**CHƯƠNG 4 – OPERATING SYSTEMS SECURITY**

**An toàn trong hệ điều hành**

Tiêu chí CIA

C : Tính bí mật ( Confidentiality )

I : Tính toàn vẹn ( Integrity )

A : Tính khả dụng ( Availability)

**Thiết kế nhân hệ điều hành**

3 kiểu :

* Monolithic Kernel : tất cả các thành phần của hệ điều hành (quản lý bộ nhớ, hệ thống file, giao tiếp thiết bị…) được tích hợp trong cùng một không gian bộ nhớ (kernel space)
  + **Ưu điểm:**
    - **Hiệu suất cao:** Vì mọi thành phần hoạt động trong cùng một không gian bộ nhớ, việc gọi chức năng không yêu cầu chuyển đổi giữa không gian người dùng và kernel.
    - **Giao tiếp nhanh:** Gọi hệ thống nhanh hơn so với Microkernel vì không có chi phí giao tiếp giữa các module qua IPC (Inter-Process Communication).
    - **Quản lý tài nguyên tốt:** Do tích hợp mọi chức năng trong kernel, tài nguyên được quản lý tập trung.
  + **Nhược điểm:**
    - **Độ phức tạp cao:** Việc tích hợp tất cả chức năng trong một kernel lớn khiến kernel trở nên phức tạp và khó bảo trì.
    - **Khó mở rộng:** Thêm hoặc sửa đổi tính năng mới dễ gây lỗi toàn hệ thống.
    - **Tính bảo mật kém hơn:** Một lỗi nhỏ trong một thành phần có thể ảnh hưởng đến toàn bộ hệ thống vì tất cả hoạt động trong cùng không gian bộ nhớ.
* Microkernel :
  + chứa các thành phần cốt lõi (như quản lý bộ nhớ cơ bản, giao tiếp liên tiến trình, và quản lý thời gian).
  + Các thành phần khác (trình điều khiển thiết bị, hệ thống file, giao tiếp mạng) được tách ra và chạy trong không gian người dùng.
  + **Ưu điểm:**
    - **Tính mô-đun cao:** Các thành phần được tách biệt, dễ dàng thêm, sửa hoặc thay thế mà không ảnh hưởng đến toàn bộ kernel.
    - **Tính ổn định cao:** Một lỗi trong một module không ảnh hưởng đến toàn bộ hệ thống vì các module hoạt động độc lập.
    - **Tính bảo mật tốt hơn:** Hạn chế quyền truy cập kernel giúp giảm nguy cơ tấn công và lỗi hệ thống.
  + **Nhược điểm:**
    - **Hiệu suất thấp hơn:** Giao tiếp giữa các thành phần qua IPC chậm hơn so với gọi trực tiếp trong Monolithic Kernel.
    - **Thiết kế phức tạp:** Việc thiết kế giao tiếp giữa các module cần lập trình kỹ lưỡng và tốn nhiều thời gian hơn.
    - **Khó triển khai:** Microkernel yêu cầu cấu hình và triển khai phức tạp hơn vì các thành phần rời rạc.
* “ Hybrid kernel” :
  + kết hợp đặc điểm của cả Monolithic Kernel và Microkernel.
  + Kernel này giữ lại nhiều thành phần trong không gian kernel như Monolithic Kernel
  + Nhưng tách một số chức năng không quan trọng ra không gian người dùng giống Microkernel.
  + **Ưu điểm:**
    - **Hiệu suất tương đối cao:** Các thành phần cốt lõi vẫn hoạt động trong kernel space, giúp giảm độ trễ khi gọi chức năng.
    - **Dễ bảo trì:** Một số thành phần không quan trọng có thể được tách ra, giúp kernel dễ bảo trì và sửa lỗi.
    - **Tính linh hoạt:** Kết hợp giữa hiệu năng cao và khả năng mô-đun hóa, dễ dàng mở rộng chức năng hoặc thêm tính năng mới.
  + **Nhược điểm:**
    - **Độ phức tạp tăng:** Sự kết hợp giữa Monolithic và Microkernel khiến thiết kế Hybrid Kernel phức tạp hơn cả hai loại trên.
    - **Giao tiếp chậm hơn Monolithic Kernel:** Dù nhanh hơn Microkernel, việc giao tiếp giữa kernel và module bên ngoài vẫn không bằng Monolithic.
    - **Khả năng lỗi khi tích hợp:** Nếu tích hợp không tốt, dễ xảy ra xung đột hoặc lỗi hệ thống.

Read : 4

Write : 2

X ( excute ) : 1