

**ĐỀ CHÍNH THỨC**  
(Đề có 4 trang)

**ĐỀ THI GIỮA KỲ**  
**Môn: Nguyên lý hệ điều hành**  
Thời gian làm bài: 50 phút, không kể thời gian phát đề

\* Thí sinh được sử dụng tài liệu giấy theo quy định;

Mã đề thi: B

**Câu 1:** Stack của luồng dùng để làm gì?

- A. Chứa biến toàn cục
- B. Chứa biến cục bộ và thông tin truyền tham số
- C. Chứa mã nguồn
- D. Chứa tất cả dữ liệu của tiến trình

**Câu 2:** Trong giải thuật người quản lý nhà băng (Banker's Algorithm), trạng thái an toàn là:

- A. Trạng thái hệ thống có thể cấp phát tài nguyên theo thứ tự nào đó
- B. Trạng thái không có tài nguyên
- C. Trạng thái không có tiến trình
- D. Trạng thái có deadlock

**Câu 3:** Trong đa luồng, các luồng của cùng một tiến trình chia sẻ những gì?

- A. Không chia sẻ gì cả
- B. Chia sẻ mã nguồn, dữ liệu và tài nguyên hệ thống
- C. Chỉ chia sẻ dữ liệu
- D. Chỉ chia sẻ mã nguồn

**Câu 4:** Producer-Consumer với buffer 5 phần tử:

empty = 5 (semaphore)

full = 0 (semaphore)

mutex = 1 (binary semaphore)

Sau khi Producer thực hiện 3 lần đưa dữ liệu vào buffer, giá trị của empty là:

- A. 2
- B. 0
- C. 3
- D. 5

**Câu 5:** Hệ thống truyền thông điệp có đặc điểm gì?

- A. Không an toàn
- B. Không cần hệ điều hành hỗ trợ
- C. Chỉ hoạt động trong một máy
- D. Các tiến trình trao đổi thông điệp thông qua kernel

**Câu 6:** Ưu điểm của mô hình đa xử lý đối xứng là:

- A. Tốn ít tài nguyên
- B. Tăng khả năng chịu lỗi và cân bằng tải tốt
- C. Dễ lập trình hơn
- D. Không cần đồng bộ hóa

**Câu 7:** Truyền thông non-blocking thích hợp trong trường hợp nào?

- A. Khi thông điệp không quan trọng
- B. Khi hệ thống yêu cầu độ trễ thấp và hiệu suất cao
- C. Cần đảm bảo thông điệp được nhận ngay lập tức
- D. Khi cần đồng bộ chặt chẽ

**Câu 8:** Truyền thông điệp blocking (đồng bộ) có nghĩa là gì?

- A. Thông điệp bị hủy bỏ

- B. Tiến trình gửi phải chờ cho đến khi thông điệp được nhận
- C. Không có quá trình truyền thông
- D. Tiến trình gửi không cần chờ đợi

**Câu 9:** File thực thi (executable file) được tạo ra sau bước nào?

- A. Link
- B. Debug
- C. Compile

D. Load

**Câu 10:** Cho mã giả của bài toán Dining Philosophers:

do {

$P(chopstick[i]);$

$P(chopstick[(i+1)\%5]);$

// eat

$V(chopstick[i]);$

$V(chopstick[(i+1)\%5]);$

// think

} while (true);

Vấn đề có thể xảy ra với mã này là:

- A. Không có vấn đề gì
- B. Chỉ một triết gia được ăn
- C. Deadlock khi tất cả triết gia cầm đũa trái
- D. Không có triết gia nào có thể ăn

**Câu 11:** Windows sử dụng cơ chế nào để tạo tiến trình mới?

- A. CreateProcess() tạo tiến trình mới hoàn toàn
- B. Copy toàn bộ tiến trình cha
- C. Không có cơ chế tạo tiến trình
- D. Fork() như Unix

**Câu 12:** Một chương trình có thể tạo ra bao nhiêu tiến trình cùng một lúc?

- A. Nhiều tiến trình
- B. Không tạo được tiến trình
- C. Chỉ một tiến trình
- D. Hai tiến trình

**Câu 13:** Lập lịch cho luồng được thực hiện ở đâu?

- A. Chỉ ở mức kernel
- B. Có thể ở cả mức user và kernel
- C. Chỉ ở mức user
- D. Không cần lập lịch

**Câu 14:**

Điều độ không độc quyền là gì?

