

## **`Chương 1, 2. Tổng quan về Hệ điều hành, Cấu trúc Hệ điều hành**

### **1. Hệ điều hành là gì?**

- Là thành phần trung gian giữa người dùng và hệ thống phần cứng máy tính.

### **2. Hệ điều hành có chức năng gì? Quản lý thiết bị, quản lý tiến trình và tài nguyên, quản lý bộ nhớ và quản lý file.**

- Thực thi chương trình người dùng dễ dàng hơn
- Sử dụng hệ thống máy tính thuận tiện hơn
- Sử dụng hệ thống máy tính một cách hiệu quả
- Giả lập một máy tính ảo
- Quản lý tài nguyên của hệ thống

### **3. Tài nguyên hệ thống gồm những gì?**

- Bộ nhớ
- Processor
- Thiết bị ngoại vi

### **4. Hệ điều hành đơn, đa nhiệm? HDH thời gian thực? Hệ điều hành tập trung?**

- Hệ điều hành đơn nhiệm:
  - Tại một thời điểm xác định, khi một chương trình được đưa vào bộ nhớ thì nó chiếm giữ mọi tài nguyên của hệ thống, và vì vậy chương trình khác không thể được đưa vào bộ nhớ trong khi nó chưa kết thúc
  - Nhưng do các thiết bị vào ra thường làm việc với tốc độ chậm, người ta dùng kỹ thuật Spooling cho phép tạo ra hiệu ứng song song các thiết bị chỉ cho phép vào ra tuần tự
- Hệ điều hành đa nhiệm:
  - HDH cho phép tại một thời điểm có nhiều chương trình ở trong bộ nhớ trong. Chúng có nhu cầu được phân phối thời gian phục vụ CPU, bộ nhớ và thiết bị ngoại vi. Như vậy CPU, bộ nhớ, thiết bị ngoại vi... là các tài nguyên được chia sẻ cho các chương trình đó. Vấn đề là làm sao đảm bảo tốt nhất tính bình đẳng khi giải quyết vấn đề phân phối tài nguyên.
- Hệ điều hành thời gian thực:
  - Mỗi tiến trình khi đã xâm nhập vào hệ thống thì ở bất kỳ lúc nào đều được phân phối CPU
- Hệ điều hành tập trung:
  - Trên một hệ thống máy tính chỉ có một HDH duy nhất cài ở máy chủ. Các máy trạm được khởi động nhờ máy chủ và nó chỉ làm chức năng nhập/xuất dữ liệu. Mọi xử lý đều tập trung ở máy chủ

### **5. Các tính chất (yếu tố) của HDH: hiệu suất, bảo vệ và an ninh, tính chính xác, khả năng bảo trì, thương mại, chuẩn và hệ thống mở**

- Tin cậy
- An toàn
- Khái quát theo thời gian
- Hiệu quả
- Thuận tiện

### **6. Cấu trúc phân lớp của HDH: ý nghĩa, lời gọi hệ thống là gì, hoạt động thế nào?**

- Ý nghĩa:
  - Trong máy tính, một lời gọi hệ thống là việc một chương trình máy tính yêu cầu một dịch vụ từ nhân của hệ điều hành mà nó được thực thi. Điều này có thể bao gồm các dịch vụ liên quan đến phần cứng (ví dụ: truy cập ổ đĩa cứng), tạo và thực thi các tiến trình mới, giao tiếp với các dịch vụ nhân tích hợp như lập lịch xử lý. Các lời gọi hệ thống cung cấp một giao diện quan trọng giữa một tiến trình và hệ điều hành.

