**CHƯƠNG 1 : TỔNG QUAN VỀ HỆ ĐIỀU HÀNH**

**1. Khái niệm hệ điều hành**

- Cấu trúc phân lớp của hệ thống tính toán

+ Các thành phần của hệ thống tính toán : Application programs, System programs, Hardware, User

+ HĐH : Nằm giữa phần cứng của hệ thống và các chương trình ứng dụng, để cung cấp một môi trường để người

sử dụng có thể thực hiện các chương trình ứng dụng làm cho máy tính dễ sử dụng hơn, thuận lợi hơn và hiệu quả hơn.

- Chức năng hệ điều hành

+ Giả lập một máy ảo.

+ Quản lý tài nguyên của hệ thống.

**2. Lịch sử phát triển HĐH**

**3. Định nghĩa và phân loại HĐH**

Trên các góc độ quan sát khác nhau sẽ có các định nghĩa khác nhau

- Người sử dụng : HĐH là chương trình phục vụ khai thác hệ thống tính toán một cách thuận lợi

- Người quản lý : HĐH là chương trình phục vụ quản lý chặt chẽ và khai thác tối ưu các tài nguyên của hệ thống.

- QĐ kỹ thuật : HĐH là một chương trình trang bị cho một máy tính cụ thể để tạo ra máy tính logic mới với tài nguyên

mới và khả năng mới.

- QĐ hệ thống : HĐH là một hệ thống mô hình hóa, mô phỏng hoạt động của máy tính, của người sử dụng và của các thao tác

viên, hoạt động trong chế độ đối thoại nhằm tạo môi trường khai thác thuận lợi hệ thống máy tính và quản lý tối ưu tài

nguyên của hệ thống.

Phân loại :

- Hệ thống xử lý theo lô đơn chương trình

Thực hiện các chương trình (Job) lần lượt theo những chỉ thị đã định trước, khi 1 Job kết thúc hệ thống tự động

thực hiện những Job tiếp theo mà không cần sự can thiệp từ quên ngoài.

-> Đòi hỏi tổ chức hàng đợi công việc (job queue)

-> Vấn đề : Khi nhập thiết bị vào ra -> Processor rơi vào trạng thái chờ đợi

- Hệ thống xử lý theo lô đa chương trình

Cho phép thực hiện nhiều chương trình đồng thời

-> Ưu điểm : Tiết kiệm bộ nhớ (không cần nạp toàn bộ chương trình vào bộ nhớ), hạn chế thời gian rảnh rỗi của

Processor.

- Hệ thống phân chia thời gian thực

Chia sẻ thời gian của Processor cho các chương trình/tiến trình đang sẵn sàng thực hiện

Thường được gọi : hệ điều hành đa nhiệm ( Windows)

- Hệ thống xử lý thời gian thực : Được sử dụng chủ yếu trong lĩnh vực điều khiển

- Hệ thống song song : Xây dựng cho hệ thống có nhiều bộ vi xử lý.

- Hệ thống phân tán : Mỗi bộ xử lý có một bộ nhớ cục riêng và trao đổi thông qua các đường truyền thông.

-> Hệ thống phân tán được sử dụng để : Chia sẻ tài nguyên, tăng tốc độ tính toán, an toàn.

**4. Các khái niệm trong HĐH**

- Tiến trình (Process) : là một chương trình đang thực hiện

Tiến trình >< chương trình

Chương trình : Thực thể thụ động, chứa đựng các chỉ thị điều khiển của máy tính thực hiện một nhiệm vụ nào đó.

Tiến trình : Trạng thái động của chương trình.

-> 1 tiến trình có thê bao gồm nhiều luồng ( tiểu trình)

- Luồng (Thread) : là chuỗi (luồng) lệnh được thực hiện trong tiến trình

Tiến trình chỉ chứa 1 luồng : (Heavyweight Process)

Tiến trình gồm nhiều luồng : (Lightweighy Process)

Mô hình đa luồng : (Multi\_Threading)

Các luồng hoạt động song song, chia sẻ biến toàn cục của tiến trình.

