

BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2024 – 2025



Sharing is learning



 **BAN HỌC TẬP**

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

 **CONTACT**

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit

 **TEAM TIẾNG ANH**

english.with.bht@gmail.com

 *creative.owl.se*

 *english.with.bht*

TRAINING

HỆ ĐIỀU HÀNH

 **Thời gian:** 19:00 thứ Năm ngày 17/10/2024

 **Địa điểm:** Microsoft Teams – Code: w2dsy1q

 **Trainers:** Nguyễn Quý Phong – KTPM2023.2

Lê Ngô Thanh Toàn – KHMT2023.4



Sharing is learning

MỤC LỤC

Chương 1. Tổng quan hệ điều hành

Chương 2. Cấu trúc hệ điều hành

Chương 3. Quản lý tiến trình

Chương 4. Định thời CPU



Sharing is learning

CHƯƠNG 1. TỔNG QUAN HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản về hệ điều hành

II. Phân loại hệ điều hành



Sharing is learning

CHƯƠNG 1. TỔNG QUAN HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản

Khái niệm:

- Là chương trình trung gian giữa phần cứng và người sử dụng.
- Điều khiển và phối hợp việc sử dụng phần cứng.
- Cung cấp các dịch vụ cơ bản cho các ứng dụng.

Mục tiêu:

- Giúp người dùng dễ dàng sử dụng hệ thống.
- Quản lý và cấp phát tài nguyên hệ thống hiệu quả.



CHƯƠNG 1. TỔNG QUAN HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản

Lợi ích:

- Quản lý phần cứng máy tính.
- Kết nối các thiết bị phần cứng với nhau.
- Tương tác giữa các chương trình với nhau và với phần cứng.
- Là nơi để user cài đặt các chương trình ứng dụng.
- Cung cấp giao diện cho người dùng..



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản

- Chức năng:

Phân chia thời gian xử lý và định thời CPU.

Phối hợp và đồng bộ hoạt động giữa các processes.

Quản lý tài nguyên hệ thống.

Kiểm soát truy cập, bảo vệ hệ thống.

Duy trì sự nhất quán của hệ thống, kiểm soát lỗi và phục hồi.

Cung cấp giao diện làm việc cho users.

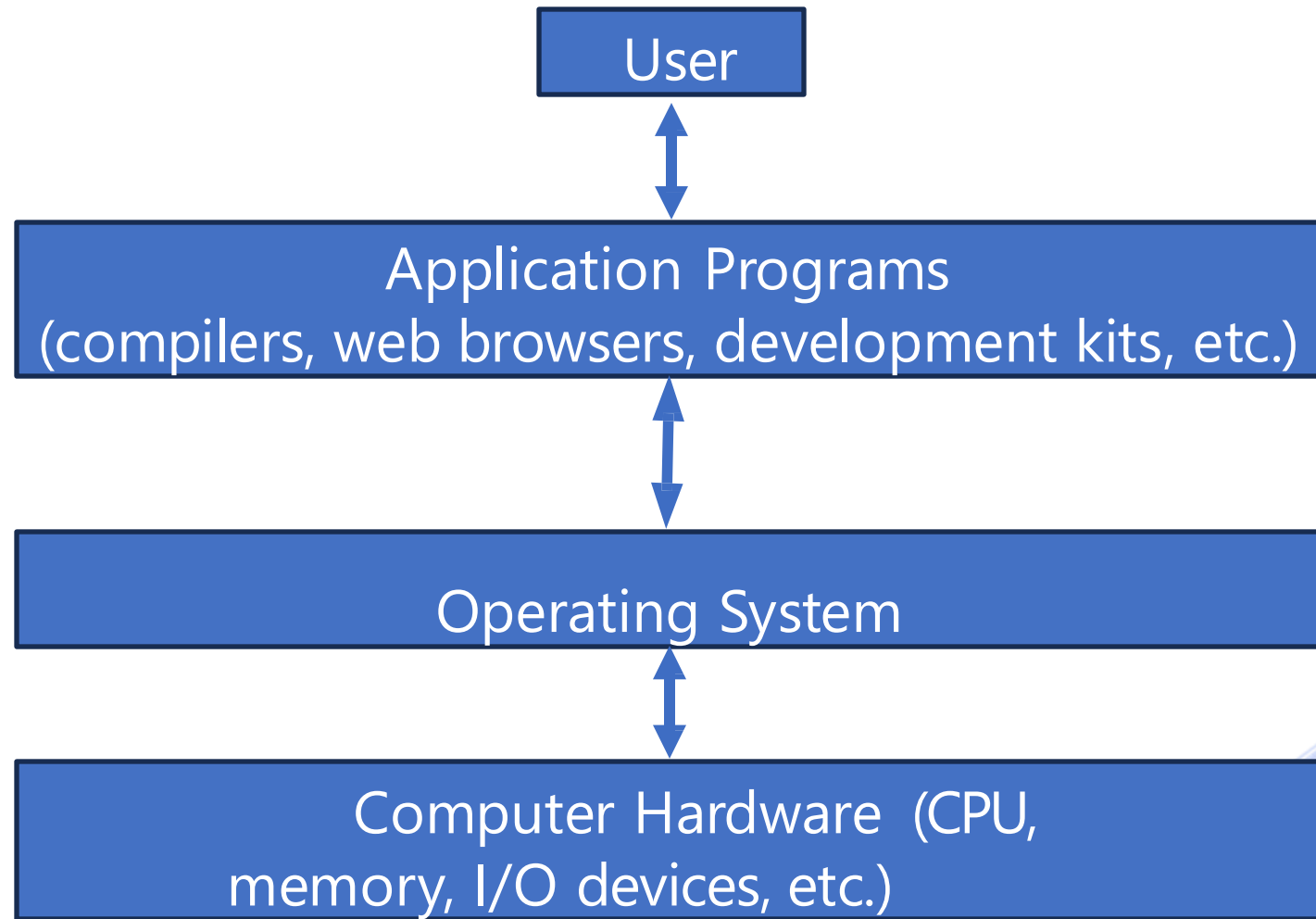


Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản

– Cấu trúc hệ thống:



Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản

1. Dưới góc độ cơ bản, hệ điều hành được định nghĩa là:

- A. Là một phần mềm chạy trên máy tính.
- ☒ B. Là một chương trình quản lý phần cứng máy tính.
- C. Là một chương trình bảo vệ phần cứng máy tính.
- D. Là một phần mềm quản lý các phần mềm khác.

3. Chức năng của hệ điều hành là gì?

- A. Cấp phát tài nguyên phần cứng cho các ứng dụng.
- B. Điều khiển, định thời thực thi các chương trình.
- C. Hỗ trợ người dùng giao tiếp với máy tính.
- ☒ D. Tất cả các tính năng trên.

2. Trong phân lớp hệ thống máy tính. Hệ điều hành thuộc vị trí nào:

- A. Hệ điều hành thuộc lớp cuối cùng, kế trên là lớp phần cứng.
- B. Hệ điều hành thuộc lớp trên cùng, phía dưới là lớp ứng dụng.
- ☒ C. Hệ điều hành nằm giữa lớp phần cứng và lớp ứng dụng.
- D. Hệ điều hành nằm giữa lớp phần cứng và lớp người dùng.

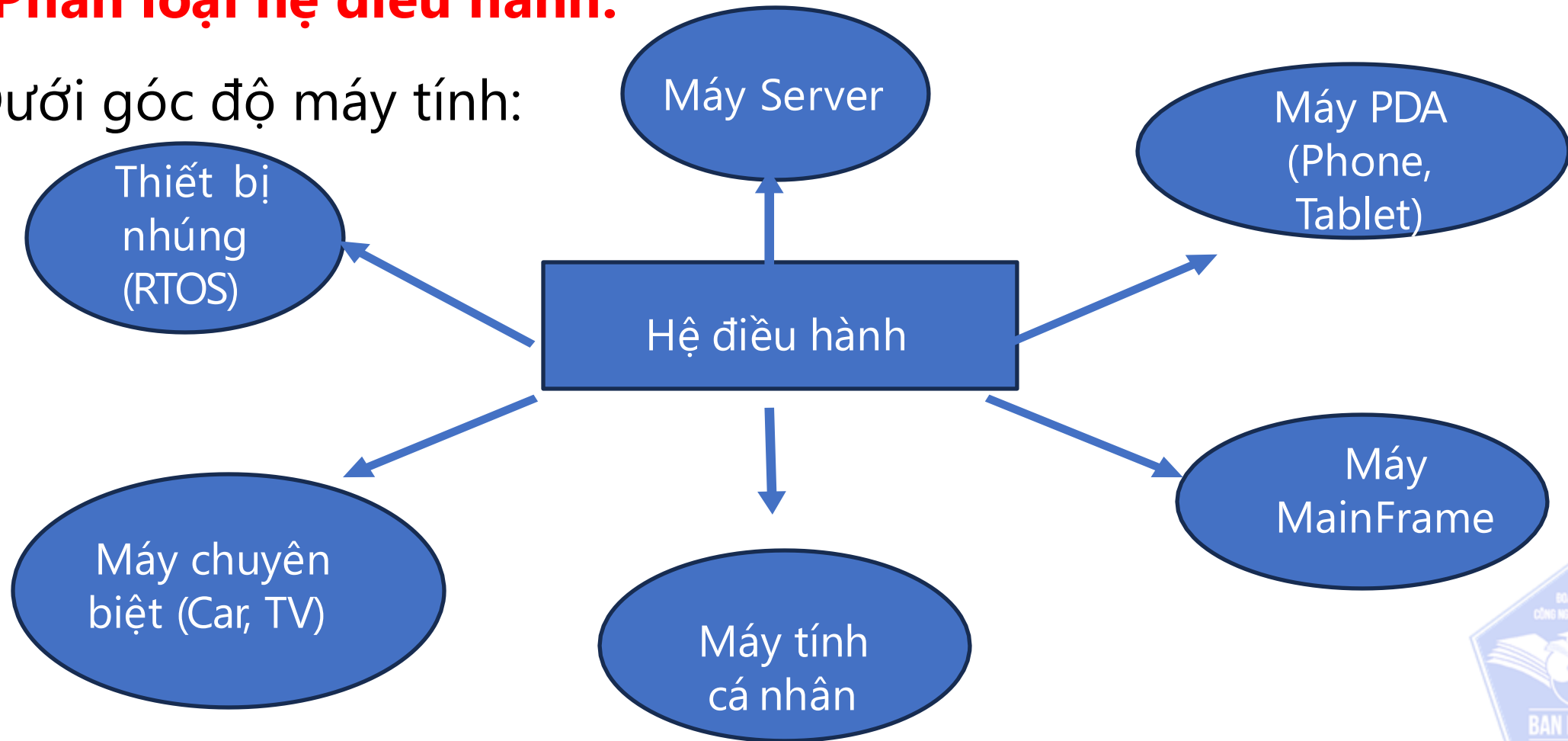


Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành:

– Dưới góc độ máy tính:

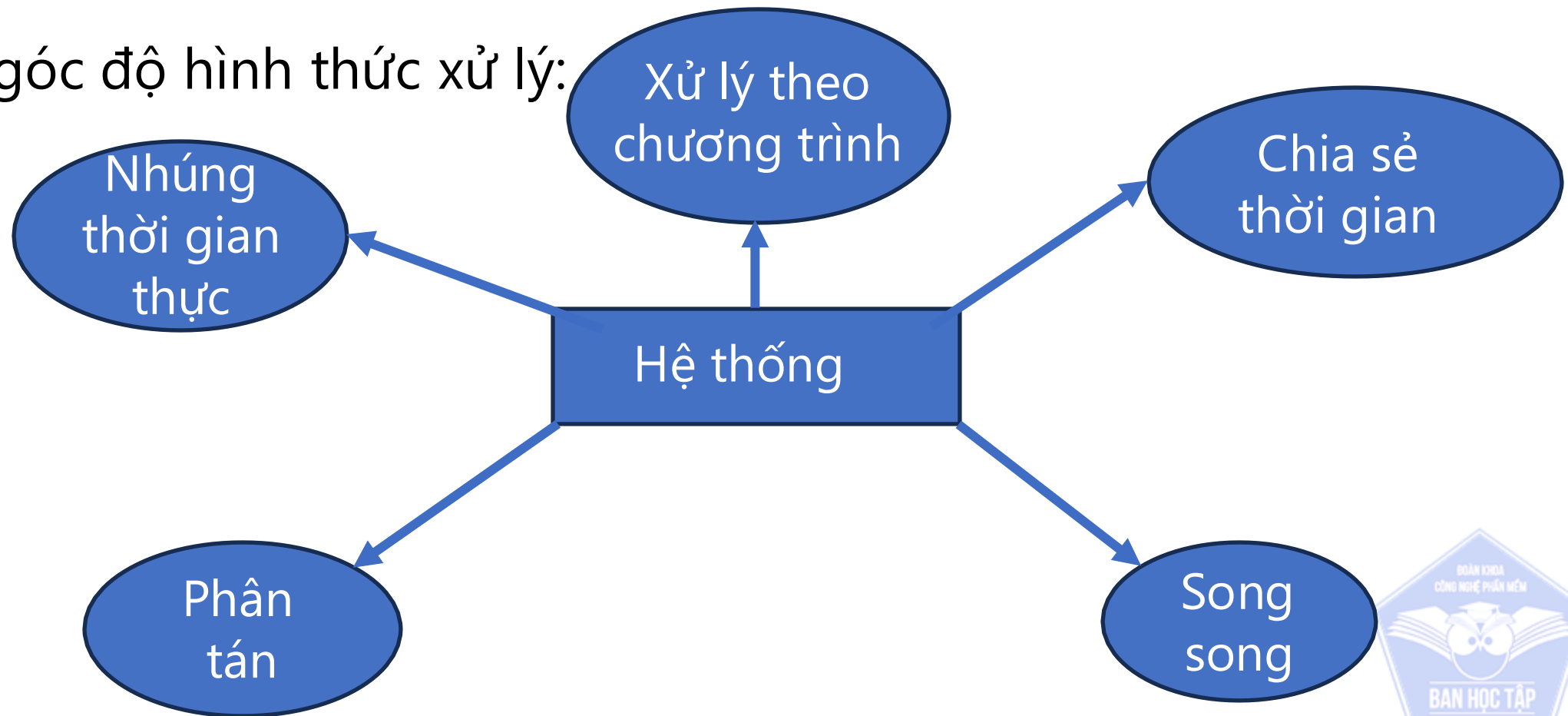


Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành:

– Dưới góc độ hình thức xử lý:



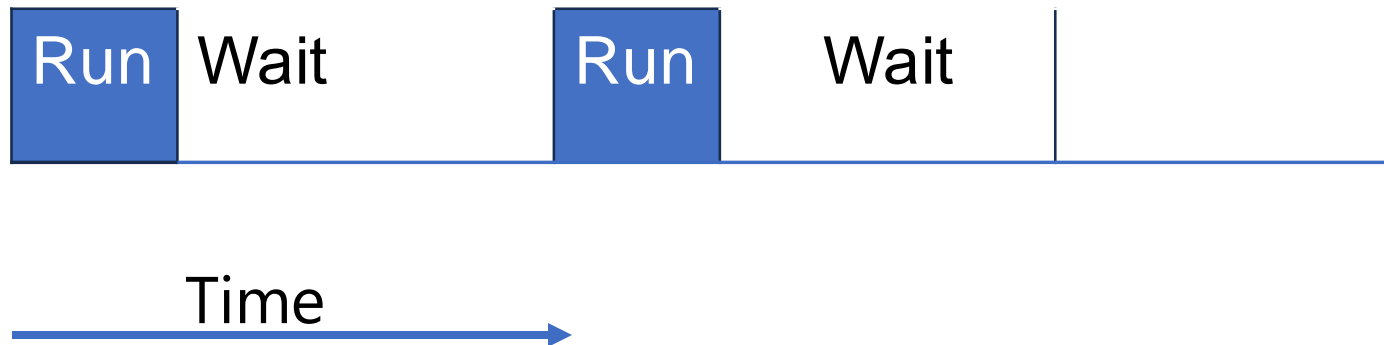
Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

a) Hệ thống đơn chương:

- Tác vụ thi hành **tuần tự**.
- Bộ giám sát **thường trực**.



Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

b) Hệ thống đa chương:

- Nhiều công việc được nạp đồng thời vào bộ nhớ chính.
 - Khi một tiến trình thực hiện I/O, một tiến trình khác được thực thi.
- > **Tận dụng được thời gian rảnh, tăng hiệu suất sử dụng CPU.**
- Lập lịch CPU
 - Quản lý bộ nhớ
 - Cấp phát tài nguyên



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

c) Hệ thống chia sẻ thời gian (Multi-tasking):

- Mở rộng của hệ thống đa chương.
- **Mỗi công việc chạy trong khoảng thời gian nhất định.**
- Các yêu cầu: tương tự hệ thống đa chương. Ngoài ra:
 - + Đồng bộ giữa các công việc.
 - + Giao tiếp giữa các công việc.
 - + Tránh deadlock.
- Quản lý hệ thống file, hệ thống lưu trữ.



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành



Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành:

d) Hệ thống song song(đa xử lý):

- Các xử lý chia sẻ bộ nhớ với nhau.
- Các công việc được các bộ xử lý thực hiện đồng thời

+ Ưu điểm:

- + Năng suất cao (các công việc được các bộ xử lý thực hiện đồng thời).
- + Sự hỏng hóc của một bộ xử lý không ảnh hưởng đến toàn bộ hệ thống.



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

d) Hệ thống song song(đa xử lý):

Đa xử lý đối xứng	Đa xử lý bất đối xứng
<ul style="list-style-type: none">•Mỗi process vận hành một bản sao hệ điều hành giống nhau.•Copy dữ liệu cho nhau khi cần.	<ul style="list-style-type: none">•Mỗi process thực thi một công việc khác nhau.•Bộ xử lý chính định thời và phân công việc cho các bộ xử lý khác.



Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

e) Hệ thống phân tán:

- Mỗi processor có bộ nhớ riêng, giao tiếp với nhau qua các kênh nối như mạng,...
- Người dùng chỉ thấy một hệ thống đơn nhất.
- Phân làm 2 loại:
 - + Client-sever
 - + Peer-to-peer



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

f) Hệ thống nhúng thời gian thực:

- Hệ thống cho **kết quả chính xác trong thời gian nhanh nhất**
- Điều khiển công nghiệp, thử nghiệm khoa học, quân sự...
- **Hard real-time:**
 - + Yêu cầu thời gian xử lý, **đáp ứng nghiêm ngặt**
 - + **Giới hạn** bộ nhớ.
- **Soft real-time:**
 - + Công việc thực hiện theo **độ ưu tiên**.
 - + Dùng trong lĩnh vực multimedia, virtual reality,...



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

1.Đâu là ưu điểm chính của hệ thống xử lý đa chương (multiprogramming system)?

- A. Chương trình khi nạp vào bộ nhớ sẽ được xử lý hoàn thành ngay lập tức.
- ☒ B. Hệ thống chạy được nhiều chương trình cùng lúc.
- C. Không cần thiết lập định thời công việc (job scheduling) và quản lý bộ nhớ.
- D. Tối ưu sử dụng bộ nhớ.

2.Trong các mô hình hệ điều hành dưới đây, loại dùng cho hệ thống có nhiều bộ xử lý cùng chia sẻ hệ thống đường truyền, dữ liệu, bộ nhớ, các thiết bị ngoại vi?

- A. Hệ thống xử lý đa chương.
- B. Hệ thống xử lý đa nhiệm.
- ☒ C. Hệ thống xử lý song song.
- D. Hệ thống xử lý thời gian thực.



Sharing is learning

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

1. Hệ thống xử lý phân tán được phân loại:

- A. Đồng bộ và bất đồng bộ.
- ☒ B. Peer-to-peer và client-server.
- C. Kết hợp và không kết hợp.
- D. Đối xứng và bất đối xứng.

2. Mục đích chính của hệ thống xử lý đa chương (multiprogramming system) là gì?

- A. Thực hiện đồng thời nhiều chương trình.
- ☒ B. Tận dụng thời gian nhàn rỗi của CPU.
- C. Chia sẻ thời gian giữa các chương trình.
- D. Tận dụng RAM, ROM khi đọc ghi.

3. Phát biểu nào sau đây KHÔNG ĐÚNG với hệ thống chia sẻ thời gian (time-sharing)?

- A. time-sharing là một hệ thống đa nhiệm (multi-tasking).
- B. Time-sharing yêu cầu thời gian chuyển đổi giữa các tác vụ rất ngắn.
- C. Time-sharing yêu cầu phải định thời CPU.
- ☒ D. Time-sharing yêu cầu hoàn thành xong nhiệm vụ 1 mới chia sẻ cho nhiệm vụ 2.

4. Hệ thống song song được phân loại như thế nào?

- ☒ A. Đa xử lý đối xứng và bất đối xứng.
- B. Đơn chương và đa chương.
- C. Client-server và peer-to-peer.
- D. Hard real-time và soft real-time.