- **Lời gọi hệ thống:**
  - Là giao diện lập trình đến các dịch vụ hệ thống được cung cấp bởi OS
  - Thường được viết bằng ngôn ngữ bậc cao (C hay C++)
  - Các chương trình thường truy nhập đến các lời gọi hệ thống thông qua giao diện chương trình ứng dụng (API) (không gọi trực tiếp các lời gọi hệ thống)
- **Hoạt động:**
  - Khi tiến trình dùng lời gọi hệ thống cần cung cấp tham số cho lời gọi hệ thống. Có ba phương pháp mà tiến trình dùng để chuyển tham số cho hệ điều hành:
    - + Chuyển tham số vào thanh ghi
    - + Lưu trữ tham số trong một bảng trong bộ nhớ và ghi địa chỉ bảng vào thanh ghi
    - + Lưu trữ tham số và stack và tham số được lấy ra bởi hệ điều hành.

## Chương 3: Quản lý tiến trình

### 1. Định nghĩa tiến trình

- Tiến trình là một chương trình đang sử lý, mỗi tiến trình có một không gian địa chỉ, một con trỏ lệnh, một tập các thanh ghi, stack riêng.

### 2. Các trạng thái của tiến trình.

- Trạng thái sẵn sàng: tiến trình chờ cấp CPU để thực hiện lệnh của mình.
- Trạng thái chạy: lệnh của tiến trình đc CPU thực hiện, với những máy tính có 1 CPU và CPU có một lõi, tại mỗi thời điểm chỉ có một tiến trình nằm trong trạng thái chạy.
- Trạng thái chờ đợi: tiến trình chờ đợi một sự kiện gì đó xảy ra.
- Trạng thái kết thúc: tiến trình ko còn nằm trong danh sách các tiến trình thực hiện nhưng vẫn chưa bị xóa.

### 3. Khối điều khiển tiến trình (PCB)

- Định danh tiến trình: mã số tiến trình, giúp phân biệt tiến trình này với tiến trình khác.
- Trạng thái tiến trình: xác định hoạt động hiện hành của tiến trình.
- Ngữ cách tiến trình: mô tả các tài nguyên tiến trình đang đc sử dụng, để phục vụ cho hoạt động hiện tại hoặc để làm cơ sở phục hồi tiến trình hoạt động trở lại.
- Thông tin giao tiếp:
- Thông tin thống kê.

### 4. Các kiểu tiến trình: độc lập, hợp tác, song song, tuần tự

- **Tiến trình độc lập:** không bị ảnh hưởng bởi tiến trình khác, không chia sẻ dữ liệu
- **Tiến trình hợp tác:** Bị ảnh hưởng bởi tiến trình khác, dùng chung dữ liệu
- **Tiến trình tuần tự:** là các tiến trình mà điểm khởi tạo của nó là điểm kết thúc của tiến trình trước đó.
- **Tiến trình song song:** là các tiến trình mà điểm khởi tạo của tiến trình này nằm ở thân của các tiến trình khác, tức là có thể khởi tạo một tiến trình mới khi các tiến trình trước đó chưa kết thúc.

## **5. Tiến trình cha, con, các tiến trình cha con dùng chung cái gì?**

Có 2 khả năng có thể xảy ra :

- Tiến trình cha tiếp tục thực thi song song với các tiến trình con của nó.
- Tiến trình cha đợi đến khi một vài hoặc tất cả các tiến trình con của nó đã hoàn tất.
- Trong khi giải bài tập thì chúng ta sẽ giải theo trường hợp 2 (Tiến trình cha đợi đến khi tiến trình con vừa gọi kết thúc mới thực thi tiếp).

## **6. Ngữ cảnh tiến trình là gì, gồm những gì?**

- Chứa giá trị các thanh ghi, trạng thái tiến trình và thông tin quản lý bộ nhớ

## **7. Nhiệm vụ của bộ điều phối, bộ phân phối**

- Hệ điều hành là bộ điều phối, nói về một quản lý vĩ mô... bộ phân phối là 1 module...

## **8. Giờ CPU là gì?**

- Là thời gian mà CPU phục vụ cho tiến trình hoạt động

## **9. Các trạng thái của tiến trình liên quan đến giờ CPU**

- Ready :Tiến trình đã có đủ tài nguyên, đang chờ được cấp CPU để chạy.
- Running :Các lệnh của tiến trình đang được thực thi.
- Waiting (hay blocked) : Tiến trình đợi I/O hoàn tất

## **10. Khái niệm lập lịch CPU:**

- Là một cách thức sắp xếp thứ tự thực hiện các tiến trình

## **11. Các thuật toán lập lịch một hàng đợi**

- 4 thuật toán: FCFS, SJN, SRN, RR.

