Bai tap chuong 4 IT007 UIT

Mai Nguyễn Nam Phương: 225211664

1. Tại sao phải định thời? Có những loại bộ định thời nào?
   - 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. Nguyên nhân là 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, Medium-Term Scheduling, Long-Term Scheduling.

2. Định thời CPU là gì? Bộ định thời nào chịu trách nhiệm thực hiện việc này?
   - Định thời CPU, còn được gọi là lập lịch CPU, là một quá trình quan trọng trong hệ điều hành đa chương trình, giúp 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.

3. Phí tổn gây ra khi định thời là gì?
   - Phí tổn gây ra khi định thời liên quan đến việc chuyển đổi giữa các tiến trình trong hệ điều hành. Khi một tiến trình được chọn để thực thi, hệ điều hành phải lưu trữ trạng thái hiện tại của tiến trình đang chạy và khôi phục trạng thái của tiến trình mới. Quá trình này gọi là context switch.

4. Trình bày các tiêu chuẩn định thời CPU?
   - Hướng người dùng (User-oriented):
     - Thời gian đáp ứng (Response time): khoảng thời gian từ lúc tiến trình gửi yêu cầu thực thi đến khi yêu cầu được đáp ứng lần đầu tiên (trong các hệ thống time-sharing, interactive system) → cực tiểu
     - Thời gian hoàn thành (Turnaround time): khoảng thời gian từ lúc một tiến trình được nạp vào hệ thống đến khi tiến trình đó kết thúc → cực tiểu
     - Thời gian đợi (Waiting time): tổng thời gian một tiến trình đợi trong ready queue → cực tiểu
   - Hướng hệ thống (System-oriented):
     - Hiệu năng sử dụng CPU (processor utilization): định thời sao cho CPU càng bận càng tốt → cực đại
     - Tính công bằng (fairness): tất cả tiến trình phải được đối xử như nhau
     - Thông lượng (throughput): số tiến trình hoàn tất công việc trong một đơn vị thời gian → cực đại

5. Kể tên các giải thuật định thời CPU?
   - 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.

FCFS :

- Mô tả :
    - Cơ chế thực thi :
        - Tiến trình nào yêu cầu CPU trước sẽ được cấp phát trước.
        - Tiến trình sẽ thực thi đến khi kết thúc hoặc bị blocked do I/O.
    - Chế độ quyết định : Non-Preemptive.
    - Hiện thực : Sử dụng hàng đợi FIFO.
        - Tiến trình đi vào được thêm vào cuối hàng đợi.
        - Tiến trình được lựa chọn để xử lý được lấy từ đầu của queue.
- Ưu điểm :
    - Sẽ không bị starvation.
    - Thuật toán này dễ cài đặt. Code đơn giản.
- Nhược điểm :
    - Thời gian chờ trung bình của FCFS thường khá dài (VD : Một process có burst-time rất dài đến trước, khi đó các process có burst-time nhỏ sẽ phải chờ 1 khoảng thời gian rất lâu mới đến lượt thực thi).
    - Lãng phí thời gian do thời gian phần cứng trống khá nhiều (convoy effect).
    - Non-preemptive. Sẽ không hoạt động tốt trong các hệ thống chia sẻ thời gian (time-sharing system) khi các user đều mong muốn được sử dụng CPU trong một khoảng thời gian và không muốn delay quá lâu.

SJF :

- Mô tả :
    - Cơ chế thực thi :
        - Định thời công việc ngắn nhất trước (Burst-time nhỏ nhất).
        - Khi CPU được tự do, nó sẽ cấp phát cho tiến trình nào yêu cầu ít thời gian nhất để kết thúc (burst-time nhỏ nhất).
        - Burst-time có được từ việc dự đoán, dựa vào các lần chạy trước của tiến trình.
        - Nếu có 2 tiến trình cùng Burst-time, tiến trình nào vào hàng đợi trước sẽ được chạy trước (không xét độ ưu tiên).
    - Chế độ quyết định : Non-Preemptive.
- Ưu điểm : Tối ưu. Cho thời gian chờ đợi trung bình tối thiểu với một tập tiến trình cho trước.
- Nhược điểm :

    - Cần phải ước lượng thời gian cần CPU tiếp theo của process (Burst time).
    - Có thể xảy ra starvation nếu số lượng process có burst time nhỏ cần được thực thi quá nhiều.

