MÔN HỌC: HỆ ĐIỀU HÀNH

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 4

1. Tại sao phải định thời? Có những loại bộ định thời nào?

Trả lời:

Mục tiêu của việc lập trình đa luồng là hướng đến việc luôn luôn phải có tiến trình sử dụng CPU, hay nói cách khác, là tối đa hoá việc sử dụng CPU. Ngoài ra, mục tiêu của các hệ thống time sharing (chia sẻ thời gian – tức các hệ thống HĐH hiện nay) là việc mang đến cho người dùng cảm giác chiếc máy tính của mình có thể làm được nhiều công việc cùng một lúc. Việc đó chỉ có thể đạt được thông qua việc chuyển quyền sử dụng CPU thật nhanh qua lại giữa các tiến trình.

Và để đạt được các mục tiêu nêu trên, **trình định thời** (**Scheduler**) sẽ lựa chọn trong các tiến trình hiện có để thực thi trên CPU (do trong một thời điểm nhất định, chỉ duy nhất có một tiến trình được quyền ở trạng thái running mà thôi) .

Có 3 bộ định thời :

- Short-Term Scheduling (hay còn gọi là Dispatcher) : Dùng để định thời cho CPU.
    - Xác định process nào trong ready queue sẽ được chiếm CPU để thực thi kế tiếp.
    - Bộ định thời Short-Term sẽ 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.
- Medium-Term Scheduling : Dùng để định thời Swaping.

    - 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.
- Long-Term Scheduling (hay còn gọi là Job Scheduler) :
    - 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 Scheduling thường cố gắng duy trì xen lẫn CPU-Bound và I/O Bound Process.

2. Định thời CPU là gì? Bộ định thời nào chịu trách nhiệm thực hiện việc này?

Trả lời:

Định thời CPU là một khái niệm quan trọng trong hệ điều hành đa chương trình, giúp làm máy tính hoạt động hiệu quả hơn. Với định thời CPU, quá trình nào sẽ được thực thi tiếp theo được xác định, giúp tối ưu hóa sử dụng tài nguyên CPU.
Bộ định thời ngắn hạn (short-term scheduler) chịu trách nhiệm thực hiện việc định thời CPU[2]. Bộ định thời này chọn các tiến trình trong bộ nhớ sẵn sàng thực thi và cấp phát CPU tới một trong các tiến trình đó

3. Phí tổn gây ra khi định thời là gì?

Trả lời:

Phí tổn gây ra khi định thời là những chi phí liên quan đến việc chuyển đổi giữa các tiến trình trong hệ thống máy tính. Khi một tiến trình được chọn để chạy, hệ điều hành cần lưu trữ trạng thái hiện tại của tiến trình đang chạy và khôi phục trạng thái của tiến trình mới. Việc này đòi hỏi một lượng thời gian và tài nguyên nhất định, do đó tạo ra chi phí.

4. Trình bày các tiêu chuẩn định thời CPU?

Trả lời:

Các thuật toán định thời CPU khác nhau có các tính chất khác nhau. Vì vậy, rất nhiều tiêu chí đã được đề ra để so sánh các thuật toán định thời CPU. Các tiêu chuẩn được liệt kê như sau :

- **Mức độ sử dụng CPU (CPU Utilization) :** Chúng ta muốn giữ làm sao cho CPU càng bận càng tốt. -> cực đại
- **Thông lượng (Throughput) :** Thông lượng được tính bằng số lượng tiến trình đã được hoàn thành trong một đơn vị thời gian. -> cực đại
- **Thời gian hoàn thành (Turnaround time) :** Khoảng thời gian từ lúc tiến trình được ghi nhận đến khi hoàn thành chính là **Turnaround Time.** -> cực tiểu
- **Thời gian đợi (Waiting time) :** Tổng thời gian tiến trình đã ở trong hàng đợi ready queue. -> cực tiểu
- **Thời gian đáp ứng (Response Time) :** Thời gian từ lúc tiến trình xuất hiện cho đến khi thực hiện tiến trình đó lần đầu tiên. -> cực tiểu
- **Tính công bằng (fairness) :** tất cả tiến trình phải được đối xử như nhau.

