1. Một hệ điều hành chứa 3 tiến trình người dùng, mỗi tiến trình yêu cầu 4 đơn vị tài nguyên R. Số lượng đơn vị R tối thiểu để không có deadlock nào xảy ra là

a.5

b.9

**c.10**

d.12

Tham khảo: https://www.gatevidyalay.com/deadlock-in-os-deadlock-problems-questions/?fbclid=IwAR1AAkFfhvwE1ng9OzWFdAC\_vbM5Ww2u0vEbC1B3rSvFPvlRxiFHRKmJdfo

→ Số lượng đơn vị mà mỗi quy trình nắm giữ = Ít hơn một đơn vị so với nhu cầu tối đa của nó → P1: 3 , P2: 3, P3:3 → R tối đa có deadlock = 3x3 = 9 → R tối thiểu không deadlock = 9 + 1 = 10

1. Khi thiết kế hệ thống bộ nhớ đệm cache của máy tính, kích thước khối bộ đệm cache (hoặc cache line) là một tham số quan trọng. Câu nào sau đây đúng trong ngữ cảnh này?

a.Kích thước khối nhỏ hơn phải chịu hình phạt bỏ lỡ (miss penalty) bộ nhớ cache thấp hơn

(nhỏ thì chịu ít hình phạt)

b.Kích thước khối nhỏ hơn có nghĩa là thẻ bộ nhớ cache nhỏ hơn

c.Kích thước khối nhỏ hơn ngụ ý vị trí không gian tốt hơn

d.Kích thước khối nhỏ hơn có nghĩa là thời gian truy cập bộ nhớ cache thấp hơn

3.Một tiến trình yêu cầu toàn bộ tài nguyên cần thiết một lần, nếu có đủ tài nguyên thì hệ thống sẽ cấp phát, nếu không đủ tài nguyên thì process sẽ bị blocked. Điều này giải quyết được điều kiện nào sau đây trong việc ngăn deadlock:

**a.Hold and Wait**

b.Mutual Exclusion

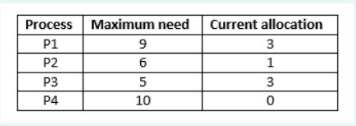
c.No Preemption

d.Circular Wait

* There must exist a process which holds some resource and waits for another resource held by some other process.

4.Một hệ thống chia sẻ 11 tape drives. Phân bổ hiện tại và yêu cầu tối đa của tape drives cho 4 tiến trình được hiển thị bên dưới:

Điều nào sau đây mô tả đúng nhất trạng thái hiện tại của hệ thống?



Available ban đầu: 11 - (3+1+3) = 4

so sánh lớn hơn = need => + allocation

|  |  |
| --- | --- |
| Need = Max - Current | Available |
| 6 (2) | 7 |
| 5 (3) | 10 |
| 2 (1) | 11 |
| 10 (4) | 11 |

1. Not Safe, Not Deadlocked

b. Not Safe, Deadlocked

**c. Safe, Not Deadlocked**

d. Safe, Deadlocked

**Lưu ý: Safe thì Not Deadlocked >< Not safe thì Deadlocked**

5. Điều nào sau đây yêu cầu trình điều khiển thiết bị?

1. Main memory
2. Cache
3. Register
4. **Disk**

6. Khẳng định nào sau đây là đúng ?

**(a) External Fragmentation tồn tại khi có đủ tổng dung lượng bộ nhớ để đáp ứng yêu cầu nhưng không gian khả dụng lại liền kề.**

**(b) Phân mảnh bộ nhớ có thể bên trong cũng như bên ngoài.**

**(c) Một giải pháp cho External Fragmentation là nén**

7.Khi xảy ra lỗi trang

a.Các tiến trình trên hệ thống đang ở trạng thái waiting

b.Tiến trình truy xuất đến trang đang có trong bộ nhớ

**c.Các tiến trình trên hệ thống thường xuyên truy cập các trang không có trong bộ nhớ**

d.Các tiến trình trên hệ thống đang ở trạng thái running

8.Các nội dung cần thiết trong mỗi mục nhập của bảng trang là

a.Truy cập thông tin phù hợp

b.Virtual page number

**c.Cả virtual page number và page frame number**

d.Page frame number

9.Thời gian để sector mong muốn xoay dưới đầu đĩa?

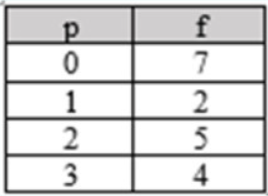
a.Seek time (thời gian di chuyển đĩa)

b.Waiting time (tgian chờ)

c.Settle time (thời gian ổn định….)

**d.Rotational latency (Thời gian xoay đầu đĩa dưới)**

11. Trong cấp phát bộ nhớ với kỹ thuật phân trang (Paging). Cho kích thước trang (page) và kích thước khung trang (frame) là 100K, địa chỉ bắt đầu cấp phát trong bộ nhớ là OK. Cho bảng trang (Page map table-PMT) của tiến trình P như sau :



Địa chỉ vật lý (Physical Address) tương ứng với địa chỉ logic (Logical Address)

|  |  |
| --- | --- |
| 3 | 20K |

là:

a. 220K.

b. 720K

C. 420K.

d. 520K.

→ 4\*100+20 + đc bắt đầu = 420

12.Điều nào sau đây không đúng đối với các kế hoạch ngăn chặn deadlock và tránh deadlock?

a.Tránh deadlock cần có kiến thức ưu tiên về các yêu cầu tài nguyên

b.Trong ngăn chặn deadlock, yêu cầu tài nguyên luôn được cấp nếu trạng thái kết quả là an toàn

c.Ngăn chặn deadlock hạn chế hơn tránh bế tắc

d.Trong trường hợp tránh deadlock, yêu cầu tài nguyên luôn được cấp, nếu trạng thái kết quả là an toàn

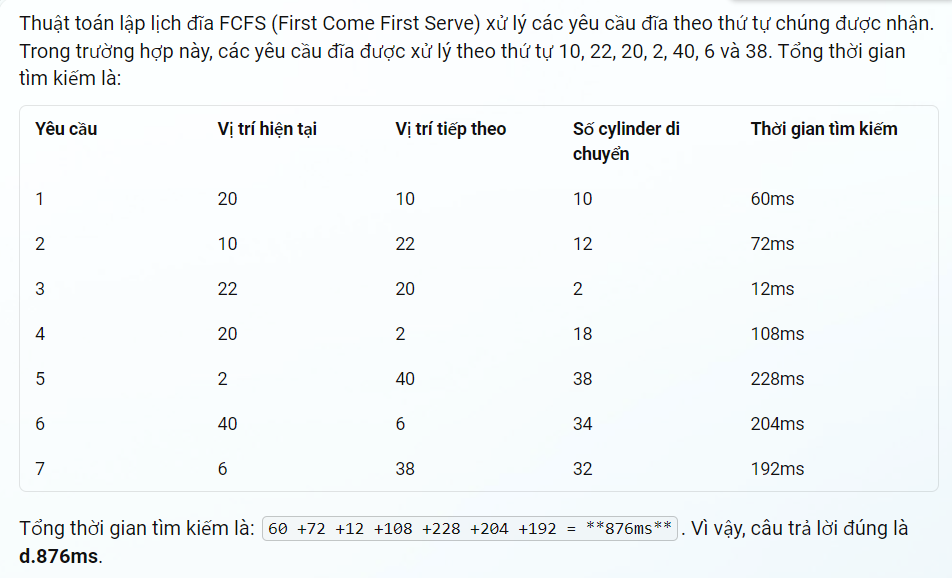
14.Yêu cầu đĩa đến trình điều khiển đĩa theo thứ tự 10, 22, 20, 2, 40, 6 và 38 tại một thời điểm khi ổ đĩa đang đọc từ cylinder 20. Thời gian tìm kiếm là 6 ms/cylinder. Nếu dùng thuật toán lập lịch đĩa FCFS thì tổng thời gian tìm kiếm là bao nhiêu?

a.850ms

b.900ms

c.360ms

**d.876ms**

****

**=> 20-10 + 22-10 + 22-20 + 20-2 + 40-2 + 40-6 + 38-6 = 146\*6 = 876**

15.Thuật toán thay thế trang nào sau đây có thể cho tỷ lệ lỗi trang tăng ngay cả khi số lượng khung được phân bổ tăng lên?