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

- I. Các thành phần của hệ điều hành**
- II. Các dịch vụ hệ điều hành cung cấp**
- III. Lời gọi hệ thống**
- IV. Các chương trình hệ thống**
- V. Cấu trúc hệ thống**



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

II. Các dịch vụ hệ điều hành cung cấp

III. Lời gọi hệ thống

IV. Các chương trình hệ thống

V. Cấu trúc hệ thống



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

- ✓ Quản lý tiến trình
- ✓ Quản lý bộ nhớ chính
- ✓ Quản lý file
- ✓ Quản lý hệ thống I/O
- ✓ Quản lý hệ thống lưu trữ thứ cấp
- ✓ Hệ thống bảo vệ
- ✓ Hệ thống thông dịch lệnh



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

✓ Quản lý tiến trình

- Tiến trình khác chương trình như thế nào?
- Các nhiệm vụ chính:
 - Tạo và hủy tiến trình
 - Tạm dừng/ thực thi tiếp tiến trình
 - Đồng bộ hoạt động các tiến trình
 - Giao tiếp giữa các tiến trình
 - Khống chế tắc nghẽn



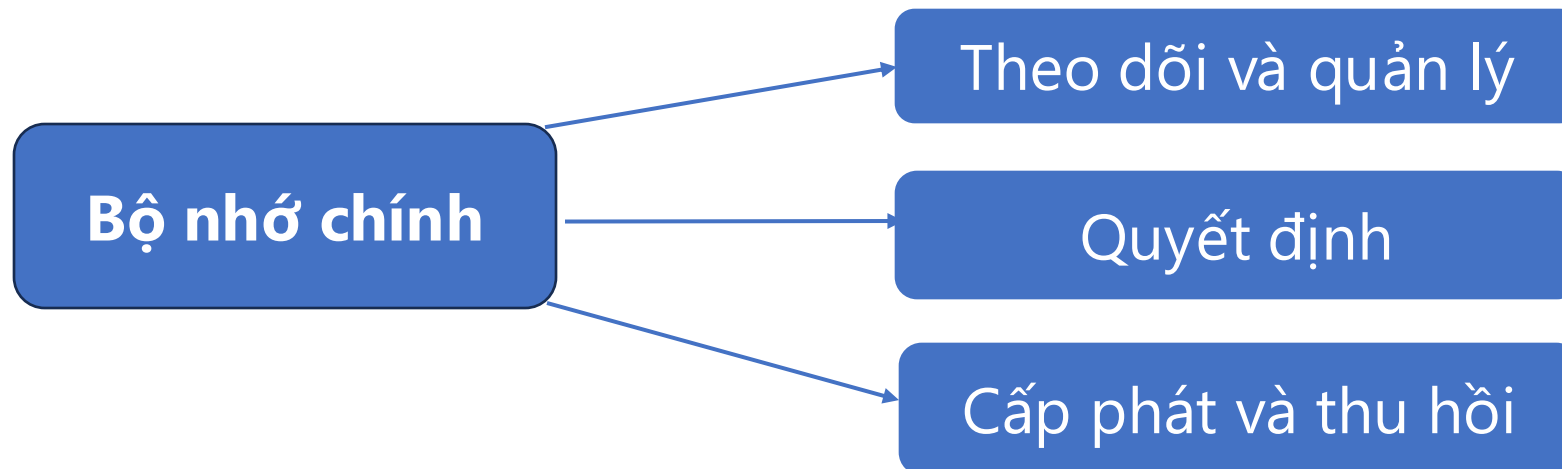
Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

✓ Quản lý bộ nhớ chính

- Bộ nhớ chính là trung tâm của các thao tác, xử lý.
- Để nâng cao hiệu suất sử dụng CPU, hệ điều hành cần quản lý bộ nhớ thích hợp



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

✓ Quản lý file

Hệ thống file: file và thư mục.

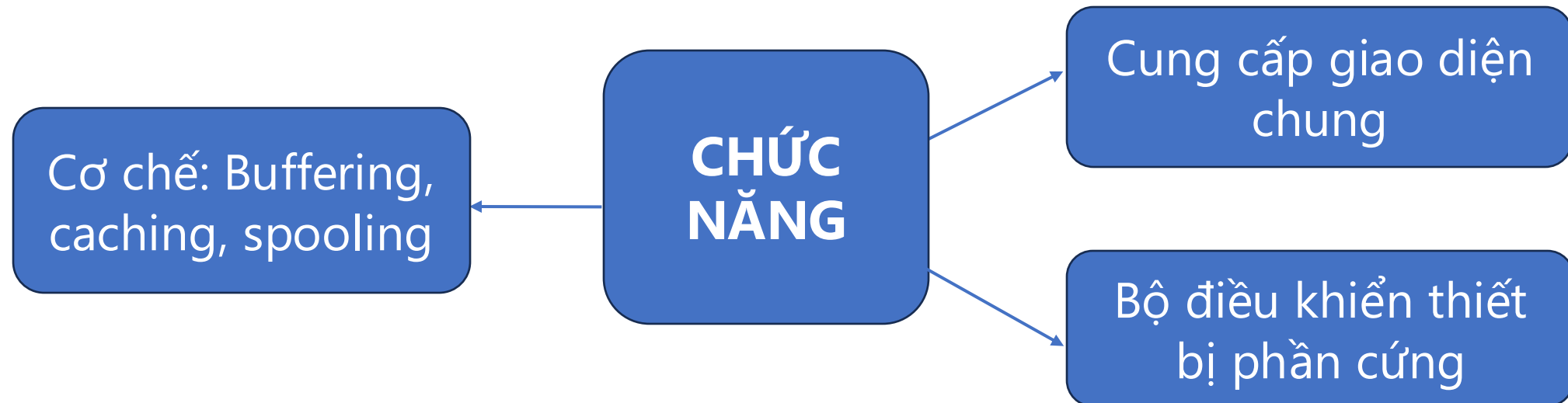


Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

- ✓ **Quản lý hệ thống I/O:** Che dấu sự khác biệt của các thiết bị I/O trước người dùng



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

✓ Quản lý hệ thống lưu trữ thứ cấp

- Lí do cần hệ thống lưu trữ thứ cấp: bộ nhớ chính nhỏ, không bền vững.
- Phương tiện lưu trữ: đĩa từ, đĩa quang (HDD, SDD).
- Các nhiệm vụ của HĐH trong quản lý đĩa:
 - Quản lý không gian trống trên đĩa
 - Cấp phát không gian lưu trữ
 - Định thời hoạt động trên đĩa

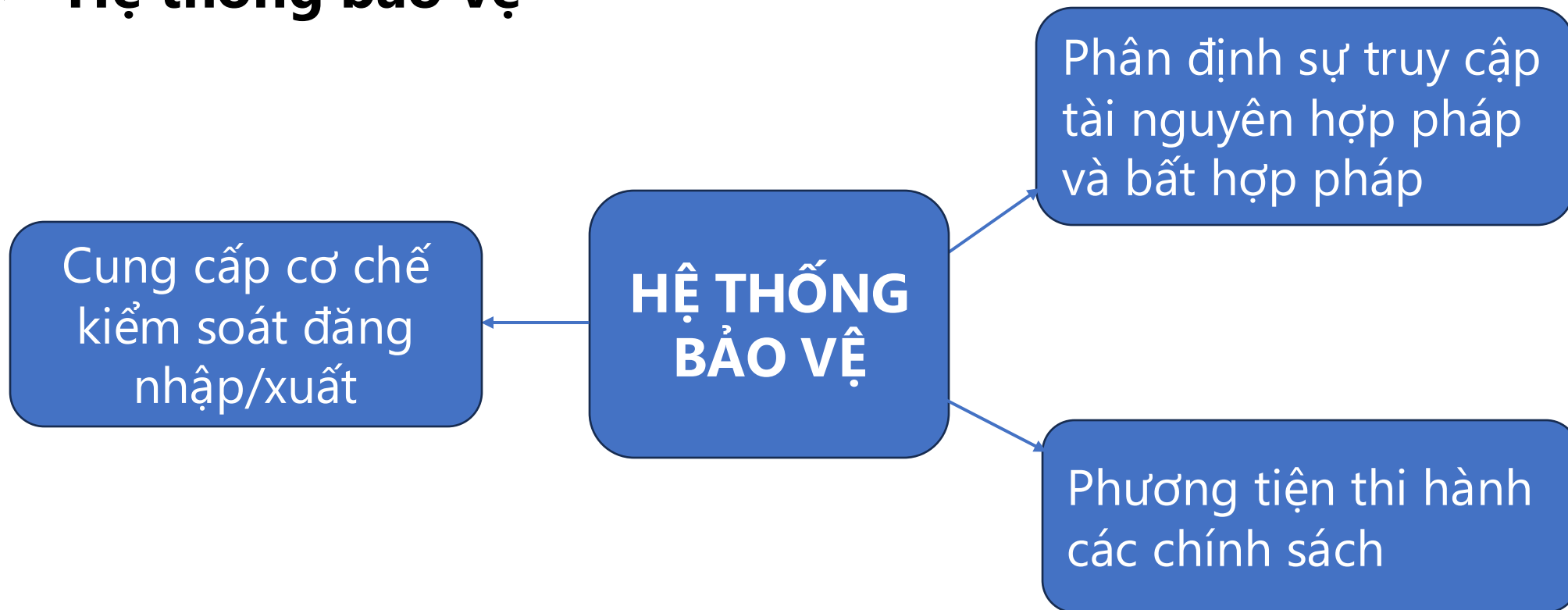


Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

✓ Hệ thống bảo vệ



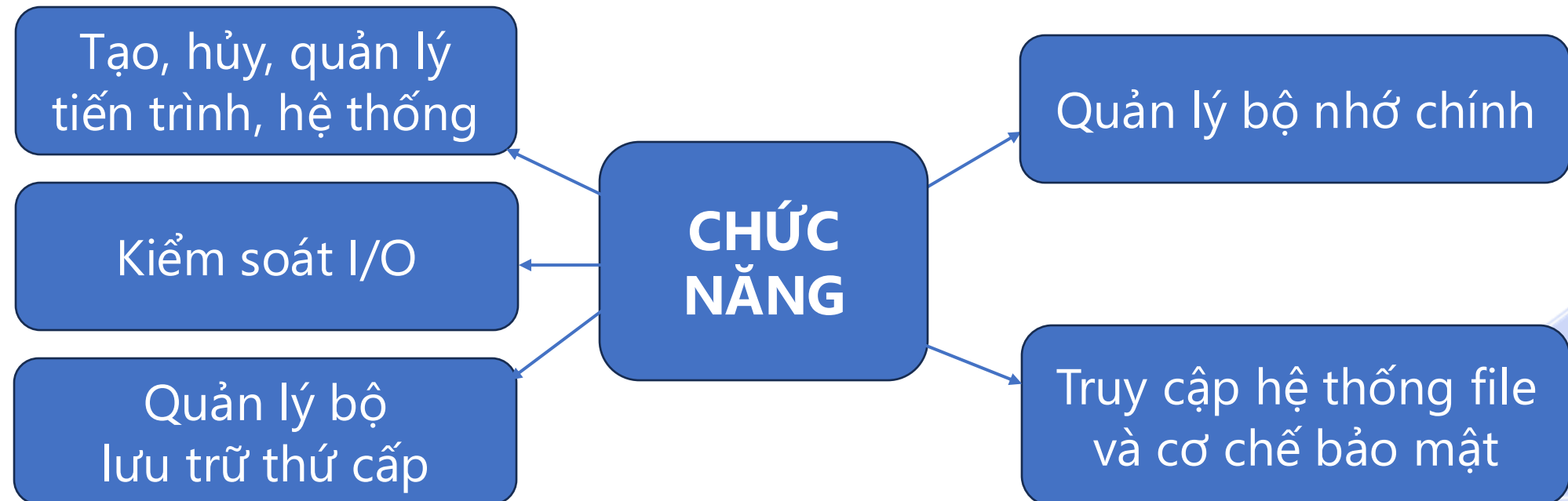
Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

✓ Hệ thống thông dịch lệnh

- Là giao diện chủ yếu giữa người dùng và OS.



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

II. Các dịch vụ hệ điều hành cung cấp

III. Lời gọi hệ thống

IV. Các chương trình hệ thống

V. Cấu trúc hệ thống



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

II. Các dịch vụ hệ điều hành cung cấp

- Thực thi chương trình
- Thực hiện các thao tác I/O theo yêu cầu của chương trình
- Các thao tác trên hệ thống file
- Trao đổi thông tin giữa các tiến trình qua hai cách:
 - Chia sẻ bộ nhớ (Shared memory)
 - Chuyển thông điệp (Message passing)
- Phát hiện lỗi (trong CPU, thiết bị I/O, ...)
- Cung cấp giao diện

Ngoài ra: cấp phát tài nguyên, kế toán, bảo vệ, an ninh, ...



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

II. Các dịch vụ hệ điều hành cung cấp

III. Lời gọi hệ thống

IV. Các chương trình hệ thống

V. Cấu trúc hệ thống



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

III. Lời gọi hệ thống

- Giao tiếp giữa tiến trình và HĐH.
- Cung cấp giao diện giữa tiến trình và HĐH.
- Thường ở dạng thư viện nhị phân hoặc hợp ngữ.
- Thông thường được viết bằng ngôn ngữ cấp cao (C hoặc C++) và hầu hết được truy cập thông qua các **Application Programming Interface (API)**.
- Ba phương pháp truyền tham số: qua stack, thanh ghi và một vùng nhớ, địa chỉ của vùng nhớ được gửi đến HĐH qua thanh ghi.



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

II. Các dịch vụ hệ điều hành cung cấp

III. Lời gọi hệ thống

IV. Các chương trình hệ thống

V. Cấu trúc hệ thống



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

IV. Các chương trình hệ thống

Người dùng chủ yếu làm việc thông qua các system program (không làm việc “trực tiếp” với các system call).

- Quản lý hệ thống file: create, delete, rename, list
- Thông tin trạng thái: date, time, dung lượng bộ nhớ trống
- Soạn thảo file: file editor
- Hỗ trợ ngôn ngữ lập trình: compiler, assembler, interpreter
- Nạp, thực thi, giúp tìm lỗi chương trình: loader, debugger
- Giao tiếp: email, talk, web browser, ...



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

- I. Các thành phần của hệ điều hành
- II. Các dịch vụ hệ điều hành cung cấp
- III. Lời gọi hệ thống
- IV. Các chương trình hệ thống
- V. Cấu trúc hệ thống



Sharing is learning

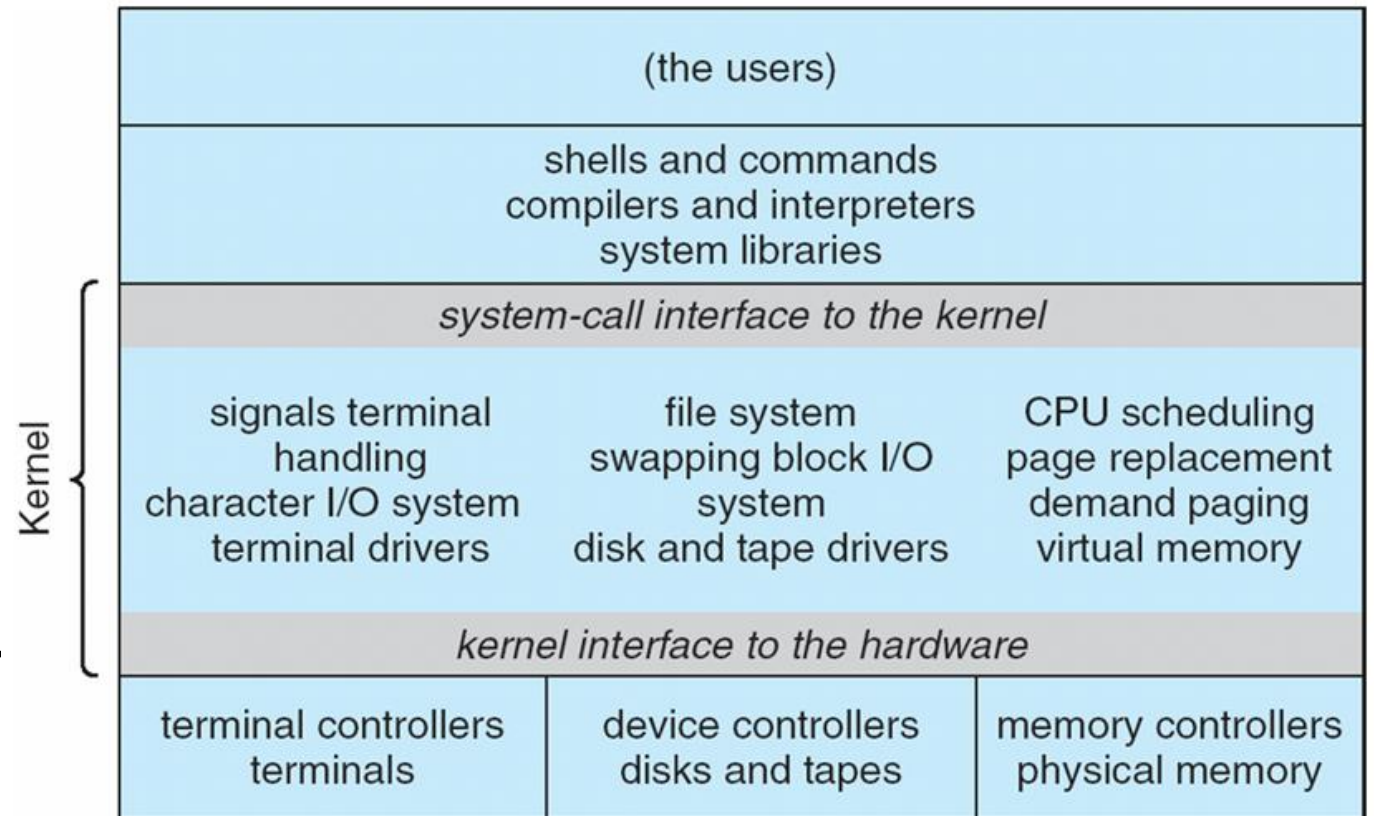
CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

V. Cấu trúc hệ thống

1. Cấu trúc Monolithic – Original Linux

- Cấu trúc đơn giản.
- Gồm 2 phần:
 - + Nhân (Kernel)
 - + System Program
- Được thiết kế theo dạng module.

Ví dụ: Hệ điều hành MS-DOS



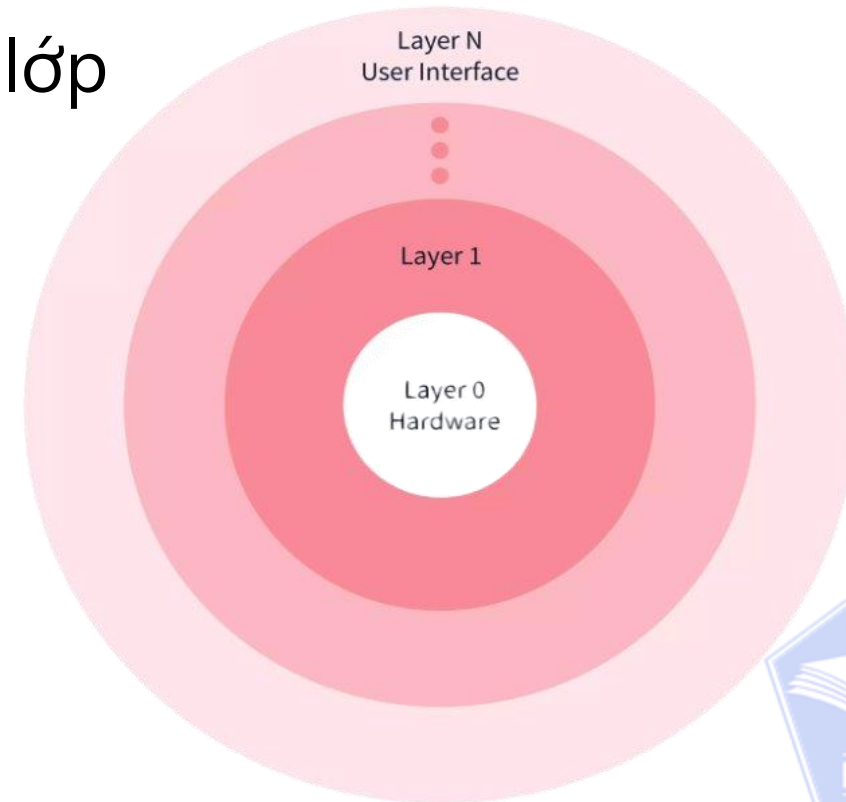
CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

V. Cấu trúc hệ thống

2. Cấu trúc Layered Approach

- Hệ điều hành được chia thành nhiều lớp
- Lớp dưới cùng: hardware
- Lớp trên cùng là giao tiếp với user
- Lớp trên chỉ phụ thuộc lớp dưới
- Một lớp chỉ có thể gọi các hàm của lớp dưới.

