



## Bài 3 : QUẢN LÝ TIẾN TRÌNH

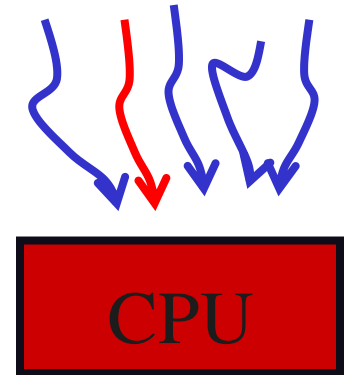
---

- **Phân chia CPU cho các tiến trình ?**
  - **Tiếp cận**
  - **Mục tiêu ?**
  - **Tổ chức ?**
  - **Chiến lược ?**
- **Trạng thái tiến trình ?**
- **Lưu trữ thông tin tiến trình ?**
- **Các thao tác trên tiến trình ?**
- **Bảo vệ tiến trình ?**
- **Trao đổi thông tin giữa các tiến trình ?**

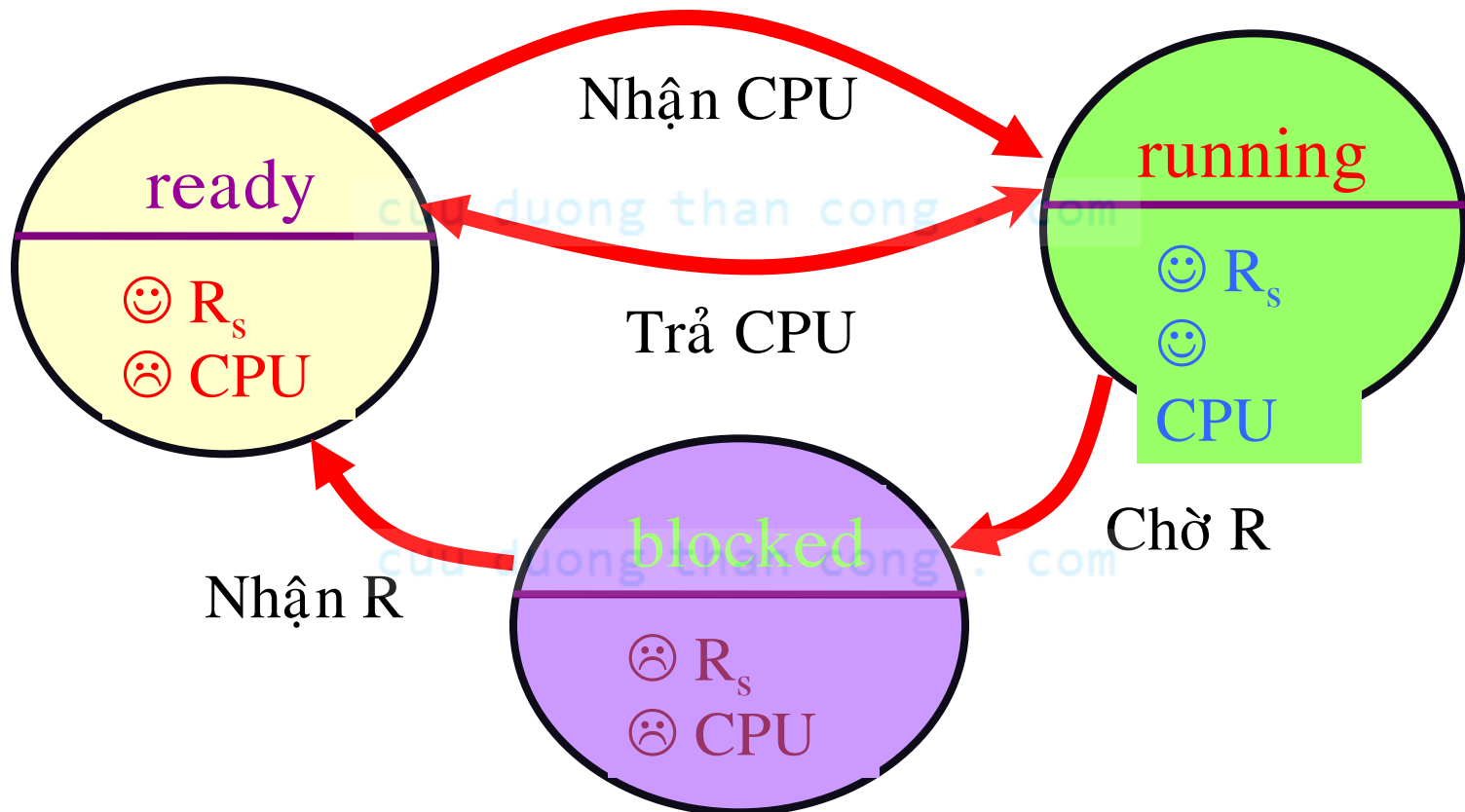
# Phân chia CPU ?

