# Họ và tên: Hà Phú Thịnh

# MSSV: 22521405

# BÀI TẬP CHƯƠNG 3

**Câu 1:** Một tiến trình bao gồm :

Trong bộ nhớ (memory) :

* Text Section (Program code) : Chứa những đoạn mã chương trình đã được biên dịch bởi compiler.
* Data Section (khu vực dữ liệu) : Chứa các biến toàn cục (global variables) và các biến tĩnh (static variables)
* Heap : Dùng để lưu trữ các bộ nhớ được cấp phát động.
* Stack : Dùng để lưu trữ các biến cục bộ (local variables).

**Câu 2*:*** Trạng thái của tiến trình :

* new : tiến trình vừa được tạo.
* ready : tiến trình đã có đủ tài nguyên, đang chờ được cấp CPU để chạy.
* running : các lệnh của tiến trình đang được thực thi.
* waiting: tiến trình đợi I/O hoàn tất.
* terminated: tiến trình đã kết thúc (đã thực thi xong).

Cách tiến trình chuyển trạng thái :

Đầu tiên, khi vừa khởi tạo, tiến trình sẽ ở trạng thái là new.

Thông qua bộ định thời dài hạn Long-term scheduling (hay còn gọi là bộ định thời công việc – Job Scheduler), tiến trình của chúng ta từ new sẽ được sắp xếp vị trí để “chui” vào trong hàng đợi ready.

Ở một số hệ điều hành có thêm bộ định thời Medium-term Scheduler. Thông qua bộ định thời này, các tiến trình sẽ được swap-out (chuyển tiến trình từ bộ nhớ chính sang bộ nhớ phụ) và swap-in (chuyển tiến trình từ bộ nhớ phụ vào bộ nhớ). Thông qua bộ định thời Short-term Scheduling (hay còn được gọi là Dispatcher), từ trạng thái ready tiến trình sẽ được sắp xếp để chuyển qua trạng thái running(là trạng thái chạy – hay trạng thái sử dụng CPU – của tiến trình).

Trong khi đang ở trạng thái running, có 3 trạng thái tiếp theo mà tiến trình có thể đạt được tiếp theo :

* waiting : Khi tiến trình đang chờ I/O (VD : Khi gọi hàm print(), scanf() trong C).
* ready : Khi tiến trình bị interrupt (bị ngắt, không cho chạy nữa) bởi Short- term Scheduler. Các lý do ngắt có thể là : Ngắt thời gian (Clock Interrupt), Ngắt ngoại vi (I/O Interrupt), Lời gọi hệ thống (Operating System Call), Signal.
* terminated : Khi ứng dụng thực thi xong : Khi gặp lệnh exit, khi thực thi lệnh cuối.

Trong khi ở trạng thái waiting, tiến trình sẽ chuyển sang trạng thái ready(vào hàng đợi ready) sau khi đã thực thi xong I/O.

**Câu 3*:*** Cộng tác giữa các tiến trình để:

* Chia sẻ dữ liệu (information sharing).
* Tăng tốc độ tính toán (computational speedup).
  + Các mạng lưới máy tính sẽ hợp với nhau để tạo thành các cluster.
  + Nếu hệ thống có nhiều CPU, chia công việc tính toán thành nhiều công việc tính toán nhỏ chạy song song.
* Thực hiện một công việc chung.
  + Xây dựng một phần mềm phức tạp bằng cách chia thành các module/process hợp tác nhau.

**Câu 4:**

Mỗi tiến trình trong hệ thống đều được cấp phát một Process Control Block (PCB). Là một trong các cấu trúc dữ liệu quan trọng nhất của hệ điều hành.

PCB chứa các thông tin liên quan đến process như :

* Trạng thái tiến trình (Process State) : new, ready, running,…
* Bộ đếm chương trình (Program Counter) : Chỉ đến địa chỉ của lệnh tiếp theo sẽ được thực thi cho tiến trình này.
* Các thanh ghi CPU (CPU Registers) : Phụ thuộc vào kiến trúc máy tính. Có thể kể đến vài loại như accumulators, index registers, stack pointers, general-purpose registers, condition-code information.
* Thông tin lập thời biểu CPU (CPU Scheduling Information) : Độ ưu tiên, con trỏ đến các hàng đợi, và các tham số của việc lập thời biểu.
* Thông tin quản lý bộ nhớ (Memory-Management Information) : Chứa page tables, segment tables, memory limits (giới hạn bộ nhớ).
* Thông tin trạng thái I/O (I/O status information) : Chứa danh sách các thiết bị I/O đã được cấp phát cho tiến trình, danh sách các file tiến trình đang mở,…
* Các thông tin quan trọng khác như : Lượng CPU, thời gian sử dụng,PID,…

**Câu 5:**

Tiểu trình : Là một đơn vị cơ bản sử dụng CPU, gồm :

* Thread ID.
* PC (Program Counter).
* Registers.
* Stack.
* Chia sẻ chung code, data, resources (file).

**Câu 6:**

Khi một tiến trình con được tạo ra từ tiến trình cha (từ câu lệnh fork()) thì tiến trình con sẽ được thực thi đến hết sau đó tiến trình cha mới thực thi câu lệnh tiếp theo.

Ngoài ra tiến trình cha còn có thể thực thi song song với tiến trình con.