**a. FIFO**

b. LRU

c. MRU

d. Optimal

→ Nghịch lý Belady

16.Cấu trúc đĩa cứng dạng GPT Partition có tối đa … Primary partition.

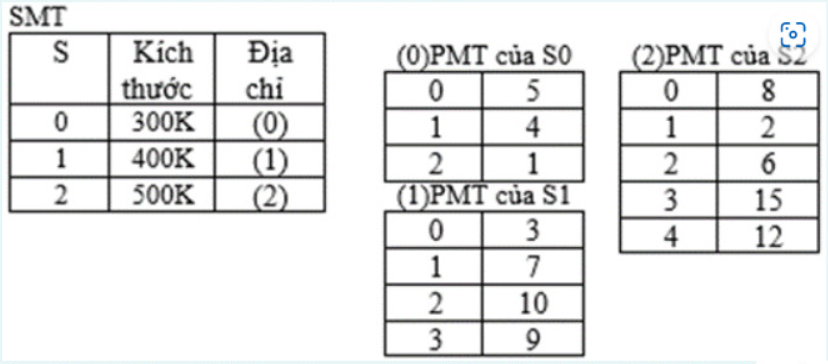
a.4.

b.64.

c.1

**d.128**

17. Trong cấp phát bộ nhớ với kỹ thuật phân đoạn kết hợp (Segmentation with paging). Địa chỉ bắt đầu cấp phát trong bộ nhớ là OK. Cho bảng phân đoạn (Segmentation map table-SMT) và các bảng trang (Page map table-PMT) của tiến trình P như sau :



Địa chỉ vật lý (Physical Address) tương ứng với địa chỉ logic (Logical Address) | 1 | 120K| là:

a**. 720K.**

b. 1020K.

c. 320K.

d. 920K

1 là 120K => p = 120K div 100K = 1; 1 -> s1 -> f= 7

d’ = 120K mod 100K = 20; pa = f \* 100 + d = 7\*100 + 20 = 720

18.Thuật toán tần suất lỗi trang (Page-Fault Frequency) để làm gì?

1. **ngăn chặn tình trạng trì trệ xảy ra**
2. tăng số frame
3. giảm số frame
4. Đếm số lỗi trang

19.Khẳng định nào sau đây là đúng ?

(a) External Fragmentation tồn tại khi có đủ tổng dung lượng bộ nhớ để đáp ứng yêu cầu nhưng không gian khả dụng lại liền kề.

(b) Phân mảnh bộ nhớ có thể bên trong cũng như bên ngoài.

(c) Một giải pháp cho External Fragmentation là nén

1. **a, b và c**
2. b và c
3. a và b
4. a

21. Đối với loại tài nguyên có nhiều instant, thuật toán được sử dụng để tránh deadlock là ......

a. sorting algorithm

b. partition algorithm

c. banker's algorithm

d. a modified resource-allocation graph

Tránh deadlock : có nhiều instance → Banker , nếu ít instance → Đồ thị cấp phát tài nguyên ( Modified Resource-Allocation Graph)

22.Trạng thái phân bổ tài nguyên không xác định bởi yếu tố nào sau đây

a.Nhu cầu tài nguyên tối tiểu của các tiến trình

b.Nhu cầu tài nguyên tối đa của các tiến trình

c.Số lượng tài nguyên được phân bổ

d.Số lượng tài nguyên có sẵn

23.Trong Hashed Page Table có chứa giá trị nào được băm (hash)?

a.Physical address

b.Frame number

**c.Page number (Theo chỉ số)**

d.Logic address

25. Bộ nhớ TLB (Translation Look-aside Buffers) là gì

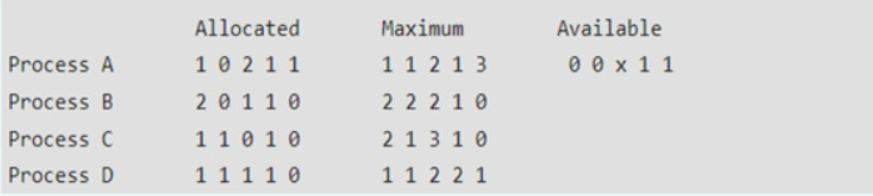
a. có tốc độ truy xuất trang chậm hơn bộ nhớ chính (nhanh hơn)- > Sai

**b. được sử dụng để lưu trữ các trang được truy cập gần hiện tại nhất**

c. chứa các trang cần tìm kiếm sau khi đã tìm trong bảng trang

d. là bộ nhớ thứ cấp

28. Một hệ thống có bốn tiến trình và 5 tài nguyên có thể phân bổ. Sự phân bổ hiện tại và nhu cầu tối đa như sau:



Giá trị nhỏ nhất của x để hệ thống trên ở trạng thái an toàn là

a. 2

b. 3

c. Không an toàn với x bất kỳ о о

d. 1

-> ưu tiên thế cao nhất , cao nhất không được-> nhỏ hơn cũng hong đc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Allocated | Maximum | Available | Need (Maxi-allo) |
| PA | 1 0 2 1 1 | 1 1 2 1 3 | 0 0 3 1 1 | 0 1 0 0 2 |
| PB | 2 0 1 1 0 | 2 2 2 1 0 | 1 1 4 2 1 | 0 2 1 0 0 |
| PC | 1 1 0 1 0 | 2 1 3 1 0 | 3 1 5 3 1 | 1 0 3 0 0 |
| PD | 1 1 1 1 0 | 1 1 2 2 1 |  | 0 0 1 1 1 |

=>Không thể tiếp tục phân bố cho PA do thiếu, nên cấp phát này ko an toàn

29.Thrashing (bế tắc) trong phân trang theo yêu cầu là gì

1. cải thiện hiệu suất hệ thống

b. vượt quá I/O trang

c. giảm mức độ đa chương trình

d. giảm I/O trang

30.Trong cấp phát bộ nhớ với kỹ thuật phân trang (Paging). Cho kích thước trang (page) và kích thước khung trang (frame) là 100K, địa chỉ bắt đầu cấp phát trong bộ nhớ là OK. Cho bảng trang (Page map table-PMT) của tiến trình P như sau :



Với địa chỉ vật lý (Physical Address) 730K, địa chỉ logic (Logical Address) tương ứng

là

a. p = 0 d = 30K

b. p = 1; d = 30K

c. p = 3 d = 30K

d. p = 2 d = 30K

Answer: Do PA là 730k mà vị trí bắt đầu là 0k -> 700k sẽ ở f = 7 - > p = 0, d = 30k

Physical Address= f\* kích thước khung trang + d

31. Chiến lược cấp phát vùng nhớ động nào phân bổ vùng nhớ nhỏ nhất đủ lớn để đáp ứng nhu cầu của tiến trình đến

a. Best-fit

b. Next-fit

C. Worst-fit

d. First-fit

32. Xem xét phần cứng phân trang với TLB. Giả sử rằng toàn bộ bảng trang và tất cả các trang đều nằm trong bộ nhớ vật lý. Mất 10 mili giây để tìm kiếm TLB và 80 mili giây để truy cập bộ nhớ vật lý. Nếu tỷ lệ truy cập TLB là 0,6, thì thời gian truy cập bộ nhớ hiệu quả (tính bằng mili giây) là \_\_\_\_\_

**a. 122**

b. 121

C. 122

d. 120

TLB\_search = 10ms

Memory\_access\_time = 80ms

EAT := (TLB\_search\_time + 2\*memory\_access\_time) \* (1- hit\_ratio) +

(TLB\_search\_time + memory\_access\_time)\* hit\_ratio

EAT = (10 + 2\*80) \* (1 - 0.6) + (10 + 80)\*0.6 = 122

35. Theo hệ thống phân lớp chức năng của hệ thống file, thành phần nào quản lý không gian đĩa trống, chuyển đổi các khối logic thành các khối vật lý và ngược lại

a. File organization module

b. Logical file system

c. Basic file system

d. Device drivers

36. Một counting semaphore được khởi tạo là 10. Sau đó, 6 hoạt động wait () và 4 hoạt động signal () đã được hoàn thành trên semaphore này. Giá trị kết quả của semaphore là