## **12. Lập lịch nhiều hàng đợi (MLQ, MLFQ)**

- MLQ: Để phân lớp các quá trình đang trong trạng thái chuẩn bị và chọn lựa quá trình chuyển sang trạng thái sử dụng có thể sử dụng các thông tin được cho bằng người tạo ra quá trình đó và các thông tin nhận được trong việc điều phối các quá trình
- MLFQ: Ý tưởng chính của giải thuật là phân lớp các tiến trình tùy theo độ ưu tiên của chúng để có cách thức điều phối thích hợp cho từng nhóm.

# **Chương 4: Đồng bộ tiến trình**

## **1. Bài toán đoạn tới hạn (đoạn găng)**

- Đảm bảo rằng khi một tiến trình đang trong đoạn tới hạn của nó, các tiến trình khác không được phép thực thi đoạn tới hạn của chúng.

## **2. Các phương pháp giải quyết bài toán đoạn tới hạn:**

- Có 5 cách:
  - + Khóa trong (tt Dekker)
  - + Kiểm tra và xác lập
  - + Đèn hiệu
  - + Dùng trình thư ký
  - + Tổ chức liên lạc giữa các tiến trình

# **Chương 5: Bế tắc.**

## **1. Hiện tượng bế tắc?**

- Hai hay nhiều tiến trình chờ vô hạn trên một sự kiện gây ra bởi một trong các tiến trình đang chờ
- Máy tính: Bế tắc là tình huống xuất hiện khi hai tiến trình phải chờ đợi nhau giải phóng tài nguyên hoặc nhiều tiến trình chờ sử dụng các tài nguyên theo một "vòng tròn".

## 2. Điều kiện xảy ra bế tắc?

- Bế tắc có thể xảy ra nếu cả bốn điều kiện sau xảy ra đồng thời :
  - Độc quyền truy xuất (Mutual Exclusion): chỉ một tiến trình được sử dụng tài nguyên tại một thời điểm.
  - Giữ và chờ (Hold and wait): một tiến trình giữ ít nhất một tài nguyên và chờ các tài nguyên khác đang được giữ bởi các tiến trình khác.
  - Không chiếm đoạt (No preemption): một tài nguyên chỉ có thể được giải phóng một cách tự nguyện bởi tiến trình giữ nó, sau khi tiến trình đó hoàn thành công việc của nó.
  - Chờ đợi vòng tròn (Circular wait)

## 3. Cách phòng bế tắc

- Yêu cầu hệ thống phải có trước một số thông tin
  - Mô hình đơn giản: mỗi tiến trình khai báo số tài nguyên lớn nhất thuộc mỗi loại mà nó cần.
  - Thuật toán tránh bế tắc kiểm tra trạng thái phân phối tài nguyên để đảm bảo rằng không bao giờ có điều kiện “chờ đợi vòng tròn” xảy ra.
  - Trạng thái phân phối tài nguyên được xác định bằng số các tài nguyên rỗi, số tài nguyên đã được phân phối và số cực đại yêu cầu của các tiến trình.

## 4. Đồ thị cấp phát tài nguyên

- Đồ thị này chứa một tập các đỉnh V và tập hợp các cạnh E. Một tập các đỉnh V được chia làm hai loại nút  $P = \{P_1, P_2, \dots, P_n\}$  là tập hợp các quá trình hoạt động trong hệ thống, và  $R = \{R_1, R_2, \dots, R_m\}$  là tập hợp chứa tất cả các loại tài nguyên trong hệ thống.

