Tổng hợp kiến thức

1, Khái niệm về thuật toán FCFS (First Come First Serve)

* Công việc nào tới trước sẽ xử lí trước
* Là thuật toán lập lịch CPU, thực hiện các yêu cầu và quy trình được xếp theo hàng đợi theo thứ tự đến của chúng

2, Đặc điểm

* Là thuật toán lập lịch CPU dễ dàng và đơn giản nhất
* Hàng đợi của tiến trình được tổ chức theo kiểu FIFO (First In First Out)
* Mọi tiến trình đều được phục vụ theo trình tự xuất hiện cho đến khi kết thúc hoặc bị ngắt
* Khi tiến trình đi vào hàng đợi sẵn sàng, PCB (Process Control Block) của tiến trình đó dẽ được liên kết vào cuối hàng đợi theo nguyên tắc FIFO
  + Hàng đợi sẵn sàng: là nơi mà các tiến trình đang chờ được CPU xử lý. Khi một tiến trình được tạo ra và sẵn sàng để thực hiện, nó sẽ được đưa vào hàng đợi sẵn sàng
  + PCB (Process Control Block – Khối điều khiển tiến trình): là một cấu trúc quan trọng trong hệ điều hành, chứa tất cả các thông tin cần thiết để quản lý và theo dõi tiến trình đó; có thể hiểu đơn giản PCB giống như “hồ sơ” của một tiến trình
* Khi CPU hoàn thành việc xử lý một tiến trình và trở nên rảnh rỗi, nó sẽ lấy tiến trình đầu tiên của hàng đợi để thực hiện xử lý tiếp theo và điều này diễn ra tuần tự

3, Ưu điểm và nhược điểm của thuật toán

* Ưu điểm:
  + Hỗ trợ các loại lập lịch không ưu tiên lẫn ưu tiên
  + Công việc luôn thực thi theo thứ tự đến
  + Dễ sử dụng, đơn giản trong triển khai
* Nhược điểm:
  + Hiệu suất kém, thời gian đợi chung khá dài nếu một tiến trình có thời gian xử lí lâu (Hiệu ứng hộ tống – Convoy Effect)
  + Thiếu tính linh hoạt và không có khả năng ưu tiên

Hiệu ứng hộ tống: xảy ra khi một tiến trình lớn có thời gian thực thi dài chặn các tiến trình nhỏ hơn, dẫn đến hệ thống hoạt động kém hiệu quả

\*Tác động của hiệu ứng hộ tống

* Tăng thời gian chờ
* Giảm hiệu suất
* Hiệu ứng nghẽn cổ chai

\*Cách khắc phục

* Sử dụng các thuật toán lập lịch ưu tiên: SJF (Short Job First) hoặc RR (Round Robin)
* Lập lịch đa cấp độ

4, Cách thức hoạt động của thuật toán

Thuật toán lập lịch trình FCFS thường được thông qua các số liệu được thu thập từ trước lập thành bảng dữ liệu và từ đó biểu diễn thành biểu đồ Gantt

* Ví dụ: Trong lịch trình này, ta có các sự kiện lần lượt là EV1, EV2, EV3, EV4 và EV5. Mỗi sự kiện sẽ có các thông số về Thời gian đến (Arrival time) và Thời gian bùng nổ (Burst time) tương ứng. Từ đó ta tính ra Thời gian hoàn thành (Completion time) và sau cùng tính Thời gian chờ (Waiting time) hay thêm Thời gian chờ trung bình (nếu bài toán yêu cầu)

|  |  |  |  |
| --- | --- | --- | --- |
| Sự kiện | Thời gian đến | Thời gian bùng nổ | Thời gian chờ |
| EV1 | 0 | 2 | 0 |
| EV2 | 3 | 4 | 8 |
| EV3 | 1 | 6 | 1 |
| EV4 | 6 | 5 | 9 |
| EV5 | 2 | 3 | 6 |

* ﻿﻿Các bước làm:
* ﻿﻿B1: Sắp xếp các sự kiện có Thời gian đến theo thứ tự từ bé đến lớn, sự kiện nào đến trước được thực thi trước
* ﻿﻿B2: Tính Thời gian hoàn thành theo phương thức sự kiện sau bằng tổng Thời gian bùng nổ của chính nó và các sự kiện trước
* ﻿﻿B3: Lập biểu đồ Gantt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EV1 | EV3 | EV5 | EV2 | EV4 |

0 2 8 11 15 20

* B4: Tính Thời gian chờ theo công thức:  
  Thời gian chờ = Thời gian bắt dầu - Thời gian đến
* ﻿﻿B5: Tính Thời gian chờ trung bình (nếu có)  
  Thời gian chờ trung bình = (0 +8 +1 +9 +6) :5 = 4,8

