

Aritmetica de pointeri

In cadrul sistemului de adresare se efectueaza operatii cu adrese (pointeri). Care sunt operatiile ARITMETICE cu pointeri permise **IN INFORMATICA** ?...

Raspuns: Orice operatie care are sens... aceasta insemnand orice operatie ce exprima ca rezultat o localizare in memorie corecta si utila ca informatie pt programator.

Aritmetica de pointeri/adrese (pointer arithmetic – operatii cu pointeri) = utilizarea de expresii aritmetice, care au ca operanzi adrese !

- adunari si scaderi de adrese ?
 - Adunare de adrese = ??? CE reprezinta ? NIMIC !!!!!
 - Scadere de adrese = ??? CE reprezinta ? $q-p$ = nr **octeti** dintre cele doua adrese de memorie (niciodata nu depasim dim memoriei ; valoarea obtinuta este o CONSTANTA NUMERICA !!!!)
- adunari si scaderi de constante la o/dintr-o adresa = necesare si utile pt accesarea elementelor dintr-un array
- inmultirea a doua adrese ?- nepermisa (in majoritatea cazurilor valoarea obtinuta este dincolo de limita maxima a memoriei posibile a fi accesata).
- Inmultirea cu o constanta (in majoritatea cazurilor valoarea obtinuta este dincolo de limita maxima a memoriei posibile a fi accesata). In plus, CE reprezinta valoarea obtinuta ?... Nimic util !!
- Impartire ?... No way !
- Singura exceptie de la regulile aritmeticii de pointeri o constituie formula de calcul a offsetului unui operand unde sunt permise adunari de valori de registri... In rest nu exista exceptii (de exemplu la nivel de adrese de variabile cu care se opereaza intr-un program).

$a[7] = *(a+7) = *(7+a) = 7[a]$ - atat in C cat si in asamblare !

Pointer arithmetic...? DOAR 3 operatii sunt permise cu POINTERI:

1). Scaderea a doua adrese

Adresa – adresa = ok (q-p = scadere de 2 pointeri = sizeof(array) sau nr de elemente (in C)/octeti (asamblare) dintre doua adrese de memorie)

2) Adunarea unei constante numerice la o adresa

Adresa + constanta numerica (identificarea unui element prin indexare – a[7]) , q+9

3). Scaderea unei constante numerice dintr-o adresa

Adresa – constanta numerica - a[-4] , p-7

- scaderea a 2 pointeri – valoare SCALARA !!! (valoare numerica constanta imediata)
- adunarea unei constante la un pointer → POINTER !
- scaderea unei constante dintr-un pointer → POINTER !

(ultimele doua utile pt referirea de elemente dintr-un array/zona de memorie)

ADUNAREA A DOI POINTERI NU ESTE PERMISA !!!!!

p+q = ??? (allowed in NASM...sometimes...!!!!!!) – dar nu inseamna ADUNARE DE POINTERI in cele din urma asa cum vom vedea...

V db 17

add edx, [EBX+ECX*2 + v -7] – OK !!!!

mov ebx, [EBX+ECX*2 - v-7] – Syntax error !!!! invalid effective address – impossible segment base multiplier

adc ecx, [EBX+ECX*2 + a+b-7] – Syntax error din cauza “a+b”; invalid effective address – impossible segment base multiplier

sub [EBX+ECX*2 + a-b-7], eax – ok, pt ca a-b este o operatie corecta cu pointeri !!!

[EBX+ECX*2 + v -7] – ok
SIB depl. const.

[EBX+ECX*2 + a-b-7]
SIB const.

mov eax, [EBX+ECX*2+(-7)] – ok.

L-value; R-value. Valoare stanga vs. valoare dreapta a unei atribuirii.

Atribuire: **i:=i+1** LHS vs. RHS

Adresa lui I \leftarrow valoarea lui I + 1

LHS(i) = adresa lui I := RHS(i) = (continutul de la adresa I) + 1

LHS (valoarea stanga a unei atribuirii L-value = adresa) := RHS (valoarea dreapta a unei atribuirii = R-Value = CONTINUT !!)

Symbol := expression_value

Identificator := expresie

Expresie_calcul_de_adresa := expresie

Dereferentierea (extragerea valorii de la o adresa) este implicita in 99% din limbaje. Exemplu exceptie – limbajul BLISS – unde dereferentierea trebuie mentionata explicit intotdeauna; $i \leftarrow *i+1$
(+ unele situatii in Algol 68)

Symbol := expression_value (99% of the cases...)

Address_computation_Expression := expression_value

In C++ $f(a+3, b-2, 2) = x+y+z$

Int& f(x,i) {...return v[i];} – Functia f returneaza o LHS (valoarea stanga)
F(a,7) = 79; inseamna ca v[7]=79 !!!

Int& j = i; // j devine ALIAS pt i

(a+2?b:c) = x+y+z ; - correct

(a+2?1:c) = x+y+z; - syntax error !!! 1:=n !!!!

Segment data

A db 17,-2

B dw -20345, "x"

Start2:

C dd 12345678h

Segment code

.....

Jmp start2 ; exemplu de salt intr-un data segment, deci pot exista etichete de cod
; intr-un segment de date !

.....