Ví dụ: Hệ điều hành THE



Sharing is learning

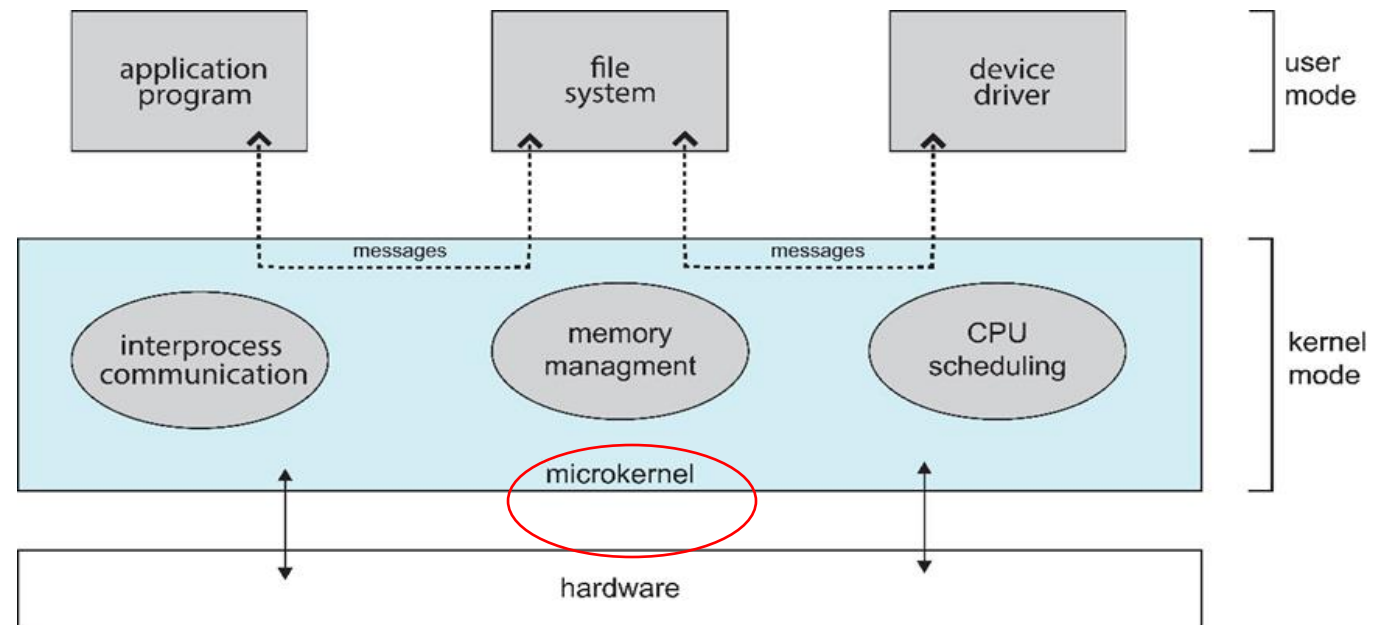
CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

V. Cấu trúc hệ thống

3. Cấu trúc Microkernel

- Chuyển một số chức năng của OS từ kernel space sang user space.
- Thu gọn kernel => microkernel.
- Giao tiếp giữa các user module qua cơ chế truyền thông điệp.

Ví dụ: CMU MACH (1980)



CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

V. Cấu trúc hệ thống

4. Cấu trúc Modules :

Nhiều HĐH hiện đại triển khai loadable kernel modules (LKMs).

- Sử dụng cách tiếp cận ***hướng đối tượng***
- Mỗi core thành phần là tách biệt nhau
- Trao đổi thông qua các interfaces
- Mỗi module như là một phần của nhân

**** Cấu trúc Modules giống với cấu trúc Layer nhưng phức tạp hơn.***



CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

V. Cấu trúc hệ thống

5. Cấu trúc Hybrid System

Hầu hết các hệ điều hành hiện đại không theo một cấu trúc thuần túy nào mà lai giữa các cấu trúc với nhau.

- Nhân Linux và Solaris theo cấu trúc kết hợp không gian địa chỉ kernel, cấu trúc monolithic và modules.
- Nhân Windows hầu như theo cấu trúc monolithic, cộng với cấu trúc microkernels cho các hệ thống cá nhân khác nhau.



CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

Trắc nghiệm khách quan

1. Hệ điều hành nào sau đây theo cấu trúc vi nhân?

- A. Linux B. THE C. QNX D. MS-DOS

=> **Đáp án: C. QNX**

2. Lời gọi hệ thống dùng để giao tiếp giữa hệ điều hành với cái gì?

- A. Người dùng B. Tiến trình
C. Phần cứng máy tính D. Chương trình

=> **Đáp án: B. Tiến trình**



Sharing is learning

CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

Trắc nghiệm khách quan

3. Một trong những nhiệm vụ của bộ nhớ chính là gì?

- A. Quản lý bộ lưu trữ thứ cấp
- B. Quyết định sẽ nạp chương trình nào khi có vùng nhớ trống.
- C. Cung cấp giao diện chung đến các trình điều khiển thiết bị.
- D. Phân định được sự truy cập tài nguyên hợp pháp và bất hợp pháp.

=> Đáp án: B. Quyết định sẽ nạp chương trình nào khi có vùng nhớ trống.



CHƯƠNG 2. CẤU TRÚC HỆ ĐIỀU HÀNH

Trắc nghiệm khách quan

4. Cho biết tên gọi của cấu trúc hệ điều hành mà các modules chức năng của nó được tách ra ngoài? Kernel chỉ có 2 chức năng chính: quản lý bộ nhớ và liên lạc giữa các tiến trình.

- A. Monolithic
- B. Layered
- C. Microkernel
- D. Hybrid

=> Đáp án: C. Microkernel



CHƯƠNG 3. QUẢN LÝ TIẾN TRÌNH

- I. Các khái niệm cơ bản.**
- II. Các trạng thái của tiến trình.**
- III. Process Control Block.**
- IV. Định thời tiến trình.**
- V. Các tác vụ đối với tiến trình.**
- VI. Giao tiếp liên tiến trình.**
- VII. Tiểu trình.**



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

I. Các khái niệm cơ bản.

- Tiến trình là một chương trình đang được hệ điều hành thực thi.
- Chương trình là thực thể bị động lưu trên đĩa. Tiến trình là thực thể chủ động.
- Chương trình trở thành tiến trình khi một tập tin thực thi được nạp vào bộ nhớ.



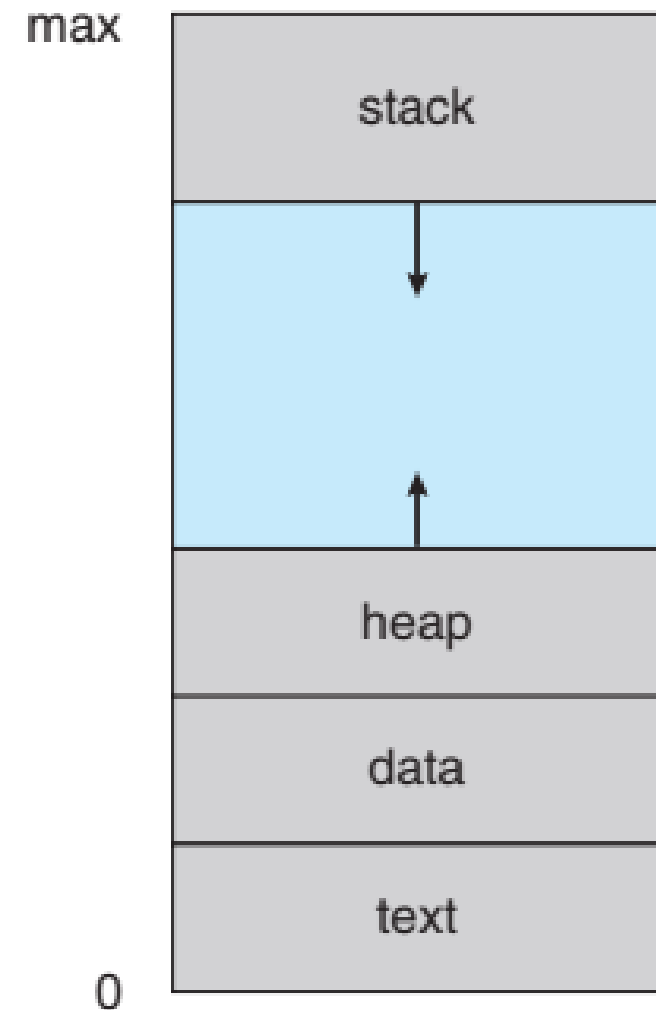
Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

I. Các khái niệm cơ bản.

Một tiến trình bao gồm:

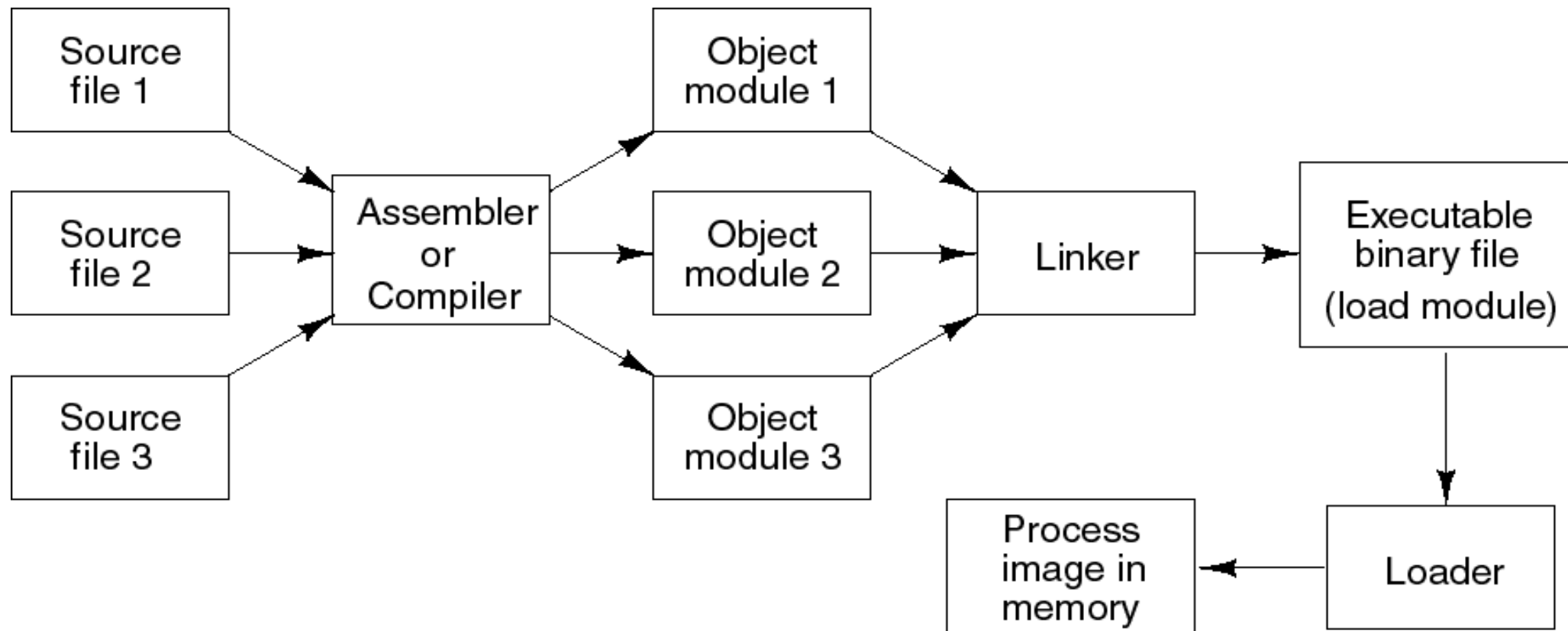
- Text section (program code)
- Data section (chứa biến toàn cục)
- Program counter, processor registers
- Heap section (chứa bộ nhớ cấp phát động)
- Stack section (chứa dữ liệu tạm thời)
 - Function parameters
 - Return address
 - Local variables



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

I. Các khái niệm cơ bản.

**Các bước nạp chương trình vào bộ nhớ:*



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

I. Các khái niệm cơ bản.

**Các bước khởi tạo tiến trình:*

- Cấp phát một định danh duy nhất cho tiến trình.
- Cấp phát không gian nhớ để nạp tiến trình.
- Khởi tạo khối dữ liệu Process Control Block (PCB) cho tiến trình.
- Thiết lập các mối liên hệ cần thiết (ví dụ: sắp PCB vào hàng đợi định thời, ...).



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.
- 2. Các trạng thái của tiến trình.**
3. Process Control Block.
4. Định thời tiến trình.
5. Các tác vụ đối với tiến trình.
6. Giao tiếp liên tiến trình.
7. Tiểu trình.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

II. Các trạng thái của tiến trình.

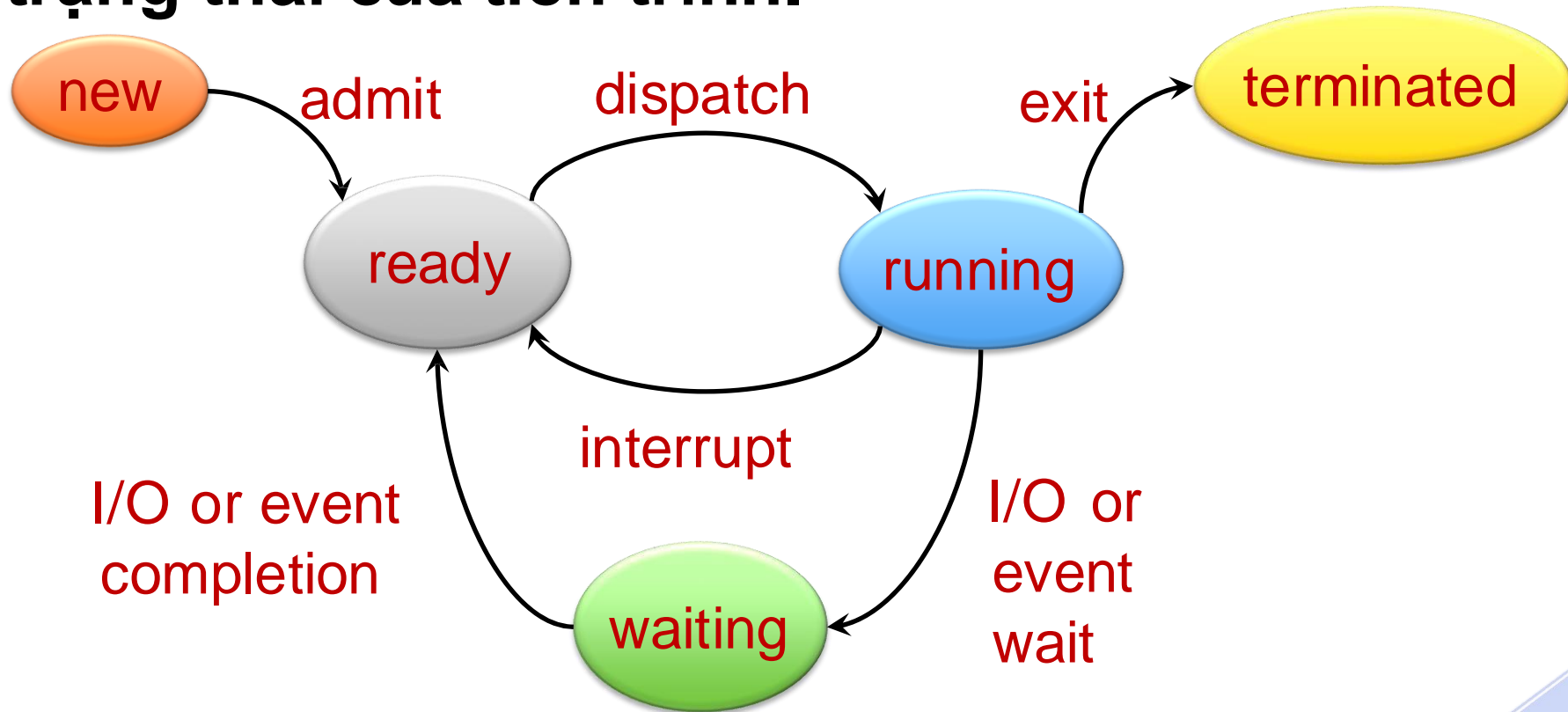
- **new**: tiến trình vừa được tạo
- **ready**: tiến trình đã có đủ tài nguyên, chỉ còn cần CPU
- **running**: các lệnh của tiến trình đang được thực thi
- **waiting** (hay **blocked**): tiến trình **đợi I/O hoàn tất**, hoặc đợi tín hiệu
- **terminated**: tiến trình đã kết thúc



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

II. Các trạng thái của tiến trình.



Chuyển đổi giữa các trạng thái của tiến trình



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

II. Các trạng thái của tiến trình.

Ví dụ:

Chuỗi trạng thái của tiến trình:

new – ready – running – waiting – ready –
running – waiting – ready – running –
waiting – ready – running – waiting –
ready – running – terminated

```
int main ()
{
    int i = 2;
    while (i < 5)
    {
        i++;
        if (i % 2 == 0)
        {
            printf ("Hello");
            printf ("Hi");
        }
        else
        {
            printf ("Bye");
        }
    }
    exit (0);
}
```



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.
2. Các trạng thái của tiến trình.
- 3. Process Control Block.**
4. Định thời tiến trình.
5. Các tác vụ đối với tiến trình.
6. Giao tiếp liên tiến trình.
7. Tiểu trình.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

III. Process Control Block

Mỗi tiến trình trong hệ thống đều được cấp phát một **Process Control Block** (PCB)

- PCB là một trong các cấu trúc dữ liệu quan trọng nhất của hệ điều hành

PCB gồm:

- Trạng thái tiến trình: new, ready, running,...
- Bộ đếm chương trình
- Các thanh ghi
- Thông tin lập thời biểu CPU: độ ưu tiên, ...
- Thông tin quản lý bộ nhớ
- Thông tin lượng CPU, thời gian sử dụng
- Thông tin trạng thái I/O

PCB

pointer	process state
process number	
program counter	
registers	
memory limits	
list of open files	
⋮	

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.
2. Các trạng thái của tiến trình.
3. Process Control Block.
- 4. Định thời tiến trình.**
5. Các tác vụ đối với tiến trình.
6. Giao tiếp liên tiến trình.
7. Tiểu trình.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

IV. Định thời tiến trình

a) Yêu cầu đối với hệ điều hành về quản lý tiến trình

- Hỗ trợ sự thực thi luân phiên giữa nhiều tiến trình
 - ❖ Hiệu suất sử dụng CPU: cực đại
 - ❖ Thời gian đáp ứng: cực tiểu
- Phân phối tài nguyên hệ thống hợp lý
- Tránh deadlock, trì hoãn vô hạn định
- Cung cấp cơ chế giao tiếp và đồng bộ hoạt động các tiến trình
- Cung cấp cơ chế hỗ trợ user tạo/kết thúc tiến trình



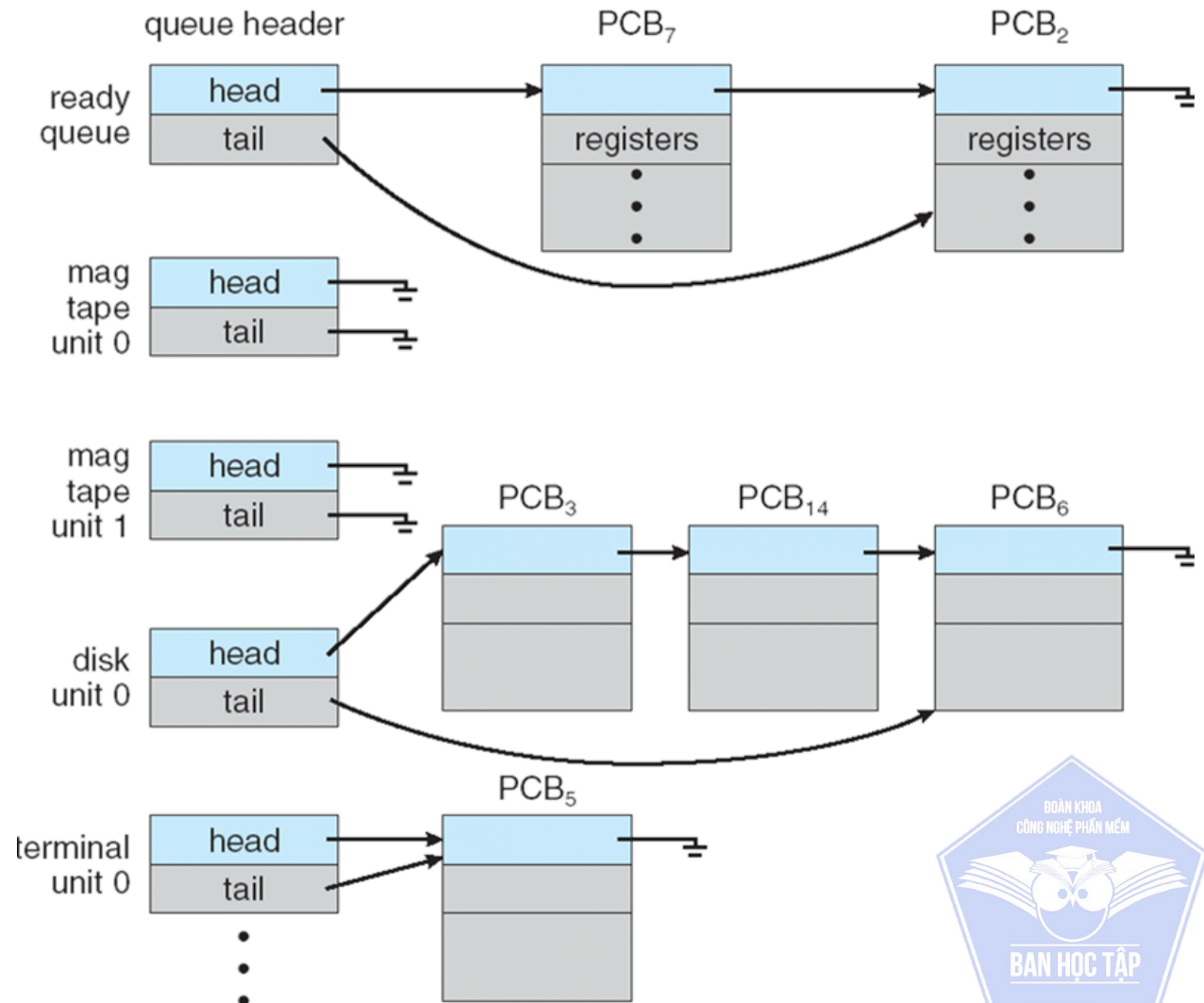
Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

IV. Định thời tiến trình

b) Các hàng đợi định thời

- Hàng đợi công việc - Job queue
- Hàng đợi sẵn sàng - Ready queue
- Hàng đợi thiết bị - Device queues
- ...



