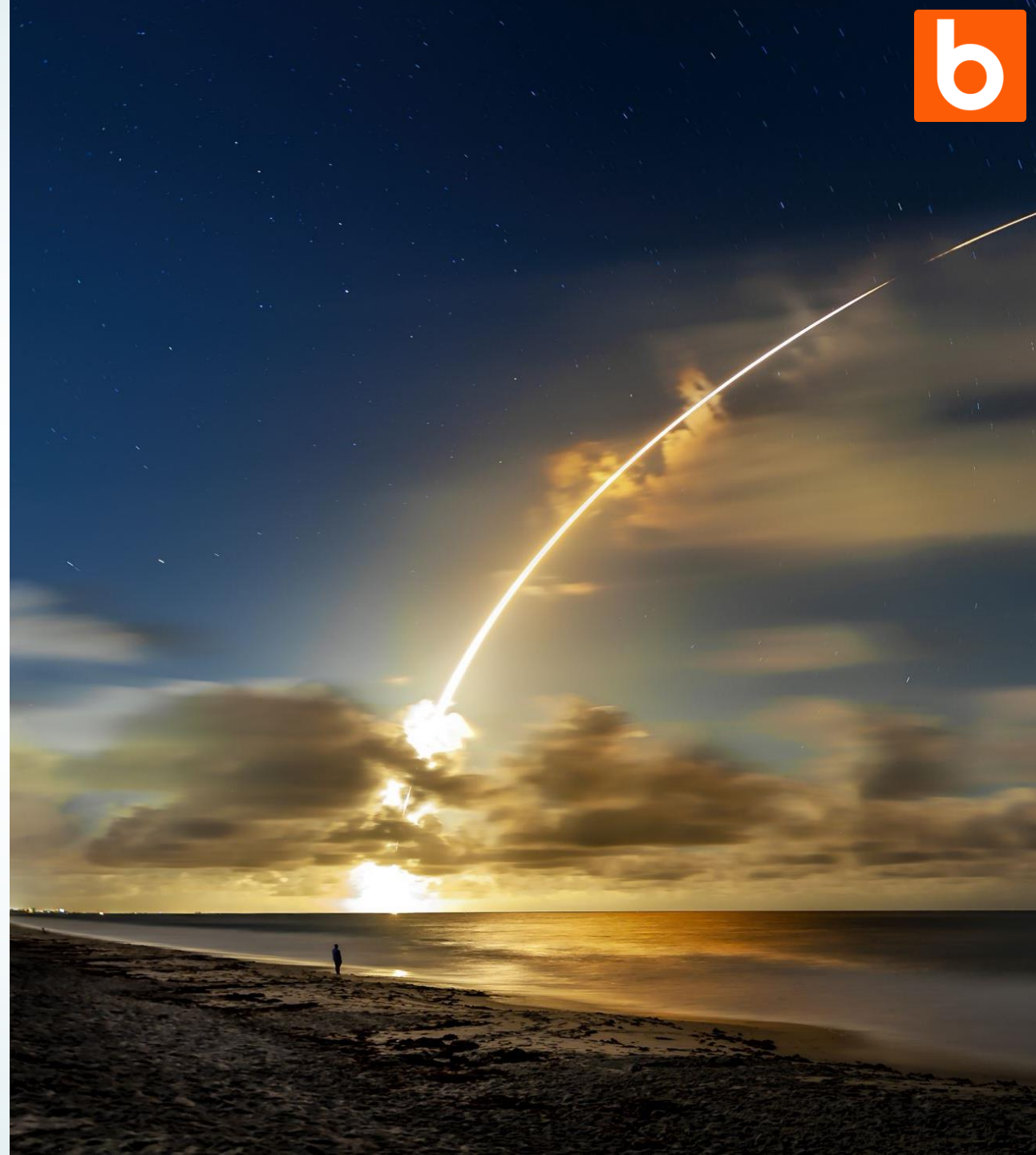


Bài 26: Hàm đệ quy

- ✓ Khái niệm, đặc điểm
- ✓ Mục đích sử dụng
- ✓ Lưu ý
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành



Khái niệm

- Định nghĩa: hàm đệ quy là hàm gọi lại chính nó một cách trực tiếp hoặc gián tiếp thông qua các hàm khác
- Đệ quy là một chủ đề phức tạp sẽ được phân tích cụ thể hơn trong môn cấu trúc dữ liệu và giải thuật

Đặc điểm

- Một hàm đệ quy chỉ biết cách giải quyết các trường hợp cơ bản nhất hay còn gọi trường hợp cơ sở, điểm dừng
- Với các lời gọi hàm trùng với trường hợp cơ sở này thì hàm đệ quy đơn giản chỉ cần đưa ra giá trị cơ sở tương ứng
- Với lời gọi hàm phức tạp hơn, hàm đệ quy sẽ chia vấn đề thành 2 phần, phần thứ nhất hàm tự biết cách giải quyết, phần thứ hai nó không biết cách giải quyết
- Ở phần thứ hai này, vấn đề cần xử lý có vẻ giống hệt vấn đề ban đầu nhưng nhỏ hơn một chút. Để giải quyết, hàm đệ quy gọi lại chính nó

Đặc điểm

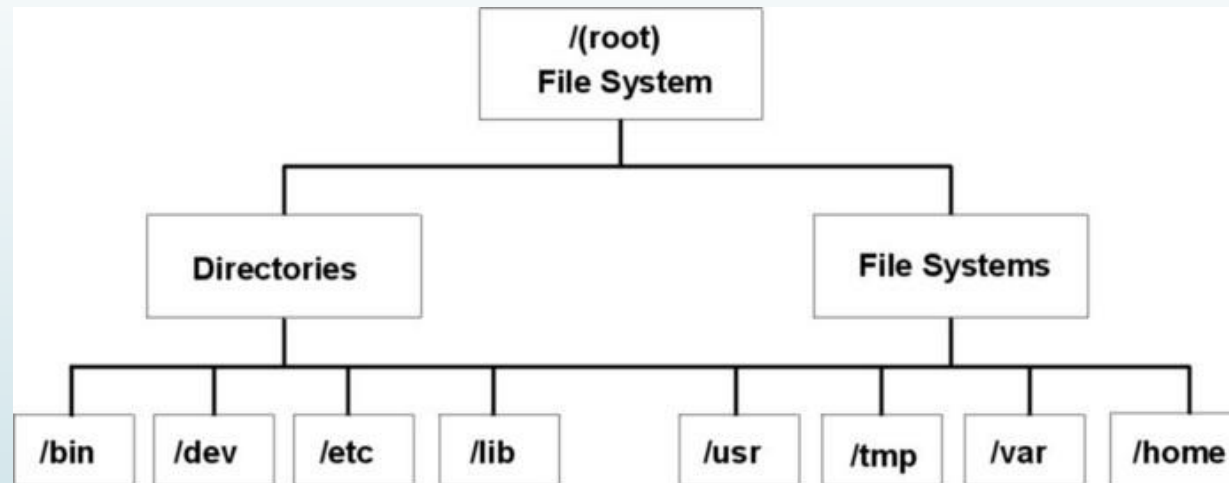
- Lúc này, hành động gọi lại chính nó được gọi là lời gọi đệ quy, bước đệ quy. Hàm chứa lời gọi đệ quy là hàm đệ quy
- Khi gọi đệ quy, lời gọi phía trước sẽ chờ đợi kết quả thực hiện của các bước đệ quy phía sau
- Để hàm đệ quy có thể kết thúc, mỗi lần đệ quy vấn đề phải trở nên đơn giản hơn và tiến dần đến trường hợp cơ sở hay điểm dừng

Mục đích sử dụng

- Giải quyết các vấn đề có thể chia nhỏ nhưng vẫn là vấn đề ban đầu
- Đặc biệt hữu ích khi vấn đề có nhiều nhánh và quá phức tạp để giải quyết chúng bằng vòng lặp
- Các cấu trúc dữ liệu điển hình như cây, đồ thị thường xuyên áp dụng đệ quy để giải quyết vấn đề

