BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2023 – 2024

ĐOÀN KHOA
CÔNG NGHỆ PHẦN MỀM
BAN HỌC TẬP

Sharing is learning

.featured .item .icon

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

TRAINING

HỆ ĐIỀU HÀNH

Thời gian: 19:30 thứ Hai ngày 23/10/2023

Địa điểm: Microsoft Teams – Code: w2dsy1q

Trainers: Phan Nguyễn Tuấn Anh – KTPM2022.1

Lê Dương Minh Thiên – KHMT2022.4

Nguyễn Lâm Thanh Triết – KTPM2022.3

Sharing is learning

CÁC CHƯƠNG

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

CÁC CHƯƠNG

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản
II. Phân loại hệ điều hành

CHƯƠNG 1: TỔNG QUAN VỀ 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 VỀ 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

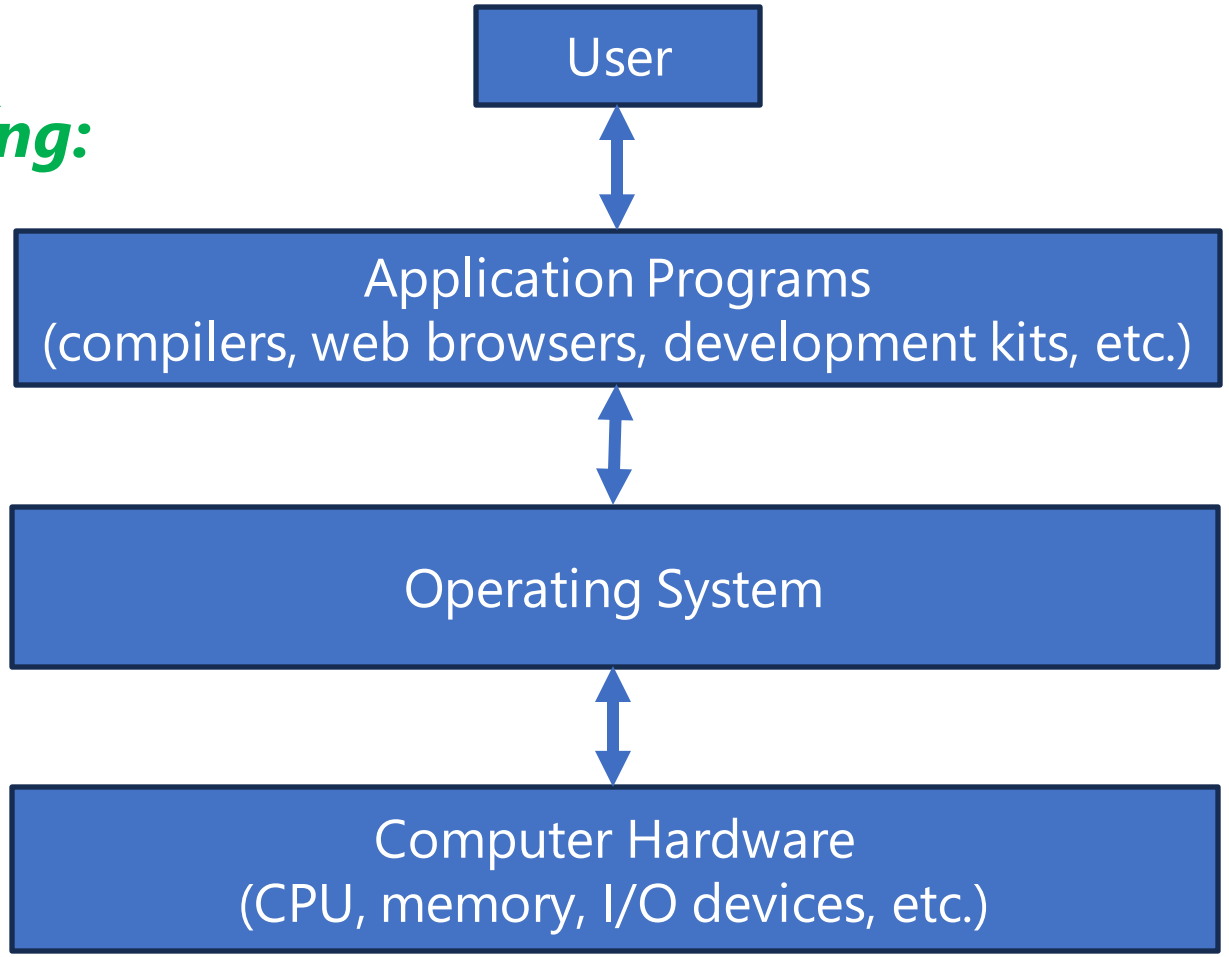
***Chức năng*:**

- 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.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản

***Cấu trúc hệ thống:**



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

I. Tổng quan cơ bản

Trắc nghiệm khách quan

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.

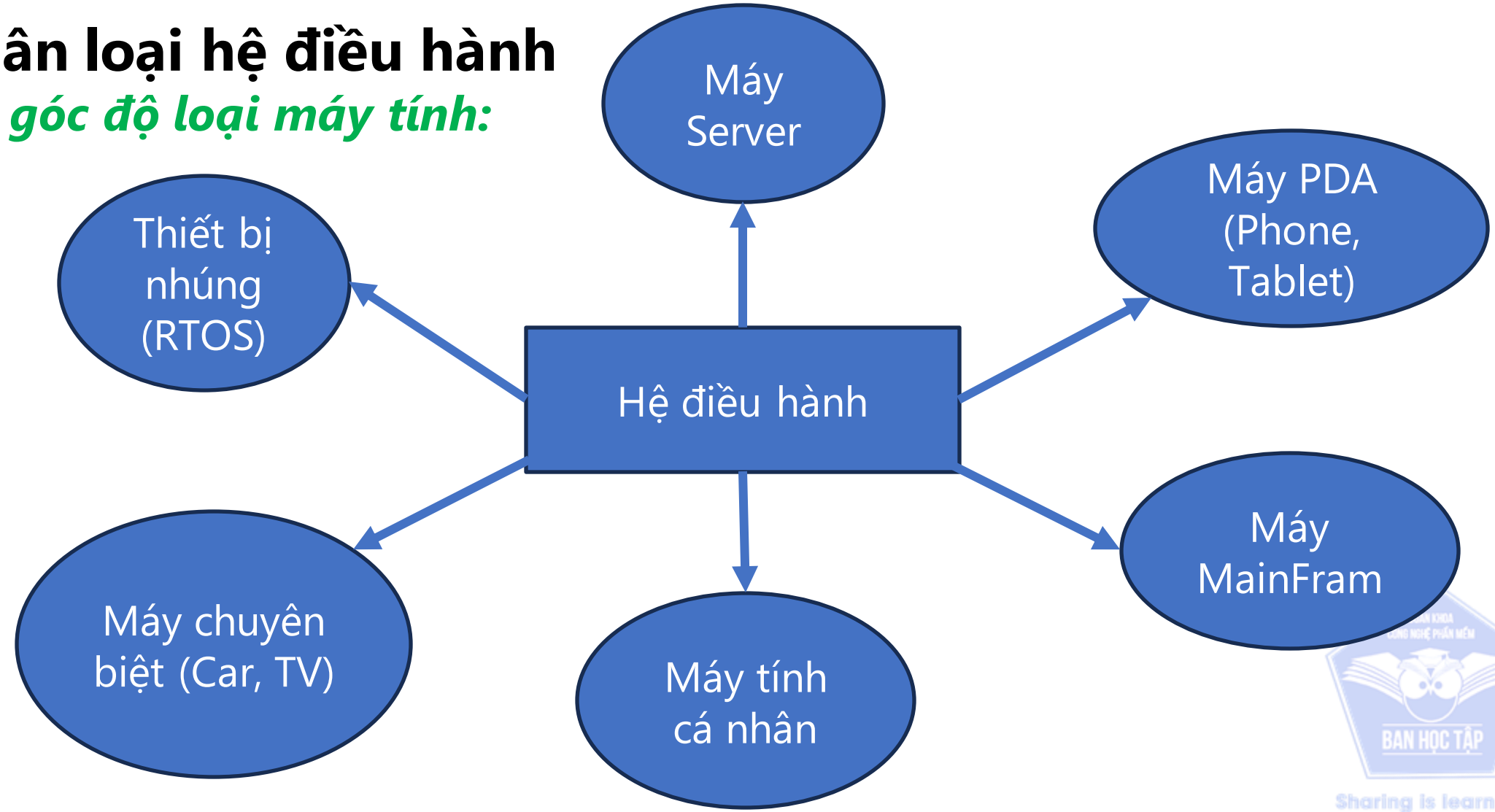
2. Trong phân lớp hệ thống máy tính. Hệ điều hành thuộc vị trí nào:

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.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

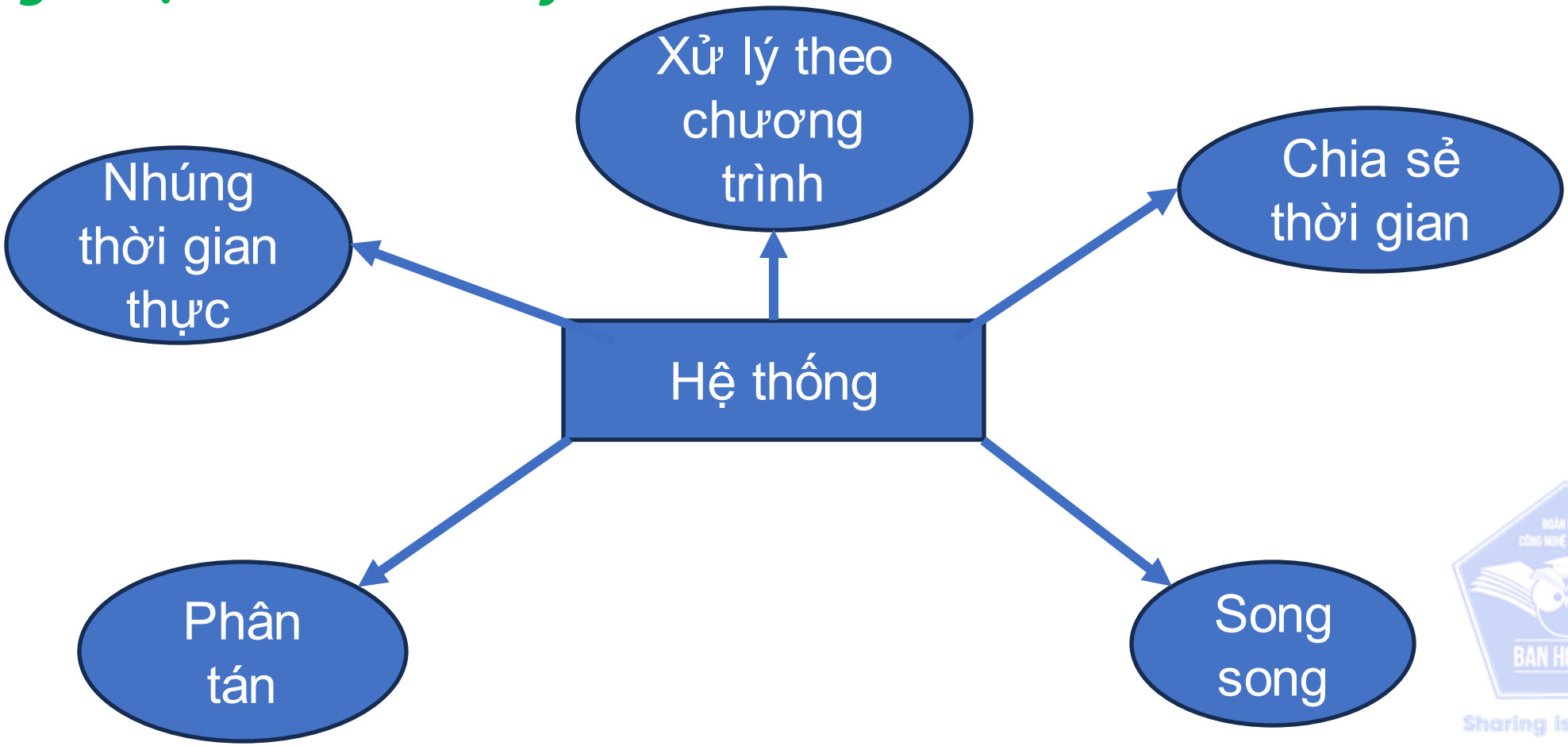
***Dưới góc độ loại máy tính:**



CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

II. Phân loại hệ điều hành

***Dưới góc độ hình thức xử lý:**



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.

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.

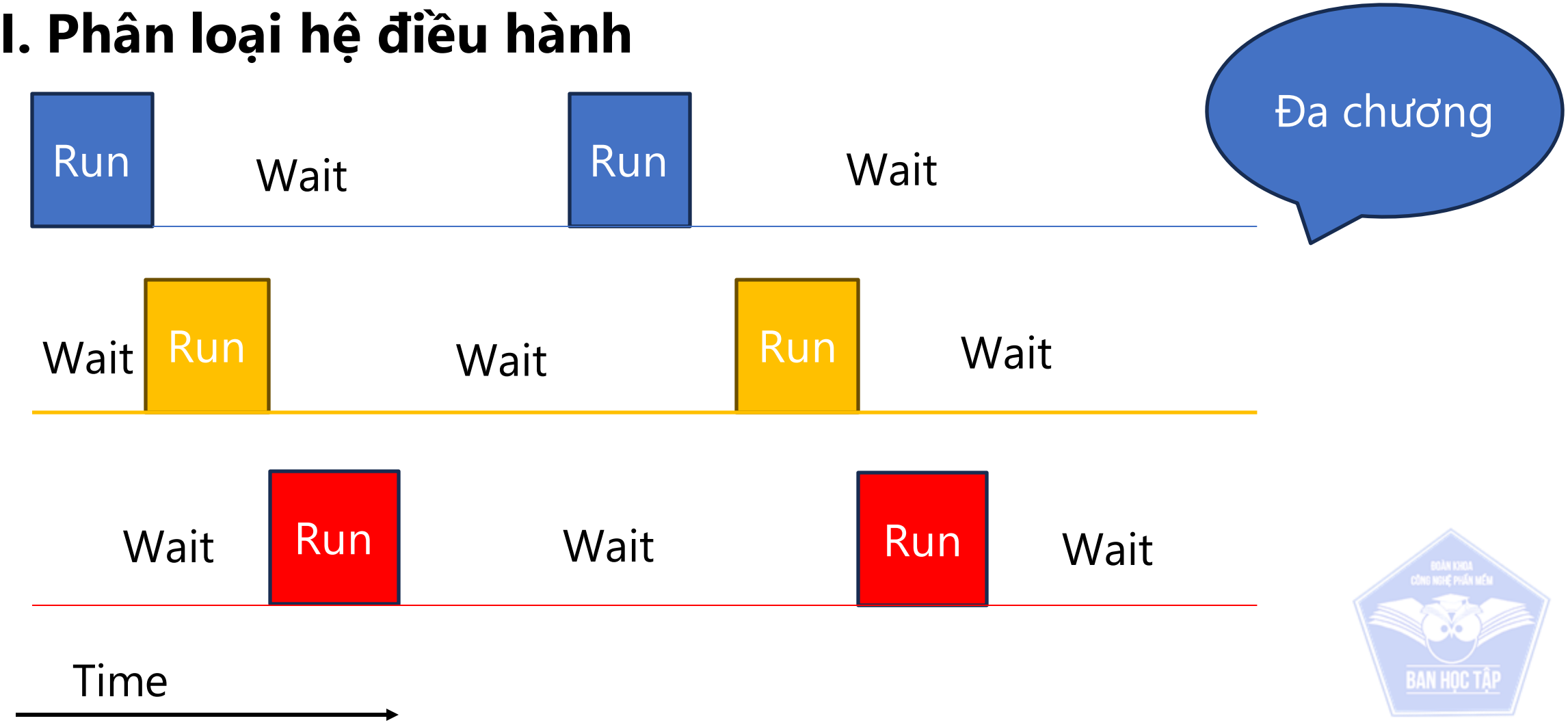
- Khi một tiến trình thực hiện I/O, một tiến trình khác được thực thi.

→ 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



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:

- Quản lý các quá trình.
  - + 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

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
• Mỗi process vận hành một bản sao hệ điều hành giống nhau. • Copy dữ liệu cho nhau khi cần.	• 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.

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

Trắc nghiệm khách quan

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

A. Chương trình khi nạp vào bộ nhớ sẽ được xử lí 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 lí 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 lí cùng chia sẻ hệ thống đường truyền, dữ liệu, bộ nhớ, các thiết bị ngoại vi?

A. Hệ thống xử lí đa chương.
B. Hệ thống xử lí đa nhiệm.
C. Hệ thống xử lí song song.
D. Hệ thống xử lí thời gian thực.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ ĐIỀU HÀNH

Trắc nghiệm khách quan

1. Hệ thống xử lí 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 lí 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.

