

KỸ THUẬT LẬP TRÌNH C/C++

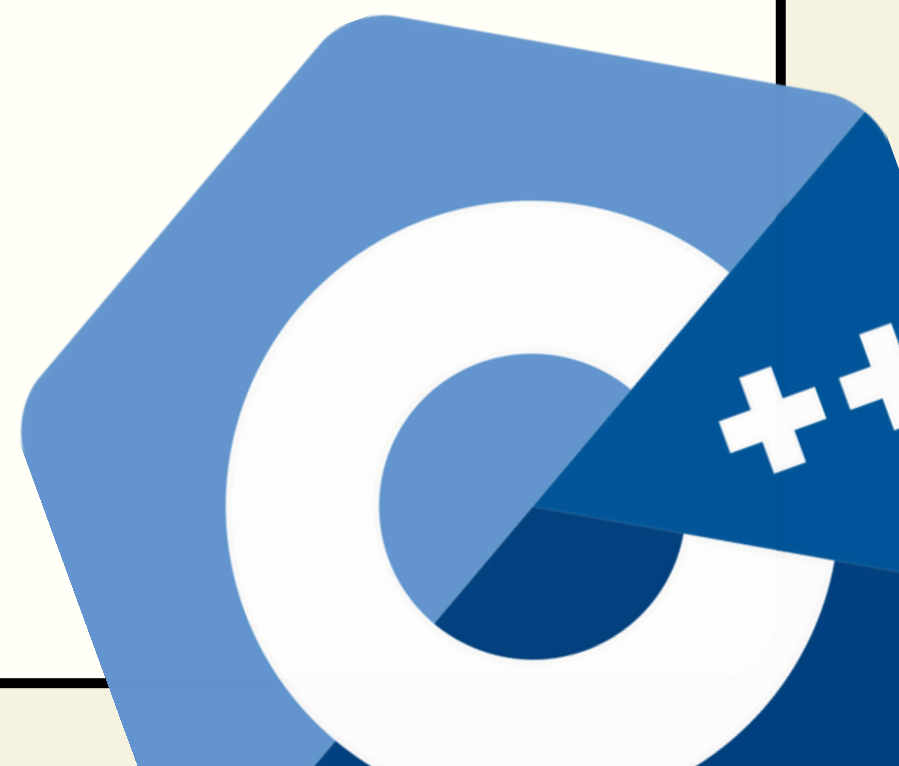
Dạy bởi 1 SVBK

Đại học Bách Khoa Hà Nội
Kỹ thuật Điện tử - Viễn thông

2024-25

NỘI DUNG

- for
- while
- do - while
- break & continue



GIỚI THIỆU CHUNG

Trong buổi học trước, ta đã học về các **cấu trúc điều khiển và rẽ nhánh** như **if, else, else if và switch - case**. Bây giờ chúng ta sẽ tìm hiểu về các **vòng lặp** trong C và ứng dụng của nó.

Vòng lặp trong lập trình được sử dụng để **lặp lại một khối mã cho đến khi điều kiện được chỉ định được đáp ứng**. Câu lệnh vòng lặp cho phép lập trình viên thực thi một câu lệnh hoặc nhóm câu lệnh nhiều lần mà **không cần lặp lại mã**.

VÒNG LẶP FOR

CÚ PHÁP

```
for (biểu thức khởi tạo; biểu thức logic; biểu thức biến đổi) {  
    khối biểu thức  
}
```

Trong ngôn ngữ lập trình C, vòng lặp for là một cấu trúc điều khiển được sử dụng để lặp lại một khối mã một số lần xác định, dựa trên một điều kiện cụ thể. Nó rất phổ biến và tiện lợi vì cú pháp của nó cho phép khởi tạo, kiểm tra điều kiện và cập nhật biến điều khiển trong cùng một dòng.

VÒNG LẶP FOR

VÍ DỤ

```
for (int i = 0; i < 5; i++) {  
    printf("i = %d\n", i);  
}
```

OUTPUT

```
i = 0  
i = 1  
i = 2  
i = 3  
i = 4
```

GIẢI THÍCH

int i = 0: Khởi tạo biến i với giá trị 0.

i < 5: Điều kiện kiểm tra, vòng lặp chạy khi i nhỏ hơn 5.

i++: Tăng i lên 1 sau mỗi lần lặp.

Khối mã **printf("i = %d\n", i);** in giá trị của i mỗi lần lặp.

VÒNG LẶP FOR

CHÚ Ý

Các thành phần trong for có thể linh hoạt:

- Một hoặc cả ba phần (khởi_tạo, điều_kiện, cập_nhật) đều có thể bỏ qua, nhưng **dấu “;” vẫn phải giữ lại.**
- Ta có thể dùng nhiều biến trong for, miễn là chúng được **phân tách bằng dấu phẩy.**

VÍ DỤ

Vòng lặp vô hạn: không có điều kiện dừng.

```
int main() {  
    for (;;) {  
        printf("Vòng lặp vô hạn!\n");  
    }  
    return 0;  
}
```

Nhiều biến khởi tạo trong vòng lặp.

```
int main() {  
    for (int i = 0, j = 10; i < j; i++, j--) {  
        printf("i = %d, j = %d\n", i, j);  
    }  
    return 0;  
}
```

VÒNG LẶP FOR - LỒNG NHAU

Vòng lặp for lồng nhau (**nested loop**) là một vòng lặp for bên trong một vòng lặp for khác. Mỗi lần **vòng lặp bên ngoài (outer loop)** chạy một lần, **vòng lặp bên trong (inner loop)** sẽ chạy toàn bộ số lần quy định của nó.

