**BÀI TẬP NGUYÊN LÝ HỆ ĐIỀU HÀNH**

Bài 1:

1. Cho 5 tiến trình P1-P5 có thời gian tới và thời gian thực hiện như trong bảng. Hãy tính thời gian chờ trung bình và thời gian hoàn thành trung bình.

|  |  |  |  |
| --- | --- | --- | --- |
| **Process** | **Arrival time** | **Burst time** | **Priority** |
| P1 | 0.0 | 11 | 1 |
| P2 | 3.0 | 7 | 3 |
| P3 | 5.0 | 4 | 2 |
| P4 | 5.0 | 2 | 3 |
| P5 | 2.0 | 8 | 2 |

FCFS:

SJF độc quyền:

SJF không độc quyền:

Đáp án:

Lượng tử thời gian: q = 3

FCFS (đến cùng lúc): 15s FCFS(có thời gian đến): 14,2s

SJF độc quyền: 10s SJF không độc quyền: 8.4s

Priority độc quyền:13,6 priority không độc quyền: 13,6

RR: 17.6

FCFS:

Biểu đồ Gant:

P1 P5 P2 P3 P4 P1 P5 P2 P4 P3

0 11 19 26 30 32 0 11 19 26 28 33

- Thời gian chờ trung bình:

Tc1 = 0, Tc2 = 19 – 3 = 16, Tc3 = 26 – 5 = 22, Tc4 = 30 – 5 = 25, Tc5 = 11 – 2 =9

(P1 đến trước => Giải quyết trước nên Tc1 = 0. Tc2 đến vào 3ms nhưng phải chờ P1 và P5 vào trước nên thời gian chờ là 19 – 3 = 16, Tc3 cũng vậy …)

Tc = (Tc1 + Tc2 + Tc3 + Tc4 + Tc5) / 5 = 14.2

Thời gian hoàn thành trung bình = Tc + T\_burst = 14.2 + (11+ 7 + 4 + 2 +8)/5 = 20.6

* SJF độc quyền:

P1 P4 P3 P2 P5

0 11 13 17 24 32

Thời gian chờ trung bình của các tiến trình: P1 = 0, P2 = 17 – 3 = 14, P3 = 13 – 5 = 8, P4 = 11 – 5 = 6, P5 = 24 – 2 = 22

(P1 đến trước => Giải quyết trước hết 11 ms. Trong 11ms, có 4 tiến trình đến và đang chờ trong hàng đợi. CPU sẽ giải quyết tiến trình nào có burst time ngắn đến dài, theo thứ tự P4, P3, P2, P5. Thời gian chờ trung bình của các tiến trình tính tượng tự như FCFS)

Thời gian chờ trung bình: T\_c = (0 + 14 +8 + 6 + 22) / 5 = 10 (s)

Thời gian hoàn thành trung bình: 10 + (11 + 7 + 4 + 2 +8)/5 = 16.4(s)

- SJF không độc quyền:

P1 P5 P2 P4 P3 P2 P5 P1

0 2 3 5 7 11 16 23 32

(P1 đến trước => Sau 2ms, P5 đến nên giải quyết P5 tiếp theo. P1 lúc đó còn 11 – 2 = 9s cần thiết. Tại 3ms, P2 đến nên CPU phục vụ P2, P5 còn 8 – 1 = 7s. Đến 5ms, P3 và P4 cùng đến nhưng P4 có burst time ngắn hơn nên thực hiện trước. Đến 7ms, còn lại các tiến trình P3: 4s, P2: 7 – 2 = 5ms, P5 = 7s, P1 = 9s => Thực hiện theo P3, P2, P5, P1)

Thời gian chờ trung bình của các tiến trình: Tính từ lúc phải chờ đến lúc thực hiện tiếp

Tc1 = 23 – 2 = 21, Tc2 = 11 – 5 = 6, Tc3 = 7 – 5 = 2, Tc4 = 0 (vì nó đến lúc 5ms), Tc5 = 16 – 3 = 13

