

# BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

## TRAINING GIỮA KỲ HỌC KỲ II NĂM HỌC 2023 – 2024



**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](mailto:bht.cnpm.uit@gmail.com)  
[fb.com/bhtcnpm](https://fb.com/bhtcnpm)  
[fb.com/groups/bht.cnpm.uit](https://fb.com/groups/bht.cnpm.uit)

### TEAM TIẾNG ANH

[english.with.bht@gmail.com](mailto:english.with.bht@gmail.com)  
 [creative.owl.se](https://creative.owl.se)  
 [english.with.bht](https://english.with.bht)

**TRAINING**

# HỆ ĐIỀU HÀNH

- ⌚ Thời gian:** 19:30 thứ Hai ngày 13/4/2024
- 👉 Địa điểm:** Microsoft Teams – Code: w2dsy1q
- 👤 Trainers:** Trương Đoàn Vũ – MMTT 2022.3  
Huỳnh Lê Đan Linh – KTPM 2022.3



**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**



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**



Sharing is learning

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

**I. Tổng quan cơ bản**

**II. Phân loại hệ điều hành**



Sharing is learning

## **Câu 1: Mục Tiêu Chính Của Hệ Điều Hành Là Gì ?**

- A. Giúp người dùng dễ dàng sử dụng hệ thống
- B. Quản lý và cấp phát tài nguyên hệ thống một cách hiệu quả
- C. Giúp cho chúng ta xem phim và truy cập internet
- D. Đáp án A và B đúng

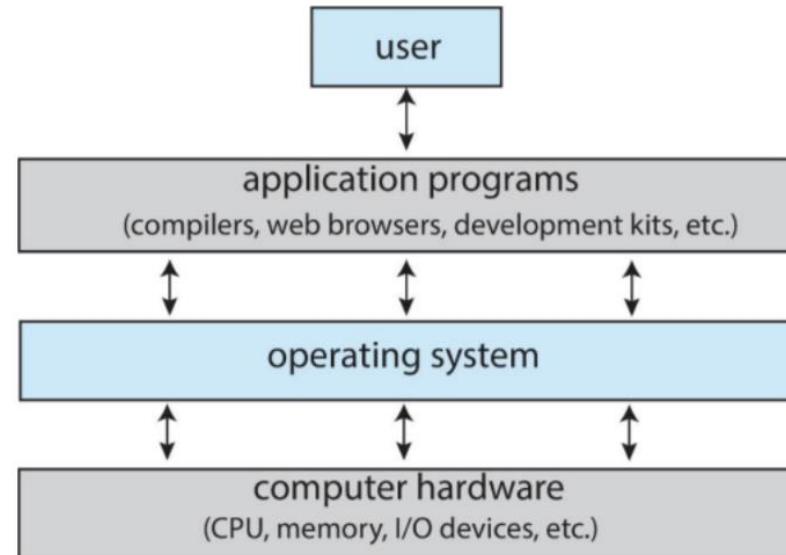
**Chọn D**



Sharing is learning

## Câu 2 : Bộ phận nào không phải là một thành phần chính trong cấu trúc hệ thống máy tính

- A. Chương trình ứng dụng
- B. Hệ Điều Hành
- C. Phần Cứng
- D. Màn Hình



**Chọn D**



Sharing is learning

### Câu 3: Nội dung nào không phải là một chức năng chính của hệ điều hành ?

- A. Quản lý thông tin website
- B. Phân chia thời gian xử lý và định thời CPU
- C. Phối hợp và đồng bộ giữa các tiến trình
- D. Quản lý tài nguyên hệ thống

### Các chức năng của hệ điều hành

- 1. Phân chia thời gian xử lý và định thời CPU.
- 2. Phối hợp và đồng bộ hoạt động giữa các processes.
- 3. Quản lý tài nguyên hệ thống.
- 4. Kiểm soát truy cập, bảo vệ hệ thống.
- 5. Duy trì sự nhất quán của hệ thống, kiểm soát lỗi và phục hồi.
- 6. Cung cấp giao diện làm việc cho users.

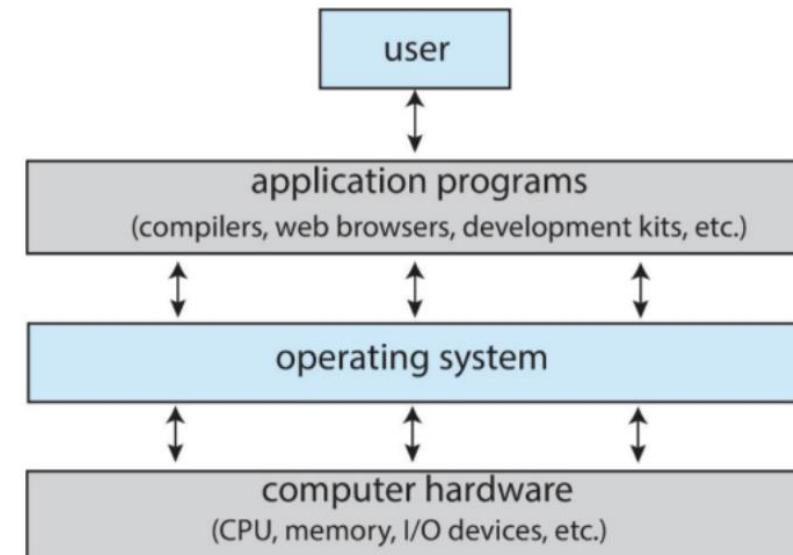
**Chọn A**



Sharing is learning

## Câu 4 : Hệ điều hành là chương trình trung gian giữa .. với phần cứng máy tính

- A. Các chương trình Ứng dụng
- B. Người sử dụng
- C. Phần mềm máy tính
- D. Bộ Xử Lý Trung Tâm



**Chọn A**



Sharing is learning

## Câu 5 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.

**Chọn B**



Sharing is learning

## Câu 6 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.

### \***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. => **Chọn D**



## Câu 7 : Đặc điểm nào dưới đây không phải là đặc điểm của hệ thống đa chương

- A. Nhiều công việc được nạp đồng thời vào bộ nhớ chính
- B. Có bộ giám sát thường trực
- C. Khi một tiến trình thực hiện I/O, một tiến trình khác được thực hiện
- D. Tận dụng được thời gian rảnh , tăng hiệu suất sử dụng CPU



Sharing is learning

## Hệ Thống Đơn Chương

- Chỉ một công việc được nạp vào bộ nhớ tại một thời điểm
- Tác vụ thi hành tuần tự.
- Bộ giám sát thường trực.

## 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.**

## Câu 7 : Đặc điểm nào dưới đây không phải là đặc điểm của hệ thống đa chương

- A. Nhiều công việc được nạp đồng thời vào bộ nhớ chính
- B. Có bộ giám sát thường trực
- C. Khi một tiến trình thực hiện I/O, một tiến trình khác được thực hiện
- D. Tận dụng được thời gian rảnh , tăng hiệu suất sử dụng CPU

**Chọn B**



Sharing is learning

## Câu 8 ( GK1 2023) : Chọn các phát biểu sai trong các phát biểu dưới đây

- A . Hệ điều hành là chương trình trung gian giữa phần cứng máy tính và người sử dụng, có chức năng điều khiển và phối hợp việc sử dụng phần cứng và cung cấp các dịch vụ cơ bản cho các ứng dụng
- B. Hệ điều hành hoạt động theo định hướng ngắt
- C. Có hai chế độ cơ bản trong hệ điều hành là chế độ người dùng và chế độ hạt nhân
- D. Hệ thống đơn chương cho phép nhiều công việc được nạp đồng thời vào bộ nhớ

**Chọn D**



Sharing is learning

## Câu 9 ( GKI 2022) : Đặc điểm nào dưới đây không phải là đặc điểm của hệ thống đa chương

- A. Nhiều công việc được nạp đồng thời vào bộ nhớ chính
- B. Tác vụ được thực thi tuần tự
- C. Khi một tiến trình thực hiện I/O, một tiến trình khác được thực hiện
- D. Tận dụng được thời gian rảnh , tăng hiệu suất sử dụng CPU

**Chọn B**



Sharing is learning

## Câu 10 : Lựa chọn nào dưới đây không phải là yêu cầu của hệ thống chia sẻ thời gian ?

- A. Quản lý kết nối giữa các máy tính
- B. Quản lý tiến trình
- C. Quản lý bộ nhớ
- D. Quản lý hệ thống lưu trữ

### Yêu cầu của hệ thống chia sẻ thời gian

- Định thời công việc
- Quản lý bộ nhớ
- Quản lý các quá trình
- Quản lý hệ thống file , hệ thống lưu trữ
- Cấp phát hợp lý các tài nguyên

**Chọn A**



Sharing is learning

## Câu 11 ( GKI 2019) : Đặc điểm nào dưới đây không phải là đặc điểm của hệ thống song song

- A. Hai hay nhiều bộ xử lí cùng chia sẻ một bộ nhớ
- B. Độ tin cao vì khi một bộ xử lí hỏng thì công việc của nó được chia sẻ giữa các bộ xử lí còn lại
- C. Có hai loại hệ thống song song là đa xử lí đối xứng và đa xử lí bất đối xứng
- D. Thời gian đáp ứng xử lí rất nghiêm ngặt , tất cả dữ liệu nằm trong bộ nhớ chính

**Chọn D** vì đây là đặc điểm của hệ thống nhúng thời gian thực



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**



Sharing is learning

## Câu Hỏi Trắc Nghiệm Chương II

**Câu 1: Thành phần nào không phải là thành phần của hệ điều hành**

- A. Quản lý tiểu trình
- B. Quản lý tiến trình
- C. Hệ thống bảo vệ
- D. Quản lý file

**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

**Chọn A**



Sharing is learning

## Câu 2 : Để hoàn thành công việc một tiến trình không cần tài nguyên gì

- A. CPU
- B. Màn hình
- C. Bộ Nhớ
- D. Thanh ghi

**Để hoàn thành một công việc một tiến trình cần**

- CPU
- Bộ Nhớ
- File
- Thiết bị I/O

**Chọn B**



Sharing is learning

## Câu Hỏi Trắc Nghiệm Chương II

### Câu 3 : Chọn đáp án đúng theo thứ tự tăng dần truy cập bộ nhớ

- A. Ram ,SSD ,Cache ,HDD
- B. Cache ,Ram ,SSD ,HDD
- C. HDD, SSD, RAM, Cache
- D. Ram ,Cache ,HDD , SSD

**Chọn C**



Sharing is learning

## Câu 4 : Quyết định sẽ nạp khi có vùng nhớ trống là chức năng của thành phần nào trong hệ điều hành

- A. Quản lý hệ thống I/O
- B. Quản lý tiến trình
- C. Quản lý hệ thống lưu trữ thứ cấp
- D. Quản lý bộ nhớ chính

### Nhiệm Vụ Của Quản Lý Bộ Nhớ Chính

- Theo dõi và quản lý các vùng nhớ trống và đã cấp phát
- Quyết định sẽ nạp chương trình khi có vùng nhớ trống
- Cấp phát và thu hồi các vùng nhớ cần thiết

**Chọn D**



Sharing is learning

**Câu 5 : Cung cấp giao diện chung đến các trình điều khiển thiết bị là chức năng của thành phần nào trong hệ điều hành ?**

- A. Hệ thống bảo vệ
- B. Quản lý hệ thống I/O
- C. Quản lý bộ nhớ chính
- D. Quản lý hệ thống lưu trữ thứ cấp

### **Chức năng của hệ thống I/O**

- Cơ chế : buffering , caching ,spooling
- Cung cấp giao diện chung đến các trình điều khiển thiết bị
- Bộ điều khiển các thiết bị phần cứng

### **Chọn B**



Sharing is learning

## Câu 5 : Hệ thống dịch lệnh (Shell /Terminal ) là gì ?

- A. là giao diện chủ yếu giữa người dùng và phần cứng máy tính
- B. Là giao diện chủ yếu giữa tiến trình và hệ điều hành
- C. Là giao diện chủ yếu giữa người dùng và phần cứng máy tính
- D. Là giao diện chủ yếu giữa người dùng và hệ điều hành

**Hệ thống dịch lệnh** là giao diện chủ yếu giữa người dùng và hệ điều hành

**Chọn D**



Sharing is learning

## Câu 6 : Ý nào trong các ý sau không phải là do một dịch vụ hệ điều hành cung cấp

- A. Phát hiện lỗi
- B. Thực thi chương trình
- C. Cung cấp internet
- D. Cấp phát tài nguyên

Các dịch vụ cơ bản của Hệ Điều hành

- 1. Thực thi chương trình
- 2. Thực hiện các thao tác I/O theo yêu cầu của chương trình
- 3. Các thao tác trên hệ thống file
- 4. Phát hiện lỗi
- 5. Cấp phát tài nguyên , kế toán , bảo vệ , an ninh
- 6. Giao diện người dùng

**Chọn C**



Sharing is learning

# Câu 7 : Lệnh nào không phải là một lệnh gọi hệ thống trong Unix ?

- A. Block()
- B. Chmod()
- C. Getpid()
- D. Chown()

**Chọn A**

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()



Sharing is learning

**Câu 8 : Để tạo một tiến trình mới trên hệ điều hành windows cần sử dụng lời gọi hệ thống nào ?**

- A. Exit()
- B.Fork()
- C. CreateProcess()
- D.ExitProcess()

**Chọn C**



Sharing is learning

## Câu 9 : Để giao tiếp giữa tiến trình và hệ điều hành ta dùng

- A. Lời gọi hệ thống
- B. Chương trình ứng dụng
- C. Chương trình hệ thống
- D. Thông điệp

Chọn A



Sharing is learning

**Câu 10 Ý nào sau đây không phải là phương pháp truyền tham số khi sử dụng lệnh gọi hệ thống ?**

- A. Qua Stack
- B. Qua một vùng nhớ , địa chỉ của vùng nhớ được gửi đến hệ điều hành qua thanh ghi
- C. Qua message
- D. Qua thanh ghi



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

1. Hệ điều hành đơn nhiệm và hệ điều hành đa nhiệm được phân loại dưới góc độ nào?

- A. Loại máy tính
- B. Số chương trình được sử dụng cùng lúc.**
- C. Hình thức xử lý
- D. Số lượng người dùng truy xuất tài nguyên cùng lúc.



Sharing is learning

# Câu 1. Hệ điều hành đơn nhiệm và hệ điều hành đa nhiệm được phân loại dưới góc độ nào?

- A. Loại máy tính
- B. Số chương trình được sử dụng cùng lúc.
- C. Hình thức xử lý
- D. Số lượng người dùng truy xuất tài nguyên cùng lúc.