CÁC CHƯƠNG

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

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

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

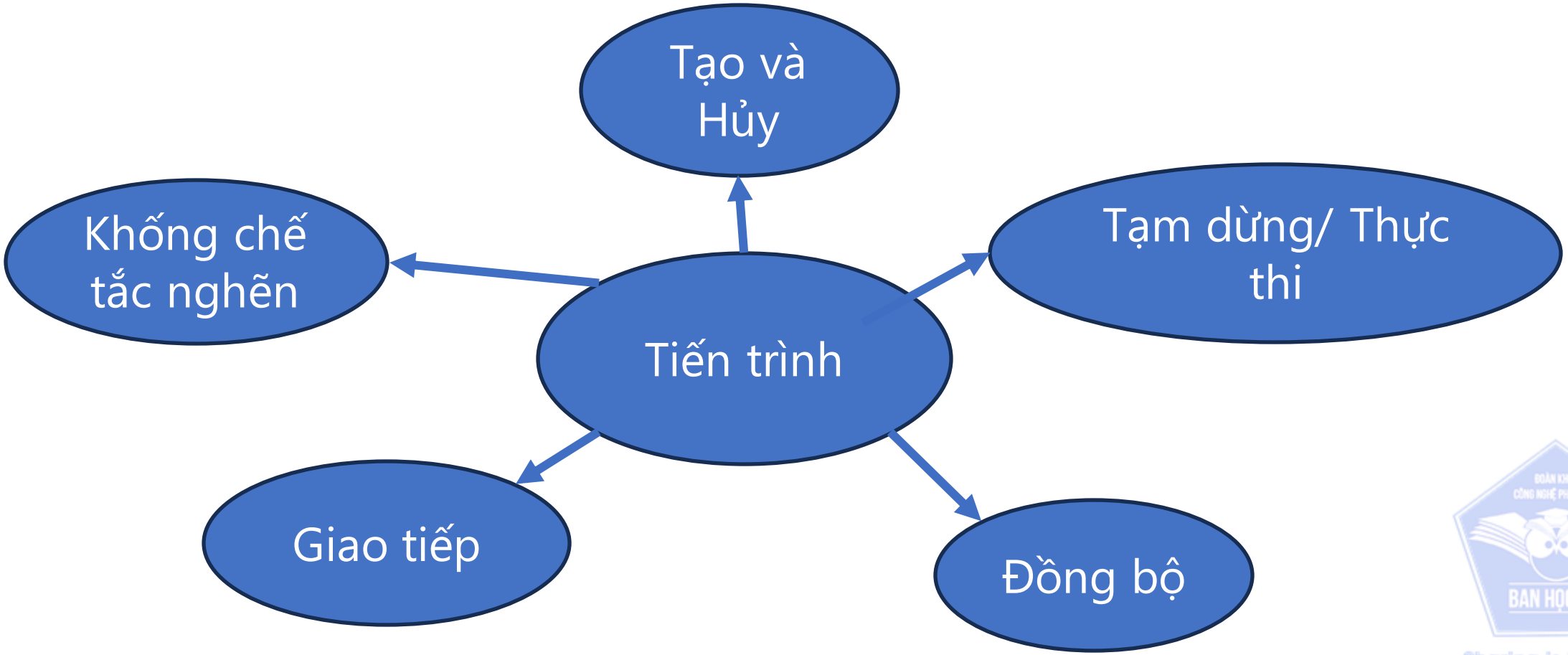
I. Các thành phần của hệ điều hành

1. Quản lý tiến trình
2. Quản lý bộ nhớ chính
3. Quản lý file
4. Quản lý hệ thống I/O
5. Quản lý hệ thống lưu trữ thứ cấp
6. Hệ thống bảo vệ
7. Hệ thống thông dịch lệnh

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

1. Quản lý tiến trình:

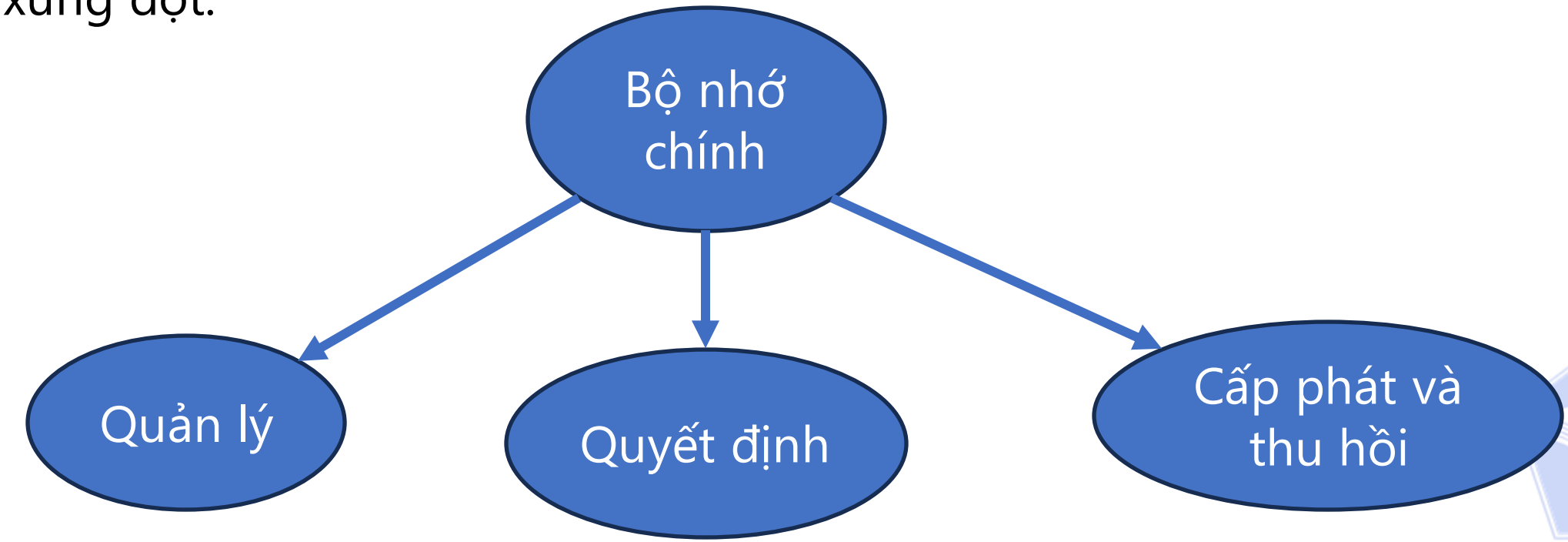


CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

2. Quản lý bộ nhớ chính:

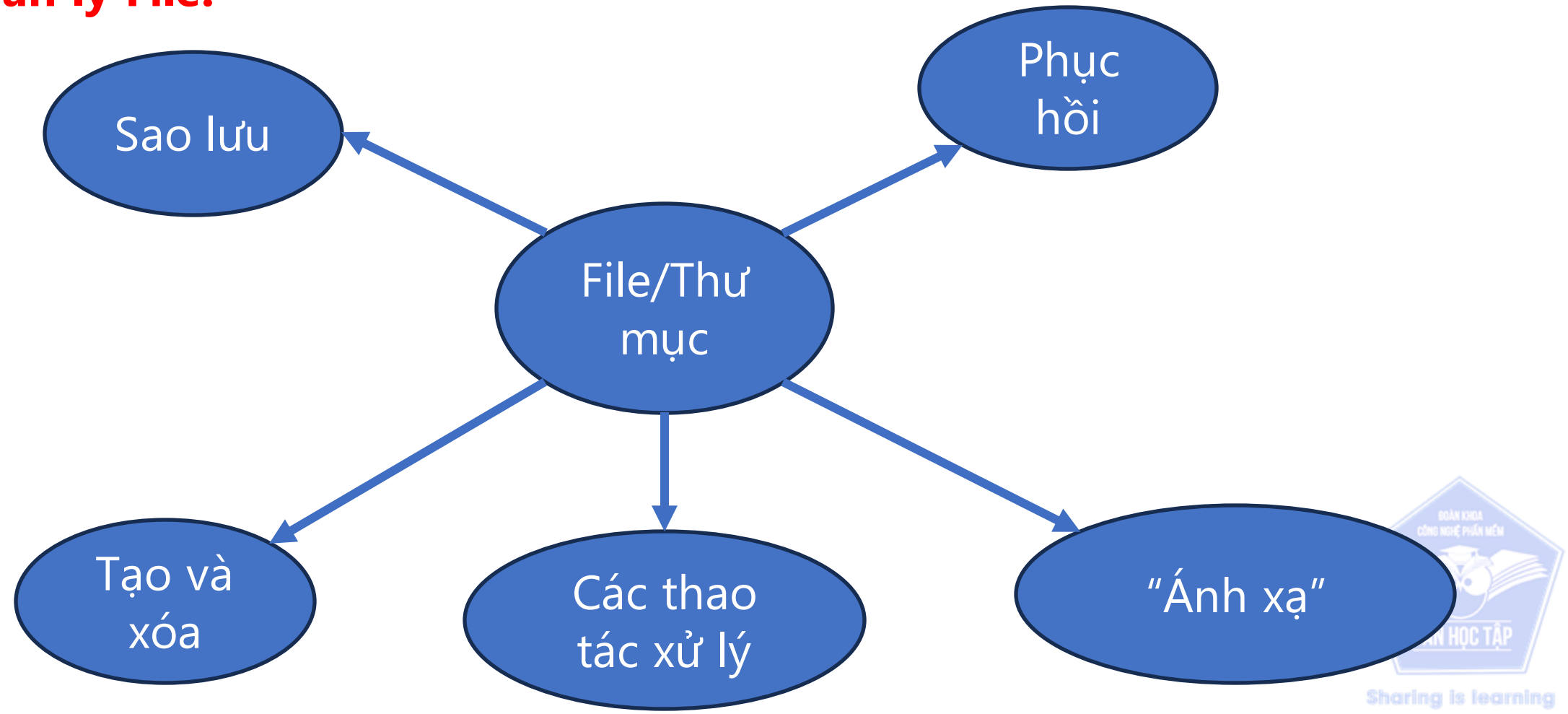
- Bộ nhớ chính là trung tâm của các thao tác, xử lý
- Hệ điều hành cần quản lý bộ nhớ đã cấp phát cho mỗi tiến trình để tránh xung đột.



CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

3. Quản lý File:

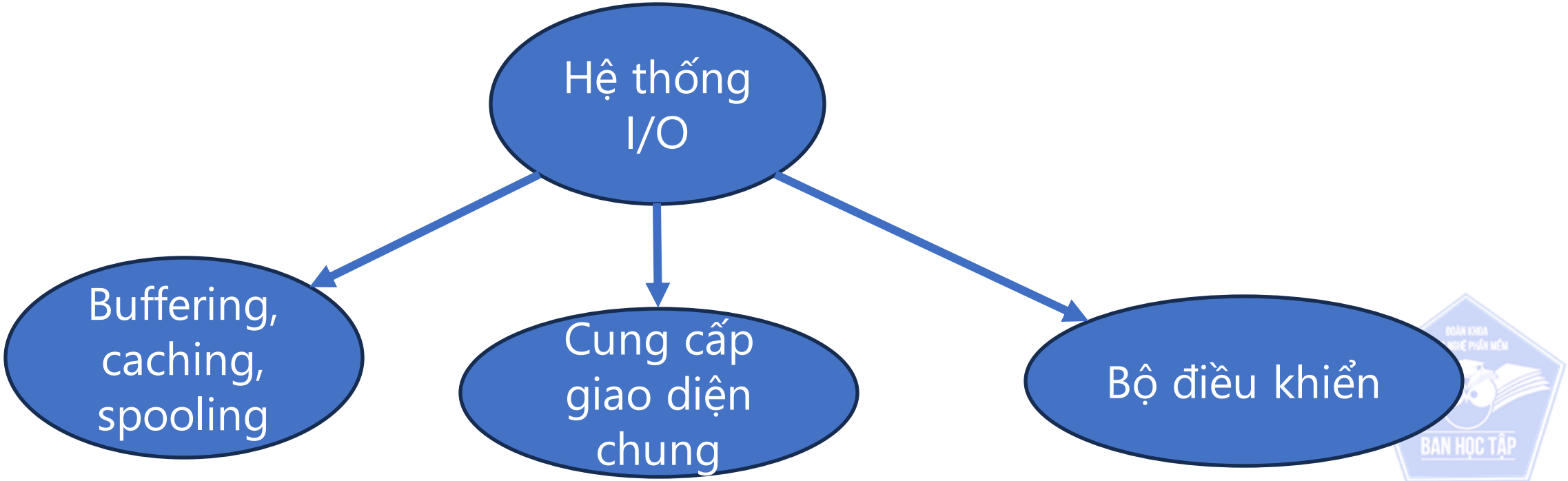


CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

4. Quản lý thiết bị I/O:

- Che dấu sự khác của các thiết bị I/O trước người dùng.

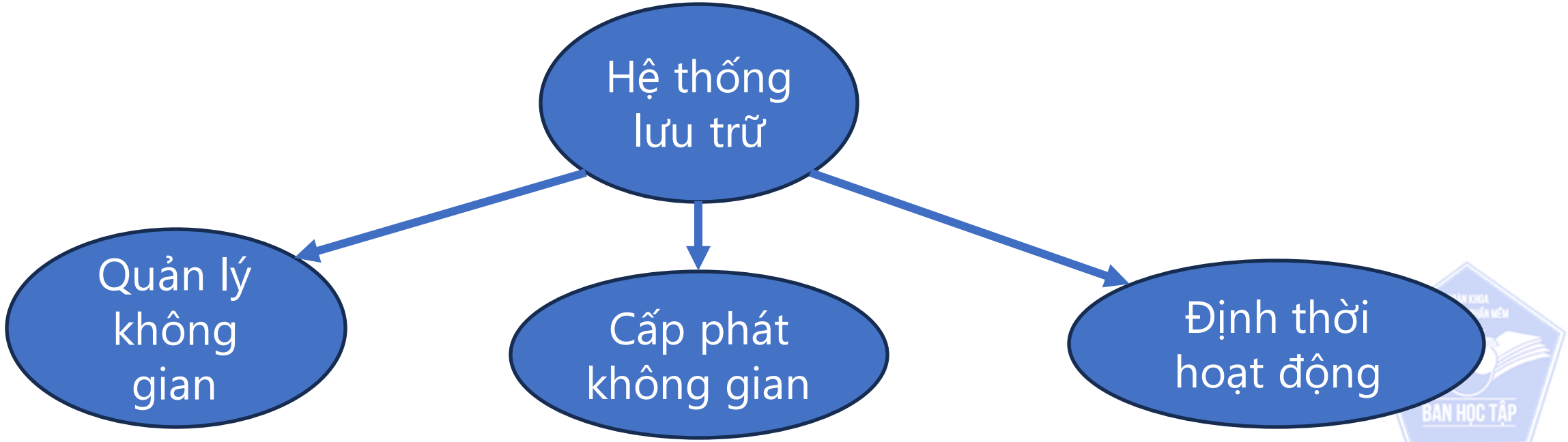


CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

5. Quản lý hệ thống lưu trữ thứ cấp:

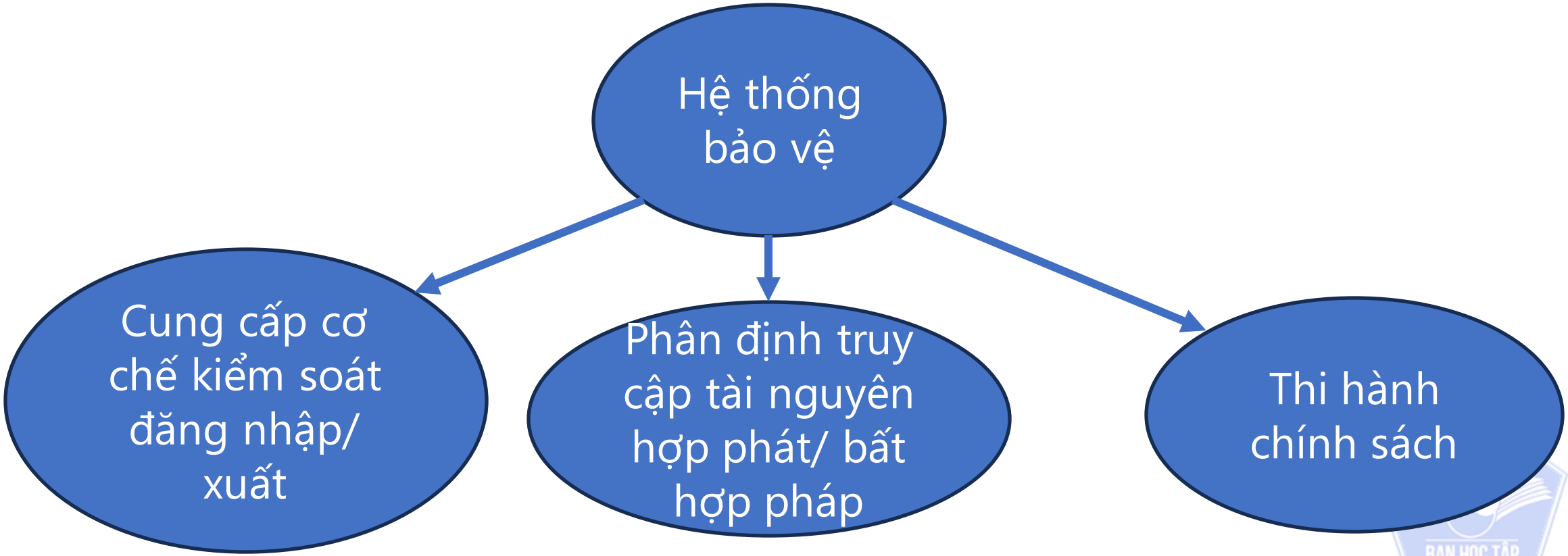
- Lưu trữ bền vững dữ liệu, chương trình.
- Phương tiện lưu trữ: đĩa từ, đĩa quang (HDD, SDD).



CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

6. Quản lý hệ thống bảo vệ:

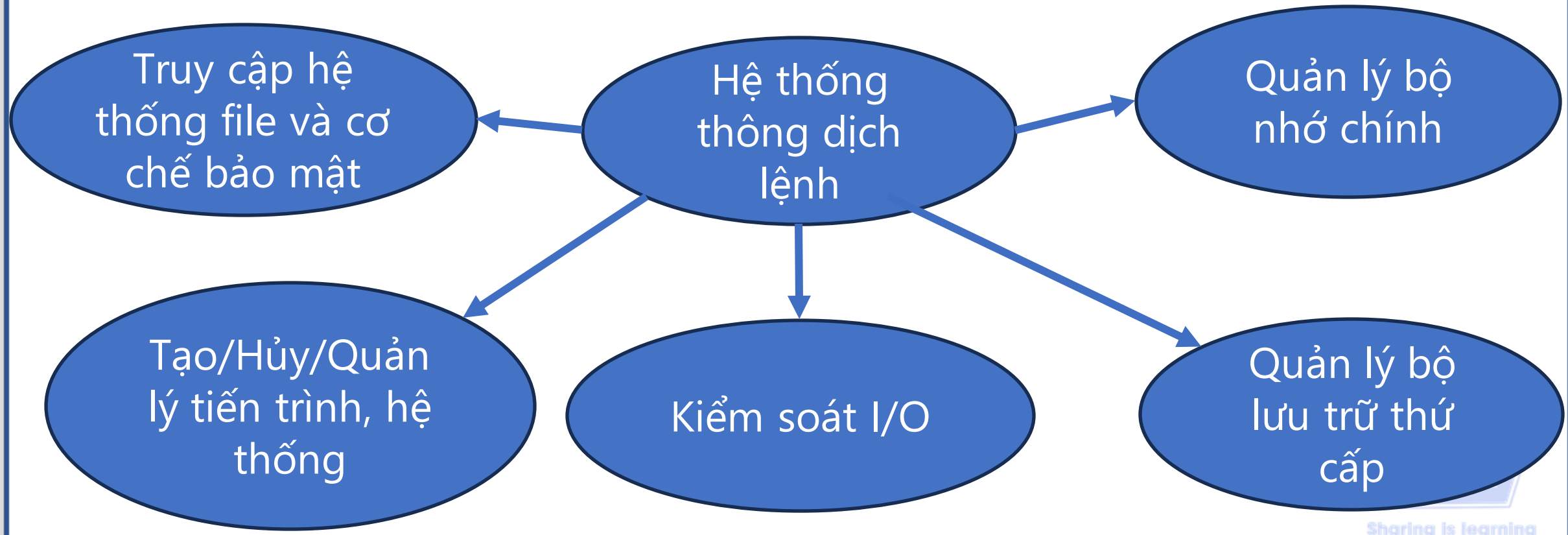


CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

I. Các thành phần của hệ điều hành

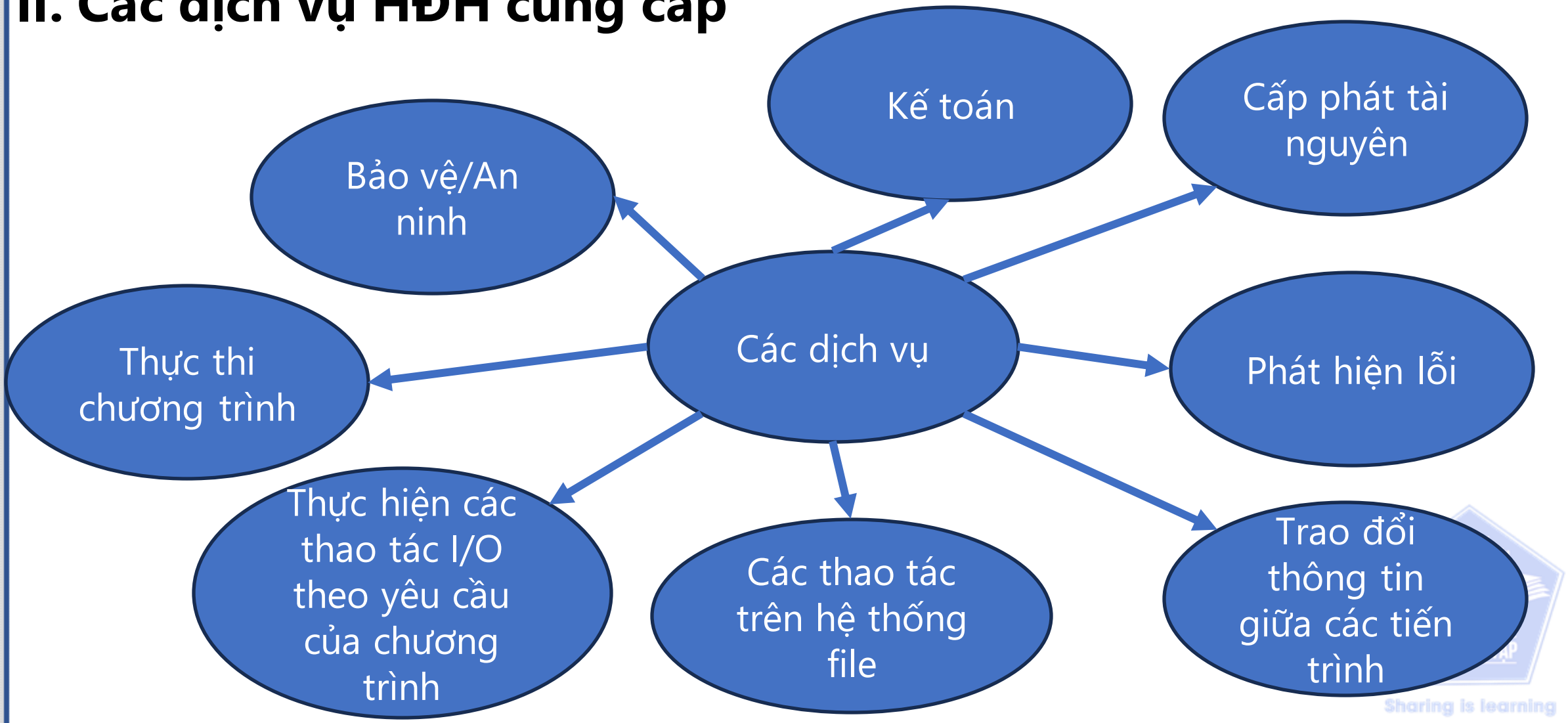
7. Quản lý hệ thống thông dịch lệnh:

- Là giao diện cơ bản để người dung giao tiếp với HĐH.



CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

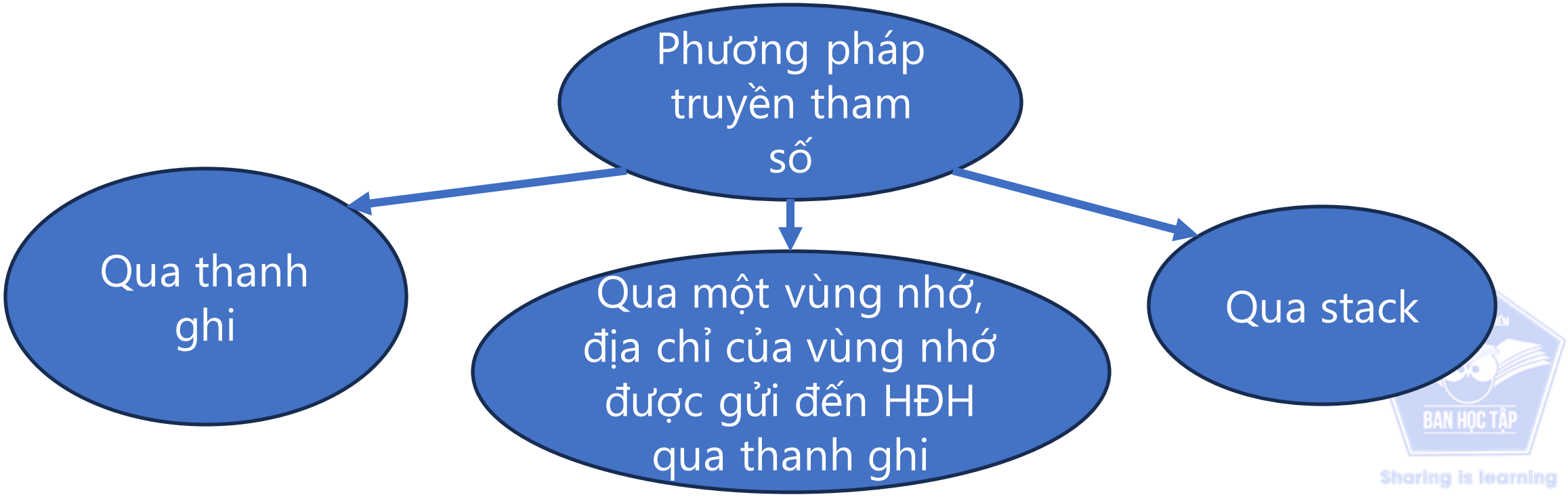
II. Các dịch vụ HĐH cung cấp



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 các lệnh hợp ngữ.



CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

4. 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).

- **Các chương trình hệ thống** (system program # application program) gồm:

➢ 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,.

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

5. Cấu trúc hệ thống

5.1. Cấu trúc Monolithic-Original UNIX

- UNIX **giới hạn** về chức năng phần cứng nên Original UNIX cũng có cấu trúc **rất giới hạn.**

- UNIX gồm 2 phần tách rời:

➢ Nhân: Cung cấp file system, CPU scheduling,..

➢ System program.

⇒ LINUX dựa theo cấu trúc monolithic được thiết kế theo dạng mô-đun.

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

5. Cấu trúc hệ thống

5.2. 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 và các hàm của nó được gọi bởi lớp trên.
- **VD:** Hệ điều hành THE
- **Lợi ích** của phân lớp:
  - Gỡ rối (debugger)
  - Kiểm tra hệ thống
  - Thay đổi chức năng

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

5. Cấu trúc hệ thống

5.3. Cấu trúc Microkernels

- Phân chia module theo microkernel.
- Chuyển một số chức năng của OS từ kernel space sang user space.
- Thu gọn kernel => microkernel.
- Microkernels có chức năng tối thiểu:
  - Quản lý tiến trình, bộ nhớ.
  - Cơ chế giao tiếp giữa các tiến trình.
- Giao tiếp giữa các user module qua cơ chế truyền thông điệp.

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

5. Cấu trúc hệ thống

5.4. Cấu trúc Module

- Nhiều hệ điều hành hiện đại triển khai các 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.

- **VD**: Linux, Solaris,...

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

5. Cấu trúc hệ thống

5.5. 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.

- Cấu trúc lai là sự kết hợp nhiều cách tiếp cận để giải quyết các nhu cầu về hiệu suất, bảo mật, nhu cầu sử dụng.
- 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 liền khối, cộng với cấu trúc vi nhân cho các hệ thống cá nhân khác nhau.

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

Trắc nghiệm khách quan

1. Hệ thống thông dịch lệnh là giao diện giữa hệ điều hành và …?

A. Người dùng
B. Các phần mềm.
C. Dữ liệu.
D. Bộ nhớ.

2. Hệ thống bảo vệ có nhiệm vụ gì?

A. Cung cấp cơ chế kiểm soát đăng xuất, nhập xuất.
B. Phân định được sự truy cập tài nguyên hợp pháp và bất hợp pháp.
C. Phương tiện thi hành các chính sách.
D. Tất cả các phương án trên.

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

A. Phân định được sử truy cập tài nguyên hợp pháp và bất hợp pháp.
B. Quản lý không gian trống trên đĩa
C. Quyết định sẽ nạp chương trình nào khi có vùng nhớ trống.
D. Tạm dùng/ thực thi tiếp tiến trình.

4. Khi nào cần sự dụng đến System call (lời gọi hệ thống)?

A. Khi một người dùng yêu cầu dịch vụ nào đó từ Kernel của Hệ Điều Hành.
B. Khi một chương trình yêu cầu dịch vụ nào đó từ Kernel của Hệ Điều Hành.
C. Khi một Hệ điều hành cần sự trợ giúp từ chương trình.
D. Khi Hệ điều hành cần sự trợ giúp từ người dung.

CHƯƠNG 2: CẤU TRÚC HỆ ĐIỀU HÀNH

Trắc nghiệm khách quan

1. Cho biết tên gọi của kiến trúc Hệ điều hành mà các Modules chức năng của nó được phân chia thành từng lớp giao tiếp với Kernel.

A. Simple OS.
B. Monolithic OS.
C. Layered OS.
D. Microkernel OS.

2. Cho biết tên gọi của kiến 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. Simple OS.
B. Monolithic OS.
C. Layered OS.
D. Microkernel OS.

3. Cho biết tên gọi của kiến trúc Hệ Điều Hành mà tất cả các modules chức năng của nó được gom hết vào Kernel

A. Simple OS.
B. Monolithic OS.
C. Layered OS.
D. Microkernel OS.

CÁC CHƯƠNG

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

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.

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.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. 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.

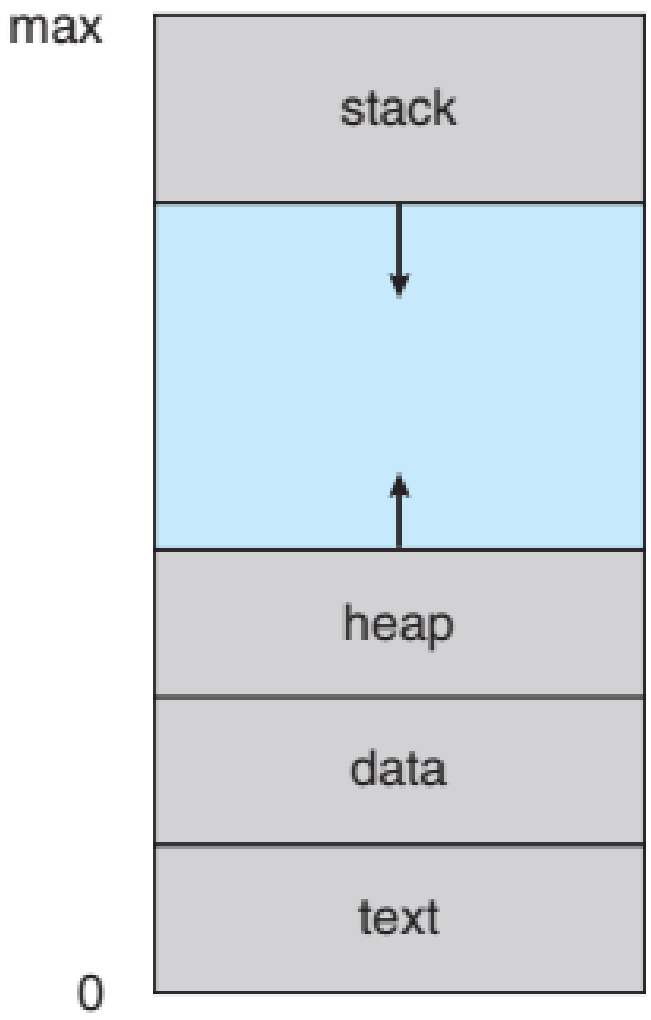
- Chương trình trở thành tiến trình khi một tập tin thực thi được nạp vào bộ nhớ.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.

Một tiến trình bao gồm:

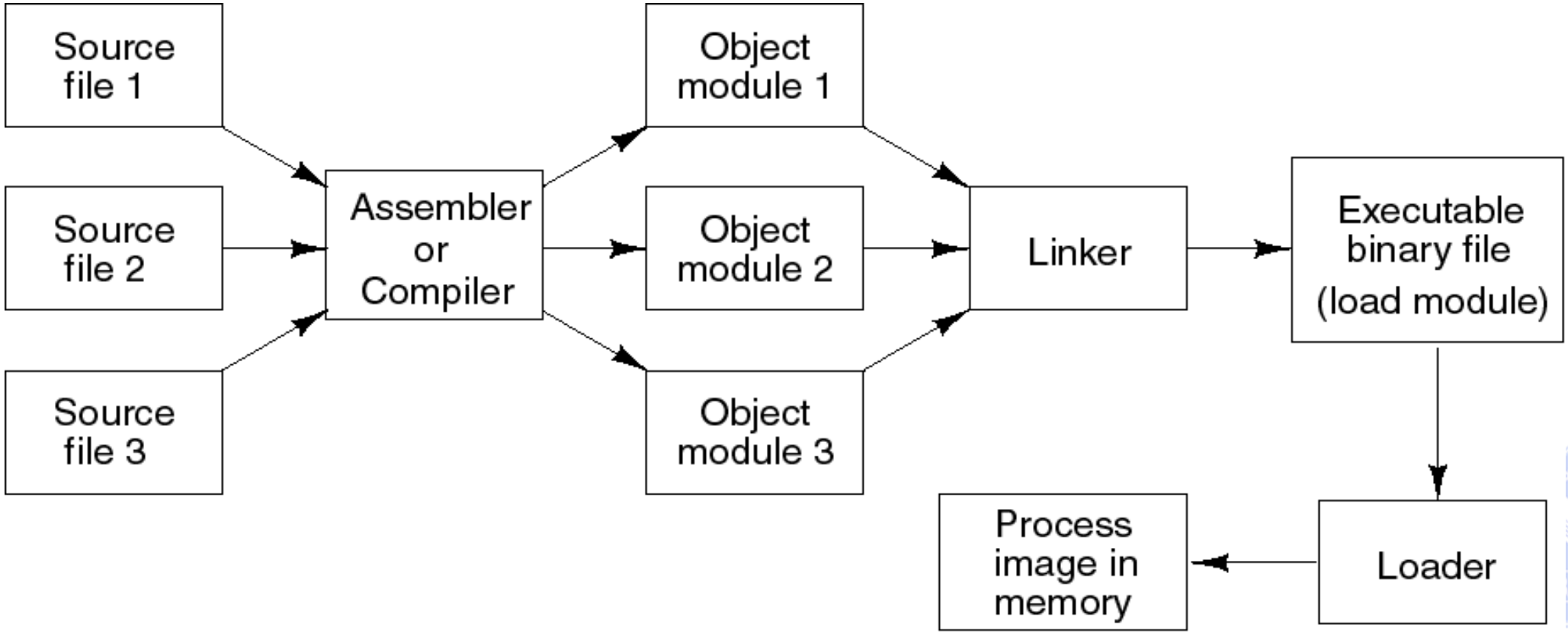
- Text section (program code)
- Data section (chứa biến toàn cục)
- Program counter, processor registers
- Heap section (chứa bộ nhớ cấp phát động)
- Stack section (chứa dữ liệu tạm thời)
  - Function parameters
  - Return address
  - Local variables



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. Các khái niệm cơ bản.

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



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

1. 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.