SRTF :

- Mô tả :
    - Cơ chế thực thi :
        - (Tương tự SJF).
        - Nếu một tiến trình mới được đưa vào danh sách với chiều dài sử dụng CPU cho lần tiếp theo **nhỏ hơn** (lưu ý, chỉ **nhỏ hơn,** nếu burst-time bằng thì không preempt) thời gian còn lại của tiến trình đang xử lý, nó sẽ dừng hoạt động tiến trình hiện hành (**preempt**).
    - Chế độ quyết định : Preemptive.
- Ưu điểm :
    - Preemptive. Thời gian đáp ứng nhanh cho các tác vụ nhỏ.
    - Tránh việc một tác vụ lớn độc chiếm CPU.
    - Thời gian chờ đợi trung bình thường sẽ nhỏ hơn SJF.
- Nhược điểm :
    - (Các nhược điểm của SJF).
    - Tăng thời gian hoàn thành trung bình.

Priority Scheduling :

- Mô tả :
    - Cơ chế hoạt động :
        - Mỗi tiến trình sẽ được gán 1 độ ưu tiên.
        - CPU sẽ được cấp cho tiến trình có độ ưu tiên cao nhất.
        - Định thời sử dụng độ ưu tiên có thể là :
            - Preemptive : Khi một tiến trình mới xuất hiện có độ ưu tiên cao hơn, nó sẽ preempt tiến trình đang chạy.
            - Non-Preemptive : Tiến trình đang chạy sẽ tiếp tục chạy.
        - Nếu có 2 tiến trình cùng độ ưu tiên, thì tiến trình nào đến trước sẽ được chạy trước. Burst-time không được áp dụng để so sánh ở đây.
    - Chế độ quyết định : Non-Preemptive hoặc Preemptive.
- Ưu điểm :
    - Các tác vụ quan trọng sẽ được thực thi trước.
- Nhược điểm :
    - Có thể xảy ra starvation : Các process có độ ưu tiên thấp có thể không bao giờ được thực thi (giải pháp : aging – Độ ưu tiên của process sẽ tăng theo thời gian).

Round Robin :

- Mô tả :
    - Cơ chế hoạt động :
        - Mỗi tiến trình nhận được một đơn vị nhỏ thời gian CPU (time-slice, quantum time), thông thường từ 10-100msec để thực thi.
        - CPU Schedulers sẽ chọn 1 tiến trình từ ready queue và “lên dây cót” một quantum cho tiến trình, sau đó cho tiến trình chạy. Lúc này, sẽ có 2 khả năng có thể xảy ra :

        - Thời gian chạy > Quantum : Khi đó, tiến trình sẽ bị interrupt và CPU Schedulers sẽ chọn tiếp tiến trình tiếp theo.
        - Thời gian chạy < Quantum : Tiến trình tiếp theo sẽ ngay lập tức được thực thi tiếp (không cần chờ hết quantum time của tiến trình trước), và tiến trình tiếp theo đó cũng được gán 1 quantum time.
    - Phụ thuộc nhiều vào quantum time :
        - Quantum time ngắn thì đáp ứng nhanh, tuy nhiên overhead lớn do chuyển ngữ cảnh nhiều. Quantum time phải > thời gian chuyển ngữ cảnh (context switch).
        - Quantum time dài thì đáp ứng chậm, tuy nhiên thông lượng (throughput) sẽ cao. Và khi quantum time quá lớn RR->FCFS (Quantum time lớn -> Không bao giờ bị ngắt -> Ai vào trước làm trước -> FCFS).
    - Khi cả tiến trình vừa thực thi xong và tiến trình mới cũng arrive vào cùng một thời điểm, thì tiến trình mới sẽ vào hàng đợi trước rồi mới đến tiến trình cũ.
    - Các tiến trình đều có độ ưu tiên giống nhau.
  - Chế độ quyết định : Preemptive.
- Ưu điểm :
  - Thời gian đáp ứng trung bình thường thấp -> Thích hợp cho các hệ thống time-sharing.
  - Không xảy ra tình trạng starvation.
- Nhược điểm :
  - Thời gian chờ đợi trung bình thường khá lớn.
  - Chuyển ngữ cảnh nhiều -> Hao phí cao.
  - Hiệu suất thuật toán phụ thuộc nhiều vào việc chọn quantum time.
  - Không thể sử dụng thuật toán nếu muốn các ứng dụng có độ ưu tiên khác nhau.