Thời gian chờ trung bình: T\_c = (21 + 6 + 2 + 0 + 13) /5 = 8.4

Thời gian hoàn thành trung bình: 8.4 + (11 + 7 + 4 + 2 +8)/5 = 14.8

- Không có độ ưu tiên nên không tính được priority độc quyền và không độc quyền

- Round Robin: Lượng tử thời gian q = 3

Biểu đồ Gantt: P1 -> P5 -> P2 -> P3 -> P4

P1 P5 P2 P1 P3 P4 P5 P2 P1 P3 P5 P2 P1

0 3 6 9 12 15 17 20 23 26 27 29 30 32

(Theo thuật toán, ta sẽ tính theo các mốc thời gian:

Tại 0ms, P1 vào và chiếm dụng CPU trong thời gian q.

Sau 3ms, P1 phải chuyển đến cuối hang đợi. Trong từ 0 – 3ms, có P5 và P2 vào. Hàng đợi lúc này sẽ là P5 P2 P1. P5 vào trước, thực hiện 3ms, đến 6ms, lúc đó có them P3, P4 nên hang đợi sẽ là P2 P1 P3 P4 P5.

Tại 6ms, P2 chiếm dụng CPU. Xong rồi chuyển xuống hang đợi: P1P3P4P5P2

Tại 9ms, P1 chiếm dụng 3ms CPU. Hàng đợi lúc này: P3P4P5P2P1

Tại 12ms, P3 chiếm dụng, …)

Thời gian chờ của từng tiến trình: Tc1 = 0 + (9 – 3) + (23-12) + (30-26) = 21

Tc2 = (6 – 0) + (20 – 9) + (29 – 23) - 3 = 20 (P2 có arrival times là 2ms)

Tc3 = 12 + (26 – 15) - 5 = 18

Tc4 = 15 – 5 = 10

Tc5 = 3 + (17 – 6) + (27 – 20) – 2 = 19

Thời gian chờ trung bình: Tc = (21 + 20 + 18 + 10 + 19)/3 = 17.6

Thời gian hoàn thành trung bình: Tht = 17.6 + (11 + 7 + 4 + 2 +8)/5 = 24

1. Tính giá trị X1, X2, X3 khi thực thi theo SJF độc quyền (P1 vào lúc 2 ms, P2: 0 ms, P3: 4 ms). Biết X1 = X2 = X3 = X = 10. Q = 3. Thứ tự vào là P1, P2, P3

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiến trình** | **P1** | **P2** | **P3** |
| **Công việc** | **X2 = X1 + 2X2**  **X1 = 2X1 \* X** | **X1 = X1 + X3**  **X3 = X2 + 2X1** | **X3 = X2 + 2X3**  **X2 = X – X2** |
| Mã Assembly | Load reg1, x1  Load reg2, x2  Load reg3, 2  Mul reg2, reg3  Add reg1, reg2  Save x2, reg1  Load reg1, x1  Mul reg1,reg3  Load reg, x  Mul reg1,reg  Save x1, reg1 | Load reg1, x1  Load reg2, x2  Load reg3, x3  Add reg1, reg3  Save x1, reg1  Load reg4, 2  Mul reg1, reg4  Add reg2, reg1  Save x3, reg2 | Load reg3, x3  Load reg2, x2  Load reg1, 2  Mul reg3, reg1  Add reg3, reg2  Save x3, reg3  Load Reg, x  Sub reg, reg2  Save x2, reg |

Ta có burst time của các tiến trình như sau:

|  |  |
| --- | --- |
| P1 | 11 |
| P2 | 9 |
| P3 | 9 |

* Theo SJF độc quyền, ta sẽ có bảng Gantt như sau:

P2 P3 P1

0 9 18 29

(P2 vào trước, sau 9ms thì P3 được ưu tiên trước P1)

X1 = X2 = X3 = X = 10,

Từ 0-9ms:

Reg1 = X1 = 10, reg2 = X2 = 10, reg3 = X3 = 10, reg1 = reg1 + reg3 = 10 + 10 = 20, X1 = reg1 = 20, reg4 = 2, reg1 = reg1 \* reg4 = 40, reg2 = reg2 + reg1 = 10 + 40 = 50, X3 = reg2 = 50