Mục đích sử dụng

- Ví dụ: tìm kiếm các file trong hệ thống file của hệ điều hành



Các lưu ý với hàm đệ quy

- Không áp dụng đệ quy với hàm main vì mục đích chính của hàm này là kích hoạt, khởi chạy chương trình ứng dụng
- Các vấn đề có thể giải quyết bằng vòng lặp thì nên giải quyết bằng vòng lặp
- Nếu không có yêu cầu gì về hiệu năng, ta có thể sử dụng đệ quy
- Dùng đệ quy để giải quyết một vấn đề phức tạp có thể tốn rất nhiều tài nguyên, chi phí(thời gian, CPU, bộ nhớ)

Ví dụ minh họa

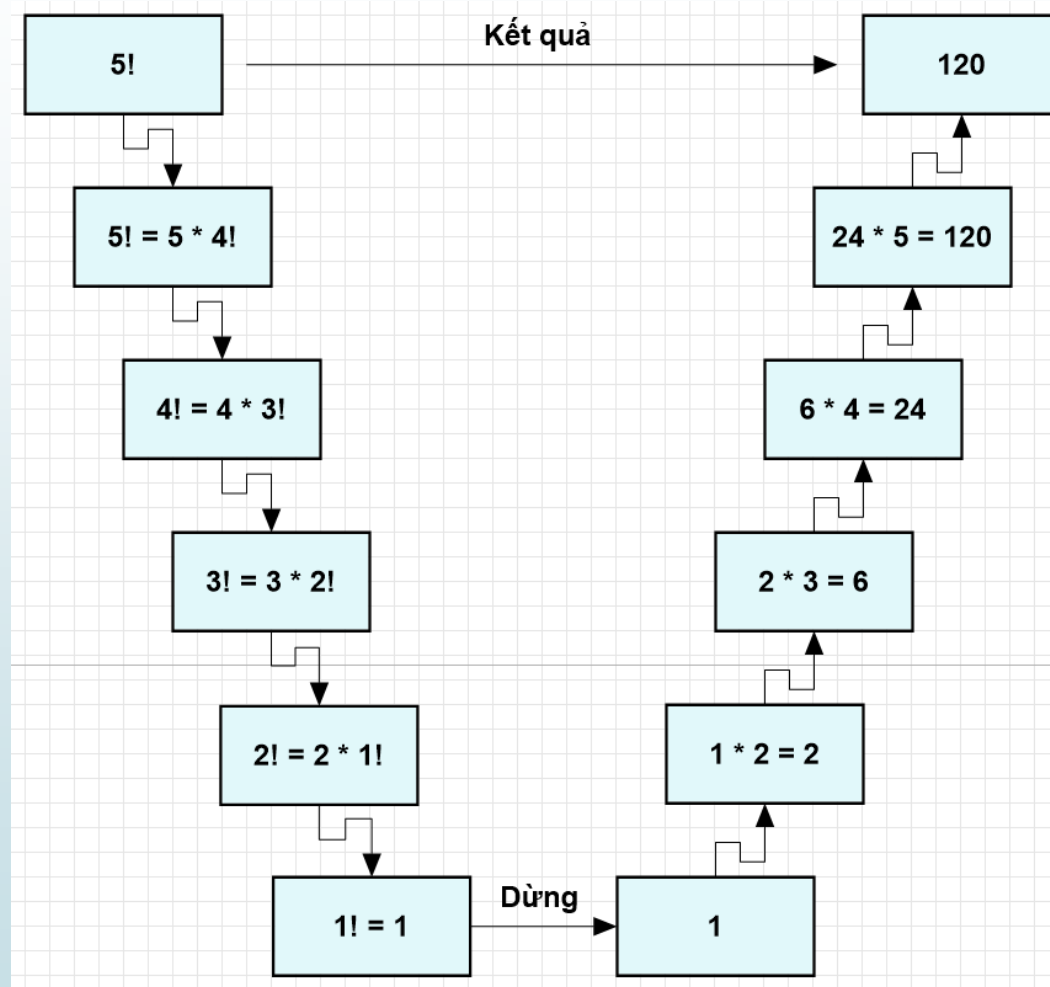
Ví dụ 1: Tính $n!$

➤ $5! = 5 * 4 * 3 * 2 * 1$

➤ $5! = 5 * (4 * 3 * 2 * 1) = 5 * 4!$

➤ Vậy để tính $n!$, ta lấy $n * (n-1)!$

Minh họa bằng hình ảnh