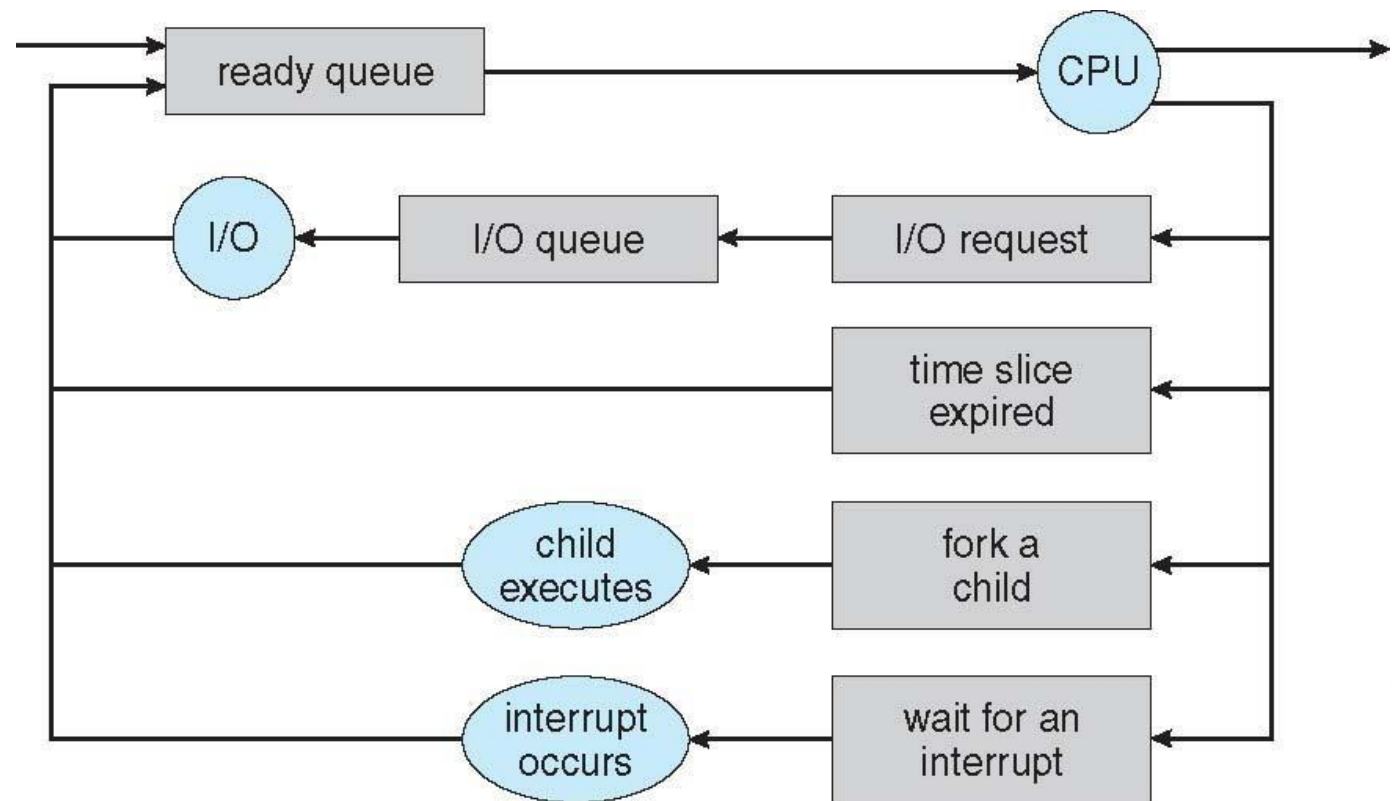
Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

IV. Định thời tiến trình

b) Các hàng đợi định thời

**Lưu đồ hàng đợi của định thời tiến trình*



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

IV. Định thời tiến trình

c) Các loại bộ định thời

Phân loại bộ định thời

- **Bộ định thời công việc** (Job scheduler) hay **bộ định thời dài** (long-term scheduler)
- **Bộ định thời CPU** hay **bộ định thời ngắn**

Phân loại tiến trình

- Các tiến trình có thể mô tả như:
 - **tiến trình hướng I/O**
 - **tiến trình hướng CPU**
- Thời gian thực hiện khác nhau
→ kết hợp hài hòa giữa chúng



Sharing is learning

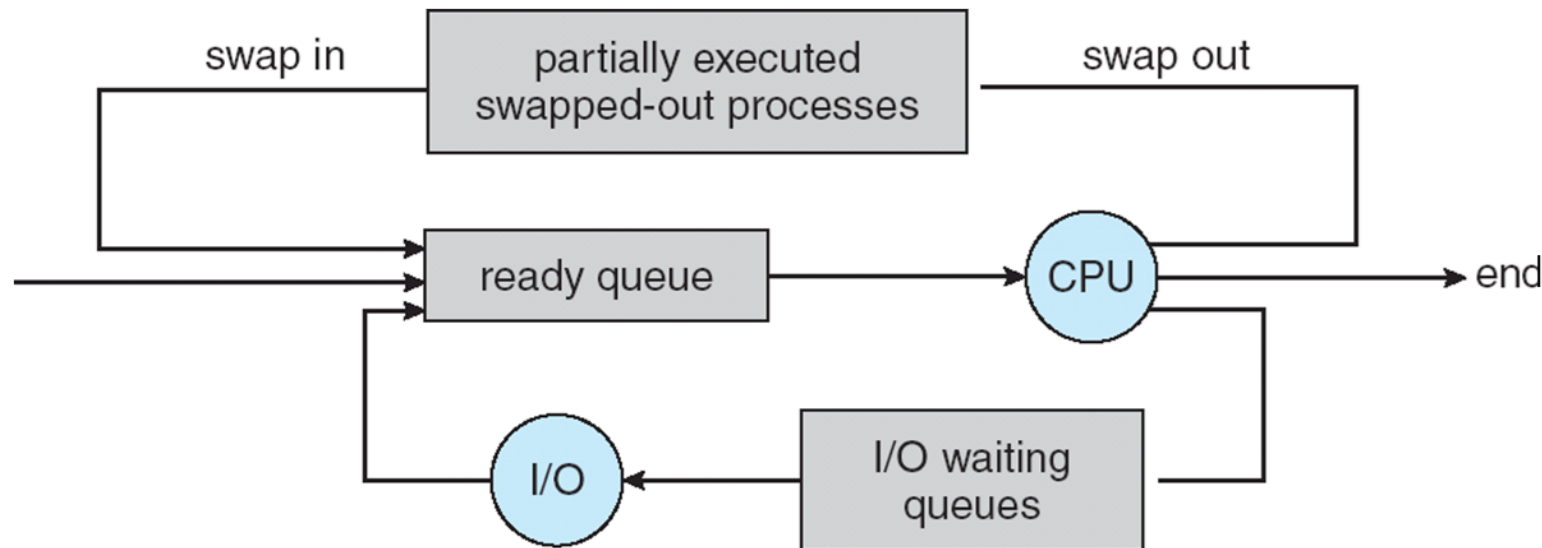
CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

IV. Định thời tiến trình

c) Các loại bộ định thời

Đôi khi hệ điều hành (như time-sharing system) có thêm **medium-term scheduling** để điều chỉnh mức độ đa chương của hệ thống.

- Chuyển tiến trình từ bộ nhớ sang đĩa (swap out)
- Chuyển tiến trình từ đĩa vào bộ nhớ (swap in)



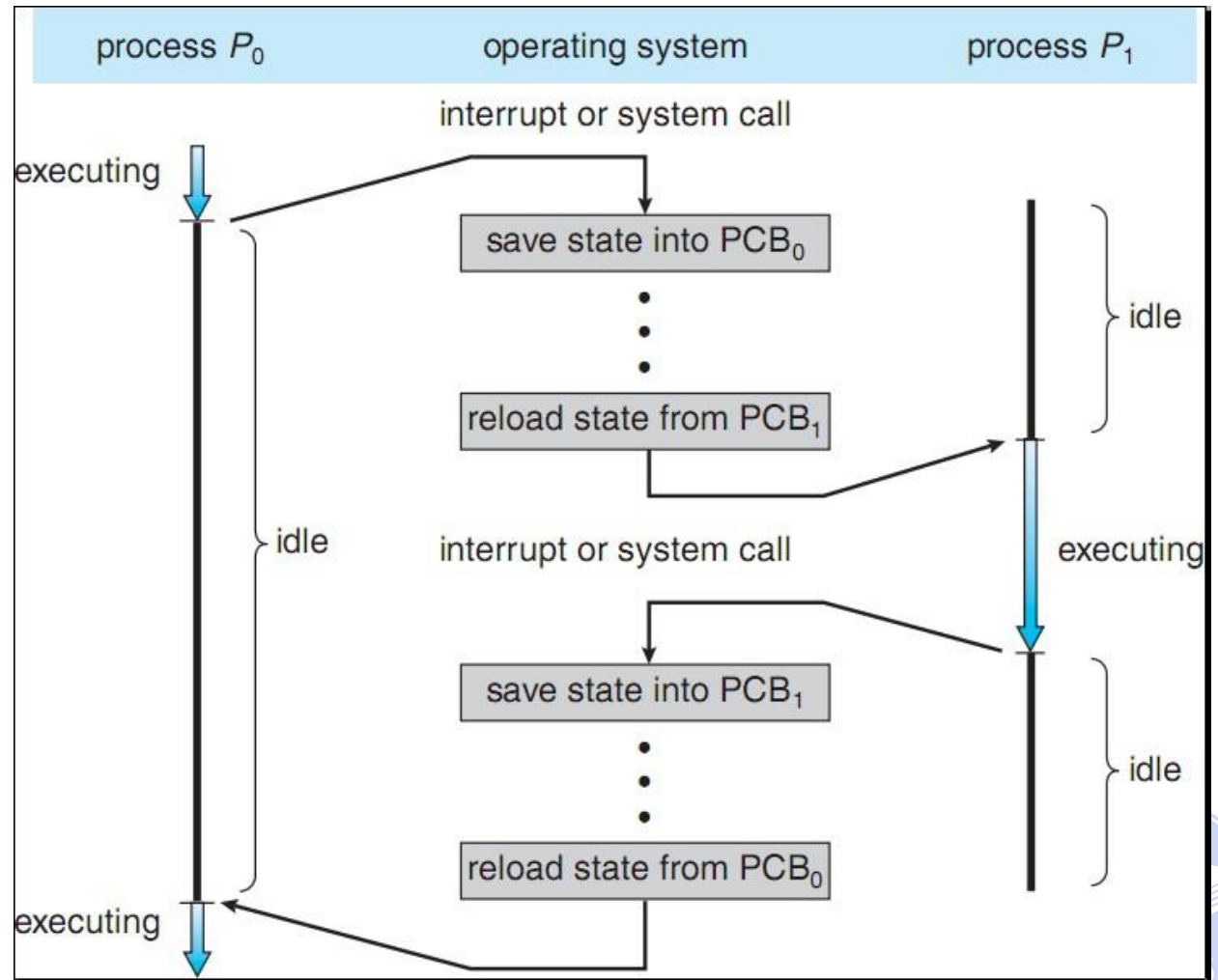
Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

IV. Định thời tiến trình

d) Chuyển ngữ cảnh

Quá trình CPU chuyển từ tiến trình này đến tiến trình khác



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.
2. Các trạng thái của tiến trình.
3. Process Control Block.
4. Định thời tiến trình.
- 5. Các tác vụ đối với tiến trình.**
6. Giao tiếp liên tiến trình.
7. Tiểu trình.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình

Có 2 tác vụ chính:

- Tạo tiến trình mới
- Kết thúc tiến trình



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình.

- Tạo tiến trình mới:

Tiến trình con nhận tài nguyên từ HĐH hoặc từ tiến trình cha

❖ Chia sẻ tài nguyên của tiến trình cha

- Tiến trình cha và con chia sẻ mọi tài nguyên
- Tiến trình con chia sẻ một phần tài nguyên của cha

❖ Trình thực thi

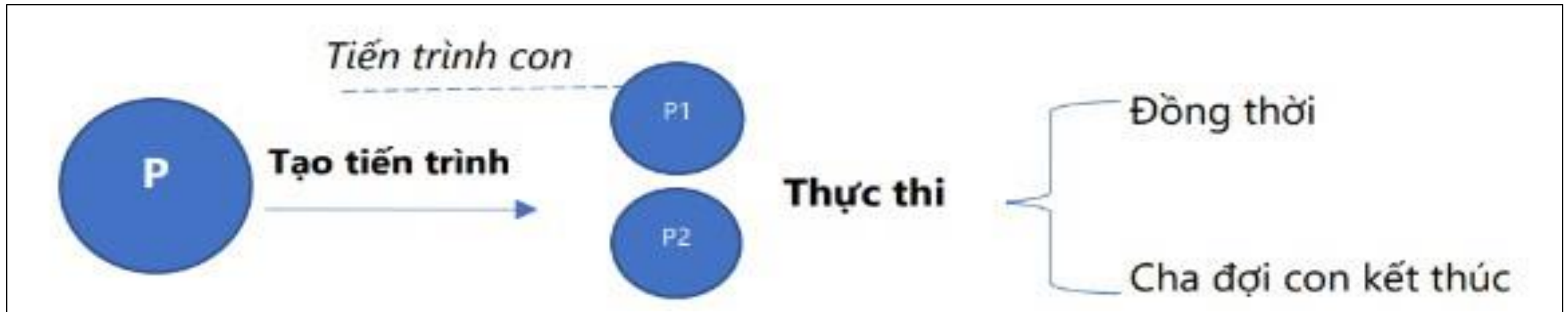
- Tiến trình cha và con thực thi đồng thời (concurrently)
- Tiến trình cha đợi đến khi các tiến trình con kết thúc



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình.

- Tạo tiến trình mới:



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình

- Kết thúc tiến trình:

❖ Tiến trình tự kết thúc.

Tiến trình kết thúc khi thực thi lệnh cuối và gọi system routine exit.

❖ **Tiến trình kết thúc do tiến trình khác** (có đủ quyền, vd: tiến trình cha của nó).

Gọi system routine abort với tham số là pid(process identifier) của tiến trình cần được kết thúc.

⇒ Hệ điều hành thu hồi tất cả các tài nguyên của tiến trình kết thúc (vùng nhớ, I/O buffer,..).



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình

- Cộng tác giữa các tiến trình:

Trong tiến trình thực thi, các tiến trình có thể cộng tác để hoàn thành công việc, nhằm:

- Chia sẻ dữ liệu
- Tăng tốc tính toán
- Thực hiện một công việc chung

⇒ Sự cộng tác yêu cầu hệ điều hành hỗ trợ cơ chế giao tiếp và cơ chế đồng bộ hoạt động của các tiến trình.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình

Hàm fork()

- Fork() tạo ra tiến trình mới bằng cách nhân bản (duplicate) tiến trình gọi hàm này
- Tiến trình ban đầu gọi là tiến trình cha (parent process)
- Tiến trình được nhân bản ra được gọi là tiến trình con (child process), là một bản sao giống với tiến trình cha tạo ra nó (kể cả trạng thái thực thi)

Giá trị
trả về

- PID của tiến trình con nếu tạo được tiến trình con
- -1 nếu tạo tiến trình con bị lỗi
- 0 cho tiến trình con



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình

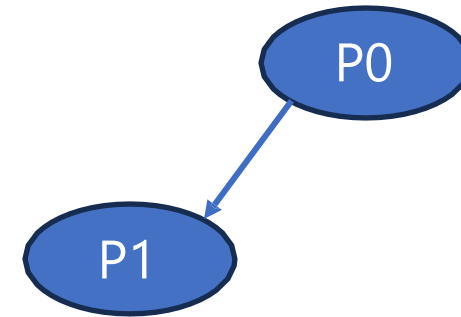
Ví dụ:

```
#include <stdio.h>
int main() {
    fork();
    printf("BHT CNPM");
}
```

⇒ Cho biết output?

Hướng dẫn làm

Cây tiến trình:



OUTPUT: BHT CNPMBHT CNPM



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình

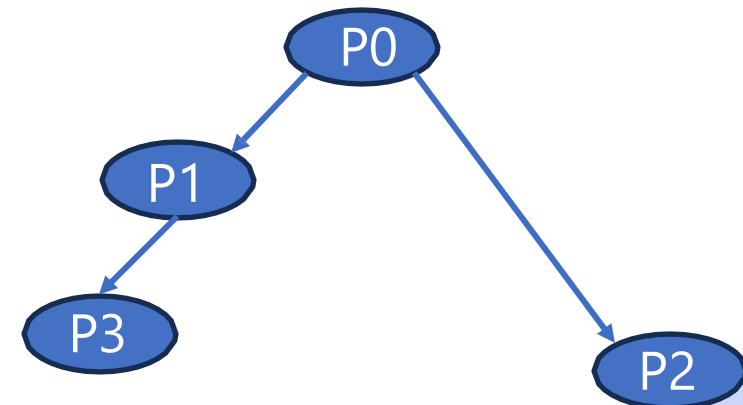
Ví dụ:

```
#include <stdio.h>
int main() {
    fork();
    fork();
    printf("BHT CNPM");
}
```

⇒ Cho biết output?

Hướng dẫn làm

Cây tiến trình:



OUTPUT: BHT CNPMBHT
CNPM BHT CNPMBHT
CNPM



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

V. Các tác vụ đối với tiến trình

Ví dụ:

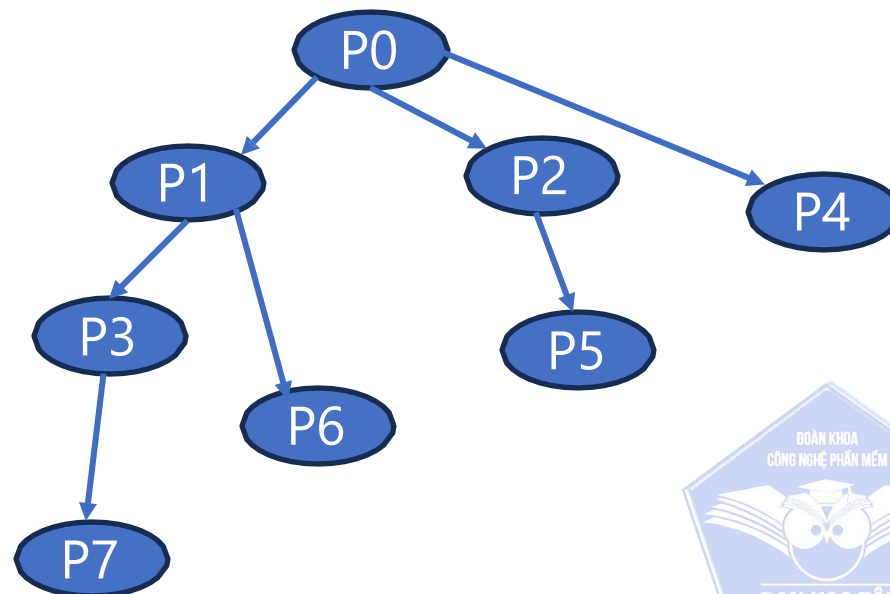
```
#include <stdio.h>
int main() {
    printf("Hello, Alo")
    fork();
    fork();
    printf("Hi");
    fork();
    printf("Alo");
}
```

⇒ Cho biết có bao nhiêu chữ Hello, Hi, Alo?

- A. 1,4,8 B. 1,4,6
C. 1,2,9 **D. 1,4,9**

Hướng dẫn làm

Cây tiến trình:



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.
2. Các trạng thái của tiến trình.
3. Process Control Block.
4. Định thời tiến trình.
5. Các tác vụ đối với tiến trình.
- 6. Giao tiếp liên tiến trình.**
7. Tiểu trình.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VI. Giao tiếp liên tiến trình.