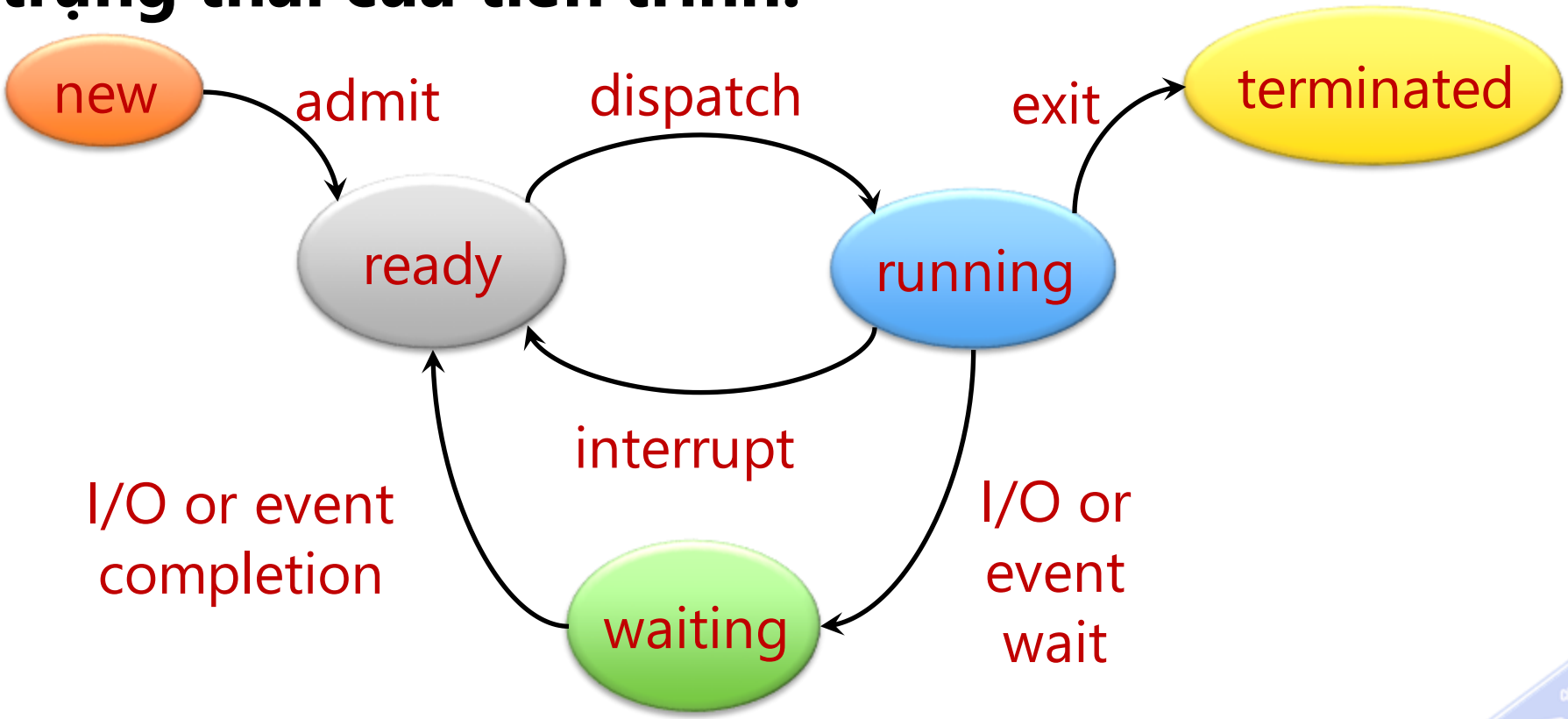
CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

2. 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

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

2. Các trạng thái của tiến trình.



Chuyển đổi giữa các trạng thái của tiến trình

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

2. 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 – terminated

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

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.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

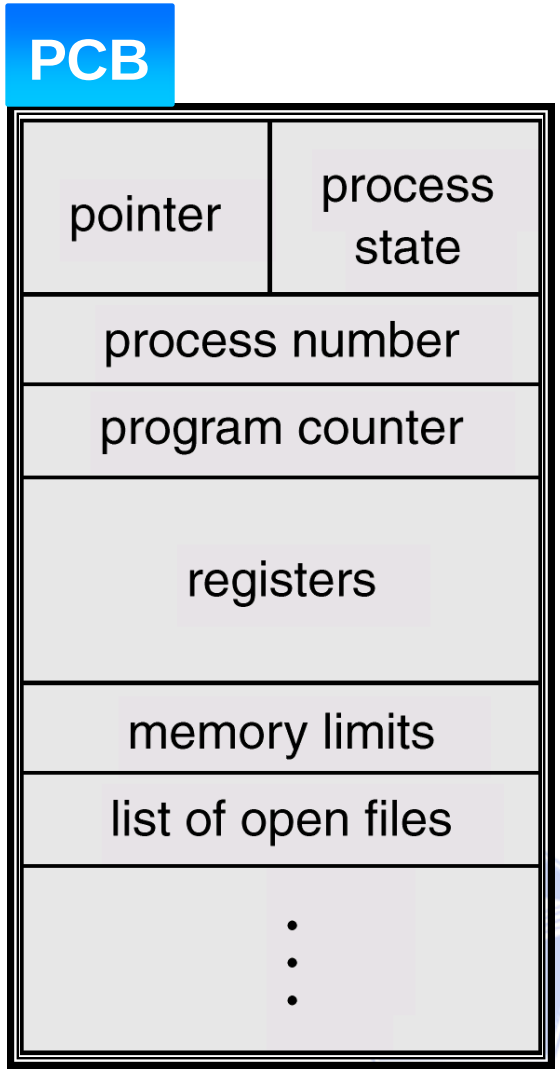
3. Process Control Block

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



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.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

4. Định thời tiến trình

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

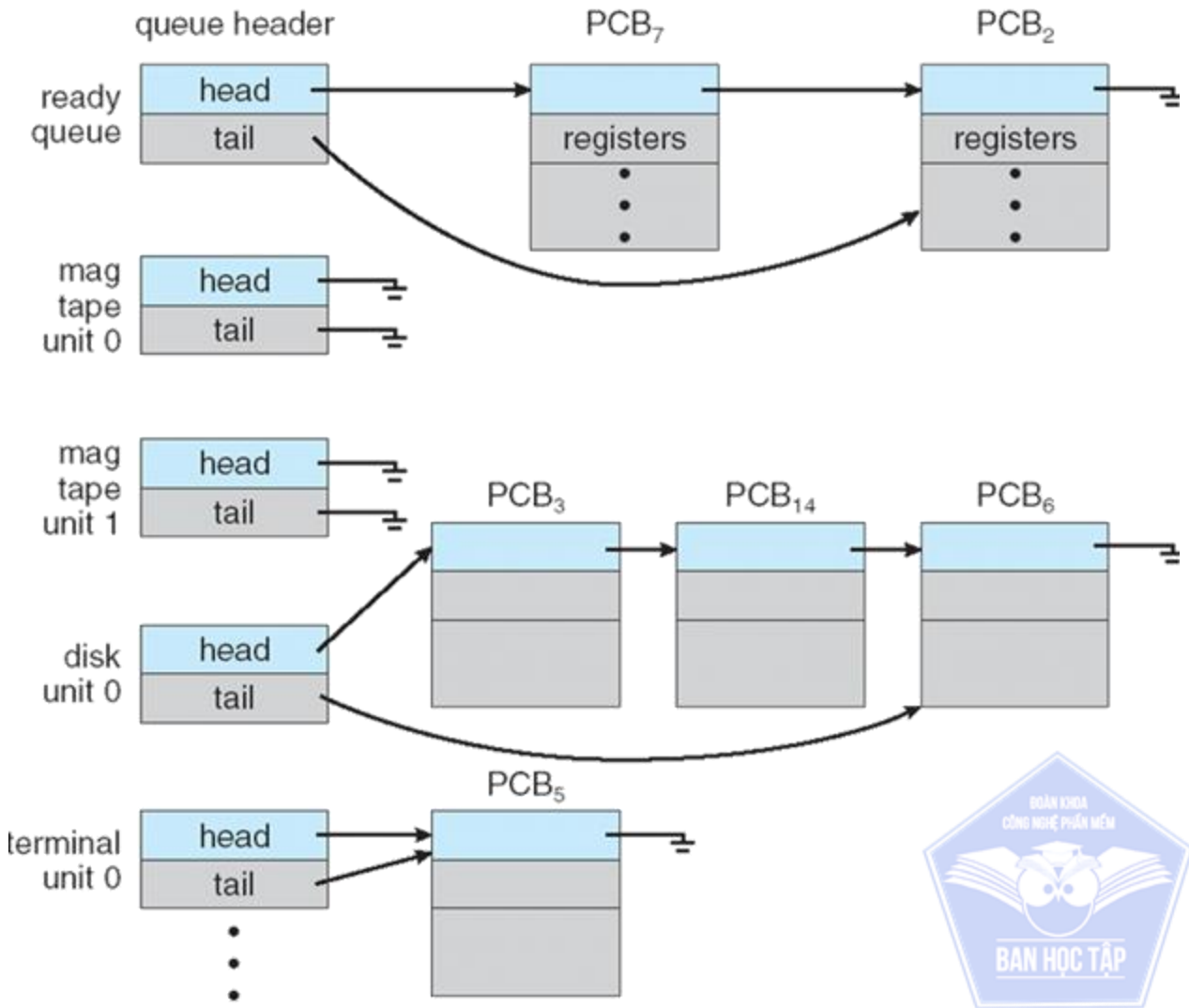
- Hỗ trợ sự thực thi luân phiên giữa nhiều tiến trình
  - ❖ Hiệu suất sử dụng CPU: cực đại
  - ❖ Thời gian đáp ứng: cực tiểu
- Phân phối tài nguyên hệ thống hợp lý
- Tránh deadlock, trì hoãn vô hạn định
- Cung cấp cơ chế giao tiếp và đồng bộ hoạt động các tiến trình
- Cung cấp cơ chế hỗ trợ user tạo/kết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

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

4. 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
- …

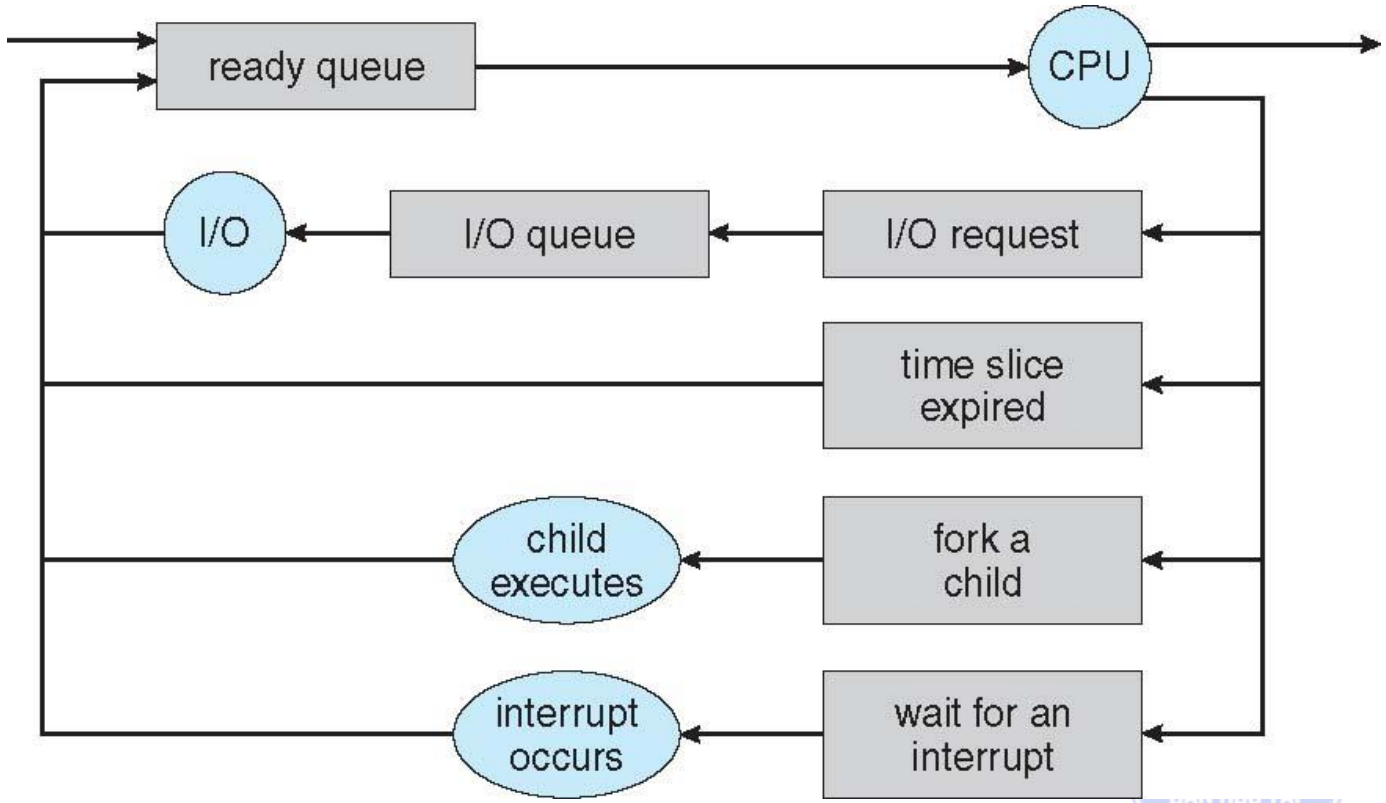


CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

4. 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

4. 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

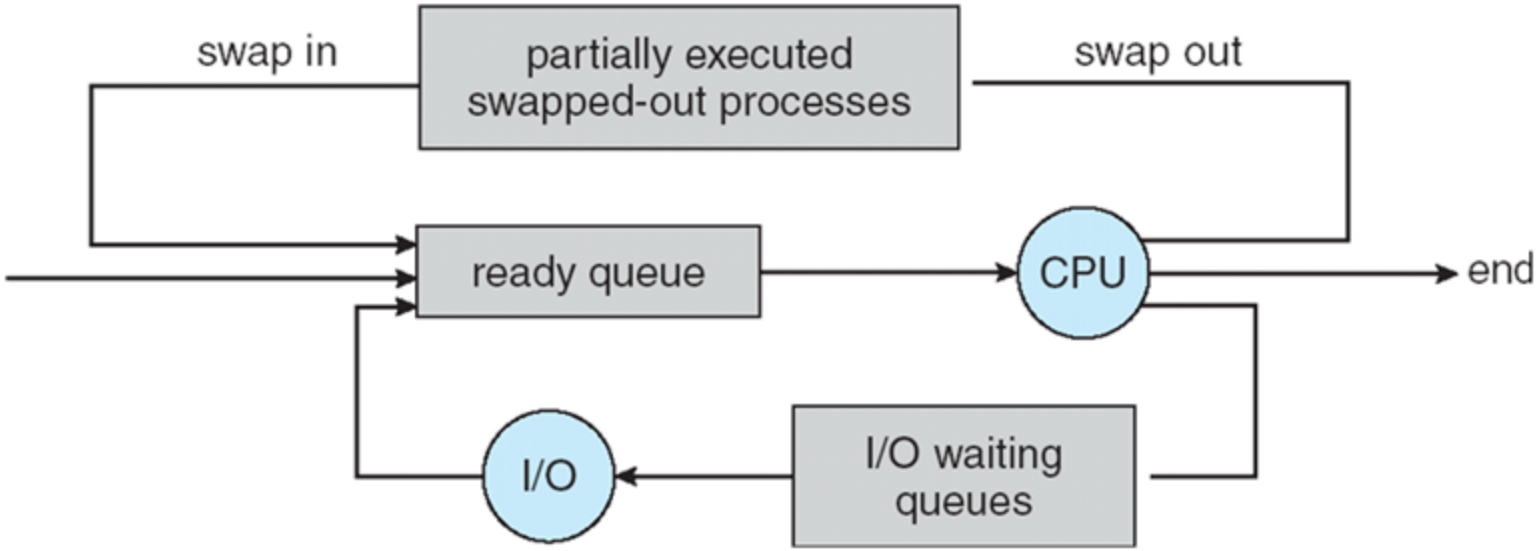
CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

4. 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)

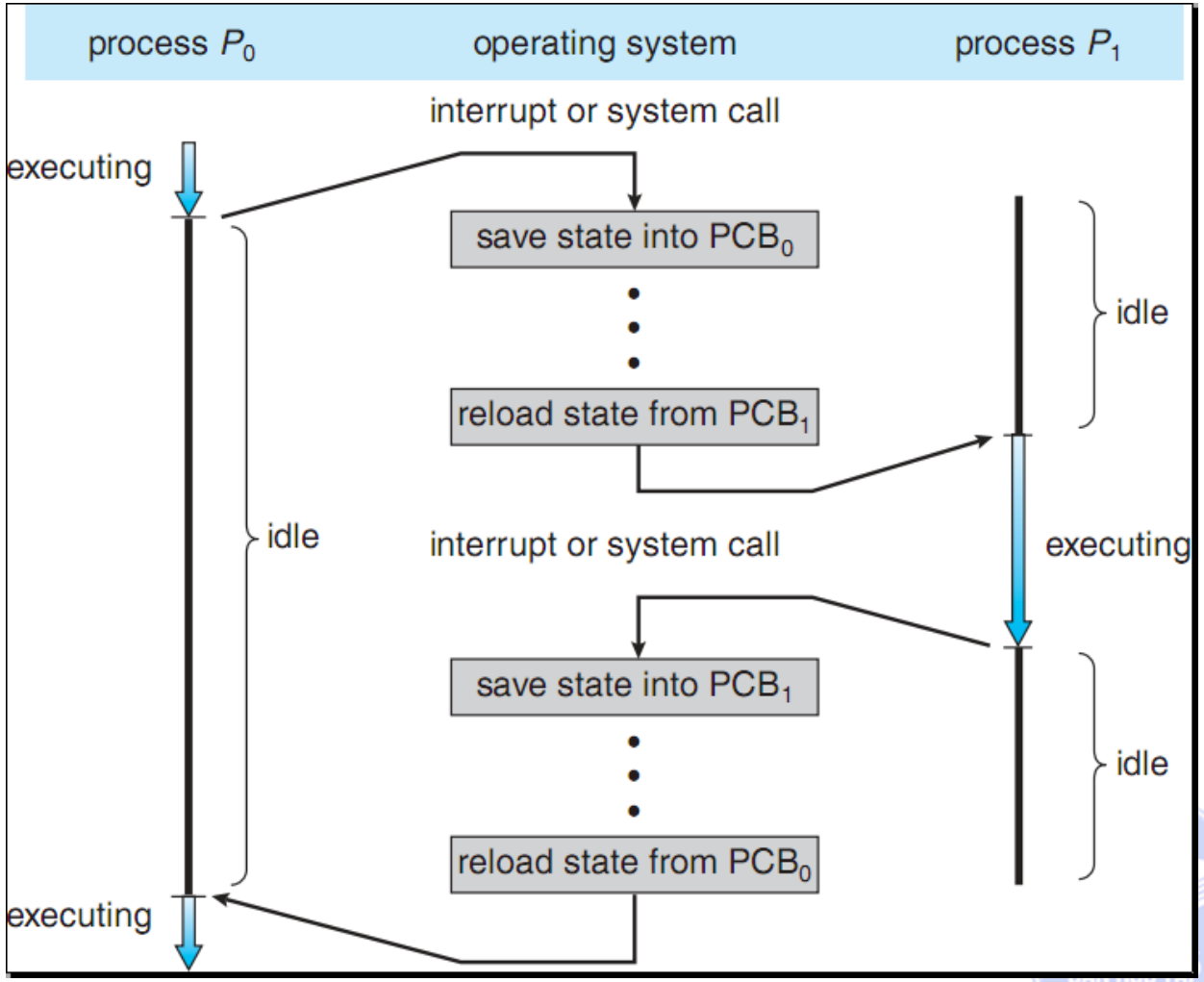


CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

4. 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.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. Các tác vụ đối với tiến trình

