

1. Cuestiones generales relacionadas con un SO:

- ¿Qué es el núcleo (*kernel*) de un SO?
- ¿Qué es un modelo de memoria (interpretación del espacio de memoria) para un programa? Explique los diferentes modelos de memoria para la arquitectura IA-32.
- ¿Cómo funciona el mecanismo de tratamiento de interrupciones mediante interrupciones vectorizadas? Explique que parte es realizada por el hardware y que parte por el software.
- Describa detalladamente los pasos que lleva a cabo el SO cuando un programa solicita una llamada al sistema.

○ *El núcleo de un sistema operativo es la parte del sistema que obtiene todos los privilegios frente a la ejecución de programas, ofreciendo servicios. Con la vez que un programa necesita ejecutar código de SO lo hace en este modo presentando así el mínimo de impedimentos posibles; por ejemplo, el código de user no*

○ *Un modelo de memoria es la forma en que está organizada la memoria con respecto a los programas de usuario la gestión de la memoria se ve recodificada según el modelo usado anteriormente, según el modelo el espacio de direcciones será mayor o menor.*

Los modelos son:

- Flat → espacio de memoria = nº direcciones
- Segmented → espacio de memoria = nº de segmentos usados · desplaz. de cada segmento
- Real address → igual que segmented pero con restricción de tamaño.

○ El esquema es el siguiente.

Hardware

- Produce la interrupción
- CPU termina la instrucción
- Comprueba si hay interrupciones
- Verifica que la hay
- Cambia a modo kernel

Software

- Guardamos PC, PSW en la pila.
- Cargamos PC con la dirección del vector de interrupciones correspondiente
- Salvamos los registros del programa actual en el PCB y bloqueamos el proceso
- Resolvemos la interrupción

- Recuperarlos PC y PSW
- Cambiarlos a modo usuario
- Continuar la ejecución del proceso

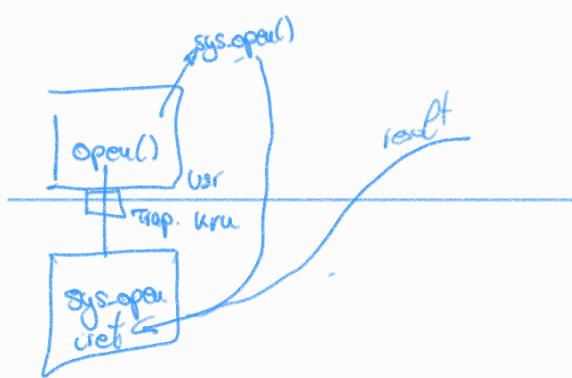
El vector de interrupciones es una estructura del sistema que guarda en cada casilla la dirección de memoria de la instrucción inicial de cada ISR.

- o) Cuando se produce una llamada al sistema, si es bloqueante, el proceso que estaba ejecutándose pasa a estar bloqueado, esto se hace mediante un cambio de contexto, cuando el sistema ha resuelto la llamada (pasa a kernel, toma la información del gestor general de llamadas, resuelve, vuelve a usuario), el proceso que estaba bloqueado pasa a listo y podrá volver a ejecutarse.

Context switch:

- Salvar registros
- PCB a cola de bloqueados
- Planificador pasa de la cola de listos
- Se ejecuta de nuevo el dispatcher que es el encargado de hacer las dos primeras pasos y cargar los nuevos registros en la CPU

Ejemplo:



2. Explique tres responsabilidades asignadas al gestor de memoria de un SO y tres asignadas al gestor de procesos.

Gestor de memoria

- Debe proporcionar la protección de las zonas de memoria que lo necesiten, es decir, no permitir que se manipulen zonas de memoria con código ajeno del SO;
 - o para que un proceso no entre a la zona de memoria de otro
- Debe gestionar la zona de memoria compartida entre procesos, es decir, determinar la cantidad y qué procesos pueden hacer qué; un ejemplo es un acceso concurrente.
- Debe gestionar la asignación y liberación de memoria así como la cantidad a asignar de la misma. Si no estuviera habría un despliegue de superposiciones de escritura; es decir, interferencias.