**a. 8**

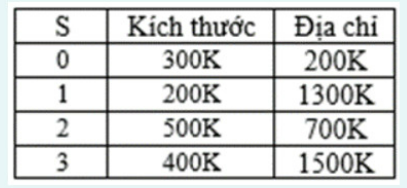
b. 10

c. 12

d. 0

Semaphore = counting - wait + signal = 10 - 6 + 4

38.Trong cấp phát bộ nhớ với kỹ thuật phân đoạn (Segmentation). Địa chỉ bắt đầu cấp phát trong bộ nhớ là OK. Cho bảng phân đoạn (Segmentation map table-SMT) của tiến trình P như sau :



Địa chỉ vật lý (Physical Address) tương ứng với địa chỉ logic (Logical Address) [

2 | 120K ] là:

a. 1620K

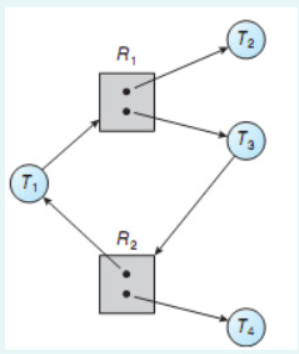
**b. 820K.**

C. 320K.

d. 1420K.

s2 có logic là 120K, < limit (500K) S = 120K + 700K = 820K

39. Cho đồ thị cấp phát tài nguyên của một hệ thống như sau.



Phát biểu nào đúng?

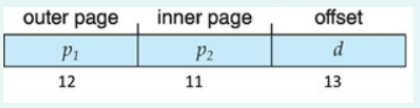
a. Hệ thống có thể không có deadlock (Không có chu trình)

b. Hệ thống có deadlock nếu xóa cạnh R1 được gán cho T2

c. Hệ thống có deadlock

d. Hệ thống không có deadlock nếu xóa cạnh R1 được gán cho T2

40.Trong hệ thống có địa chỉ ảo 36 bit, sử dụng bảng trang 2 cấp để dịch địa chỉ ảo sang địa chỉ vật lý theo cấu trúc (12bit | 11bit | 13bit) như hình vẽ. Hỏi có bao nhiêu bảng trang cấp 2 (inner page)?



a. 8192

b. 1024

**c. 4096**

d. 2048

p1 trỏ đến p2 → số bảng trang của p2 = p1 = 2^12

41.Chức năng thích hợp nhất của Bộ quản lý bộ nhớ (MMU) là gì?

a. Nó là một bộ nhớ liên kết để lưu trữ TLB (translation lookaside buffer)

b. Nó là một kỹ thuật hỗ trợ đa chương trình bằng cách tạo các phân vùng động

c. Nó là một con chip (thiết bị phần cứng) để ánh xạ địa chỉ ảo sang địa chỉ thực

d. Nó là một thuật toán để cấp phát và phân bổ bộ nhớ chính cho một tiến trình

42.Giả sử rằng ta có một chuỗi tham chiếu trang cho một t/trình với m khung (ban đầu tất cả đều trống). Chuỗi tham chiếu trang có độ dài p; n số trang riêng biệt xuất hiện trong đó. Giới hạn dưới về số lỗi trang là mấy (với bất kỳ thuật toán thay thế trang nào):

a. p-n

b. p

C. n

d. m

43. Đồ thị cấp phát tài nguyên "Wait-for" là gì? xem Slide

a. Có đỉnh là các tiến trình và các tài nguyên, cạnh là các yêu cầu/ cấp phát tài nguyên

b. Có các đỉnh là các tiến trình và cạnh mô tả tiến trình Pi đang chờ tiến trình Pj

c. Có các đỉnh là các tài nguyên và cạnh mô tả tiến trình Pi đang được cấp phát tài nguyên Rj

d. Có các đỉnh là các tiến trình và cạnh mô tả tiến trình Pi yêu cầu tài nguyên Rj

44.Thuật toán thay thế trang nào có hiện tượng bất thường Belady?

a. Optimal

b. LRU

C. FIFO

d. Both LRU and Optimal

45. Thực hiện vô hiệu hóa (ngăn) 1 trong số các điều kiện: Mutual exclusion, Hold and Wait, No Preemption, và Circular Wait liên quan đến thuật toán gì về deadlock

a. deadlock prevention (ngừa)

b. deadlock deletion (xóa)

c. deadlock avoidance (tránh)

d. deadlock detection (phát hiện)

46. Điều nào sau đây KHÔNG được chia sẻ bởi các thread của cùng một tiến trình?

a. Stack

b. Address Space

C. Message Queue

d. File Descriptor Table

47. Hãy xem xét một đĩa có 8 bề mặt, 64 track trên mỗi bề mặt và 256 sector trên mỗi track. 512 byte dữ liệu được lưu trữ theo cách nối tiếp bit trong một sector. Dung lượng của đĩa và số bit cần thiết để chỉ định một sector cụ thể trong đĩa tương ứng là bao nhiêu?

a. 256MB, 17bits

b. 64GB, 28bits

c. 64MB, 17bits

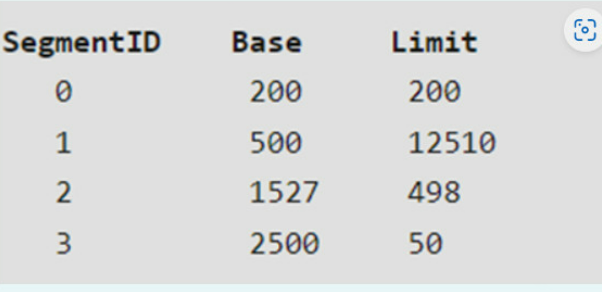
d. 256MB, 19bits

→ dung lượng: 8\*64\*256\*512 = 2^26 byte = 64MB

→ Số bit: 3 + 6 + 8 = 17 bits

48.Hãy xem xét bảng phân đoạn sau trong lược đồ phân đoạn:

SegmentID



Điều gì sẽ xảy ra nếu địa chỉ logic được yêu cầu là - Segment ID 2 và offset 1000?

a. Tìm nạp mục nhập tại địa chỉ vật lý 1498 cho segment Id2

b. Một trap được tạo ra

c. Deadlock

d. Tìm nạp mục nhập tại địa chỉ vật lý 2527 cho segment Id2

Giới hạn ở segment ID 2 là [0, 497] mà offset là 1000 nên không nằm đúng.

