**Hệ điều hành (Bài tập 2)**

**2312678 – Nguyễn Lê Bảo Long – CTK47A**

**1) Tìm hiểu, trình bày và cho ví dụ về hoạt động của tiến trình:**

**- Quá trình tạo mới tiến trình (Process Creation)**

Quá trình tạo mới tiến trình (Process Creation) là quá trình mà hệ điều hành tạo ra một tiến trình mới để thực hiện một tác vụ hoặc một công việc nhất định. Đây là giai đoạn khởi tạo một tiến trình mới, trong đó hệ điều hành cấp phát tài nguyên cho tiến trình mới, như bộ nhớ, quyền truy cập, thông tin về tiến trình, v.v.

Các bước tạo tiến trình có thể bao gồm:

 **Tạo bộ nhớ và tài nguyên**: Hệ điều hành cấp phát bộ nhớ cho tiến trình mới và cung cấp tài nguyên cần thiết như bộ xử lý, bộ nhớ, I/O, v.v.

 **Xác định ID tiến trình (PID)**: Mỗi tiến trình khi được tạo ra sẽ được cấp một ID duy nhất (PID), giúp hệ điều hành phân biệt các tiến trình với nhau.

 **Chọn kiểu tiến trình**: Tùy thuộc vào nhu cầu hệ thống, tiến trình có thể là tiến trình nền (background process) hoặc tiến trình tương tác (interactive process).

 **Khởi tạo các trạng thái và thông tin**: Hệ điều hành lưu trữ thông tin như trạng thái của tiến trình, các tệp mở, và các dữ liệu liên quan khác.

Ví dụ:

* Trong hệ điều hành Linux/Unix, khi một tiến trình cha gọi lệnh fork(), một tiến trình con sẽ được tạo ra. Tiến trình con này sẽ sao chép hầu hết các tài nguyên của tiến trình cha, nhưng có một PID mới và có thể thực thi độc lập.

**- Quá trình kết thúc tiến trình (Process Termination)**

Quá trình kết thúc tiến trình (Process Termination) là quá trình mà một tiến trình kết thúc công việc của mình và giải phóng tài nguyên mà nó đã sử dụng. Sau khi một tiến trình hoàn thành công việc của mình hoặc gặp lỗi nghiêm trọng, hệ điều hành sẽ thu hồi tài nguyên và xóa bỏ tiến trình khỏi bộ nhớ.

Quá trình kết thúc tiến trình bao gồm các bước:

* **Giải phóng tài nguyên**: Khi tiến trình kết thúc, tất cả tài nguyên (như bộ nhớ, tệp, kết nối mạng) mà tiến trình đã sử dụng sẽ được giải phóng, giúp tránh tình trạng tài nguyên bị chiếm dụng không cần thiết.
* **Trả lại mã trạng thái (exit status)**: Tiến trình có thể trả về mã trạng thái cho tiến trình cha để cho biết kết quả của quá trình thực thi. Mã này có thể là 0 (thành công) hoặc một giá trị khác (lỗi).
* **Xóa tiến trình khỏi bảng tiến trình**: Sau khi tiến trình kết thúc, hệ điều hành sẽ loại bỏ tiến trình khỏi bảng tiến trình và giải phóng mọi tài nguyên liên quan.

Ví dụ:

* Trong hệ điều hành Linux/Unix, khi một tiến trình gọi lệnh exit(), tiến trình đó sẽ kết thúc và hệ điều hành sẽ giải phóng tài nguyên của nó. Nếu tiến trình là tiến trình con, thì khi tiến trình cha thực hiện lệnh wait(), nó sẽ nhận kết quả trạng thái của tiến trình con.

**2) Tìm hiểu mô hình Shared memory của Interprocess Communication**

**- Khái niệm Interprocess Communication**

**Interprocess Communication (IPC)**, hay **Giao tiếp giữa các tiến trình**, là một tập hợp các kỹ thuật cho phép các tiến trình trong một hệ thống máy tính giao tiếp với nhau và chia sẻ dữ liệu. Trong một hệ thống đa tiến trình, mỗi tiến trình có không gian bộ nhớ riêng biệt, do đó chúng cần phải sử dụng IPC để trao đổi thông tin hoặc đồng bộ hóa công việc.

IPC có thể được thực hiện thông qua nhiều cơ chế khác nhau, bao gồm:

* **Shared Memory**: Các tiến trình chia sẻ một vùng bộ nhớ chung để đọc và ghi dữ liệu.
* **Message Passing**: Các tiến trình giao tiếp qua việc gửi và nhận thông điệp (thông qua các kênh hoặc hộp thư).
* **Pipes**: Dữ liệu được truyền từ tiến trình này sang tiến trình khác qua một kênh (pipe).
* **Semaphores**: Một phương pháp đồng bộ hóa giúp điều khiển quyền truy cập đến tài nguyên chia sẻ.

**- Bài toán Producer-Consumer**

**Producer-Consumer** là một bài toán mẫu trong lý thuyết đồng bộ hóa tiến trình. Trong bài toán này, có hai loại tiến trình:

* **Producer (nhà sản xuất)**: Tiến trình này sản xuất các dữ liệu hoặc hàng hóa và lưu chúng vào một bộ đệm chung (Shared Buffer).
* **Consumer (người tiêu dùng)**: Tiến trình này lấy các dữ liệu hoặc hàng hóa từ bộ đệm để xử lý.