**CHAPTER 8**

8.1- Operating System OverviewThe Operating System as a User/Computer InterfaceThe Operating System as Resource ManagerTypes of Operating Systems

**Operating System (OS) ServicesThe most important system program**Masks the details of the hardware from the programmer and provides the programmer with a convenient interface for using the system**The OS typically provides services in the following areas:**Program creationProgram executionAccess to I/O devicesControlled access to filesSystem accessError detection and responseAccounting

**InterfacesInstruction Set Architecture (ISA)**Defines the machine language instructions that a computer can follow**.** Boundary between hardware and software**Application Binary Interface (ABI)**Defines a standard for binary portability across programsDefines the system call interface to the operating system and the hardware resources and services available in a system through the user ISA**Application Programming Interface (API)**Gives a program access to the hardware resources and services available in a system through the user ISA supplemented with high-level language (HLL) library callsUsing an API enables application software to be ported easily to other systems that support the same API

**Operating System as Resource Manager**Máy tính là một tập hợp các tài nguyên để di chuyển, lưu trữ, xử lý dữ liệu và kiểm soát các chức năng này. Hệ điều hành (OS) chịu trách nhiệm quản lý các tài nguyên đó.

Hệ điều hành với vai trò là một cơ chế kiểm soát có hai điểm đặc biệt:

* Hệ điều hành hoạt động giống như phần mềm máy tính thông thường – nó là một chương trình được bộ xử lý thực thi.
* Hệ điều hành thường xuyên nhường quyền kiểm soát và phải phụ thuộc vào bộ xử lý để có thể giành lại quyền kiểm soát.

**Types of Operating SystemsInteractive systemNgười dùng hoặc lập trình viên tương tác trực tiếp với máy tính để yêu cầu thực thi một công việc hoặc thực hiện một giao dịch. Tùy vào ứng dụng, người dùng có thể giao tiếp với máy tính trong quá trình thực thi công việc.**

**Batch systemNgược lại với chế độ tương tác, chương trình của người dùng được gộp chung với các chương trình khác và được gửi lên bởi một người vận hành máy tính. Sau khi chương trình hoàn thành, kết quả sẽ được in ra cho người dùng.**

**Early SystemsFrom the late 1940s to the mid-1950s the programmer interacted directly with the computer hardware – there was no OS**Processors were run from a console consisting of display lights, toggle switches, some form of input device and a printer**Các vấn đề:**

1. **Lập lịch (Scheduling)**
   * Sử dụng bảng đăng ký để đặt trước thời gian sử dụng bộ xử lý.
   * Lãng phí thời gian rảnh của máy tính nếu người dùng hoàn thành sớm.
   * Người dùng có thể bị buộc dừng trước khi giải quyết xong vấn đề.
2. **Thời gian thiết lập (Setup Time)**
   * Chạy một chương trình yêu cầu nhiều bước:
     + Nạp trình biên dịch và chương trình nguồn vào bộ nhớ.
     + Lưu chương trình đã biên dịch.
     + Nạp và liên kết chương trình đối tượng với các hàm chung.

**Simple Batch System:**Memory Layout for a Resident Monitor**From the View of the Processor . . .Processor executes instructions from the portion of main memory containing the monitorThe monitor handles setup and scheduling Job control language (JCL)Desirable Hardware FeaturesMemory protection**The user program must not modify the monitor's memory. If an error occurs, the processor detects it, transfers control to the monitor, which then aborts the job, prints an error message, and loads the next job.

**Timer:** A timer prevents a job from monopolizing the system by triggering an interrupt to return control to the monitor.

**Interrupts**: Gives the OS more flexibility in relinquishing control to and regaining control from user programs

**Privileged instructions**: Lệnh đặc biệt chỉ có thể thực thi bởi monitor; nếu user program thực thi, sẽ gây lỗi. I/O instructions là lệnh đặc quyền để monitor kiểm soát thiết bị I/O.

System Utilization ExampleThe processor is often idle -> Multiple jobs can be carried out.