- **1 CPU vật lý** : làm thế nào để tạo ảo giác mỗi tiến trình sở hữu CPU riêng của mình ?
- **Dispatcher** luân chuyển CPU giữa các tiến trình:
  - Ngữ cảnh xử lý riêng biệt cho mỗi tiến trình (PCB)

```
while(1)
{
    interrupt  $P_{cur}$ 
    save state  $P_{cur}$ 
    Scheduler gets  $P_{next}$ 
    load state  $P_{next}$ 
    jump to it
}
```

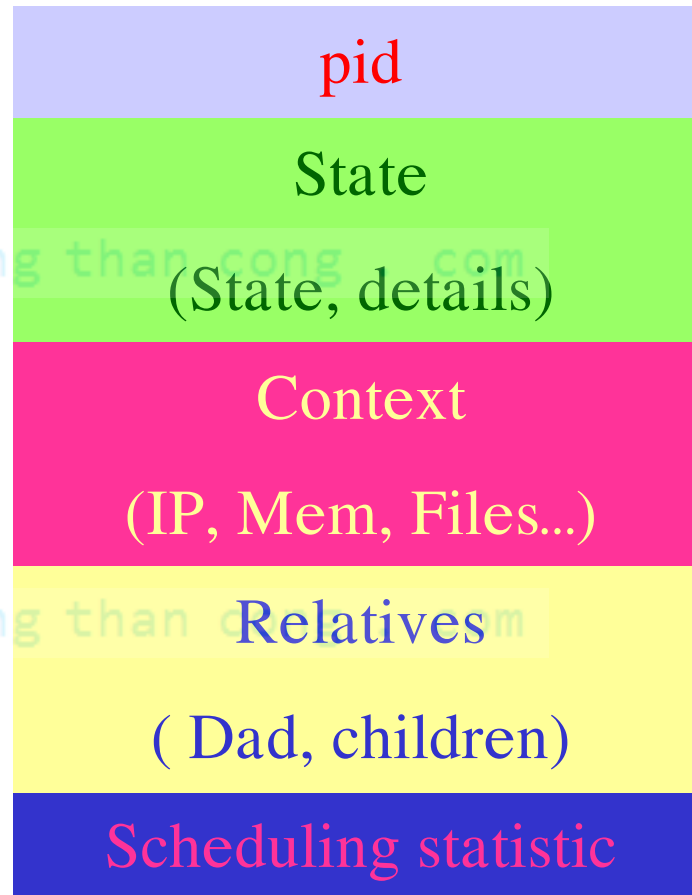


# Trạng thái tiến trình ?



# Khối quản lý tiến trình trong mô hình multiprocesses

Process control Block  
PCB

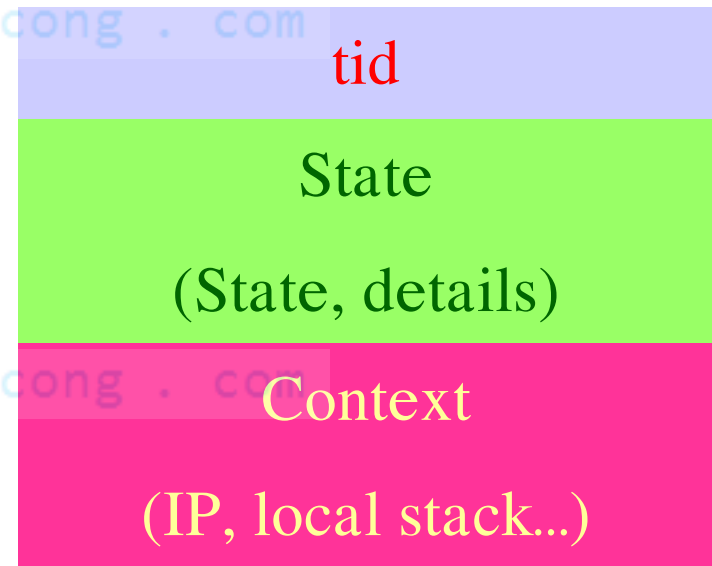