# Chương 6: Bộ nhớ chính

## 1. Nhiệm vụ quản lý bộ nhớ của HDH

- Quản lý tiến trình
- Quản lý bộ nhớ
- Quản lý thiết bị vào ra

## 2. Khái niệm địa chỉ luận lý (logic), địa chỉ vật lý (physical)

- Địa chỉ luận lý (logic): là địa chỉ do bộ xử lý tạo ra.
- Địa chỉ vật lý (physical): là địa chỉ thực trong bộ nhớ chính, địa chỉ vật lý còn gọi là địa chỉ tuyệt đối/địa chỉ thực.

## 3. Các cấu trúc cơ bản của chương trình (tuyến tính, phân đoạn, overlay, động)

- Phân đoạn
  - Một chương trình được người lập trình chia thành nhiều phân đoạn, mỗi phân đoạn có ngữ nghĩa khác nhau và hệ điều hành có thể nạp các phân đoạn vào bộ nhớ tại các vị trí không liên tục.
- Overlay:
  - Chỉ giữ trong bộ nhớ những câu lệnh và dữ liệu cần trong bất cứ thời điểm nào
  - Cần khi tiến trình lớn hơn kích cỡ bộ nhớ được cấp cho nó
  - Được thực thi bởi người dùng, không cần sự hỗ trợ đặc biệt từ hệ điều hành, thiết kế lập trình của cấu trúc overlay tương đối phức tạp.

## 4. Các sơ đồ quản lý bộ nhớ (phân hoạch, trao đổi, phân đoạn, phân trang)

## 5. Các thuật toán FirstFit, BestFit, WorstFit

- First-fit: chọn đoạn trống đầu tiên đủ lớn
- Best-fit: chọn đoạn trống nhỏ nhất nhưng đủ lớn để thỏa mãn nhu cầu
- Worst-fit: chọn đoạn trống lớn nhất.

## **Chương 7: Bộ nhớ ảo**

### **1. Bộ nhớ ảo là gì? Đặc điểm**

- Bộ nhớ ảo là một vùng thuộc không gian lưu trữ của bộ nhớ phụ hệ thống máy tính (chẳng hạn như ổ cứng HDD hoặc SSD). Nó hoạt động như thể là một phần của RAM hoặc bộ nhớ chính hệ thống.
- Đặc điểm: Bộ nhớ ảo khiến cho việc lập trình ứng dụng dễ dàng hơn bởi chúng làm ẩn đi sự phân mảnh bộ nhớ vật lý, bằng cách ủy thác cho kernel hệ điều hành sự quản lý phân cấp bộ nhớ (loại bỏ việc chương trình máy tính tự xử lý việc sử dụng chồng ô nhớ). Mỗi khi ứng dụng sử dụng không gian bộ nhớ được chỉ định bởi ứng dụng, bộ nhớ ảo sẽ ngăn ngừa việc phân bổ lại ô nhớ hoặc cho truy cập ô nhớ có địa chỉ ảo tương ứng.

### **2. Các thuật toán thay trang FIFO, Optimal, LRU.**

- **LRU:** cài đặt bằng ngăn xếp
  - Lưu giữ số hiệu trang trong một ngăn xếp (Cài đặt một danh sách liên kết kép)
  - Tham chiếu trang (chuyển lên đầu ngăn xếp - Cần thay đổi tổng cộng 6 con trỏ)
  - Không đòi hỏi tìm kiếm khi thay trang
- **Optimal:** Thuật toán tối ưu
  - Thay trang sẽ không được sử dụng trong thời gian dài

## **Chương 8: Hệ thống tập tin**

### **1. Phương pháp cấp phát liên tục, danh sách móc nối (liệt kê), chỉ số trong quản lý đĩa từ**

- **Phương pháp cấp phát liên tục:**
  - Mỗi file lưu trong một tập các khối liên tục trên đĩa
  - Đơn giản – chỉ cần điểm bắt đầu (block #) và kích cỡ (số các blocks)
  - Truy nhập ngẫu nhiên • Không tận dụng không gian tối ưu
  - Các file không thể thay đổi kích cỡ
- **Phân phối móc nối (liên kết)**
  - Một file có thể là một danh sách các khối đĩa: các khối đĩa có thể nằm rải rác
- **Phân phối chỉ số:**
  - Khối index: chứa toàn bộ con trỏ đến các khối đĩa
  - Cần bảng chỉ số
  - Truy cập ngẫu nhiên
  - Phân phối động mà không sinh ra phân mảnh ngoài (chỉ tốn không gian cho bảng index)
  - Ánh xạ từ không gian luận lý sang không gian vật lý trong một file kích cỡ tối đa là 512KB và kích cỡ khối là 1024B chúng ta chỉ cần một khối cho bảng index.