- A. Chỉ chạy một tiến trình
- B. Không có scheduling
- C. CPU có thể bị thu hồi từ tiến trình đang chạy
- D. CPU không thể bị thu hồi từ tiến trình đang chạy

**Câu 15:** Chuyển đổi ngữ cảnh (Context switching) là gì?

- A. Chuyển đổi giữa các chương trình
- B. Chuyển đổi thiết bị vào/ra
- C. Chuyển đổi bộ nhớ
- D. Chuyển đổi trạng thái của CPU giữa các tiến trình khác nhau

**Câu 16:** Phát hiện deadlock thường được thực hiện:

- A. Không bao giờ
- B. Định kỳ
- C. Liên tục
- D. Chỉ khi khởi động hệ thống

**Câu 17:** Tiêu chí đánh giá thuật toán điều phối CPU dựa trên thông tin nào?

- A. Xét đến mức độ sử dụng CPU và thời gian chờ

- B. Xét số tiến trình hoàn thành và thời gian đáp ứng  
C. Xét thời gian chờ đợi  
D. Xét mức độ sử dụng CPU, số tiến trình hoàn thành, thời gian chờ, thời gian đáp ứng

**Câu 18:** Job scheduler (bộ lập lịch công việc) quyết định điều gì trong hệ thống?

- A. Tốc độ CPU  
B. Kích thước bộ nhớ RAM  
C. Số lượng thiết bị I/O  
D. Mức độ đa chương trình

**Câu 19:** PCB (Process Control Block) là gì?

- A. Một thiết bị phần cứng  
B. Một phần của bộ nhớ RAM  
C. Một chương trình máy tính  
D. Một cấu trúc dữ liệu chứa thông tin về một tiến trình

**Câu 20:** Semaphore khác Mutex ở điểm nào?

- A. Semaphore không có hàng đợi  
B. Semaphore không thể dùng cho đồng bộ hóa  
C. Semaphore có thể có giá trị lớn hơn 1  
D. Semaphore chỉ dùng cho một tiến trình

**Câu 21:** Theo quan điểm của người sử dụng thông thường, yêu cầu quan trọng nhất đối với hệ điều hành là gì?

- A. Dễ sử dụng và giao diện thân thiện  
B. Hiệu suất cao và tối ưu tài nguyên  
C. Khả năng bảo mật cao  
D. Quản lý tài nguyên hiệu quả

**Câu 22:** Xét theo số chương trình có thể thực hiện đồng thời, hệ điều hành được chia thành:

- A. Đơn nhiệm và đa nhiệm  
B. Real-time và không real-time  
C. Một người dùng và nhiều người dùng  
D. Batch processing và time sharing

**Câu 23:** Theo góc độ kỹ thuật, đâu là yêu cầu quan trọng của hệ điều hành?

- A. Quản lý tài nguyên hiệu quả và tối ưu  
B. Chi phí thấp  
C. Dễ cài đặt  
D. Giao diện người dùng đẹp

**Câu 24:**

Thành phần nào sau đây KHÔNG thuộc kiến trúc Von Neumann?

- A. Thiết bị Input/Output  
B. Card đồ họa (GPU)  
C. Bộ xử lý trung tâm (CPU)  
D. Bộ nhớ chính (Main Memory)

**Câu 25:** Thành phần nào tương tác trực tiếp với phần cứng để quản lý tài nguyên hệ thống?

- A. Database System  
B. Users  
C. Operating System  
D. Application Programs

**Câu 26:** Thuật toán SJF (Shortest Job First):

Cho 4 tiến trình có thời gian hoạt động là: P1: 6ms, P2: 8ms, P3: 7ms, P4: 3ms

Các tiến trình đều đến tại thời điểm 0.

