**BÀI THỰC HÀNH BUỔI 3**

**BT1. Khởi tạo và Theo dõi Process**

1. Mô tả: Viết một chương trình C tạo một process con bằng cách sử dụng fork(). Sau khi tạo process, tiến trình con sẽ in ra PID của nó và tiến trình cha in ra PID của tiến trình con.

**BT2. Sử dụng exec và Biến Môi Trường**

1. Mô tả: Viết một chương trình trong đó tiến trình con sẽ thay thế chương trình đang chạy bằng một chương trình khác, ví dụ ls hoặc date, sử dụng hàm execlp() hoặc execvp().
2. Yêu cầu:
   * Truyền các biến môi trường vào chương trình và thực hiện các công việc khác nhau (ví dụ biến môi trường được truyền vào là 1 thì thực hiện lệnh ls, nếu là 2 thì thực hiện lệnh date).
   * Giải thích điều gì xảy ra với tiến trình ban đầu sau khi exec được gọi.

**BT3. Tín hiệu Process**

1. Mô tả: Viết một chương trình tạo hai process cha - con. Tiến trình cha sẽ gửi tín hiệu SIGUSR1 tới tiến trình con sau một khoảng thời gian.
2. Yêu cầu:
   * Sử dụng signal() để định nghĩa hành vi của tiến trình con khi nhận tín hiệu SIGUSR1.
   * In ra một thông báo khi tiến trình con nhận được tín hiệu.

**BT4. Ghi nhận Trạng thái Process**

1. Mô tả: Viết một chương trình tạo một tiến trình con, sau đó tiến trình cha sử dụng wait() để chờ tiến trình con kết thúc và ghi nhận trạng thái kết thúc của nó.
2. Yêu cầu:
   * Trong tiến trình con, chọn trạng thái kết thúc bằng cách gọi exit() với các giá trị khác nhau.
   * Sử dụng WIFEXITED() và WEXITSTATUS() để xác nhận và ghi nhận mã thoát của tiến trình con.

**BT5. Tạo Process Zombie và Orphan**

1. Mô tả: Viết một chương trình tạo ra một tiến trình zombie và một tiến trình orphan để hiểu về các loại tiến trình đặc biệt này.
2. Yêu cầu:
   * Để tạo zombie, cho tiến trình cha kết thúc sớm trong khi tiến trình con vẫn còn chạy, và sử dụng ps để quan sát trạng thái zombie của tiến trình.
   * Để tạo orphan, cho tiến trình con ngủ trong một khoảng thời gian dài hơn tiến trình cha, để tiến trình con trở thành orphan.
   * Giải thích trạng thái zombie và orphan, cùng lý do tại sao chúng xuất hiện trong Linux.