

## **Câu hỏi ôn tập**

### **1. Cho biết mục đích của system call**

Cho phép chương trình của người dùng yêu cầu dịch vụ của hệ điều hành

### **2. Các hoạt động chính của hệ điều hành trong quản lý quy trình**

Tạo và xóa cả process của người dùng và hệ thống.

Đình chỉ và tiếp tục các process.

Cung cấp cơ chế đồng bộ hóa process.

Cung cấp cơ chế giao tiếp process.

Cung cấp cơ chế xử lý tình trạng khóa chết (deadlock).

### **3. Các hoạt động chính của hệ điều hành trong quản lý bộ nhớ**

Theo dõi các phần của bộ nhớ hiện đang được sử dụng và được sử dụng bởi ai.

Quyết định quá trình và dữ liệu nào sẽ được di chuyển vào và ra khỏi bộ nhớ.

Phân bổ và giải phóng không gian bộ nhớ khi cần thiết.

### **4. 5 chức năng chính của hđh trong quản lý hệ thống tập tin – quản lý file – quản lý tập tin**

Tạo và xóa tập tin

Quản lý thư mục

Thao tác tập tin và thư mục

Phân quyền truy cập

Quản lý đĩa và không gian lưu trữ

### **5. Các hoạt động chính của hđh trong quản lý thiết bị lưu trữ (mass - storage)**

Cấp phát không gian lưu trữ

Định thời đĩa (Disk scheduling)

### **6. Địa chỉ logic hay địa chỉ ảo là địa chỉ**

Được tạo ra bởi tính toán của CPU

## **7. Địa chỉ vật lý là địa chỉ**

Được tạo ra bởi đơn vị quản lý bộ nhớ (MMU)

## **8. HDH sử dụng cặp con trỏ base-limit trong không gian bộ nhớ logic của một quy trình để**

Qui định vùng nhớ mà một quy trình có thể truy xuất

## **9. Nguyên lý, đặc điểm cấp phát bộ nhớ kiểu MFT**

Bộ nhớ được chia thành nhiều phân vùng có kích thước cố định

Mỗi process sử dụng một phân vùng và sau khi kết thúc sẽ trả lại phân vùng đó cho hệ thống

## **10. Cấp phát bộ nhớ theo kiểu MVT**

Phần bộ nhớ khả dụng gọi là hole được cấp cho một phần vừa đủ cho quy trình, phần còn lại tạo thành một hole khác

## **11. Bộ định thời dài hạn có nhiệm vụ**

Chọn các process từ process pool để đưa vào bộ nhớ

## **12. Bộ định thời ngắn hạn có nhiệm vụ**

Chọn từ trong số các quá trình đã sẵn sàng để thực thi và cấp phát CPU cho một trong số chúng.

## **13. Trên phần lớn Window và Unix thì bộ định thời dài hạn**

Không có bộ định thời dài hạn

## **14. Trên một số hdh chia sẻ thời gian người ta sử dụng bộ định thời trung gian làm gì**

Thực hiện swapping process

## **15. Khi nào process cha kết thúc process con**

Vượt quá tài nguyên được cấp

Nhiệm vụ không còn cần thiết

Process cha đã thoát

## **16. Khi process cha tạo ra một process con thì có thể**

Cha và con chia sẻ tất cả các tài nguyên

Con chia sẻ một tập hợp con của các tài nguyên của cha

Cha và con không chia sẻ tài nguyên nào

### **17. Thư viện Win32 là**

Thư viện cung cấp trên hệ thống window ở mức kernel

### **18. Thư viện Pthread là thread mở rộng của chuẩn POSIX là thư viện được cấp ở mức**

Cả kernel và user

### **19. Quan hệ giữa kernel thread và user thread trên hhd linux là**

One-to-one

### **20. Hai kỹ thuật tạo thread trong java**

Một cách tiếp cận là tạo một lớp mới được phát sinh từ lớp Thread và ghi đè phương thức run().

Một cách tiếp cận khác và phổ biến hơn là định nghĩa một lớp thực hiện interface Runnable.

### **21. Chức năng thư viện thread**

Thư viện thread cung cấp các công cụ để tạo, quản lý và đồng bộ hóa các luồng thực thi độc lập trong một chương trình đa nhiệm.

### **22. Cho biết 2 phương pháp để thực hiện 1 thư viện Thread**

Cung cấp một thư viện hoàn toàn ở user space mà không có hỗ trợ từ kernel. Tất cả các mã và cấu trúc dữ liệu cho thư viện tồn tại trong user space. Điều này có nghĩa là gọi một hàm trong thư viện dẫn đến một lời gọi hàm cục bộ trong user space và không phải là một system call.