# PCB và TCB trong mô hình multithreads

PCB



Thread Control Block  
TCB





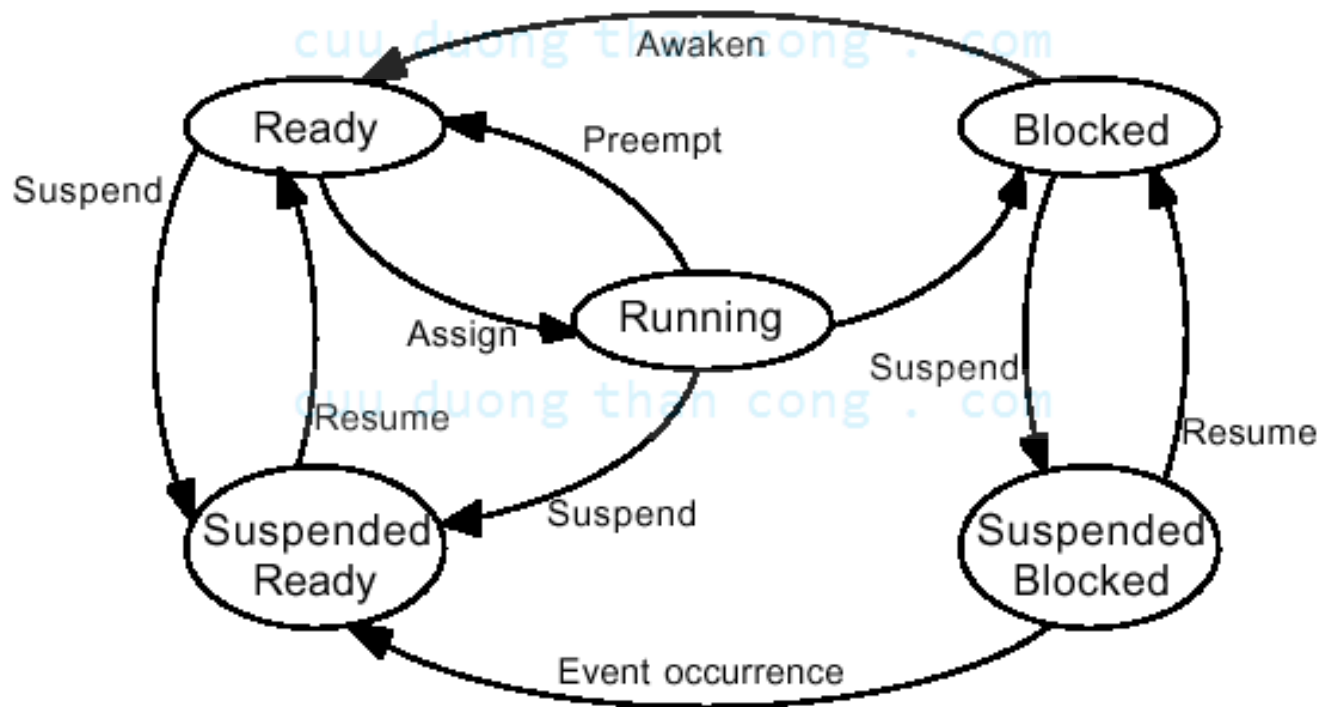
## Các thao tác trên tiến trình

---

- **Tạo lập tiến trình :**
  - **Cấp phát tài nguyên cho tiến trình con ?**
  - **Hoạt động của cha và con độc lập**
- **Kết thúc tiến trình :**
  - **Thu hồi tài nguyên ?**
  - **Ép buộc kết thúc ?**
- **Thay đổi trạng thái tiến trình :**  
**Assign(), Block(), Awake(), Resume(), Suspend()**

# Trạng thái tiến trình ?

- Có nhu cầu Suspend & Resume :
  - Hệ thống quá tải
  - Kiểm soát hoạt động của tiến trình con