5. Kể tên các giải thuật định thời CPU?

Trả lời:

- First-Come, First-Served (FCFS).
- Shortest-Job-First Scheduling (SJF).
- Preemptive SJF (hay Shortest-Remaining-Time First – SRTF).
- Priority Scheduling.
- Round-Robin (RR).
- Highest Response Ratio Next (HRRN).
- Multilevel Queue.
- Multilevel Feedback Queue.

6. **Mô tả và nêu ưu điểm, nhược điểm của từng giải thuật định thời sau: FCFS, SJF, SRTF, RR, Priority Scheduling, HRRN, MQ, MFQ.**

Trả lời:

- FCFS (First Come First Serve):
  - Mô tả: Giải thuật đơn giản và dễ hiểu nhất, xử lý các tiến trình theo thứ tự chúng đến.
  - Ưu điểm: Cung cấp sự công bằng bằng cách đối xử với tất cả các tiến trình một cách công bằng và cho chúng cơ hội chạy.
  - Nhược điểm: Tiến trình có thời gian thực thi ít hơn thường phải chờ đợi lâu.
- SJF (Shortest Job First):
  - Mô tả: Ưu tiên các công việc ngắn hơn.
  - Ưu điểm: Có lẽ là tối ưu, vì nó mang lại thời gian chờ trung bình tối thiểu cho một tập hợp các tiến trình.
  - Nhược điểm: SJF có thể gây ra tình trạng đói nếu các tiến trình ngắn hơn không ngừng đến.
- SRTF (Shortest Remaining Time First):
  - Mô tả: Tiến trình có thời gian hoàn thành ngắn nhất được chọn để thực thi đầu tiên.
  - Ưu điểm: Tạo ra thời gian chờ trung bình tối thiểu.
  - Nhược điểm: Không thể triển khai được vì thời gian burst của các tiến trình không thể biết trước.
- RR (Round Robin):
  - Mô tả: Mỗi tiến trình nhận được một phần CPU. RR có tính chu kỳ, vì vậy không có tình trạng đói.
  - Ưu điểm: Mỗi tiến trình nhận được một phần CPU. RR có tính chu kỳ, vì vậy không có tình trạng đói.
  - Nhược điểm: Đặt quantum quá ngắn sẽ làm tăng overhead và làm giảm hiệu suất CPU, nhưng đặt nó quá dài có thể gây ra phản ứng kém đối với các tiến trình ngắn.
- Priority Scheduling:
  - Mô tả: Cung cấp một cơ chế tốt để xác định rõ ràng mức độ quan trọng của mỗi tiến trình.
  - Ưu điểm: Cho phép gán các mức ưu tiên khác nhau cho các tiến trình dựa trên mức độ quan trọng, khẩn cấp hoặc các tiêu chí khác.
  - Nhược điểm: Nếu các tiến trình ưu tiên cao sử dụng nhiều thời gian CPU, các tiến trình ưu tiên thấp hơn có thể bị đói và bị hoãn lại vô thời hạn[1].

- HRRN (Highest Response Ratio Next):
    - Mô tả: Giải thuật này luôn chọn tiến trình có tỷ lệ phản hồi cao nhất để thực thi.
    - Ưu điểm: Thực hiện tốt hơn SJF. Nó không chỉ ưu tiên các công việc ngắn hơn mà còn giới hạn thời gian chờ của các công việc dài hơn.
    - Nhược điểm: Không thể triển khai được vì thời gian burst của các tiến trình không thể biết trước.
- MQ (Multilevel Queue Scheduling):
    - Mô tả: Cho phép áp dụng lịch biểu riêng biệt cho các loại tiến trình khác nhau.
    - Ưu điểm: Overhead lịch biểu thấp. Cho phép aging, do đó không có tình trạng đói.
    - Nhược điểm: Không linh hoạt. Nó cũng yêu cầu một số phương tiện để chọn các giá trị cho tất cả các thông số để xác định lịch biểu tốt nhất, do đó nó cũng là thuật toán phức tạp nhất.
- MFQ (Multilevel Feedback Queue Scheduling):
    - Mô tả: Tương tự như MLQ, nhưng trong MFQ, các tiến trình có thể di chuyển giữa các hàng đợi.
    - Ưu điểm: Overhead lịch biểu thấp. Cho phép aging, do đó không có tình trạng đói.
    - Nhược điểm: Không linh hoạt. Nó cũng yêu cầu một số phương tiện để chọn các giá trị cho tất cả các thông số để xác định lịch biểu tốt nhất, do đó nó cũng là thuật toán phức tạp nhất.