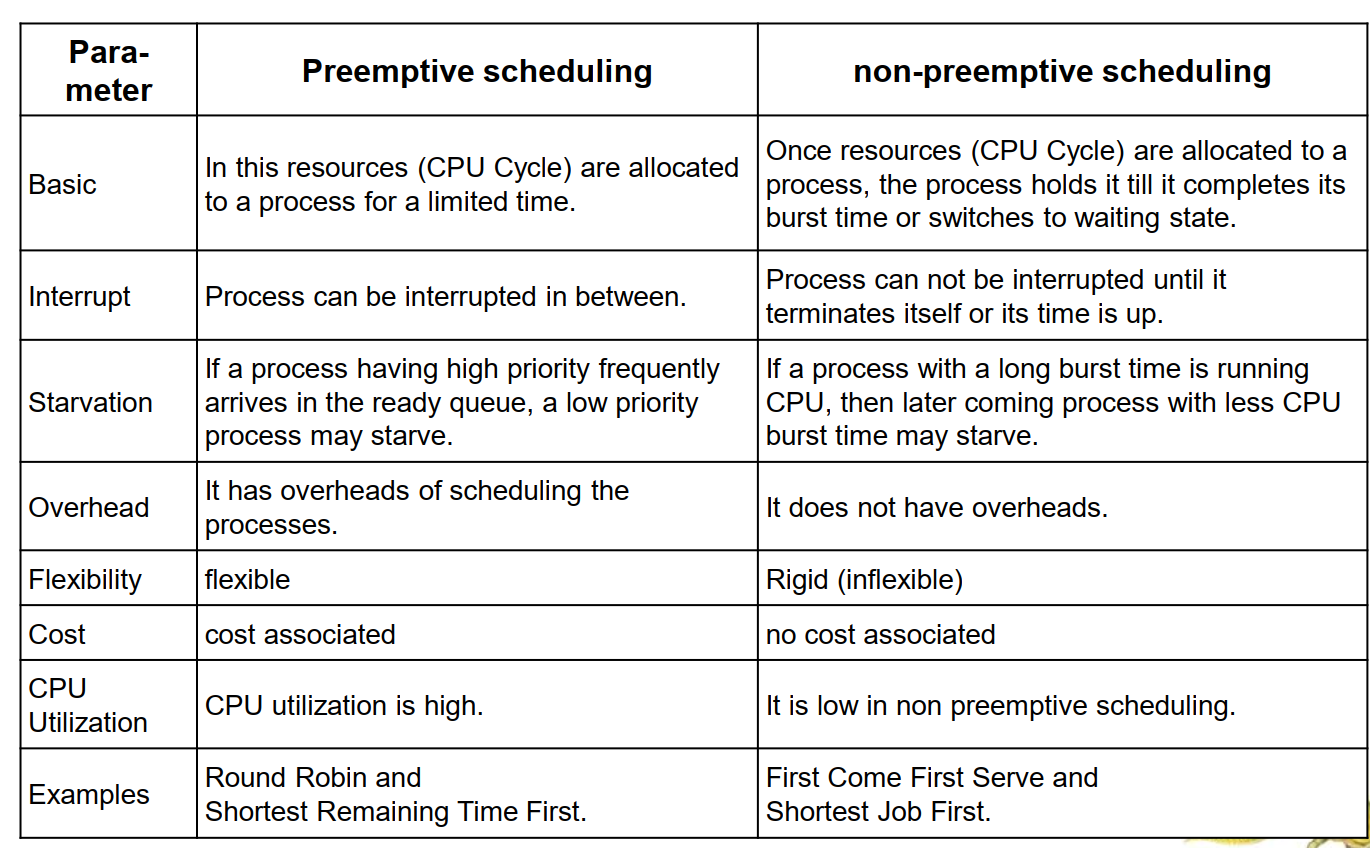
1. **When the process issues an I/O request \_\_\_\_\_\_**
2. It is placed in an I/O queue
3. It is placed in the ready queue
4. It is placed in the Job queue
5. It is placed in a waiting queue



1. **Thuật toán lập lịch dạng non-preemptive không có đặc điểm nào sau đây:**
2. flexible
3. the process not holds it still it completes
4. can not be interrupt until it terminates
5. CPU utilization is low

Lập lịch preemptive đạt được tính linh hoạt (flexible) bằng cách cho phép các tiến trình quan trọng truy cập vào CPU khi chúng đến hàng đợi sẵn sàng, bất kể tiến

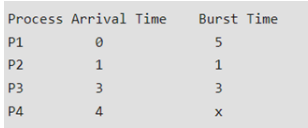
trình nào đang được thực thi hiện tại. Lập lịch non-preemptive được gọi là cứng nhắc vì ngay cả khi một tiến trình quan trọng đi vào hàng đợi sẵn sàng thì CPU đang chạy tiến trình đó không bị xáo trộn.



1. **In Unix operating system, when a process creates a new process using the fork() system call, which of the following state is shared between the parent process and child process?**
2. Both Heap and Stack
3. Stack
4. Shared memory segments
5. Heap

|  |
| --- |
| Note:Heap and stack are being duplicated |

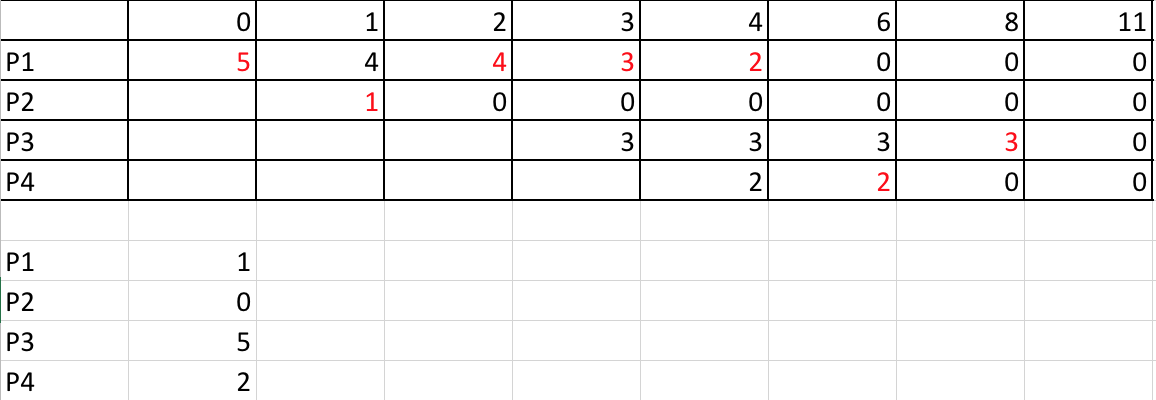
1. **Xem xét các tiến trình CPU sau với thời gian đến (tính bằng mili giây) và Burst Time CPU (tính bằng mili giây) ngoại trừ tiến trình P4 như được đưa ra bên dưới:**

****

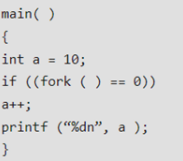
**Nếu thời gian chờ trung bình trên tất cả tiến trình là 2 mili giây và thuật toán lập lịch Shortest Remaining Time First (Preemptive SJF) được sử dụng để lập lịch cho các tiến trình, hãy tìm giá trị của x? => Average waiting time = 2**

