Clase 2
26.08.2022

Introducción II

> **Palabras Claves**: Sistema Operativo, Hardware, System Call, Interacción, Modos de Ejecución, Protección, Kernel

Funciones principales de un Sistema Operativo

Resumiendo de antes, las funciones de un sistema operativo son:

- Brindar abstracciones de alto nivel a los procesos de usuario.
- Administrar eficientemente el uso de la CPU.
- Administrar eficientemente el uso de la memoria.
- Brindar asistencia al proceso de entrada/salida.

Problemas que un sistema operativo (el kernel) debe evitar

Básicamente son tres cosas las que debe evitar:

- Qué un proceso de apropie de la CPU
- Que un proceso intente ejecutar instrucciones privileagiadas (ejemplo: entrada/salida)
- Que un proceso intente acceder a una posición de memoria fuera de su espacio permitido. **Hay que proteger los espacios de dirección, es decir, que ninguno burle los espacios de dirección.**

Para evitar estos tres problemas, el sistema operativo debe gestionar **y controlar** el uso de la CPU, detectar los intentos de ejecución de instrucciones privilegiadas, los accesos ilegales a memoria y **proteger el vector de interrupciones**, así como las *rutinas de atención de interrupciones.* Esto es mucho muy importante.

> **Las interrupciones rompen el ciclo de intrucciones.**

Para todo esto, las aquitecturas brindan diferente **apoyo por hardware** para no perder al sistema operativo.

El apoyo del Hardware

El **hardware** da apoyo bajo tres ejes:

- Los **modos de ejecución**: define limitaciones en el conjunto de intrucciones que se puede ejecutar en cada modo. Se habla de 4 anillos o 2 anillos, el más conocido es este último dónde se representan al **modo usuario y al modo kernel.***
- La **interrupción de Clock**, que evitan que un proceso se apropie de la CPU. Básicamente cada cierto tiempo el sistema operativo aparece en la CPU para gestionar los recursos y fijarse como va todo.

- La **protección de la memoria**, dónde se definen límites de memoria a los que cada proceso puede acceder (un registro base y uno límite).

Modos de ejecución: Modo usuario, modo Kernel.

Hay un **bit** en el CPU (en el controlador) que indica cuál es el modo de ejecución actual.

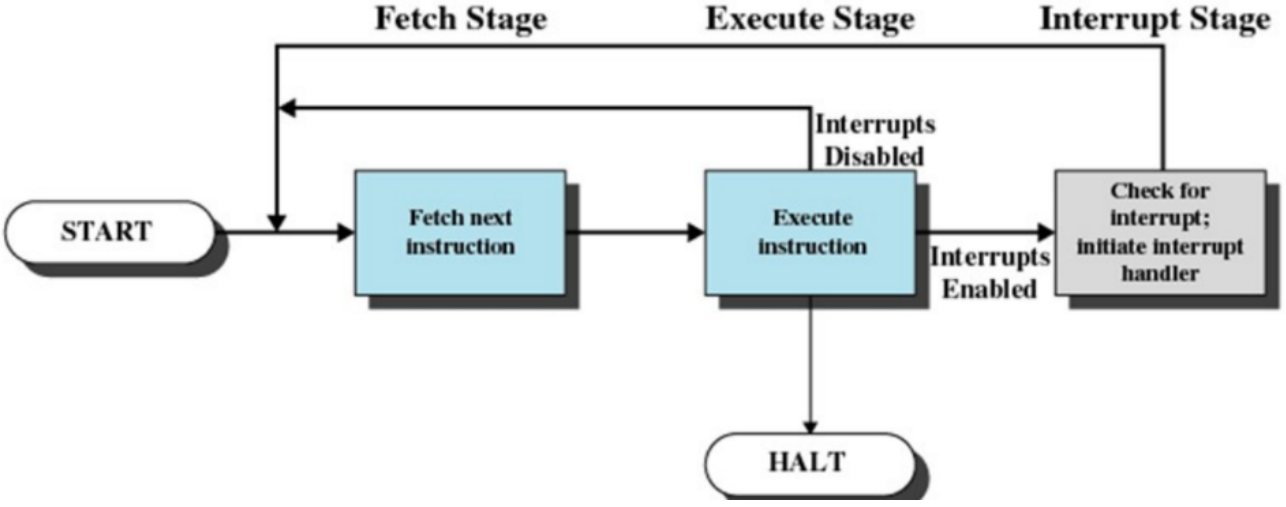
Las instrucciones privilegiadas (que necesitan acceder a estructurar del Kernel o deben ejecutar código que no es del proceso) deben ejecutarse en modo **supervisor o modo KERNEL**.
-> En este modo se ejecuta la porción del sistema operativo que corresponde al kernel.

El modo **usuario** es aquel en dónde el proceso puede acceder *sólo* a su espacio de direcciones (es decir, las direcciones propias).
->En este modo se ejecuta el resto del sistema operativo y los programas de usuarios.

3 situaciones de cambio en el bit:

- Cuándo se **arranca el sistema operativo**, este lo hace con **el bit en modo supervisor**.
- Cada vez que comienza a **ejecutarse un proceso de usuario**, este **bit se pone en modo usuario,** mediante una intrucción especial.
- Cuándo hay un **trap o una interrupción** el bit de modo se pone en **modo Kernel**.
  - ***Esta es la única forma de pasar a modo Kernel***
  - No es el proceso de usuario quién hace el cambio explícitamente, es la interrupción.
  - **La exepción ocurre en compilación y la interrupción es externa al coso.**

¿Cómo funciona esto último? Cuándo el **proceso de usuario intenta**, por sí mismo, ejecutar intrucciones que pueden causar problemas (las llamadas a instrucciones privilegiadas) el **hardware** lo detecta como una **operación ilegal** y produce un **trap al SO**.