Từ 9 – 18ms:

R3 = X3 = 50, r2 = X2 = 10, r1 = 2, r3 = r3\* r1 = 50\* 2 = 100, r3 = r3 + r2 = 110, ,X3 = r3 = 110, r = 10, r = r – r2 = 0, X2 = 0

Từ 18ms đến 29ms:

R1 = X1 = 20, r2 = X2 = 0, r3 = 2, r2 = r2\*r3 = 0, r1 = r1 + r2 = 20, X2 = r1 = 20, r1 = X1 = 20, r1 = r1\*r3 = 40, r = x = 10, r1 \*= r = 400, x1 = r1 = 400

Vậy X1 = 400, X2 = 20, X3 = 110

* Với q = 3, ta sẽ có biểu đồ gantt như sau:

P2 P1 P3 P2 P1 P3 P2 P1 P3 P1

0 3 6 9 12 15 18 21 24 27 29

Từ 0 – 3 ms: P2 thực hiện 3 lệnh đầu

R1 = X1 = 10, r2 = x2 = 10, r3 = x3 = 10

Từ 3- 6 ms: P1 thực hiện 3 lệnh đầu: r1 = x1 = 10, r2 = x2 = 10, r3 = 2

Từ 6 – 9 ms: P3 thực hiện 3 lệnh đầu: r3 = x3 = 10, r2 = x2 = 10, r1 = 2

Từ 9 – 12ms: P2 thực hiện 3 lệnh tiếp theo: r1 = r1 + r3 = 12, x1 = r1 = 12, r4 = 2

Từ 12 – 15ms: P1 thực hiện 3 lệnh tiếp theo: r2 = r2\*r3 = 100, r1 += r2 = 112, x2 = r1 = 112

Từ 15 – 18 ms: P3 thực hiện 3 lệnh tiếp theo: r3 \*= r1 = 1120, r3 += r2 = 1232, x3 = r3 = 1232

Từ 18 – 21ms: P2 thực hiện 3 lệnh cuối: r1 \*= r4 = 1120 \* 2 = 2240, r2 += r1 = 112 + 2240 = 2360, x3 = r2 = 2360

Từ 21 – 24ms: P1 thực hiện 3 lệnh tiếp theo: r1 = x1 = 12, r1 \*= r3 = 12\*1232 = 14784, r = x = 10

Từ 24 – 27ms: P3 thực hiện 3 lệnh cuối: r = x = 10, r -= r2 = 10 – 2360 = -2350, x2 = r = -2350

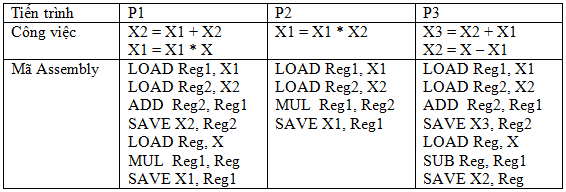
Tuwf27 – 29ms, P1 thực hiện 2 lệnh cuối: r1 \*= r = 14784 \* (-2350) = -34742400, x1 = r1 = -34742400

Vậy x1 = -34742400, x2 = -2350, x3 = 2360

Câu 2:

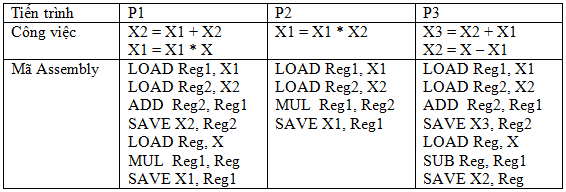
Cho các tiến trình P1, P2, P3 thực hiện các công việc trong bảng dưới. Giả sử: X1, X2, X3, X là các biến dùng chung; ban đầu X1 = X2 = X3 = X = 10. Tính giá trị X1, X2, X3 sau khi 3 tiến trình kết thúc biết hệ thống sử dụng thuật toán lập lịch sau:

* SJF (Shortest Job First) độc quyền biết các tiến trình đến hàng đợi ready như sau: P1 đến tại giây 0, P2 đến tại mili giây 5, P3 đến tại mili giây 4 và thời gian thực hiện mỗi lệnh Assembly là 1 mili giây. (X1 = 2000, X2 = -1990, X3 = 2020)
* RR với thời gian lượng tử bằng 2 lệnh Assembly và thứ tự đến của các tiến trình là P2, P3, P1(X1 = 100, X2 = 0, X3 = 20)
* SJF (Shortest Job First) Không độc quyền biết các tiến trình đến hàng đợi ready như sau: P1 đến tại giây 0, P2 đến tại mili giây 2, P3 đến tại mili giây 3 và thời gian thực hiện mỗi lệnh Assembly là 1 mili giây.(X1 = 100, X2 = -90, X3 = 120)



Câu 3

Cho các tiến trình P1, P2, P3 thực hiện các công việc trong bảng dưới:



* Trong trường hợp các biến đều là các biến riêng của mỗi tiến trình và đều được khởi tạo = 10. Sử dụng thuật toán RR để lập lịch với q bằng thời gian thực hiện 2 câu lệnh hợp ngữ, thứ tự đến của các tiến trình là P3, P1, P2. Tính giá trị của mỗi biến.
* Thực hiện với SJF không độc quyền với p1 vào lúc 0(ml s), p2 vào lúc 1(ml s), p3 vào lúc 2(ml s). Các biến X1, X2, X3, X là biến dùng chung. X1 = X2 = X3 = 10 = X.

Câu 4

Một tiến trình ở trong đọan găng có thể ở những trạng thái nào. Giải thích thông qua một ví dụ cụ thể.

Câu 5

Thao tác nguyên tử là gì. Cho ví dụ. Tại sao trong giải pháp Test&Set, thao tác Test&Set(lock) phải là thao tác nguyên tử.

Câu 6

Bài toán Cây cầu cũ: Để tránh sụp đổ, người ta chỉ có cho phép tối đa 3 xe lưu thông đồng thời qua một cây cầu rất cũ. Tại mỗi thời điểm, chỉ cho phép tối đa 3 xe lưu thông trên cầu. Mỗi chiếc xe khi đến đầu cầu sẽ kiểm tra điều kiện lên cầu, và khi đã qua cầu được sẽ báo hiệu kết thúc. Mỗi xe hoạt động như một tiến trình Car(). Mô tả tiến trình Car(), chỉ ra đoạn găng và các đoạn bên ngoài đoạn găng. Giải thích.

Câu 7

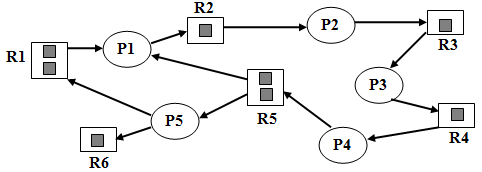
Để vượt qua sông, các nhân viên Microsof và các Linux hacker cùng sử dụng một bến sông và phải chia sẻ một số thuyền đặc biệt. Mỗi chiếc thuyền này chỉ cho phép chở 1 lần 4 người, và phải có đủ 4 người mới khởi hành được. Để bảo đảm an toàn cho cả 2 phía, cần tuân thủ các luật sau :

* Không chấp nhận 3 nhân viên Microsoft và 1 Linux hacker trên cùng một chiếc thuyền.
* Ngược lại, không chấp nhận 3 Linux hacker và 1 nhân viên Microsoft trên cùng một chiếc thuyền.
* Tất cả các trường hợp kết hợp khác đều hợp pháp.
* Thuyền chỉ khởihành khi đã có đủ 4 hành khách.

Mô tả tiến trình Hacker và tiến trình nhân viên (Employee) **theo mã giả**. Chỉ ra vấn đề đồng bộ và đoạn găng trong bài toán.

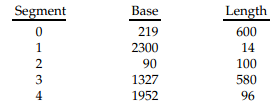
**Câu 8**

Cho sơ đồ phân phối tài nguyên như trong hình vẽ. Hệ thống có khóa chết không. Giải thích.



**Câu 9**

Trong kỹ thuật phân đoạn, cho bảng phân đoạn như hình vẽ, giải thích và tính địa chỉ vật lý của các địa chỉ lôgic sau: (0,430), (1,10), (2,500), (3,400), (4,112).