- IPC – Inter Process Communication là cơ chế cung cấp bởi hệ điều hành nhằm giúp các tiến trình:
 - Giao tiếp với nhau.
 - Đồng bộ hoạt động.
- Hai mô hình IPC:
 - Shared memory
 - Message passing

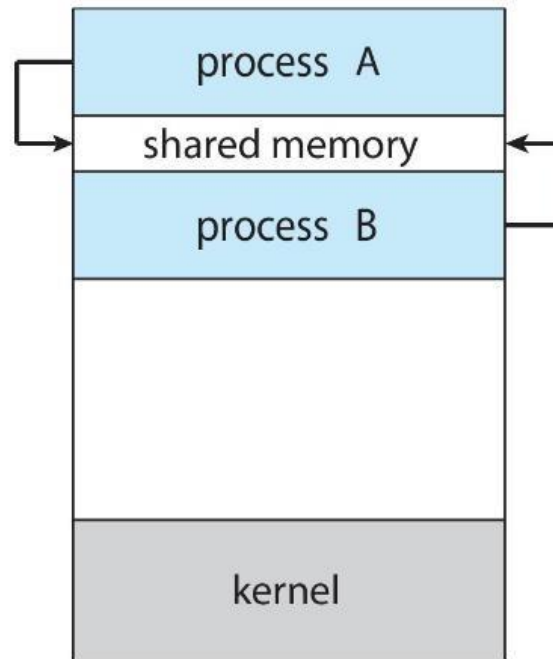


Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

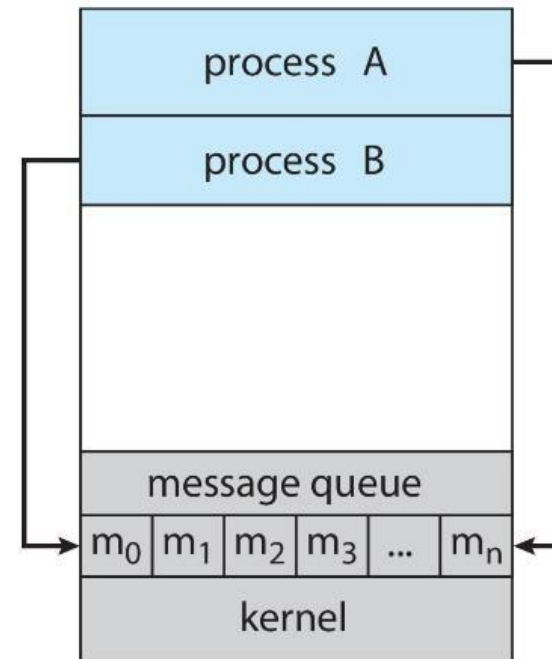
VI. Giao tiếp liên tiến trình.

(a) Shared memory.



(a)

(b) Message passing.



(b)



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VI. Giao tiếp liên tiến trình.

Share Memory

- Một vùng nhớ dùng chung (được chia sẻ chung) giữa các tiến trình cần giao tiếp với nhau.
- Quá trình giao tiếp được thực hiện dưới sự điều khiển của các tiến trình, không phải của hệ điều hành.
- Cần có cơ chế đồng bộ hoạt động của các tiến trình khi chúng cùng truy xuất bộ nhớ dùng chung.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VI. Giao tiếp liên tiến trình.

Message Passing

- Đặt tên (Naming)

➤ Giao tiếp trực tiếp

- send(P, msg): gửi thông điệp đến tiến trình P
- receive(Q, msg): nhận thông điệp đến từ tiến trình Q

➤ Giao tiếp gián tiếp: thông qua mailbox hay port

- send(A, msg): gửi thông điệp đến mailbox A
- receive(Q, msg): nhận thông điệp từ mailbox B

- **Đồng bộ hóa (Synchronization):** blocking send, non-blocking send, blocking receive, nonblocking receive.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.
2. Các trạng thái của tiến trình.
3. Process Control Block.
4. Định thời tiến trình.
5. Các tác vụ đối với tiến trình.
6. Giao tiếp liên tiến trình.
- 7. Tiểu trình.**

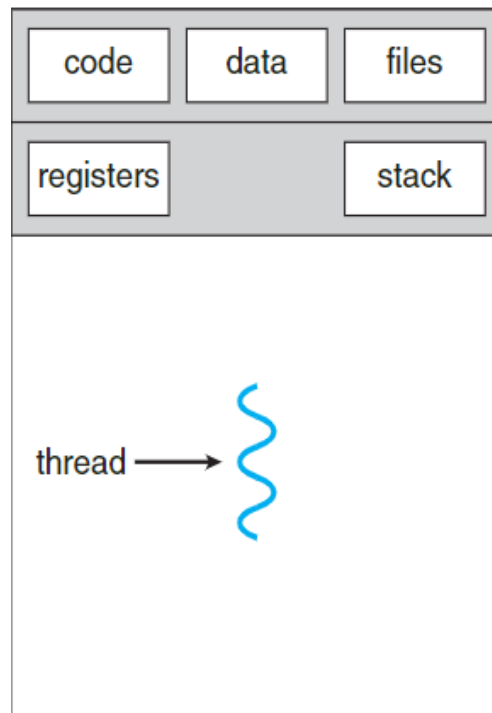


Sharing is learning

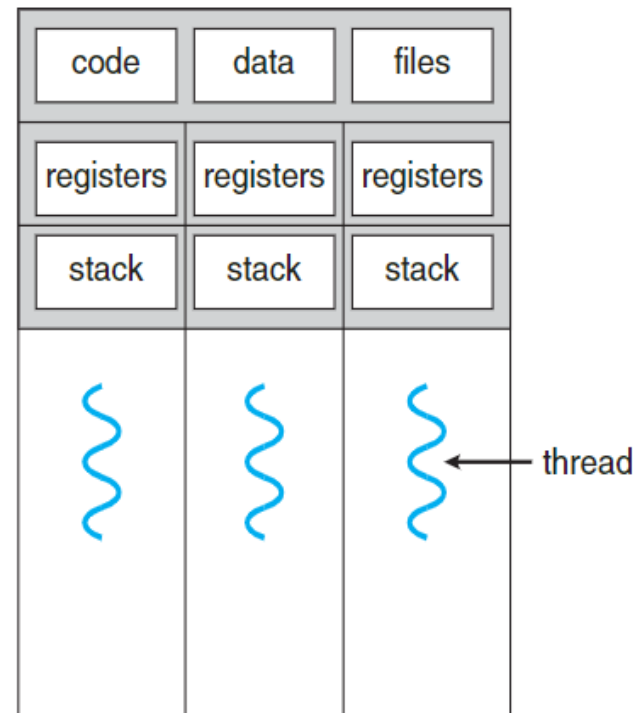
CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VII. Tiểu trình.

Tiểu trình là một đơn vị cơ bản sử dụng CPU gồm: Thread ID, PC, Registers, Stack và chia sẻ chung code, data, resources.



single-threaded process



multithreaded process



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VII. Tiểu trình.

Lợi ích của tiến trình đa luồng:

- Đáp ứng nhanh: cho phép chương trình tiếp tục thực thi khi một bộ phận bị khóa hoặc một hoạt động dài
- Chia sẻ tài nguyên: tiết kiệm không gian nhớ
- Kinh tế: tạo và chuyển ngữ cảnh nhanh hơn tiến trình
- Trong multiprocessor: có thể thực hiện song song.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VII. Tiểu trình.

Các mô hình đa tiểu trình:

- Nhiều – Một (Many-to-One)
- Một – Một (One-to-One)
- Nhiều – Nhiều (Many-to-Many)



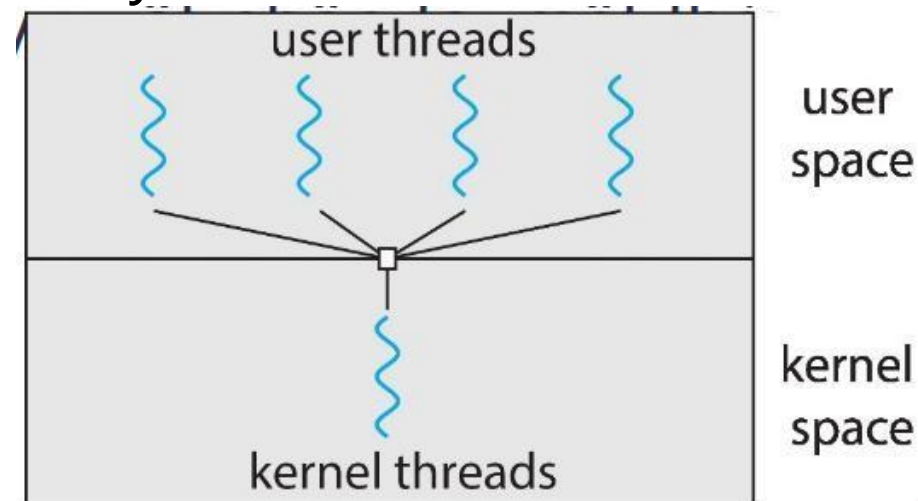
Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VII. Tiểu trình.

Nhiều – Một (Many-to-One)

- Nhiều tiểu trình người dùng được ánh xạ đến một tiểu trình hạt nhân.
- Một tiểu trình bị block sẽ dẫn đến tất cả tiểu trình bị block.
- Các tiểu trình không thể chạy song song trên các hệ thống đa lõi bởi vì chỉ có một tiểu trình có thể truy xuất nhân tại một thời điểm.
- Rất ít hệ thống sử dụng mô hình này.



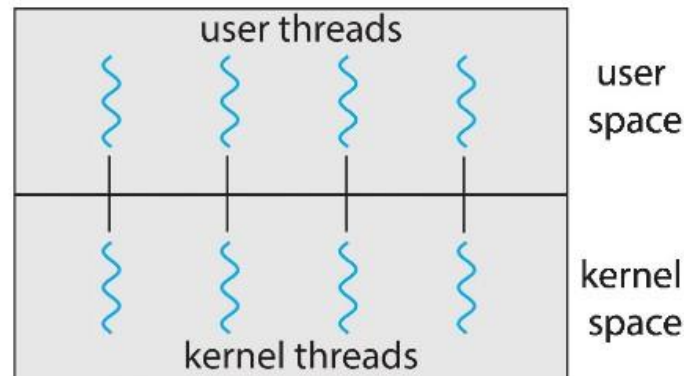
Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VII. Tiểu trình.

Một – Một (One-to-One): Mỗi tiểu trình người dùng ứng với một tiểu trình hạt nhân.

- Tạo một tiểu trình người dùng cũng đồng thời tạo một tiểu trình hạt nhân.
- Tính đồng thời (concurrency) tốt hơn mô hình nhiều – một vì các tiểu trình khác vẫn hoạt động bình thường khi một tiểu trình bị block.
- **Nhược điểm:** Số lượng tiểu trình của mỗi tiến trình có thể bị hạn chế.
- Nhiều hệ điều hành sử dụng: Windows, Linux.



Sharing is learning

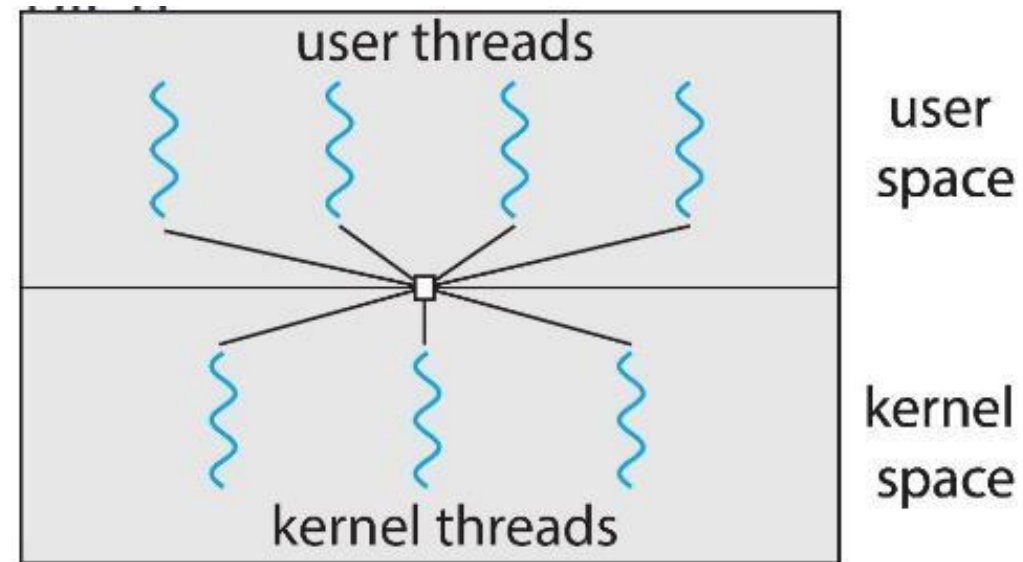
CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

VII. Tiểu trình.

Nhiều – Nhiều (Many-to-Many)

Các tiểu trình người dùng được ánh xạ với nhiều tiểu trình hạt nhân.

- Cho phép hệ điều hành tạo đủ số lượng tiểu trình hạt nhân \Rightarrow Giải quyết được hạn chế của 2 mô hình trên.
- Khó cài đặt nên ít phổ biến.



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

**Trắc nghiệm khách quan*

Câu 1: Tiến trình đang ở trạng thái running không thể chuyển sang trạng thái nào dưới đây?

- ☒ A. New
- ☐ B. Ready
- ☐ C. Waiting
- ☐ D. Terminated



Sharing is learning

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

**Trắc nghiệm khách quan*

Câu 2: Chọn phát biểu **SAI** trong các phát biểu sau:

- A. Cây tiến trình là một cách thể hiện quan hệ giữa tiến trình cha và tiến trình con
- ☒ B. Không gian địa chỉ của tiến trình con luôn được nhân bản từ tiến trình cha
- C. Tiến trình con có thể chia sẻ một phần hoặc toàn bộ tài nguyên của tiến trình cha
- D. Tiến trình cha có thể kết thúc tiến trình con



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

I. Các khái niệm cơ bản về các bộ định thời.

II. Các loại bộ định thời.

III. Các tiêu chuẩn định thời.

IV. Các giải thuật định thời.



CHƯƠNG 4. ĐỊNH THỜI CPU

I. Các khái niệm cơ bản về các bộ định thời.

II. Các loại bộ định thời.

III. Các tiêu chuẩn định thời.

IV. Các giải thuật định thời.



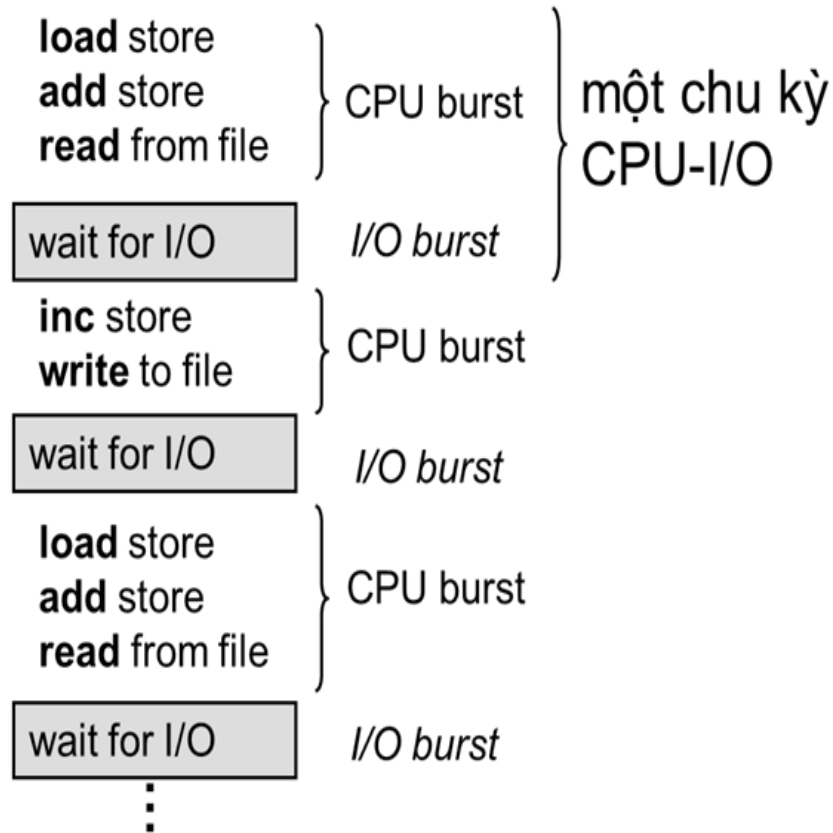
Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

I. Các khái niệm cơ bản về các bộ định thời

- **Định thời** là chiến lược lựa chọn tiến trình phù hợp để được thực thi sao cho đạt được hiệu quả cao nhất.

- **Service time** là thời gian một tiến trình cần CPU trong một chu kỳ CPU - I/O (hay còn gọi là **burst time**).



Process	Arrival Time	Service Time
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2



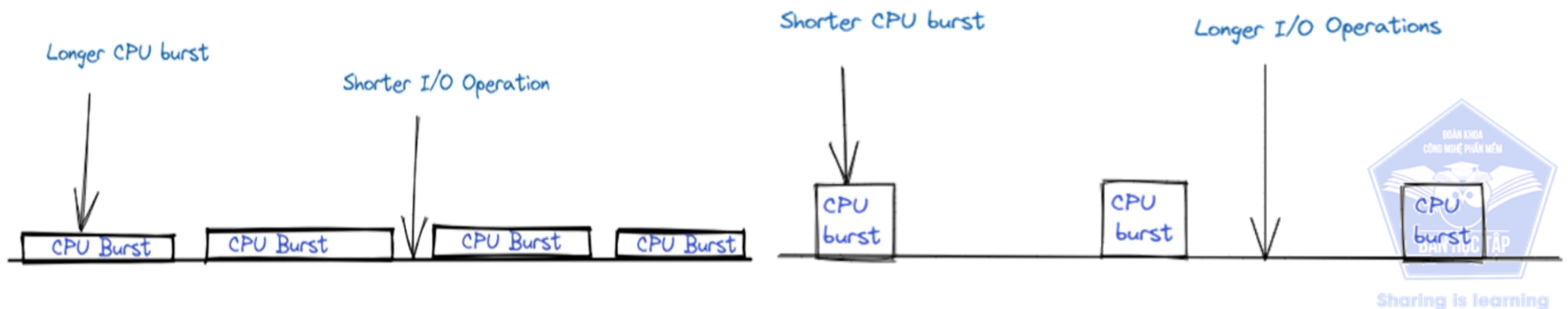
Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

I. Các khái niệm cơ bản về các bộ định thời

- Tiến trình có service time lớn được gọi là các **tiến trình hướng CPU (CPU-bound process)**.

Tiến trình hướng I/O yêu cầu thời gian thực thi trên ngoại vi nhiều hơn, thời gian hoàn thành chương trình phụ thuộc chu kỳ đợi cho các thao tác nhập/xuất.



CHƯƠNG 4. ĐỊNH THỜI CPU

I. Các khái niệm cơ bản về các bộ định thời.

II. Các loại bộ định thời.

III. Các tiêu chuẩn định thời.

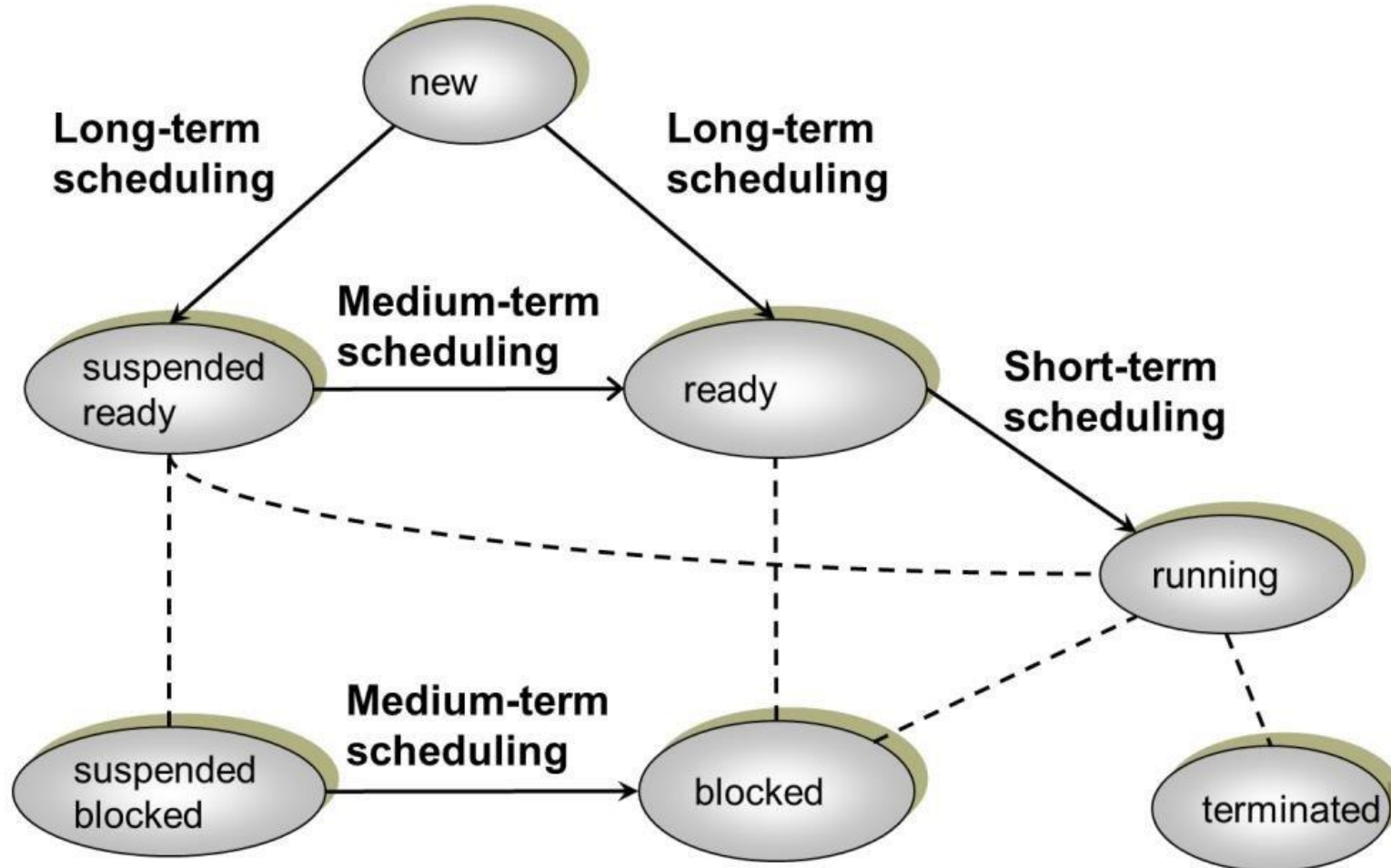
IV. Các giải thuật định thời.



