

Thông tin dưới đây được dịch từ tài liệu StudyGuide tại trang OS-Book của sách giáo trình chính và phần Summary của sách. Mọi lỗi sai sót hay góp ý, xin gửi về cho tôi qua email trantrungtin@tdtu.edu.vn

Study Guide for Lecture notes ch05

- Quá trình thực thi của tiến trình là một chu kỳ đan xen giữa thực thi trên CPU và chờ đợi nhập xuất. Bộ lập lịch CPU ra quyết định khi một tiến trình:

- Chuyển từ Running sang waiting (không phải do quyền ưu tiên / nonpreemptive)
- Chuyển từ running sang ready (bộ lập lịch giữ quyền ưu tiên / preemptive)
- Chuyển từ waiting sang ready (bộ lập lịch giữ quyền ưu tiên / preemptive)
- Kết thúc (không phải do quyền ưu tiên / nonpreemptive)

- Bộ điều phối (dispatcher) giao quyền sử dụng CPU đến tiến trình được chọn lựa bởi Bộ lập lịch ngắn hạn (short-term scheduler)

- Trễ điều phối (Dispatch latency) là thời gian để bộ điều phối dừng một tiến trình và chuyển sang một tiến trình khác.

- Giải thuật lập lịch được chọn tùy vào tiêu chí tối ưu (ví dụ như thông lượng, thời gian quay vòng, .v.v.)

- FCFS, SJF, Shortest-Remaining-Time-First (preemptive SJF), Round Robin, Priority.

- Xác định độ dài của lượt sử dụng CPU tiếp theo: tính trung bình theo cấp số nhân (Exponential Averaging):

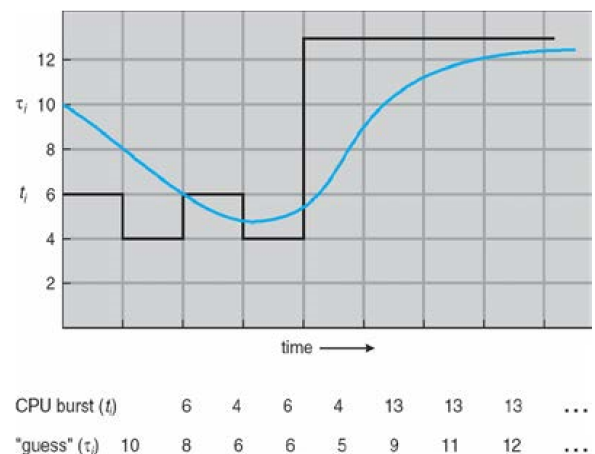
- t_n = thời gian sử dụng CPU thực tế ở lượt chạy thứ n .
- τ_{n+1} = thời gian dự báo của lượt chạy CPU tiếp theo.
- Giá trị α , $0 \leq \alpha \leq 1$ (thông thường $\alpha = 0.5$)
- Khi đó: $\tau_{n+1} = \alpha * t_n + (1-\alpha)\tau_n$

- Lập lịch Ưu tiên (Priority Scheduling) có thể dẫn đến sự thiếu thốn tài nguyên (starvation), và có thể giải quyết bằng giá trị tuổi (aging) của một tiến trình (tiến trình tồn tại trong hệ thống càng lâu thì độ ưu tiên sẽ càng tăng lên)

- Với lập lịch xoay vòng (Round Robin), một phần thời gian nếu thiết lập quá nhỏ sẽ dẫn đến một số lần chuyển ngữ cảnh rất lớn.

- Phần thời gian nên chọn sao cho 80% tiến trình có một lượt cần chạy CPU ngắn hơn thời gian một phần.

- Hàng chờ đa mức (Multilevel Queues) và Hàng chờ đa mức có phản hồi (Multilevel Feedback Queues) có nhiều tiến trình chờ ở các hàng chờ có mức ưu tiên khác nhau.



- Trong hàng đợi Phản hồi, mức độ ưu tiên không cố định → Các tiến trình có thể được thăng cấp hoặc hạ cấp đến một hàng đợi khác.
 - Hàng chờ phản hồi có thể áp dụng các thuật toán lập lịch khác nhau ở các mức khác nhau.
- Lập lịch cho vi xử lý đa nhân (Multiprocessor Scheduling) có thể thực hiện bởi nhiều cách:
- Đa xử lý bất đối xứng (Asymmetric multiprocessing): chỉ có một bộ xử lý truy cập cấu trúc dữ liệu của hệ thống → không cần chia sẻ dữ liệu.
 - Đa xử lý đối xứng (Symmetric multiprocessing): mỗi bộ xử lý sẽ tự lập lịch (đây là phương pháp thông dụng hiện nay).
 - Bộ xử lý thân thuộc (Processor affinity): một tiến trình chạy trên một bộ xử lý thường sẽ tiếp tục chạy trên cùng bộ vi xử lý đó. Vì vậy bộ nhớ của bộ xử lý đó sẽ chứa dữ liệu cụ thể bộ xử lý cụ thể.
- Công thức Little (Little's Formula) có thể xác định thời gian đợi trung bình của mỗi tiến trình trong tất cả giải thuật lập lịch.
- $n = \lambda \times W$
 - Trong đó n = độ dài trung bình của hàng chờ; W = thời gian đợi trung bình trong hàng chờ; λ = tần suất trung bình tiến trình đến hàng đợi.
- Giả lập (Simulations) là nhưng mô hình được lập trình của một hệ thống máy tính với các đồng hồ khả biến.
- Được sử dụng để thu thập thông tin thống kê hiệu quả các giải thuật.
 - Chạy giả lập thì chính xác hơn các mô hình hàng đợi.
 - Mặc dù chính xác, chi phí vẫn cao và rủi ro.