## **An ninh trật tự cho môi trường đa tiến trình !**

---

- **Bảo vệ tiến trình :**
  - **Ngăn cản các tiến trình xâm phạm tài nguyên, can thiệp vào xử lý của nhau => KGĐC riêng biệt, 2 mode xử lý**
  - **Bảo đảm quyền tiến triển xử lý cho mỗi tiến trình => công bằng trong các chiến lược phân phối tài nguyên.**
- **Trao đổi thông tin , phối hợp hoạt động ?**
  - **Nhu cầu ?**
  - **Vấn đề ?      => Chương kế tiếp**
  - **Giải pháp ?**

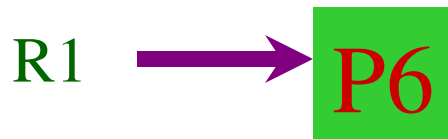
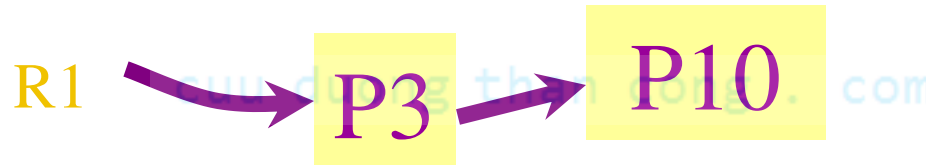


## Các danh sách tiến trình

Ready List



Waiting Lists





# Điều phối tiến trình

---

- Mục tiêu ?
- Các cấp độ điều phối
- Thời điểm ra quyết định điều phối ?
- Đánh giá chiến lược điều phối ?
- Một số chiến lược điều phối



## Điều phối tiến trình

---

**SCHEDULER**

chọn một tiến trình  
nhận cpu

**DISPATCHER**

chuyển đổi ngữ  
cảnh



## Chuyển đổi ngữ cảnh (context switching)

---

- **Kịch bản :**
  - Lưu ngữ cảnh tiến trình hiện hành
  - Nạp ngữ cảnh tiến trình được chọn kế tiếp
- **Chi tiết cụ thể phụ thuộc vào phần cứng**
  - general-purpose & floating point registers, co-processor state...
- **Chi phí chuyển đổi ngữ cảnh :**
  - Giữa các tiến trình ?
  - Giữa các tiểu trình ?

# Chuyển đổi ngữ cảnh giữa các tiến trình



- Chuyển đổi mode xử lý
- Chuyển đổi IP và các thanh ghi khác của CPU
- Chuyển đổi không gian địa chỉ

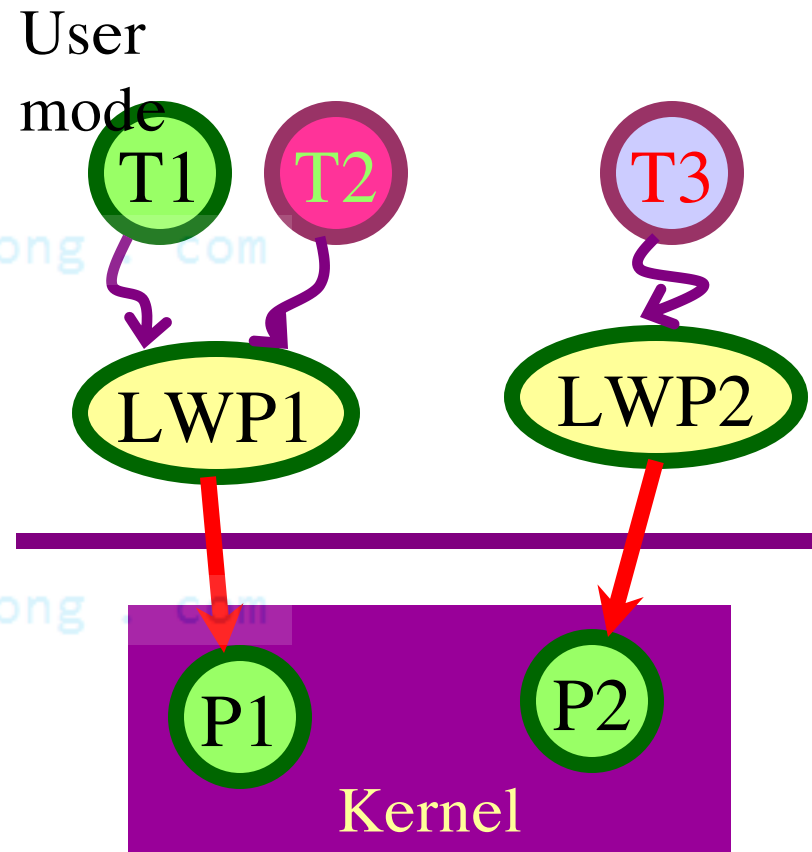
## Tiểu trình hạt nhân (Kernel thread)



