**Sistema Operativo**

Es software: Necesita procesador y memoria para ejecutarse

· Gestiona el HW

· Controla la ejecución de los procesos

· Interfaz entre aplicaciones y HW

· Actúa como intermediario entre un usuario de una computadora y el HW de la misma

**Objetivos de los S.O**:

· Comodidad: Hacer más fácil el uso del hardware (PC, servidor, switch, router, controlador específico)

· Eficiencia: Hacer un uso más eficiente de los recursos del sistema

· Evolución: Permitir la introducción de nuevas funciones al sistema sin interferir con funciones anteriores

**Componentes de un SO**

Kernel, Shell (GUI / CUI o CLI), Herramientas (Editores, Compiladores, Librerías, etc).

**Kernel** (Núcleo)

· “Porción de código”

· Que se encuentra en memoria principal

· Que se encarga de la administración de los recursos.

Implementa servicios esenciales:

· Manejo de memoria

· Manejo de la CPU

· Administración de procesos

· Comunicación y Concurrencia

· Gestión de la E/S

**Servicios de un SO**:

· Administración y planificación del procesador

o Multiplexación de la carga de trabajo

o Imparcialidad, “justicia” en la ejecución

o (Fairness)

o Que no haya bloqueos

o Manejo de Prioridades

· Administración de Memoria

o Administración de memoria eficientemente

o Memoria física vs memoria virtual. Jerarquías de memoria

o Protección de programas que compiten o se ejecutan concurrentemente

· Administración del almacenamiento–Sistema de archivos

o Acceso a medios de almacenamiento externos

· Administración de dispositivos

o Ocultamiento de dependencias de HW

o Administración de accesos simultáneos

· Detección de errores y respuestas

o Errores de HW internos y Externos

§ Errores de Memoria/CPU

§ Errores de Dispositivos

o Errores de SW

§ Errores Aritméticos

§ Acceso no permitido a direcciones de memoria

o Incapacidad del SO para conceder una solicitud de una aplicación

· Interacción del Usuario (Shell)

· Contabilidad

o Recoger estadísticas del uso

o Monitorear parámetros de rendimiento

o Anticipar necesidades de mejoras futuras

o Dar elementos si es necesario facturar tiempo de procesamiento

**Problemas que un SO debe evitar**:

· Que un proceso se apropie de la CPU, gestionando el uso de la CPU

· Que un proceso intente ejecutar instrucciones de E/S ilegales, detectándolos.

· Que un proceso intente acceder a una posición de memoria fuera de su espacio declarado, detectando accesos a espacios ilegales de memoria.

– Proteger los espacios de direcciones y el vector de interrupciones (como por ejemplo las rutinas de atención de interrupciones)

**Apoyo del hardware**:

· Modos de Ejecución: Define limitaciones en el conjunto de instrucciones que se puede ejecutar en cada modo

· Interrupción de Clock: Se debe evitar que un proceso se apropie de la CPU

· Protección de la Memoria: Se deben definir límites de memoria a los que puede acceder cada proceso (registros base y límite)

**Modos de ejecución**

· El bit en la CPU indica el modo actual

· Las instrucciones privilegiadas deben ejecutarse en modo Supervisor o Kernel

– Necesitan acceder a estructuras del kernel, o ejecutar código que no es del proceso

· En modo Usuario, el proceso puede acceder sólo a su espacio de direcciones, es decir a las direcciones “propias”.

· El kernel del SO se ejecuta en modo supervisor

· El resto del SO y los programas de usuario se ejecutan en modo usuario (subconjunto de instrucciones permitidas)

Tener en cuenta que…

· Cuando se arranque el sistema, arranca con el bit en modo supervisor.

· Cada vez que comienza a ejecutarse un proceso de usuario, este bit se DEBE PONER en modo usuario.

– Mediante una Instrucción especial.

· Cuando hay un trap o una interrupción, el bit de modo se pone en modo Kernel. – Única forma de pasar a Modo Kernel – No es el proceso de usuario quien hace el cambio explicitamente.

· Cuando el proceso de usuario intenta por sí mismo ejecutar instrucciones que pueden causar problemas (las llamadas instrucciones privilegiadas), el HW lo detecta como una operación ilegal y produce un trap al SO

**Protección de la memoria**

· Delimitar el espacio de direcciones del proceso

· Poner límites a las direcciones que puede utilizar un proceso

–Por ejemplo: Uso de un registro base y un registro límite

–El kernel carga estos registros por medio de instrucciones privilegiadas. Esta acción sólo puede realizarse en modo Kernel

· La memoria principal aloja al SO y a los procesos de usuario

o El kernel debe proteger para que los procesos de usuario no puedan acceder donde no les corresponde

o El kernel debe proteger el espacio de direcciones de un proceso del acceso de otros procesos.

**Protección de la E/S**

· Las instrucciones de E/S se definen como privilegiadas.