- Tài nguyên hệ thống : Là tất cả những gì cần thiết cho một tiến trình có thể thực hiện được.

Không gian : không gian lưu trữ của hệ thống

Thời gian : thời gian thực hiện lệnh/truy xuất dữ liệu.

\*Tài nguyên hệ thống bao gồm :

+ Bộ nhớ : phân biệt bộ nhớ (vùng vật lý chứa dữ liệu) và truy cập tới bộ nhớ (quá trình tìm đến dữ liệu trên

bộ nhớ)

+ Processor (Vi xử lý) :

+ Thiết bị ngoại vi

\*Phân loại tài nguyên :

Kiểu tài nguyên :

+ Tài nguyên vật lý : các thiết bị vật lý

+ Tài nguyên logic : biến nhớ, thiết bị ảo

Khả năng dùng chung :

Tài nguyên ảo : VD Máy in ảo

- Bộ xử lý lệnh (Shell)

Tiến trình đặc biệt : nơi giao tiếp giữa người dùng và HĐH

Nhiệm vụ : nhận lệnh của người sử dụng, phân tích lệnh, phát sinh tiến trình mới để thực hiện yêu cầu của lệnh.

Nhận lệnh : thông qua cơ chế dòng lệnh

- Nhân hệ thống (Kernel)

User -> Shell -> Kernel -> Hardware

Là một chương trình máy tính ở mức thấp nhất, giao tiếp với phần cứng, điều khiển mọi thứ khác, nó là hạt nhân

trái tim của HĐH. Chuyên đổi các yêu cầu đầu vào/đầu ra thành một tập lệnh cho CPU

**5. Cấu trúc HĐH**

5.1 Các thành phần của hệ thống

- Quản lý tiến trình

- Quán lý bộ nhớ chính

- Quản lý hệ thống vào ra

- Quản lý file

- Quản lý bộ nhớ lưu trữ

- Hệ thống trao đổi dữ liệu ( hệ thống phân tán)

- Bảo vệ hệ thống

- Giao diện người dùng

5.2 Dịch vụ HĐH

Dịch vụ chính yếu :

- Thi hành chương trình

- Thao tác nhập xuất

- Thao tác trên hệ thống tập tin

- Truyền thông

- Phát hiện lỗi

Dịch vụ phụ trợ :

- Cung cấp tài nguyên

- Thống kê báo cáo

- Bảo vệ

5.3 Lời gọi hệ thống : tạo môi trường giao tiếp giữa chương trình của người sử dụng và HĐH

5.4 Các cấu trúc hệ thống

**6. Tính chất cơ bản của HĐH**

- Độ tin cậy cao : mọi hoạt động/thông báo của HĐH phải chuẩn xác

- An toàn : dữ liệu và chương trình phải được bảo vệ

- Hiệu quả : các tài nguyên phải được khai thác triệt để, hệ thống cần phải duy trì được tính đồng bộ

- Tổng quát theo thời gian : hệ thống phải có tính kế thừa, hệ thống cần phải có khả năng thích nghi với

những thay đổi có thể xảy ra

- Thuận tiện : dễ sử dụng, nhiều mức với hiệu quả khác nhau, có hệ thống trợ giúp phong phú

**7. Vấn đề xây dựng HĐH**

Nguyên tắc xây dựng hệ điều hành :

- Nguyên tắc modul

- Nguyên tắc tương đối trong định vị

- Nguyên tắc macroprocessor

- Nguyên tắc khởi tạo trong cài đặt

- Nguyên tắc lặp chức năng

- Nguyên tắc giá trị chuẩn

- Nguyên tắc bảo vệ nhiều mức

**CHƯƠNG 2 : QUẢN LÝ TIẾN TRÌNH**

\*Tiến trình : Chương trình đang được thực hiện được cung cấp tài nguyên khi bắt đầu và trong khi thực hiện gọi là tiến trình

Hệ thống là tập các tiến trình đồng thời :

+ Tiến trình hệ điều hành : Thực hiện mã lệnh hệ thống

+ Tiến trình người dùng : Thực hiện mã lệnh người dùng

Trách nhiệm của HĐH : Đảm bảo hoạt động của tiến trình và tiểu trình (luồng)

**1. Tiến trình (Process)**

\*Khái niệm tiến trình

- K/n : 1) là một dãy các chuyên đổi trạng thái của hệ thống, chuyển từ trạng thái này sang trạng thái khác được thực hiện theo yêu

cầu nằm trong chương trình của người sử dụng, được xuất phát từ một trạng thái ban đầu.