Luân: Thế x dô mà chứ ai gãnh đâu mà tính :>

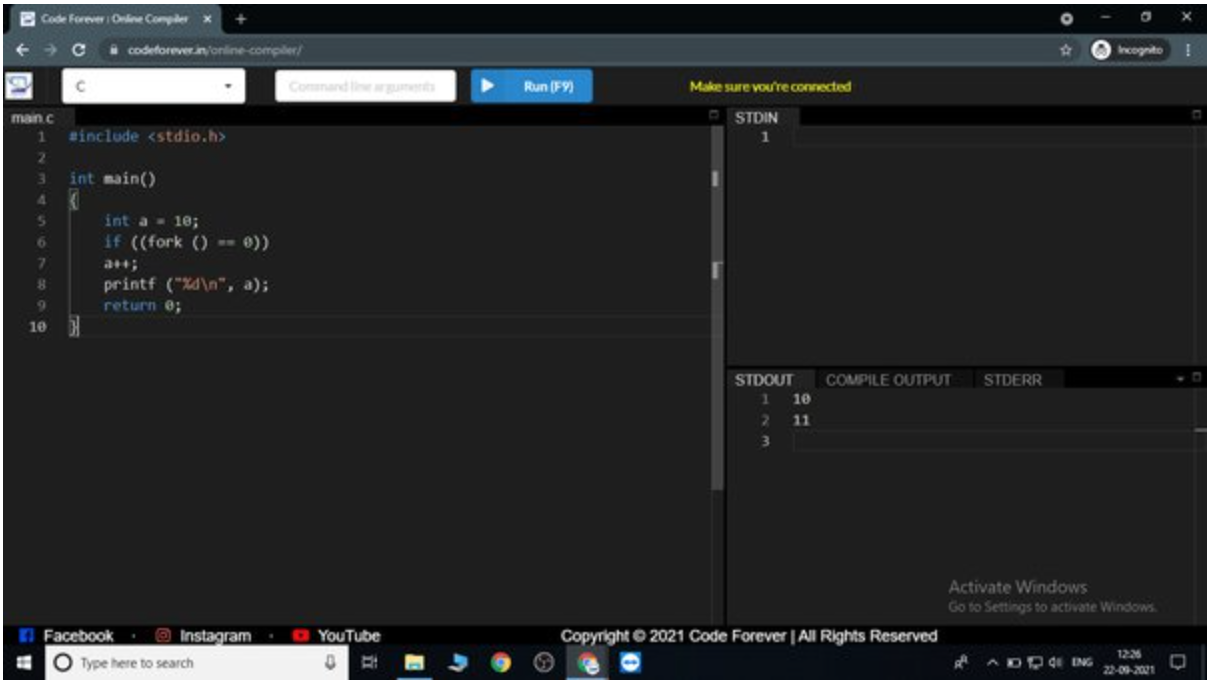
1. 2
2. 5
3. 1
4. 4

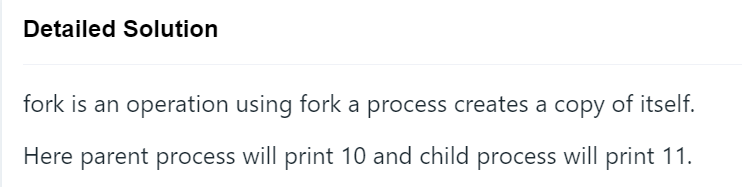


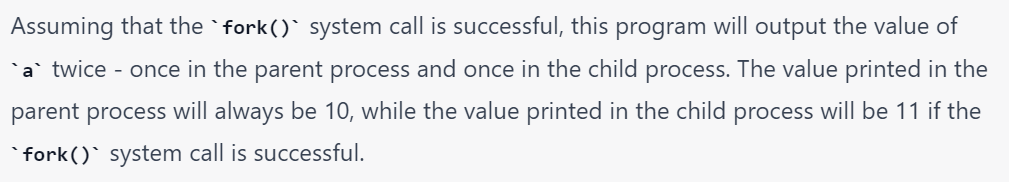
1. **Kết quả của chương trình sau là gì?**

****

1. 10
2. 11 và 10
3. 10 và 11
4. 11/



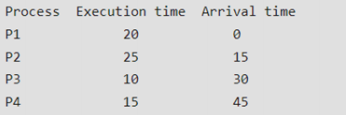




1. **Bộ nhớ nào là secondary storage**
2. ROM
3. hard disk (or hard drive)
4. RAM
5. register
6. **Cơ chế phần cứng cho phép 1 thiết bị thông báo cho CPU là \_\_\_\_\_**
7. interrupt
8. polling
9. none of the above
10. system call



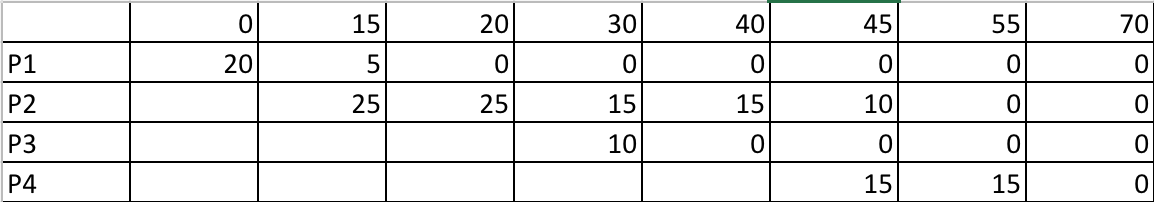
1. **Hệ điều hành sử dụng thuật toán lập lịch Shortest Remaining Time first (SRTF) - tức SJF Preemptive. Xem xét thời gian đến và thời gian thực hiện cho các tiến trình sau:**

****

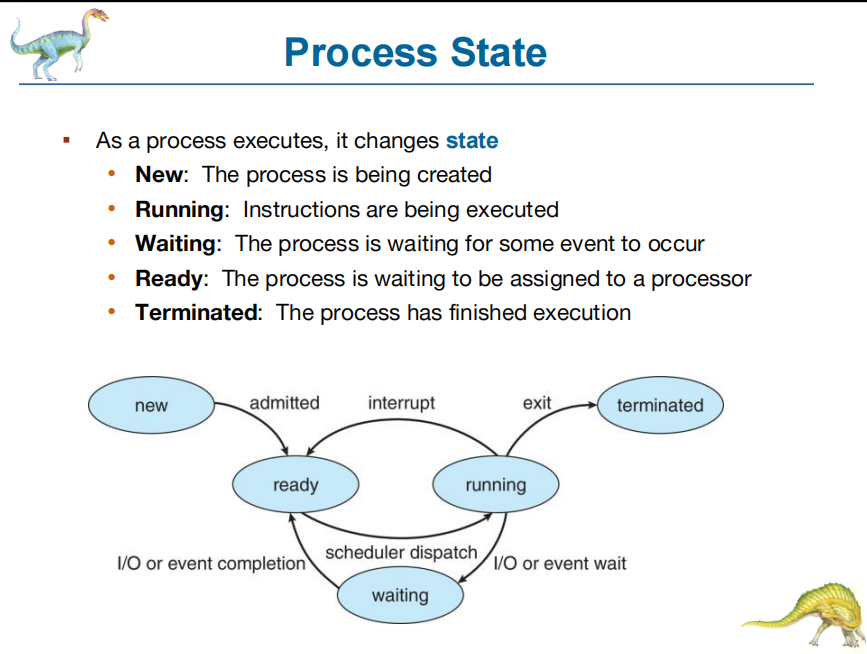
**Tổng thời gian chờ đợi cho quá trình P2 là bao nhiêu? (P4 – 10)**

**Tổng thời gian Turnarround cho quá trình P3 là bao nhiêu? – 10**

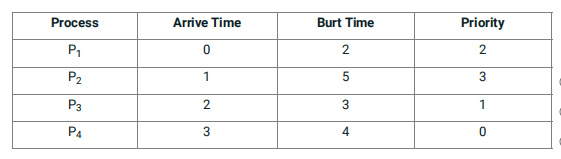
1. 5
2. 55
3. 40
4. 15



1. **Khi một tiến trình đang chạy mà cần input/output dữ liệu thì sẽ chuyển sang trạng thái nào?**
2. Switching
3. ready
4. waiting
5. input/output



1. **Xem xét thông tin của các tiến trình trong bảng. Hệ điều hành sử dụng thuật toán lập lịch Priority Preemptive, tính waiting time trung bình cho các tiến trình này.**

****

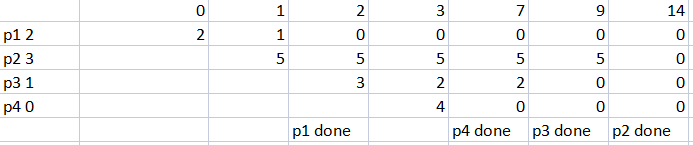
1. 2
2. 1.75
3. 3.75
4. 2.5

Luân: Câu này t làm ra 3 huhu 😭

Muyxue: cỵ cũng ra 3 huhu :<

Hân: em cũng ra 3

Cô bé tóc đỏ: t cũng vậy!!!



avg waiting time = (0+8+4+0)/4 = 3

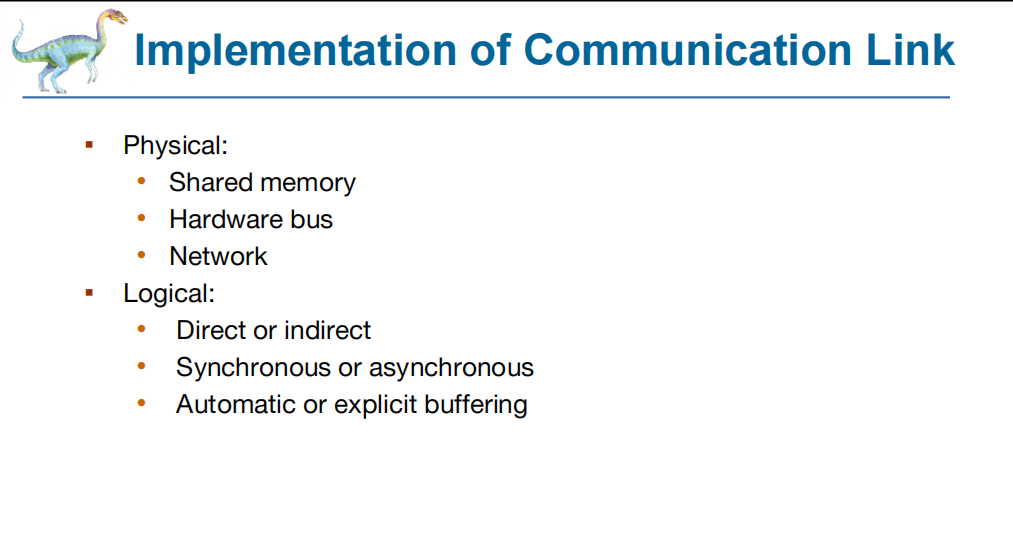
1. **Hãy xem xét một tập hợp n nhiệm vụ với các thời gian chạy đã biết r1, r2 … rn sẽ được chạy trên máy đơn xử lý. Thuật toán lập lịch CPU nào sau đây sẽ dẫn đến throughput (thông lượng) tối đa?**
2. Shortest job first
3. FCFS
4. Round robin
5. Priority

Thông lượng có nghĩa là tổng số tác vụ được thực hiện trên mỗi đơn vị thời gian, tức là tổng thời gian chờ và thời gian bùng nổ.