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

II. Các loại bộ định thời



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

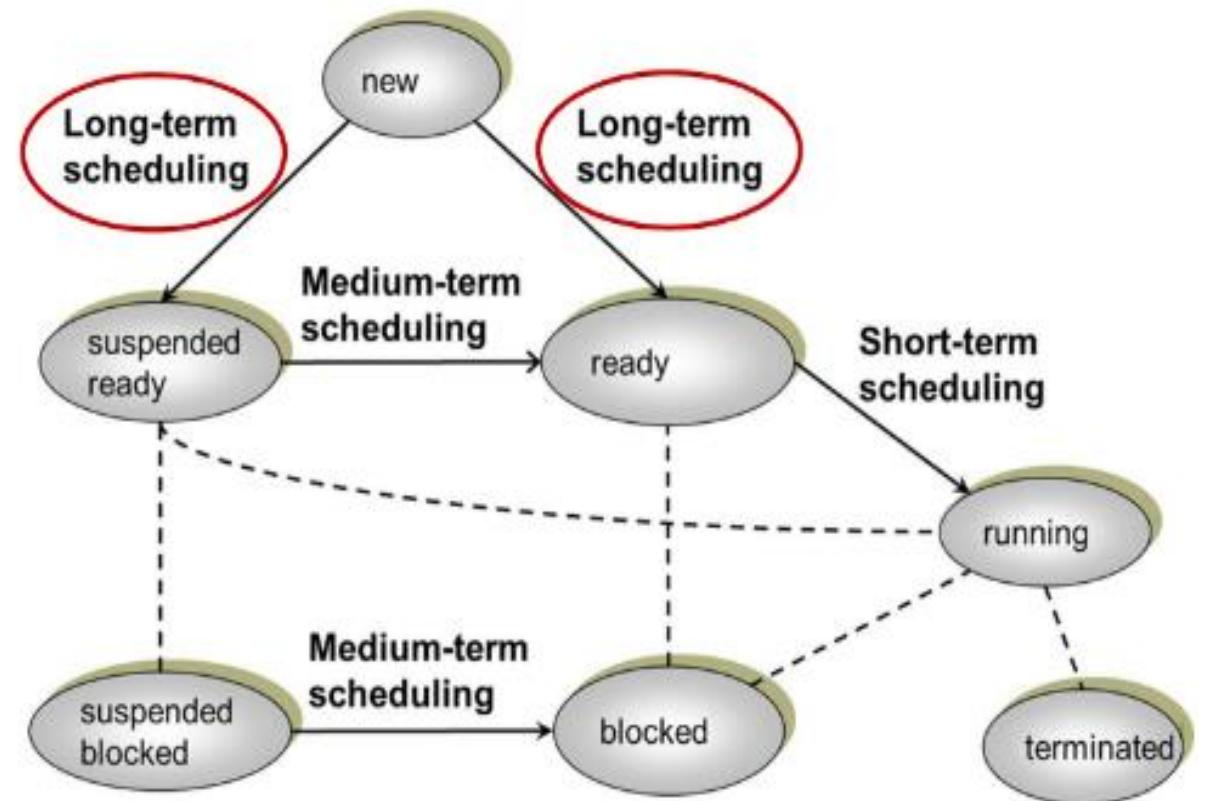
II. Các loại bộ định thời

1. Bộ định thời dài

- Xác định chương trình nào được chấp nhận nạp vào hệ thống để thực thi

= > điều khiển mức độ đa chương của hệ thống

- Thường cố gắng duy trì xen lẫn giữa tiến trình hưởng CPU và tiến trình hưởng I/O.

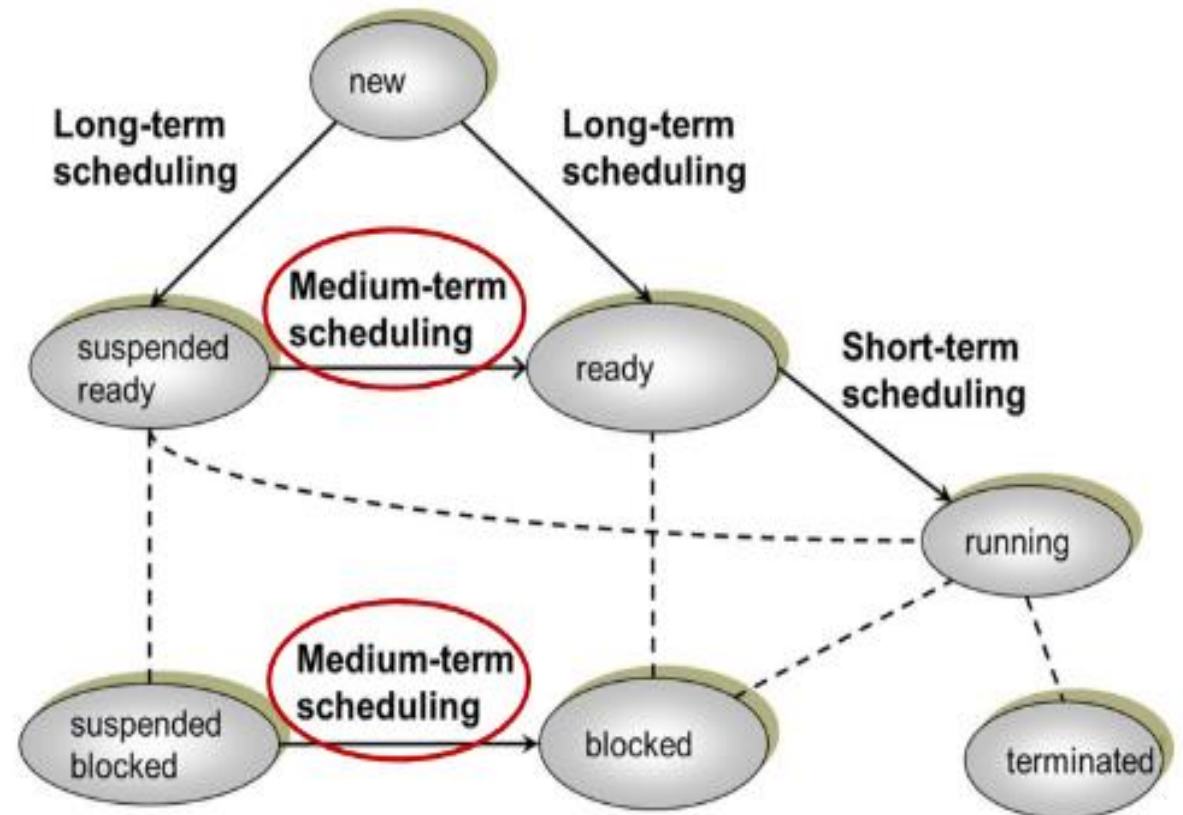


CHƯƠNG 4. ĐỊNH THỜI CPU

II. Các loại bộ định thời

2. Bộ định thời vừa

- Quyết định tiến trình nào được đưa vào (swap in) và đưa ra khỏi (swap out) bộ nhớ chính trong quá trình thực thi của hệ thống.
- Được thực hiện bởi phần quản lý bộ nhớ

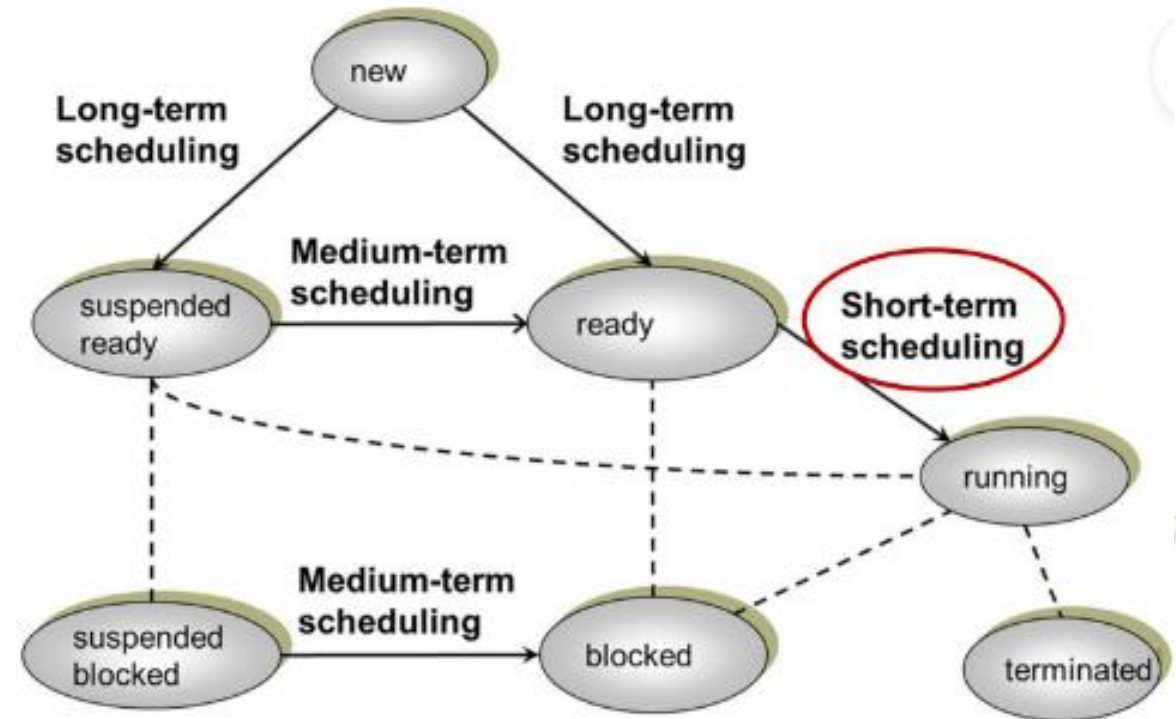


CHƯƠNG 4. ĐỊNH THỜI CPU

II. Các loại bộ định thời

3. Bộ định thời ngắn

- Xác định process nào trong ready queue sẽ được chiếm CPU để thực thi kế tiếp
- Bộ định thời ngắn gọi khi:
 - + Ngắt thời gian
 - + Ngắt ngoại
 - + Lời gọi hệ thống
 - + Tín hiệu đồng bộ hóa



CHƯƠNG 4. ĐỊNH THỜI CPU

I. Các khái niệm cơ bản về các bộ định thời.

II. Các loại bộ định thời.

III. Các tiêu chuẩn định thời.

IV. Các giải thuật định thời.



CHƯƠNG 4. ĐỊNH THỜI CPU

III. Các tiêu chuẩn định thời

1. Hướng người dùng

- Thời gian đáp ứng (Response time): khoảng thời gian từ lúc tiến trình gửi yêu cầu thực thi đến khi yêu cầu được đáp ứng lần đầu tiên (trong các hệ thống time-sharing, interactive system).
- Thời gian hoàn thành (Turnaround time): khoảng thời gian từ lúc một tiến trình được nạp vào hệ thống đến khi tiến trình đó kết thúc.
- Thời gian đợi (Waiting time): tổng thời gian một tiến trình đợi trong ready queue

=> **ĐỀU YÊU CẦU CỰC TIỂU.**

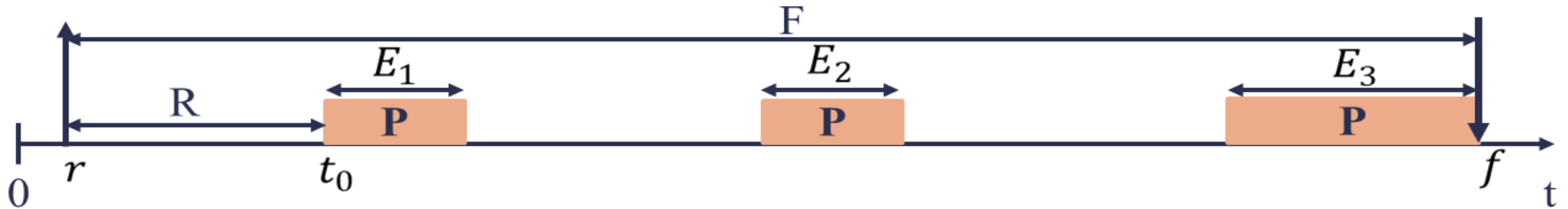


Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

III. Các tiêu chuẩn định thời

1. Hướng người dùng



Giả sử: r là thời điểm xuất hiện của P trong hệ thống (Arrival Time/Release Time), t_0 là thời điểm P được thực thi lần đầu tiên, f là thời điểm tiến trình P đã hoàn thành việc thực thi.

Thì: R (thời gian đáp ứng) $= t_0 - r$, E (thời gian chờ) $= f - (E_1 + E_2 + E_3) - r$, F (thời gian hoàn thành) $= f - r$.



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

III. Các tiêu chuẩn định thời

1. Hướng hệ thống

- **Hiệu năng CPU:** làm cho CPU càng bận càng tốt => CỰC ĐẠI.
- **Tính công bằng:** tất cả các tiến trình được đối xử như nhau.
- **Thông lượng:** số tiến trình hoàn tất công việc trong một đơn vị thời gian => CỰC ĐẠI



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

I. Các khái niệm cơ bản về các bộ định thời.

II. Các loại bộ định thời.

III. Các tiêu chuẩn định thời.

IV. Các giải thuật định thời.



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

Một giải thuật định thời thông thường bao gồm hai yếu tố:

- **Hàm chọn lựa (selection function):** mô tả cách thức (căn cứ) để chọn tiến trình nào trong ready queue được thực thi (Các hàm chọn lựa thường được xây dựng dựa trên độ ưu tiên, yêu cầu về tài nguyên, đặc điểm thực thi của tiến trình,...).
- **Chế độ quyết định (decision mode):** quyết định thời điểm thực hiện hàm chọn lựa để định thời.
 - Không trưng dụng
 - Trưng dụng



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

Chế độ quyết định:

- Không trưng dụng (Non-preemptive): Khi ở trạng thái running, process sẽ thực thi cho đến khi kết thúc hoặc bị blocked do yêu cầu I/O.
- Trưng dụng (Preemptive): Process đang thực thi (running) có thể bị ngắt nửa chừng và chuyển về trạng thái ready.

⇒ Chi phí preemptive cao hơn non-preemptive nhưng đánh đổi lại bằng thời gian đáp ứng tốt hơn vì không có trường hợp một process độc chiếm CPU quá lâu.



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

1. First-Come, First-Served (FCFS)

Hàm lựa chọn:

- Tiến trình nào yêu cầu CPU trước sẽ được cấp phát CPU trước.
- Tiến trình sẽ thực thi đến khi kết thúc hoặc bị blocked do I/O.

Chế độ quyết định: **không trưng dụng (non-preemptive)**.

Hiện thực: sử dụng hàng đợi **FIFO (FIFO queues)**

- Tiến trình mới xuất hiện được thêm vào cuối hàng đợi.
- Tiến trình được lựa chọn để xử lý được lấy từ đầu của hàng đợi.



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

1. First-Come, First-Served (FCFS)

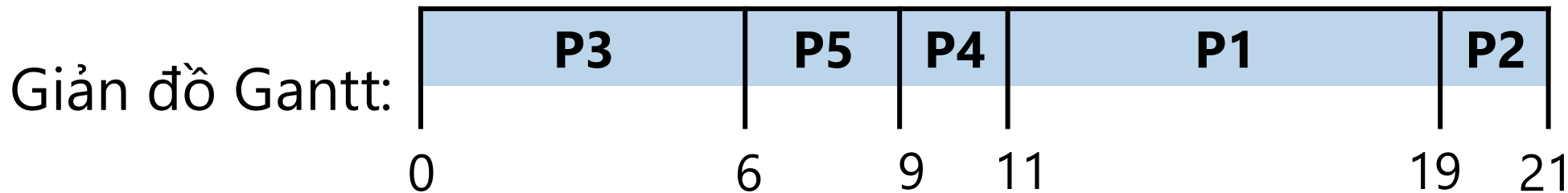
Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

Thời gian đáp ứng = Thời gian chờ
= t khi P được thực thi - Arrival t

Kết quả:

$P1=1, P2=8, P3=0, P4=5, P5=4$

Thời gian đáp ứng/ đợi trung bình:
 $(P1+P2+P3+P4+P5)/5 = 3.6$



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

1. First-Come, First-Served (FCFS)

Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

Thời gian hoàn thành:

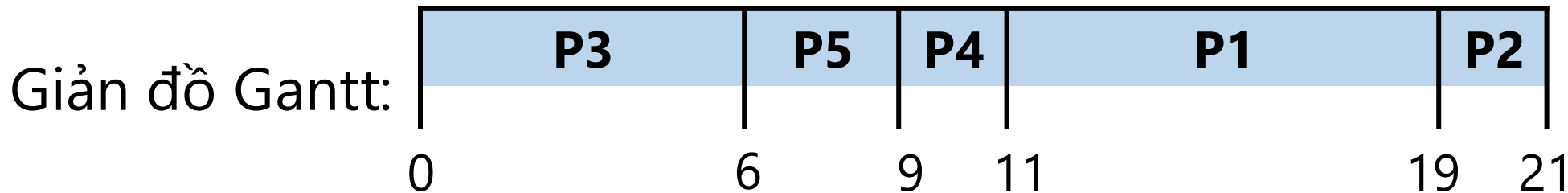
$$P1 = 19 - 10 = 9$$

$$P2 = 21 - 11 = 10$$

$$P3 = 6, P4 = 7, P5 = 7$$

Thời gian hoàn thành trung bình:

$$(P1 + P2 + P3 + P4 + P5) / 5 = 7.8$$



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

2. Shortest-Job-First (SJF)

Hàm lựa chọn:

- Khi CPU trống, HĐH sẽ chọn tiến trình có CPU Burst ngắn nhất để được thực thi tiếp theo.
- Giải thuật này sử dụng chiều dài thời gian thực thi của tiến trình làm căn cứ để chọn lựa.