**Chọn B**



Sharing is learning

**Câu 2. “Quyết định sẽ nạp chương trình nào khi có vùng nhớ trống” là chức năng của thành phần nào trong hệ điều hành?**

- A. Quản lý tiến trình
- B. Quản lý hệ thống lưu trữ thứ cấp
- C. Quản lý hệ thống I/O
- D. Quản lý bộ nhớ chính

**Chọn D**



Sharing is learning

# Câu 3 Thành phần nào sau đây không phải là một thành phần của hệ điều hành

- A. Quản lý hệ thống File
- B. Quản lý bộ nhớ
- C. Quản lý kernel
- D. Quản lý tiến trình

**Chọn C**



Sharing is learning

#### Câu 4. Đặc điểm của cấu trúc vi nhân là gì?

- A. Một số chức năng của hệ điều hành được chuyển từ kernel space sang user space.
- B. Hệ điều hành được chia thành nhiều lớp (layer).
- C. Sử dụng cách tiếp cận hướng đối tượng, mỗi core thành phần là tách biệt nhau.
- D. Các module trao đổi với nhau thông qua các interfaces.

**Chọn A**



Sharing is learning

## Câu 5. Cấu trúc của một hệ thống máy tính gồm có những thành phần chính nào:

- A. Bộ vi xử lý (CPU), Bộ nhớ (Main Memory), Hệ điều hành (OS) và Các thiết bị nhập xuất (I/O devices)
- B. Phần cứng (Hardware), Hệ điều hành (OS), Bộ nhớ (Main Memory) và Các chương trình ứng dụng (Application Programs)
- C. Bộ vi xử lý (CPU), Bộ nhớ (Main Memory), Hệ điều hành (OS) và Các chương trình ứng dụng (Application Programs)
- D. Phần cứng (Hardware), Hệ điều hành (OS), Các chương trình ứng dụng (Application Programs) và Người dùng (Users)

**Chọn D**



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**



# 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

- 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

## 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.



Sharing is learning

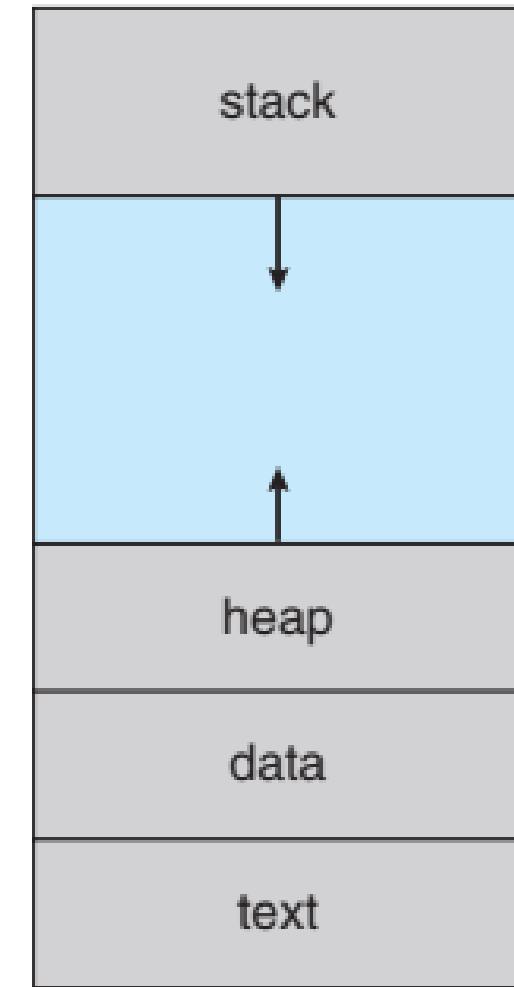
# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 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

max



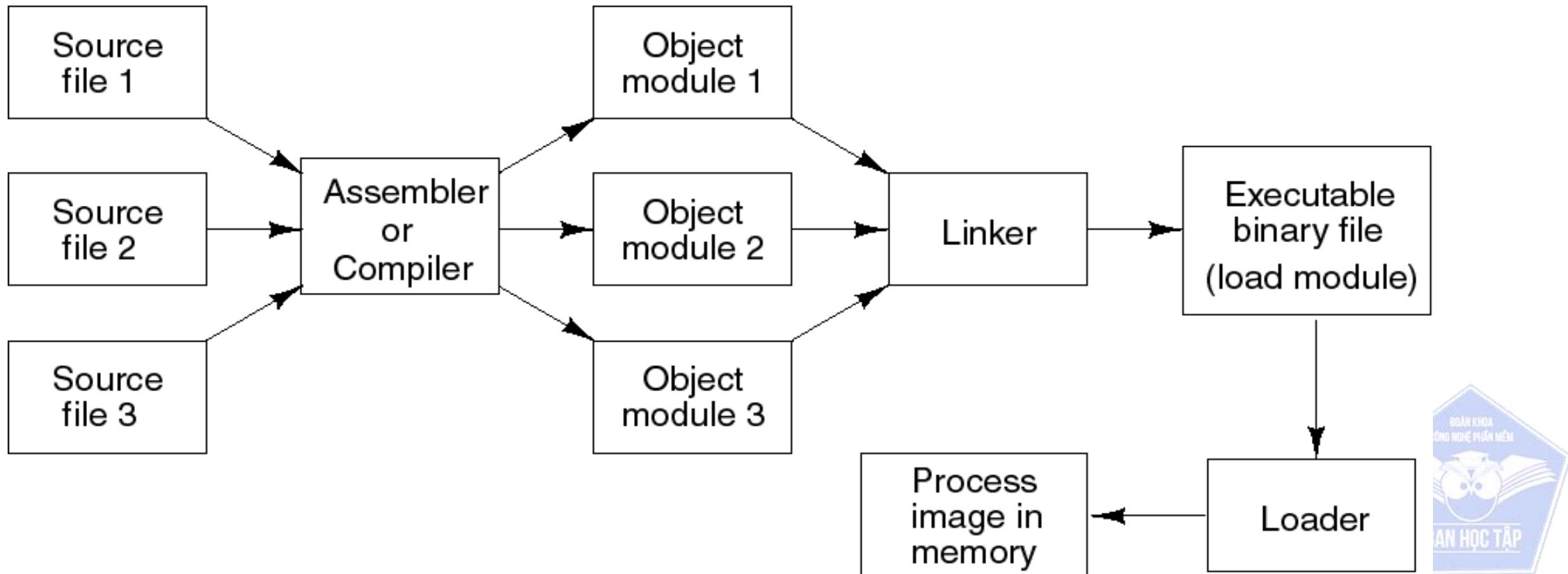
0



# 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, ...).



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

## 2. Các trạng thái của tiến trình.

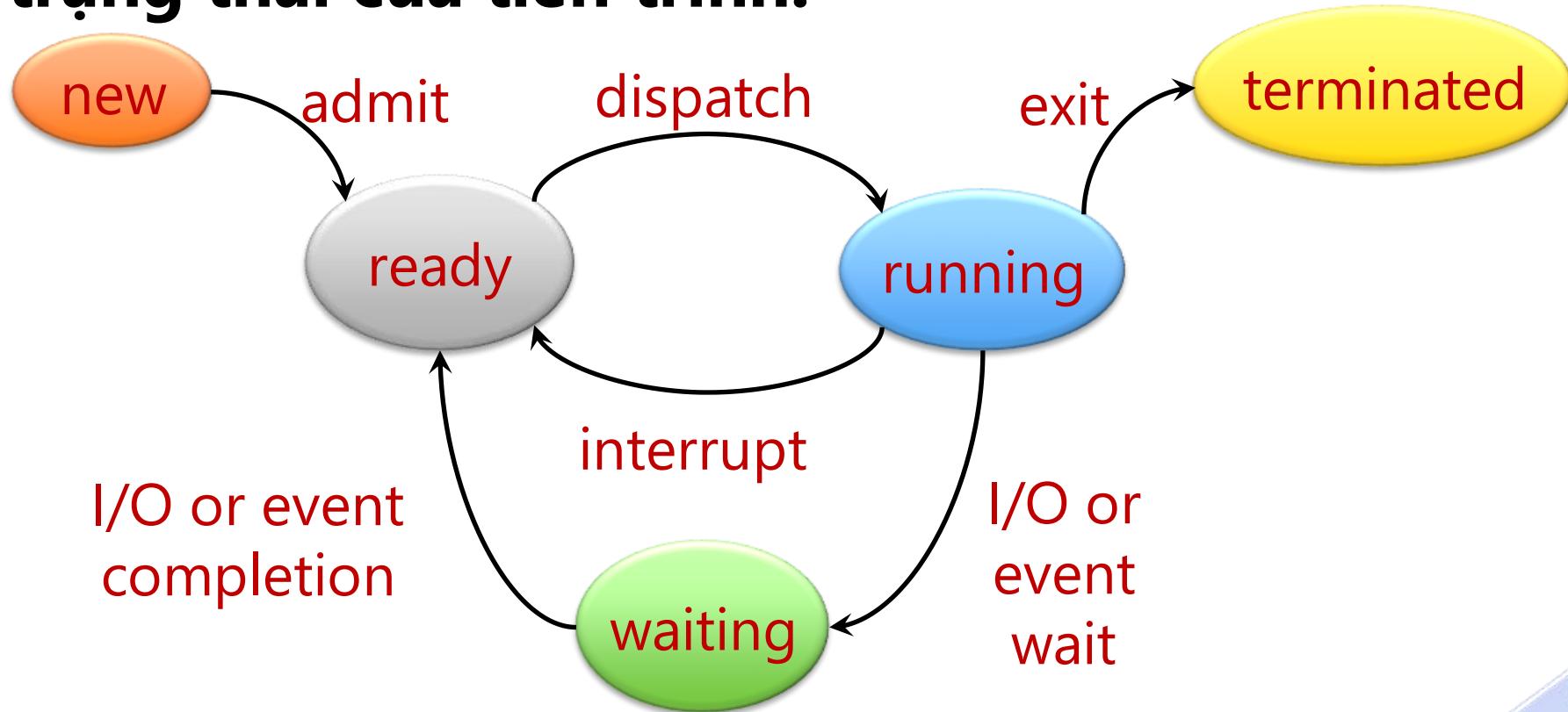
- **new**: tiến trình vừa được tạo
- **ready**: tiến trình đã có đủ tài nguyên, chỉ còn cần CPU
- **running**: các lệnh của tiến trình đang được thực thi
- **waiting** (hay **blocked**): tiến trình **đợi I/O hoàn tất**, hoặc đợi tín hiệu
- **terminated**: tiến trình đã kết thúc



Sharing is learning

# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 2. Các trạng thái của tiến trình.