**Câu 10a:** Trong kỹ thuật phân trang 2 mức. Với địa chỉ logic là 32bit, cho p1 = 10, tính địa p2 và d khi biết một frame trong trong bộ nhớ vật lý có dung lượng 2048byte.

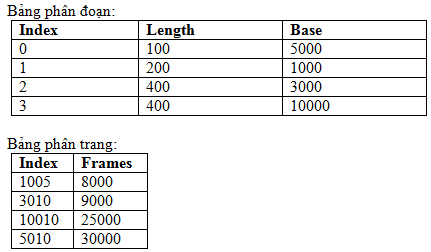
**Câu 10b:**

Trong kỹ thuật phân trang 2 mức, với địa chỉ logic: p1 = 5, p2 = 7, d = 10. Cho biết địa chỉ logic nào sau đây là sai, vì sao?

1. 10, 100, 1000
2. 40, 20, 100
3. 30, 150, 300
4. 25, 120, 200
5. 20, 110, 1300

**Câu 11**

Mô tả quá trình biên dịch địa chỉ trong kỹ thuật quản lý bộ nhớ phân đoạn kết hợp với phân trang. Cho bảng phân đoạn và bảng phân trang như hình vẽ, xác định địa chỉ vật lý của các địa chỉ lôgic sau: (1,5,10), (2, 1000, 30).



**Câu 12:** Các page CPU cần đọc lần lượt là :

1 2 3 2 4 1 2 3 0 3 1 5 7 1 3 2 1 2 4 5

Bộ nhớ vật lý có 3 frame (ban đầu đang trống)

Sử dụng thuật toán thay thế trang FIFO và LRU để xem mỗi thuật toán có bao nhiêu lỗi trang.

**Câu 13:** Các cylinder mà CPU cần đọc:**100, 10, 40, 150, 130, 2, 68, 70, 180, 160, 63**

Đầu đọc đang ở vị trí cylinder **90**. Tổng số cylinder là 200 (0-199)

Tính quãng đường đi đầu đọc cần di chuyển trong các thuật toán: FCFS, SSTF, Scan (C-Scan), Look (C-Look)

Head: 90

=> đến 100 = 10

Đến 10 = 90

Đến 40 = 30

Đến 150 = 110 => 20=> 128 => 66 => 2 => 110 => 20 => 97

10 => 90 => 30 => =110 => 20=> 128 => 66 => 2 => 110 => 20 => 97

= 683.

**Câu 14:** Quản lý phân đoạn trong bộ nhớ cho biết địa chỉ vật lý của các địa chỉ logic sau và giải thích (1,30), (0, 500), (2, 400), (3, 10).

|  |  |
| --- | --- |
| **Limit** | **Base** |
| 600 | 246 |
| 23 | 2000 |
| 350 | 2200 |
| 12 | 1200 |

**Câu 15:** Trong phân trang kết hợp phân đoạn, cho địa chỉ logic như sau (1,10, 100), (0, 40, 4), (3, 5, 19), (2, 7, 8) hãy cho biết địa chỉ vật lý, giải thích?

|  |  |
| --- | --- |
| **Limit** | **Base** |
| 600 | 246 |
| 23 | 2000 |
| 350 | 2200 |
| 12 | 1200 |

|  |  |
| --- | --- |
| **Index** | **Frames** |
| 2010 | 400 |
| 280 | 8000 |
| 1205 | 9000 |
| 2300 | 6000 |

**Bài 16:** Cho các page CPU cần đọc như sau:

0 1 2 1 3 4 2 3 0 1 0 3 5 2 3 1 0 2 0 1 6 2 3 1

Giả sử bộ nhớ trong có 3 frames. Sử dụng thuật toán FIFO, LRU để thay thế trang khi gặp lỗi trang. Có bao nhiêu lỗi trang?

**Câu 17:** Trong ổ cứng có 200 cylinder (0-199)

Các cylinder cần đọc trong hàng đợi queue: 70, 100, 12, 44, 178, 130, 65, 110

Đầu đọc đang ở vị trí head = 87.