Cách thực hiện thư viện luồng cấp kernel được hỗ trợ trực tiếp bởi hệ điều hành. Mã và cấu trúc dữ liệu cho thư viện tồn tại trong không gian kernel. Gọi một hàm trong API của thư viện thường dẫn đến một lệnh gọi hệ thống đến kernel

### **23. Cho biết 3 chức năng chính của hệ điều hành**

Một chương trình quản lý phần cứng máy tính

Cung cấp cơ sở cho các chương trình ứng dụng

Hoạt động như một trung gian giữa người dùng máy tính và phần cứng.

**24. Chỉ thị nào sau đây có thể thực hiện ở user mode**

Access I/O device

**25. Thế nào là thiết bị lưu trữ nonvolatile**

Là một loại thiết bị lưu trữ dữ liệu có khả năng giữ lại dữ liệu mà không cần nguồn điện liên tục

Một số ví dụ về thiết bị lưu trữ nonvolatile bao gồm ổ đĩa cứng (HDD), ổ đĩa SSD, thẻ nhớ, đĩa CD/DVD và băng từ

**26. Cho biết các ưu điểm của MultiProcessor system**

Tăng thời lượng xử lý – thực hiện nhiều công việc trong thời gian ngắn;  
Tiết kiệm tài nguyên hệ thống – vì có thể chia sẻ các thiết bị ngoại vi, bộ nhớ và nguồn điện;  
Tăng độ tin cậy – nếu các chức năng được phân phối đúng đắn giữa các bộ xử lý, thì sự cố của một bộ xử lý sẽ không làm đình trệ hệ thống, chỉ làm nó chậm lại.

**27. Trong một hệ thống Asymmetric multiprocessing**

Một processor định thời và cấp phát công việc

**28. Cho biết các trạng thái có thể có của 1 process**

New: Quá trình đang được tạo ra.

Running: Các lệnh đang được thực thi.

Waiting: Quá trình đang chờ đợi một sự kiện nào đó xảy ra (như hoàn thành I/O hoặc nhận tín hiệu).

Ready: Quá trình đang chờ được phân bổ cho một bộ xử lý.

Terminated: Quá trình đã hoàn thành thực thi.

**29. Điều gì là vấn đề chính để làm nên sự khác biệt của bộ định thời dài hạn và bộ định thời ngắn hạn**

Sự phân biệt chính giữa hai bộ lập lịch này nằm ở tần suất thực thi.

**30. Một I/O bound process là gì và 1 CPU bound process là gì?**

Một I/O bound process là một tiến trình tốn nhiều thời gian để thực hiện các hoạt động đầu vào và đầu ra

CPU bound process là một tiến trình tốn nhiều thời gian để thực hiện các hoạt động tính toán trên CPU.

**31. Symmetric Multiprocessing là gì**

Tất cả process có quan hệ ngang hàng

**32. Cho biết phân mảnh ngoài (external fragmentation) khi cấp phát bộ nhớ là gì**

Hiện tượng phân mảnh bên ngoài xảy ra khi tổng không gian nhớ đủ để đáp ứng yêu cầu, nhưng các không gian khả dụng không liên tục; bộ nhớ bị phân mảnh thành một số lỗ nhỏ.

**33. Cho biết phân mảnh trong (internal fragmentation) khi cấp phát bộ nhớ là gì**

Nếu bộ nhớ được phân bổ cho một quy trình lớn hơn so với bộ nhớ yêu cầu, thì bộ giữa hai số này là internal fragmentation, là bộ nhớ nằm trong một phân vùng nhưng không được sử dụng.

**34. Để giải quyết phân mảnh ngoài (external fragmentation) của bộ nhớ, hệ điều hành sử dụng phương pháp Compaction. Cho biết ý tưởng chính của phương pháp này và điều kiện để thực hiện nó?**

Một giải pháp cho vấn đề external fragmentation bên ngoài là nén. Mục tiêu là xáo trộn nội dung bộ nhớ để đặt tất cả các bộ nhớ trống cùng với nhau thành một khối lớn. Tuy nhiên, việc nén không phải lúc nào cũng khả thi.

Điều kiện: phải có một trình quản lý bộ nhớ (memory manager) có khả năng tái sử dụng các vùng bộ nhớ đã được giải phóng và sắp xếp chúng liên tiếp nhau

**35. Hệ điều hành hỗ trợ cơ chế hợp tác (cooperation) giữa các process. Cho biết vài lý do cũng như sự hữu ích khi các process liên kết với nhau?**

Chia sẻ thông tin. Vì nhiều người dùng có thể quan tâm đến cùng một thông tin (một tập tin chia sẻ...)