- Khái niệm tiểu trình được xây dựng bên trong hạt nhân
- Dispatcher làm việc với đơn vị là tiểu trình

## Tiểu trình người dùng (User thread)

- Khái niệm tiểu trình được hỗ trợ bởi một thư viện hoạt động trong user mode
- Dispatcher của hạt nhân làm việc với đơn vị là tiến trình
- ThreadDispatcher làm việc với đơn vị là tiểu trình
  - $P \leftrightarrow LWP - T$
- Không cần chuyển đổi chế độ xử lý khi chuyển đổi các tiểu trình cùng thuộc 1 tiến trình.



Kernel  
mode



## Lựa chọn tiến trình ?

---

- **Tác vụ của Scheduler**
- **Mục tiêu ?**
  - Sử dụng CPU hiệu quả
  - Đảm bảo tất cả các tiến trình đều tiến triển xử lý
- **Tiêu chuẩn lựa chọn ?**
  - Tất cả các tiến trình đều như nhau ?
  - Đề xuất một độ ưu tiên cho mỗi tiến trình ?
- **Thời điểm lựa chọn ? (Thời điểm kích hoạt Scheduler())**





## Mục tiêu điều phối

---

- **Hiệu quả (Efficiency)**

- ↓ **Thời gian**

- ↓ **Đáp ứng (Response time)**

- ↓ **Hoàn tất (Turnaround Time =  $T_{\text{quit}} - T_{\text{arrive}}$ ):**

- ↓ **Chờ (Waiting Time =  $T_{\text{in Ready}}$ ) :**

- ↑ **Thông lượng (Throughput = # jobs/s )**

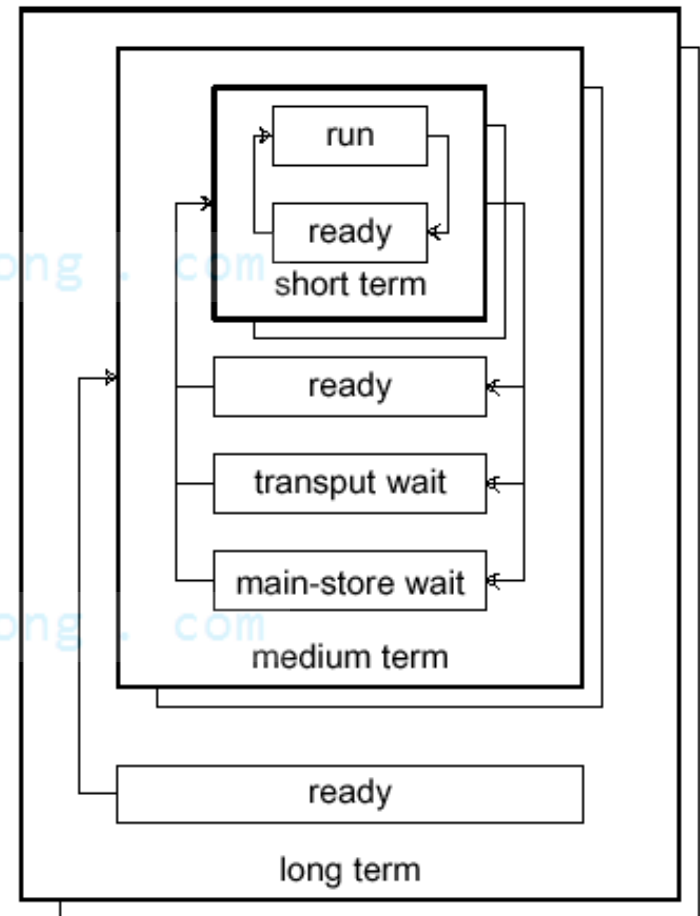
- ↑ **Hiệu suất Tài nguyên**

- ↓ **Chi phí chuyển đổi**

- **Công bằng ( Fairness) : Tất cả các tiến trình đều có cơ hội nhận CPU**

## Các cấp độ điều phối

- **Longterm scheduling** : chọn tiến trình kế tiếp được khởi động (mang vào bộ nhớ và nhận trạng thái ready)
- **Mediumterm scheduling** : quyết định chuyển tiến trình đang running sang trạng thái blocked.
- **Shortterm scheduling** : chọn 1 tiến trình ở trạng thái ready để chuyển sang trạng thái running.





