

Biến là gì? Vì sao cần biến khi lập trình

Biến là 1 nh ngh a giúp chúng ta truy xu t vào vùng nh thông qua 1 tên, tên ó c g i là bi n

Vì sao dùng biến trong C:

Trong l p trình c n có các giá tr input và output c a bài toán và ngoài ra c n các thành ph n l u tr tính toán trong thu t toán.

l u tr các thành ph n u vào ho c u ra, ho c các thành ph n trung gian tính toán, áp d ng trong các thu t toán thì c n dùng t i bi n, vì c khai báo bi n b n ch t là vì c yêu c u h th ng c p phát cho chúng ta 1 vùng nh l u tr các giá tr u vào và u ra, trung gian c a bi n.

Các kiểu dữ liệu dữ liệu cơ bản trong C:

5 ki u c b n: int, float, double, char, void; trong C

Ki u int là s nguyên 0,1,2,3,4, -5,-6

Tu vào h th ng c p phát ki u d li u này là 2byte hay là 4 byte

Float dùng l u tr s th c 0.124, 0.5235, 4434.5253

Th ng c c p phát là 4byte ng sau s th p phân là 6 giá tr 0.124324

Ki u double th ng l u tr s th c và c c p phát 8 bytes và chính xác ng sau d u ch m th p phân là 10 giá tr .

Ki u char là ki u d li u dùng l u tr ký t và ki u char th ng c c p phát 1 byte

Ki u void là ki u d li u ko có ki u d li u nào c , khi khai báo void thì h th ng ko c p phát b nh

Các từ khoá dẫn xuất: G m 3 t khoá

Ki u **unsigned** áp d ng cho ki u int

Khi khai báo **unsigned int** thì s c c p phát th ng là 2 byte or 4 byte tu vào h th ng và các giá tr này c l u tr là s nguyên d ng

Ki u **short int** là ki u d li u mà khi ta khai báo thì h th ng s c c p phát **1 n a vùng nh** so v i ki u int, tu vào h th ng mà c c p phát 1 byte hay là 2 byte

T khoá **long** có th áp d ng cho 2 ki u int ho c double, v i long int thì ta s c g p **2 l n vùng nh** cho ki u int or double

Store class

Các l p l u tr c a l b i n

G m 2 ph n:

- ✓ Timeline c a l b i n.
- ✓ Scope c a l b i n.

K t h p 2 cái trên c ng v i cái này là nh ngh a c a l b i n

- ✓ Memory c a l b i n.

Ph n 4 là phân bi t s khác nhau gi a bi n toàn c c và bi n c c b

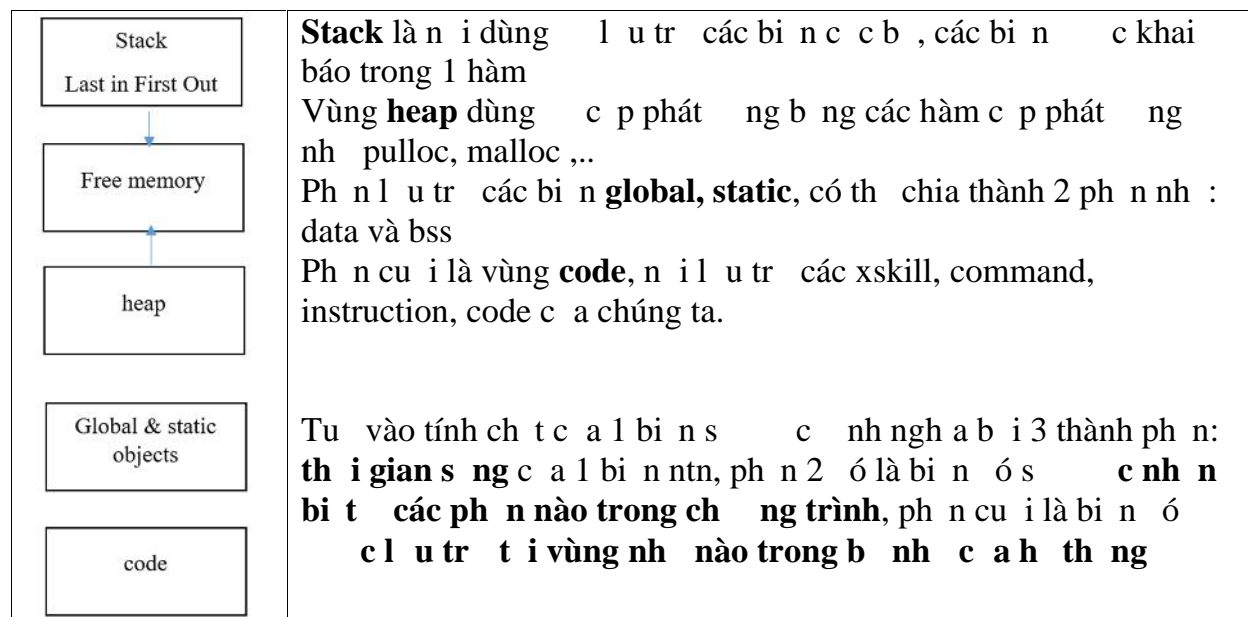
Th i gian s ng c a l b i n, khi nào thì l b i n c c p phát trên b nh và khi nào l b i n ó c xoá b trên b nh

nh ngh a timeline c a l b i n là th i gian s ng c a bi n ó có giá tr trong b nh (t n t i trong b nh) (valid)

T m v c c a l b i n (scope), l b i n trong ch ng trình c nh n bi t nh ng âu, c tìm th y hay nh n bi t nh ng ph n ch ng trình nào thì t ó ta nh ngh a ra scope c a l b i n, là l ph n c a ch ng trình n i có th nh n bi t ra l b i n c a chúng ta, nh n bi t ra bi n mà chúng ta kh i t o và s d ng c cái bi n ó.

Th i gian s ng và t m v c c a bi n c h p l i g i là **l p l u tr c a l b i n**, ph n t i p theo là bi n c l u tr trong b nh ntn

H th ng phân chia vùng nh nh sau:



Biến Local và biến Static

Tây, ta phân ra 2 loại biến, đó là biến global và biến local

Biến **global** là các biến khai báo không nằm trong bất cứ hàm nào cả và biến này sẽ tồn tại trong file mà khai báo biến đó.

Ti p theo là bi n **local**, là 1 bi n c khai báo trong 1 hàm và bi n này ch có kh n ng c nh n bi t trong 1 hàm. Bi n này c c p phát khi mà hàm ó c g i và k t thúc (b free kh i h th ng khi hàm ó k t thúc)

Và các biến global sẽ có 1 trục vùng nh là **vùng data hay vùng bss** phụ thuộc vào biến đó có kernel hay là character:

- ✓ n u b i n g l o b a l c k h i t o t r c v ù n g d a t a ,
- ✓ N u b i n m à c h a c k h i t o s c l u t r v ù n g b s s

Thị trường sản phẩm toàn cầu là khi chi phí vận chuyển thị trường toàn cầu có lợi hơn và có lợi. Khi chi phí vận chuyển thị trường toàn cầu có lợi hơn chi phí vận chuyển.

Th i gian s ng c a bi n local c b t u t khi hàm mà khai báo bi n c g i và s k t thúc, b free i khi mà hàm ó k t thúc.

→ T ng k t l i p h n l p l u t r c a l b i n, c nh ngh a g m 2 p h n là timeline c a l b i n, scope c a l b i n.

→ Trong bài này, chúng ta quy định cấu trúc vùng nào và đưa vào lập trình trên chúng ta phân biệt 2 loại biến là biến global và biến local