Có 2 tác vụ chính:

- Tạo tiến trình mới
- Kết thúc tiến trình

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. 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

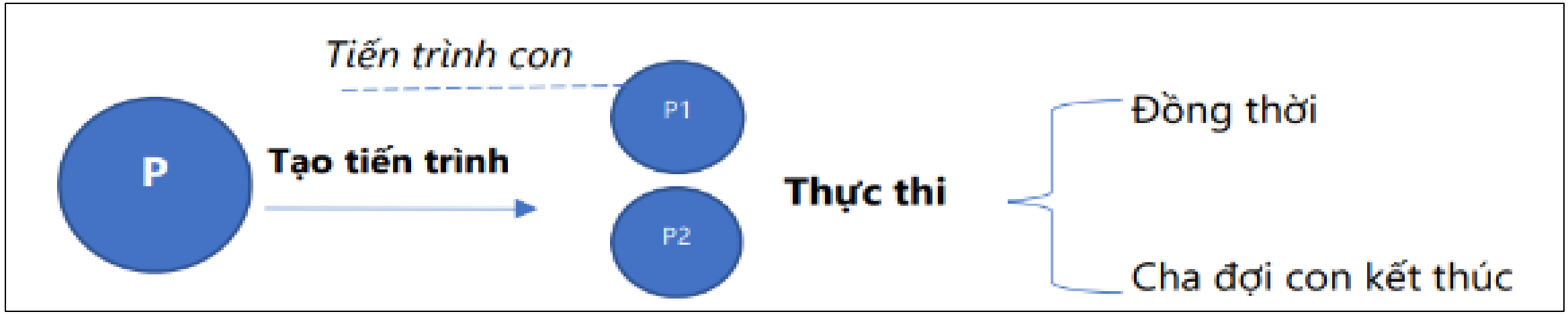
❖ **Trình thực thi**

- 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

5. Các tác vụ đối với tiến trình.

- Tạo tiến trình mới:



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. 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,..).

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. 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

$\Longrightarrow$ 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.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. 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

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. Các tác vụ đối với tiến trình

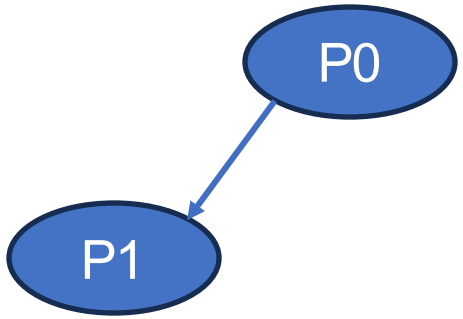
Ví dụ:

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

⟹ Cho biết output?

Hướng dẫn làm

Cây tiến trình:



OUTPUT: BHT CNPMBHT CNPM

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. Các tác vụ đối với tiến trình