Lập lịch Shortest job first là một chính sách lập lịch chọn quá trình chờ đợi với thời gian thực hiện nhỏ nhất để thực hiện tiếp theo.

Do đó, trong lập lịch Shortest job first, các công việc ngắn nhất được thực hiện trước. Điều này có nghĩa là việc sử dụng CPU là tối đa. Vì vậy, số lượng nhiệm vụ tối đa được hoàn thành.

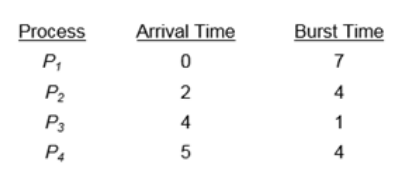
1. **Which is physical link in implementation of Communication Link between processes?**
2. indirect
3. direct
4. synchronous
5. network



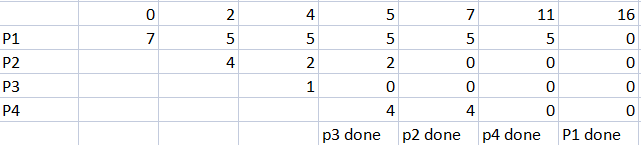
1. **The number of processes in memory means:**
2. **MultiTasking (đa nhiệm):** Cho phép một quá trình (process) thực hiện nhiều tác vụ (task) khác nhau đồng thời trong cùng một thời điểm bằng cách chia sẻ tài nguyên của quá trình đó, chẳng hạn như bộ nhớ, bộ vi xử lý, các thiết bị ngoại vi, vv.
3. **MultiThreading (đa luồng)**: Chạy đồng thời nhiều luồng
4. **MultiProcessing:** Cho phép nhiều quá trình (process) thực hiện đồng thời các tác vụ khác nhau trên nhiều bộ xử lý (CPU) hoặc các lõi xử lý độc lập (cores) trong cùng một máy tính.
5. Multiprogramming (đa chương) : Giữ nhiều chương trình trong bộ nhớ cùng 1 thời điểm. => tăng CPU ultilization, CPU scheduling
6. **Thuật toán Round Robin với q rất lớn sẽ trở thành thuật toán gì**
7. Priority
8. Shortest Job First non-preemptive
9. FCFS
10. Shortest Job First preemptive



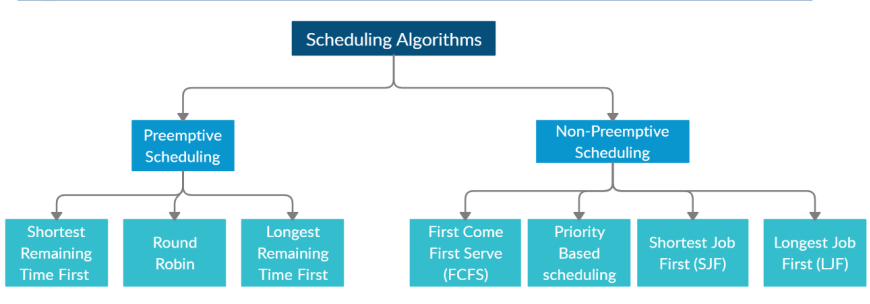
1. **A process can be terminated because of \_\_\_\_**
2. pending alarm
3. signals and signal handlers
4. killed by another process and normal exit
5. waiting another process
6. **Sử dụng thuật toán lập lịch SJF - preemptive cho các tiến trình theo thông tin dưới bảng. Thời gian chờ trung bình là bao nhiêu?**