Chuyển đổi giữa các trạng thái của tiến trình



Sharing is learning

# 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 >  
terminated

```
int main(int argc, char** argv) {  
    int a, i = 2;  
    while (i <=5) {  
        i++;  
        if (i % 2 == 0)  
        {  
            a = 5*4;  
        } else {  
            a = 5*9;  
            printf ("Gia tri so a = %d", &a);  
        }  
    }  
    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

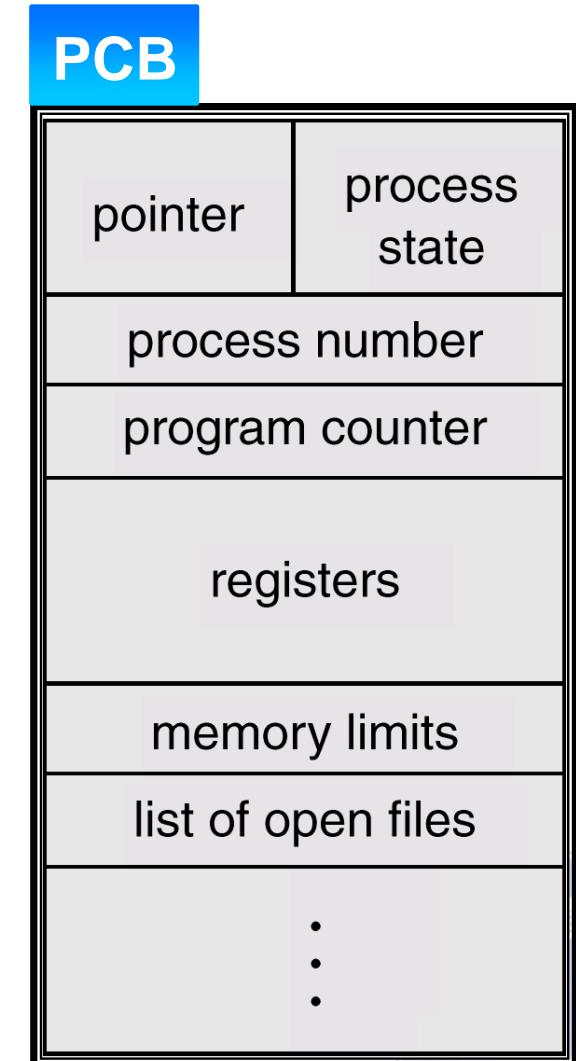
## 3. Process Control Block

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.



Sharing is learning

# 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



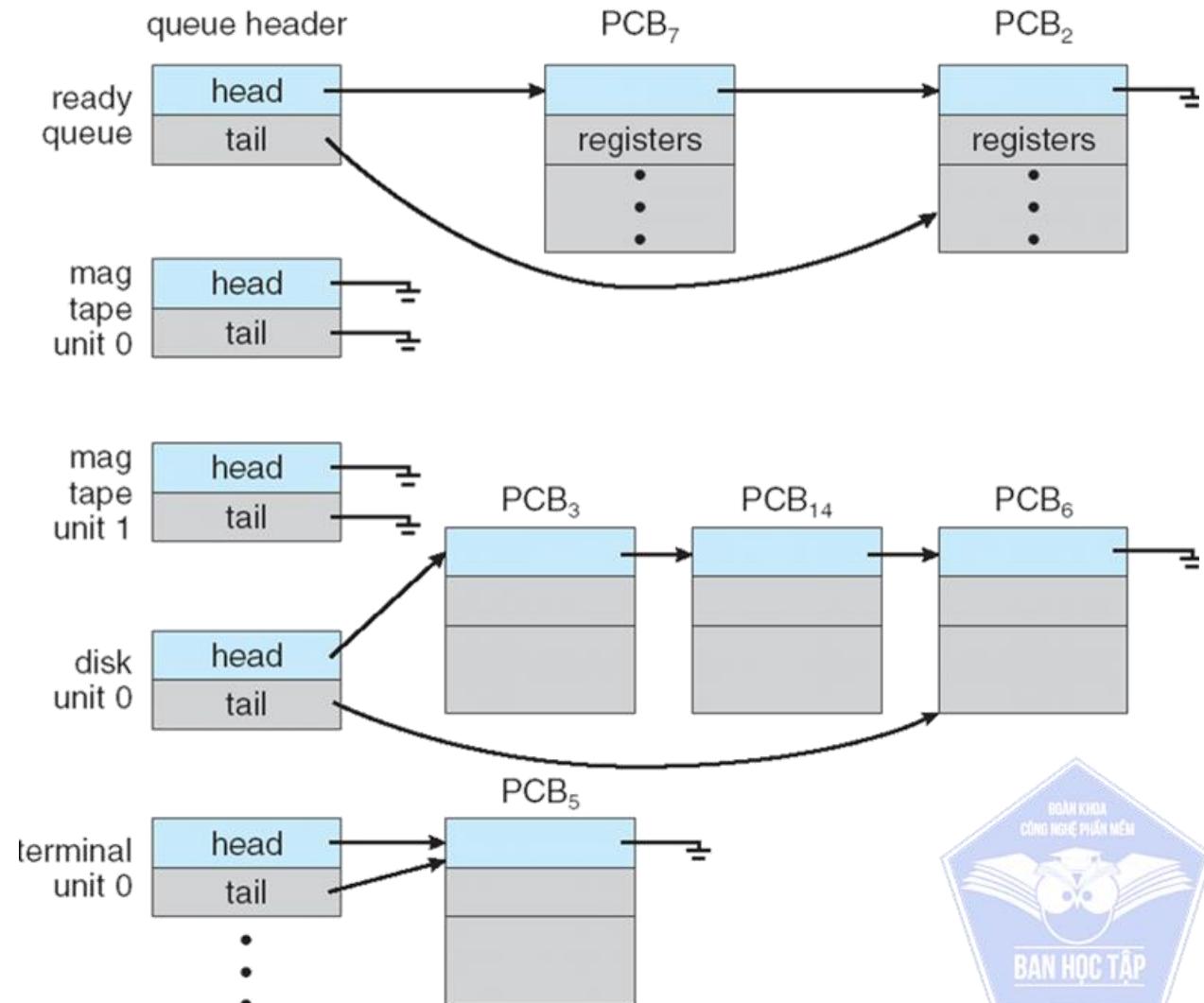
Sharing is learning

# 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
- ...



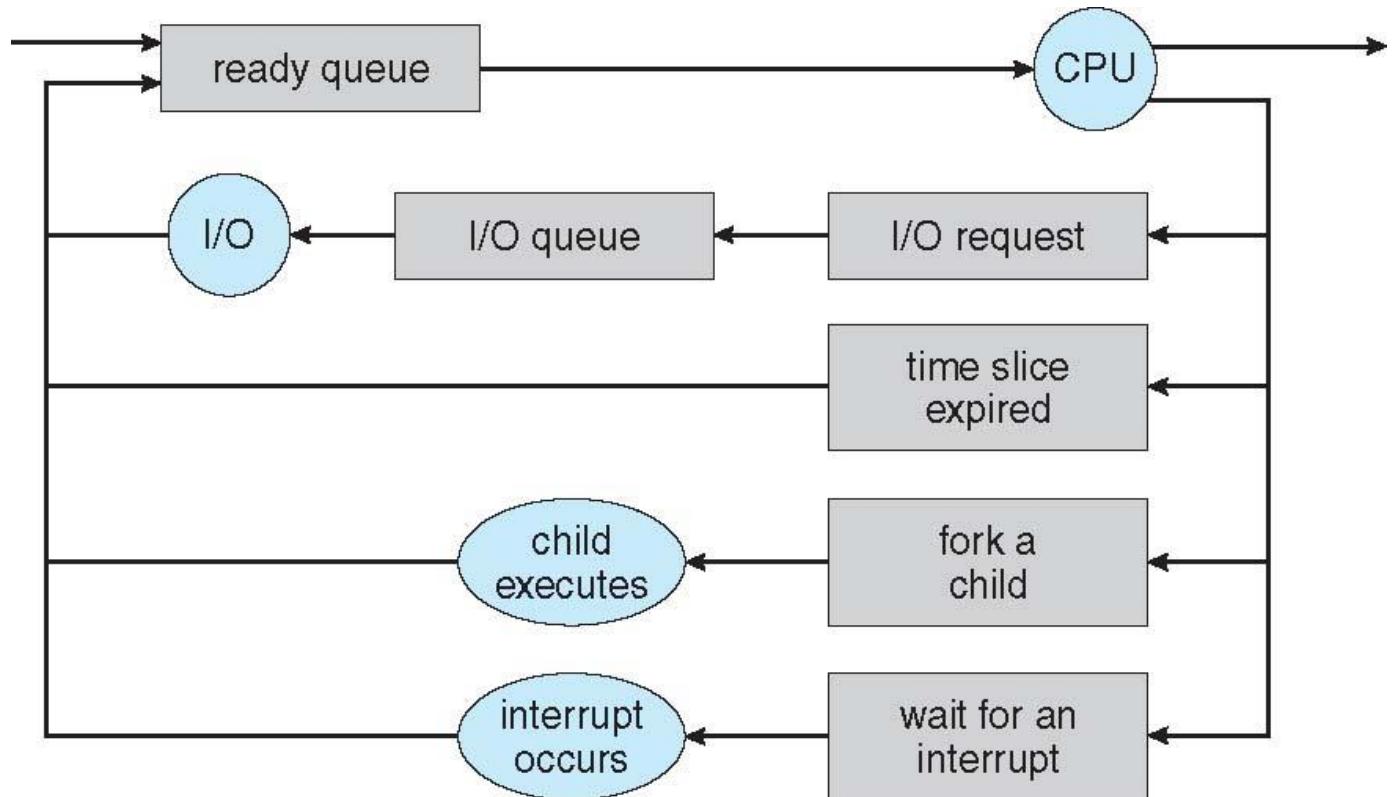
Sharing is learning

# 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



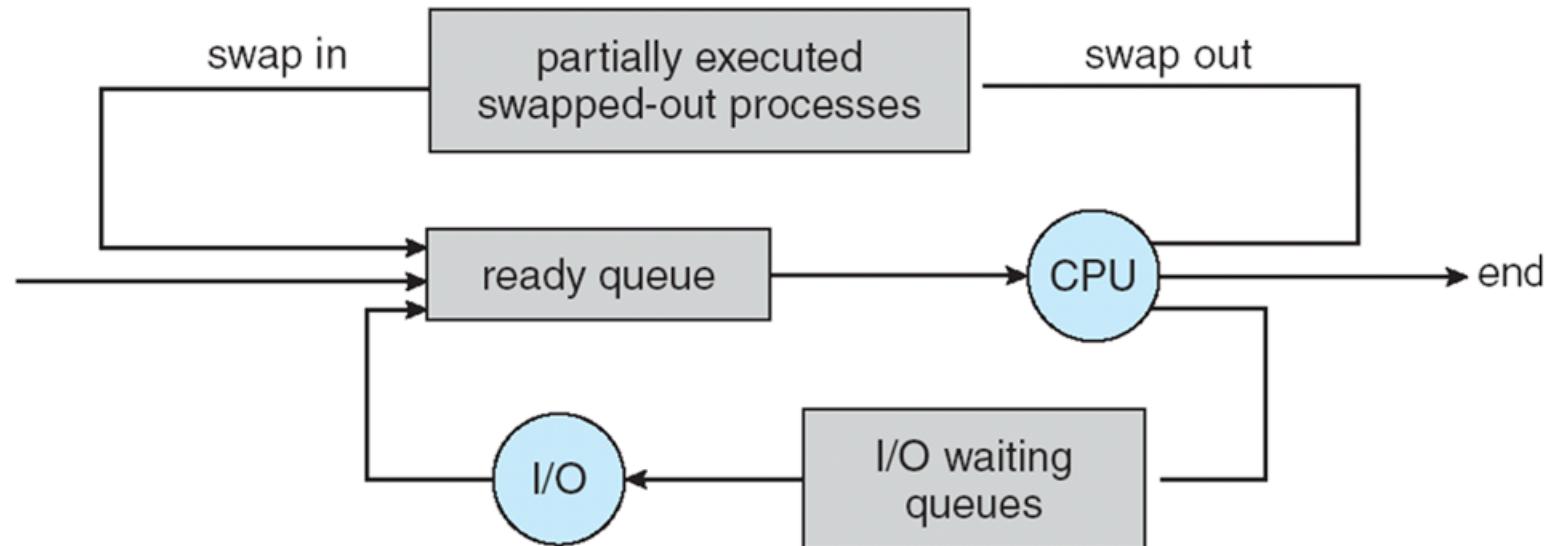
# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 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)

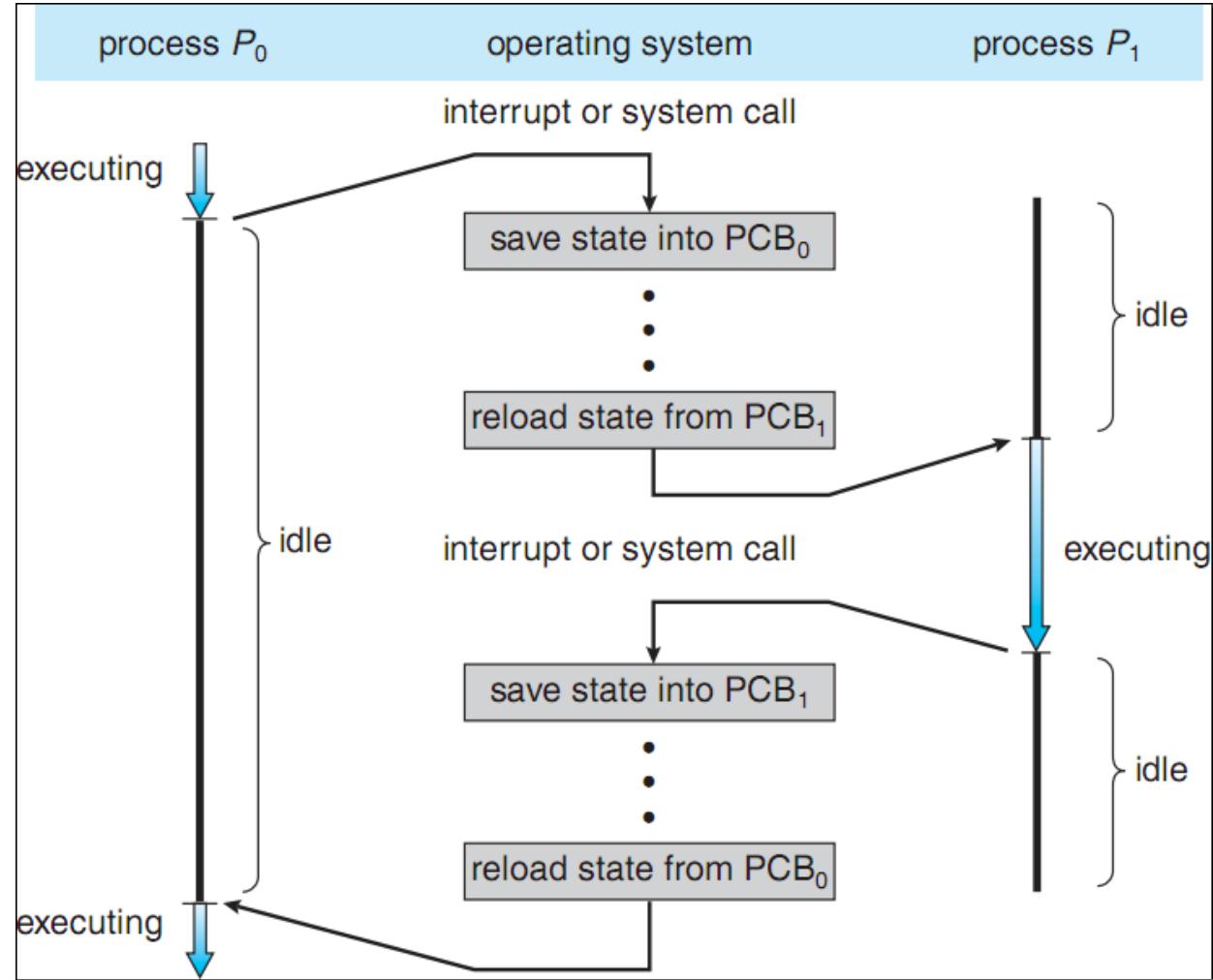


# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 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.



Sharing is learning

# 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



Sharing is learning

# 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

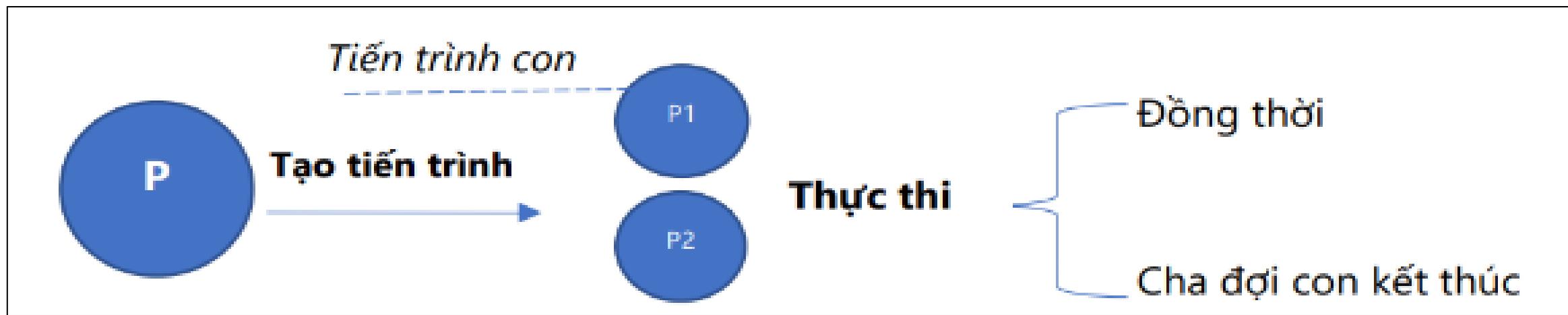


Sharing is learning

# 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,...).



Sharing is learning

# 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

⇒ 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

## 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



Sharing is learning

# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 5. Các tác vụ đối với tiến trình

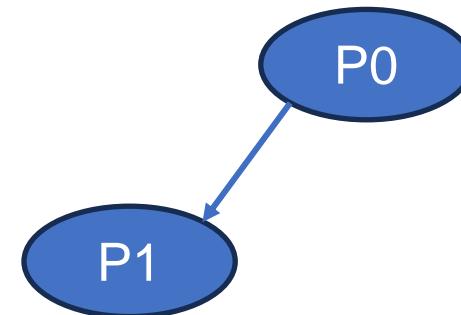
Ví dụ:

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

⇒ Cho biết output?

Hướng dẫn làm

Cây tiến trình:



OUTPUT: BHT CNPMBHT CNPM



Sharing is learning

# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 5. Các tác vụ đối với tiến trình

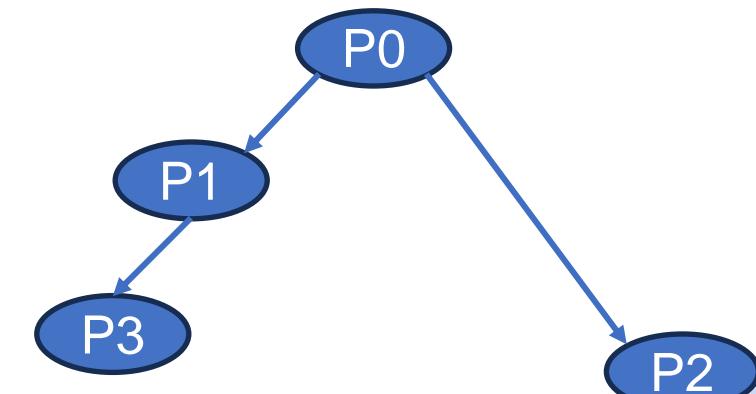
Ví dụ:

```
#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:

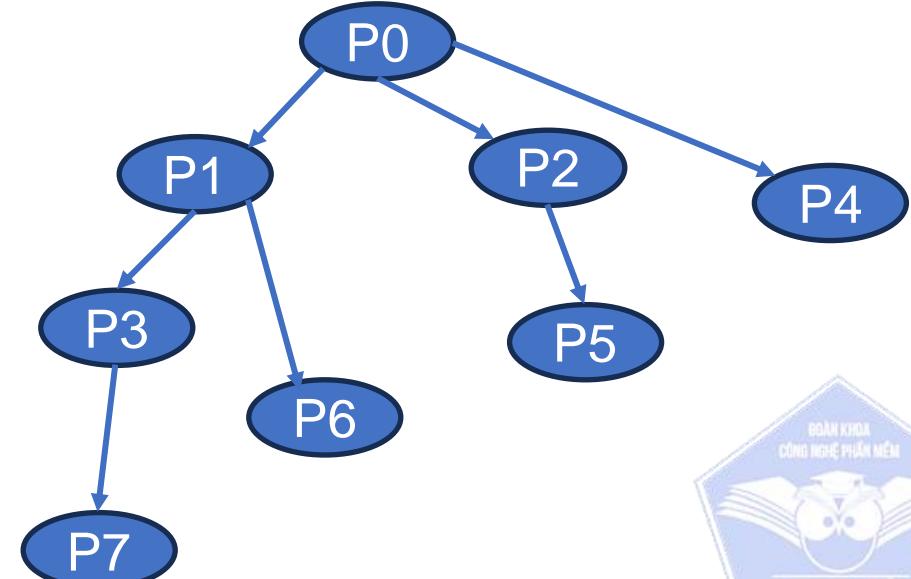
```
#include <stdio.h>
int main() {
    printf("Hello, Alo")
    fork();
    fork();
    printf("Hi");
    fork();
    printf("Alo");
}
```

⇒ 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:



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

## 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

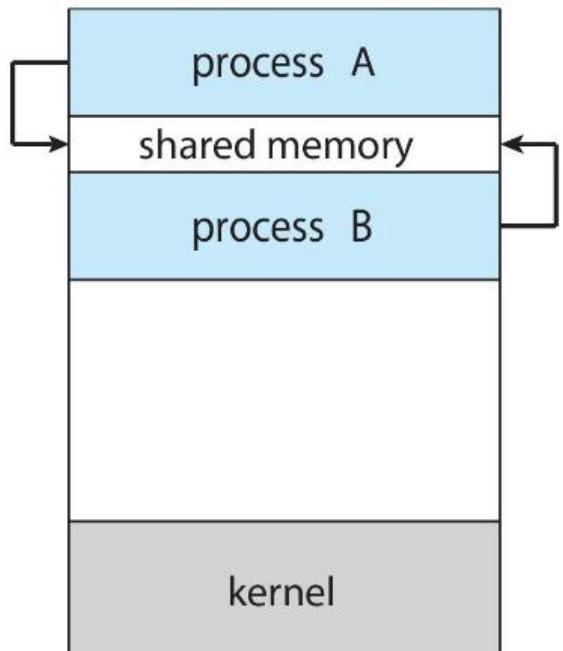


Sharing is learning

# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

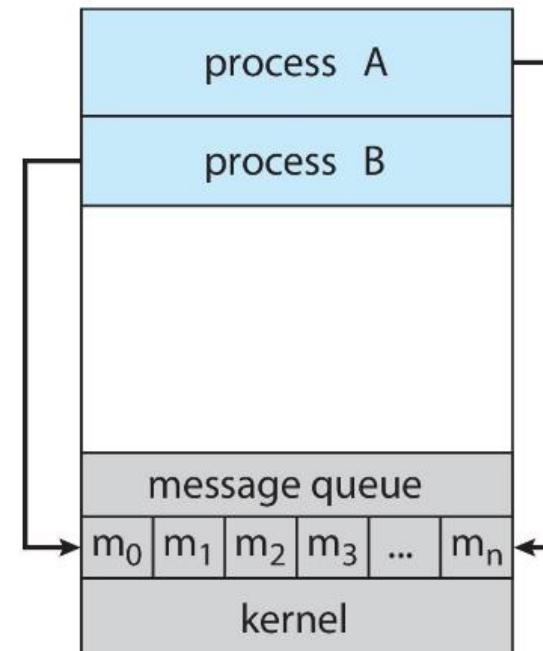
## 6. Giao tiếp liên tiến trình.

(a) Shared memory.



(a)

(b) Message passing.



(b)



Sharing is learning

# 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.



Sharing is learning

# 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
- 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.**

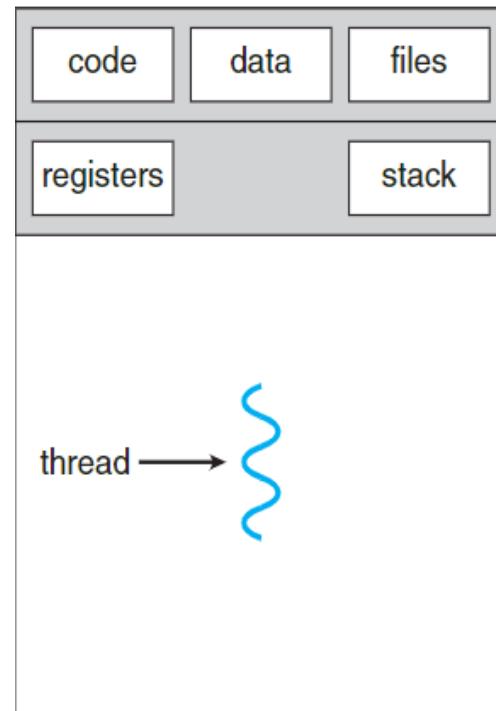


Sharing is learning

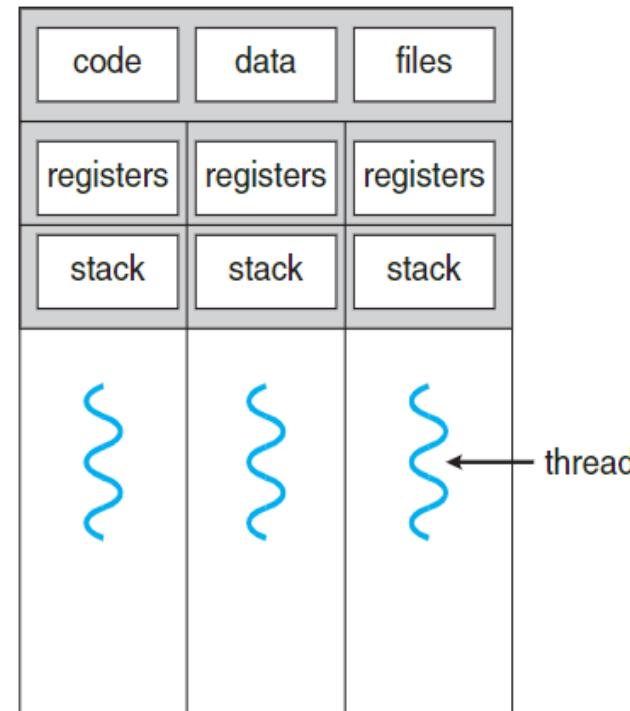
# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 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, resources.



single-threaded process



multithreaded process



Sharing is learning

# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 7. Tiểu trình.

### Lợi ích của tiểu 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

## 7. Tiểu trình.

### Các mô hình đa tiểu trình:

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



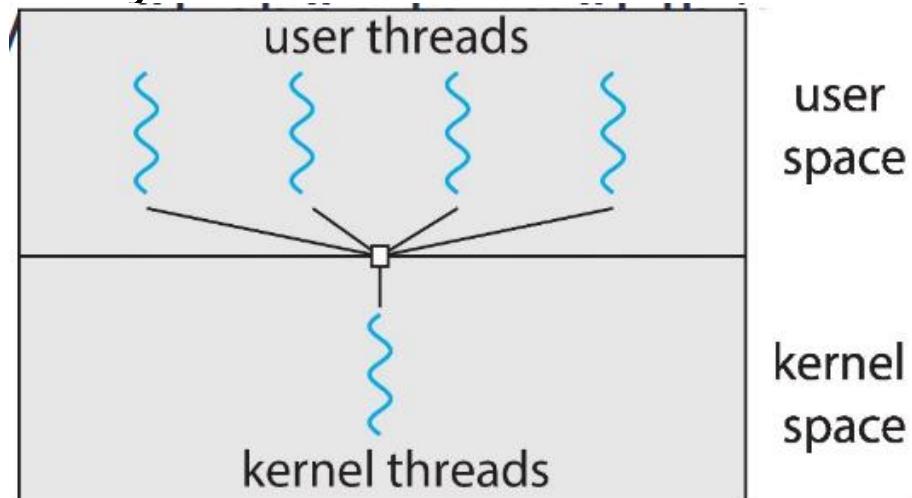
Sharing is learning

# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 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.

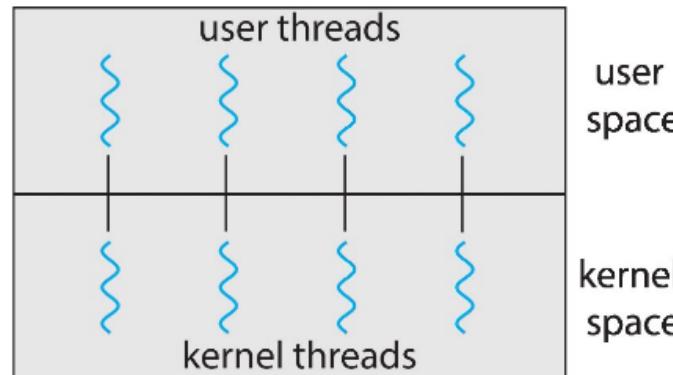


# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 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.



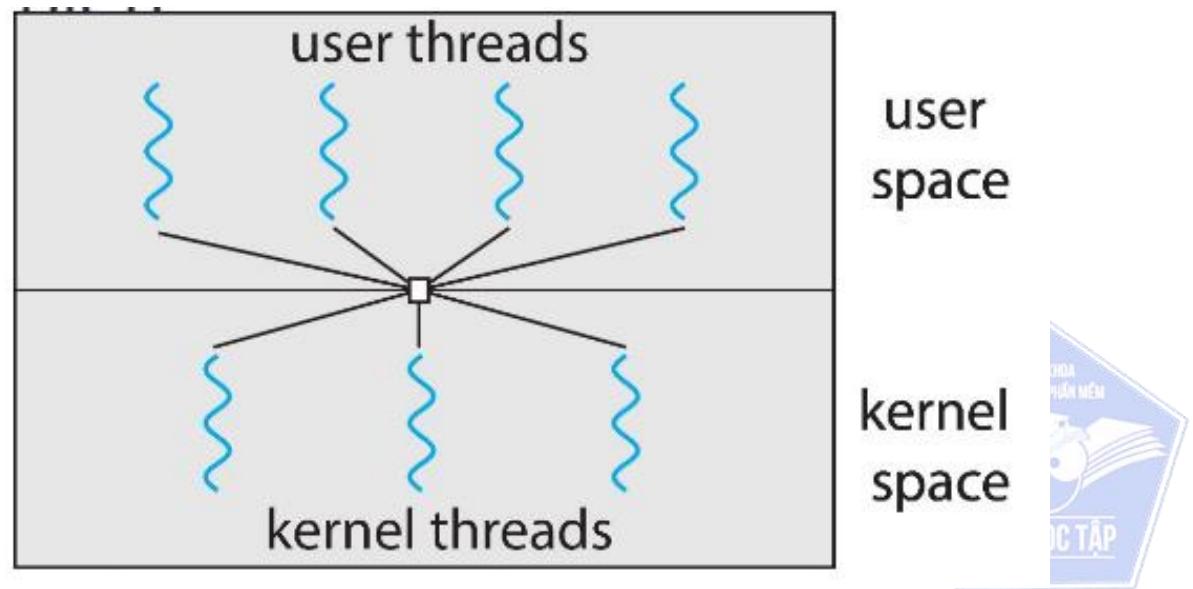
# CHƯƠNG 3: QUẢN LÝ TIẾN TRÌNH

## 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  $\Rightarrow$  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