Gestor de procesos

- Gestionar la creación de los usuarios, es decir, impedir la creación de más usuarios con valores basura
- Gestionar la eliminación de los usuarios liberando todo su memoria ocupada.
- Gestionar la comunicación y sincronización de los usuarios mediante uso de las zonas de memoria compartidas gestionadas por el gestor de memoria

3. ¿Cómo gestionaría el sistema operativo la posibilidad de anidamiento de interrupciones?

Para el SO, una interrupción no es más que un proceso más, pero que se ejecuta en modo kernel. Por tanto, una interrupción dentro de otra se resolvería como si un proceso tuviera una interrupción; sin embargo, no habría cambios de modo y la interrupción provocante tendría mayor prioridad de ejecución en la cola de listas.

4. Contraste las ventajas e inconvenientes de una arquitectura de SO monolítica frente a una arquitectura microkernel.

Monolítico

Inconvenientes:

- La caída de un módulo provoca la caída del sistema. En sistemas microkernel esto no es así, sólo se caerá una parte del mismo y se salvaría llevando el proceso de servicio.

- Dependencias entre módulos complejos que no ocultan información
- Estas dependencias pueden confundir al leer o interpretar código

Microkernel:

Ventajas:

- Fiabilidad: explicado arriba
- Extensibilidad: como se pueden cargar varios programadores, se admite la inclusión de nuevos servicios como procesos de usr.

Inconvenientes:

- Peor rendimiento. Debido a que para obtener un servicio es necesario hacer múltiples cambios de modo y de espacios de direcciones.

5. Cuestiones relacionadas con virtualización:

- ¿Qué se entiende actualmente por virtualización mediante hipervisor?
- ¿Qué clases de hipervisores existen de manera general y qué ventajas e inconvenientes plantea una clase con respecto a la otra?

• Cuando nos referimos a virtualización mediante hipervisor nos referimos a una virtualización asistida por HW, es decir, tenemos un HW que permite la virtualización donde cada elemento HW puede ser proporcionado por el hipervisor al SO invitado. Este hipervisor es tipo un host y debe proporcionar:

- Almacenamiento independiente de SO para proporcionar recursos al VM
- Switching de VE's para asignar recursos gestionados por el hipervisor.

Es decir, la virtualización es una capa software (firmware hardware) que está situada entre el hardware real y el SO invitado y que proporciona un entorno de trabajo (VM) para ejecutar el SO invitado.

o) Native: Esté implementado sobre el hardware y proporciona los recursos H/w virtualizados a la máquina virtual.

Es mucho más rápido que el anterior, pero como inconveniente muestra ocupado todo el hardware.

Invitado: Consiste en que nuestro hipervisor se ejecuta sobre el so real como un programa de usuario más y nuestro sistema operativo invitado va a usar la abstracción proporcionada por el hipervisor para trabajar.

Como inconveniente es mucho más lento que el tipo native, ya que hay múltiples niveles de abstracción entre so invitado y hardware real luego no permite un rendimiento alto de la máquina.

Como ventaja, no ocupa todo el computador luego no es necesario tener hasta la máquina dedicada a la virtualización.

6. Cuestiones relacionadas con RTOS:

- o ¿Qué característica distingue esencialmente a un proceso de tiempo real de otro que no lo es?
- o ¿Cuáles son los factores determinantes del tiempo de respuesta en un RTOS, e.d. define determinismo y reactividad?

o) Los procesos de tiempo real son aquellos que deben ser resueltos en un intervalo de tiempo; para ello tenemos concretamente los subintervalos, el primero que hace referencia a la tasa de datos, que puede ser periódica o no, y que si no se cumple se ejecutará en el segundo intervalo correspondiente a la ejecución. Si alguno de ellos no se cumple, se desechará dicho programa.

Un ejemplo de ellos son los sistemas aeronáuticos.

o) Los factores determinantes del tiempo de respuesta son el determinismo y la reactividad. Ambos afectan a cuántos procesos se resuelven correctamente de todos los que se generen:

- Determinismo: tiempo que tarda el R/S en captar que ha habido una interrupción.

- Reactividad: tiempo que tarda el R/S en ejecutar la ISR y recuperar la interrupción.

Si ambos son altos, es muy probable que el sistema sea poco estable.