**Time Sharing Systems**

Used when the user interacts directly with the computerProcessor’s time is shared among multiple usersMultiple users simultaneously access the system through terminals, with the OS interleaving the execution of each user program in a short burst or quantum (time slice, time slot) of computation

**Batch Multiprogramming versus Time Sharing**A close-up of a computer program

AI-generated content may be incorrect.

**8.2- SchedulingThe key to multiprogramming**

A close-up of a text

AI-generated content may be incorrect.

**Program**: executable file stored in external memory**Process**: program in execution

**Long Term SchedulingSwapped-out process:** Process stored in RAM must be write to disk. It shields it’s memory to swapped-in process.**Xác định** chương trình cần xử lý.

**Chuyển đổi** chương trình thành tiến trình.

**Có thể** đưa vào hàng đợi swapped-out (nếu có).

**Xử lý theo lô**: Đưa tiến trình vào hàng đợi, thực thi khi tài nguyên có sẵn.

**Chia sẻ thời gian**: Nhận tiến trình nếu tài nguyên còn trống, từ chối nếu hệ thống đầy.

**Medium-Term Scheduling  
 and Short-Term SchedulingMTS**: Swapping Function:

* Swapping-in manages the degree of multiprogramming.
* Swapping-in decisions consider the memory requirements of swapped-out processes.

**STS:** Also known as the dispatcher (trình điều phối)Executes frequently and makes the fine-grained decision of which job to execute next

**Process Control Block (PCB)**

**Program counter** : Index of instruction which will be fetched when the process re-executes.**Context data:** Values of registers at the time the process is blocked

**8.3- Memory Management**

Memory Management Swapping Partioning Paging Virtual Memory Translation Lookaside Buffer Segmentation**Memory Manager is a part of OS which bears responsibility to manage computer memory at the system level and some above techniques can be applied.**

**Memory Management:  
SwappingWhy?Memory has larger size ->Processes request more and more memory, more processes need to run -> Memory is not enough to supply -> A selected process must be swapped out to disk in order to load new process (SWAP)**

**PartitioningSmaller process needs smaller memory -> Unequal-size partition is better.**

**PagingFrame size= Page size**

**At a time, only one instruction of the current process executes -> Only necessary part of each process is loaded -> Many processes can be loaded.**

**Programs are divided into small fixed chunk (ex. 4KB). At a time, only some pages of each process are loaded to memory (frames)**

**Memory is divided also to frames**

**Logical and Physical AddressesEach program address is expressed as a logical address which is a pair of (page, offset)**

**Virtual Memory: Demand PagingEach page of a process is brought in only when it is neededPrinciple of localityWhen working with a large process execution may be confined (limited) to a small section of a program (subroutine)It is better use of memory to load in just a few pagesIf the program references data or branches to an instruction on a page not in main memory, a page fault is triggered which tells the OS to bring in the desired page.**

**Advantages:**More processes can be maintained in memoryTime is saved because unused pages are not swapped in and out of memory**Disadvantages:**When one page is brought in, another page must be thrown out **(page replacement)**If a page is thrown out just before it is about to be used the OS will have to go get the page again**Thrashing** (đánh bại- hệ thống trì trệ)When the processor spends most of its time swapping pages rather than executing instructions

**Paging: Inverted Page Table Structure**

A large table is used to store all pages of all procsses

Paging: Operation of Paging and Translation Lookaside Buffer (TLB)**TLB** là một phần cứng bao gồm một số thanh ghi. Một phần của **bảng trang (page table)** được sao chép vào đó để tăng hiệu suất chuyển đổi **địa chỉ ảo** sang **địa chỉ vật lý**.

**SegmentationProgram is divided in to segments (data, code, stack, heap segments)Usually visible to the programmer**

**Cung cấp cách tổ chức chương trình, dữ liệu và gán quyền bảo vệ.**

**Cho phép lập trình viên xem bộ nhớ dưới dạng nhiều không gian địa chỉ (segments).**

**Mỗi segment có thể được chia thành nhiều trang (pages).**

**Advantages:Simplifies managing dynamic data structures.**

**Allows independent modification and compilation of programs without re-linking or re-loading everything.**

**Segments can be shared among processes.**

**Each segment can have individual protection.**