RELACIÓN SO: TEMA 1

Autor: Daniel Pérez Ruiz

1. Cuestiones generales relacionadas con un SO:

- *¿Qué es el núcleo (kernel) de un SO?*:
    - El *kernel* es una parte del SO que contiene las funciones del sistema operativo más frecuentemente usadas, y en cierto momento, otras porciones del sistema operativo actualmente en uso.
- *¿Qué es un modelo de memoria para un programa? Explique los diferentes modelos de memoria para la arquitectura IA-32.*
    - Un modelo de memoria describe las interacciones de los hilos a través de la memoria y su uso compartido de los datos.
    - Para la ia 32 son el flat memory model, segmented memory model y real-adress mode memory model:
        - **Flat memory model.** Es un espacio lineal con direcciones consecutivas en el rango [0,2^32-1] (linear address space). Permite direccionar con granularidad de un byte.
        - **Segmented memory model.** Los programas ven el espacio de memoria como un grupo de espacios independientes llamados segmentos.
        - **Real-address mode memory model.**Proporcionado por compatibilidad hacia atrás con el procesador 8086.
- *¿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*:
    - Se mantiene una tabla de vectores de interrupción (direcciones de comienzo de las distintas RSI) y a cada interrupción se le asocia un número que será el índice por el cual se accederá a la tabla y se recuperará la información de la dirección de comienzo.
    - Necesita señales de conformidad o handshaking para sincronizar al procesador con la interfaz, ya que esta última tiene que indicarle al procesador cuando va a enviarle el índice que necesita para buscar el vector de interrupción (INT) y el procesador deberá enviar otra señal para indicar que se ha reconocido la interrupción (INTA#).
    - *PARTES REALIZADAS POR EL HARDWARE:*
        - ↪ [El controlador de dispositivo u otro sistema hardware genera una interrupción] ↪
        - ↪ [El procesador termina la ejecución de la instrucción actual] ↪
        - ↪ [El procesador indica el reconocimiento de la interrupción] ↪
        - ↪ [El procesador apila el PSW y el PC en la pila de control] ↪
        - ↪ [El procesador carga un nuevo valor en el PC basado en la interrupción] ↪
    - *PARTES REALIZADAS POR EL SOFTWARE*:
        - ↪ [Salva el resto de la información del estado del proceso] ↪
        - ↪ [Procesa la interrupción] ↪
        - ↪ [Restaura la información del estado del proceso] ↪
        - ↪ [Restaura los antiguos PSW y PC]

- *Describa detalladamente los pasos que lleva a cabo el SO cuando un programa solicita una llamada al sistema.*
    - Se asocia un número con cada llamada al sistema proporcionada por el SO.
    - El interfaz de llamadas invoca la llamada requerida por el programador y devuelve el estado de finalización y los valores de retorno.
    - Paso de parámetros:
        - En registros de la CPU.
        - Parámetros en memoria paso la dirección del bloque en un registro.
        - Parámetros en una pila.

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

- *RESPONSABILIDADES ASIGNADAS AL GESTOR DE MEMORIA*:
    - **Aislamiento de procesos.** El sistema operativo debe evitar que los procesos independientes interfieran en la memoria de otro proceso, tanto datos como instrucciones.
    - **Asignación y gestión automática.** Los programas deben tener una asignación dinámica de memoria por demanda, en cualquier nivel de la jerarquía de memoria. La asignación debe ser transparente al programador. Por tanto, el programador no debe preocuparse de aspectos rela- cionados con limitaciones de memoria, y el sistema operativo puede lograr incrementar la efi- ciencia, asignando memoria a los trabajos sólo cuando se necesiten.
    - **Soporte a la programación modular.** Los programadores deben ser capaces de definir módulos de programación y crear, destruir, y alterar el tamaño de los módulos dinámicamente.
    - **Protección y control de acceso.** La compartición de memoria, en cualquier nivel de la jerarquía de memoria, permite que un programa direccione un espacio de memoria de otro proceso. Esto es deseable cuando se necesita la compartición por parte de determinadas aplicaciones. Otras veces, esta característica amenaza la integridad de los programas e incluso del propio sistema operativo. El sistema operativo debe permitir que varios usuarios puedan acceder de distintas formas a porciones de memoria.
    - **Almacenamiento a largo plazo.** Muchas aplicaciones requieren formas de almacenar la información durante largos periodos de tiempo, después de que el computador se haya apagado.

- *RESPONSABILIADADES ASIGNADAS AL GESTOR DE PROCESOS:*
    - **Inapropiada sincronización.** Es frecuente el hecho de que una rutina se suspenda esperando por algún evento en el sistema. Por ejemplo, un programa que inicia una lectura de E/S debe esperar hasta que los datos estén disponibles en un buffer antes de proceder. En este caso, se necesita una señal procedente de otra rutina. El diseño inapropiado del mecanismo de señali- zación puede provocar que las señales se pierdan o se reciban señales duplicadas.