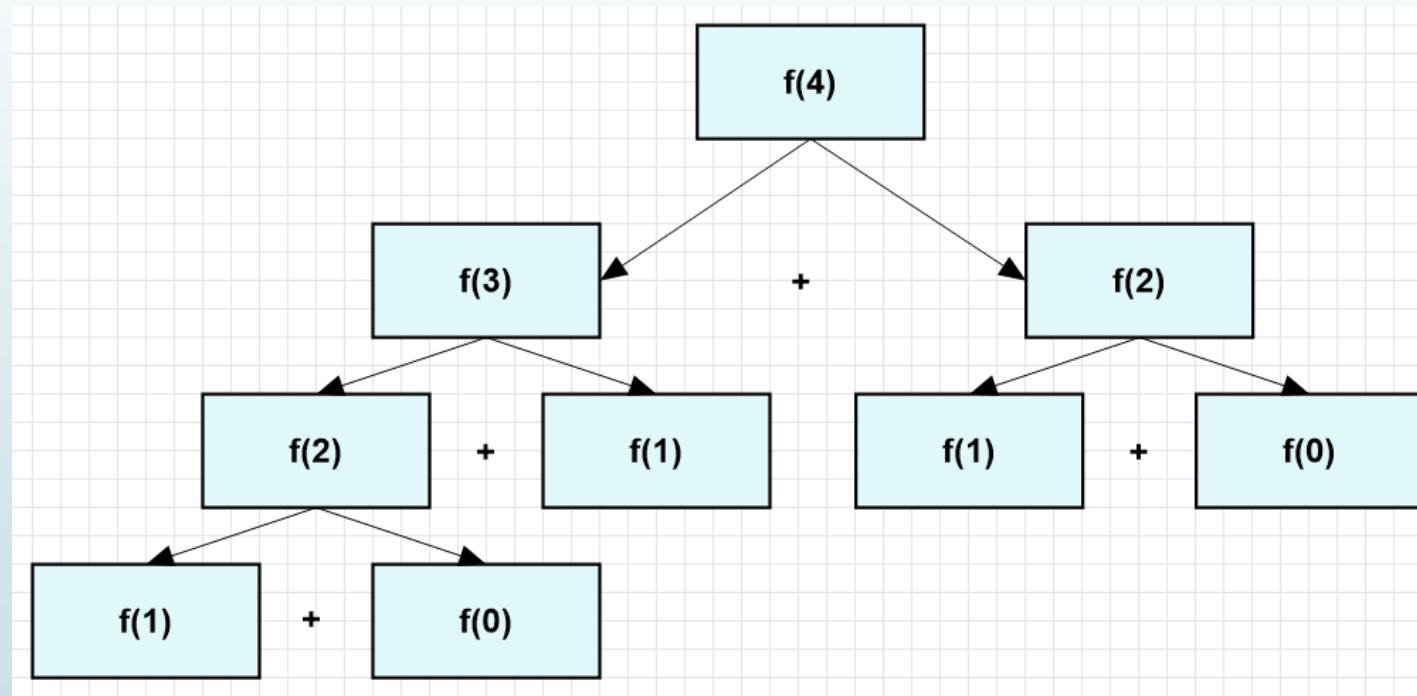


Code mẫu tính $n!$

```
unsigned long long factorial(unsigned int n) {  
    if (n < 2) { // trường hợp cơ sở  
        return 1;  
    }  
    else { // bước đệ quy  
        return n * factorial(n - 1);  
    }  
}
```

Ví dụ tìm số Fibonacci $f(n)$

Số Fibonacci được định nghĩa như sau: $f_0 = 0$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$



Code mẫu tìm số $f(n)$

```
unsigned long long fibonacci(unsigned int n) {  
    if (n == 0 || n == 1) { // điểm dừng  
        return n;  
    }  
    else { // bước đệ quy  
        return fibonacci(n - 1) + fibonacci(n - 2);  
    }  
}
```

Nội dung tiếp theo

Đệ quy vs vòng lặp