****

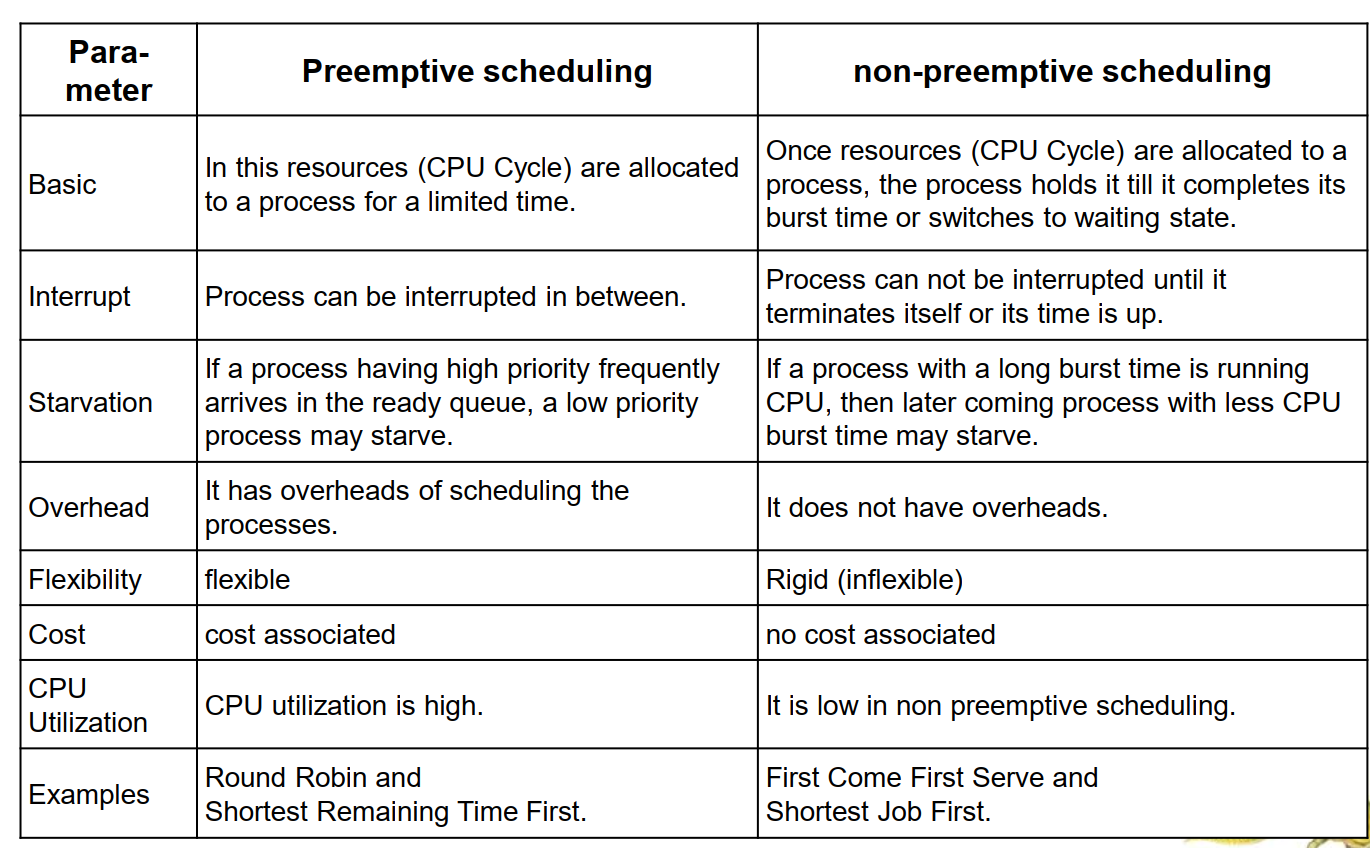
1. 7
2. 4
3. 3
4. 6



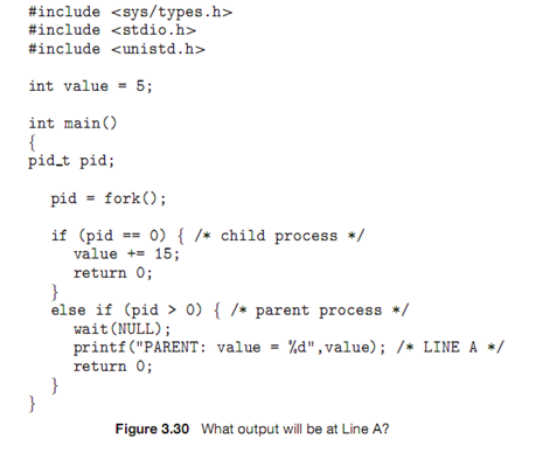
1. **What define the state of a process?**
2. the activity just executed by the process
3. the final activity of the process
4. the current activity of the process
5. the activity to next be executed by the process
6. **Bộ nhớ nào lớn nhất mà CPU có thể truy xuất trực tiếp?**
7. cache
8. hard disk
9. register
10. main memory
11. **Thuật toán lập lịch nào sau đây là preemptive (không độc quyền)**
12. Priority
13. Round Robin with q very large (=FCFS)
14. FCFS
15. Priority with RR q = 3



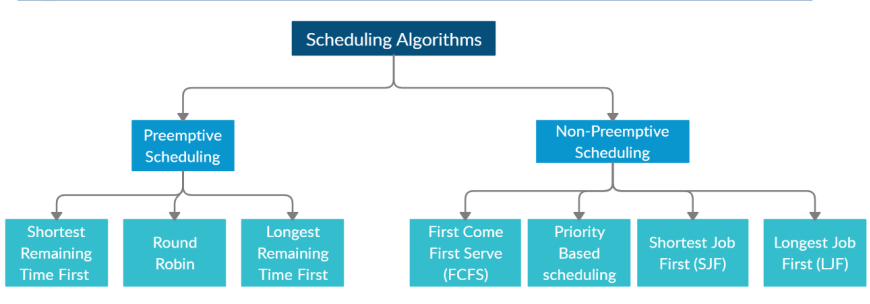
1. **Thuật toán lập lịch dạng preemptive (không độc quyền) không có đặc điểm nào sau đây:**
2. does not have overheads
3. CPU utilization is high
4. interrupted in between
5. flexible



1. **Using the program shown below, explain what the output will be at LINE A**

****

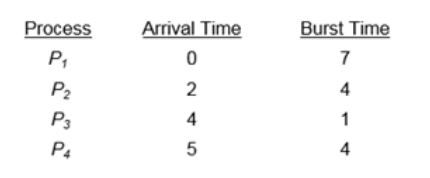
1. 15
2. 20
3. 5
4. 0
5. **Thuật toán lập lịch nào sau đây là non-preemptive (độc quyền)?**
6. Shortest Job First preemptive
7. Priority with Round Robin q = 3
8. Priority
9. Round Robin with q = 3



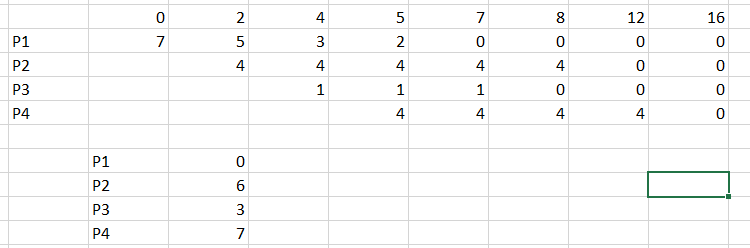
1. **Nếu khoảng thời gian quantum (time-slice) được sử dụng trong thuật toán lập lịch Round-robin nhiều hơn thời gian tối đa cần thiết để thực hiện bất kỳ tiến trình nào, thì thuật toán sẽ**
2. Trở thành Shortest job first
3. Trở thành First come first serve
4. Không có cái nào ở trên
5. Trở thành lập lịch Priority
6. **The systems which allow only one process execution at a time, are called \_\_\_\_**
7. uniprocessing systems
8. unithread systems
9. uniprogramming systems
10. unitasking systems
11. **Giao tiếp giữa các quá trình có thể được thực hiện thông qua \_\_\_\_\_\_**
12. messages
13. mails
14. system calls
15. traps
16. **Thuật toán lập lịch nào tối ưu để giảm thiểu thời gian chờ trung bình của 1 tập hợp tiến trình?**
17. FCFS
18. Shortest job first
19. Round robin
20. Priority
21. **Tiêu chí nào sau đây không phải là tiêu chí tối ưu hóa trong thiết kế thuật toán lập lịch CPU**
22. Minimum waiting time
23. Minimum turnaround time
24. Minimum CPU utilization
25. Maximum throughput

|  |
| --- |
| Note:  Tiêu chí tối ưu hóa trong thiết kế thuật toán lập lịch CPU:   * Maximum: CPU utilization, throughput * Minimum: TAT, WT, Response Time |

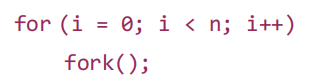
1. **Sử dụng thuật toán lập lịch SJF - non-preemptive cho các tiến trình theo thông tin dưới bảng. Thời gian chờ trung bình là bao nhiêu?**