7. Đặc điểm của định thời trên hệ thống có nhiều bộ xử lý? Khi nào cần phải thực hiện cân bằng tải?

Trả Lời:
Định thời trên hệ thống có nhiều bộ xử lý có các đặc điểm sau[1][2]:

- Xử lý các công việc thực sự đồng thời[2].
- Mỗi bộ xử lý có bộ nhớ riêng[2].
- Mỗi bộ xử lý có đường truyền dữ liệu riêng[2].

Cân bằng tải là một phương pháp phân phối khối lượng tải trên nhiều máy tính hoặc một cụm máy tính để có thể sử dụng tối ưu các nguồn lực, tối đa hóa thông lượng, giảm thời gian đáp ứng và tránh tình trạng quá tải trên máy chủ[3]. Cần thực hiện cân bằng tải khi:

- Tăng khả năng đáp ứng[4].

- Tránh tình trạng quá tải trên máy chủ[4].
- Đảm bảo tính linh hoạt và mở rộng cho hệ thống[4].
- Tăng độ tin cậy và khả năng dự phòng cho hệ thống[4].
- Tăng tính bảo mật cho hệ thống[4].

8. Đặc điểm định thời theo thời gian thực?

Trả lời:

- **Tính bị động**: Hệ thống phản ứng dựa trên các sự kiện từ môi trường ngoại vi.
- **Tính nhanh nhạy**: Hệ thống phản ứng nhanh chóng đối với các sự kiện.
- **Tính đồng thời**: Hệ thống có khả năng xử lý nhiều tác vụ cùng một lúc.
- **Tính tiền định**: Hành vi của hệ thống có thể dự đoán được trước.
- **Độ trễ ngắt tối thiểu và độ trễ chuyển luồng tối thiểu**: Một hệ điều hành thời gian thực được đánh giá cao hơn về mức độ nhanh nhạy hoặc có thể dự đoán được nó như thế nào so với số lượng công việc nó có thể thực hiện trong một khoảng thời gian nhất định.

9. Mô tả các đặc điểm cơ bản của bộ định thời CFS trên Linux?

Trả lời:

Bộ định thời CFS (Completely Fair Scheduler) trên Linux có các đặc điểm cơ bản sau:

- **Công bằng**: CFS đảm bảo sự công bằng giữa tất cả các tiến trình thông thường trong hệ thống.
- **Phân loại tiến trình**: Linux phân loại tiến trình dựa theo đặc điểm hoạt động của tiến trình theo một trong hai loại: I/O-Bound process và Processor-Bound process.
    - **I/O-Bound process**: Là tiến trình sử dụng phần lớn thời gian cho việc chờ các hoạt động input/output.
    - **Processor-Bound process**: Là tiến trình sử dụng phần lớn thời gian của CPU để thực thi mã lệnh.
- **Tiến trình Real-time**: Trong quá trình phát triển, Linux kernel hỗ trợ tiến trình real-time, là các tiến trình yêu cầu đáp ứng lập lịch một cách nghiêm ngặt để đảm bảo xử lý và tốc độ phản hồi ở mức độ thời gian thực.

- **Tiến trình Normal:** Là các tiến trình không đòi tốc độ phản hồi real-time, nên có độ ưu tiên thấp hơn tất cả các tiến trình real-time trong hệ thống. Một normal process được chia thành 2 loại sau:
    - **Interactive Process:** Là các tiến trình tương tác với người dùng.
    - **Batch Process:** Là các tiến trình không tương tác với người dùng.

10. Mô tả các đặc điểm cơ bản của định thời trên Windows?

Trả lời:

Định thời trên hệ điều hành Windows có các đặc điểm cơ bản sau[1][2][3]:

- **Quản lý chia sẻ tài nguyên:** Windows quản lý việc chia sẻ các tài nguyên của hệ thống như bộ nhớ, CPU, thiết bị ngoại vi[3].
- **Giả lập máy tính mở rộng:** Windows ẩn các chi tiết phần cứng và cung cấp cho người dùng giao diện đơn giản, dễ sử dụng và hoàn toàn không bị lệ thuộc vào thiết bị phần cứng[3].
- **Đa chương:** Bằng cách chuyển đổi CPU giữa các quá trình, hệ điều hành có thể làm máy tính hoạt động nhiều hơn[2].
- **Bộ định thời CPU:** Bất cứ khi nào CPU rảnh, hệ điều hành phải chọn một trong những quá trình trong hàng đợi sẵn sàng để thực thi[2].
- **Chu kỳ CPU-I/O:** Sự thành công của việc định thời biểu CPU phụ thuộc vào thuộc tính được xem xét sau đây của quá trình. Việc thực thi quá trình chứa một chu kỳ (cycle) thực thi CPU và chờ đợi nhập/xuất[2].