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



# 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**



Sharing is learning

# 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.



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

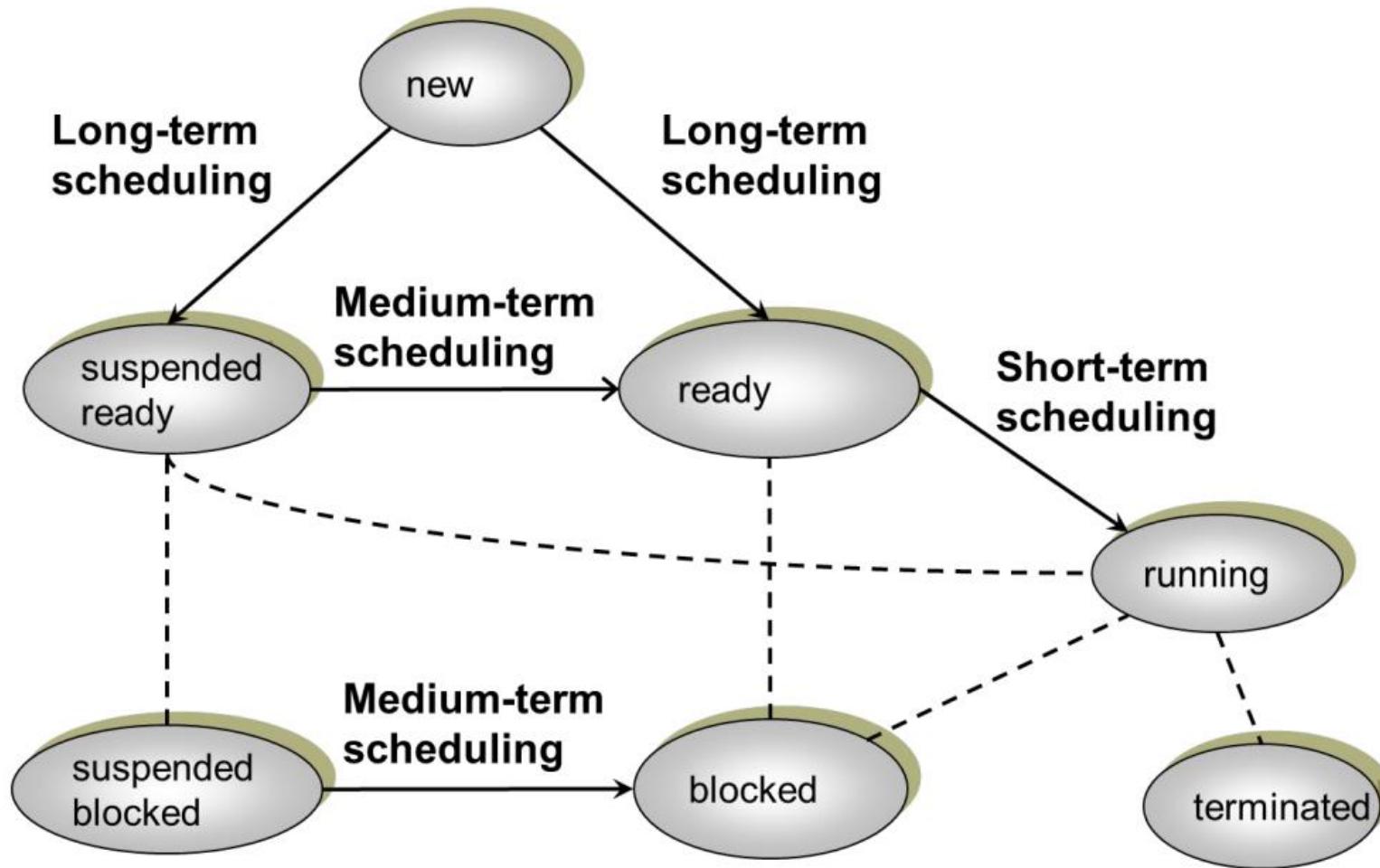
- 1. Các loại bộ định thời.**
2. Các tiêu chuẩn định thời.
3. Các yếu tố của giải thuật định thời.
4. Các giải thuật định thời.



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 1. Các loại bộ định thời



Sharing is learning

# 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 lẩn 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.



Sharing is learning

# 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.



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

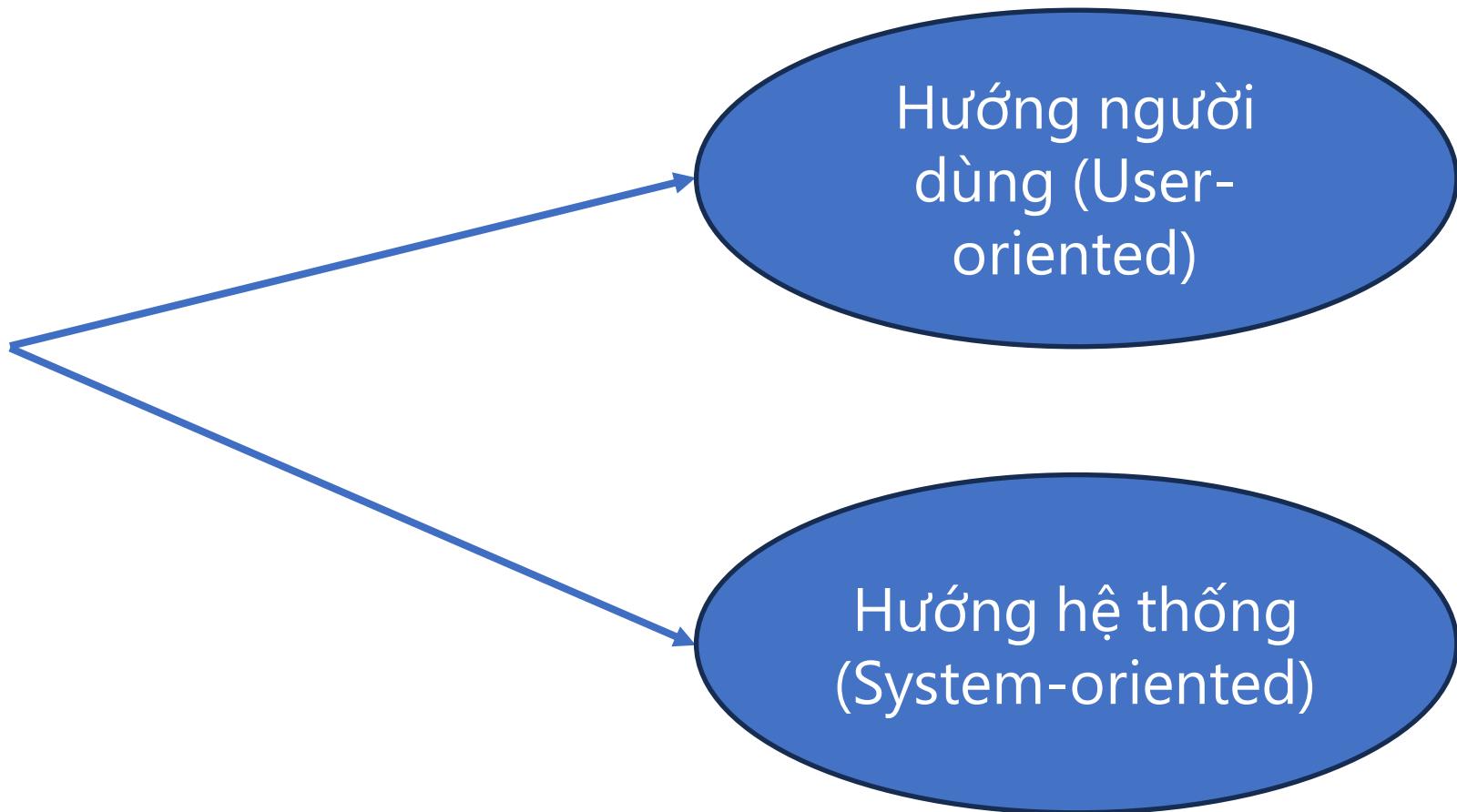
1. Các loại bộ định thời.
- 2. Các tiêu chuẩn định thời.**
3. Các yếu tố của giải thuật định thời.
4. Các giải thuật định thời.



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 2. Các tiêu chuẩn định thời CPU:



Sharing is learning

# 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.

⇒ **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.

⇒ **CỰC ĐẠI**



Sharing is learning

# 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.



Sharing is learning

# 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,...)



Sharing is learning

# 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 trung dung (Non-preemptive): Khi ở trạng thái running, process sẽ thực thi cho đến khi kết thúc hoặc bị blocked do yêu cầu I/O.
  - Trung dung (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.



Sharing is learning

# 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.



Sharing is learning

# 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 Robin
- e) Highest-Response-Ratio-Next
- f) Multilevel Queue và Multilevel Feedback Queue



Sharing is learning

# 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**



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

**\*Tóm tắt công thức**

**Response\_time = Start\_time - Arrival\_time**

**Waiting\_time = Finish\_time – Arrival\_time – Burst\_time (TỔNG QUÁT)**

Với non-preemptive: Waiting\_time = Response\_time

**Turnaround\_time = Finish\_time - Arrival\_time**



Sharing is learning

# 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 Robin
- e) Highest-Response-Ratio-Next
- f) Multilevel Queue và Multilevel Feedback Queue



Sharing is learning

# 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.



Sharing is learning

# 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.



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### a) *First-Come-First-Serve*

Giản đồ Gantt

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



**Thời gian đợi:**

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



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### a) *First-Come-First-Serve*

Giản đồ Gantt

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



**Thời gian đáp ứng:**

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



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### a) *First-Come-First-Serve*

Giản đồ Gantt

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



**Thời gian hoàn thành:**

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



Sharing is learning

# 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 Robin
- e) Highest-Response-Ratio-Next
- f) Multilevel Queue và Multilevel Feedback Queue



Sharing is learning

# 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.



Sharing is learning

# 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.



Sharing is learning

# 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.



Sharing is learning

# 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.



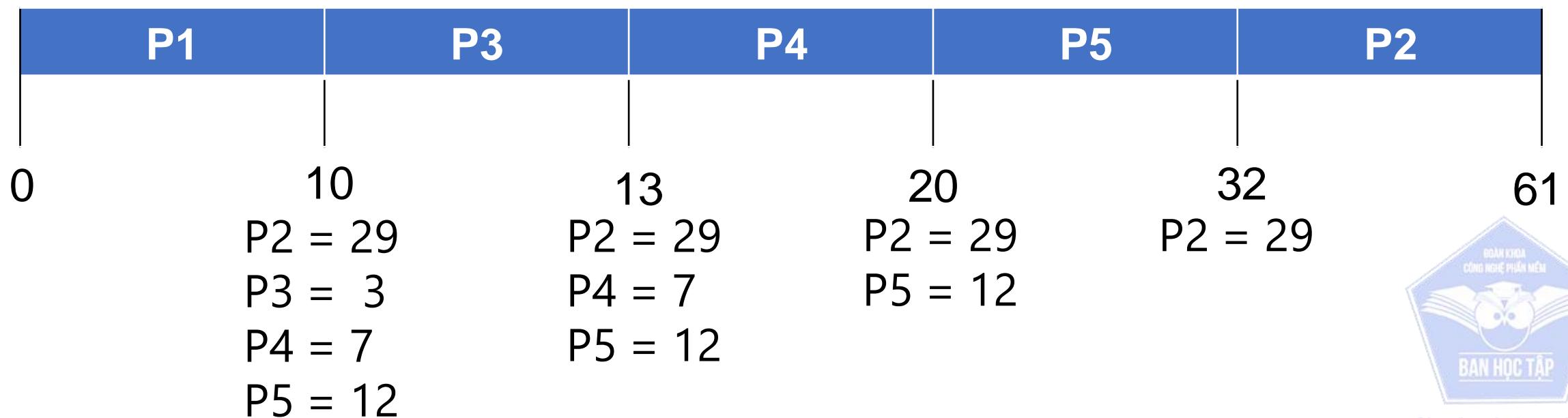
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.1) Shortest-Job-First

Giản đồ Gantt

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



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.1) Shortest-Job-First

Giản đồ Gantt

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



#### Thời gian đợi:

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



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.1) Shortest-Job-First

Giản đồ Gantt

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



### Thời gian đáp ứng:

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



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.1) Shortest-Job-First

Giản đồ Gantt

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



**Thời gian hoàn thành:**

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



Sharing is learning

# 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.



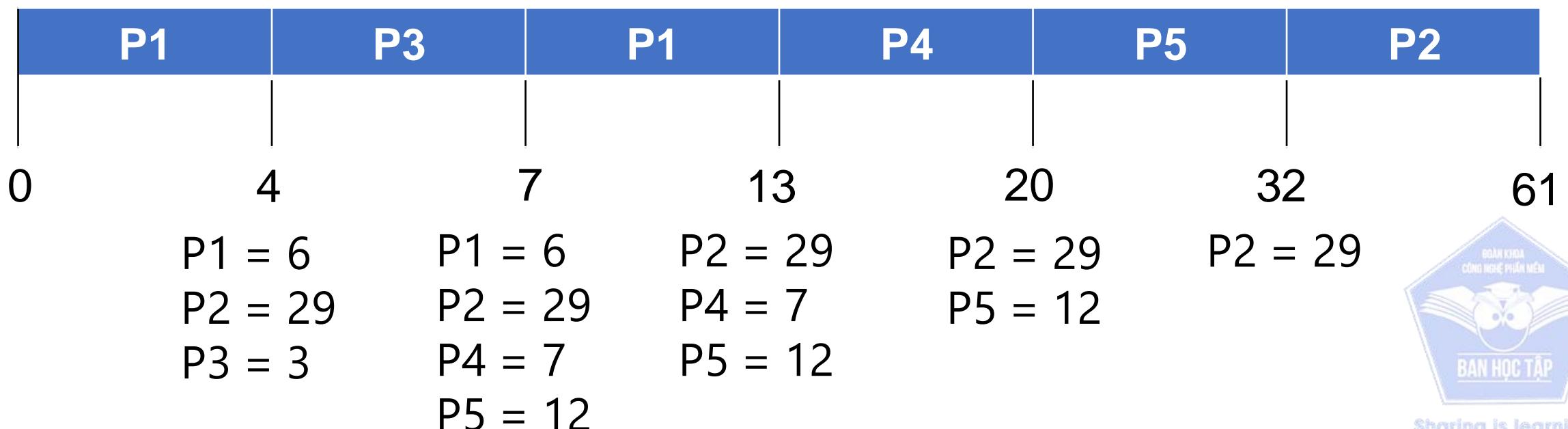
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.2) SRTF (Preemptive SJF)