2) Là chương trình đang được thực hiện (tài nguyên tối thiểu cần có : bộ nhớ cho mã chương trình và dữ liệu, các thanh ghi

của Processor phục vụ cho quá trình thực hiện các chương trình).

Tiến trình >< chương trình

Chương trình : là một thực thể thụ động. Bao gồm : mã chương trình, Dữ liệu(biến được lưu trữ và sử dụng trong bộ nhớ),

thư viện liên kết động. Khi chương trình thực hiện, tài nguyên tối thiểu cần có là bộ nhớ và các thanh ghi.

Tiến trình : là thực thể chủ động (bộ đếm lệnh, tập tài nguyên)

- Dịch và thực hiện một chương trình : HĐH khởi tạo một tiến trình và phân phối vùng nhớ cho nó.

Bắt đầu thực hiện tại \_Start() (hàm này gọi tới hàm main()), khi hàm main() kết thúc OS gọi tới hàm \_exit() để hủy bỏ tiến trình

và thu hồi tài nguyên. => Tiến trình là chương trình đang thực thi

- Trạng thái của tiến trình : Khi thực hiện tiến trình thay đổi trạng thái, có 5 trạng thái của tiến trình

+ Khởi tạo (New) : Tiến trình đang được khởi tạo

+ Sẵn sàng (Ready) : Tiến trình đang đợi sử dụng Process vật lý

+ Thực hiện (Running) : Các câu lệnh của tiến trình đang được thực hiện

+ Chờ đợi ( Waiting) : Tiến trình đang chờ đợi một sự kiện nào đó xuất hiện (sự hoàn thành thao tác vào ra)

+ Kết thúc (Terminated) : Tiến trình thực hiện xong

=> Trạng thái của tiến trình là một phần trong hoạt động hiện tại của tiến trình

- Lưu đồ thay đổi trạng thái của tiến trình (Slide 14/340 Chương 2)

Hệ thống có 1 Processor :

+ Có duy nhất một tiến trình ở trạng thái thực hiện

+ Có thể có nhiều tiến trình ở trạng thái chờ đợi hoặc sẵn sàng

- Khối điều khiển tiến trình PCB (Process Control Block)

Mỗi tiến trình được thể hiện trong một hệ thống bởi một khối điều khiển tiến trình

PCB : cấu trúc thông tin cho phép xác định duy nhất một tiến trình

- Tiến trình đơn luồng và tiến trình đa luồng

Tiến trình đơn luồng : Là tiến trình thực hiện chỉ một luồng thực thi

=> Cho phép thực hiện 1 nhiệm vụ tại 1 thời điểm

Tiến trình đa luồng : Là tiến trình có nhiều luồng thực thi

=> Cho phép thực hiện nhiều hơn một nhiệm vụ tại một thời điểm

\*Điều phối tiến trình

#Mục đích : Sử dụng tối đa thời gian của CPU

=> Cần có nhiều tiến trình trong hệ thống

#Vấn đề : Luân chuyển CPU giữa các tiến trình

=> Phải có hàng đợi cho các tiến trinh

- Hệ thống có nhiều hàng đợi dành cho tiến trình (Slide 21/340):

+ Job-queue : Tập các tiến trình trong hệ thống

+ Ready-queue : Tập các tiến trình tồn tại trong bộ nhớ, đang sẵn sàng và chờ đợi để được thực hiện

+ Devices queue : Tập các tiến trình đang chờ đợi một thiết bị vào ra. Phân biệt hàng đợi cho từng thiết bị

