

1. Why do modern operating systems need interrupts?

Optimize CPU usage.

Support multitasking by handling tasks in parallel.

Reduce system latency when responding to peripheral devices.

2. When is an interrupt better than polling for checking new data?

A fast response to an event is needed (e.g., keyboard, network).

Avoiding CPU resource wastage due to continuous checking (as in polling).

The system has multiple peripheral devices operating simultaneously.

Polling is only suitable when data needs to be checked frequently with minimal delay, such as checking the CPU cache.

3. When is Programmed I/O better than DMA?

Small and infrequent data transfers (keyboard, mouse, sensors).

Simple hardware that does not support DMA.

No need for high-speed data transfer or when the CPU is not overloaded.

DMA is more suitable for large data transfers (hard drives, network cards).

4. What type of processes make multiprogramming better than time-sharing?

Multiprogramming is better when:

Processes require minimal user interaction.

CPU utilization needs to be optimized by overlapping I/O wait times.

The system is designed for batch processing.

Time-sharing is better when:

Quick user response is required.

CPU time is divided into time slices to ensure fair process execution.

5. Can user applications access the operating system's memory? Explain briefly.

No, they cannot directly access it.

The operating system uses protected memory regions to prevent user processes from corrupting system data.

Access is only possible through system calls (API requests).

6. List three CPU scheduling algorithms.

First-Come, First-Served (FCFS): Tiến trình đến trước sẽ được xử lý trước.

Shortest Job First (SJF): Tiến trình có thời gian xử lý ngắn nhất sẽ được xử lý trước.

Priority Scheduling: Tiến trình có mức độ ưu tiên cao nhất sẽ được xử lý trước.

Round Robin (RR): Mỗi tiến trình được cấp một khoảng thời gian nhất định (time quantum) để thực thi theo vòng tròn.

Multilevel Queue Scheduling: Tiến trình được phân loại vào các hàng đợi khác nhau và mỗi hàng đợi có chính sách lập lịch riêng.

Multilevel Feedback Queue Scheduling: Tiến trình có thể di chuyển giữa các hàng đợi khác nhau dựa trên hiệu suất và thời gian xử lý.

Rate Monotonic Scheduling (RMS): Tiến trình có chu kỳ ngắn hơn sẽ được ưu tiên cao hơn.

Earliest Deadline First (EDF): Tiến trình có thời hạn gần nhất sẽ được ưu tiên cao nhất.

7. Among the three scheduling algorithms listed, which one is the best for response time?

Round Robin (RR) is the best for response time. Since it allocates CPU time in time slices, each process gets a chance to execute early, reducing response time compared to FCFS and SJF (in SJF, longer processes may wait too long).

8. Three processes start at times 1, 2, and 3. The CPU order is: 1, 2, 3. If CPU scheduling does not follow arrival order, which scheduling method might be used?

10. Which operating system supports file access tracking based on the most recent access time (last access time) with more than one visitor?

Unix/Linux can store last access time information for files and can track the number of users accessing files using commands like `who`, `last`, and `finger`. In summary, Unix/Linux can manage file access with multiple visitors.

11. What is a major difference between Multiprogramming system and Time-sharing system?

Multiprogramming

- Maximize CPU utilization by organizing programs so that the CPU always has one to execute.

- Sometimes processes have to wait a long time for another time-consuming process.

Time-sharing

- The CPU executes multiple processes by switching among them, but the switches occur frequently, providing the user with a fast response time.
- Processes no longer have to wait for the last task to end to get the processor.

12. List 3 possible causes for an operation to change from USER MODE to SYSTEM MODE

System Call: When a process requests a service from the operating system, such as accessing a file or requesting memory.

Interrupt: When a hardware device needs the CPU's attention, such as when an I/O device completes an operation.

Exception: When an error occurs during execution, such as a division by zero or a segmentation fault.

13. Contiguous allocation of files may cause of disk fragmentation. Is this internal or external fragmentation?

This is external fragmentation. External fragmentation occurs when there is enough total free space to store a file, but this space is scattered into small, non-contiguous blocks, making it difficult to store a new file.

1. Introduction

OS: A software layer between the hardware and the application programs/users which provides a virtual machine interface

Processes – System Abs, Threads – Processor Abs, Virtual Memory – Memory Abs, File System – Storage Abs

Spooling - Simultaneous Peripheral Operation On-Line: Improve the performance of the computer system by managing peripherals

Time-Sharing (Chia sẻ thời gian):

Minh họa: Time-sharing là kỹ thuật cho phép nhiều người dùng tương tác với máy tính cùng lúc bằng cách chia sẻ thời gian CPU. Mỗi người dùng có cảm giác như họ đang sử dụng máy tính riêng của mình.

Lợi ích: Cải thiện hiệu suất và khả năng sử dụng của hệ thống, đặc biệt là trong môi trường đa người dùng.

Multiprogramming

Khái niệm: Multiprogramming là kỹ thuật cho phép nhiều chương trình (process) được nạp vào bộ nhớ và thực thi đồng thời. Khi một chương trình đang chờ I/O, CPU có thể chuyển sang thực thi chương trình khác, giúp tăng hiệu suất sử dụng CPU.

Lợi ích:

Tăng hiệu suất: Giảm thời gian CPU nhàn rỗi bằng cách chuyển đổi giữa các chương trình khi một chương trình đang chờ I/O.

Tăng thông lượng hệ thống: Nhiều chương trình có thể được thực thi trong cùng một khoảng thời gian.