11. (Bài tập mẫu) Cho các tiến trình với thông tin ở bảng bên dưới. Biết rằng tất cả các tiến trình đều đến ở thời điểm 0 theo thứ tự từ P1 đến P5. Vẽ giản đồ Gantt, tính thời gian đợi trung bình và thời gian lưu lại trong hệ thống (turnaround time) trung bình cho các giải thuật sau:

a. **FCFS**

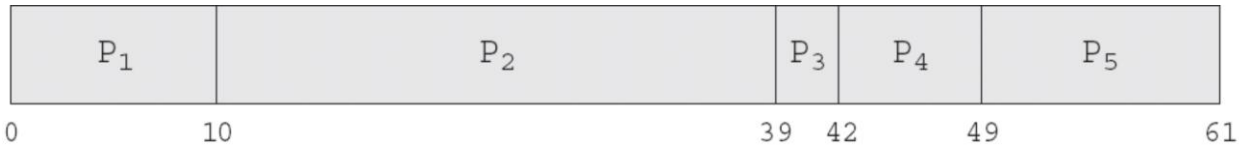
b. **SJF**

c. **RR với quantum time = 10**

Process	Burst Time
P1	10

P2	29
P3	3
P4	7
P5	12

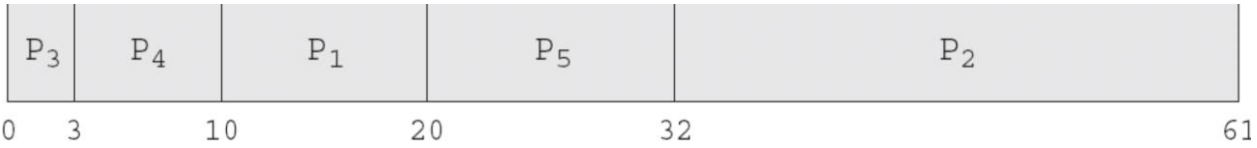
FCFS:



Thời gian đợi trung bình: (0 + 10 + 39 + 42 + 49)/5 = 28

Thời gian lưu lại trong hệ thống trung bình: (10 + 39 + 42 + 49 + 61)/5 = 40.2

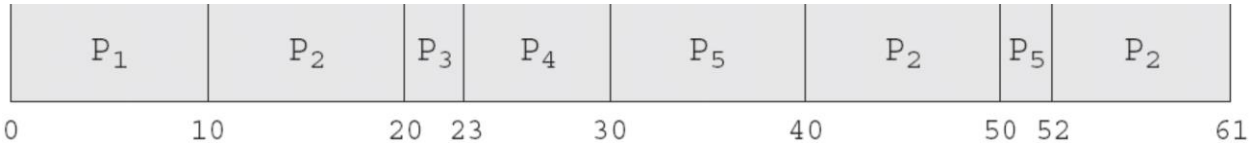
SJF



Thời gian đợi trung bình: (10 + 32 + 0 + 3 + 20)/5 = 13

Thời gian lưu lại trong hệ thống trung bình: (20 + 61 + 3 + 10 + 32)/5 = 25.2

RR với quantum time = 10



Thời gian đợi trung bình: (0 + (10 + 20 + 2) + 20 + 23 + (30 + 10))/5 = 23

Thời gian lưu lại trong hệ thống trung bình: (10 + 61 + 23 + 30 + 52)/5 = 35.2

12. Cho 5 tiến trình P1, P2, P3, P4, P5 với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau:

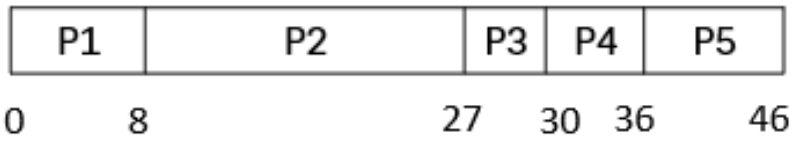
Process	Arrival Time	CPU Burst Time
P1	0	8
P2	2	19
P3	4	3
P4	5	6
P5	7	10