- Số lần vòng ngoài chạy: **m lần**
- Số lần vòng trong chạy mỗi khi vòng ngoài lặp 1 lần: **n lần**
→ Số lần chạy tổng cộng của vòng trong là: **$m \times n$ lần**

VÒNG LẶP WHILE

CÚ PHÁP

```
while (biểu thức logic) {  
    khối biểu thức  
}
```

Vòng lặp while là một cấu trúc điều khiển dùng để lặp lại một khối mã miễn là một điều kiện cụ thể còn đúng (true). Nó kiểm tra điều kiện trước khi thực hiện khối mã, vì vậy nếu điều kiện sai ngay từ đầu, khối mã sẽ không được thực thi lần nào.

VÒNG LẶP WHILE

VÍ DỤ

```
int i = 0;
while (i < 5) {
    printf("i = %d\n", i);
    i++;
}
```

OUTPUT

```
i = 0
i = 1
i = 2
i = 3
i = 4
```

GIẢI THÍCH

int i = 0: Khởi tạo biến i với giá trị 0 (khởi tạo nằm ngoài vòng lặp).

i < 5: Điều kiện kiểm tra, vòng lặp chạy khi i nhỏ hơn 5.

i++: Tăng giá trị i lên 1 trong khối mã để tránh vòng lặp vô hạn.

printf("i = %d\n", i): In giá trị của i mỗi lần lặp.

VÒNG LẶP WHILE

VÍ DỤ

```
int i = 10;  
do {  
    printf("i = %d\n", i);  
    i++;  
} while (i < 5);
```

OUTPUT

`i = 10`

GIẢI THÍCH

Vì **i = 10** và điều kiện **i < 5** sai ngay từ đầu, nhưng khối mã vẫn chạy một lần trước khi kiểm tra điều kiện.

SO SÁNH CÁC VÒNG LẶP VỚI NHAU

So sánh `for`, `while`, và `do-while`:

Tiêu chí	<code>for</code>	<code>while</code>	<code>do-while</code>
Khởi tạo	Trong cú pháp	Ngoài vòng lặp	Ngoài vòng lặp
Kiểm tra điều kiện	Trước khi chạy khối mã	Trước khi chạy khối mã	Sau khi chạy khối mã
Số lần chạy tối thiểu	0 (nếu điều kiện sai)	0 (nếu điều kiện sai)	1 (luôn chạy ít nhất 1 lần)
Khi nào dừng	Số lần lặp biết trước	Điều kiện dừng linh hoạt	Cần chạy ít nhất 1 lần

BREAK & CONTINUE

Trong ngôn ngữ lập trình C, **break** và **continue** là hai câu lệnh điều khiển luồng (control flow statements) được sử dụng để thay đổi cách thực thi của các vòng lặp như **for**, **while**, hoặc **do-while**.

BREAK & CONTINUE

	Break	Continue
Chức năng	Dùng để thoát hoàn toàn khỏi vòng lặp hiện tại, ngay lập tức chuyển luồng thực thi đến câu lệnh ngay sau vòng lặp.	Bỏ qua phần còn lại của lần lặp hiện tại và chuyển sang lần lặp tiếp theo của vòng lặp.
Cách hoạt động	Khi break được gọi, vòng lặp chứa nó sẽ dừng lại , bất kể điều kiện của vòng lặp có còn đúng hay không.	Khi continue được gọi, mã lệnh phía sau nó trong lần lặp hiện tại sẽ bị bỏ qua , và vòng lặp sẽ kiểm tra lại điều kiện để bắt đầu lần lặp tiếp theo.
Ứng dụng	Thường được để thoát khỏi vòng lặp khi một điều kiện cụ thể được thỏa mãn.	Dùng khi bạn muốn bỏ qua một số trường hợp cụ thể nhưng vẫn tiếp tục vòng lặp.

BREAK & CONTINUE

VÍ DỤ - BREAK

```
for (int i = 0; i < 10; i++) {  
    if (i == 5) {  
        break;  
    }  
    printf("%d ", i);  
}
```

OUTPUT

0 1 2 3 4

GIẢI THÍCH

Chương trình chạy vòng lặp for để **in ra các số từ 0 đến 9**, trong vòng lặp for có điều kiện **nếu i = 5 thì dừng lại và thoát vòng lặp**. Output của chương trình là các số từ 0 đến 4.

BREAK & CONTINUE

VÍ DỤ - CONTINUE

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0) {  
        continue;  
    }  
    printf("%d ", i);  
}
```

OUTPUT

1 3 5 7 9

GIẢI THÍCH

Chương trình chạy vòng lặp for để **in ra các số từ 0 đến 9**, trong vòng lặp for có điều kiện **nếu như gặp các số chẵn thì sẽ continue** - bỏ qua các giá trị đó và thực thi tiếp câu lệnh.