- Tiến trình mới tạo, được đặt trong hàng đợi sẵn sàng và đợi cho tới khi được lựa chọn để thực hiện

- Tiến trình đã được chọn và đang thực hiện :

1) Đưa ra một y/c vào ra : đợi trong một hàng đợi thiết bị

2) Tạo một tiến trình con và đợi tiến trình con kết thúc

3) Hết thời gian sử dụng CPU phải quay lại hàng đợi sẵn sàng

- Tiến trình tiếp tục chu kì (sẵn sàng, thực hiện, chờ đợi) cho tới khi kết thúc

- Bộ điều phối ( Scheduler) : lựa chọn tiến trình trong các hàng đợi

+ Điều phối công việc (Job scheduler, Long-term scheduler)

=> Vấn đề : Các chương trình thiên về vào/ra -> ít thời gian CPU, các tiến trình thiên về tính toán -> nhiều thời gian

của CPU

+ Điều phối CPU (CPU scheduler, Short-term scheduler) : Job Scheduler sẽ sắp xếp lại vị trí các Job rồi gửi sang Process

Scheduler (Slide 32/340)

+ Chuyển ngữ cảnh ( Context switch)

+ Hoán đổi tiến trình (Medium-term Scheduler)

=> Nhiệm vụ : đưa tiến trình ra khỏi bộ nhớ, sau đó đưa tiến trình quay lại và tiếp tục thực hiện

=> Mục đích : giải phóng bộ nhớ, tạo vùng nhớ tự do rộng hơn

\*Thao tác trên tiến trình

(+) Tạo tiến trình

- Tiến trình có thể tạo nhiều tiến trình mới cùng hoạt động (CreateProcess(), fork())

+ tiến trình tạo : tiến trình cha

+ tiến trình được tạo : tiến trình con

- Tiến trình con có thể tạo nhiều tiến trình con khác -> Cây tiến trình

- Vấn đề phân phối tài nguyên

+ tiến trình con lấy từ HĐH

+ tiến trình con lấy từ tiến trình cha (tất cả, hoặc 1 phần)

- Vấn đề thực hiện

+ tiến trình cha tiếp tục thực hiện đồng thời với tiến trình con

+ tiến trình cha chờ tiến trình con kết thúc rồi thực hiện

(+) Kết thúc tiến trình

- Hoàn thành câu lệnh cuối và yêu cầu HĐH xóa nó (exit)

+ gửi trả dữ liệu tới tiến trình cha

+ các tài nguyên đã được cung cấp trả lại hệ thống

- Tiến trình cha có thể kết thúc sự thực hiện của tiến trình con

+ TT cha phải biết định danh của tiến trình con

+ Sử dụng lời gọi hệ thống (abort)

- Tiến trình cha kết thúc tiến trình con khi :

+ TT con sử dụng vượt quá tài nguyên được cấp

+ Nhiệm vụ cung cấp cho tiến trình con không còn cần thiết nữa

+ TT cha kết thúc và HĐH không cho phép TT con được thực hiện

=> Cascading termination

\*Hợp tác tiến trình

- Phân loại tiến trình

+ Các TT tuần tự : Điểm bắt đầu của TT này nằm sau điểm kết thúc của tiến trình kia

+ Các TT song song : Điểm bắt đầu của TT này nằm giữa điểm kết thúc của TT kia

-> Độc lập : Không bị ảnh hưởng hoặc bị ảnh hưởng bởi các TT khác trong hệ thống

-> Có hợp tác : Ảnh hưởng hoặc bị ảnh hưởng bởi các TT khác trong hệ thống

- Hợp tác tiến trình

+ Mục đích : Chia sẻ thông tin, tăng tốc độ tính toán, Module hóa, Tiện dụng

+ Đòi hỏi cơ chế cho phép : Truyền thông giữa các tiến trình, đồng bộ hóa hoạt động của các TT

- Bài toán người sản xuất (Producer) - Người tiêu thụ (Consumer)