****

1. 6
2. 7
3. 3
4. 4

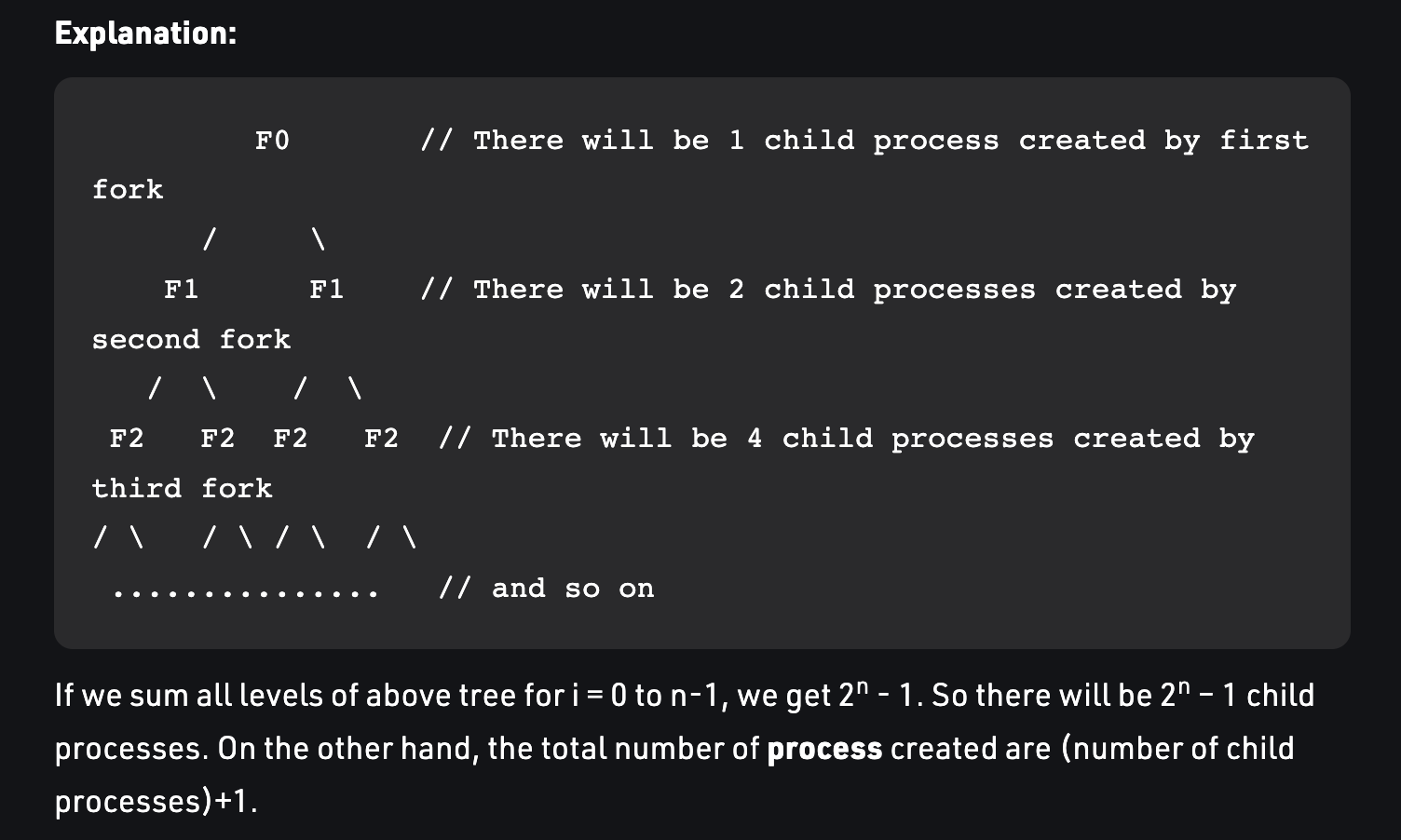


1. **When the next instruction to be executed by the current process, what provide the address of that instruction?**
2. CPU registers
3. Pipe
4. **Program counter**
5. Process stack
6. **A process stack does not contain \_\_\_\_**
7. Local variables
8. Function parameters
9. Return addresses
10. PID of child process
11. **A process executes the following code:**

****

**The total number of child processes created is:**

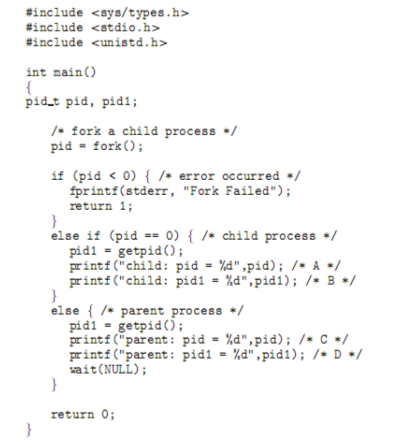
1. n
2. 2n - 1
3. 2n
4. 2(n+1) - 1



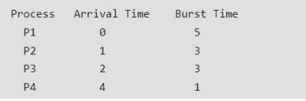
1. **Hệ điều hành thuộc dạng nào sẽ đọc và phản hồi ngay các yêu cầu?**
2. Time sharing time
3. Real time system
4. Quick response time
5. Batch system

|  |
| --- |
| **Real time system** được định nghĩa là một hệ thống xử lý dữ liệu trong đó khoảng thời gian cần thiết để xử lý và đáp ứng với đầu vào là nhỏ đến mức nó kiểm soát môi trường. Thời gian của hệ thống để đáp ứng với đầu vào và hiển thị thông tin cập nhật được yêu cầu được gọi là thời gian phản hồi. Vì vậy, trong phương pháp này, thời gian phản hồi rất ít so với xử lý trực tuyến. |

1. **Cái gì cung cấp (các) giao diện cho các dịch vụ của hệ điều hành?**
2. program
3. hardware
4. software
5. system call
6. **Using the programming below, identify the values of pid at lines A, B, C and D. (Assume that the actual pids of the parent and child are 2600 and 2603, respectively).**

****

1. A=0, B=2603, C=2603, D=2600
2. A=0, B=2600, C=2603, D=2600
3. **A=0, B=2603, C=2600, D=2600**
4. A=0, B=2600, C=2603, D=2603
5. **A multiprogrammed system:**
6. all programs in memory
7. several processes run simultaneously
8. several processes in memory simultaneously (Multiprogramming : Giữ nhiều chương trình trong bộ nhớ cùng 1 thời điểm.)
9. OS picks and begins to execute all processes
10. **Hệ điều hành sử dụng thuật toán lập lịch Shortest Remaining Time first (SRTF), tức SJF Preemptive. Xem xét thời gian đến và thời gian thực hiện cho các tiến trình sau:**

****

**Turnaround time trung bình cho các tiến trình này là bao nhiêu**

1. 5.75
2. 6.25
3. 6.00
4. 5.5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 4 | 5 | 8 | 12 |
| P1 | 5\* | 4 | 4 | 4 | 4 | 4\* | 0 |
| P2 |  | 3\* | 2\* | 0 | 0 | 0 | 0 |
| P3 |  |  | 3 | 3 | 3\* | 0 | 0 |
| P4 |  |  |  | 1\* | 0 | 0 | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P2 | P4 | P3 | P1 |

1 2 4 5 8 12

Thời gian tồn tại: trong tiến trình(**Turnaround time)**

P1: 12 **(0- đầu)**

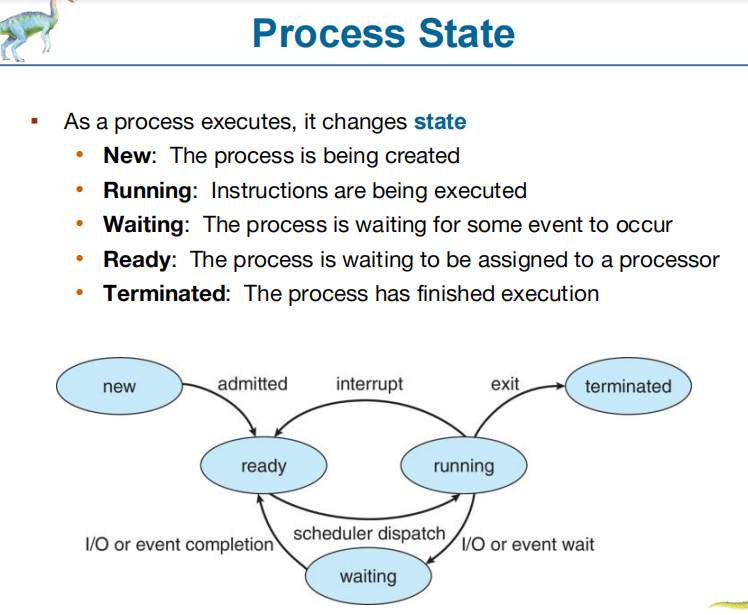
P2: 4-1 = 3

P3: 8 - 2 = 6

P4: 5 - 4 = 1

avg = (12+3+6+1)/4 = 5.5

1. **What is the action used to change the process from wait state to ready state?**
2. Interrupt
3. I/O or event complete
4. Scheduler
5. I/O or event wait



**41. CPU-bound process có nghĩa là**

a. Thời gian chiếm CPU nhiều hơn thời gian chờ trong hàng đợi

b. Thời gian thực hiện tác vụ I/O nhiều hơn thời gian chiếm CPU

c. Thời gian thực hiện tác vụ I/O ít hơn thời gian chiếm CPU

d. Thời gian thực hiện tác vụ I/O bằng thời gian chiếm CPU

**42. Which is used to implement the communication link in Client - Server system?**

a. socket

b. automatic

c. direct

d. pipe

**43. A process is the ready state, means:**

a. when process is using the CPU

b. pending alarm

c. when process is scheduled to run after some execution

d. when process is unable to run until some task has been completed

**44. Thao tác của Command Line Interface (CLI):**

a. Thao tác khác (Other operations).

b. Nhập lệnh + Tham số (Commands + Command line parameters).

c. Nhập lệnh + Tham số + ENTER (Commands+Command line parameters+ENTER).

d. Double click trên Icon (Double click on Icon) .

**45. Thành phần nào sau đây không thuộc Hệ điều hành.**

a. Hệ thống bảo vệ (Protection System).

b. Quản lý tiến trình (Process management).

c. Hệ thông dịch lệnh (Commander-Intpreter System)

d. Dịch vụ Rom Bios (Rom Bios device drivers).