - 7]            ✓  [bază + index·scală + constantă]   se poate face +(-7)
add edx, [ebx + ecx·2 + v - 7]        ✓  [bază + index·scală + constantă]
mov ebx, [ebx + ecx·2 - v - 7]        syntax error   s-a folosit - în loc de + la deplasament
adc ecx, [ebx + ecx·2 + a + b - 7]    syntax error   a+b nu avem voie să adunăm două adrese
sub [ebx + ecx·2 + a - b - 7], eax    ✓  a-b este o operație corectă
mov ax, ebx                           syntax error   sizeof destination != sizeof source
mov ebx, ch                           syntax error   sizeof destination != sizeof source
mov eax, ebx                          ✓  sizeof destination = sizeof source
```

[index · scală]

mov eax, [ebx] ✓

mov ax, [ebx] ✓ ax ← 2 octeți de la DS:[ebx] deoarece operandul sursă nu are tip asociat, operandul destinație este cel care stabilește tipul de dată al transferului

mov edx, [eax + ebx] ✓

mov edx, eax + ebx syntax error între doi registrii se folosește add, + poate face calculul doar cu valori constante de la asamblare singura excepție o reprezintă formula offset

mov edx, eax + v syntax error eax nu e constantă —"—

mov edx, [ebx + eax] ✓ nu știm care e baza și indexul

mov edx, [esp + ecx] ✓ aici esp trb să fie baza

mov edx, [ecx + esp] ✓ —"—

bază + scală · index

mov edx, [esp + 2 · ecx] ✓

mov edx [ecx + 2 · esp] syntax error esp nu poate să fie index

mov dh, [edx + ecx · 4 + 3] ✓ transferă primul octet

mov dx, [edx + ecx · 4 + 3] ✓ transferă primii doi octeți

mov eax, [eax · 3] ✓ mov eax, [eax + eax · 2]

mov eax, [ebx · 9 + 12] ✓ mov eax, [ebx + ebx · 8 + 12]

mov eax, [esp · 5] syntax error esp-ul nu poate fi index

Memoria se poate accesa în două moduri:

- utilizând numele de variabile mov eax, [v]
- utilizând formula offsetului mov eax, [ebx + ecx · 4 + 5]

Ollydbg: mov eax, [v] traduce ca mov eax, dword ptr [DS:2]

mov eax, var ✓ se încarcă offset-ul variabilei var pentru 32 biți

mov eax, [var] ✓ se încarcă 4 octeți de la conținutul care are adresa de start var

mov ax, var ✓ se încarcă 2 octeți din adresa care este pentru pe 4 octeți dar avem warning 16-bit relocation of 32 bits value

mov ax, [var] ✓ se încarcă 2 octeți de la conținutul care are adresa de start var

mov ah, [var] ✓ se încarcă 1 octet de la conținutul care are

adresa de start var

mov ah, var syntax error nu există offseturi pe 8 biți

var db 17, 18, 19, 29, 2Ah, -3

mov [var], eax ✓ se suprascriu primii 4 octeți de la var cu conținutul de la eax

A db 17

B db 18

C db 21

D db 23

mov eax, [A] ✓ 23 21 19 17

mov eax, [B-1] ✓ 23 21 19 17

mov eax, [C-2] ✓ 23 21 19 17

mov eax, [D-3] ✓ 23 21 19 17

offset_16 = [baza (BX, BP)] + [index (SI, DI)] + [constantă]

mov ah, [bx] ✓ ah ← 1 octet de la DS:[BX]

mov ax, [bx] ✓ ax ← 2 octeți de la DS:[BX]

mov eax, [bx] ✓ eax ← 4 octeți de la DS:[BX]

mov ah, [bh] syntax error deoarece bh este pe 8 biți