Tăng tốc tính toán. Nếu chúng ta muốn một nhiệm vụ cụ thể chạy nhanh hơn, chúng ta phải chia nó thành các tác vụ con, mỗi tác vụ sẽ được thực thi song song với các tác vụ khác.

Tính mô-đun. Để xây dựng hệ thống theo kiểu mô-đun - chia các chức năng của hệ thống thành các quy trình hoặc luồng riêng biệt.

Tiện lợi. Người dùng có thể làm việc trên nhiều nhiệm vụ cùng một lúc.

**36. Cho biết các điều kiện cần thiết để các process có thể hợp tác (cooperation) với nhau theo cách chia sẻ bộ nhớ (share-memory)**

Bộ nhớ chia sẻ yêu cầu hai hoặc nhiều tiến trình đồng ý với nhau.

Các tiến trình cũng có trách nhiệm đảm bảo rằng chúng không ghi vào cùng một vị trí cùng một lúc.

**37. Cho biết RPC (Remote procedure call) được sử dụng khi nào và ý nghĩa của nó là gì**

RPC (Remote Procedure Call) được sử dụng khi cần thiết để gọi một hàm hoặc thủ tục trên một máy tính từ một máy tính khác trong mạng. RPC cho phép các ứng dụng chạy trên các máy tính khác nhau giao tiếp với nhau như thể chúng đang chạy trên cùng một máy tính.

Ý nghĩa của RPC là tạo ra một phương thức gọi hàm ở xa, giúp cho các ứng dụng có thể tương tác với nhau thông qua mạng máy tính một cách dễ dàng và hiệu quả. RPC đóng vai trò quan trọng trong các ứng dụng phân tán, nơi mà các thành phần khác nhau của ứng dụng chạy trên các máy tính khác nhau trong mạng.

**38. Hàng đợi sẵn sàng (ready queue) là hàng đợi**

Các quá trình đang lưu trú trong bộ nhớ chính và sẵn sàng và chờ để thực thi được giữ trên một danh sách gọi là hàng đợi sẵn sàng

**39. Vùng đệm nhập xuất (I/O buffering) là gì và mục đích của vùng này**

Bộ đệm (Buffer) là một khu vực bộ nhớ lưu trữ dữ liệu trong quá trình chuyển đổi giữa hai thiết bị hoặc giữa một thiết bị và một ứng dụng.

Việc đệm được thực hiện với ba mục đích:

- Để giải quyết sự khác biệt về tốc độ của thiết bị.
- Để giải quyết sự khác biệt về kích thước chuyển đổi của thiết bị.
- Để duy trì "các nguyên tắc sao chép": sao chép dữ liệu vào bộ đệm nhân hệ điều hành trước khi trả lại quyền điều khiển cho ứng dụng.

**40. Cho biết nội dung của DMA command block**

Địa chỉ bộ nhớ nguồn (Source Address): Đây là địa chỉ của bộ nhớ mà dữ liệu sẽ được sao chép từ đó bởi hoạt động DMA.

Địa chỉ bộ nhớ đích (Destination Address): Đây là địa chỉ của bộ nhớ mà dữ liệu sẽ được sao chép đến bởi hoạt động DMA.

Độ dài dữ liệu (Data Length): Đây là số lượng byte của dữ liệu sẽ được chuyển đổi bởi hoạt động DMA.

Thông số cấu hình (Configuration Parameters): Đây là các thông số cấu hình khác, bao gồm chế độ hoạt động DMA (single block hoặc multi-block), độ rộng bus, tốc độ truyền dữ liệu, và các thông số khác liên quan đến hoạt động DMA.

Địa chỉ bộ điều khiển DMA (DMA Controller Address): Đây là địa chỉ của bộ điều khiển DMA, nơi mà DMACB sẽ được truyền để bắt đầu hoạt động DMA.

#### **41. Hoạt động trong cơ chế DMA**

DMA command block được ghi vào bộ nhớ

CPU viết địa chỉ của khối lệnh DMA này vào bộ điều khiển DMA, sau đó tiếp tục với các công việc khác

#### **42. Khi controller đặt một tín hiệu interrupt-request line thì CPU sẽ thực hiện**

Lưu lại trạng thái hiện tại và chuyển điều khiển đến đoạn thủ tục

#### **43. Khi một process cha tạo một process con mới thì không gian địa chỉ của hai process này có thể**

Quy trình con là một bản sao của quy trình cha (nó có chương trình và dữ liệu giống với quy trình cha).