## Thời điểm ra quyết định điều phối

---

- **Điều phối độc quyền (non-preemptive scheduling):** tiến trình được chọn độc chiếm CPU
- **Điều phối không độc quyền (preemptive scheduling):** tiến trình được chọn có thể bị « cướp » CPU bởi tiến trình có độ ưu tiên cao hơn



## Các chiến lược điều phối

---

- **FIFO**
- **RR**
- **SJF**
- **MULTILEVELFEEDBACK**
- **LOTTERY**

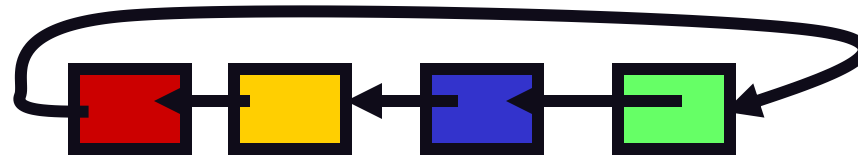
# FIFO – RR -SJF

- FIFO



cuu duong than cong . com

- RR

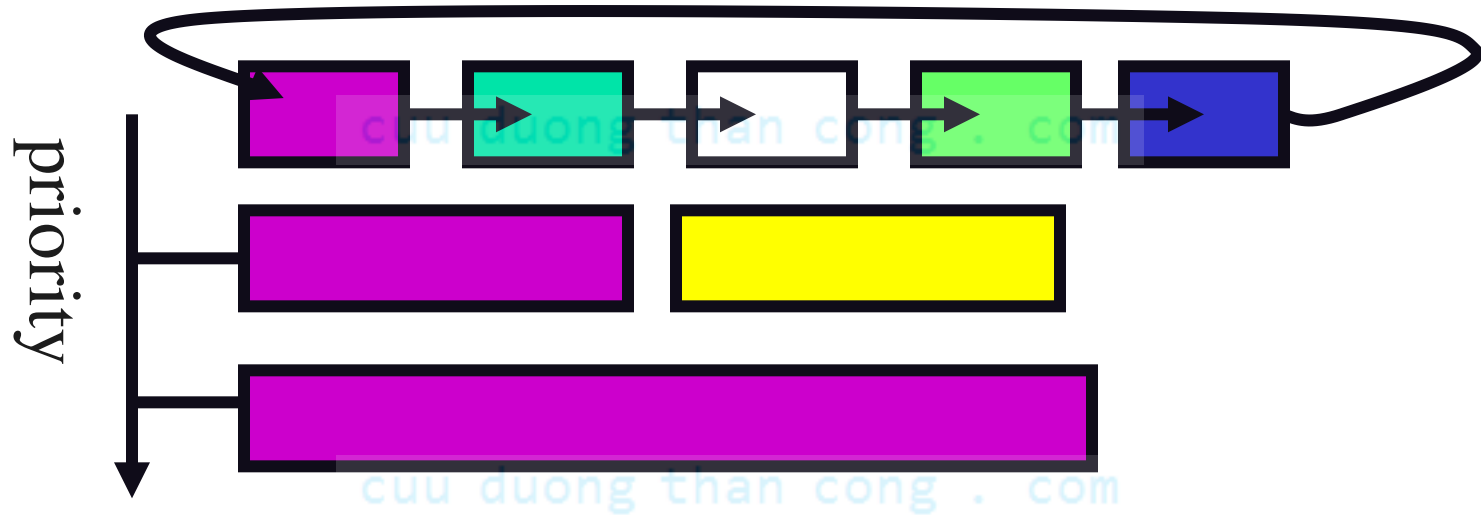


cuu duong than cong . com

- SJF



# Multilevel Feedback





# Lottery

---

1

1

1

1

P1

P2

P3

P4

P2 có 25 % cơ hội

1

7

1

1

P1

P2

P3

P4

P2 có 70 % cơ hội