Casos curiosos:

- **En WIN2000** el modo núcleo ejecuta los servicios ejecutivos. El modo usuario ejecuta los procesos de usuario. Cuando un programa se bloquea en modo usuario, a lo sumo se escribe un suceso en el registro de sucesos. **Si el bloqueo se produce estando en modo supervisor se genera la** BSOD **(pantalla azul de la muerte).** -> ¿Por qué? Porque no hay un super-kernel o alguien que se encarge de manejar lo que está pasando. En estos casos se agarra, se vuelca todo lo que está pasando en un archivo y se lanza pantalla azul.

- **Intel 8088** no tenía modo dual de operación ni protección por hardware.
- **MsDos** las aplicaciones pueden acceder directamente a las funciones básicas de entrada/salida para escribir directamente en pantalla o en disco.

Resumiendo

Modo usuario:

- Debug de procesos, definición de protocolos de comunicación gestión de aplicaciones (compilador, editor, aplicaciones de usuario.
- En este modo se llevan a cabo todas las tareas que no requieran accesos privilegiados
- En este modo no se puede interactuar con el hardware
- El proceso trabaja en su propio espacio de direcciones

Ocurre una interrupción y se pasa al:
Modo kernel:

- Gestión de procesos: Creación y terminación , planificación, intercambio, sincronización y soporte para la comunicación entre procesos
- Gestión de memoria: Reserva de espacio de direcciones para los procesos, Swapping, Gestión y páginas de segmentos
- Gestión E/S: Gestión de buffers, reserva de canales de E/S y dedispositivos de los procesos
- Funciones de soporte: Gestión de interrupciones, auditoría, monitoreo.

> Dato de color: los drivers se ejecutan en modo kernel. El kernel se va reconstruyendo con los drivers instalados.

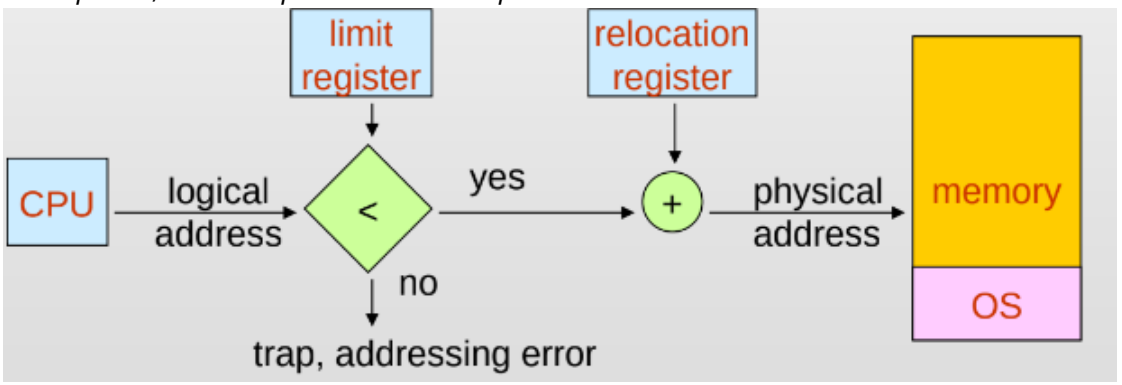
Protección de la memoria

> Delimitar el espacipo de direcciones del proceso.

Por cada programa que se ejecuta en memoria, se crea una "cápsula" dónde el programa sabe que de *pe* a *pa* le corresponde un cacho de la RAM. Es el **kernel** (mediante intrucciones privilegiadas) quién se encarga de poner limites a las direcciones que pueden utilizar un proceso (es decir: poner un registro base y uno límite)

La **memoria principal aloja al sistema operativo** y a los procesos de usuario. El kernel debe proteger el espacio de direcciones para que los procesos de usuario *no puedan acceder a dónde no les*

corresponde, inclusive pisando así otros procesos.



Las direcciones pueden ser **lógicas** o **físicas**. La dirección lógica es aquella que usa un proceso, la dirección física es ese lugarcito en RAM. Las direcciones lógicas son importantes para no tener que ir "poniéndose" de acuerdo entre programas que espacio utilizar, así el sistema operativo el que guarda los procesos entre **una base y un límite;** el procesador chequeea que esa dirección sea lógica y lo suma a un registro de relación, que es un registro en la RAM.

Protección de E/S

Las instrucciones de entrada y salida se definen como **privilegiadas**, así que deben ejecutarse en modo kernel. Los procesos de usuario realizan E/S mediante las llamadas al sistema operativo (SO es un servicio aquí), para que el kernel gestione estas entradas y salidas.

Protección del CPU

Es el CPU quien maneja **la interrupción por clock**, para evitar *ser* apropiado por algún proceso. Esto se implementa normalmente a través de un clock y un contador, es el Kernel quién le da valor al contador que se decrementa con cada tick del reloj, y al llegar a cero puede expulsar al proceso para ejecutar otro.

- Las intrucciones que modifican el funcionamiento del reloj son privilegiadas.
- Se le asigna al contador el valor que se quiera.
- Se usa también para el cálculo de la hora actual, basándone en cantidad de interrupciones ocurridas cada *tanto* tiempo y desde una fecha y hora determinadas.