Ponteiros

Ponteiros são variáveis que guardam posições de memória. Toda variável possui um espaço de memória onde seu conteúdo é armazenado ao declarar uma variável, o SO mapeia está variável para um endereço de memória reservado ao programa isto se chama alocação de memória.

Memória fisíca do computador

x> alocação de memória da variável

Espaço do programa

|  |
| --- |
|  |
|  |
|  |
|  |
| 10 |
|  |
|  |
|  |
|  |
|  |
|  |

Main(){

Int x=10;

}

Declaração de um ponteiro

Tipo \*nome;

Ex:

Int \*p;

Char \*l;

Operadores & \*

& > Acessam a posição de memória de uma variável.

Ex:

Int x=10;

Printf(“valor de x: %i”,x)

Printf (“\ posição de memória de x: %i”, &x);

|  |
| --- |
| .  .  . |
|  |
| x10- |
| .  . |
|  |
| P |
|  |
| .  .  . |

Como ponsteiros armazenam posições de memória, podemos fazer:

Int \*p;

P= &x;

Dizemos que p está apontando para x.

\*> Acessa o conteúdo da variável apontada por um ponteiro.

Ex:  
 print(“valor de x: %i “,\*p);

Assim acessamos o conteúdo de x através do ponteiro p.

|  |  |
| --- | --- |
| Int A, \*B;  A=0;  B=&A;  A++;  (\*B)++;  Printf(“%i”,A);  Printf(“\n%i”,\*B); | Int A,B;  A=0;  B=A;  A++;  B++;  Printf(“%i”,A);  Printf(“\n%i”,B); |

Int \*P,A,B;

A=10;

P=&A;

B=\*=\*p+a;

(\*p)++;

A= A +3 \*(\*P);

Int \*A, \*B;

Int X,Y,Z;

X=0;

A=&X;

B=A;

X++;

Y= \*B +\*A;

B=&Z;

\*B= \*A-Y\*2;

\*A+=\*B;

A=&Y;

\*A+=\*B;

X=2\*(\*A)-(\*B+Z);

\*B-=Z-3\*(\*A) +Z;

Printf(“%i %i %i”,X,Y,Z);