Giản đồ Gantt

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



Sharing is learning

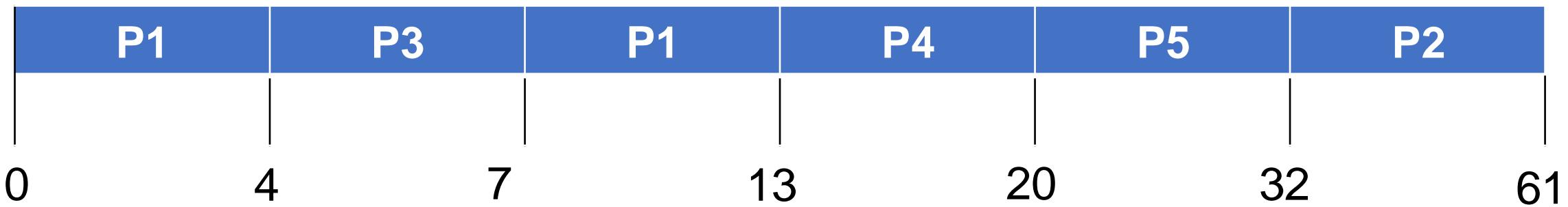
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.2) SRTF (Preemptive SJF)

Giản đồ Gantt

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



#### Thời gian đợi:

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



Sharing is learning

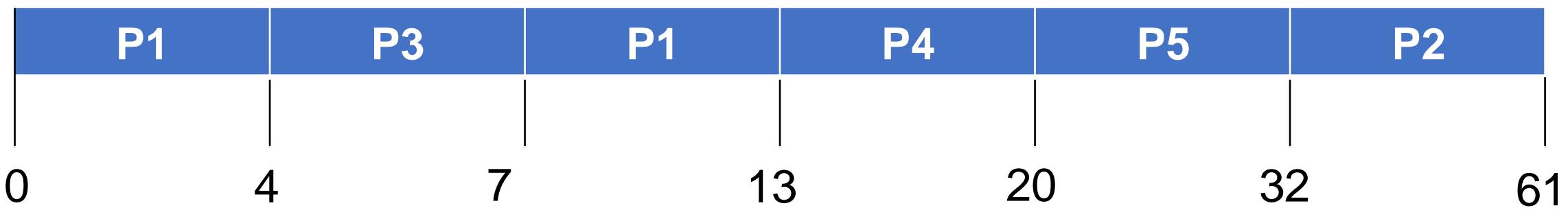
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.2) SRTF (Preemptive SJF)

Giản đồ Gantt

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



### Thời gian đáp ứng:

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



Sharing is learning

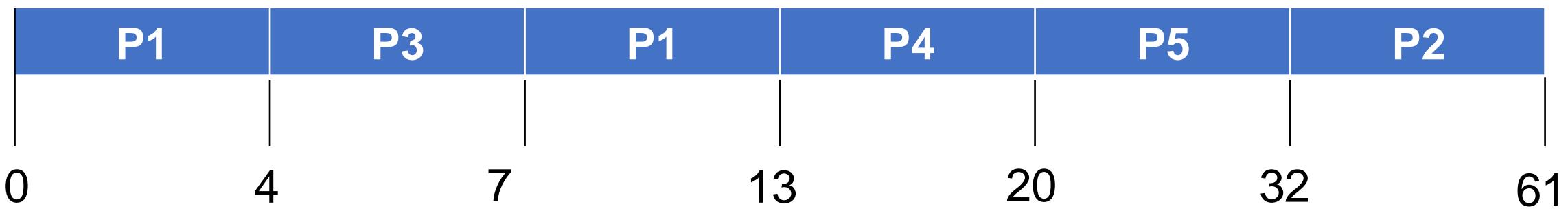
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### b.2) SRTF (Preemptive SJF)

Giản đồ Gantt

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



### Thời gian hoàn thành:

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



Sharing is learning

# 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 Robin
- e) Highest-Response-Ratio-Next
- f) Multilevel Queue và Multilevel Feedback Queue



Sharing is learning

# 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.



Sharing is learning

# 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.



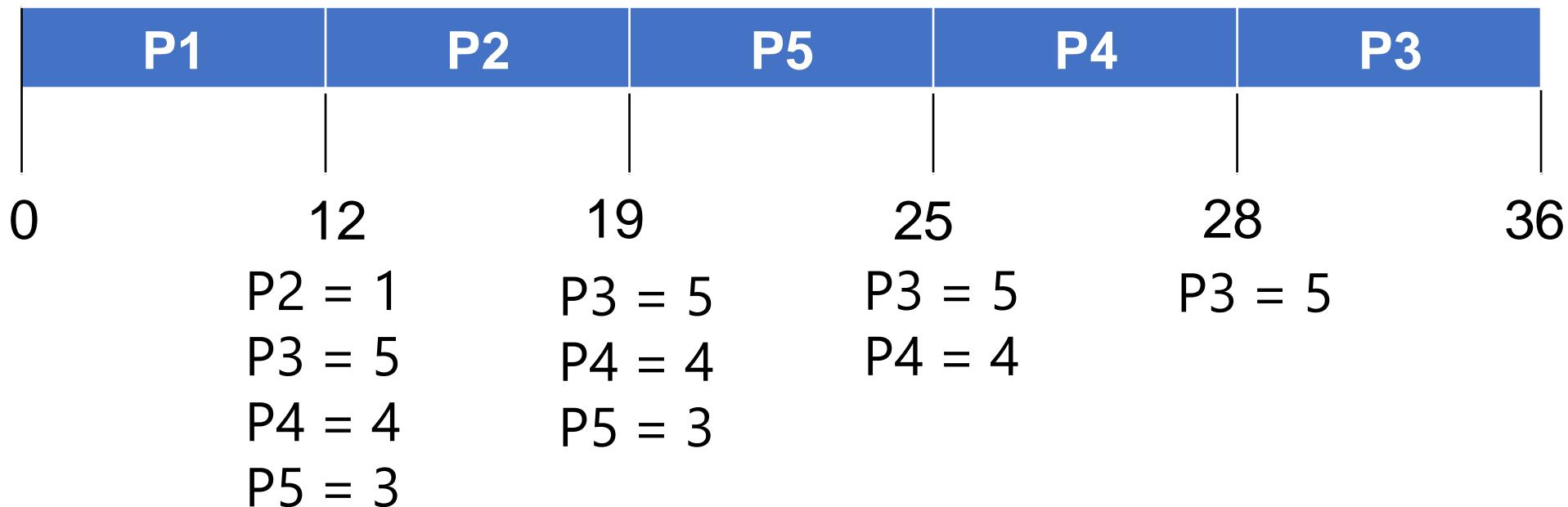
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 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



Sharing is learning

# 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:  $(0+10+23+16+7)/5=11.2$



Sharing is learning

# 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:  $(0+10+23+16+7)/5=11.2$



Sharing is learning

# 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 hoàn thành trung bình:  $(12+7+31+19+13)/5=16.4$



Sharing is learning

# 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.



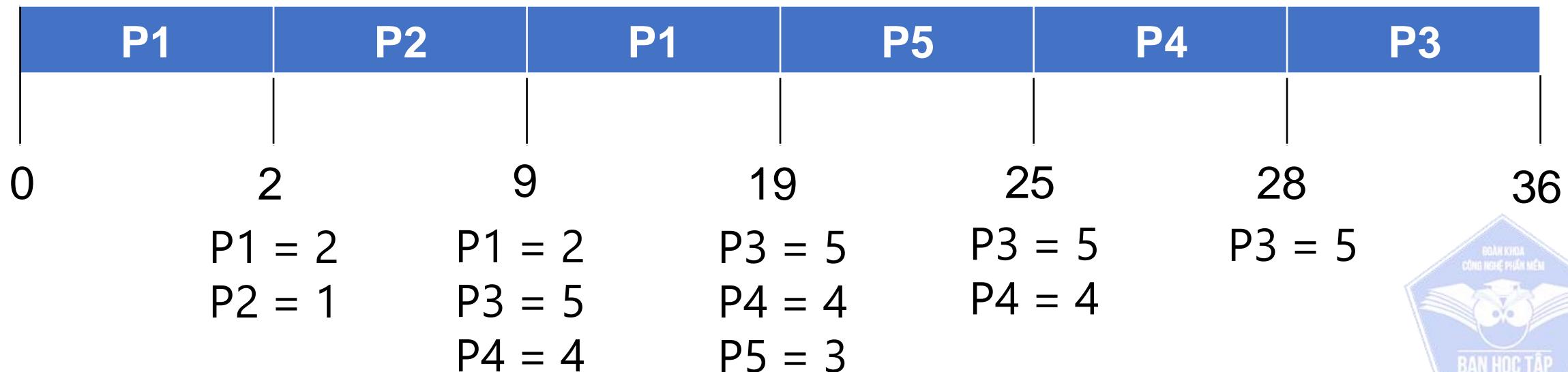
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 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



Sharing is learning

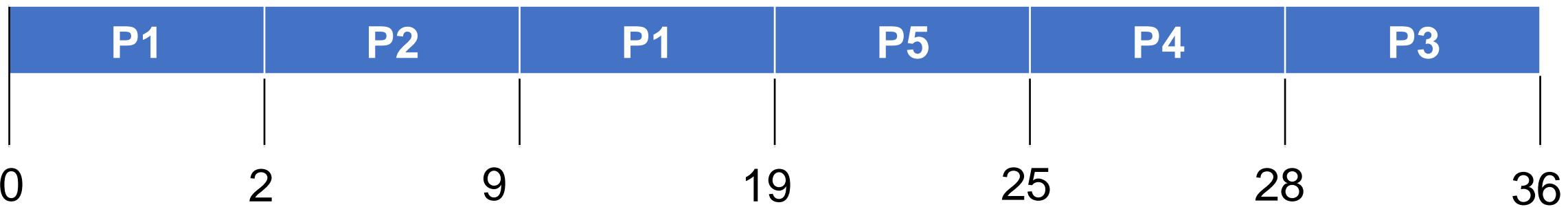
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 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:  $(7+0+23+16+7)/5=10.6$



Sharing is learning

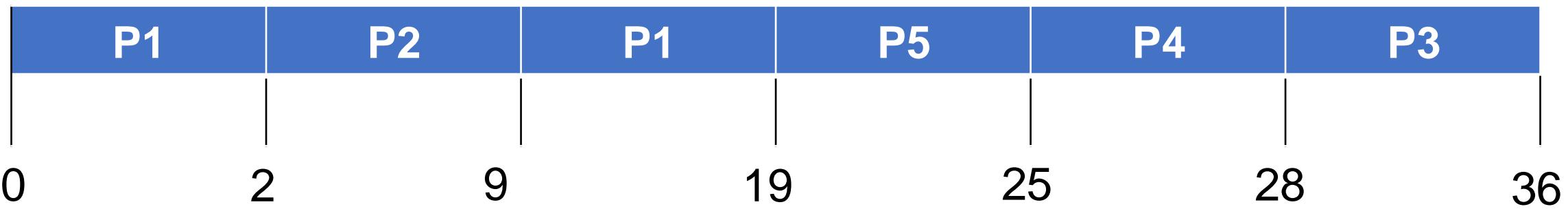
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 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:  $(0+0+23+16+7)/5=9.2$



Sharing is learning

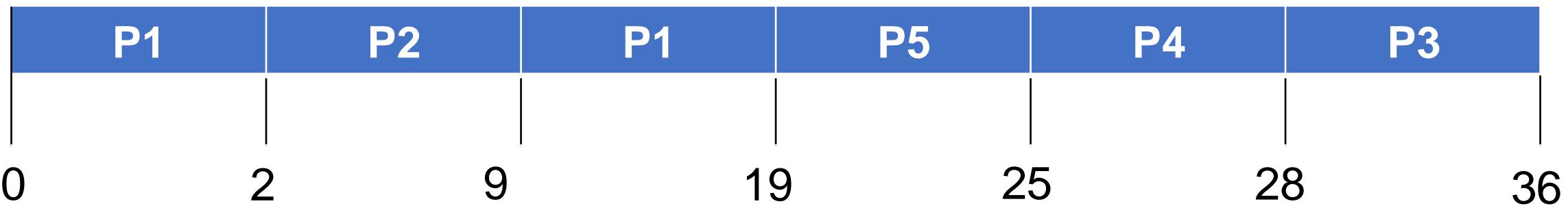
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 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 hoàn thành trung bình:  $(19+7+31+19+13)/5=17.8$



Sharing is learning

# 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 Robin**
- e) Highest-Response-Ratio-Next
- f) Multilevel Queue và Multilevel Feedback Queue



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### c) *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.



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### c) *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.

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



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### c) *Round Robin*

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.



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### c) Round Robin

(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

P1	P2	P1	P3	P2	P4	P5	P1	P3	P5
0	4	8	12	16	19	22	26	30	34
P2	P1	P3	P2	P4	P5	P1	P3	P5	
P1	P3	P2	P4	P5	P1	P3	P5		
P2	P4	P5	P1	P3	P5				
P5	P1	P3							
P1	P3								



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

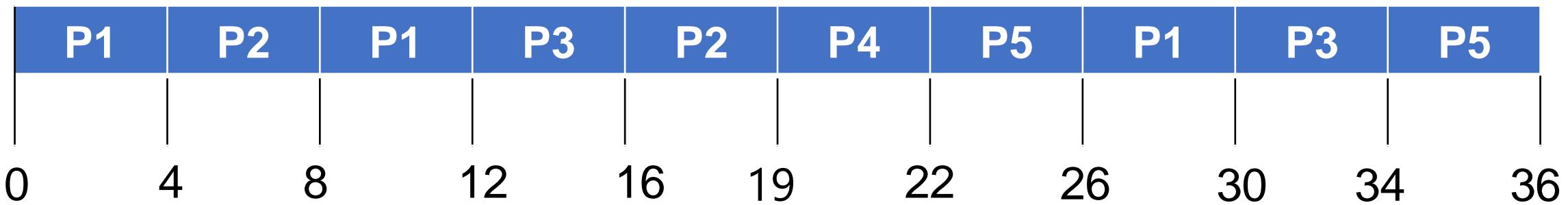
## 4. Các giải thuật định thời

### c) Round Robin

(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:  $(0+2+7+10+10)/5=5.8$



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

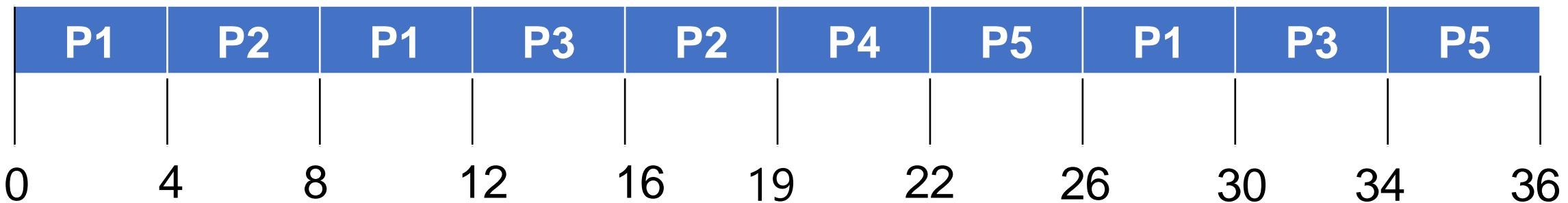
## 4. Các giải thuật định thời

### c) Round Robin

(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 đợi trung bình:  $(18 + 10 + 21 + 10 + 18) / 5 = 15.4$