Highest Response Ratio Next (HRRN) :

- Mô tả :
  - Cơ chế hoạt động :
    - Chọn process tiếp có giá trị RR (Response Ratio) **lớn nhất**.
    - Các process ngắn được ưu tiên hơn vì service time (hay burst time) nhỏ.
  - Công thức :

$$response\ ratio = \frac{waiting\ time + estimated\ run\ time}{estimated\ run\ time} = 1 + \frac{waiting\ time}{estimated\ run\ time}$$

Công thức tính RR (Response Ratio) của thuật toán HRRN.

- Ưu điểm :
  - Không xảy ra starvation.
  - Tự động cân bằng giữa việc ưu tiên một tiến trình có thời gian thực thi nhỏ và một tiến trình đã ở quá lâu trong hệ thống (aging).
- Nhược điểm :
  - Non-Preemptive.

Multilevel Queue Scheduling :

- Mô tả :
  - Cơ chế hoạt động :
    - Hàng đợi ready được chia thành nhiều hàng đợi riêng biệt theo một số tiêu chuẩn như :
      - Đặc điểm và yêu cầu định thời của process.
      - Foreground (interactive) và background process.
    - Process được gán cố định vào một hàng đợi, mỗi hàng đợi sẽ sử dụng một giải thuật riêng.
    - Có 2TH hệ điều hành định thời cho các hàng đợi :
      - Có một độ ưu tiên cố định cho từng hàng đợi (fixed priority scheduling).
        - Hàng đợi có độ ưu tiên cao hơn phải được chạy xong (empty) trước khi hàng đợi có độ ưu tiên thấp hơn được phép chạy.
        - Nếu có 1 tiến trình đi vào hàng đợi có độ ưu tiên cao hơn trong khi hàng đợi có độ ưu tiên thấp hơn đang được thực thi, hàng đợi có độ ưu tiên thấp hơn đó sẽ bị preempt.
      - Time-slice : Mỗi hàng đợi nhận được một khoảng thời gian chiếm CPU và phân phối cho các process trong hàng đợi khoảng thời gian đó.
  - Chế độ quyết định : Non-Preemptive hoặc Preemptive.
- Ưu điểm :
  - Áp dụng nhiều giải thuật định thời cho nhiều loại tiến trình có độ ưu tiên khác nhau.
  - Cho phép các CPU-Bound process được ưu tiên hơn trong việc thực thi -> Thời gian hệ thống thực thi tác vụ được cải thiện.
  - Có thể hoạt động trong cả 2 chế độ : Preemptive và Non-Preemptive.
- Nhược điểm :
  - Các hàng đợi đa cấp này cần được giám sát -> Hao phí tài nguyên hệ thống.
  - Process không thể di chuyển từ hàng đợi này sang hàng đợi khác -> Không linh động.

Multilevel Feedback Queue

- Mô tả :
  - Cơ chế hoạt động :
    - (Tương tự Multilevel Feedback Queue).
    - Điểm khác biệt : Cho phép process nhảy từ queue này đến queue khác.
- Ưu điểm :
  - Thích nghi với các tiến trình. VD : Một tiến trình nếu sử dụng quá nhiều CPU time thì sẽ xếp nó vào queue có độ ưu tiên thấp hơn.
  - Aging. VD : Một process đã xuất hiện lâu mà không được thực thi, sẽ được đưa lên 1 queue có độ ưu tiên cao hơn.
  - Thuật toán chung nhất, có thể được thiết kế để phù hợp với các hệ thống khác biệt.
- Nhược điểm :
  - Tốn tài nguyên hệ thống để duy trì các queue -> Có thể không thích hợp đối với các hệ thống nhỏ.

- Thiết kế phức tạp.

7. Đặc điểm của định thời trên hệ thống có nhiều bộ xử lý? Khi nào cần phải thực hiện cân bằng tải?
   - Định thời CPU trở nên phức tạp hơn khi hệ thống có nhiều bộ xử lý.
   - Cân bằng tải cần được thực hiện khi có sự không cân đối trong việc sử dụng các bộ xử lý. Điều này thường xảy ra khi một hoặc một số bộ xử lý có quá nhiều tải (hoặc công việc), trong khi các bộ xử lý khác lại rỗi rãi.

8. Đặc điểm định thời theo thời gian thực?

