

Bài 47: CẤP PHÁT MẢNG ĐỘNG (DYNAMICALLY ALLOCATING ARRAYS)

Xem bài học trên website để ủng hộ Kteam: [Cấp phát mảng động \(Dynamically allocating arrays\)](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài học trước, mình đã chia sẻ cho các bạn về [CẤP PHÁT & GIẢI PHÓNG VÙNG NHỚ CHO BIẾN ĐƠN TRONG C++](#). Hôm nay, chúng ta sẽ cùng tìm hiểu về **Cấp phát mảng động (Dynamically allocating arrays)**.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về:

- [CON TRỎ TRONG C++](#) (Pointer)
- [CẤP PHÁT ĐỘNG TRONG C++](#) (Dynamic memory allocation)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Cấp phát mảng động trong C++
- Xóa mảng động trong C++
- Khởi tạo mảng động trong C++
- Thay đổi kích thước mảng động

Cấp phát mảng động trong C++

Khi khai báo **mảng tĩnh**, kích thước mảng phải được **xác định tại thời điểm biên dịch** và không bao giờ thay đổi.

Cấp phát động một mảng cho phép chúng ta xác định kích thước mảng **trong thời gian chạy chương trình**.

Để cấp phát và thu hồi mảng động, C++ cung cấp toán tử **new[]** và **delete[]**:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Nhập chiều dài mảng: ";
    int length;
    cin >> length;

    int *array = new int[length]; // kích thước mảng có thể là biến số

    // sử dụng mảng
    // ...

    delete[] array; // trả lại vùng nhớ mảng array cho hệ điều hành

    return 0;
}
```

Lưu ý rằng, **Cấp phát bộ nhớ động** sử dụng vùng nhớ được quản lý bởi hệ điều hành được gọi là **heap**, vì vậy kích thước một mảng động có thể lớn hơn rất nhiều so với khai báo mảng tĩnh.

Số lượng phần tử có thể được cung cấp **trong khi chương trình đang chạy** là điểm đặc biệt của kỹ thuật **cấp phát mảng động**.

Xóa mảng động trong C++

Khi chúng ta không còn sử dụng một mảng được cấp phát động, chúng ta cần trao quyền quản lý vùng nhớ đó lại cho hệ điều hành. Đối với **mảng**, điều này được thực hiện thông qua toán tử **delete[]**:

```
int * array = new int[length]; // cấp phát mảng động length phần tử

// sử dụng mảng

delete[] array; // trả lại vùng nhớ mảng array cho hệ điều hành
```

Khởi tạo mảng động trong C++

Trước phiên bản C++11, để khởi tạo một mảng động, bạn cần lập qua các phần tử của mảng và gán giá trị cho chúng:

```
int *array = new int[4];
array[0] = 5;
array[1] = 7;
array[2] = 4;
array[3] = 8;
```

Tuy nhiên từ C++ 11, mảng động có thể được khởi tạo đơn giản hơn:

```
int *array = new int[4]{ 5, 7, 4, 8 };
```

Lưu ý rằng mảng động phải được khai báo độ dài rõ ràng khi khởi tạo:

```
int fixedArray[]{ 1, 2, 3 }; // ok

int *dynamicArray1 = new int[] {1, 2, 3}; // lỗi

int *dynamicArray2 = new int[3]{ 1, 2, 3 }; // ok
```

Ngoài ra, bạn có thể khởi tạo một mảng động có n phần tử 0:

```
int *array = new int[10](); // mảng động có 10 phần tử 0
```

Thay đổi kích thước mảng động

Cấp phát động một mảng cho phép bạn cung cấp độ dài mảng tại thời điểm chương trình đang chạy. Tuy nhiên, C++ không cung cấp sẵn tính năng thay đổi kích thước một mảng đã được cấp phát.

Để thay đổi kích thước mảng, chúng ta cần thực hiện:

- **Bước 1:** Cấp phát động một mảng mới.
- **Bước 2:** Sao chép các phần tử từ mảng cũ sang vùng nhớ mới.
- **Bước 3:** Xóa mảng cũ.

```
#include <iostream>
using namespace std;

int main()
{
    int size = 3;
    // cấp phát động và khởi tạo mảng 3 phần tử
    int *array = new int[size]{ 3, 5, 7 };

    int newSize = 6;
    // cấp phát động mảng mới 6 phần tử
    int *resize = new int[newSize];

    // sao chép phần tử
    for (int i = 0; i < size; i++)
    {
        resize[i] = array[i];
    }

    delete[] array; // Xóa mảng hiện tại

    array = resize; // Trỏ sang mảng đã resize
    size = newSize; // kích thước mảng mới
```

```
// sử dụng mảng array đã resize  
// ...  
  
delete[] array; // Thu hồi mảng  
  
system("pause");  
return 0;  
}
```

Tuy nhiên, các bước này khá phức tạp và dễ gây ra lỗi trong trường hợp phần tử mảng là những kiểu dữ liệu phức tạp.

Mặt khác, C++ cung cấp một loại mảng có thể thay đổi kích thước gọi là `std::vector` thuộc thư viện chuẩn `std`. Phần này sẽ được giới thiệu sau trong bài **LỚP DỰNG SẴN VECTOR TRONG C++**.

Kết luận

Qua bài học này, bạn đã nắm được cách Cấp phát mảng động (Dynamically allocating arrays). Với kỹ thuật này, bạn có thể sử dụng mảng với số lượng phần tử lớn hơn và có thể thay đổi trong quá trình chạy chương trình.

Lưu ý rằng khi sử dụng cấp phát động, bạn cần nắm rõ những kiến thức cơ bản về cấp phát và giải phóng vùng nhớ trong bài viết này để tránh rò rỉ bộ nhớ, cũng như những vấn đề về vùng nhớ khác.

Trong bài tiếp theo, mình sẽ giới thiệu cho các bạn khái niệm [CON TRỎ & HẰNG \(Pointers and const\) trong C++](#).

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.