Thời gian chờ trung bình là:

- a) 9 ms  
b) 8 ms  
c) 10 ms  
d) 7 ms

**Câu 27:** Lập lịch theo độ ưu tiên (Priority Scheduling - số càng nhỏ độ ưu tiên càng cao):

Cho 3 tiến trình với độ ưu tiên và thời gian hoạt động:

P1: độ ưu tiên = 3, thời gian hoạt động = 10

P2: độ ưu tiên = 1, thời gian hoạt động = 1

P3: độ ưu tiên = 2, thời gian hoạt động = 2

Thứ tự thực hiện các tiến trình là:

- a) P2, P3, P1  
b) P3, P2, P1  
c) P2, P1, P3  
d) P1, P2, P3

**Câu 28:** Cho hệ thống với 4 tiến trình và 2 tài nguyên:

Max:	A B	Allocation:	A B	Available:	A B
P0	4 3	2 1	3 1		
P1	3 3	1 1			
P2	5 4	2 1			
P3	2 2	0 0			

Sau khi P2 hoàn thành và giải phóng tài nguyên, Available sẽ là:

- A. (1,0)  
B. (5,2)  
C. (3,1)  
D. (2,1)

**Câu 29:** Dining Philosophers với: chopstick[5] = {1,1,1,1,1} (semaphores)

Khi 2 triết gia liên kề cùng cầm đũa trái, số triết gia tối đa có thể ăn cùng lúc là:

- A. 1  
B. 5  
C. 2  
D. 3

**Câu 30:** Cho hệ thống với: 4 tiến trình: P1, P2, P3, P4 và 3 loại tài nguyên: R1 (2), R2 (2), R3 (2)

Trạng thái: P1: giữ R1(1), yêu cầu R2(1); P2: giữ R2(1), yêu cầu R3(1); P3: giữ R3(1), chờ R1(1)

P4: giữ R1(1), R2(1)

Hệ thống có deadlock không?

- A. Có  
B. Cần thêm thông tin  
C. Không  
D. Không thể xác định

----- HẾT -----



## PHẦN TỰ LUẬN

**Câu 1 (10 đ)** Cho 3 tiến trình (hoặc luồng) P1, P2, P3 thực hiện song song như bên dưới. Giả thiết rằng S1, S2 là các biến Semaphores được khởi tạo giá trị bằng 0. Các biến a, b, c, X kiểu nguyên

(Các) giá trị có thể của biến X khi cả

3 tiến trình/luồng cùng kết thúc là:

- X = .....
- X = .....
- X = .....
- X = .....

Tiến trình P1	Tiến trình P2	Tiến trình P3
$a = 2$	$b = 3$	$c = 4$
$P(S1)$	$V(S1)$	$V(S1)$
$P(S1)$	$b = b * 2$	$P(S2)$
$a = b + c$	$V(S2)$	$P(S2)$
$V(S2)$		$X = b * a$

Hãy giải thích ngắn gọn

---



---



---



---



---

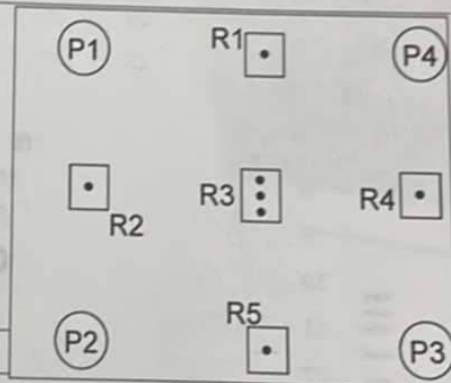
**Câu 2 (15đ):** Cho hệ thống gồm 4 tiến trình P1, P2, P3, P4 và 5 tài nguyên R1, R2, R3, R4, R5 như hình vẽ.

1. Vẽ vào hình bên đồ thị phân phối tài nguyên cho hệ thống trên khi:

- Tiến trình P1 đang sử dụng các tài nguyên R1, R2, R3;
- P2 đang sử dụng R3 và yêu cầu R2, R5;
- P3 đang sử dụng R5 và yêu cầu R3 và R4;
- P4 đang sử dụng R3, R4 và yêu cầu R1.

2. Hệ thống đang bế tắc (Đúng/sai)?.....

Giải thích (ngắn gọn)



3. Khi P1 giải phóng R3, hệ thống cung cấp tài nguyên này cho P3 ngay lập tức thì

- Hệ thống có an toàn không (Đúng/sai)? .....
- Giải thích!

---



---



---



---



---