

Biến là gì

Vì sao cần biến khi lập trình

Biến là 1 định nghĩa giúp chúng ta truy xuất vào vùng nhớ thông qua 1 tên, tên đó gọi biến

Vì sao dung biến trong C, trong lập trình cần có các giá trị input và output của bài toán và ngoài ra cần các thành phần để lưu trữ tính toán trong thuật toán

Để lưu trữ các thành phần đầu vào hoặc đầu ra, hoặc các thành phần trung gian để tính toán, áp dụng trong các thuật toán thì cần dùng tới biến, việc khai báo biến bản chất là việc yêu cầu hệ thống **cấp phát cho chúng ta 1 vùng nhớ để lưu trữ các giá trị đầu vào và đầu ra, trung gian của biến.**

Các kiểu dữ liệu dữ liệu cơ bản trong C, các kiểu dữ liệu dẫn xuất

5 kiểu cơ bản: int, float, double, char, void; trong C

Kiểu int là số nguyên 0,1,2,3,4, -5,-6

Tuỳ vào hệ thống cấp phát kiểu dữ liệu này là 2byte hay là 4 byte

Float dùng để lưu trữ số thực 0.124, 0.5235, 4434.5253

Thường được cấp phát là 4byte đằng sau số thập phân là 6 giá trị 0.124324

Kiểu double thường để lưu trữ số thực và được cấp phát 8 bytes và độ chính xác đằng sau dấu chấm thập phân là 10 giá trị.

Kiểu char là kiểu dữ liệu dùng để lưu trữ ký tự và kiểu char thường được cấp phát 1 byte

Kiểu void là kiểu dữ liệu ko có kiểu dữ liệu nào cả, khi khai báo void thì hệ thống ko cấp phát bộ nhớ

Các từ khoá dẫn xuất

Gồm 3 từ khoá

Kiểu unsigned áp dụng cho kiểu int

Khi khai báo unsigned int thì sẽ được cấp phát thường là 2 byte or 4 byte tùy vào hệ thống và các giá trị này được lưu trữ là số nguyên dương

Kiểu short int là kiểu dữ liệu mà khi ta khai báo thì hệ thống sẽ cấp phát 1 nửa vùng nhớ so với kiểu int, tùy vào hệ thống mà được cấp phát 1 byte hay là 2 byte

Từ khoá long có thể áp dụng cho 2 kiểu int hoặc double, với long int thì ta sẽ được gấp 2 lần vùng nhớ cho kiểu int or double

Store class

Các lớp lưu trữ

Gồm 4 phần: phần 1 timeline của 1 biến, phần 2 là tầm vực của 1 biến, phần 3 là các biến trong memory, 4 là phân biệt sự khác nhau giữa biến toàn cục và biến cục bộ

Thời gian sống của 1 biến, khi nào thì 1 biến được cấp phát trên bộ nhớ và khi nào 1 biến đó được xoá bỏ trên bộ nhớ

Định nghĩa timeline của 1 biến là thời gian sống của biến đó có giá trị trong bộ nhớ (tồn tại trong bộ nhớ) (valid)

Tầm vực của 1 biến (scope), 1 biến trong chương trình được nhận biết ở những đâu, được tìm thấy hay nhận biết ở những phần chương trình nào thì từ đó ta định nghĩa ra scope của 1 biến, là 1 phần của chương trình nơi có thể nhận biết ra 1 biến của chúng ta, nhận biết ra biến mà chúng ta khởi tạo và sử dụng được cái biến đó.

Thời gian sống này và tầm vực của biến được hợp lại gọi là lớp lưu trữ của 1 biến, phần tiếp theo là biến được lưu trữ trong bộ nhớ ntn

Hệ thống phân chia vùng nhớ như sau:

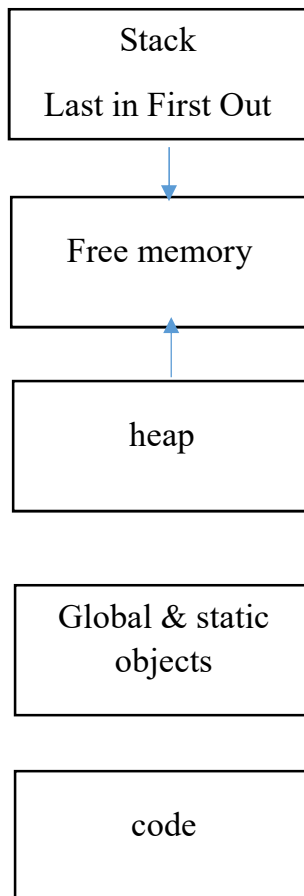
Stack là nơi dùng để lưu trữ các biến cục bộ, các biến được khai báo trong 1 hàm

Vùng heap dùng để cấp phát động bằng các hàm cấp phát động như pulloc, malloc ,..

Phần lưu trữ các biến global, static, có thể chia thành 2 phần nhỏ; data và bss

Phần cuối là vùng code, nơi lưu trữ các xskill, command, instruction, code của chúng ta.

Tuỳ vào tính chất của 1 biến sẽ được định nghĩa bởi 3 thành phần: phần 1, thời gian sống của 1 biến ntn, phần 2 đó là biến đó sẽ được nhận biết ở các phần nào trong chương trình, phần cuối là biến đó được lưu trữ tại vùng nhớ nào trong bộ nhớ của hệ thống



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

Biến global là các biến được khai báo ko nằm trong bất cứ hàm nào cả và biến này sẽ được nhận thấy ở trong file mà khai báo biến đó.

Tiếp theo là biến local, là 1 biến được khai báo trong 1 hàm và biến này chỉ có khả năng nhận biết trong 1 hàm. Biến này được cấp phát khi mà hàm đó được gọi và kết thúc (bị free khỏi hệ thống khi hàm đó kết thúc)

Và các biến global sẽ được lưu trữ ở vùng nhớ là data hay vùng bs phụ thuộc vào biến đó được khởi tạo hay là chưa khởi tạo, nếu biến global được khởi tạo trước ở vùng data, nếu biến mà chưa được khởi tạo sẽ được lưu trữ ở vùng bss

Thời gian sống của biến global là khi chương trình được khởi chạy thì biến global được khởi tạo và cấp phát. Khi chương trình kết thúc thì biến global được free khỏi hệ thống.

Đối với biến local, thời gian sống của 1 biến được bắt đầu từ khi hàm mà khai báo biến được gọi và sẽ kết thúc, bị free đi khi mà hàm đó kết thúc.

Tổng kết lại phần lớp lưu trữ của 1 biến, được định nghĩa gồm 2 phần là timeline của 1 biến, scope của 1 biến. khi mà lớp lưu trữ này thì nó sẽ quy định biến đó được lưu trữ ở vùng nào và dựa vào lớp lưu trữ chúng ta phân biệt được 2 loại biến đó là biến global và biến local