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

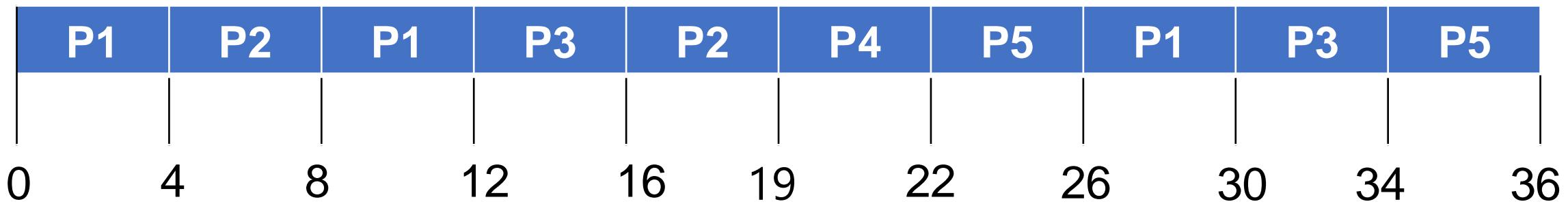
## 4. Các giải thuật định thời

### c) Round Robin

(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 hoàn thành trung bình:  $(30+17+29+13+24)/5=22.6$



Sharing is learning

# CHƯƠNG 4: ĐỊNH THỜI CPU

## 4. Các giải thuật định thời

### c) *Round Robin*

- ❑ 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 Robin
- e) **Highest-Response-Ratio-Next**
- f) Multilevel Queue và Multilevel Feedback Queue



Sharing is learning

# 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:

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



Sharing is learning

# 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 Robin
- e) Highest-Response-Ratio-Next
- f) **Multilevel Queue và Multilevel Feedback Queue**



Sharing is learning

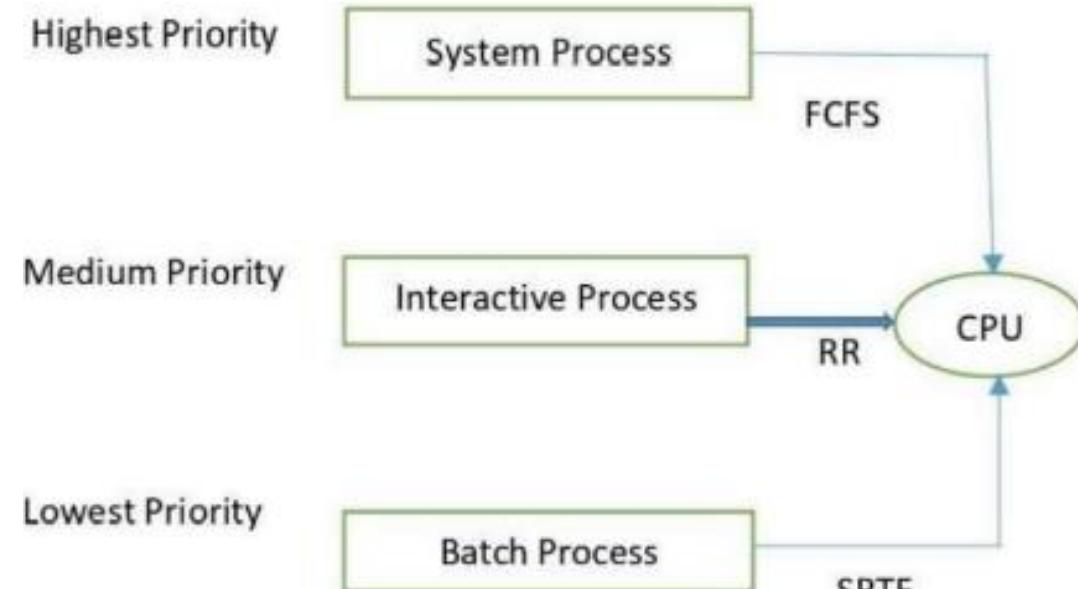
# CHƯƠNG 4: ĐỊNH THỜI CPU

## 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,...



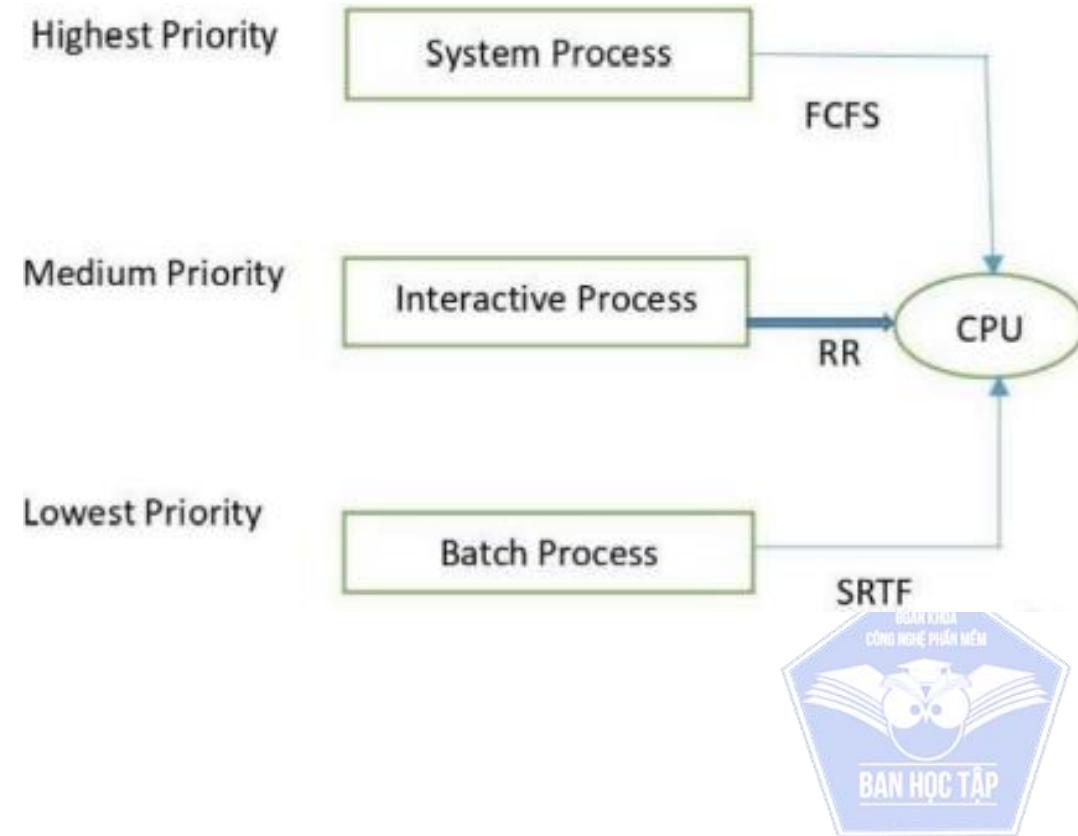
Sharing is learning

# 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 phõi 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

## 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



Sharing is learning

# 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 hàng đợi cao hoặc thấp hơn ?
- Khi process yêu cầu được xử lý thì hàng đợi nào là hợp lý nhất ?



Sharing is learning

# 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.



Sharing is learning

# 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)



Sharing is learning

# GIẢI ĐỀ THI THỬ



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

1. Hệ điều hành đơn nhiệm và hệ điều hành đa nhiệm được phân loại dưới góc độ nào?

- A. Loại máy tính
- B. Số chương trình được sử dụng cùng lúc.**
- C. Hình thức xử lý
- D. Số lượng người dùng truy xuất tài nguyên cùng lúc.



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

2. “Quyết định sẽ nạp chương trình nào khi có vùng nhớ trống” là chức năng của thành phần nào trong hệ điều hành?
- A. Quản lý tiến trình
  - B. Quản lý hệ thống lưu trữ thứ cấp
  - C. Quản lý hệ thống I/O
  - D. Quản lý bộ nhớ chính**



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

3. Thành phần nào sau đây không phải là một thành phần của hệ điều hành

- A. Quản lý hệ thống File
- B. Quản lý bộ nhớ
- C. **Quản lý kernel**
- D. Quản lý tiến trình



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

4. Đặc điểm của cấu trúc vi nhân là gì?

- A. Một số chức năng của hệ điều hành được chuyển từ kernel space sang user space.
- B. Hệ điều hành được chia thành nhiều lớp (layer).
- C. Sử dụng cách tiếp cận hướng đối tượng, mỗi core thành phần là tách biệt nhau.
- D. Các module trao đổi với nhau thông qua các interfaces.



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

5. Cấu trúc của một hệ thống máy tính gồm có những thành phần chính nào:

- A. Bộ vi xử lý (CPU), Bộ nhớ (Main Memory), Hệ điều hành (OS) và Các thiết bị nhập xuất (I/O devices)
- B. Phần cứng (Hardware), Hệ điều hành (OS), Bộ nhớ (Main Memory) và Các chương trình ứng dụng (Application Programs)
- C. Bộ vi xử lý (CPU), Bộ nhớ (Main Memory), Hệ điều hành (OS) và Các chương trình ứng dụng (Application Programs)
- D. Phần cứng (Hardware), Hệ điều hành (OS), Các chương trình ứng dụng (Application Programs) và Người dùng (Users)**

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

### 6. Chọn chuỗi trạng thái đúng với đoạn chương trình dưới đây:

```
int main(int argc, char** argv) {  
    int a, i = 2;  
    while (i <=5) {  
        i++;  
        if (i % 2 == 0)  
        {  
            a = 5*4;  
        } else {  
            a = 5*9;  
            printf ("Gia tri so a = %d", &a);  
        }  
    }  
    exit(0);  
}
```

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

### 6. Chọn chuỗi trạng thái đúng với đoạn chương trình dưới đây:

- A. new > ready > running > waiting > ready > running > waiting > terminated
- B. new > ready > running > waiting > ready > running > terminated
- C. **new > ready > running > waiting > ready > running > waiting > ready > running > terminated**
- D. new > ready > running > waiting > ready > running > waiting > ready > running > waiting > ready > running > terminated



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

7. Hệ điều hành nào dưới đây KHÔNG sử dụng kiến trúc vi nhân?

- A. MS-DOS
- B. Windows NT
- C. Tru64 UNIX
- D. Apple MacOS Server



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

8. Khi thực hiện giải thuật định thời Round Robin với 10 tiến trình đang nằm trong hàng đợi ready và time quantum = 9 ms thì thời gian lâu nhất mà một tiến trình có thể phải chờ đợi cho đến khi nó được đáp ứng là bao nhiêu?

- A. 10 ms
- B. **81 ms**
- C. 90 ms
- D. 100 ms



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

9. Tiến trình ở trạng thái ready khi nào?

- A. Tiến trình vừa được tạo.
- B. Tiến trình đã có đủ tài nguyên, chỉ còn cần CPU.**
- C. Tiến trình đã kết thúc.
- D. Các lệnh của tiến trình đang được thực thi.



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

10. Hệ điều hành điều phối tiến trình theo hướng vì lợi ích của hệ thống (System-oriented), tiêu chí nào KHÔNG thuộc hướng này?

- A. Sử dụng CPU (processor utilization): sao cho hiệu quả nhất.
- B. Công bằng (fairness) nhất đối với các tiến trình.
- C. **Thời gian chờ (Waiting time) sao cho dài nhất.**
- D. Thông lượng (throughput) tiến trình sao cho nhiều nhất



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

11. Chọn phát biểu SAI trong các phát biểu về giải thuật định thời bên dưới?

- A. **SJF là giải thuật định thời sử dụng độ ưu tiên với độ ưu tiên là thời-gian-sử-dụng-CPU-dự-đoán.**
- B. Không thể sử dụng giải thuật Round Robin nếu muốn các tiến trình có độ ưu tiên khác nhau.
- C. Khi thực hiện định thời theo độ ưu tiên, tiến trình có độ ưu tiên thấp có thể không bao giờ được thực thi.
- D. Giải thuật Multilevel Queue cho phép tiến trình di chuyển giữa các hàng đợi khác nhau.



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

12. Trong các bộ định thời sau, bộ định thời nào sẽ xác định process nào trong ready queue sẽ được chiếm CPU để thực thi kế tiếp:

- A. **Short-term Scheduler**
- B. Long-term Scheduler
- C. Medium-term Scheduler
- D. Tất cả đều có thể thực hiện được yêu cầu trên.