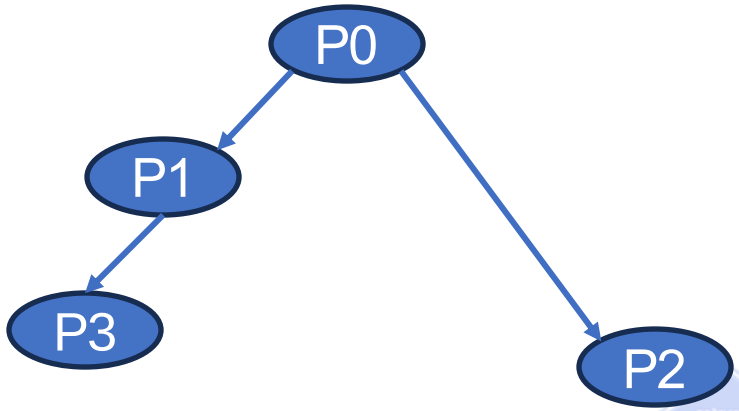
Ví dụ:

```
#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

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

5. Các tác vụ đối với tiến trình

Ví dụ:

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

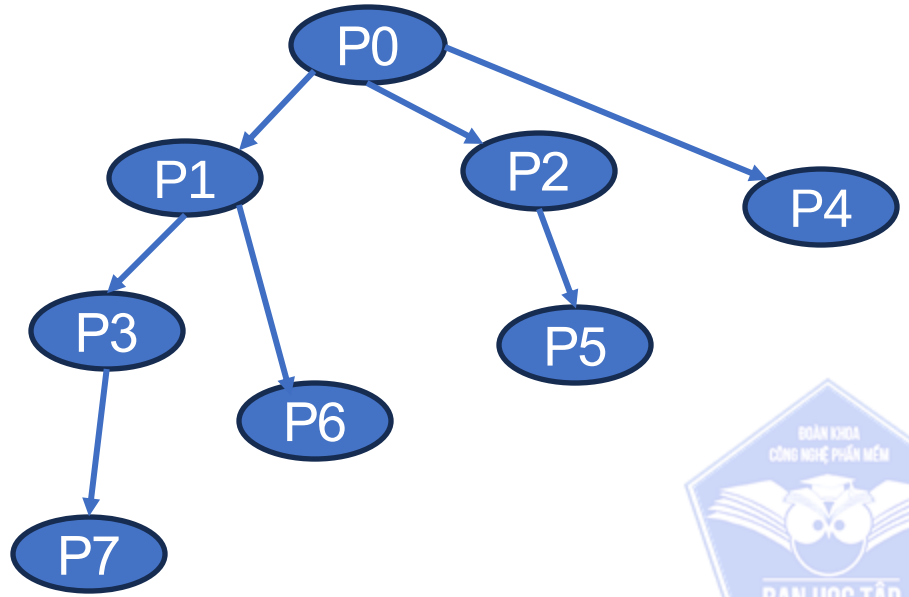
⟹ Cho biết có bao nhiêu chữ Hi, Hello, Alo?

A. 1,4,8 B.1,4,6

C. 1,2,9 D. 1,4,9

Hướng dẫn làm

Cây tiến trình:



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.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

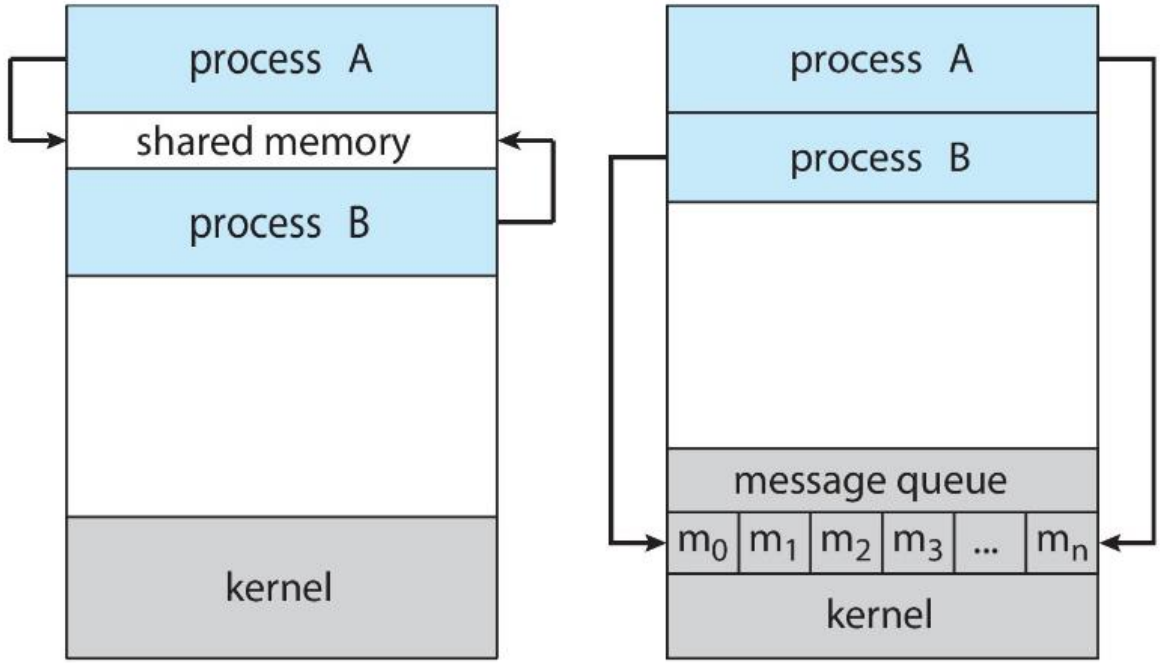
6. Giao tiếp liên tiến trình.

- 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

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

6. Giao tiếp liên tiến trình.

(a) Shared memory. (b) Message passing.



CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

6. 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.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

6. 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

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

6. Giao tiếp liên tiến trình.

Message Passing

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

➢ Tạo vùng đệm (Buffering): dùng queue để tạm chứa các message.

- Khả năng chứa là 0 (Zero capacity hay no buffering).
- Bounded capacity: độ dài của queue là giới hạn.
- Unbounded capacity: độ dài của queue là không giới hạn.

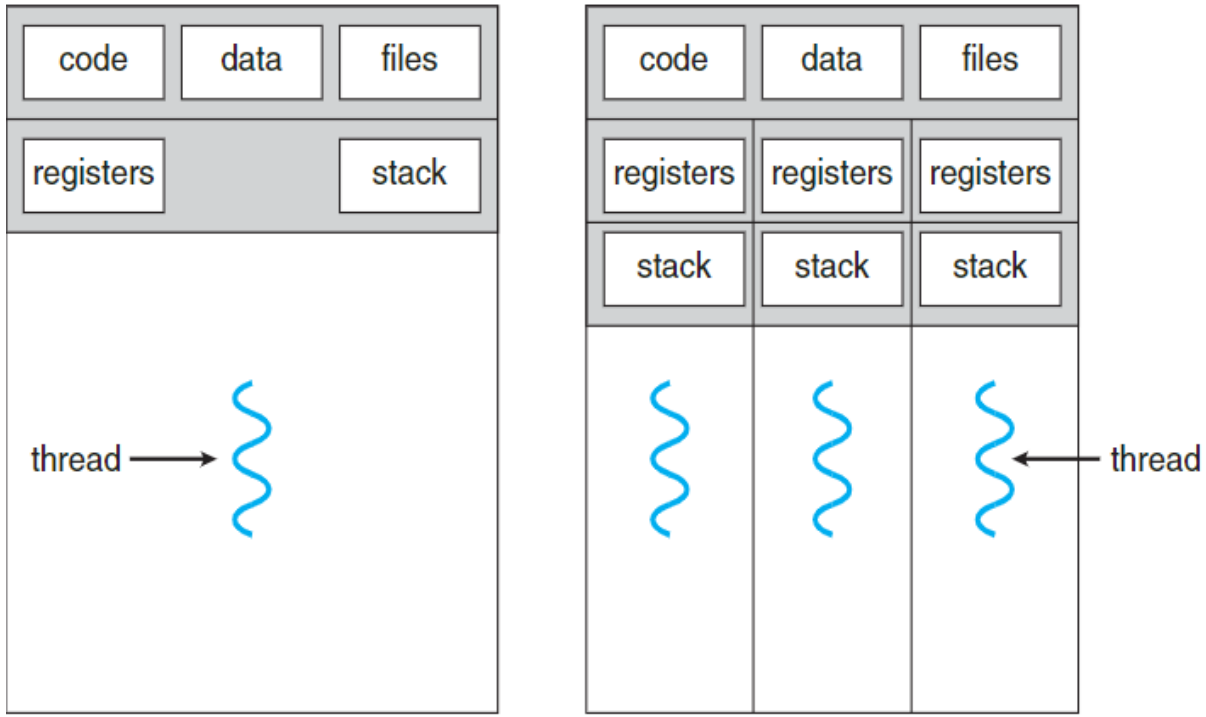
CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

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.**

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

7. 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,resourses.



single-threaded process

multithreaded process

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

7. 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.

Ví dụ: Trong Solaris 2, tạo process chậm hơn 30 lần, chuyển chậm hơn 5 lần so với thread.

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

7. Tiểu trình.

Các mô hình đa tiểu trình:

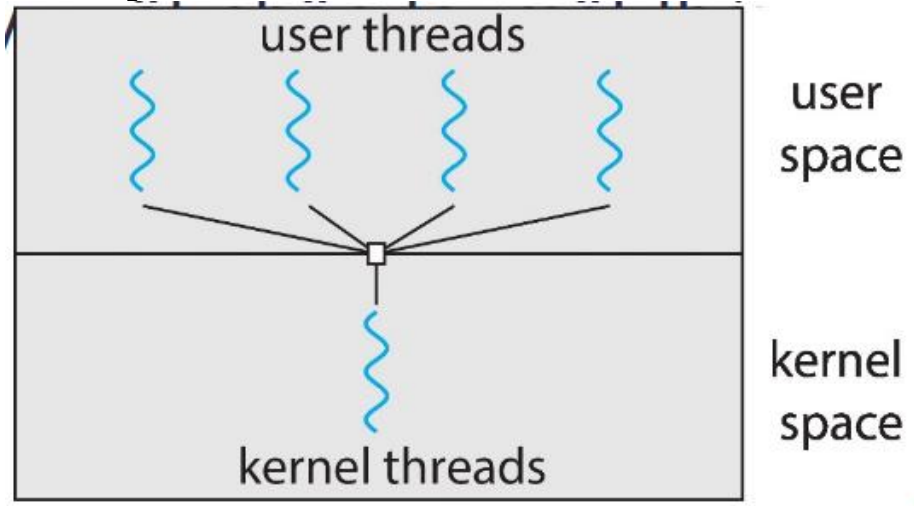
- Nhiều – Một (Many-to-One)
- Một – Một (One-to-One)
- Nhiều – Nhiều (Many-to-Many)

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

7. 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.

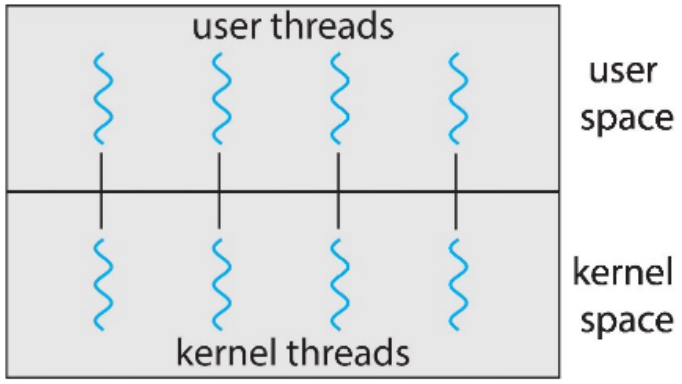


CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

7. 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.



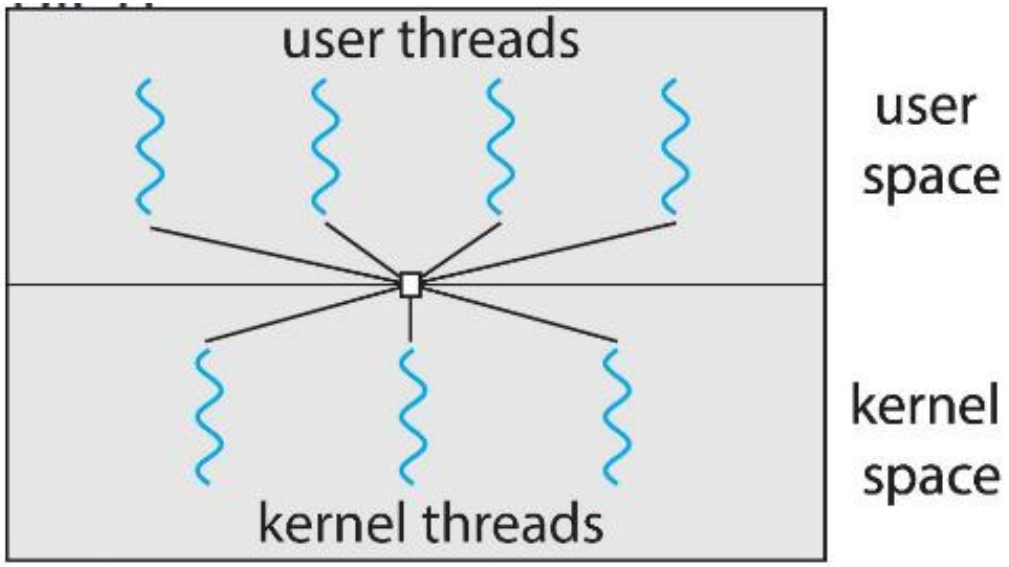
CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

7. 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 ⟹ Giải quyết được hạn chế của 2 mô hình trên.
- Khó cài đặt nên ít phổ biến.



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

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

CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

Câu 3: Cho đoạn mã nguồn sau:

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

Cho biết OUTPUT khi chương trình chạy:

CÁC CHƯƠNG

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

CHƯƠNG 4: ĐỊNH THỜI CPU

1. Các loại bộ định thời.
2. Các tiêu chuẩn định thời.
3. Các yếu tố của giải thuật định thời.
4. Các giải thuật định thời.

CHƯƠNG 4: ĐỊNH THỜI CPU

1. Các loại bộ định thời.

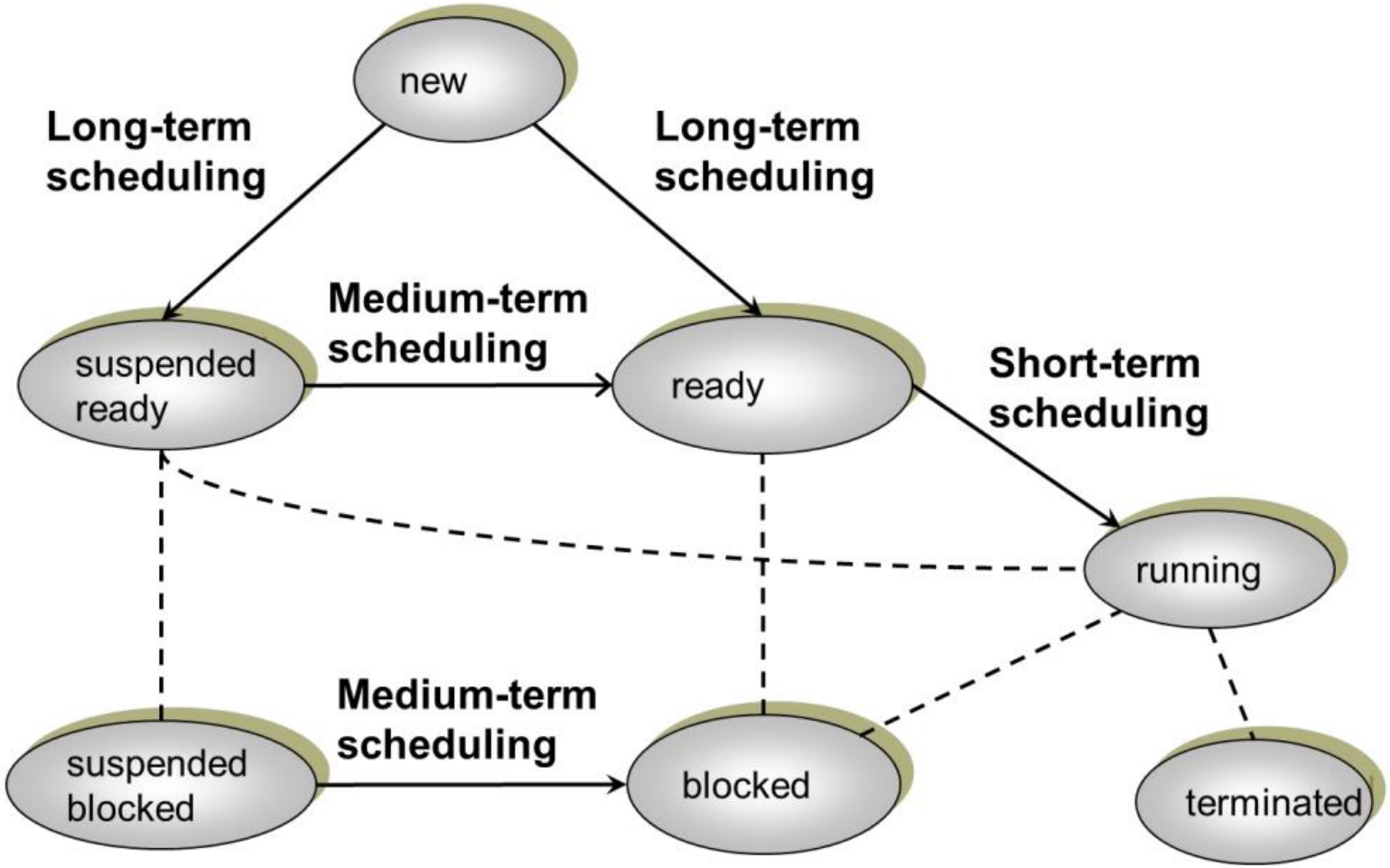
2. Các tiêu chuẩn định thời.

3. Các yếu tố của giải thuật định thời.

4. Các giải thuật định thời.

CHƯƠNG 4: ĐỊNH THỜI CPU

1. Các loại bộ định thời



CHƯƠNG 4: ĐỊNH THỜI CPU

1. Các loại bộ định thời

Long-term scheduling

- Xác định chương trình nào được chấp nhận nạp vào hệ thống để thực thi
- Điều khiển mức độ multiprogramming của hệ thống.
- Long term scheduler thường cố gắng duy trì xen lẫn CPU-bound và I/O - bound process .

Medium-term scheduling

- Process nào được đưa vào (swap in), đưa ra khỏi (swap out) bộ nhớ chính.
- Được thực hiện bởi phần quản lý bộ nhớ và được thảo luận ở phần quản lý bộ nhớ.

CHƯƠNG 4: ĐỊNH THỜI CPU

1. Các loại bộ định thời

Short-term scheduling

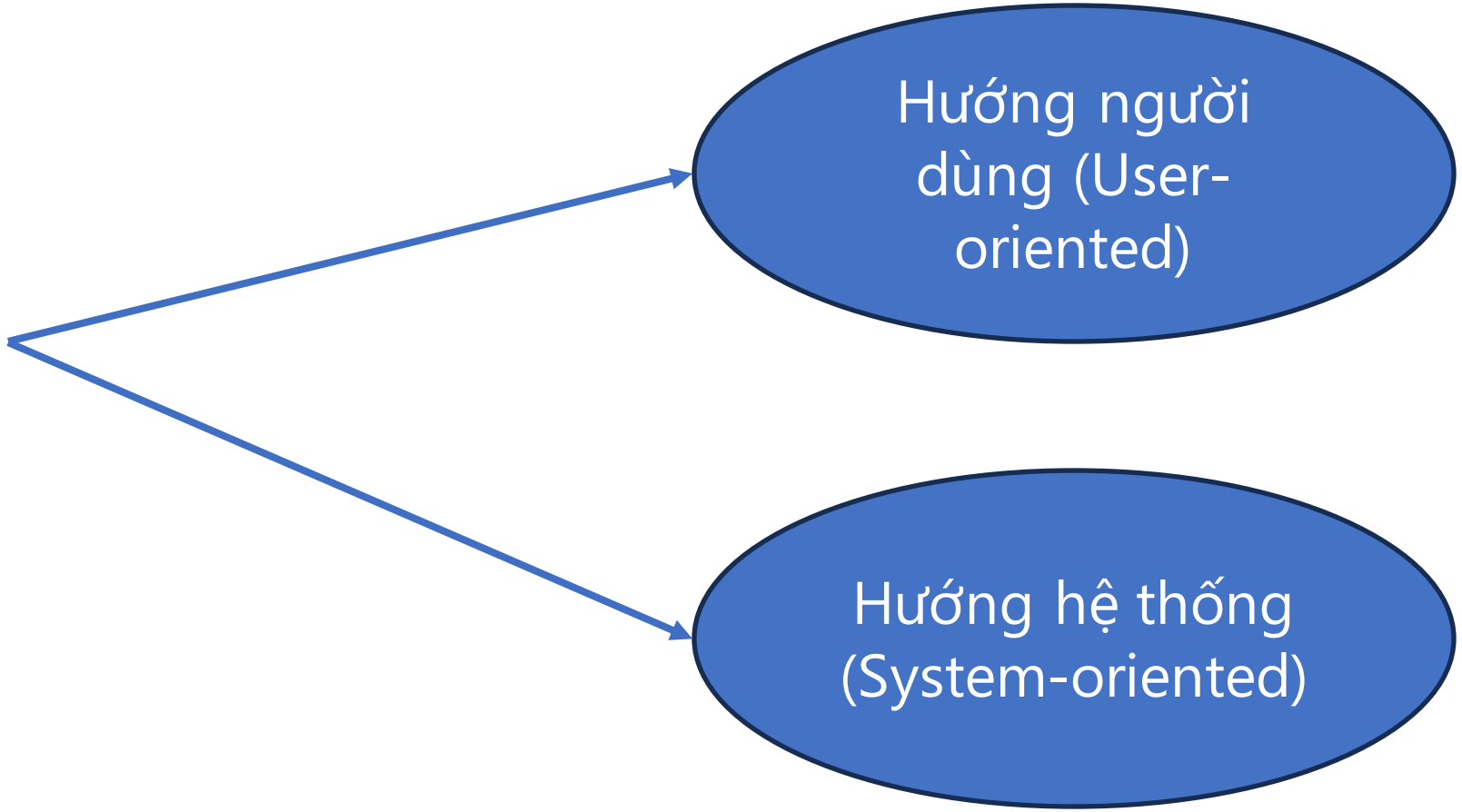
- Xác định process nào trong ready queue sẽ được chiếm CPU để thực thi kế tiếp (còn được gọi là định thời CPU, CPU scheduling).
- Bộ định thời short-term được gọi mỗi khi có một trong các sự kiện/interrupt sau xảy ra:
  - Ngắt thời gian (clock interrupt).
  - Ngắt ngoại vi (I/O interrupt).
  - Lời gọi hệ thống (operating system call).
  - Signal.

CHƯƠNG 4: ĐỊNH THỜI CPU

1. Các loại bộ định thời.
2. **Các tiêu chuẩn định thời.**
3. Các yếu tố của giải thuật định thời.
4. Các giải thuật định thời.

CHƯƠNG 4: ĐỊNH THỜI CPU

2. Các tiêu chuẩn định thời CPU:



CHƯƠNG 4: ĐỊNH THỜI CPU

2. Các tiêu chuẩn định thời CPU:

***Hướng người dùng*:**

- Thời gian đáp ứng: thời gian process nạp vào hệ thống -> yêu cầu đầu tiên được đáp ứng.
- Thời gian hoàn thành: thời gian process nạp vào hệ thống -> process kết thúc.
- Thời gian chờ: tổng thời gian một process đời trong ready queue.

$\Longrightarrow$ **CỰC TIỂU**

CHƯƠNG 4: ĐỊNH THỜI CPU

2. Các tiêu chuẩn định thời CPU:

***Hướng hệ thống*:**

- Sử dụng CPU: CPU càng bận càng tốt.
- Công bằng: các process phải được đối xử như nhau.
- Thông lượng: số process được hoàn tất trong một đơn vị thời gian là cực đại.

$\Longrightarrow$ **CỰC ĐẠI**

CHƯƠNG 4: ĐỊNH THỜI CPU

1. Các khái niệm cơ bản và các loại bộ định thời.
2. Các tiêu chuẩn định thời.
3. **Các yếu tố của giải thuật định thời.**
4. Các giải thuật định thời.
5. Định thời tiểu trình.
6. Định thời đa bộ xử lý.
7. Định thời trên các hệ điều hành hiện đại.

CHƯƠNG 4: ĐỊNH THỜI CPU

3. Các yếu tố của giải thuật định thời.

Hàm chọn lựa (selection function)

- Dùng để chọn process nào trong ready queue được thực thi(thường dựa trên độ ưu tiên,yêu cầu tài nguyên, đặc điểm thực thi của process,..)

CHƯƠNG 4: ĐỊNH THỜI CPU

3. Các yếu tố của giải thuật định thời.

Chế độ quyết định (decision mode)

- Chọn thời điểm thực hiện hàm chọn lựa để định thời.
- Hai chế độ:

➢ **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 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

1. Các khái niệm cơ bản và các loại bộ định thời.
2. Các tiêu chuẩn định thời.
3. Các yếu tố của giải thuật định thời.
4. **Các giải thuật định thời.**
5. Định thời tiểu trình.
6. Định thời đa bộ xử lý.
7. Định thời trên các hệ điều hành hiện đại.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve
b) Shortest-Job-First và Shortest-Remaining-Time-First
c) Priority Scheduling
d) Round Ronin
e) Highest-Response-Ratio-Next
f) Multilevel Queue và Multilevel Feedback Queue

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

Tóm tắt công thức

- Thời điểm tiến trình bắt đầu được thực thi là **Start_time**
- Thời điểm tiến trình kết thúc thực thi là **Finish_time**
- Thời điểm tiến trình xuất hiện là **Arrival_Time**
- Thời gian thực thi là **Burst_time**
- Thời gian đáp ứng là **Response_time**
- Thời gian chờ là **Waiting_time**
- Thời gian hoàn thành là **Turnaround_time**

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

Tóm tắt công thức

Start_time: tùy thuộc vào giải thuật