Tính quãng đường đầu đọc đi? FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK.

**Bài 17 (chương 5 deadlock):**

Cho 6 tiến trình: P0,…, P5 và 3 loại tài nguyên: A (10 cá thể), B (8 cá thể), C (7 cá thể).

Tại thời điểm T0 các tài nguyên được cấp phát và số lượng tài nguyên lớn nhất được yêu cầu của mỗi tiến trình được thể hiện như trong bảng dưới đây.

|  |  |  |
| --- | --- | --- |
| **Process** | **Allocation**  **(A, B,C)** | **Max**  **(A, B, C)** |
| P0 | (1,0,2) | (6,4,2) |
| P1 | (2,1,0) | (2,2,2) |
| P2 | (0,0,2) | (1,1,2) |
| P3 | (3,2,1) | (5,3,2) |
| P4 | (2,3,0) | (8,6,5) |
| P5 | (1,1,1) | (9,8,5) |

Sử dụng giải thuật chủ nhà băng kiểm tra xem tại thời điểm T0 có xảy ra deadlock hay không?

**Hướng dẫn câu 6**

**Car(int direction)** /\* **direction** xác định hướng di chuyển của mỗi chiếc xe.\*/  
{

RuntoBridge();ArriveBridge(direction);   
PassBridge();Exit Bridge();  
RunfromBridge();

}

Đoạn găng: là đoạn đi trên cầu PassBridge()

* + Cầu là tài nguyên chung. Các tiến trình (xe) chia sẻ cầu để thực hiện công việc
  + Khi số tiến trình vượt qua 3 thì hệ thống xảy ra mâu thuẫn

**Hướng dẫn câu 7**

**Hacker()**  
{

RuntoRiver(); // Đi đến bờ sông **HackerArrives ();** // Kiểm tra điều kiện xuống thuyền  
CrossRiver(); // Khởi hành qua sông

}

**Employee()**  
{

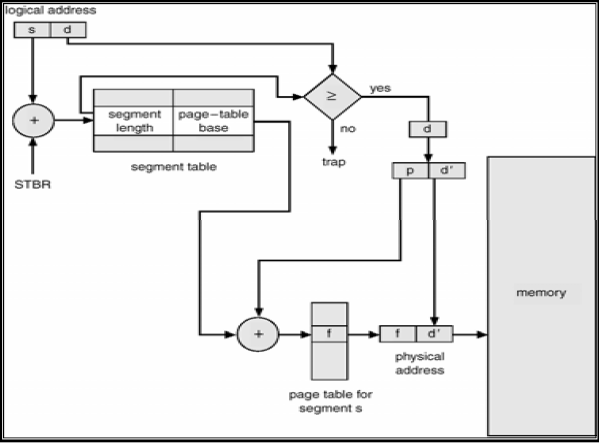
RuntoRiver(); // Đi đến bờ sông **EmployeeArrives ();** // Kiểm tra điều kiện xuống thuyền  
CrossRiver(); // Khởi hành qua sông

}

* Đoạn găng và đồng bộ
  + Đoạn găng: CrossRiver(). Giải thích: thuyền là tài nguyên tranh chấp. Khi có nhiều hơn số tiến trình (Hacker, Embloyee) được phép trong đoạn găng sẽ xảy ra mâu thuẫn (chìm thuyền)
  + Đồng bộ: đủ 4 người và 2 hacker, 2 nhân viên. Giải thích: Yêu cầu đồng bộ xảy ra khi kiểm tra điều kiện xuống thuyền, phải có đúng 2 tiến trình nhân viên và 2 tiến trình hacker thì mới thực hiện CrossRiver().

**Goi y cau 10**

* Quá trình biên dịch địa chỉ



* Tính địa chỉ vật lý:
  + (1, 5, 10) => chỉ số đoạn = 1, địa chỉ trang = 5 + địa chỉ cơ sở của đoạn 1 = 1000+5, địa chỉ vật lý = địa chỉ trang vật lý + offset = 8010
  + (2, 1000, 30) không hợp lệ vì 1000 > độ dài đoạn 2 (400)