Vẽ sơ đồ Gantt và tính thời gian chờ trung bình, thời gian đáp ứng trung bình, thời gian lưu lại trong hệ thống (turnaround time) trung bình cho các giải thuật sau:

a. FCFS
b. SJF preemptive
c. RR với quantum time = 6.

a. FCFS:

Giản đồ Gantt



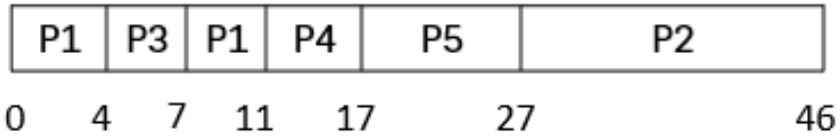
Thời gian chờ trung bình : (0 + 6 + 23 + 25 + 29)/5 = 16,6

Thời gian đáp ứng trung bình: (0 + 6 + 23 + 25 + 29)/5 = 16,6

Thời gian lưu lại trong hệ thống: (8 + 25 + 26 + 31 + 39)/5 = 25.8

b. SJF preemptive:

Giản đồ Gantt



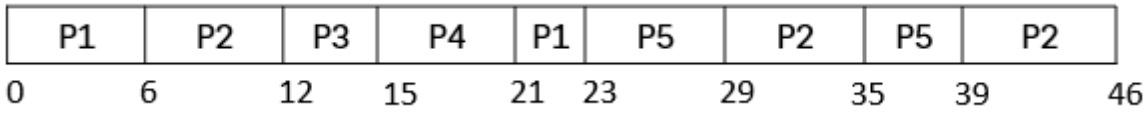
Thời gian chờ trung bình : (3+ 25 + 0 + 6 + 10)/5 = 8,8

Thời gian đáp ứng trung bình: (0 + 25 + 0 + 6 + 10)/5 = 8,2

Thời gian lưu lại trong hệ thống: (11 + 44 + 3 + 12 + 20)/5 = 18

c. RR với quantum time = 6:

Giản đồ Gantt



Thời gian chờ trung bình : (15 + 25 + 8 + 10 + 22)/5 = 16

Thời gian đáp ứng trung bình: (0 + 4 + 8 + 10 + 16)/5 = 7,6

Thời gian lưu lại trong hệ thống: (23 + 44 + 11 + 16 + 32)/5 = 25,2

13. (Bài tập mẫu) Cho 5 tiến trình P1, P2, P3, P4, P5 với thời gian vào hàng đợi ready và thời gian cần CPU tương ứng như bảng sau:

Process	Arrival Time	Burst Time
P1	0	13
P2	4	9

P3	6	4
P4	7	20
P5	12	10

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình, thời gian lưu lại trong hệ thống (turnaround time – thời gian hoàn thành) trung bình khi thực hiện các giải thuật định thời sau:

a) **Round Robin với quantum time = 5**

b) **SRTF**

Có nhận xét gì về tính hiệu quả của hai giải thuật trên?

a. Round Robin với quantum time = 5

Giản đồ Gantt:

P1	P2	P1	P3	P4	P2	P5	P1	P4	P5	P4

0 5 10 15 19 24 28 33 36 41 46 56

Thời gian đáp ứng trung bình: (0 + 1 + 9 + 12 + 16)/5 = 7.6

Thời gian đợi trung bình: ((5 + 18) + (1 + 14) + 9 + (12 + 12 + 5) + (16 + 8))/5 = 20

Thời gian hoàn thành trung bình: (36 + 24 + 13 + 49 + 34)/5 = 31.2

b. SRTF

Giản đồ Gantt:

P1	P3	P1	P2	P5	P4

0 6 10 17 26 36
56

Thời gian đáp ứng trung bình: (0 + 13 + 0 + 29 + 14)/5 = 11.2

Thời gian đợi trung bình: (4 + 13 + 0 + 29 + 14)/5 = 12

Thời gian hoàn thành trung bình: (17 + 22 + 4 + 49 + 24)/5 = 23.2

Nhận xét về hai giải thuật trên:

- SRTF hiệu quả hơn (tốt hơn) Round Robin nếu xét trên các tiêu chuẩn thời gian đợi (trung bình) và thời gian hoàn thành (trung bình).
- Round Robin cho thời gian đáp ứng (trung bình) tốt hơn SRTF.