- **Violación de la exclusión mutua.** Frecuentemente, más de un programa o usuario intentan hacer uso de recursos compartidos simultáneamente. Por ejemplo, dos usuarios podrían inten- tar editar el mismo fichero a la vez. Si estos accesos no se controlan, podría ocurrir un error. Debe existir algún tipo de mecanismo de exclusión mutua que permita que sólo una rutina en un momento determinado actualice un fichero. Es difícil verificar que la implementación de la exclusión mutua es correcta en todas las posibles secuencias de eventos.

- **Operación no determinista de un programa.** Los resultados de un programa particular nor- malmente dependen sólo de la entrada a dicho programa y no de las actividades de otro pro- grama en un sistema compartido. Pero cuando los programas comparten memoria, y sus ejecu- ciones son entrelazadas por el procesador, podrían interferir entre ellos, sobreescribiendo zonas de memoria comunes de una forma impredecible. Por tanto, el orden en el que diversos programas se planifican puede afectar a la salida de cualquier programa particular.

- **Interbloqueos.** Es posible que dos o más programas se queden bloqueados esperándose entre sí. Por ejemplo, dos programas podrían requerir dos dispositivos de E/S para llevar a cabo una determinada operación (por ejemplo, una copia de un disco o una cinta). Uno de los progra- mas ha tomado control de uno de los dispositivos y el otro programa tiene control del otro dis- positivo. Cada uno de ellos está esperando a que el otro programa libere el recurso que no po- seen. Dicho interbloqueo puede depender de la temporización de la asignación y liberación de recursos.

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

- Arquitectura monolítica:
  - Explicación:
    - Las dependencias entre los distintos módulos son complejas salvo para algunos elementos bien establecidos (¡NO oculta información!).
    - El modelo de obtención de servicios es la llamada a procedimiento.
  - Problemas:
    - La fuerte dependencia entre módulos provoca dificultades a la hora de comprender el código y de realizar modificaciones.
    - Al ser un solo programa cargado en memoria el fallo de un módulo puede provocar la caida del sistema.
- Arquitectura microkernel:
  - Ventajas:
    - Fiabilidad. Un error en un módulo que se implemente como proceso de usuario solo provoca una caida parcial del sistema que puede recuperarse levantando el proceso de servicio.
    - Extensibilidad. Podemos incluir nuevos servicios como procesos de usuario.
  - Problemas:
    - Peor rendimiento que la arquitectura monolítica porque para obtener un servicio se realizan más cambios de modo y de espacio de direcciones.

4. Cuestiones relacionas con virtualización:

- ¿Qué se entiende actualmente por virtualización mediante hipervisor?
    - Un **VMM (Virtual Machine Monitor)** o **hipervisor** es el software que proporciona la abstracción de la máquina real a las VMs.
- ¿Qué clases de hipervisores existen de manera general y que ventajas e inconvenientes plantea una clase con respecto a la otra?
    - **Tipo 1**, native o bare-metal. Se ejecutan directamente en el host HW para proporcionar VM a los SO invitados. Ej. Xen y Vmware ESX/ESXi.
    - **Tipo 2**, hosted. Se ejecutan sobre un SO convencional como el resto de programas y el SO invitado se ejecuta sobre la abstracción proporcionada por el hipervisor. Ej. VMware Workstation y VirtualBox.

5. Cuestiones relacionadas con RTOS:

- ¿Qué característica distingue esencialmente a un proceso de tiempo real de otro que no lo es?
    - Estos procesos se ejecutan en "tiempo real", por lo que tienen un tiempo límite de ejecución. No basta sólo con que la operación sea correcta.
- ¿Cuáles son los factores determinantes del tiempo de respuesta en un RTOS, e.d. define determinismo y reactividad?
    - **Determinismo**: Realiza operaciones en instantes de tiempo jos predeterminados o dentro de intervalos de tiempo predeterminados. En un sistema operativo de tiempo real, las solicitudes de servicio de los procesos son dirigidas por eventos externos y temporizaciones.
    - **Reactividad**: Se preocupa de cuánto tiempo tarda el sistema operativo, después del reconocimiento, en servir la interrupción. La reactividad incluye los siguientes aspectos:
        1. La cantidad de tiempo necesario para manejar inicialmente la interrupción y comenzar a eje- cutar la rutina de servicio de la interrupción (RSI). Si la ejecución de la RSI necesita un cam- bio de proceso, entonces el retardo será mayor que si la RSI puede ser ejecutada dentro del contexto del proceso actual.
        2. La cantidad de tiempo necesario para realizar la RSI. Esto depende, generalmente, de la plata- forma hardware.
        3. El efecto del anidamiento de interrupciones. Si una RSI puede ser interrumpida por la llegada de otra interrupción, entonces el servicio se retrasará.