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

**13. Chọn phát biểu **SAI** trong các phát biểu sau đây?**

- A. Chuyển ngữ cảnh là quá trình CPU chuyển từ tiến trình này sang tiến trình khác.
- B. Chương trình trở thành tiến trình khi một tập tin thực thi được nạp vào bộ nhớ.
- C. Có ba phương pháp truyền tham số khi sử dụng lời gọi hệ thống: qua thanh ghi, qua một vùng nhớ và qua stack.
- D. Lời gọi hệ thống cung cấp giao diện giữa người dùng và hệ điều hành**



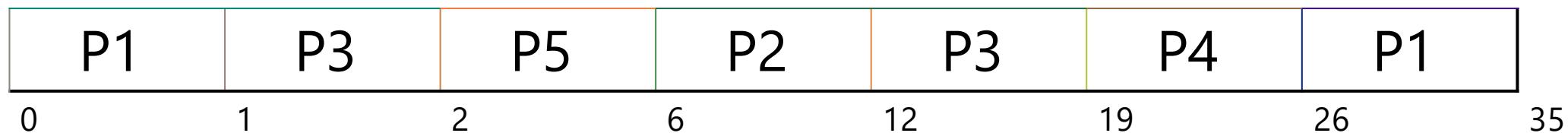
Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

Cho giản đồ GANTT và bảng thời gian chờ sau đây, trả lời câu hỏi 14 và 15:

**Gian đồ GANTT**



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

Cho giản đồ GANTT và bảng thời gian chờ sau đây, trả lời câu hỏi 14 và 15:

Bảng thời gian chờ

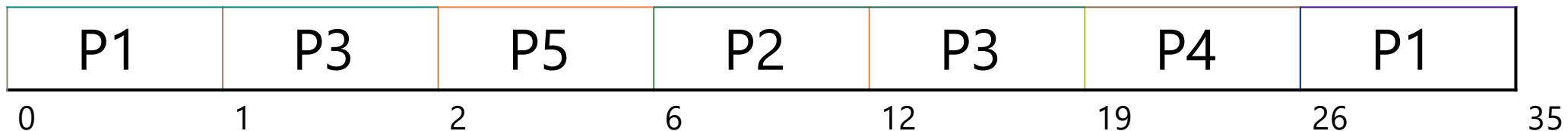
Process	Waiting time
P1	25
P2	2
P3	10
P4	12
P5	0



Sharing is learning

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm



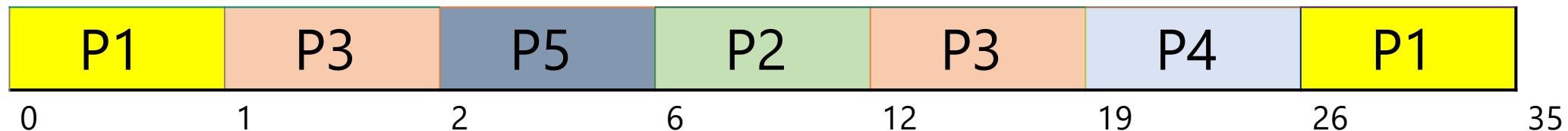
14. Biết giải thuật định thời sử dụng là SRTF, nếu dùng giải thuật FCFS đối với tập tiến trình trên thì tiến trình hoạt động vào giây thứ 20 là:

- A. P1
- B. P2
- C. P3
- D. P4
- E. P5

Process	Waiting time
P1	25
P2	2
P3	10
P4	12
P5	0

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm



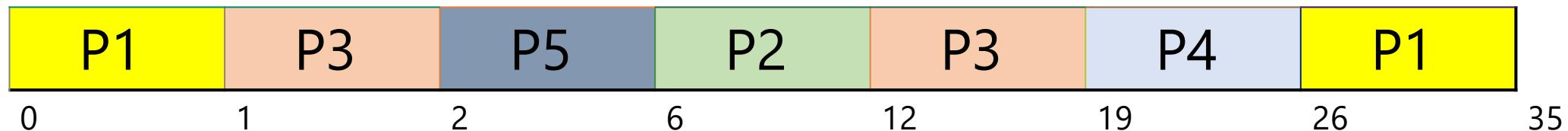
Burst time = Tổng các khoảng thời gian tiến trình được thực thi trên giản đồ

- P1:  $(1 - 0) + (35 - 26) = 10$
- P2:  $(12 - 6) = 6$
- P3:  $(2 - 1) + (19 - 12) = 8$
- P4:  $(26 - 19) = 7$
- P5:  $(6 - 2) = 4$

Process	Waiting time
P1	25
P2	2
P3	10
P4	12
P5	0

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm



Arrival time = Thời gian hoàn thành - thời gian thực thi - thời gian chờ

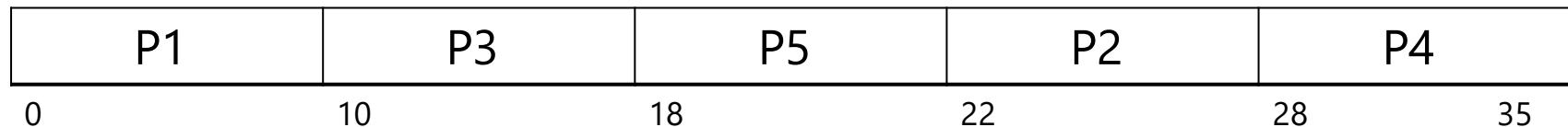
- P1:  $35 - 10 - 25 = 0$
- P2:  $12 - 6 - 2 = 4$
- P3:  $19 - 8 - 10 = 1$
- P4:  $26 - 7 - 12 = 7$
- P5:  $6 - 4 - 0 = 2$

Process	Waiting time
P1	25
P2	2
P3	10
P4	12
P5	0

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

**FCFS**



14. Biết giải thuật định thời sử dụng là SRTF, nếu dùng giải thuật FCFS đối với tập tiến trình trên thì tiến trình hoạt động vào giây thứ 20 là:

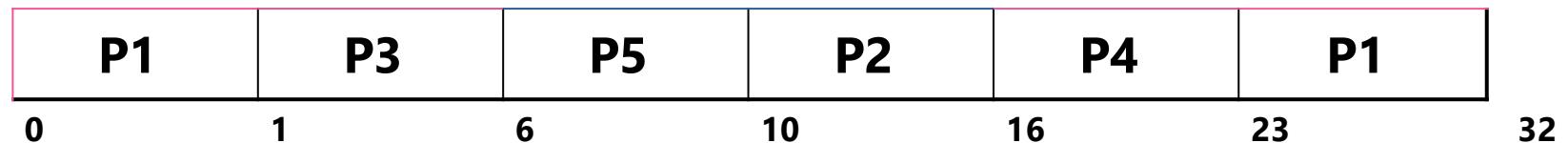
- A. P1
- B. P2
- C. P3
- D. P4
- E. **P5**

Process	Arrival Time	Burst - time
P1	0	10
P2	4	6
P3	1	8
P4	7	7
P5	2	4

# GIẢI ĐỀ THI THỬ

## A. Trắc nghiệm

SRTF



15. Nếu tiến trình P3 thực thi ít đi 03 giây, giản đồ GANTT cho giải thuật SRTF lúc này là:

Process	Arrival Time	Burst - time
P1	0	10
P2	4	6
P3	1	5
P4	7	7
P5	2	4

# GIẢI ĐỀ THI THỬ

## B. Tự luận

Cho 5 tiến trình P1, P2, P3, P4, P5 với thời gian vào Ready List và thời gian cần CPU tương ứng như bảng sau:

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 - thời gian hoàn thành) trung bình cho các giải thuật?

- First Come First Serve
- Priority (preemptive)
- Round Robin (quantum time = 4)

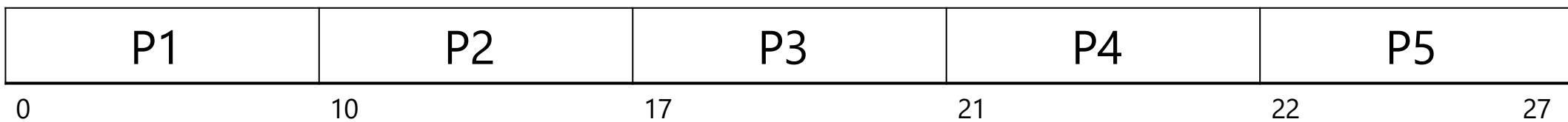
Process	Arrival Time	Burst - time	Priority
P1	0	10	5
P2	2	7	2
P3	3	4	3
P4	6	1	1
P5	7	5	4

# GIẢI ĐỀ THI THỬ

## B. Tự luận

### 1. FCFS

Process	Arrival Time	Burst - time	Priority
P1	0	10	5
P2	2	7	2
P3	3	4	3
P4	6	1	1
P5	7	5	4



Sharing is learning

# GIẢI ĐỀ THI THỬ

## B. Tự luận

### 1. FCFS

Process	Đáp ứng	Turnaround	Đợi
P1	0	10	0
P2	8	15	8
P3	14	18	14
P4	15	16	15
P5	15	20	15
Trung bình	10.4	15.8	10.4



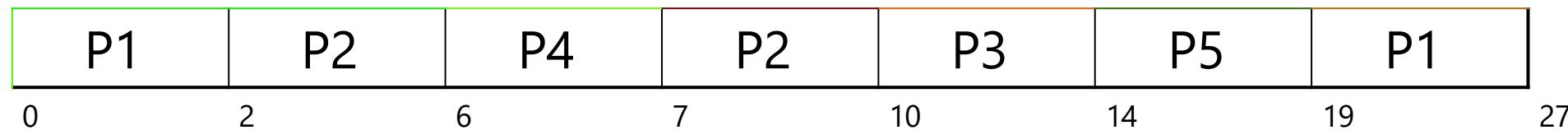
Sharing is learning

# GIẢI ĐỀ THI THỬ

## B. Tự luận

### 2. Priority (preemptive)

Process	Arrival Time	Burst - time	Priority
P1	0	10	5
P2	2	7	2
P3	3	4	3
P4	6	1	1
P5	7	5	4



Sharing is learning

# GIẢI ĐỀ THI THỬ

## B. Tự luận

### 2. Priority (preemptive)

Process	Đáp ứng	Turnaround	Đợi
P1	0	27	17
P2	0	8	1
P3	7	11	7
P4	0	1	0
P5	7	12	7
Trung bình	2.8	11.8	6.4



Sharing is learning

# GIẢI ĐỀ THI THỬ

## B. Tự luận

### 3. Round Robin ( $q = 4$ )

Process	Arrival Time	Burst - time	Priority
P1	0	10	5
P2	2	7	2
P3	3	4	3
P4	6	1	1
P5	7	5	4

P1	P2	P3	P1	P4	P5	P2	P1	P5
0	4	8	12	16	17	21	24	26
<b>P1 (10)</b>	<b>P2 (7)</b>	<b>P3(4)</b>	<b>P1(6)</b>	<b>P4(1)</b>	<b>P5(5)</b>	<b>P2(3)</b>	<b>P1(2)</b>	<b>P5(1)</b>
P3 (4)	P1(6)	P4(1)	P5(5)	P2(3)	P1(2)	P5(1)		
P1 (6)	P4(1)	P5(5)	P2(3)	P1(2)	P5(1)			
		P5(5)	P2(3)	P1(2)				
		P2(3)						



Sharing is learning

# GIẢI ĐỀ THI THỬ

## B. Tự luận

### 3. Round Robin ( $q = 4$ )

Process	Đáp ứng	Turnaround	Đợi
P1	0	26	16
P2	2	22	15
P3	5	9	5
P4	10	11	10
P5	10	20	15
Trung bình	5.4	17.6	12.2



Sharing is learning

# GIẢI ĐỀ THI THỬ

## B. Tự luận

Biết rằng khi chạy đoạn chương trình trên, tất cả các lệnh fork() đều được thực hiện thành công và chương trình kết thúc bình thường, hãy trả lời các câu hỏi sau đây

1. Có tất cả bao nhiêu tiến trình (kể cả cha) được tạo ra?
2. Có tất cả bao nhiêu từ "hello" được in ra?
3. Trong số các tiến trình in ra từ "hi", mỗi tiến trình chuyển sang trạng thái waiting bao nhiêu lần?
4. Có bao nhiêu giá trị khác nhau được in ra bởi câu lệnh in đậm?

```
int x = 10;  
int main()  
{ int i;  
printf("hi\n");  
int f = fork();  
if (f > 0)  
    printf("%d\n", f);  
for (i = 0; i < 2; i++) {  
    int k = fork();  
    printf("%d\n", k);  
    if (i == 0)  
        fork();  
    printf("hello\n"); }  
printf("bye\n");  
return 0; }
```

# GIẢI ĐỀ THI THỬ

## B. Tự luận

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

- Dòng 5: 2 tiến trình
- Dòng 9, i = 0: 4 tiến trình
- Dòng 12, i = 0: 8 tiến trình
- Dòng 9, i = 1: 16 tiến trình

Tổng cộng có 16 tiến trình được tạo ra

```
1. int x = 10;
2. int main()
3. { int i;
4.   printf("hi\n");
5.   int f = fork();
6.   if (f > 0)
7.     printf("%d\n", f);
8.   for (i = 0; i < 2; i++) {
9.     int k = fork();
10.    printf("%d\n", k);
11.    if (i == 0)
12.      fork();
13.    printf("hello\n"); }
14.   printf("bye\n");
15.   return 0; }
```

# GIẢI ĐỀ THI THỬ

## B. Tự luận

2. Có tất cả bao nhiêu từ "hello" được in ra?

- Vòng  $i = 0$ : 8 chữ
  - Vòng  $i = 1$ : 16 chữ
- => 24 từ "hello"

```
1. int x = 10;
2. int main()
3. { int i;
4.     printf("hi\n");
5.     int f = fork();
6.     if (f > 0)
7.         printf("%d\n", f);
8.     for (i = 0; i < 2; i++) {
9.         int k = fork();
10.        printf("%d\n", k);
11.        if (i == 0)
12.            fork();
13.        printf("hello\n"); }
14.    printf("bye\n");
15.    return 0; }
```

# GIẢI ĐỀ THI THỬ

## B. Tự luận

3. Trong số các tiến trình in ra từ "hi", mỗi tiến trình chuyển sang trạng thái waiting bao nhiêu lần?

- Có 1 tiến trình in ra từ "hi" - tiến trình đầu tiên
  - Khi fork(), đây là tiến trình cha => thực thi dòng 7
  - Trong 2 lần lặp: thực thi dòng 10 và 13
  - Thực thi dòng 16
- => Có 7 lần dùng I/O => waiting 7 lần

```
1. int x = 10;
2. int main()
3. { int i;
4.     printf("hi\n");
5.     int f = fork();
6.     if (f > 0)
7.         printf("%d\n", f);
8.     for (i = 0; i < 2; i++) {
9.         int k = fork();
10.        printf("%d\n", k);
11.        if (i == 0)
12.            fork();
13.        printf("hello\n"); }
14.    printf("bye\n");
15.    return 0; }
```

# GIẢI ĐỀ THI THỬ

## B. Tự luận

4. Có bao nhiêu giá trị khác nhau được in ra bởi câu lệnh in đậm?

Mỗi lần fork() trả về giá trị 0 cho tiến trình con và trả về ID của tiến trình con cho tiến trình cha của nó.

Vòng i = 0: Dòng 10 thực thi sau khi có 2 tiến trình mới được sinh ra => 2 giá trị mới

Vòng i = 1: Dòng 10 thực thi sau khi có 8 tiến trình mới được sinh ra => 8 giá trị mới

Thêm giá trị 0

=> Có 11 giá trị khác nhau được in ra bởi câu lệnh in đậm

```
1. int x = 10;  
2. int main()  
3. { int i;  
4.     printf("hi\n");  
5.     int f = fork();  
6.     if (f > 0)  
7.         printf("%d\n", f);  
8.     for (i = 0; i < 2; i++) {  
9.         int k = fork();  
10.        printf("%d\n", k);  
11.        if (i == 0)  
12.            fork();  
13.        printf("hello\n"); }  
14.    printf("bye\n");  
15.    return 0; }
```



Sharing is learning

# BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

## TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2023 – 2024



# 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](https://www.facebook.com/bhtcnpm)  
[fb.com/groups/bht.cnpm.uit](https://www.facebook.com/groups/bht.cnpm.uit)