NachOS – Another Completely Heuristic Operating System . là một phần mềm giả lập hệ điều hành với kiến trúc MIPS ( Million Instruction Per Second ) . Được viết bằng C++ cho MIPS , NachOS chạy như một user-process trên hệ điều hành máy chủ.

Các thành phần của NachOS :

* NachOS Machine : NachOS mô phỏng đại khái gần giống với kiến trúc MIPS, có thanh ghi , bộ nhớ và CPU . NachOS Machine được thực thi bởi Machine Object khi NachOS được khởi động, nó chứa các methods như Run, ReadRegister, WriteRegister ,… .
* NachOS Threads : Trong NachOS, một lớp Threads được định nghĩa và có thể có những trạng thái như là ready, running, blocked hoặc just created. Thread object cũng có chứa các methods như PutThreadToSleep, YieldCPU, ThreadFork, ThreadStackAllocate, … .
* NachOS UserPrograms : NachOS chạy chương trình người dung trong chính không gian bảo mật của chúng . NachOS có thể chạy bất kỳ tệp nhị phân MIPS nào . NachOS yêu cầu thực thi tệp phải ở định dạng “Noff” trong khi các tệp đa phần được lưu dưới dạng “a.out” hay còn gọi là định dạng “Coff”.

Hiện nay , các hệ điều hành của máy tính “hiện đại” thường tách riêng bộ nhớ ảo thành 2 khu vực : không gian nhân (kernel space) và không gian người dùng (user space) . Sự tách biệt này nhằm mục đích bảo vệ bộ nhớ và phần cứng khỏi những phần mềm hoặc lỗi độc hại .

|  |  |
| --- | --- |
| Kernel | User |
| Chạy nhân hệ điều hành đặc quyền , phần mở rộng của nhân và đa số các drivers. | Vùng bộ nhớ chỉ để chạy ứng dụng phần mềm hoặc để thực thi một số drivers. |
| Mã thực thi có quyền truy cập vào bất kỳ không gian địa chỉ nào của memory và phần cứng. | Mã thực thi bị giới hạn truy cập và không thể truy cập trực tiếp đến kernel space được. (Truy cập gián tiếp qua một phần nhỏ của kernel gọi là the system call). |
| Bất kỳ sự “ không ổn định “ nào bên trong mã thực thi kernel đều có thể dẫn đến lỗi hệ thống hoàn toàn. | Nếu có xảy ra trục trặc hoặc vấn đề thì đều có thể khôi phục được vì truy cập bị hạn chế và giới hạn trong không gian hệ thống mà user đang hoạt động. |

Kernel space code có đặc tính để run trong kernel mode, gọi là mã thực thi Ring 0.

Điển hình trong kiến trúc x86 có 4 Ring :

* Ring 0 : Kernel Mode. ( Tương tác trực tiếp với phần cứng vật lý như CPU hoặc bộ nhớ )
* Ring 1 : Có thể sử dụng bởi máy ảo hoặc drivers.
* Ring 2 : Sử dụng bởi drivers.
* Ring 3 : Là những ứng dụng.

Đặc quyền sắp xếp từ thấp đến cao : Ring 3 > Ring 2 > Ring 1 > Ring 0 .

Riêng Linux chỉ sử dụng 2 Ring là : Ring 0 ( cho Kernel ) và Ring 3 ( cho User )