## 2. Phương pháp bitmap trong quản lý đĩa từ (không gian trống)

- Không gian đĩa được chia thành các khối (block) và được đánh số từ 0 ... max. Mỗi đĩa sử dụng 1 bit để đánh dấu trạng thái. Khối đĩa nào đã sử dụng thì bit tương ứng có giá trị bằng 1, chưa được sử dụng thì có giá trị bằng 0.
- Tập hợp các ký hiệu 0, 1 tạo thành 1 bit vector (bitmap). Đọc thông tin trong bitmap hệ điều hành có thể xác định được không gian tự do trên đĩa.

## 3. Phương pháp lập nhóm, đếm trong quản lý đĩa (không gian trống)

- Phương pháp lập nhóm cho phép hệ thống nhóm các khối đĩa tự do liên tiếp thành 1 nhóm. Khối đĩa tự do đầu tiên trong nhóm lưu trữ địa chỉ của các khối đĩa tự do trong nhóm. Khối đĩa cuối cùng trong nhóm lưu trữ địa chỉ của khối đĩa tự do đầu tiên của nhóm tiếp theo.
- Phương pháp đếm là sự biến đổi của phương pháp lập nhóm.
- Trong phương pháp này, hệ thống lập danh sách quản lý địa chỉ của các khối đĩa tự do đầu tiên và số lượng các khối đĩa tự do liên tục kế tiếp các khối đĩa đó.

## 4. Một số kiểu tổ chức file của hệ điều hành: tổ chức thư mục hai mức, tổ chức thư mục theo cấu trúc cây, tổ chức theo đồ thị không chu trình.

- **Thư mục hai mức:**
  - Mỗi người sử dụng có 1 thư mục riêng, khi làm việc với file chỉ duyệt thư mục riêng. Khi log in, hệ thống sẽ kiểm tra và cho phép người sử dụng làm việc với thư mục riêng. Khi thêm một người dùng
    - Hệ thống tạo bởi phần tử mới trong Master file directory
    - Tạo ra Use file directory
  - Giải quyết vấn đề trùng tên;
  - Hiệu quả khi người dùng đọc lập. Khó khăn khi muốn dùng chung file
- **Thư mục cấu trúc cây:**
  - Tồn tại một đường dẫn (tương đối/tuyệt đối) đến một file
  - Thư mục con là file được xử lý đặc biệt (bit đánh dấu)
  - Các thao tác tạo/xóa/duyệt... thực hiện trên thư mục hiện thời
  - Xóa thư mục con => Xóa hết các cây con của nó
- **Tổ chức theo đồ thị không chu trình:**
  - Người dùng có thể link đến một file của người dùng khác
  - Khi duyệt thư mục (backup) file có thể duyệt nhiều lần
  - Xóa file: liên kết/nội dung (người tạo file/liên kết cuối)

## Chương 9: Quản lý thiết bị

1. Nguyên lý hoạt động?
2. Chiến lược quản lý thiết bị và vào/ra (Polling, Ngắt, DMA)
3. Trình điều khiển thiết bị
4. Cấu trúc vật lý của đĩa từ (disk, platter, head)
5. Cấu trúc luận lý của đĩa từ (track, cylinder, sector, cluster)
6. Định thời truy cập đĩa: các phương pháp First Come First Serve (FCFS), Shortest-Seek-Time First (SSTF), SCAN, C-SCAN (Circular SCAN), C-LOOK.