

Aritmética de apunadores

martes, 12 de octubre de 2021 07:13 a. m.

Adición de un apunador un número entero (El apunador se moverá $n * \text{sizeof}(\text{tipo de dato})$ posiciones de memoria hacia adelante).

```
int var=45; //&var=0x405
int* ptrVar=&var; // ptr=0x405;
ptrVar+=n; //ptr=0x405+n*4
```

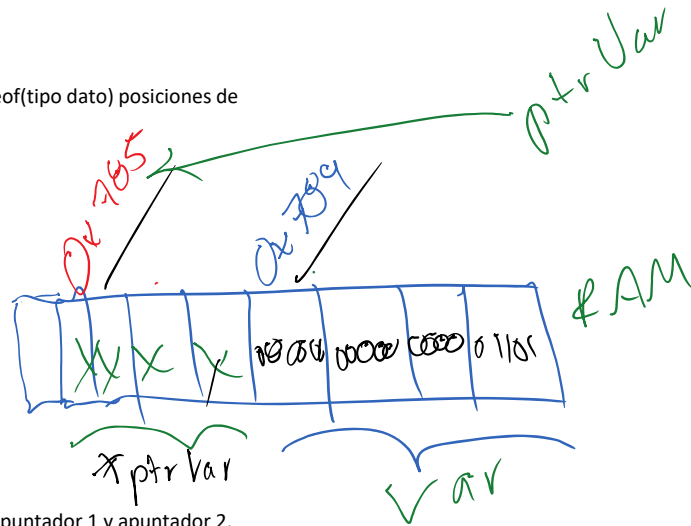
```
char letra='u';
char* ptrLetra=&letra;
ptrLetra+=n; //ptr=0x405+n*1
```

//Si es double, el incremento será de $8 * n$ posiciones.

Sustracción de un apunador y un número entero (se hará un decremento en $n * \text{sizeof}(\text{tipo dato})$ posiciones de memoria);

```
int var=13; //&var=0x789
int* ptrVar=&var; // ptr=0x789;
//ptrVar-=n; //ptr=0x789-n*4
(*ptr)-=n; //decremento pero del valor al cual se apunta----> var=12
*ptr-=n; //ptr=0x785 ----> X basura;

char letra='u';
char* ptrLetra=&letra;
ptrLetra-=n; //ptr=0x405-n
```



Resta de dos apunadores

int valorDistancia=puntero1-puntero2; //Es la distancia que hay que entre el apunador 1 y apunador 2.

Nota: Si queremos modificar la dirección a la cual apunta un apunador, a no se requiere el operador '*'

Otras operaciones validas que se tienen al manejar apunadores son:

- Comparación entre dos o más apunadores del mismo tipo de dato.
 $\text{ptr1} == \text{ptr2};$ $\text{ptr1} < \text{ptr2};$ $\text{ptr2} != \text{ptr1}$
- Asignación a cero
 $\text{int}^* \text{apuntador} = \text{NULL};$ $\text{int}^* \text{apuntador} = 0;$ $\text{void}^* \text{ptr} = \text{NULL}$
- Comparación contra 0
 $\text{ptr} != \text{NULL};$ $\text{ptr} == 0;$
- Corrimientos en memoria

	0x200	0x201	0x202	0x203	0x204	0x205
$\text{char}^* \text{puntero} = 0x200; \text{puntero}++;$
$\text{int}^* \text{puntero} = 0x200; \text{puntero}++;$
$\text{short}^* \text{puntero} = 0x200; \text{puntero}++;$

Que no se debe hacer con apunadores

No deben sumar apunadores:

$\text{ptr1} + \text{ptr2};$

No se pueden multiplicar/dividir/sumar/restar apunadores con valores float/double

$\text{float ptr} = \text{ptr1} / 9.5;$

Realizar asignaciones distintas al tipo de dato del apunador(solamente es posible con un cast)

$\text{int}^* \text{ptr} = '4';$

En ciertos compiladores no se pueden inicializar apunadores en valores fijos:

```
char* ptrChar=0x21;
```

Ejemplo:

```
#include<stdio.h>
void main(){
    int *ptr=0; //declaración apuntador a NULL
    int var=255;
    ptr=&var;
    printf("La dirección a la cual apunta mi apuntador es %d\n", ptr);
    if(ptr==NULL){
        puts("El apuntador no es valido");
        ptr=&var;
        puts("Se ha corregido la dirección de memoria del apuntador");
    }else if(ptr>0){
        ptr+=4; //ptr se mueve a la dirección de memoria anterior
        printf("La nueva dirección de mi apuntador es %d\n", ptr);
    }
}
```