Slide 47-48-49-50

\*Truyền thông liên tiến trình

- Trao đổi giữa các tiến trình

+ Mô hình bộ nhớ dùng chung : Các tiến trình chia sẻ vùng nhớ chính, Mã cài đặt được viết tường minh bởi người lập trình ứng dụng,

+ Mô hình truyền thông liên tiến trình (Interprocess communication) : Là cơ chế cho phép các TT truyền thông và đồng bộ các hoạt

động, thường được sử dụng trong các hệ phân tán khi các TT truyền thông nằm trên các máy khác nhau, đảm bảo bởi hệ thống truyền thông

điệp

- Hệ thống truyền thông điệp : Cho phép các tiến trình trao đổi với nhau thông qua các biến dùng chung

+ Yêu câu 2 thao tác cơ bản : Send(msg), Receive(msg)

+ Nếu 2 tiến trình P và Q muốn trao đổi, chúng cần : Thiết lập 1 liên kết truyền thông giữa chúng, Trao đổi các messages nhờ các

thao tác Send/Receive

- Hệ thống truyền thông điệp và bài toán Producer Comsumer

+ Truyền thông trực tiếp :

# Các TT phải gọi tên các TT nhận/gửi 1 cách tường minh

(Send(P,Message)) - gửi 1 thông báo tới tiến trình P

(Receive(Q,Message)) - nhận 1 thông báo từ tiến trình Q

# Tính chất của liên kết truyền thông : Được thiết lập tự động, 1 liên kết gắn với 1 cặp TT truyền thông, chỉ tồn tại 1 LK

giữa cặp tiến trình, liên kết là 1 chiều nhưng thường 2 chiều

+ Truyền thông gián tiếp : Casc thông điệp được gửi/nhận tới/từ các hòm thư (mailboxes), cổng (ports). Mỗi hòm thư có định danh

duy nhất , các TT có thể trao đổi nếu chúng dùng chung hòm thư.

# Tính chất các liên kết : Được thiết lập chỉ khi các TT dùng chung hòm thư, 1 LK có thể gắn với nhiều tiến trình, mỗi cặp

TT có thể dùng chung nheiefu liên kết truyền thông, liên kết có thể 1 hay 2 chiều.

# Các thao tác : Tạo hòm thư, Gửi/nhận thông báo qua hòm thư : Send(A,msg), Receive(A,msg), Hủy bỏ hòm thư.

- Vấn đề đồng bộ hóa :

+ Truyền thông điệp có thể phải chờ đợi (Blocking) hoặc không chờ đợi (non Blocking) : Blocking - truyền thông đồng bộ, non

Blocking - truyền thông không đồng bộ.

+ Các thủ tục send() và receive() có thể phải chờ đợi hoặc không.

# Blocking Send() : TT gửi thông báo và chờ tới khi msg được nhận bởi TT nhận hoặc bởi hòm thư.

# non Blocking Send() : TT gửi thông báo xong và tiếp tục làm việc ( chả cần biết thông báo gửi thế nào)

# Blocking receive() : TT nhận phải đợi cho tới khi có thông điệp

# non Blocking receive() : TT nhận trả về một thông điệp có giá trị, hoặc 1 giá trị null

**4.Tài nguyên găng và điều độ tiến trình**

4.1 Khái niệm tài nguyên găng

\* Khái niệm :

- Là những tài nguyên hạn chế về khả năng dùng chung

- Cần đồng thời cho nhiều tiến trình

- Có thể là thiết bị vật lí hay là dữ liệu dùng chung

=> Vấn đề : Dùng chung tài nguyên găng có thể dẫn tới việc không đảm bảo tính toàn vẹn dữ liệu -> Đòi hỏi cơ chế đồng bộ hóa

các tiến trình

\* Điều kiện cạnh tranh : Nhiều TT cùng truy cập vào trật tự của các truy nhập -> Làm cho chương trình không xác định

\* Ngăn ngừa ĐK cạnh tranh :

- Chỉ 1 TT truy nhập tới dữ liệu dùng chung tại 1 thờ điểm

- Đoạn lệnh truy nhập tới dữ liệu dùng chung trong các TT phải thực hiện theo thứ tự xác định