**Câu 7:**

Trong quá trình thực thi thì tiến trình khi chạy từ chương trình trên đã trải qua những trạng thái như sau: new – ready – running – waiting – ready – running – waiting – ready – running – waiting – ready – running – waiting – ready – running – terminated

**Câu 8:**

Chương trình đã trải qua các trạng thái:

new-ready-running-waiting-ready-running-waiting-ready-running-terminated

**Câu 9:**

Chương trình đã trải qua các trạng thái:

new-ready-running-waiting-ready-running-waiting-ready-running-waiting-ready- running-waiting-ready-running-terminated

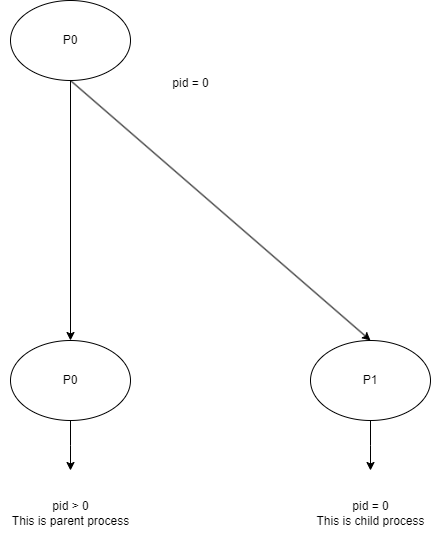
**Câu 10:**

Chương trình đã trải qua các trạng thái:

new-ready-running-waiting-ready-running-waiting-ready-running-waiting-ready- running-waiting-ready-running-terminated

**Câu 11:**

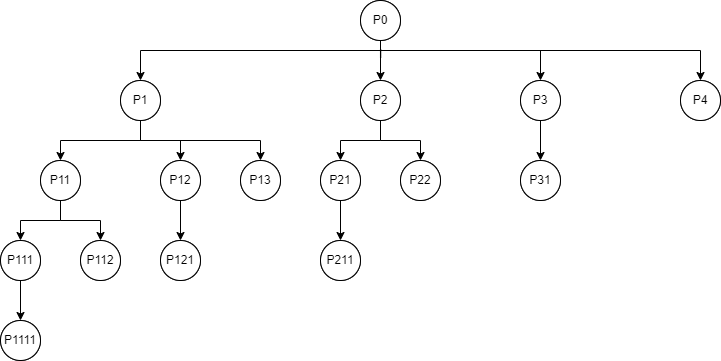
Khi chạy đoạn chương trình trên, khi chạy hết sẽ có 2 process được sinh ra bao gồm 1 tiến trình cha và 1 tiền trình con. Theo chương trình trên thì tiến trình cha sẽ in ra dòng chữ “This is parent process”; và tiến trình con sẽ in ra dòng chữ “This is child process”. Cây tiến trình khi thực thi đoạn chương trình trên như sau:



**Câu 12:**

Sau khi chạy hết chương trình trên thì có 16 tiến trình được sinh ra bao gồm 1 tiến

trình cha và 15 tiến trình con Cây tiến trình:

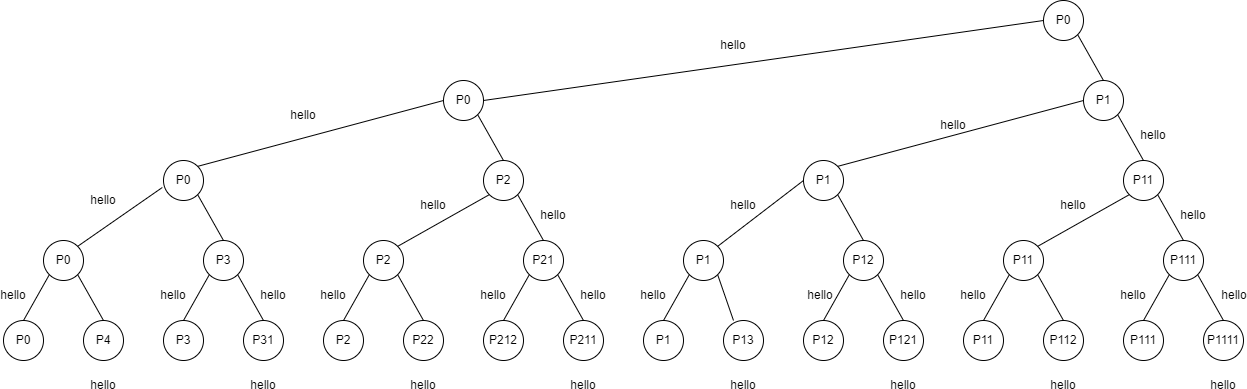


**Câu 13**

Chương trình sẽ in ra: (hello 30 lần)

hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello hello

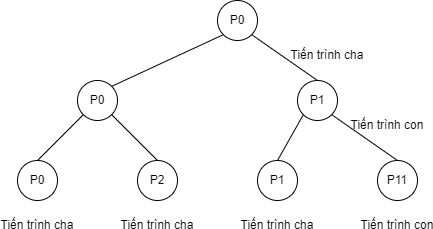
Cây tiến trình:



**Câu 14:**

Chương trình sẽ in ra: Tiến trình cha

Tiến trình con Tiến trình con Tiến trình cha Tiến trình cha Tiến trình cha Cây tiến trình:



**Câu 15:**

Ta có:

* Process cha: u = a – 5

v = &a

* Process con: x = a + 5

y = &a

Địa chỉ vật lý của biến a trong process cha và process con phải khác nhau. Nhưng chương trình lại truy cập bộ nhớ ảo (giả sử chương trình chạy trên hệ điều hành sử dụng bộ nhớ ảo). Nên tiến trình con sẽ sao chép tiến trình tiến trình và địa chỉ ảo trong biến a không thay đổi trong tiến trình con(vậy địa chỉ biến a của tiến trình cha và con là giống nhau).

Suy ra mối quan hệ giữa u, v và x, y: x = u + 10

v = y