Finish_time = Start_time + Burst_time (chỉ đúng với cơ chế Nonpreemptive)

Response_time = Start_time - Arrival_time

Waiting_time = Response_time = Start_time - Arrival_time (chỉ đúng với cơ chế Nonpreemptive)

Turnaround_time = Finish_time - Arrival_time

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve

b) Shortest-Job-First và Shortest-Remaining-Time-First

c) Priority Scheduling

d) Round Ronin

e) Highest-Response-Ratio-Next

f) Multilevel Queue và Multilevel Feedback Queue

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve

- Tiến trình nào yêu cầu CPU trước sẽ được cấp phát CPU trước.
- Process sẽ thực thi đến khi kết thúc hoặc bị blocked do I/O.
- Chế độ quyết định: **non-preemptive.**
- Sử dụng **hàng đợi FIFO** (FIFO queues):
  - Tiến trình đi vào được **thêm vào cuối** hàng đợi.
  - Tiến trình được lựa chọn để xử lý được lấy từ đầu của queues.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve

Cho 5 tiến trình với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt

Thời gian đợi:

- P1=0, P2=8, P3=35, P4=37, P5=42
- Thời gian đợi trung bình: (0+8+35+37+42)/5=24.4

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt

Thời gian đáp ứng:

- P1=0, P2=8, P3=35, P4=37, P5=42
- Thời gian đáp ứng trung bình: (0+8+35+37+42)/5=24.4

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt

Thời gian hoàn thành:

- P1=10, P2=37, P3=38, P4=44, P5=54
- Thời gian đáp ứng trung bình: (10+37+38+44+54)/5=36.6

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve

b) Shortest-Job-First và Shortest-Remaining-Time-First

c) Priority Scheduling

d) Round Ronin

e) Highest-Response-Ratio-Next

f) Multilevel Queue và Multilevel Feedback Queue

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b) Shortest-Job-First và Shortest-Remaining-Time-First

- Thời gian sử dụng CPU chính là độ dài của CPU burst (burst time).
- Tương ứng với mỗi process cần có độ dài của CPU burst tiếp theo.
- Hàm lựa chọn: chọn process có độ dài CPU burst nhỏ nhất.
- SJF tối ưu trong việc giảm thời gian đợi trung bình.
- **Nhược điểm:** Cần phải ước lượng thời gian cần CPU tiếp theo của process.
- Có thể xảy ra tình trạng "đói" (starvation) đối với các process có CPU-burst lớn khi có nhiều process với CPU-burst nhỏ đến hệ thống.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.1) Shortest-Job-First

- Định thời biểu công việc ngắn nhất trước.

- Khi CPU được tự do, nó sẽ cấp phát cho tiến trình yêu cầu ít thời gian nhất để kết thúc (tiến trình ngắn nhất).

- Liên quan đến chiều dài thời gian sử dụng CPU cho lần tiếp theo của mỗi tiến trình.

- Sử dụng những chiều dài này để lập lịch cho tiến trình với thời gian ngắn nhất.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.1) Shortest-Job-First

- Non-preemptive

Khi CPU được trao cho quá trình nó không nhường cho đến khi nó kết thúc chu kỳ xử lý của nó.

- Preemptive

Nếu một tiến trình mới được đưa vào danh sách với chiều dài sử dụng CPU cho lần tiếp theo nhỏ hơn thời gian còn lại của tiến trình đang xử lý, nó sẽ dừng hoạt động tiến trình hiện hành.

→ Shortest-Remaining-Time-First (SRTF).

- SJF là tối ưu – cho thời gian chờ đợi trung bình tối thiểu với một tập tiến trình cho trước.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.1) Shortest-Job-First

Cho 5 tiến trình với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình.

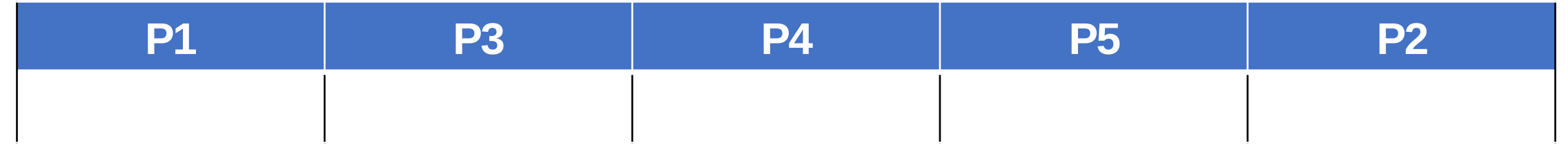
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.1) Shortest-Job-First

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



0

10
P2 = 29
P3 = 3
P4 = 7
P5 = 12

13
P2 = 29
P4 = 7
P5 = 12

20
P2 = 29
P5 = 12

32
P2 = 29

61

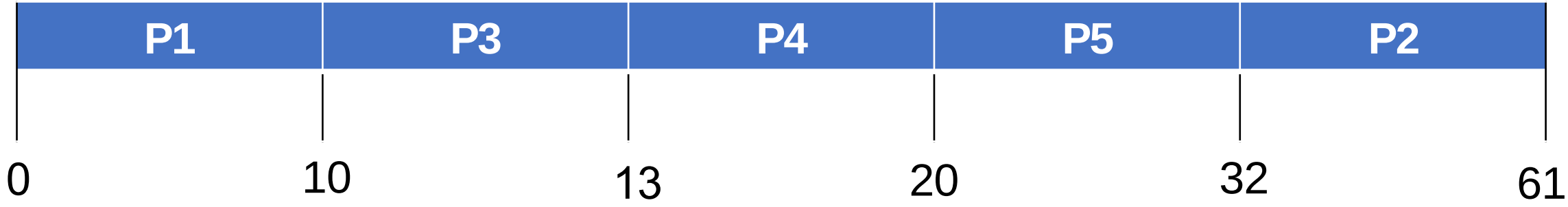
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.1) Shortest-Job-First

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



Thời gian đợi:

- P1=0, P2=30, P3=6, P4=8, P5=13
- Thời gian đợi trung bình: (0+30+5+8+13)/5=11.4

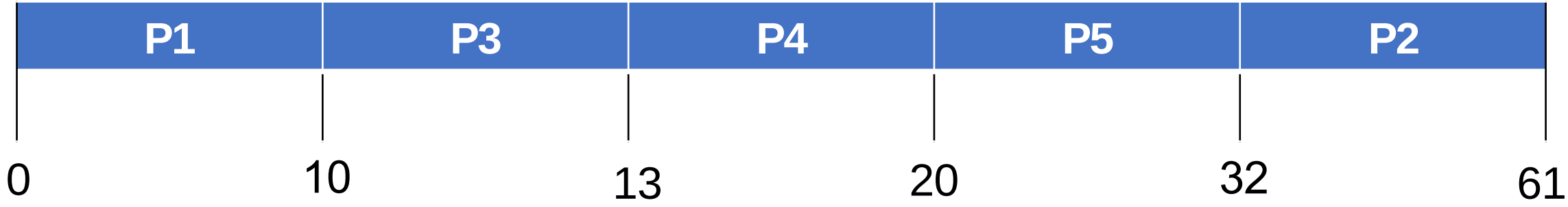
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.1) Shortest-Job-First

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



Thời gian đáp ứng:

- P1=0, P2=30, P3=6, P4=8, P5=13
- Thời gian đáp ứng trung bình: (0+30+5+8+13)/5=11.4

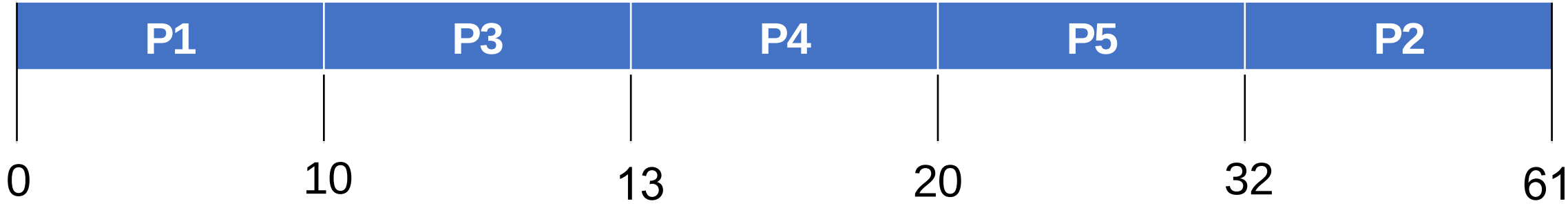
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.1) Shortest-Job-First

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



Thời gian hoàn thành:

- P1=10, P2=59, P3=9, P4=15, P5=25
- Thời gian đáp ứng trung bình: (10+59+9+15+25)/5=23.6

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.2) SRTF (Preemptive SJF)

Cho 5 tiến trình với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình.

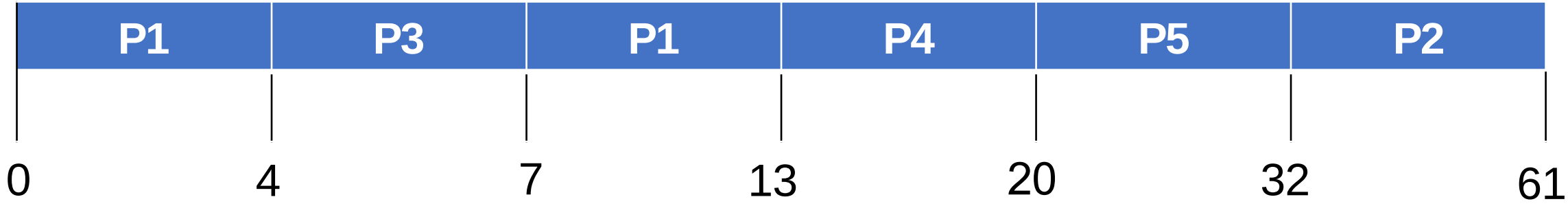
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.2) SRTF (Preemptive SJF)

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



P1 = 6
P2 = 29
P3 = 3

P1 = 6
P2 = 29
P4 = 7
P5 = 12

P2 = 29
P4 = 7
P5 = 12

P2 = 29
P5 = 12

P2 = 29

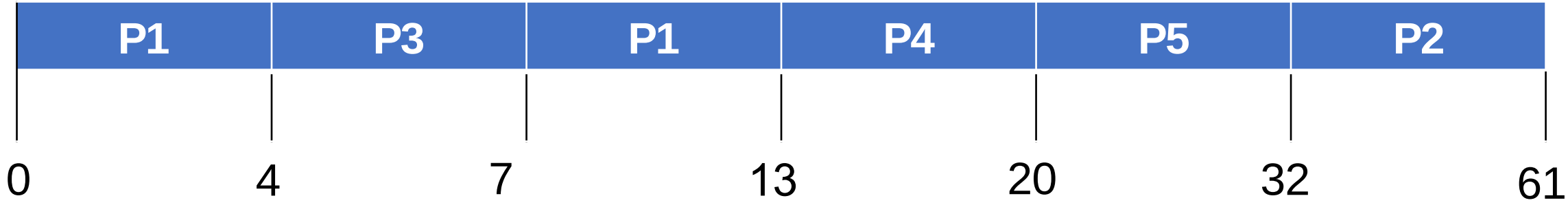
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.2) SRTF (Preemptive SJF)

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



Thời gian đợi:

- P1=3, P2=30, P3=0, P4=8, P5=13
- Thời gian đợi trung bình: (3+30+0+8+13)/5=10.8

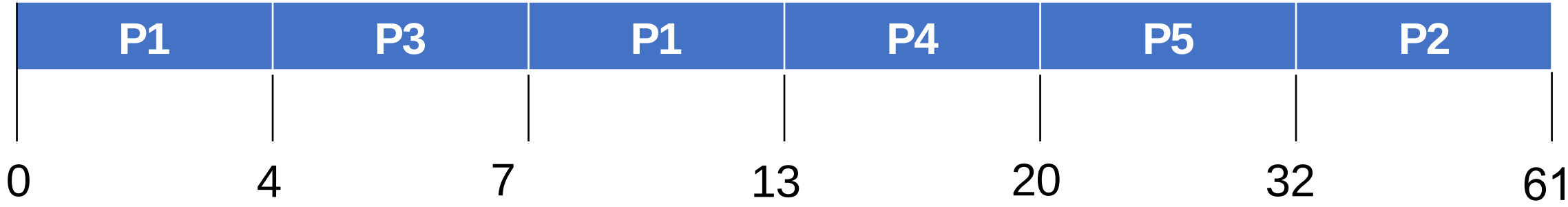
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.2) SRTF (Preemptive SJF)

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



Thời gian đáp ứng:

- P1=0, P2=30, P3=0, P4=8, P5=13
- Thời gian đáp ứng trung bình: (0+30+0+8+13)/5=10.2

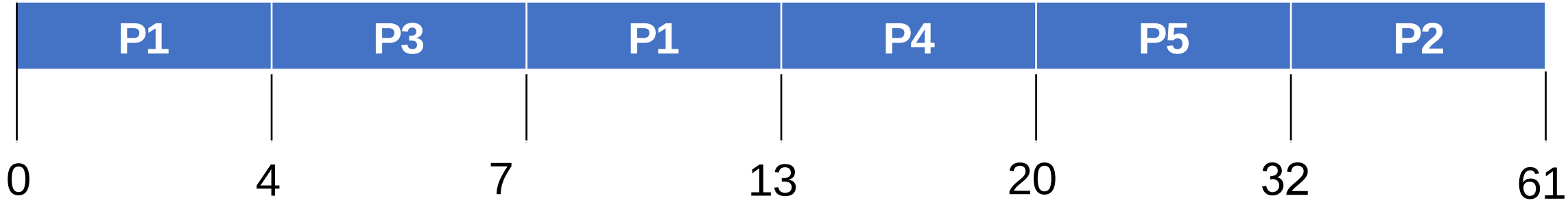
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

b.2) SRTF (Preemptive SJF)

Process	Arrival Time	Burst Time
P1	0	10
P2	2	29
P3	4	3
P4	5	7
P5	7	12

Giản đồ Gantt



Thời gian hoàn thành:

- P1=13, P2=59, P3=3, P4=15, P5=25
- Thời gian đáp ứng trung bình: (13+59+3+15+25)/5=23

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve
b) Shortest-Job-First và Shortest-Remaining-Time-First
c) Priority Scheduling
d) Round Ronin
e) Highest-Response-Ratio-Next
f) Multilevel Queue và Multilevel Feedback Queue

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c) 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.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.1) Priority Scheduling (non-preemptive)

Cho 5 tiến trình với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình.

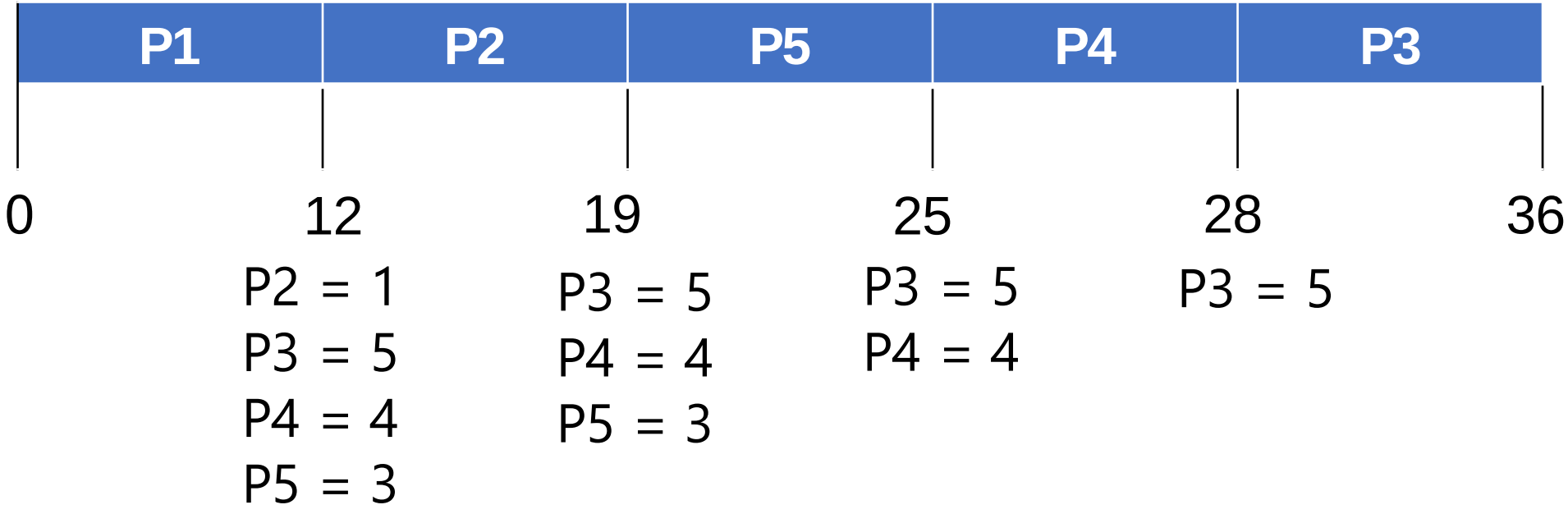
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.1) Priority Scheduling (non-preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3



CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.1) Priority Scheduling (non-preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3

Thời gian đợi:

- P1=0, P2=10, P3=23, P4=16, P5=7
- Thời gian đợi trung bình: (0+10+23+16+7)/5=11.2

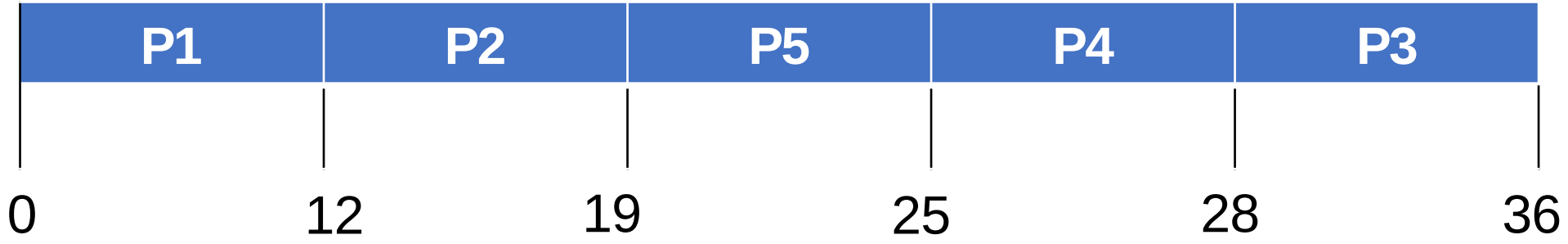
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.1) Priority Scheduling (non-preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3



Thời gian đáp ứng:

- P1=0, P2=10, P3=23, P4=16, P5=7
- Thời gian đáp ứng trung bình: (0+10+23+16+7)/5=11.2

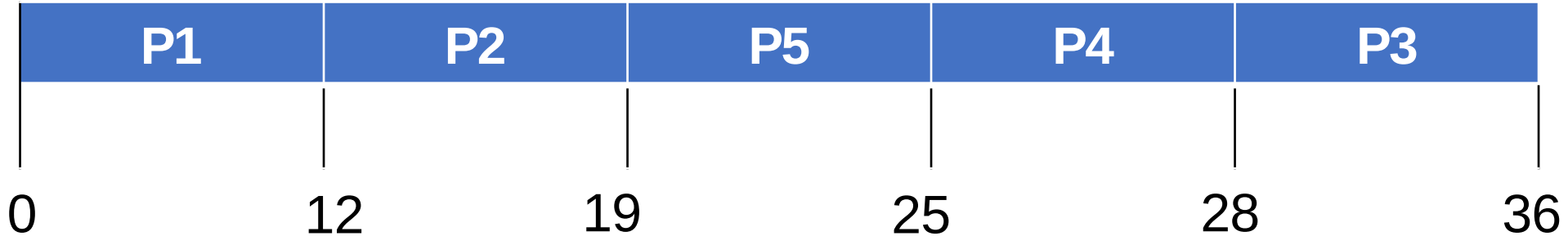