\* Đoạn găng :

- Là chương trình sử dụng tài nguyên găng ( Đoạn c.trình dùng truy nhập và thao tác trên dữ liệu dùng chung)

- Khi có nhiều TT sử dụng tài nguyên găng thì phải điều độ

-> Mục đích : đảm bảo chỉ có 1 TT tại 1 thời điểm sử dụng tài nguyên găng ( không có quá 1 TT nằm trong đoạn găng)

\* Yêu cầu của chương trình điều độ :

- Loại trừ lẫn nhau : Mỗi thời điểm, tài nguyên găng không phải phục vụ 1 số lượng TT vượt quá khả năng của nó

-> VD : 1 TT đang trong đoạn găng thì không TT nào khác được vào đoạn găng

- Tiến triển : Nếu tài nguyên găng còn khả năng phục vụ mà vẫn có TT muốn vào đoạn găng thì TT đó phải được sử dụng tài nguyên

găng

- Chờ đợi hữu hạn : Nếu tài nguyên găng hết khả năng phục vụ mà có tiến trình muốn vào đoạn găng thì phải xếp vào hàng đợi và

chờ đợi trong thời gian hữu hạn để được vào đoạn găng

\* Quy ước : [?] - cứ thấy thế nào ý ? =)))

\* Các phương pháp điều độ :

- Phương pháp khóa trong

- Phương pháp kiểm tra và xác lập

- Kỹ thuật đèn báo

- Monitor

4.2 Phương pháp khóa trong

\* Nguyên tắc :

- Mỗi TT dùng 1 byte trong bộ nhớ CHUNG để làm khóa : TT vào đoạn găng -> đóng khóa, TT thoát khỏi đoạn găng -> mở khóa, TT muốn

vào đoạn găng phải kiểm tra khóa của TT còn lại, đang khóa -> đợi, khóa mở -> được quyền vào đoạn găng

\* Thuật toán điều độ & Thuật toán Dekker

4.3 Phương pháp kiểm tra và xác lập

\* Nguyễn tắc :

- Sử dụng sự hỗ trợ từ phần cứng

- Phần cứng cung cấp các câu lệnh xử lí không tách rời

-> Xử lí không tách rời là gì ? Khối lệnh không thể bị ngắt trong khi đang thực hiện, khối lệnh gọi đồng thời sẽ theo một thứ tự

bất kì

\* Thuật toán với TestAndSet và thuật toán với Swap

4.4 Kĩ thuật đèn báo (Semaphore)

\* Khái niệm :

- Là một biến nguyên S, khởi tạo bằng số khả năng phục vụ của tài nguyên nó điều độ

- Chỉ có thể thay đổi giá trị bằng 2 thao tác cơ bản P và V

+ Thao tác P(S) ( Wait(S) -> S--)

+ Thao tác V(S) ( Sign(S) -> S++)

- Đèn báo là công cụ điều độ tổng quát

\* Nhận xét :

- Dễ dàng cho hệ thống phức tạp

- Không tồn tại hiện tượng chờ đợi tích cực

- Hiệu quả sử dụng phụ thuộc vào người dùng

- Các phép xử lí PS, VS không phân chia được

=> Bản thân PS, VS cũng là 2 tài nguyên găng => cần được điều độ

4.5 Đồng bộ tiến trình

Các ví dụ về đồng bộ tiến trình, các bài toán :

- Bài toán người sản xuất - người tiêu thụ ( Producer - Consumer)

- Bữa ăn tối của triết gia ( Dining Philosophers)

- Người đọc và biên tập viên ( Readers - Writers)

- Người thợ cắt tóc ngủ gật ( Sleeping Barber)

- Bathroom Problem

- Đồng bộ theo Barriers

4.6 Công cụ điều độ cao cấp

\* Giới thiệu : là một kiểu dữ liệu đặc biệt, được đề xuất bởi HOARE 1974. Bao gồm các thủ tục,

dữ liệu cục bộ, đoạn mã khởi tạo.