Quy trình con có một chương trình mới được tải vào nó

#### **44. Race condition là gì**

Race condition là tình trạng xảy ra khi nhiều quy trình hoặc luồng thực thi trong chương trình cùng truy cập và thay đổi dữ liệu chung một cách đồng thời và không có sự điều khiển đúng đắn nào, dẫn đến kết quả của chương trình không đáng tin cậy và không thể dự đoán trước được

#### **45. Trong một hệ thống có các process tương tác với nhau (cooperating process), hãy cho biết làm thế nào để ngăn chặn race condition xảy ra**

Mutual exclusion (Đồng quyền): Nếu quá trình  $P_i$  đang thực thi trong critical section của nó, thì không có quá trình nào khác có thể thực thi trong critical section của chúng.

Progress (Quy trình): Nếu không có quá trình nào đang thực thi trong critical section của nó và một số quá trình muốn vào critical section của chúng, thì chỉ có những quá trình đang không thực thi trong phần còn lại của chúng mới có thể tham gia vào quyết định về việc quá trình nào sẽ vào critical section của mình tiếp theo, và việc lựa chọn này không thể được trì hoãn vô thời hạn.

Bounded waiting (Giới hạn thời gian chờ đợi): Tồn tại một giới hạn, hoặc giới hạn, về số lần mà các quá trình khác được phép vào critical section của họ sau khi một quá trình đã yêu cầu vào critical section của mình và trước khi yêu cầu đó được chấp nhận.

#### **46. Critical section của một process là vùng mà ở đó**

Process có thể thay đổi dữ liệu của nó

#### **47. Cho biết ý nghĩa các thành phần trong một đồ thị cấp phát tài nguyên (resource-allocation graph)**

Đỉnh của đồ thị: Gồm hai loại: tập gồm tất cả quy trình trong hệ thống (kí hiệu là  $P$ ), tập gồm tất cả các loại tài nguyên (kí hiệu là  $R$ ) trong hệ thống

Request Edge – Cạnh yêu cầu: Thể hiện quy trình  $P_i$  đang yêu cầu tài nguyên  $R_j$

Assignment Edge – Cạnh gán: Thể hiện tài nguyên  $R_j$  đã được cấp cho quy trình  $P_i$

#### **48. Nếu một process có nhiều thread thì các thread này**

Có cùng đoạn mã chương trình, dữ liệu và tài nguyên hệ thống

#### **49. Cho biết ý nghĩa vắn tắt của các mô hình quan hệ giữa user thread và kernel thread**

One – to – one: Ánh xạ mỗi luồng người dùng vào một luồng nhân hệ điều hành.

Many – to – one: Ánh xạ nhiều luồng cấp người dùng vào một luồng nhân hệ điều hành.

Many – to – many: Đa luồng nhiều luồng cấp người dùng đến một số luồng nhân hệ điều hành nhỏ hơn hoặc bằng.



**50. Preemptive kernel là phương pháp**

Cho phép một process đang chạy ở kernel mode ngừng và CPU phục vụ process khác.

**51. Nonpreemptive kernel là phương pháp**

Kernel-mode process sẽ thực hiện cho tới khi nó tự động giải phóng CPU

**52. Cho biết điều kiện để có loại trừ tương hỗ (mutual exclusion) trong vấn đề deadlock là gì**

Một quy trình không bao giờ cần phải đợi một tài nguyên có thể chia sẻ.

**53. Cho biết lợi ích của các nhà phát triển hđh thiết kế các hệ thống con nhập xuất (I/O subsystem) độc lập vs phần cứng**

Họ có thể thiết kế các thiết bị mới để tương thích với giao diện điều khiển chủ của máy chủ hiện có.

Viết các trình điều khiển thiết bị để giao tiếp giữa phần cứng mới và các hệ điều hành phổ biến.

**54. Các phương pháp xử lý deadlock của hệ điều hành**

Sử dụng một giao thức để ngăn hoặc tránh deadlock, đảm bảo không bao giờ rơi vào deadlock

Cho phép hệ thống rơi vào deadlock phát hiện và khôi phục

Bỏ qua vấn đề deadlock

**55. Cho 2 ví dụ minh họa chương trình nhiều thread hoạt động tốt hơn so với chương trình một thread**

Word có thể vừa in tài liệu sử dụng một thread nền, vừa cùng lúc chạy một thread khác nhận dữ liệu vào từ người dùng để gõ một văn bản mới

Xử lý hình ảnh và âm thanh trong ứng dụng đa phương tiện trên máy tính