Summary Chapter 05 of book “OS concepts 10th edition”

- Định thời CPU (scheduling) là tác vụ chọn tiến trình đang chờ từ hàng đợi “sẵn sàng” và phân bổ CPU cho nó. CPU được phân bổ cho tiến trình được chọn bởi bộ điều phối (dispatcher).
- Các thuật toán lập lịch có thể giữ quyền ưu tiên (preemptive), khi đó CPU có thể được thu hồi từ tiến trình đang giữ cho dù tiến trình vẫn đang cần CPU) hoặc không có quyền ưu tiên (khi đó một tiến trình phải tự nguyện từ bỏ quyền sử dụng CPU). Hầu như tất cả các hệ điều hành hiện đại đều giữ quyền ưu tiên.
- Các thuật toán lập lịch (Scheduling algorithms) có thể được đánh giá theo năm tiêu chí sau đây: (1) tận dụng CPU, (2) thông lượng (số tiến trình hoàn tất trong một đơn vị thời gian), (3) thời gian quay vòng, (4) thời gian chờ đợi và (5) thời gian đáp ứng.
- Lập lịch “Đến trước được phục vụ trước” (FCFS / First-come, first-served) là cách lập lịch đơn giản nhất, nhưng nó có thể để cho các tiến trình ngắn phải chờ các tiến trình rất dài.
- Lập lịch “Tác vụ ngắn nhất trước” (SJF / Shortest-job-first) được chứng minh là tối ưu, tạo ra thời gian chờ trung bình nhỏ nhất. Tuy nhiên, thực hiện SJF là rất khó bởi vì dự đoán thời gian sử dụng CPU là một việc rất khó khăn.
- Lập lịch “Xoay vòng” (RR / Round-robin) phân bổ CPU cho mỗi tiến trình trong một phần thời gian nhất định. Nếu tiến trình không từ bỏ CPU trước khi phần thời gian phân bổ cho nó hết hạn, tiến trình này sẽ bị thu hồi quyền sử dụng CPU và một tiến trình khác được lên lịch để chạy cũng với một phần thời gian tương tự.
- Lập lịch ưu tiên (Priority) gán cho mỗi tiến trình một mức độ ưu tiên và CPU được phân bổ đến tiến trình có độ ưu tiên cao nhất. Các tiến trình có cùng mức độ ưu tiên có thể được lên lịch tuần tự theo FCFS hoặc sử dụng lập lịch RR.

- Lập lịch hàng chờ đa mức phân chia các tiến trình thành nhiều hàng đợi riêng biệt được sắp xếp theo mức độ ưu tiên và bộ lập lịch tiến hành với các tiến trình trong hàng đợi ưu tiên cao nhất. Các thuật toán lập lịch khác nhau có thể được áp dụng trong mỗi hàng đợi.
- Hàng chờ phản hồi đa cấp tương tự như hàng chờ đa cấp, ngoại trừ, một tiến trình có thể di chuyển giữa các hàng chờ khác nhau.
- Bộ xử lý đa nhân có một hoặc nhiều CPU trên cùng một chip vật lý, và mỗi CPU có thể có nhiều hơn một tiểu trình phân cứng. Từ góc nhìn của hệ điều hành, mỗi tiểu trình phân cứng được xem như là một CPU luận lý.
- Cân bằng tải trên các hệ thống đa nhân sẽ cân bằng tải giữa các nhân CPU, mặc dù việc di chuyển các tiểu trình giữa các nhân để cân bằng tải có thể làm cho nội dung bộ nhớ cache bị vô hiệu và vì vậy có thể làm tăng thời gian truy cập bộ nhớ.
- Lập lịch thời gian thực mềm (Soft real-time) ưu tiên cho các tác vụ thời gian thực hơn các tiến trình không cần thời gian thực. Lập lịch thời gian thực cứng (Hard real-time) cung cấp khoảng thời gian đảm bảo cho tác vụ thời gian thực.
- Lập lịch “Thời gian thực tỷ lệ đơn điệu” (Rate-monotonic real-time) sử dụng một chính sách ưu tiên tĩnh và bộ lập lịch giữ quyền ưu tiên thu hồi CPU.
- Lập lịch “Tác vụ tới hạn” (EDF / Earliest-deadline-first) chỉ định các ưu tiên theo thời hạn chót. tiến trình nào càng gần thời hạn chót, mức độ ưu tiên càng cao; tiến trình nào càng xa thời hạn chót, mức độ ưu tiên càng thấp.
- Lập lịch theo tỷ lệ phân bổ (Proportional share) chia cho tất cả các ứng dụng T phần thời gian. Nếu một ứng dụng được phân bổ N phần thời gian, nó được đảm bảo có được tỉ lệ N/T của tổng thời gian xử lý.
- Linux sử dụng bộ lập lịch “hoàn toàn công bằng” (CFS / completely fair scheduler), chỉ định tỷ lệ thời gian sử dụng cho từng tác vụ. Tỷ lệ này dựa trên giá trị thời gian chạy giả lập (vruntime) đã gán cho mỗi tác vụ.
- Bộ lập lịch Windows giữ quyền ưu tiên (preemptive), và có 32 cấp độ ưu tiên để xác lập thứ tự định thời tiểu trình.
- Solaris xác định sáu lớp lập lịch duy nhất và được ánh xạ vào độ ưu tiên toàn cục. Tiểu trình chuyên tính toán (CPU-intensive) được gán mức độ ưu tiên thấp hơn (với phần thời gian dài hơn), và các tiểu trình I / O thường được chỉ định mức độ ưu tiên cao hơn (với phần thời gian ngắn hơn.)
- Mô hình hóa và mô phỏng có thể được sử dụng để đánh giá một thuật toán lập lịch CPU.