9. Mô tả các đặc điểm cơ bản của bộ định thời CFS trên Linux?
   - Định thời theo lớp: Mỗi lớp được gán một độ ưu tiên cụ thể. Bộ định thời chọn tác vụ có độ ưu tiên cao nhất trong lớp có độ ưu tiên cao nhất. Thời gian sử dụng CPU của mỗi tác vụ không dựa trên quantum time cố định mà dựa trên tỷ lệ giờ CPU. Nhân Linux cài đặt sẵn 2 lớp: default và real-time. Các lớp khác có thể được thêm vào.
   - Thời gian sử dụng CPU: Được tính dựa trên giá trị nice được gán cho mỗi tác vụ, có giá trị từ -20 đến 19. Giá trị thấp hơn có độ ưu tiên cao hơn.
   - Target latency: Khoảng thời gian mà một tiến trình cần được chạy ít nhất một lần. Target latency có thể tăng lên nếu số lượng tiến trình tăng lên.
   - Virtual run time: Mỗi tác vụ có giá trị virtual run time riêng, được kết hợp với một hệ số đặc biệt dựa trên độ ưu tiên. Các tiến trình có độ ưu tiên bình thường có virtual run time tương đương với thời gian chạy thực tế. Bộ định thời chọn tiến trình có virtual run time nhỏ nhất để thực thi tiếp.

10. Mô tả các đặc điểm cơ bản của định thời trên Windows?
    - Định thời theo độ ưu tiên: Tác vụ có độ ưu tiên cao nhất luôn được chạy tiếp. Tiến trình sẽ được thực thi cho đến khi block bởi system call, hết quantum time, hoặc bị thay thế bởi một tiến trình khác có độ ưu tiên cao hơn.
    - Sử dụng 32 độ ưu tiên: Được chia thành 2 lớp: variable (1-15) và real-time (16-31). Độ ưu tiên 0 dành cho quản lý bộ nhớ. Mỗi độ ưu tiên có hàng đợi riêng.
    - Idle thread: Được chạy nếu không có bất cứ tác vụ nào trong hàng đợi.
    - Các hàm thư viện Windows API: Cung cấp cho tiến trình các lớp ưu tiên sau: REALTIME_PRIORITY_CLASS, HIGH_PRIORITY_CLASS, ABOVE_NORMAL_PRIORITY_CLASS, NORMAL_PRIORITY_CLASS, BELOW_NORMAL_PRIORITY_CLASS, IDLE_PRIORITY_CLASS.
    - Tiến trình có thể có các độ ưu tiên tương đối sau: TIME_CRITICAL, HIGHEST, ABOVE_NORMAL, NORMAL, BELOW_NORMAL, LOWEST, IDLE.
    - Lớp ưu tiên và độ ưu tiên tương đối: Có thể kết hợp để xác định giá trị ưu tiên. Độ ưu tiên cơ sở (lúc khởi tạo) là NORMAL bên trong lớp.
    - Khi hết quantum: Độ ưu tiên có thể giảm nhưng không nhỏ hơn độ ưu tiên cơ sở.
    - Windows 7 có thêm user-mode scheduling (UMS): UMS cho phép ứng dụng tạo và quản lý tiểu trình độc lập với nhân. Hiệu quả hơn trong trường hợp có nhiều tiểu trình. Định thời UMS được thực hiện với sự hỗ trợ của các thư viện như C++ Concurrent Runtime (ConcRT).

11. (Bài tập mẫu)

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

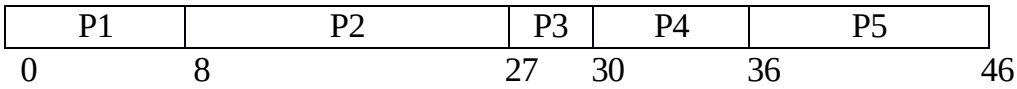
Vẽ sơ đồ Gantt và tính thời gian chờ trung bình, thời gian đáp ứng trung bình, thời gian lưu lại trong hệ thống (turnaround time) trung bình cho các giải thuật sau:

a. FCFS

b. SJF preemptive

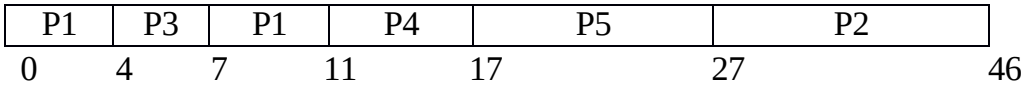
c. RR với quantum time = 6.

a.