Chế độ quyết định: **không trưng dụng (non-preemptive) và trưng dụng (preemptive)**



CHƯƠNG 4. ĐỊNH THỜI CPU

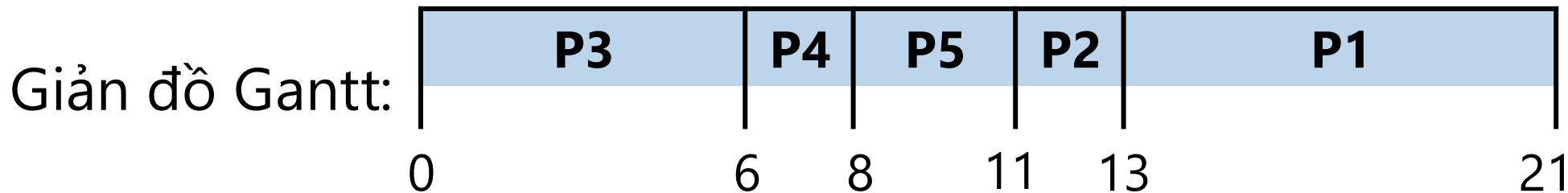
IV. Các giải thuật định thời

2. Shortest-Job-First (SJF) - không trưng dụng

Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

Thời gian đáp ứng = Thời gian chờ:
 $P1 = 3, P2 = 0, P3 = 0, P4 = 2, P5 = 6$

Thời gian đáp ứng/ đợi trung bình:
 $(P1 + P2 + P3 + P4 + P5) / 5 = 2.2$



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

2. Shortest-Job-First (SJF) - không trưng dụng

Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

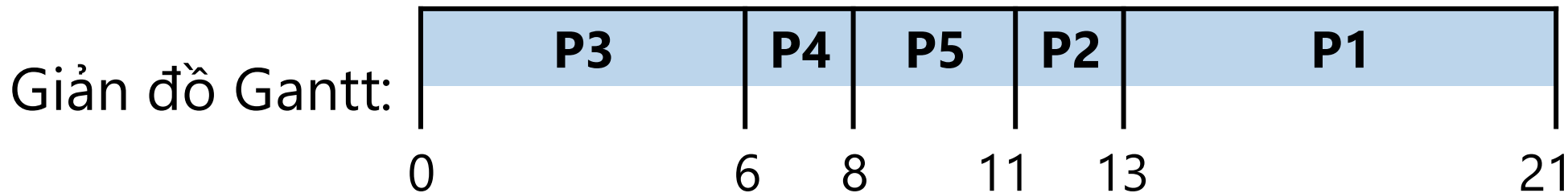
Thời gian hoàn thành:

$P1 = 11, P2 = 2, P3 = 6$

$P4 = 4, P5 = 9$

Thời gian hoàn thành trung bình:

$(P1 + P2 + P3 + P4 + P5) / 5 = 6.4$



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

2. Shortest-Remaining-Time-First (SRTF) - SJF trưng dụng

Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

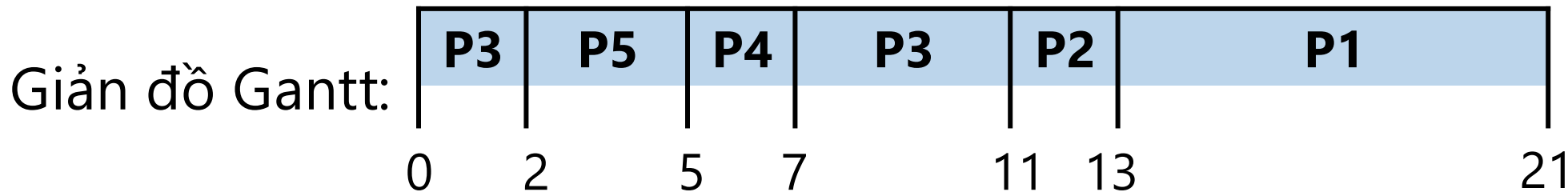
Khi $t = 0$: chỉ có RT_{P3}

Khi $t = 2$: $RT_{P3} = 4$, $RT_{P5} = 3$

Khi $t = 4$: $RT_{P3} = 4$, $RT_{P5} = 1$, $RT_{P4} = 2$

Khi $t = 10$: $RT_{P3} = 1$, $RT_{P1} = 8$

Khi $t = 11$: $RT_{P1} = 8$, $RT_{P2} = 2$



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

2. Shortest-Remaining-Time-First (SRTF) - SJF trưng dụng

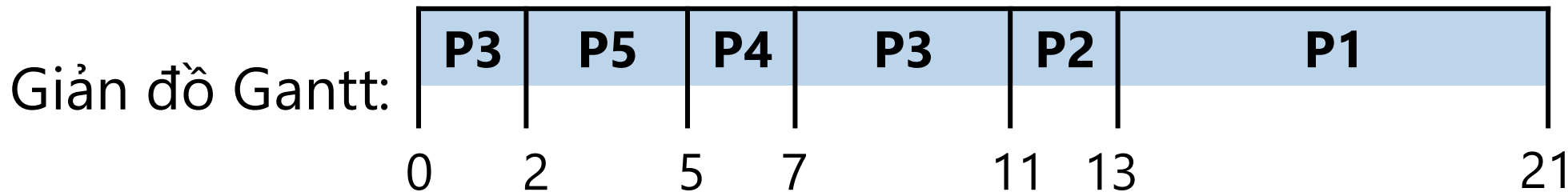
Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

Thời gian đáp ứng:

$P1 = 3, P2 = 0, P3 = 0, P4 = 1, P5 = 0$

Thời gian đáp ứng trung bình:

$(P1 + P2 + P3 + P4 + P5) / 5 = 0.8$



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

2. Shortest-Remaining-Time-First (SRTF) - SJF trưng dụng

Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

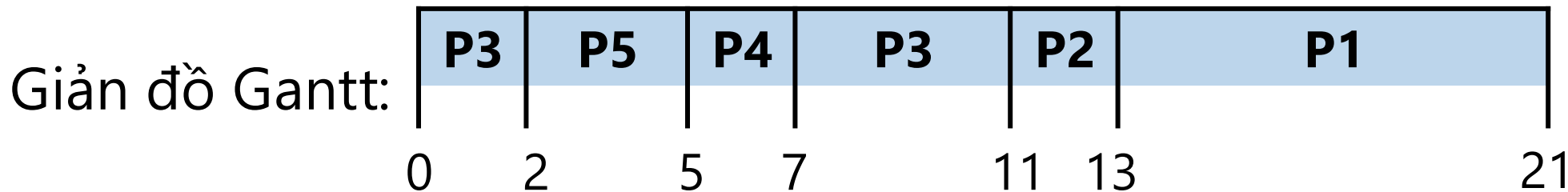
Thời gian đợi = Completion time – Arrival t – Burst t

Kết quả:

P1 = 3, P2 = 0, P3 = 5, P4 = 1, P5 = 0

Thời gian đợi trung bình:

$$(P1 + P2 + P3 + P4 + P5) / 5 = 1.8$$



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

2. Shortest-Remaining-Time-First (SRTF) - SJF trưng dụng

Process	Arrival Time	Burst time
P1	10	8
P2	11	2
P3	0	6
P4	4	2
P5	2	3

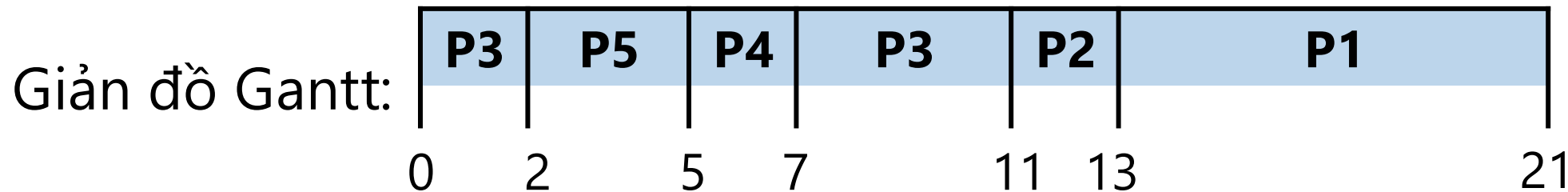
Thời gian hoàn thành:

$P1 = 11, P2 = 2, P3 = 11$

$P4 = 3, P5 = 3$

Thời gian hoàn thành trung bình:

$(P1 + P2 + P3 + P4 + P5) / 5 = 6$



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

3. Priority scheduling

- Mỗi process sẽ được gán một độ ưu tiên.
- CPU sẽ được cấp cho process có độ ưu tiên cao nhất.
- Định thời sử dụng độ ưu tiên có thể: Preemptive hoặc Non-preemptive.
- Khi thực hiện non-preemptive, về cơ bản nó giống như FCFS, priority chỉ có chức năng sắp xếp thứ tự các tiến trình trong hàng đợi.



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

3. Priority scheduling

- Hạn chế: trì hoãn vô hạn định, tiến trình có độ ưu tiên thấp có thể không bao giờ được thực thi (do có những tiến trình độ ưu tiên cao hơn liên tục xuất hiện).
- Giải pháp: làm mới (aging) – độ ưu tiên của tiến trình sẽ tăng theo thời gian.



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

3. Priority scheduling (không trưng dụng)

Process	Arrival Time	Burst time	Priority
P1	10	8	1
P2	11	2	2
P3	0	6	3
P4	4	2	4
P5	2	3	5

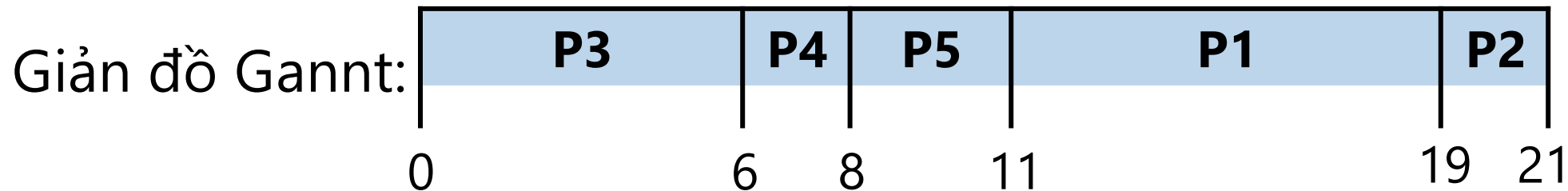
Thời gian đáp ứng:

$P1 = 1, P2 = 8, P3 = 0$

$P4 = 2, P5 = 6$

Thời gian đáp ứng trung bình: 3.4

Thời gian đợi, hoàn thành?



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

3. Priority scheduling (trưng dụng)

Process	Arrival Time	Burst time	Priority
P1	10	8	1
P2	11	2	2
P3	0	6	3
P4	4	2	4
P5	2	3	5

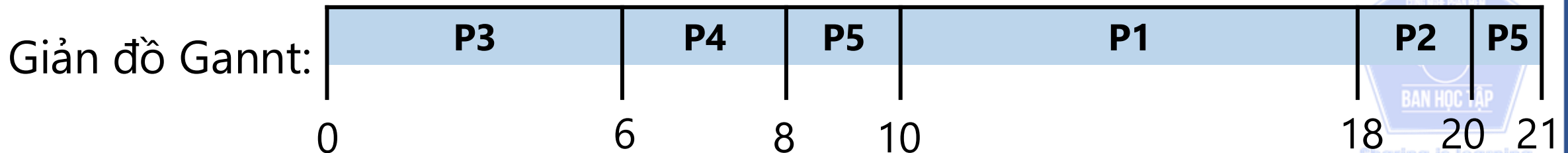
Thời gian đợi:

$P1 = 0, P2 = 7, P3 = 0$

$P4 = 2, P5 = 16$

Thời gian đợi trung bình: 5

Thời gian đáp ứng, hoàn thành?



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

4. Round robin

- Mỗi process nhận được một đơn vị nhỏ thời gian CPU (time slice, quantum time), thông thường từ 10-100 msec để thực thi.
- Sau khoảng thời gian đó, process bị đoạt quyền và trở về cuối hàng đợi ready.
- Nếu có n process trong hàng đợi ready và quantum time = q thì không có process nào phải chờ đợi quá $(n - 1)q$ đơn vị thời gian.



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

4. Round robin

Hiệu suất:

- Nếu q lớn: RR chính là FCFS.
- Nếu q nhỏ: q không được quá nhỏ bởi vì phải tốn chi phí chuyển ngữ cảnh.

RR sử dụng một giả thiết ngầm là tất cả các tiến trình đều có tầm quan trọng ngang nhau.

Ưu điểm: thời gian đáp ứng nhỏ.

Hạn chế: Thời gian chờ đợi trung bình và thời gian hoàn thành trung bình của giải thuật RR thường khá lớn.



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

4. Round robin

Process	Arrival Time	Burst time
P1	8	8
P2	9	2
P3	0	6
P4	4	2
P5	2	3

Giả sử $q = 2$:

Hàng đợi khi $t = 2$: P5 P3

Hàng đợi khi $t = 4$: P3 P4 P5

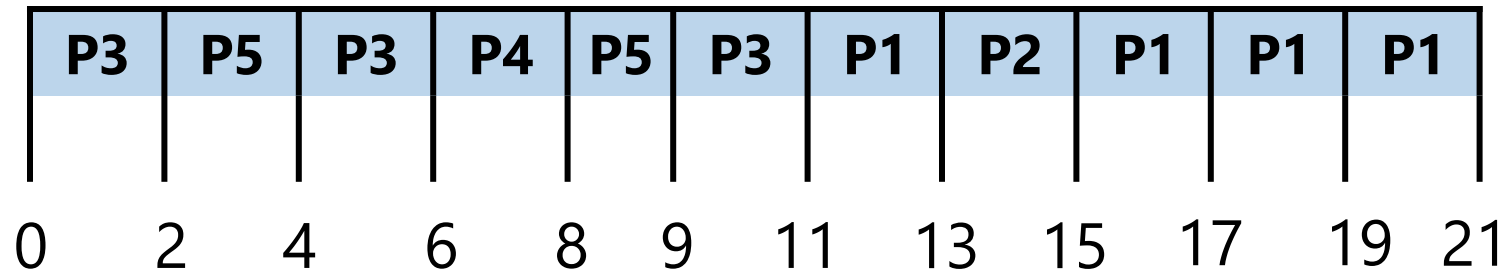
Hàng đợi khi $t = 6$: P4 P5 P3

Hàng đợi khi $t = 8$: P5 P3 P1

Hàng đợi khi $t = 9$: P3 P1 P2

Hàng đợi khi $t = 11$: P1 P2 ...

Giải đồ Gantt:



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

4. Round robin

Process	Arrival Time	Burst time
P1	8	8
P2	9	2
P3	0	6
P4	4	2
P5	2	3

Thời gian đáp ứng:

$P1 = 3, P2 = 4, P3 = 0, P4 = 2, P5 = 0$

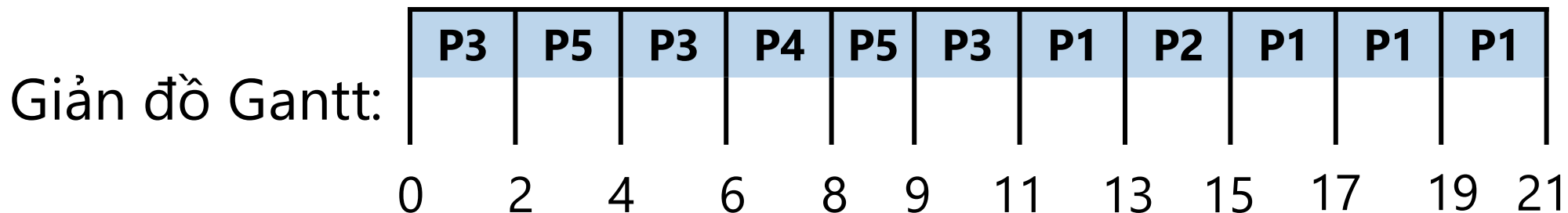
Thời gian đáp ứng trung bình: 2.8

Thời gian đợi:

$P1 = 5, P2 = 4, P3 = 5, P4 = 2, P5 = 4$

Thời gian đợi trung bình: 4

Thời gian hoàn thành?



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

4. Round robin

- Khi thực hiện process switch thì OS sẽ sử dụng CPU chứ không phải process của người dùng.
- Performance tùy thuộc vào kích thước của quantum time (còn gọi là time slice), và hàm phụ thuộc này không đơn giản.
- Time slice ngắn thì đáp ứng nhanh.

Vấn đề: có nhiều chuyển ngữ cảnh. Phí tổn sẽ cao.

- Time slice dài hơn thì throughput tốt hơn, nhưng thời gian đáp ứng lớn.



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

5. Highest-Response-Ratio-Next

Hàm lựa chọn:

- Chọn process có tỉ lệ phản hồi cao nhất.
- Các process ngắn được ưu tiên hơn.

Công thức:

$$RR = \frac{\text{time spent waiting} + \text{expected service time}}{\text{expected service time}}$$



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

6. Multilevel Queue

Ready queue được chia thành nhiều hàng đợi riêng biệt theo một số tiêu chuẩn sau:

- Đặc điểm và yêu cầu định thời của tiến trình
- Phân loại tiến trình: Foreground (interactive) và background,...

Tiến trình được gán cố định vào một hàng đợi, mỗi hàng đợi sử dụng giải thuật định thời riêng.



CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

6. Multilevel Queue

Hệ điều hành cần phải định thời cho các hàng đợi

- Fixed priority scheduling: phục vụ từng hàng đợi có độ ưu tiên cao đến thấp.

Vấn đề: Có thể gây ra tình trạng starvation (đói tài nguyên).

- Time slice: Mỗi hàng đợi được nhận một khoảng thời gian chiếm CPU và phân phối cho các process trong hàng đợi khoảng thời gian đó.

Ví dụ: 80% cho hàng đợi foreground định thời bằng Round Robin và 20% cho hàng đợi background định thời bằng giải thuật FCFS.



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

7. Multilevel Feedback Queue

- Phân loại tiến trình dựa trên các đặc tính về CPU burst.
- Sử dụng chế độ trưng dụng (preemptive).
- Sau một khoảng thời gian nào đó, các tiến trình hướng I/O và tiến trình interactive sẽ ở các hàng đợi có độ ưu tiên cao hơn còn các tiến trình hướng CPU sẽ ở các hàng đợi có độ ưu tiên thấp hơn.
- Một tiến trình đã chờ quá lâu ở một hàng đợi có độ ưu tiên thấp có thể được chuyển đến hàng đợi có độ ưu tiên cao hơn (cơ chế aging).



Sharing is learning

CHƯƠNG 4. ĐỊNH THỜI CPU

IV. Các giải thuật định thời

7. Multilevel Feedback Queue

Ưu và nhược điểm

- Là thuật toán định thời phổ biến và phức tạp nhất.
- Những quá trình trong thời gian t nào đó được đáp ứng nhanh

Yêu cầu cần giải quyết

- Số lượng hàng đợi bao nhiêu là thích hợp?
- Dùng giải thuật nào ở mỗi hàng đợi?
- Làm sao để xác định thời điểm để chuyển một process đến hàng đợi cao hoặc thấp hơn ?
- Khi process yêu cầu được xử lý thì hàng đợi nào là hợp lý nhất ?



Sharing is learning

GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

1. Bộ định thời nào xác định tiến trình nào trong ready queue sẽ được chiếm CPU để thực thi kế tiếp?

- A. Bộ định thời dài
- B. Bộ định thời vừa
- C. Bộ định thời ngắn
- D. Cả B và C đều đúng

=> Đáp án: C. Bộ định thời ngắn



GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

2. Hệ thống thông dịch lệnh là giao diện chủ yếu giữa hệ điều hành và cái gì?

- A. Phần cứng
- B. Tiến trình
- C. Phần mềm
- D. Người dùng

=> **Đáp án: D. Người dùng**



GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

3. Chọn phát biểu **SAI** trong các phát biểu sau:

- A. Hệ thống song song gồm hai hoặc nhiều bộ xử lý cùng chia sẻ một bộ nhớ.
- B. Hệ thống song song được chia làm hai loại là hệ thống đa xử lý đối xứng và hệ thống đa xử lý bất đối xứng.
- C. Hệ thống đa chương có đặc điểm: nhiều công việc được nạp đồng thời vào bộ nhớ chính, khi một tiến trình thực hiện I/O thì tiến trình khác thực thi.
- D. Hệ thống nhúng thời gian thực là một hệ thống cho kết quả chính xác nhất trong thời gian hiện tại.

=> **Đáp án: D. Hệ thống nhúng thời gian thực là một hệ thống cho kết quả chính xác nhất trong thời gian hiện tại.**



GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

4. Cấu trúc Modules là một trong những dạng cấu trúc của hệ điều hành. Đặc điểm nào sau đây KHÔNG phải của cấu trúc Modules:

- A. Sử dụng cách tiếp cận hướng đối tượng.
- B. Mỗi core thành phần tách biệt nhau.
- C. Giao tiếp giữa các user module qua cơ chế truyền thông điệp.
- D. Mỗi module như một phần của nhân.

=> Đáp án: C. Giao tiếp giữa các user module qua cơ chế truyền thông điệp.



GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

5. Điều kiện nào sau đây **KHÔNG** cần cho hoạt động đa chương của hệ điều hành?

- A. Định thời CPU
- B. Quản lý bộ nhớ
- C. Ứng dụng được lập trình đa nhiệm
- D. Cấp phát tài nguyên

=> **Đáp án: C. Ứng dụng được lập trình đa nhiệm**



GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

6. Giải thuật định thời nào sau đây chỉ có chế độ quyết định là không trưng dụng (non – preemptive):

- A. First-Come, First-Served (FCFS)
- B. Shortest-Job-First (SJF)
- C. Priority scheduling
- D. Round robin

=> Đáp án: A. First-Come, First-Served (FCFS)



Sharing is learning

GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

7. Mô hình nào sau đây **KHÔNG** phải mô hình đa tiểu trình:

- A. Một - Một
- B. Một - Nhiều
- C. Nhiều - Một
- D. Nhiều - Nhiều

=> **Đáp án: B. Một - Nhiều**



Sharing is learning

GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

8. Mô hình nào dưới đây giúp các tiến trình giao tiếp với nhau:

- A. Message passing
- B. Process Control Block
- C. Application Programming Interface
- D. System call

=> Đáp án: A. Message passing



Sharing is learning

GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

9. Chọn phát biểu **SAI** trong các phát biểu sau:

- A. Tiến trình con sẽ chia sẻ một phần tài nguyên của tiến trình cha.
- B. Tiến trình chỉ kết thúc khi thực thi lệnh cuối và gọi system routine exit.
- C. Không gian địa chỉ của tiến trình con được nhân bản từ cha hoặc được khởi tạo từ template.
- D. Tiến trình cha và con có thể thực thi đồng thời.

=> **Đáp án: B. Tiến trình chỉ kết thúc khi thực thi lệnh cuối và gọi system routine exit.**



GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

10. Chọn phát biểu ĐÚNG trong các phát biểu sau:

- A. Hệ điều hành là chương trình trung gian giữa phần cứng và phần mềm.
- B. Hệ điều hành không cung cấp dịch vụ phát hiện lỗi.
- C. CPU, Memory, thiết bị I/O ... là một phần của hệ điều hành.
- D. Mục đích chính của hệ thống xử lý đa chương là chia sẻ thời gian giữa các chương trình.