14. (Bài tập mẫu) Cho 5 tiến trình P1, P2, P3, P4, P5 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	13	4
P2	4	9	3
P3	6	4	1
P4	7	17	2
P5	12	9	5

Vẽ giản đồ Gantt và tính thời gian đợi trung bình, thời gian đáp ứng trung bình, thời gian lưu lại trong hệ thống (turnaround time – thời gian hoàn thành) trung bình khi thực hiện giải thuật định thời Preemptive Priority (độ ưu tiên 1 > 2 > 3 …)

Giản đồ Gantt:

P1	P2	P3	P4	P2	P1	P5

0 4 6 10 27 34
43 52

Thời gian đáp ứng trung bình: (0 + 0 + 0 + 3 + 31)/5 = 6.8

Thời gian đợi trung bình: (30 + 21 + 0 + 3 + 31)/5 = 17

Thời gian hoàn thành trung bình: (43 + 30 + 4 + 20 + 40)/5 = 27.4

15. Sử dụng các giải thuật FCFS, SJF, SRTF, Priority –Pre, RR (10) để tính các giá trị thời gian đợi, thời gian đáp ứng, thời gian hoàn thành trung bình và vẽ giản đồ Gantt cho các tiến trình sau:

Process	Arrival Time	Burst Time	Priority
P1	0	20	20
P2	25	25	30
P3	20	25	15
P4	35	15	35
P5	10	35	5
P6	15	50	10

FCFS:

Giản đồ Gantt

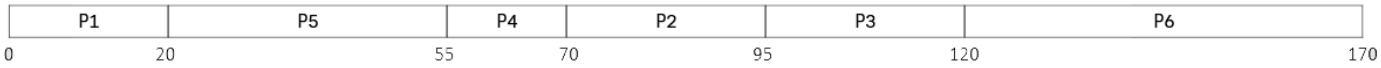
Thời gian đợi trung bình: (0 + 105+ 85 + 120 + 10 + 40)/6 = 60

Thời gian đáp ứng trung bình: (0 + 105+ 85 + 120 + 10 + 40)/6 = 60

Thời gian hoàn thành trung bình: (20 + 130 + 110 + 135 + 45 + 90)/6 = 88,33

SJF:

Giản đồ Gauntt



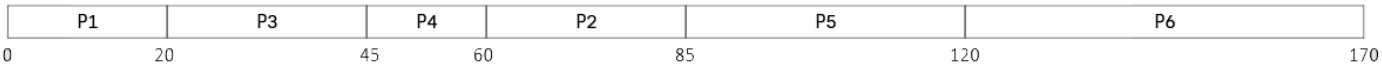
Thời gian đợi trung bình: (0 + 45 + 75 + 20 + 10 + 105)/6 = 42,5

Thời gian đáp ứng trung bình: (0 + 45 + 75 + 20 + 10 + 105)/6 = 42,5

Thời gian hoàn thành trung bình: (20 + 70 + 100 + 35 + 45 + 155)/6 = 70,833

SRTF:

Giản đồ Gantt



Thời gian đợi trung bình: (0 + 35 + 0 + 10 + 75 + 105)/6 = 37,5

Thời gian đáp ứng trung bình: (0 + 35 + 0 + 10 + 75 + 105)/6 = 37,5

Thời gian hoàn thành trung bình: (20 + 60 + 25 + 25 + 110 + 155)/6 = 65,833

Priority-Pre (5>10>…):

Giản đồ Gantt

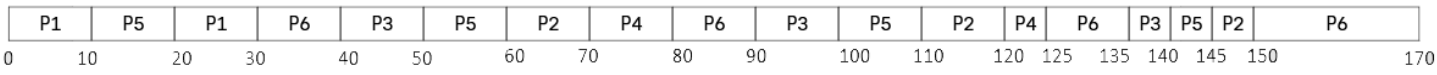
Thời gian đợi trung bình: (110 + 105 + 75 + 140 + 0 + 30)/6 = 76,67

Thời gian đáp ứng trung bình: (0 + 105 + 75 + 140 + 0 + 30)/6 = 58,33

Thời gian hoàn thành trung bình: (130 + 130 + 100 + 155 + 35 + 80)/6 = 105

RR(10):