5, Lưu đồ thuật toán

6, Các bước xây dựng ứng dụng lịch trình dựa trên FCFS

Các bước xây dựng ứng dụng lịch trình dựa trên FCFS

* Thiết kế Cấu Trúc Dữ Liệu:
  + Sự kiện: Mỗi sự kiện sẽ được biểu diễn bởi một đối tượng có các thuộc tính:
    - ID: Mã định danh duy nhất
    - Tên sự kiện
    - Thời gian bắt đầu
    - Thời gian kết thúc
    - Mô tả (tùy chọn)
  + Danh sách sự kiện: Sử dụng một danh sách (list) hoặc hàng đợi (queue) để lưu trữ các sự kiện. Danh sách này sẽ được sắp xếp theo thời gian bắt đầu tăng dần để đảm bảo tuân thủ nguyên tắc FCFS
* Giao Diện Người Dùng:
  + Thêm sự kiện: Cho phép người dùng nhập thông tin của một sự kiện mới và thêm vào danh sách
  + Sửa sự kiện: Cho phép người dùng chỉnh sửa thông tin của một sự kiện đã tồn tại
  + Xóa sự kiện: Cho phép người dùng xóa một sự kiện khỏi danh sách
  + Xem lịch: Hiển thị lịch trình các sự kiện đã được thêm vào, sắp xếp theo thứ tự thời gian
  + Tìm kiếm sự kiện: Cho phép người dùng tìm kiếm một sự kiện cụ thể dựa trên tên hoặc khoảng thời gian
* Xử lý Logic:
  + Thêm sự kiện:
    - Tạo một đối tượng sự kiện mới
    - Thêm đối tượng này vào cuối danh sách sự kiện
    - Sắp xếp lại danh sách theo thời gian bắt đầu tăng dần
  + Sửa sự kiện:
    - Tìm sự kiện cần sửa dựa trên ID
    - Cập nhật thông tin mới cho sự kiện đó
    - Sắp xếp lại danh sách nếu cần
  + Xóa sự kiện:
    - Tìm sự kiện cần xóa dựa trên ID
    - Xóa sự kiện khỏi danh sách
  + Hiển thị lịch:
    - Duyệt qua danh sách sự kiện và hiển thị thông tin của từng sự kiện trên giao diện
    - Sử dụng một thư viện đồ họa để tạo giao diện lịch trực quan
* Lưu trữ Dữ Liệu:
  + Lưu trữ cục bộ: Sử dụng một file văn bản hoặc một cơ sở dữ liệu đơn giản như SQLite để lưu trữ thông tin các sự kiện
  + Lưu trữ đám mây: Sử dụng các dịch vụ đám mây như Firebase, Google Cloud để lưu trữ và đồng bộ dữ liệu giữa các thiết bị
* Các Tính Năng Nâng Cao:
  + Thông báo: Gửi thông báo cho người dùng trước khi một sự kiện bắt đầu.
  + Lặp lại sự kiện: Cho phép tạo các sự kiện lặp lại hàng ngày, hàng tuần, hoặc hàng tháng.
  + Chia sẻ lịch: Cho phép người dùng chia sẻ lịch trình của mình với người khác
  + Tích hợp với các ứng dụng khác: Ví dụ, tích hợp với Google Calendar để đồng bộ dữ liệu

\*\*Lưu Ý:

* Hiệu suất: Khi danh sách sự kiện trở nên lớn, việc ắp xếp lại liên tục có thể ảnh hưởng đến hiệu suất. Có thể cân nhắc sử dụng các cấu trúc dữ liệu khác như cây nhị phân cân bằng để cải thiện hiệu suất tìm kiếm và sắp xếp
* Giao diện người dùng: Giao diện người dùng nên trựcquan, dễ sử dụng và phù hợp với thiết bị.
* Tính năng: Có thể mở rộng ứng dụng với nhiều tính năng khác như tạo nhóm sự kiện, quản lý nhiều lịch, tích hợp với các ứng dụng khác
* Các thư viện hữu ích:
  + Python: Tkinter (tạo giao diện), SQLite (cơ sở dữ liệu), datetime (xử lý thời gian)
  + JavaScript: React, Angular (tạo giao diện web), LocalStorage (lưu trữ cục bộ)

Bằng cách kết hợp các yếu tố trên, bạn có thể xây dựng một ứng dụng lịch trình đơn giản nhưng hiệu quả dựa trên thuật toán FCFS