

Giúp SV ôn tập trọng tâm được tổng hợp và phân loại theo từng chủ đề kiến thức.

Nội dung này được thiết kế để giúp bạn hệ thống hóa lại kiến thức HK251. Lưu ý: Các em vẫn phải học và tìm hiểu toàn bộ nội dung môn học theo đề cương.

I. Tổng quan về Hệ Điều Hành & Cấu trúc Máy tính

- Mục tiêu và vai trò: Hiểu rõ các mục tiêu chính của HĐH (sử dụng hiệu quả phần cứng, giao diện người dùng...) và vai trò của Device Controller trong việc giao tiếp thiết bị.
- Chế độ hoạt động (Modes): Phân biệt rõ User Mode và Kernel Mode. Cần biết các thao tác nào (như I/O, tính toán số học, system calls) được thực hiện ở chế độ nào.
- Cấu trúc HĐH: Nắm vững ưu/nhược điểm của các cấu trúc:
 - Đơn khối (Monolithic): Hiệu suất, khả năng bảo trì.
 - Phân lớp (Layered): Lợi ích trong việc bảo trì và gỡ lỗi.
 - Modular (Mô-đul):
 - Microservice:
- System Calls: Cách truyền tham số cho system calls (qua thanh ghi, stack, v.v.).
- Cơ chế đa nhiệm (Multitasking) vs Đa chương (Multiprogramming).
- Time-sharing.

II. Quản lý Process (Process Management)

- Vòng đời quá trình:
 - Các trạng thái của process: Ready, Running, Waiting (Blocked). Hiểu rõ điều kiện chuyển đổi giữa các trạng thái này (do I/O, do hết thời gian CPU, do ngắt).
 - Process Control Block (PCB): Các thành phần chứa trong PCB (PC, thanh ghi, trạng thái...).
- Tạo quá trình (Fork):
 - Cơ chế hoạt động của lệnh fork().
 - Phân tích mã nguồn C: Dự đoán output của biến số khi fork() được gọi, mối quan hệ cha-con.
- Giao tiếp IPC:
 - Các cơ chế: Shared Memory vs. Message Passing.
 - Hiểu khi nào nên dùng cơ chế nào (dựa trên tốc độ, lượng dữ liệu).

III. Luồng (Threads) & Đa luồng

- Khái niệm: Lợi ích của việc sử dụng thread so với process.
- Phân loại: Phân biệt User-level thread và Kernel-level thread (ai quản lý, khả năng chia sẻ tài nguyên).
- Mô hình ánh xạ: One-to-One, Many-to-One, Many-to-Many.
- Chia sẻ tài nguyên: Các thread trong cùng process chia sẻ những gì (Code, Data, Heap) và sở hữu riêng cái gì (Stack, Registers).

IV. Định thời CPU (CPU Scheduling)

- Các giải thuật định thời: Cần nắm vững nguyên lý hoạt động và cách tính toán của:
 - FCFS (First-Come, First-Served).
 - SJF (Shortest Job First) & SRTF (Shortest Remaining Time First).
 - Round Robin (RR): Khái niệm Quantum time.
 - Priority Scheduling (Preemptive & Non-preemptive).
 - Multi-level Queue (MLQ).
- Bài tập tính toán: Luyện tập tính Thời gian chờ trung bình (Average Waiting Time) và Thời gian hoàn thành/xoay vòng (Turnaround Time) dựa trên bảng dữ liệu Process, Arrival Time, Burst Time cho trước.

V. Đồng bộ hóa (Synchronization) & Deadlock

- Cơ chế đồng bộ:
 - Critical Section (Vùng tranh chấp): Điều kiện loại trừ tương hỗ (Mutual Exclusion).
 - Semaphore: Phân biệt Binary Semaphore và Counting Semaphore. Cách hoạt động của wait() và signal().
 - Mutex: Sự khác biệt giữa Mutex và Semaphore.
 - Test-and-Set: Bản chất phần cứng và tác dụng.
- Các bài toán kinh điển:
 - Dining Philosophers (Triết gia ăn tối): Nguyên nhân gây Deadlock.
 - Producer-Consumer: Cách phối hợp hoạt động.
- Vấn đề tiềm ẩn: Race condition, Deadlock, Starvation.

VI. Quản lý Bộ nhớ (Memory Management)

- Địa chỉ: Phân biệt địa chỉ luận lý (Logical/Virtual) và địa chỉ vật lý (Physical).
- Phân trang (Paging):
 - Cấu trúc địa chỉ: Page number, Offset.
 - Bảng phân trang (Page Table) và vai trò của TLB (Translation Lookaside Buffer) trong việc tăng tốc dịch địa chỉ.
 - Page Fault: Khi nào xảy ra? Hệ thống xử lý như thế nào?
- Thay thế trang (Page Replacement):
 - Các giải thuật: FIFO, LRU (Least Recently Used), OPT (Optimal).
 - Bài tập: Tính số lượng lỗi trang (Page faults) dựa trên chuỗi tham chiếu (Reference String) và số lượng Frame cho trước.
 - So sánh hiệu năng giữa các giải thuật.
- Phân đoạn (Segmentation):
 - Cách dịch địa chỉ: <Segment-number, Offset>.
 - Bài tập kiểm tra địa chỉ hợp lệ dựa trên bảng Segment (Base, Limit).
- Phân mảnh: Phân mảnh nội (Internal) vs Phân mảnh ngoại (External) trong các chiến lược cấp phát.

VII. Hệ thống tập tin & I/O (File System)

- Phương pháp cấp phát:
 - Liên tục (Contiguous Allocation).
 - Danh sách liên kết (Linked Allocation).
 - Chỉ mục (Indexed Allocation).
 - So sánh về tốc độ truy cập ngẫu nhiên và lãng phí không gian.
- File System ảo: Khái niệm procfs.
- I/O: Lợi ích của việc thực thi đồng thời CPU và I/O.

VIII. Kiến thức mở rộng & Tính toán khác

- Định luật Amdahl: Bài tập tính Speedup của hệ thống Multicore dựa trên tỉ lệ phần trăm tuần tự (serial) và song song (parallel).

Lời khuyên ôn tập:

1. Ưu tiên bài tập tính toán: Đề thi có tỷ lệ đáng kể các câu hỏi yêu cầu tính toán (Lập lịch CPU, Page Fault, Speedup, Địa chỉ bộ nhớ). Hãy chắc chắn bạn cầm máy tính bấm nhanh các dạng này.
2. Đọc kỹ Code C: Có câu hỏi yêu cầu phân tích luồng chạy của fork(), hãy vẽ sơ đồ cây process để không bị nhầm lẫn giữa cha và con.
3. So sánh giải thuật: Đừng chỉ học thuộc lòng, hãy lập bảng so sánh ưu nhược điểm (ví dụ: RR tốt cho time-sharing, SJF tối ưu waiting time nhưng dễ gây starvation).

Sửa lần cuối: Thứ Bảy, 29 tháng 11 2025, 3:08 PM

Tổ kỹ thuật / Technician

Email : ddthu@hcmut.edu.vn

Quý Thầy/Cô chưa có tài khoản (hoặc quên mật khẩu) nhà trường vui lòng liên hệ Trung tâm Dữ liệu & Công nghệ Thông tin, phòng 109 nhà A5 để được hỗ trợ.

(For HCMUT account, please contact to : Data and Information Technology Center)

Email : dl-cntt@hcmut.edu.vn

ĐT (Tel.) : (84-8) 38647256 - 7200

Copyright 2007-2023 BKEL - Phát triển dựa trên Moodle