Giản đồ Gantt



Thời gian đợi trung bình: (10 + 100 + 95 + 75 + 100 + 105)/6 = 80,833

Thời gian đáp ứng trung bình: (0 + 35 + 85 + 35 + 0 + 15)/6 = 28,33

Thời gian hoàn thành trung bình: (30 + 145 + 120 + 90 + 135 + 155)/6 = 112,5

16. Xét tập các tiến trình sau (với thời gian yêu cầu CPU và độ ưu tiên kèm theo). Vẽ giản đồ Gantt và tính thời gian đợi trung bình và thời gian lưu lại trong hệ thống trung bình (turnaround time) cho các giải thuật sau:

a. SJF Preemptive

b. RR với quantum time = 2

c. Preemptive Priority (độ ưu tiên 1 > 2 > ...)

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

a. SJF Preemptive:

Giản đồ Gantt

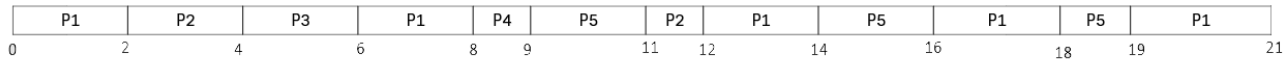
Thời gian đợi trung bình: (11 + 1 + 3 + 0 + 3)/5 = 3,4

Thời gian đáp ứng trung bình: (0 + 0 + 3 + 0 + 3)/5 = 1,2

Thời gian hoàn thành trung bình: (21 + 4 + 5 + 1 + 8)/5 = 7,8

b. RR với quantum = 2:

Giản đồ Gantt



Thời gian đợi trung bình: (11 + 8 + 2 + 5 + 10)/5 = 7,2

Thời gian đáp ứng trung bình: (0 + 1 + 2 + 5 + 5)/5 = 2,6

Thời gian hoàn thành trung bình: (21 + 11 + 4 + 6 + 15)/5 = 11,4

c. Preemptive Priority (độ ưu tiên 1 > 2 > ...):

Giản đồ Gantt

Thời gian đợi trung bình: (6 + 3 + 0 + 1 + 12)/5 = 8,4

Thời gian đáp ứng trung bình: (0 + 0 + 0 + 1 + 12)/5 = 2,6

Thời gian hoàn thành trung bình: (16 + 6 + 2 + 2 + 17)/5 = 8,6

17. 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 cho các giải thuật sau:

a. FCFS

b. SJF preemptive

c. RR với quantum time = 10

a. FCFS

Giản đồ Gantt

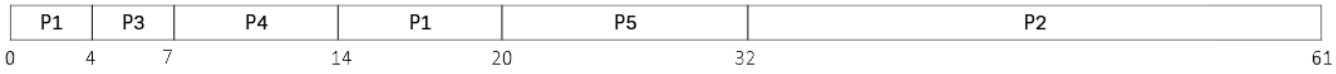
Thời gian đợi trung bình: (0 + 8 + 35 + 37 + 42)/5 = 24,4

Thời gian đáp ứng trung bình: (0 + 8 + 35 + 37 + 42)/5 = 24,4

Thời gian hoàn thành trung bình: (10 + 37 + 38 + 44 + 54)/5 = 36,6

b. SJF preemptive

Giản đồ Gantt



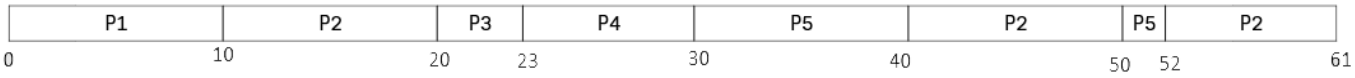
Thời gian đợi trung bình: (10 + 30 + 0 + 3 + 13)/5 = 11,2

Thời gian đáp ứng trung bình: (0 + 30 + 0 + 3 + 13)/5 = 9,2

Thời gian hoàn thành trung bình: (20 + 59 + 3 + 10 + 25)/5 = 23,4

c. RR với quantum time = 10

Giản đồ Gantt



Thời gian đợi trung bình: (0 + 30 + 16 + 18 + 33)/5 = 19,4

Thời gian đáp ứng trung bình: (0 + 8 + 16 + 19 + 23)/5 = 13,2

Thời gian hoàn thành trung bình: (10 + 59 + 19 + 25 + 55)/5 = 33,6