=> Đáp án: D. Mục đích chính của hệ thống xử lý đa chương là chia sẻ thời gian giữa các chương trình.



GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

11. Chuỗi trạng thái của tiến trình khi chạy chương trình trên lần lượt là:

Đáp án: C. new – ready – running – waiting
– ready – running – waiting – ready – running
– waiting – ready – running – terminated.

```
#include <stdio.h>
int main ()
{
    int a;
    for (int i = 0; i <= 1; i++)
    {
        if (i % 2 == 0)
        {
            scanf("%d", &a);
            printf ("a = %d\n", a);
        }
        else printf ("Bye");
    }
    exit (0);
}
```

GIẢI ĐỀ THI THỬ

I. Trắc nghiệm

12. Chọn phát biểu ĐÚNG trong các phát biểu sau về giải thuật định thời:

- A. Giải thuật Multilevel Queue cho phép tiến trình chuyển từ hàng đợi này sang hàng đợi khác.
- B. Giải thuật priority scheduling được tạo ra nhằm giải quyết vấn đề trì hoãn vô hạn định của giải thuật Shortest Job First.
- C. Ưu điểm của giải thuật Round robin là thời gian hoàn thành thường nhỏ.
- D. Giải thuật Round Robin sử dụng một giả thiết ngầm là tất cả các tiến trình đều có tầm quan trọng ngang nhau.

=> Đáp án: D. Giải thuật Round Robin sử dụng một giả thiết ngầm là tất cả các tiến trình đều có tầm quan trọng ngang nhau.



Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 1: Phát biểu sau **ĐÚNG** hay **SAI**?

a. Ưu điểm của hệ thống song song (đa xử lý) là năng suất cao và khi một bộ xử lý hỏng thì không ảnh hưởng đến toàn bộ hệ thống.

=> **ĐÚNG**

b. Nhân của hệ điều hành Linux và Solaris theo cấu trúc kết hợp không gian địa chỉ kernel, cấu trúc monolithic và modules.

=> **ĐÚNG**



Sharing is learning

GIẢI ĐỀ THI THỬ

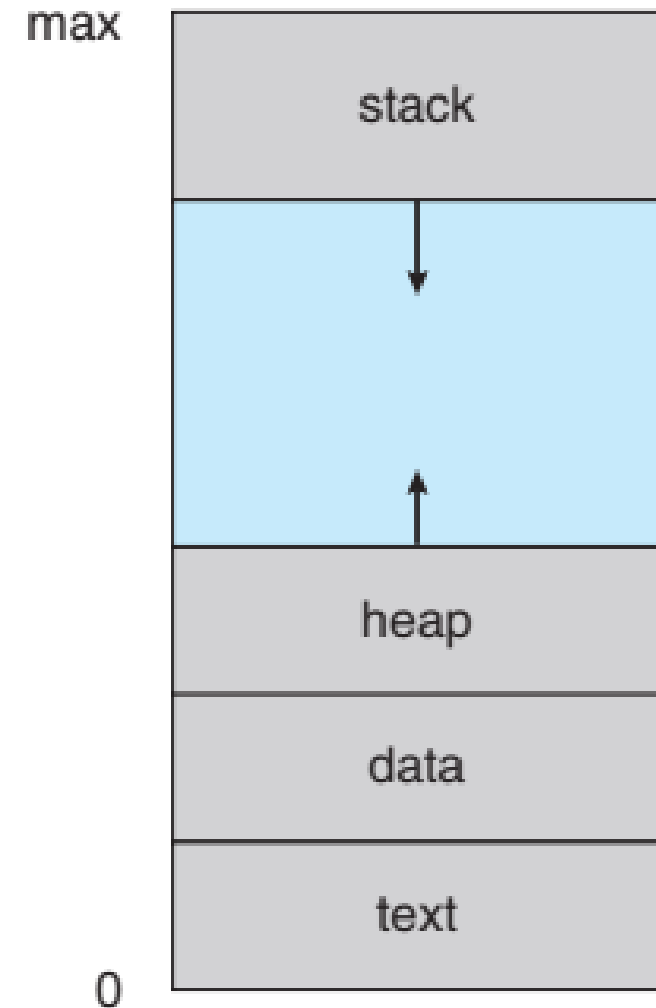
II. Tự luận

Câu 2:

a. Khi chương trình trên được nạp vào bộ nhớ, biến x và i sẽ nằm ở đâu?

Bộ nhớ có 4 phần:

- Stack
- Heap
- Data section
- Text section



GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 2:

a. Khi chương trình trên được nạp vào bộ nhớ, biến x và i sẽ nằm ở đâu?

- Biến x: data section
- Biến i: stack

```
int x = 2;
int main()
{
    printf("Hello\n");
    int i = 0;
    while (i < x)
    {
        printf("Hi\n");
        int k = 0;
        if (i % 2 == 0)
            k = fork();
        if (k > 0)
        {
            fork();
            printf("Hi\n");
        }
        else
            fork();
        i++;
    }
    printf("Bye\n");
}
```



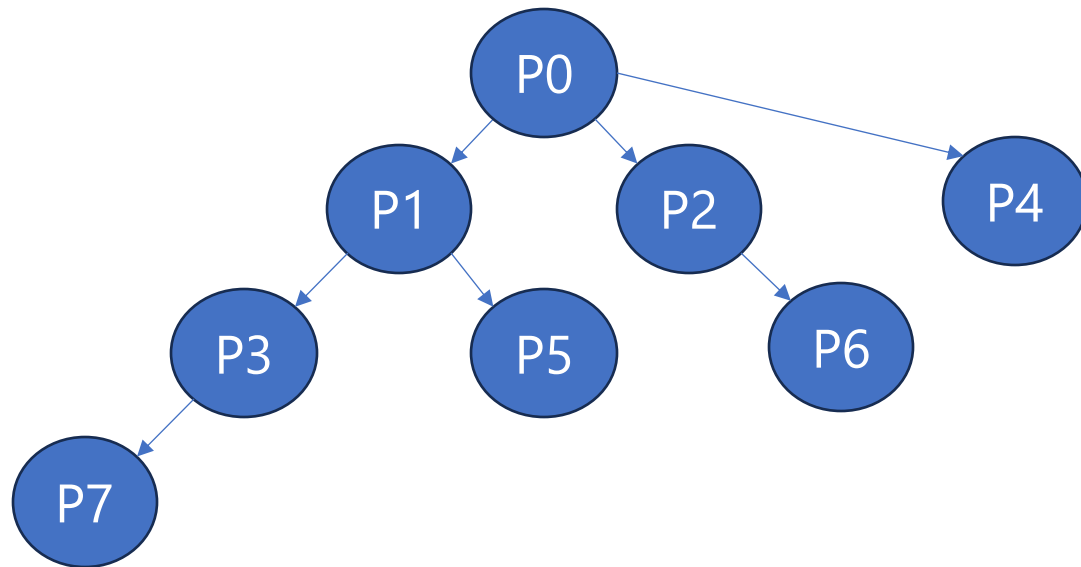
Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 2:

b. Có bao nhiêu tiến trình (kể cả tiến trình cha) được tạo ra?



Đáp án: 8 tiến trình

```
int x = 2;
int main()
{
    printf("Hello\n");
    int i = 0;
    while (i < x)
    {
        printf("Hi\n");
        int k = 0;
        if (i % 2 == 0)
            k = fork();
        if (k > 0)
        {
            fork();
            printf("Hi\n");
        }
        else
            fork();
        i++;
    }
    printf("Bye\n");
}
```



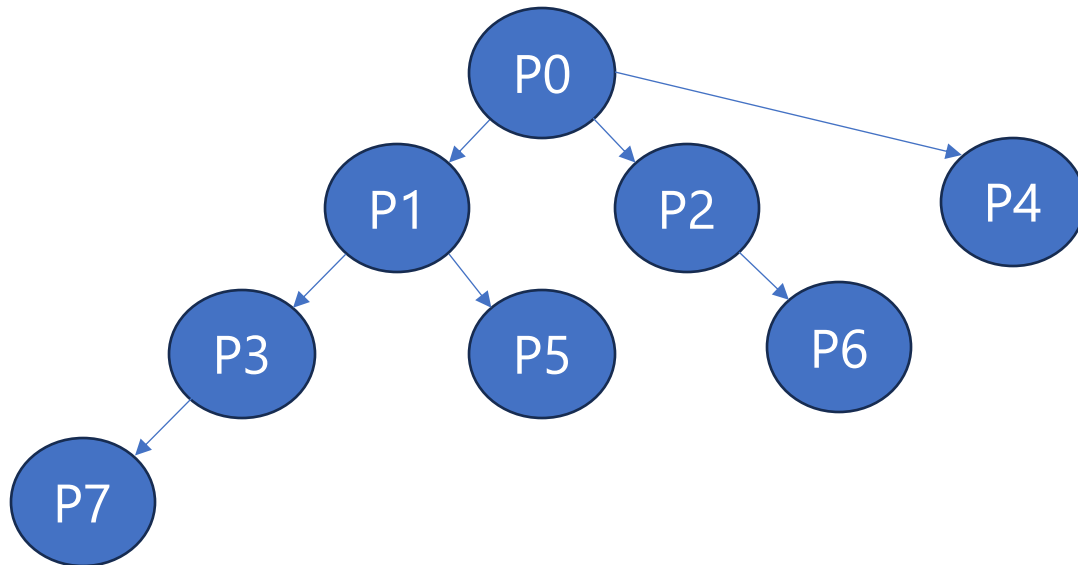
Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 2:

c. Có bao nhiêu từ "Hi" được in ra ?



Đáp án: 7 chữ "Hi"

```
int x = 2;
int main()
{
    printf("Hello\n");
    int i = 0;
    while (i < x)
    {
        printf("Hi\n");
        int k = 0;
        if (i % 2 == 0)
            k = fork();
        if (k > 0)
        {
            fork();
            printf("Hi\n");
        }
        else
            fork();
        i++;
    }
    printf("Bye\n");
}
```



Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 2:

d. Tất cả các tiến trình trong đoạn chương trình trên nằm trong hàng đợi waiting bao nhiêu lần?

Đáp án: 16 lần gồm:

- 7 lần `printf(Hi)`
- 1 lần `printf>Hello)`
- 8 lần `printf(Bye)`

```
int x = 2;
int main()
{
    printf("Hello\n");
    int i = 0;
    while (i < x)
    {
        printf("Hi\n");
        int k = 0;
        if (i % 2 == 0)
            k = fork();
        if (k > 0)
        {
            fork();
            printf("Hi\n");
        }
        else
            fork();
        i++;
    }
    printf("Bye\n");
}
```



Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 3a:

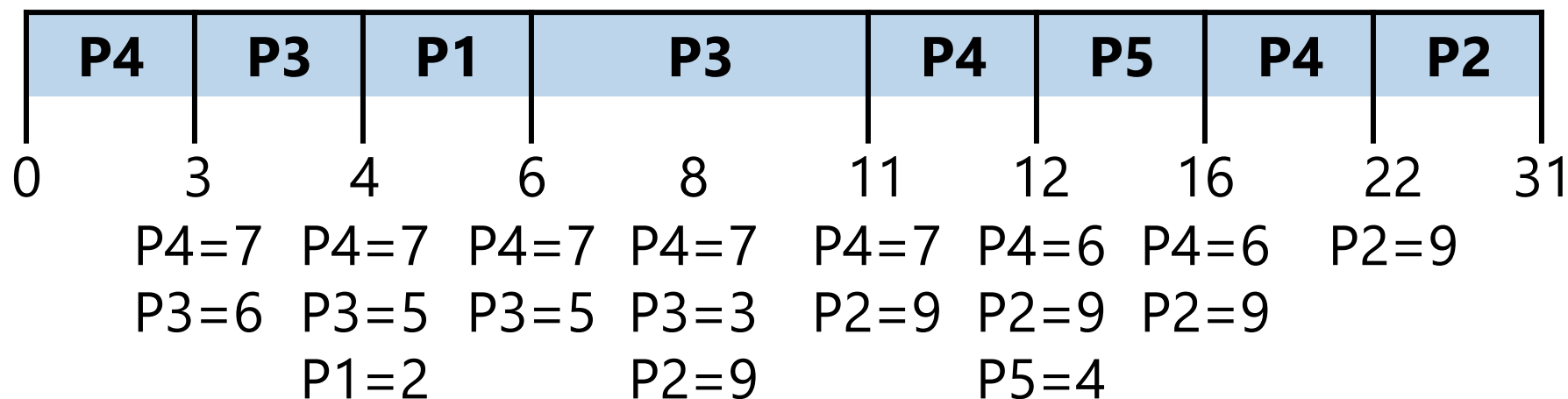
Thời gian đáp ứng:

$P1 = 0, P2 = 14, P3 = 0, P4 = 0, P5 = 0$

Thời gian đáp ứng trung bình: 2.8

Process	Arrival Time	Burst time
P1	4	2
P2	8	9
P3	3	6
P4	0	10
P5	12	4

Giải đồ Gantt



GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 3a:

Thời gian đợi:

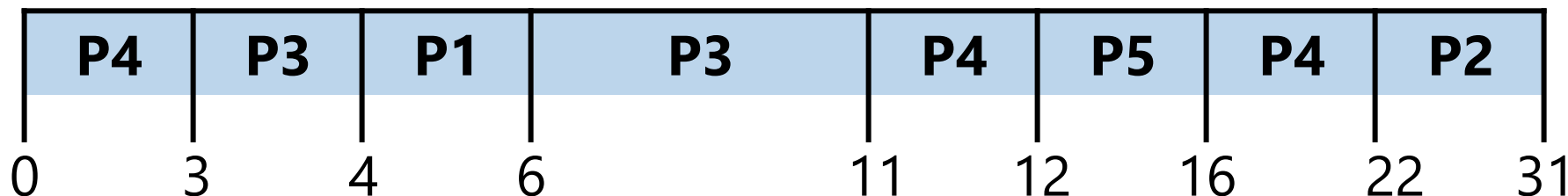
$P1 = 0, P2 = 14, P3 = 2, P4 = 12, P5 = 0$

Thời gian đợi trung bình:

$$(P1 + P2 + P3 + P4 + P5) / 5 = 5.6$$

Process	Arrival Time	Burst time
P1	4	2
P2	8	9
P3	3	6
P4	0	10
P5	12	4

Giản đồ Gantt



Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 3a:

Thời gian hoàn thành:

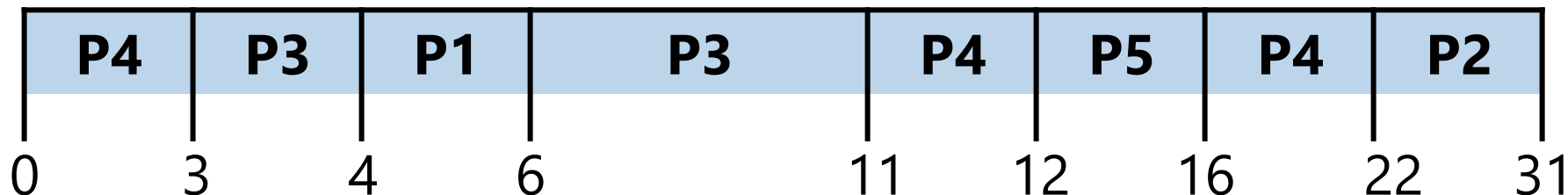
$P1 = 2, P2 = 23, P3 = 8, P4 = 22, P5 = 4$

Thời gian hoàn thành trung bình:

$$(P1 + P2 + P3 + P4 + P5) / 5 = 11.8$$

Process	Arrival Time	Burst time
P1	4	2
P2	8	9
P3	3	6
P4	0	10
P5	12	4

Giản đồ Gantt



Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 3b:

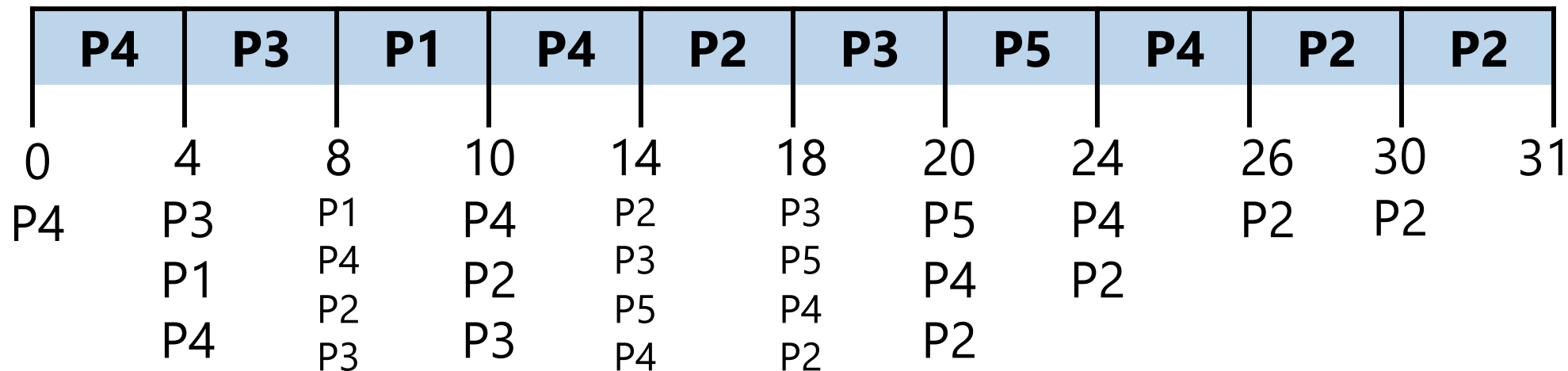
Thời gian đáp ứng:

$P1 = 4, P2 = 6, P3 = 1, P4 = 0, P5 = 8$

Thời gian đáp ứng trung bình: 3.8

Process	Arrival Time	Burst time
P1	4	2
P2	8	9
P3	3	6
P4	0	10
P5	12	4

Giản đồ Gantt



Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 3b:

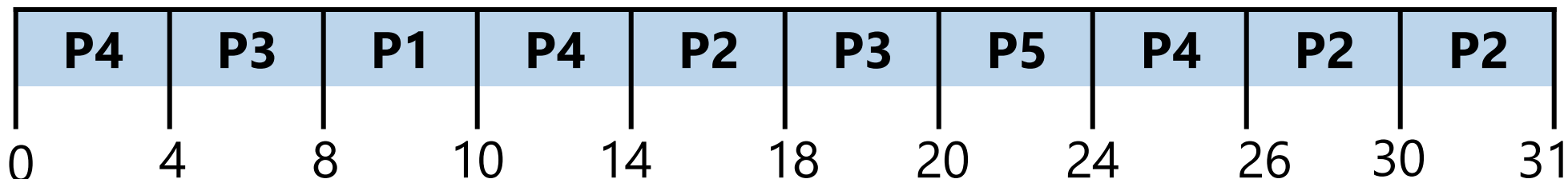
Thời gian đợi:

$P1=4, P2=14, P3=11, P4=16, P5=8$

Thời gian đợi trung bình: 10.6

Process	Arrival Time	Burst time
P1	4	2
P2	8	9
P3	3	6
P4	0	10
P5	12	4

Giản đồ Gantt



Sharing is learning

GIẢI ĐỀ THI THỬ

II. Tự luận

Câu 3b:

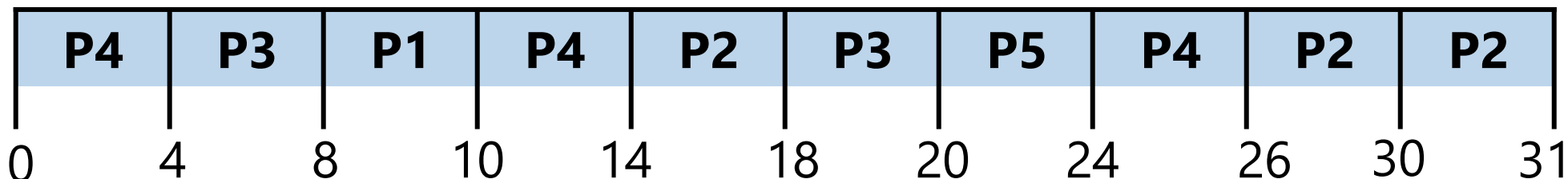
Thời gian hoàn thành:

$P1=6, P2=23, P3=17, P4=26, P5=12$

Thời gian hoàn thành trung bình: 16.8

Process	Arrival Time	Burst time
P1	4	2
P2	8	9
P3	3	6
P4	0	10
P5	12	4

Giản đồ Gantt



Sharing is learning



- Training
- Giải đáp
- Chia sẻ



- Design ấn phẩm
- Viết content
- Chụp ảnh



- Instagram
- TikTok
- Dịch thuật
- Thi thử



BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ II NĂM HỌC 2023 – 2024



Sharing is learning

HẾT

**CẢM ƠN CÁC BẠN ĐÃ THEO DÕI
CHÚC CÁC BẠN CÓ KẾT QUẢ THI THẬT TỐT!**

 **BAN HỌC TẬP**

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

 **CONTACT**

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit

TEAM TIẾNG ANH

english.with.bht@gmail.com

 *creative.owl.se*

 *english.with.bht*