OS type: Parallel OS (Multiprogramming), Real-Time OS (require deadline), Distributed OS (many small computers), Embedded OS (Phone)

2-----

Kiến trúc Von Neumann:

Minh họa: Đây là mô hình kiến trúc máy tính cơ bản, trong đó dữ liệu và các lệnh được lưu trữ trong cùng một bộ nhớ.

Thành phần chính: Bao gồm bộ xử lý trung tâm (CPU), bộ nhớ, và các thiết bị đầu vào/đầu ra (I/O).

Von Neumann Machine

Nguyên lý hoạt động:

Lưu trữ chương trình: Cả dữ liệu và lệnh đều được lưu trữ trong bộ nhớ, cho phép CPU truy cập và thực thi lệnh một cách tuần tự.

Chu trình lệnh: CPU thực hiện các lệnh theo chu trình nạp lệnh, giải mã lệnh, và thực thi lệnh.

Lợi ích:

Đơn giản hóa thiết kế: Kiến trúc này giúp đơn giản hóa thiết kế và xây dựng máy tính.

Tăng tính linh hoạt: Cho phép các chương trình tự sửa đổi và tạo ra các chương trình khác.

Processor modes (0 = system mode / 1 = user mode)

Exceptions

Traps: là một exception đặc biệt buộc Program Counter (PC) chuyển đến một địa chỉ đã biết và đặt chế độ CPU vào chế độ hệ thống (system mode).

Interrupt (Ngắt):

Minh họa: Khi một thiết bị hoàn thành một thao tác I/O, nó sẽ gửi một tín hiệu ngắt đến CPU để thông báo rằng thao tác đã hoàn thành.

Ưu điểm: Giảm thiểu thời gian chờ đợi của CPU vì CPU không cần phải kiểm tra liên tục trạng thái của thiết bị.

Nhược điểm: Cần phần cứng hỗ trợ và có thể gây ra sự gián đoạn trong quá trình thực thi của CPU.

Polling (Thăm dò):

Minh họa: CPU kiểm tra trạng thái của thiết bị theo chu kỳ để xem liệu thao tác I/O đã hoàn thành hay chưa.

Ưu điểm: Đơn giản và không cần phần cứng hỗ trợ đặc biệt.

Nhược điểm: Tốn thời gian CPU vì CPU phải kiểm tra trạng thái của thiết bị liên tục, dẫn đến lãng phí tài nguyên.

PIO: L+S*B (L-load, S-store, B-byte)

CPU phải kiểm tra thanh ghi trạng thái và gửi hoặc nhận dữ liệu.

PIO phù hợp với các khối dữ liệu nhỏ và có chi phí thấp hơn DMA.

DMA: 4L+4000

Thiết lập thanh ghi base, count, kiểm tra trạng thái và xử lý ngắt.

DMA hiệu quả hơn khi di chuyển các khối dữ liệu lớn.

Boot Protocol

CPU is hard-wired to start executing from a known address in memory

ROM contains the “boot” code

The boot loader can then load the rest of the operating system from disk.

3---processes

Activation Record: include local variables, parameters, return address, previous activation record

Process = Code + Global + Stack + ... + Heap

Process Control Block là một cấu trúc dữ liệu được hệ điều hành sử dụng để lưu trữ thông tin về một tiến trình cụ thể. Đặc biệt process id (PID) khác nhau tức cha con khác nhau

Process Creation: using fork(), <>0: parent, =0: child

Signal: là cơ chế mà hệ điều hành UNIX sử dụng để thông báo cho chương trình người dùng khi một sự kiện quan tâm xảy ra

Interprocess Communication - IPC): Cho phép các tiến trình giao tiếp và đồng bộ hóa hành động của chúng mà không cần sử dụng các biến chia sẻ (lệnh send() và receive())

4—scheduling

CPU Scheduling decisions: 1+4 = nonpreemptive, all=preemptive

1. Switches from running to waiting state
2. Switches from running to ready state
3. Switches from waiting to ready
4. Terminates

Dispatcher là module chịu trách nhiệm chuyển quyền điều khiển CPU cho tiến trình được chọn bởi bộ lập lịch ngắn hạn (short-term scheduler).

Scheduling Criteria:

CPU Utilization: Giữ cho CPU bận rộn nhất có thể.

Throughput: Số lượng tiến trình hoàn thành trong một đơn vị thời gian.

Turnaround Time: Thời gian để hoàn thành một tiến trình cụ thể.

Waiting Time: Thời gian một tiến trình chờ trong hàng đợi sẵn sàng.

Response Time: Thời gian từ khi một yêu cầu được gửi đến khi có phản hồi đầu tiên (không phải đầu ra).

Linux-Scheduling

Hai thuật toán lập lịch:

Time-sharing: Dựa trên hệ thống ưu tiên dựa trên tín dụng (credit-based). Tiến trình có nhiều tín dụng nhất sẽ được lập lịch tiếp theo. Khi tín dụng của một tiến trình bằng 0, tiến trình khác sẽ được chọn. Khi tất cả các tiến trình đều có tín dụng bằng 0, hệ thống sẽ cấp lại tín dụng dựa trên các yếu tố như mức độ ưu tiên và lịch sử.

Real-time: Hỗ trợ thời gian thực mềm (soft real-time) và tuân thủ chuẩn Posix.1b với hai lớp: First-Come, First-Served (FCFS) và Round Robin (RR). Tiến trình có mức độ ưu tiên cao nhất sẽ được thực thi trước.

Ưu tiên và độ dài time-slice:

Mối quan hệ giữa mức độ ưu tiên và độ dài time-slice: Tiến trình có mức độ ưu tiên cao hơn sẽ có time-slice dài hơn, giúp đảm bảo tiến trình quan trọng được thực thi kịp thời.