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.1) Priority Scheduling (non-preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3



Thời gian hoàn thành:

- P1=12, P2=7, P3=31, P4=19, P5=13
- Thời gian đáp ứng trung bình: (12+7+31+19+13)/5=16.4

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.2) Priority Scheduling (preemptive)

Cho 5 tiến trình với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình.

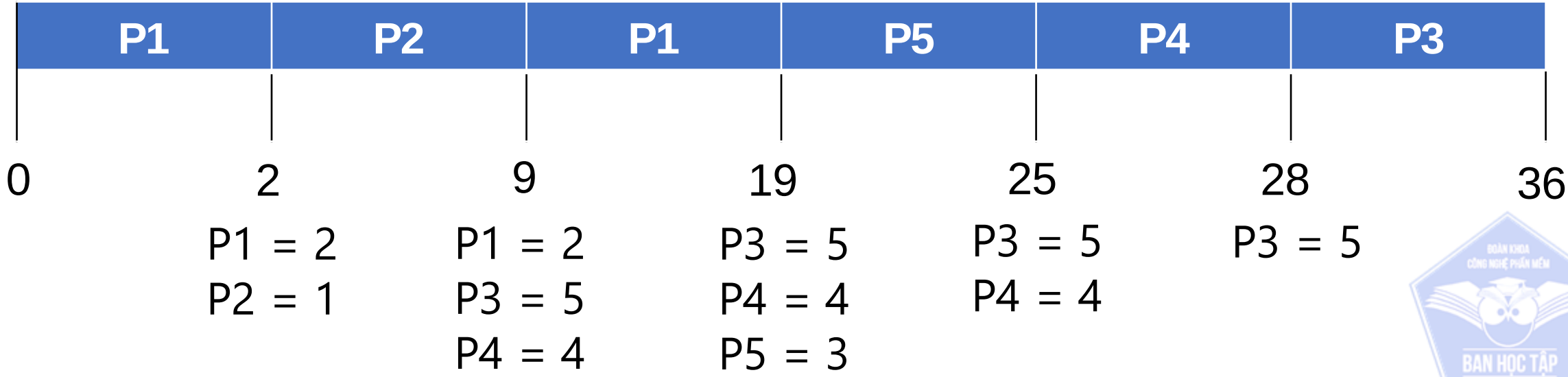
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.2) Priority Scheduling (preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3



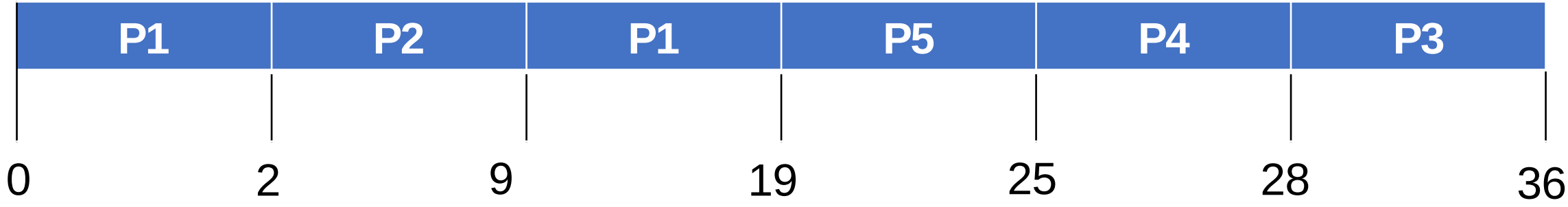
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.2) Priority Scheduling (preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3



Thời gian đợi:

- P1=7, P2=0, P3=23, P4=16, P5=7
- Thời gian đợi trung bình: (7+0+23+16+7)/5=10.6

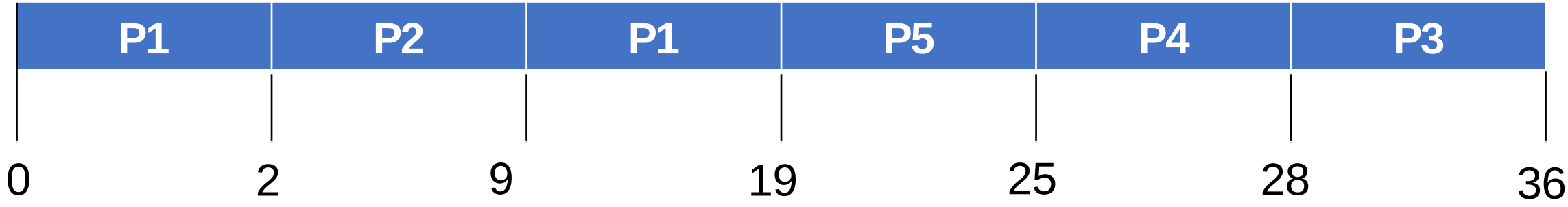
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.2) Priority Scheduling (preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3



Thời gian đáp ứng:

- P1=0, P2=0, P3=23, P4=16, P5=7
- Thời gian đáp ứng trung bình: (0+0+23+16+7)/5=9.2

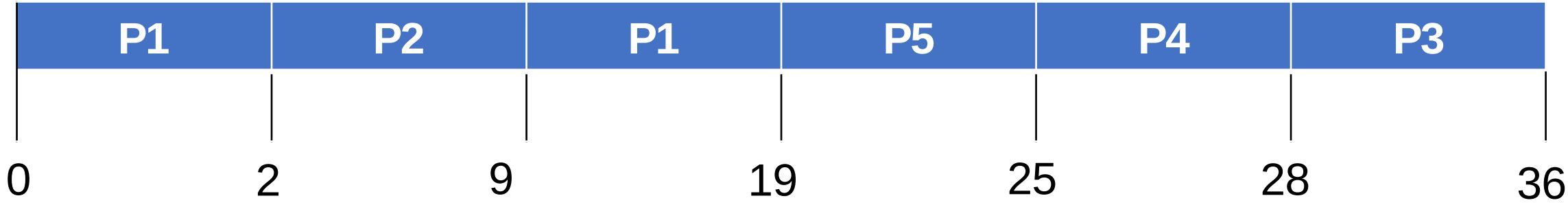
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c.2) Priority Scheduling (preemptive)

Giản đồ Gantt

Process	Arrival Time	Burst Time	Priority
P1	0	12	2
P2	2	7	1
P3	5	8	5
P4	9	3	4
P5	12	6	3



Thời gian hoàn thành:

- P1=19, P2=7, P3=31, P4=19, P5=13
- Thời gian đáp ứng trung bình: (19+7+31+19+13)/5=17.8

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve
b) Shortest-Job-First và Shortest-Remaining-Time-First
c) Priority Scheduling
d) Round Ronin
e) Highest-Response-Ratio-Next
f) Multilevel Queue và Multilevel Feedback Queue

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c) Round Ronin

- 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

4. Các giải thuật định thời

c) Round Ronin

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.

Thời gian chờ đợi trung bình của giải thuật RR thường khá lớn nhưng thời gian đáp ứng nhỏ.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c) Round Ronin

Cho 5 tiến trình với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau: **(quantum time = 4)**

Process	Arrival Time	Burst Time
P1	0	12
P2	2	7
P3	5	8
P4	9	3
P5	12	6

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình.

CHƯƠNG 4: ĐỊNH THỜI CPU

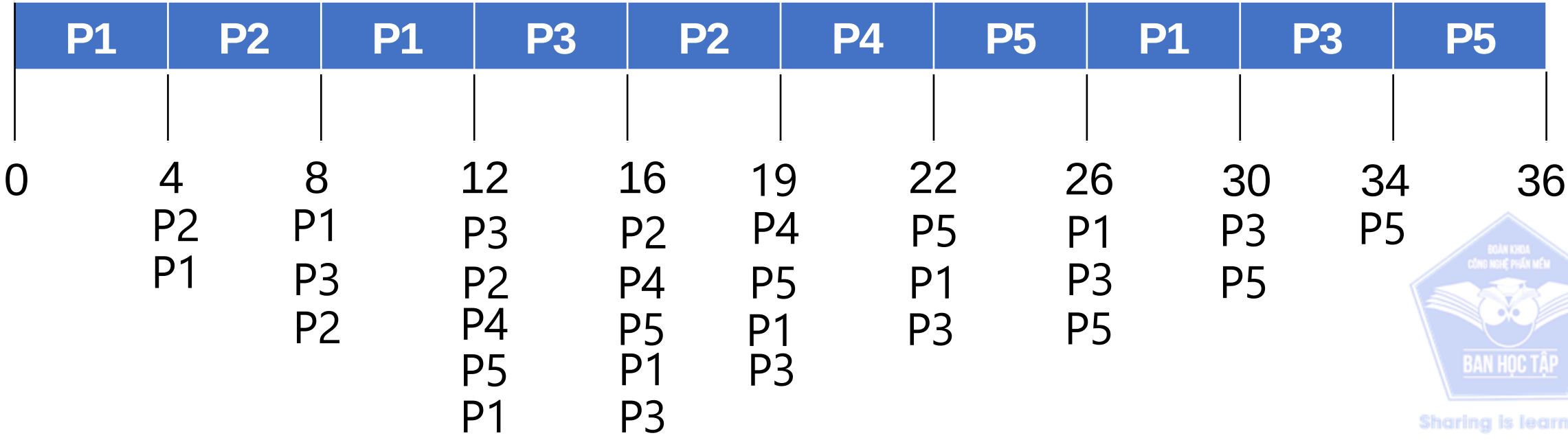
4. Các giải thuật định thời

c) Round Ronin

(quantum time = 4)

Giản đồ Gantt

Process	Arrival Time	Burst Time
P1	0	12
P2	2	7
P3	5	8
P4	9	3
P5	12	6



CHƯƠNG 4: ĐỊNH THỜI CPU

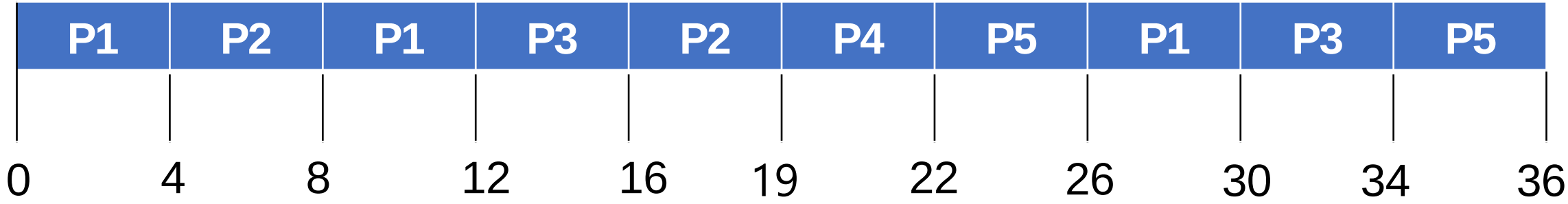
4. Các giải thuật định thời

c) Round Ronin

(quantum time = 4)

Giản đồ Gantt

Process	Arrival Time	Burst Time
P1	0	12
P2	2	7
P3	5	8
P4	9	3
P5	12	6



Thời gian đáp ứng:

- P1=0, P2=2, P3=7, P4=10, P5=10
- Thời gian đáp ứng trung bình: (0+2+7+10+10)/5=5.8

CHƯƠNG 4: ĐỊNH THỜI CPU

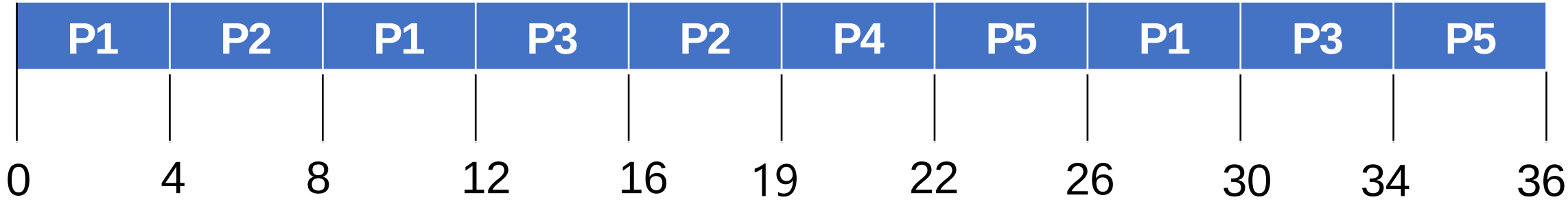
4. Các giải thuật định thời

c) Round Ronin

(quantum time = 4)

Giản đồ Gantt

Process	Arrival Time	Burst Time
P1	0	12
P2	2	7
P3	5	8
P4	9	3
P5	12	6



Thời gian đợi:

- P1=4+14=18, P2=2+8=10, P3=7+14=21, P4=10, P5=10+8=18
- Thời gian đáp ứng trung bình: (18+10+21+10+18)/5=15.4

CHƯƠNG 4: ĐỊNH THỜI CPU

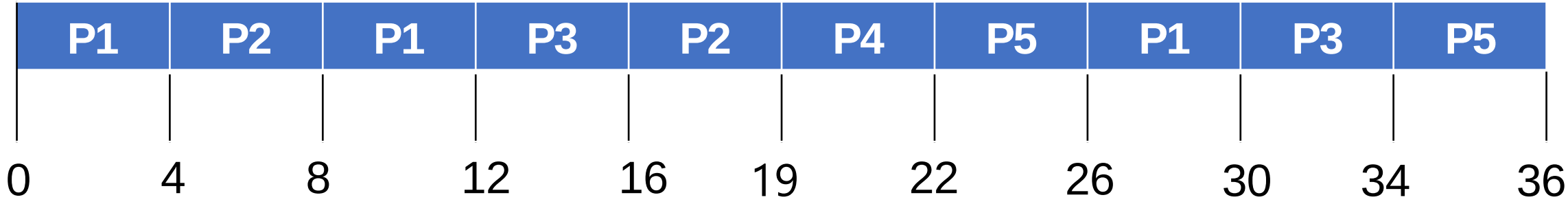
4. Các giải thuật định thời

c) Round Ronin

(quantum time = 4)

Giản đồ Gantt

Process	Arrival Time	Burst Time
P1	0	12
P2	2	7
P3	5	8
P4	9	3
P5	12	6



Thời gian hoàn thành:

- P1=30, P2=17, P3=29, P4=13, P5=24
- Thời gian đáp ứng trung bình: (30+17+29+13+24)/5=22.6

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

c) Round Ronin

- ❑ Khi thực hiện process switch thì OS sẽ sử dụng CPU chứ không phải process của người dùng.
  - Dừng thực thi, lưu tất cả thông tin, nạp thông tin của process sắp thực thi.
- ❑ 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
  - Nếu time slice quá lớn, RR trở thành FCFS.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

a) First-Come-First-Serve
b) Shortest-Job-First và Shortest-Remaining-Time-First
c) Priority Scheduling
d) Round Ronin
e) Highest-Response-Ratio-Next
f) Multilevel Queue và Multilevel Feedback Queue

CHƯƠNG 4: ĐỊNH THỜI CPU

4.Các giải thuật định thời

e) 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:

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

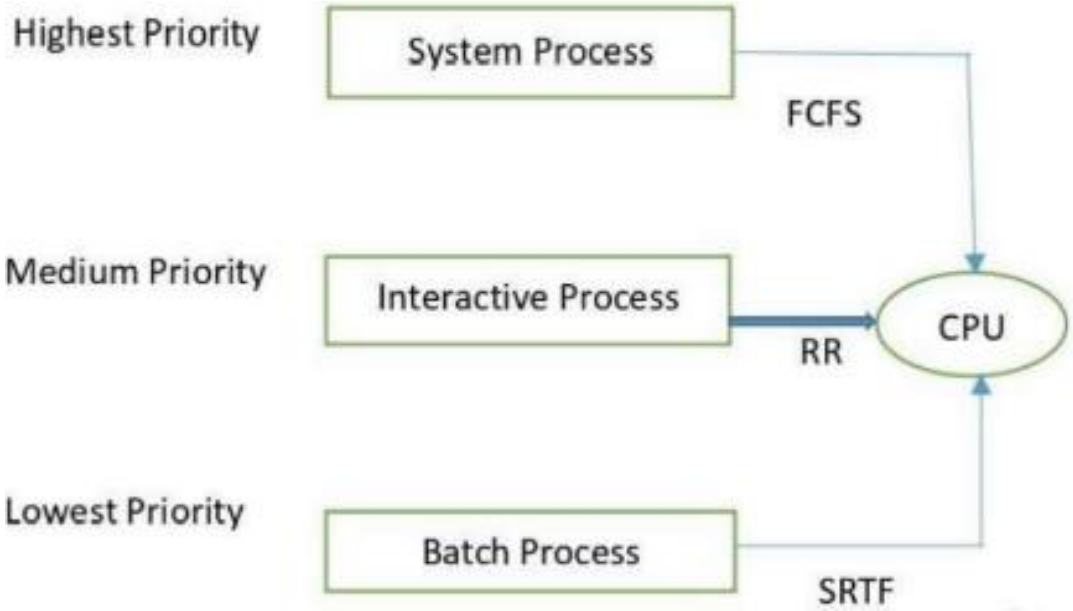
CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

f) Multilevel Queue và Multilevel Feedback Queue

Hàng đợi ready được chia thành các hàng đợi riêng biệt dựa vào các tiêu chuẩn như:

- Đặc điểm và yêu cầu định thời của process
- Foreground và background process,…

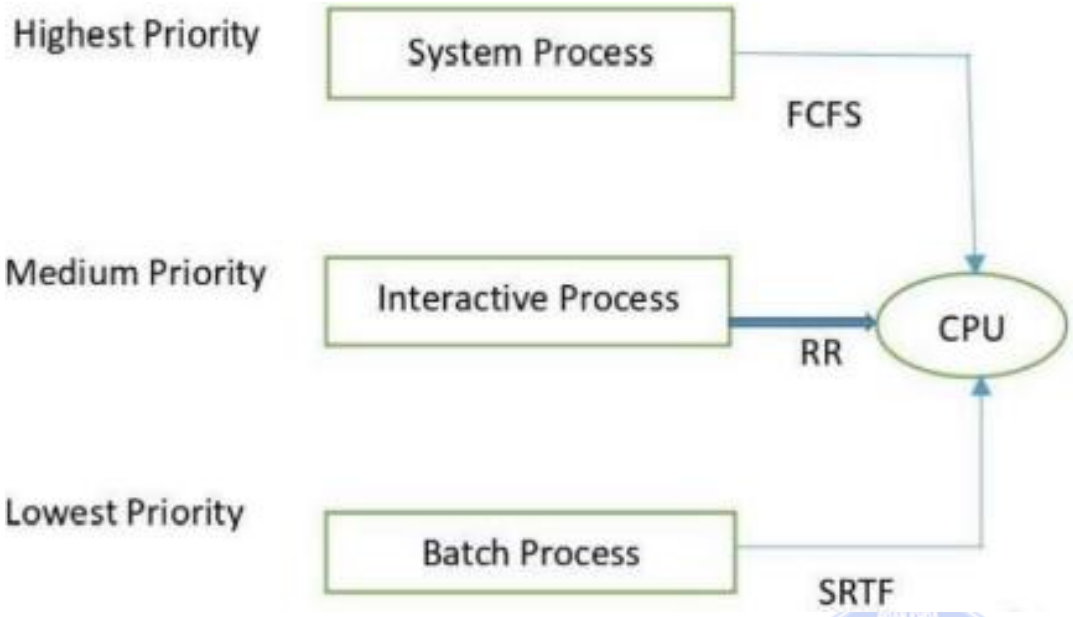


CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

f) Multilevel Queue và Multilevel Feedback Queue

- Process được gán cố định vào một hàng đợi, mỗi hàng đợi sử dụng giải thuật định thời riêng
- Quá trình được chạy ở chế độ giao tiếp(foreground hay interactive process) được ưu tiên hơn so với quá trình chạy nền (background process)



CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

f) Multilevel Queue và Multilevel Feedback 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.

CHƯƠNG 4: ĐỊNH THỜI CPU

4. Các giải thuật định thời

f) Multilevel Queue và Multilevel Feedback Queue

Vấn đề của Multilevel queue

Process không thể chuyển từ hàng đợi này sang hàng đợi khác

→ ***Giải pháp***: Cơ chế feedback

Cơ chế feedback

- Giải quyết được vấn đề của Multilevel queue
- Nếu một quá trình dùng quá nhiều thời gian của CPU thì nó bị di chuyển đến hàng đợi có độ ưu tiên thấp
- Ngược lại, khi chờ lâu trong hàng đợi có độ ưu tiên thấp quá lâu thì nó có thể được di chuyển sang hàng chờ có độ ưu tiên cao

CHƯƠNG 4: ĐỊNH THỜI CPU

4.Các giải thuật định thời

f) Multilevel Queue và 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 hang đợi cao hoặc thấp hơn ?
- Khi process yêu cầu được xử lý thì hàng đợi nào là hợp lý nhất ?

CHƯƠNG 4: ĐỊNH THỜI CPU

**Trắc nghiệm khách quan*

Câu 1: Chọn phát biểu **SAI** trong các phát biểu sau:

A. Trong giải thuật SJF có thể xảy ra tình trạng "đói" (starvation) đối với các tiến trình có CPU-burst nhỏ khi có nhiều tiến trình với CPU burst lớn đến hệ thống.

B. Trong giải thuật Multilevel Queue, hàng đợi ready được chia thành nhiều hàng đợi, mỗi hàng đợi sử dụng giải thuật định thời riêng.

C. Giải thuật Multilevel Feedback Queue cho phép các tiến trình di chuyển một cách thích hợp giữa các hàng đợi khác nhau.

D. Không thể sử dụng giải thuật Round Robin nếu muốn các tiến trình có độ ưu tiên khác nhau.

CHƯƠNG 4: ĐỊNH THỜI CPU

**Trắc nghiệm khách quan*

Câu 2: Loại hàng đợi nào được sử dụng khi hiện thực giải thuật định thời FCFS?

A. FIFO (First In, First Out)

B. LIFO (Last In, First Out)

C. FINO (First In, Never Out)

D. FILO (First In, Last Out)

BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2023 – 2024

ĐOÀN KHOA CÔNG NGHỆ PHẦN MỀM
BAN HỌC TẬP
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