# BÀI 14: QUẢN LÝ TIẾN TRÌNH

## Process Management & Signals

# 🎯 Mục tiêu bài học

1. **Khái niệm:** Process ID (PID), Cha-Con (Parent-Child).
2. **System Calls:** `fork()`, `exec()`, `wait()`, `exit()`.
3. **Zombie Process:** Hiểu "Bóng ma" là gì và cách xử lý.
4. **Signal:** Xử lý tín hiệu ngắt (Ctrl+C) trong code C.

# 1. Process là gì?

Là một chương trình đang chạy. Mỗi process có một không gian bộ nhớ riêng (Virtual Memory).

- **PID 1 (Init):** Cha của mọi process.
- **Lệnh `ps`:** Xem danh sách process.
- **Lệnh `top`:** Xem tài nguyên (CPU/RAM) real-time.

# 2. Tạo tiến trình mới ( `fork` )

`fork()` tạo ra một bản sao y hệt của tiến trình cha.

```
pid_t pid = fork();
if (pid == 0) {
    // Đây là tiến trình CON (Child)
    printf("I am Child\n");
} else if (pid > 0) {
    // Đây là tiến trình CHA (Parent)
    printf("I am Parent, Child ID is %d\n", pid);
} else {
    // Lỗi
    perror("Fork failed");
}
```

# 3. Zombie Process (Tiến trình ma)

Khi con chết ( `exit` ) mà cha chưa kịp nhận xác ( `wait` ), con sẽ trở thành **Zombie** ( `Z+` trong `ps` ).

- Nó không tốn RAM/CPU, nhưng tốn 1 slot trong bảng PID.
- **Giải pháp:** Cha phải gọi `wait()` hoặc `waitpid()` để thu dọn tài nguyên của con.

# 4. Signals (Tín hiệu)

Cách Kernel báo cho Process biết có sự kiện (ví dụ User bấm Ctrl+C).

- `SIGINT` (2): Interrupt (Ctrl+C).
- `SIGKILL` (9): Giết ngay lập tức (Không thể chặn).
- `SIGTERM` (15): Yêu cầu dừng lịch sự.

**Xử lý Signal trong C:**

```c
#include <signal.h>

void my_handler(int signum) {
    printf("Nhan duoc tin hieu %d. Dang thoat...\n", signum);
    // Cleanup code (đóng file, tắt đèn...)
    exit(0);
}

int main() {
    signal(SIGINT, my_handler);  // Đăng ký hàm xử lý Ctrl+C
    while(1);  // Vòng lặp vô tận
}
```

# 🛠️ PHẦN THỰC HÀNH (LAB 14)

## App quản lý đa tiến trình

# Yêu cầu

Viết chương trình `my_shell` mô phỏng Terminal đơn giản:

1. Hiện dấu nhắc `MyShell>`.
2. Nhập lệnh (ví dụ `ls`, `date`).
3. Chương trình dùng `fork()` tạo con.
4. Con dùng `execlp()` để thực thi lệnh vừa nhập.
5. Cha dùng `wait()` chờ con chạy xong rồi hiện lại dấu nhắc.

# Q & A

## Hẹn gặp lại ở Bài 15: Đa luồng!