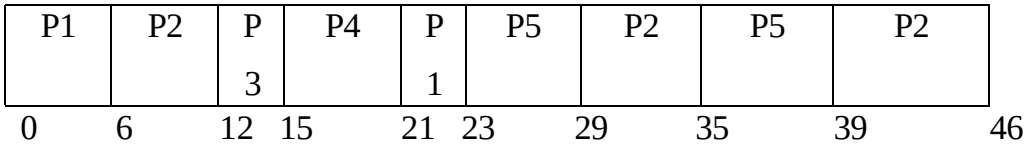
- Thời gian chờ trung bình: (0+6+23+25+29)/5 = 10.6
- Thời gian đáp ứng trung bình: (0+6+23+25+29)/5 = 10.6
- Thời gian hoàn thành trung bình: (8+25+26+31+39)/5= 25.8

b.



- 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 hoàn thành trung bình: (11+44+3+12+20)/5= 18

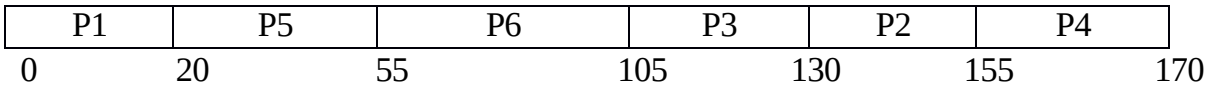
c.



- 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 hoàn thành trung bình: (23+44+11+16+32)/5= 25.2

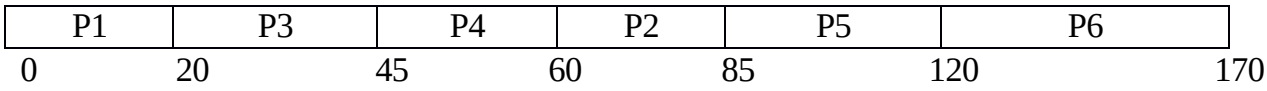
13. (Bài tập mẫu)
14. (Bài tập mẫu)
15.

- FCFS:



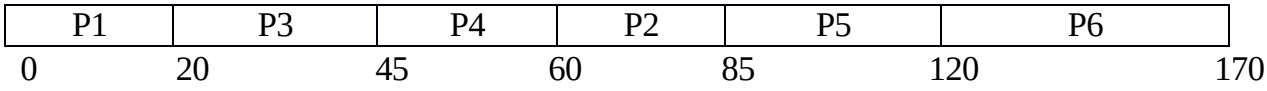
  - Thời gian chờ trung bình: (0+105+85+120+10+40)/6 = 60
  - Thời gian đáp ứng trung bình: (0+105+85+120+10+40)/6 = 60
  - Thời gian hoàn thành trung bình: (20+130+110+135+45+90)/6= 88.33

- SJF:



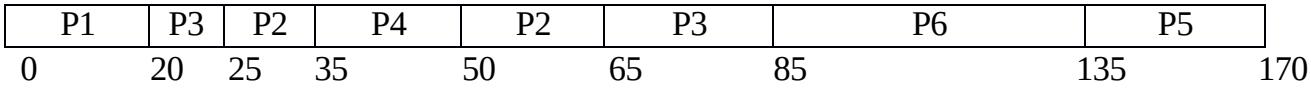
  - Thời gian chờ trung bình: (0+35+0+10+75+105)/6 = 37.5
  - Thời gian đáp ứng trung bình: (0+35+0+10+75+105)/6 = 37.5
  - Thời gian hoàn thành trung bình: (20+60+25+25+110+155)/6= 65.83

- SJF:



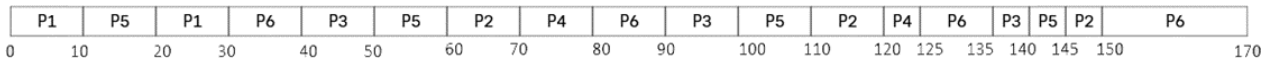
  - Thời gian chờ trung bình: (0+35+0+10+75+105)/6 = 37.5
  - Thời gian đáp ứng trung bình: (0+35+0+10+75+105)/6 = 37.5
  - Thời gian hoàn thành trung bình: (20+60+25+25+110+155)/6= 65.83

- Priority - Pre:



  - Thời gian chờ trung bình: (0+15+40+0+125+70)/6 = 41.67
  - Thời gian đáp ứng trung bình: (0+0+0+0+125+70)/6 = 32.5
  - Thời gian hoàn thành trung bình: (20+40+65+15+160+120)/6= 70

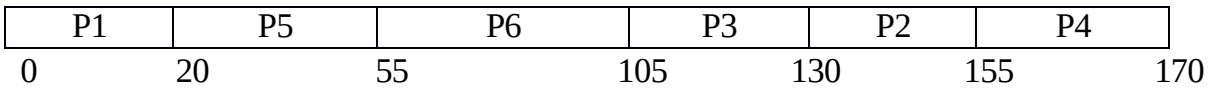
- RR:



- Thời gian chờ trung bình: (10+100+95+75+100+105)/6 = 80.83
- 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

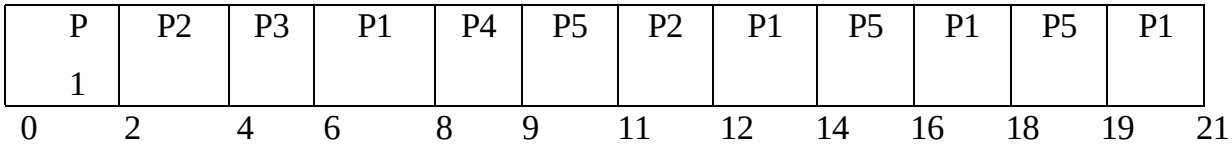
16.

- SJF Preemptive:



- Thời gian chờ trung bình: (11+0+3+1+3)/5 = 3.6
- Thời gian đáp ứng trung bình: (0+0+3+1+3)/5 = 1.2
- Thời gian hoàn thành trung bình: (21+3+5+2+8)/5= 7.8

- RR:



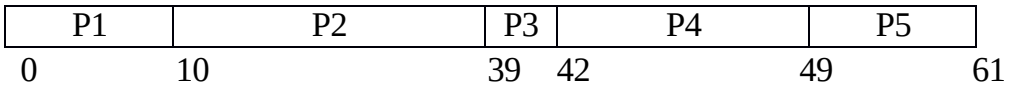
- Thời gian chờ trung bình: (11+8+2+5+10)/5 = 7.2
- Thời gian đáp ứng trung bình: (0+1+2+5+5)/5 = 2.6
- Thời gian hoàn thành trung bình: (21+11+4+6+15)/5= 11.4

- Preemptive Priority:

- Thời gian chờ 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

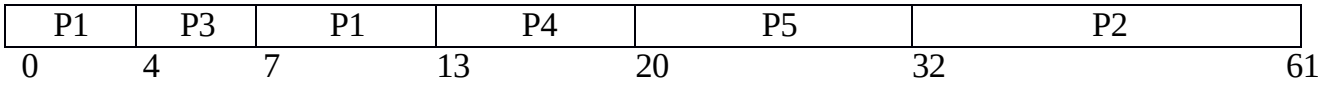
17.

- FCFS:



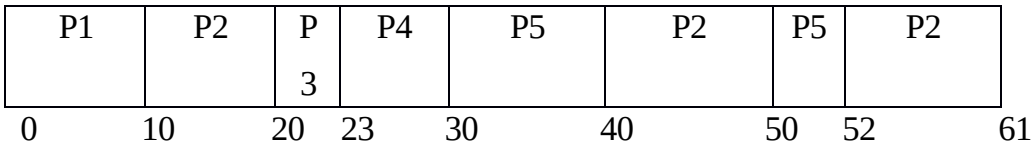
- Thời gian chờ trung bình: (0+8+35+37+42)/5 = 24.4
- Thời gian đáp ứng trung bình: (0+8+35+37+42)/5 = 24.4
- Thời gian hoàn thành trung bình: (10+37+38+44+54)/5= 36.6

- SRTF:



  - Thời gian chờ trung bình: (3+30+0+8+13)/5 = 10.8
  - Thời gian đáp ứng trung bình: (0+30+0+8+13)/5 = 10.2
  - Thời gian hoàn thành trung bình: (13+59+3+15+25)/5= 23

- RR:



  - Thời gian chờ trung bình: (0+30+16+18+33)/5 = 19.4
  - Thời gian đáp ứng trung bình: (0+8+16+18+23)/5 = 13
  - Thời gian hoàn thành trung bình: (10+59+19+25+45)/5= 31.6