· Deben ejecutarse en Modo Kernel

– Se deberían gestionar en el kernel del sistema operativo

–Los procesos de usuario realizan E/S a través de llamadas al SO (es un servicio del SO)

**Protección de la CPU**

· Uso de interrupción por clock para evitar que un proceso se apropie de la CPU

· Se implementa normalmente a través de un clock y un contador.

· El kernel le da valor al contador que se decrementa con cada tick de reloj y al llegar a cero puede expulsar al proceso para ejecutar otro

· Las instrucciones que modifican el funcionamiento del reloj son privilegiadas.

· Se le asigna al contador el valor que se quiere que se ejecute un proceso.

· Se la usa también para el cálculo de la hora actual, basándose en cantidad de interrupciones ocurridas cada tanto tiempo y desde una fecha y hora determinada.

**System Calls**

· Es la forma en que los programas de usuario acceden a los servicios del SO.

· Los parámetros asociados a las llamadas pueden pasarse de varias maneras: por registros, bloques o tablas en memoria ó la pila. count=read(file, buffer, nbytes);

· Se ejecutan en modo kernel o supervisor

Categorías de system calls:

· Control de Procesos

· Manejo de archivos

· Manejo de dispositivos

· Mantenimiento de información del sistema

· Comunicaciones

System Call en Linux

· Para activar iniciar la system call se indica:

– el número de syscall que se quiere ejecutar

– los parámetros de esa syscall

· Luego se emite una interrupción para pasar a modo Kernel y gestionar la systemcall

· El manejador de interrupciones evalúa la system call deseada y la ej

Elementos Básicos de una computadora

· Procesador

· Memoria Principal

o Volátil

o Se refiere como memoria real o primaria

· Componentes de E/S

o Dispositivos de memoria secundaria

o Equipamiento de comunicación

o Monitor / teclado / mouse

· Bus Sistema

o comunicación entre procesadores, memoria,

o dispositivos de E/S

Registros del Procesador

· Visibles por el usuario

o Registros que pueden ser usados por las aplicaciones

· De Control y estado

o Para control operativo del procesador

o Usados por rutinas privilegiadas del SO para controlar la ejecución de procesos

Registros Visibles por el usuario

· Pueden ser referenciados por lenguaje de máquina

· Disponible para programas/aplicaciones

· Tipos de registros

o Datos

o Direcciones

§ Index

§ Segment pointer

§ Stack pointer

Registros de Control y Estado

· Program Counter (PC)

o Contiene la dirección de la proxima instrucción a ser ejecutada

· Instruction Register (IR)

o Contiene la instrucción a ser ejecutada

· Program Status Word (PSW)

o Contiene códigos de resultado de operaciones

o habilita/deshabilita Interrupciones

o Indica el modo de ejecución (Supervisor/usuario)

Instrucción: Fetch (leer) y Execute (ejecutar)

· El procesador busca (fetch) la instrucción en la memoria – (PC) → IR

· El PC se incrementa después de cada fetch para apuntar a la próxima instrucción –PC = PC + 4

IR - Instruction Register

· La instrucción referenciada por el PC se almacena en el IR y se ejecuta

· Categorías de instrucciones

o Procesador - Memoria

§ Transfiere datos entre procesador y memoria

o Procesador - E/S

§ Transfiere datos a/o desde periféricos

o Procesamiento de Datos

§ Operaciones aritméticas o lógicas sobre datos

o Control

§ Alterar secuencia de ejecución

Interrupciones

· Interrumpen el secuenciamiento del procesador durante la ejecución de un proceso

· Dispositivos de E/S más lentos que el procesador

o Procesador debe esperar al dispositivo

Interrupt Handler

· Programa (o rutina) que determina la naturaleza de una interrupción y realiza lo necesario para atenderla

o Por ejemplo, para un dispositivo particular de E/S

· Generalemente es parte del SO

Ciclo de interrupción

· El procesador chequea la existencia de interrupciones.

· Si no existen interrupciones, la próxima instrucción del programa es ejecutada

· Si hay pendiente alguna interrupción, se suspende la ejecución del programa actual y se ejecuta la rutina de manejo de interrupciones

Multiples Interrupciones

· Deshabilitar las interrupciones mientras una interrupción está siendo procesada.

· Definir prioridades a las interrupciones

Evolución de un S.O

· Los SO evolucionan con el objeto de:

• –Soportar nuevos tipos de HW

• –Brindar nuevos Servicios

• –Ofrecer mejoras y alternativas a problemas existentes

§ en la planificación

§ en el manejo de la memoria

§ etc

S.O. - Evolución Histórica

· Procesamiento en Serie

• No existía un SO

• Máquinas eran utilizadas desde una consola que contenía luces, interruptores, dispositivos de entrada e impresoras.

• Problemas:

§ Planificación. Alto nivel de especialización. Costos

§ Configuración: Carga del compilador, fuente, salvar el programa compilado, carga y linkeo

* Sistemas por Lotes Sencillos (batch)
  + Monitor Residente
    - Software que controla la secuencia de eventos
    - Los trabajos se colocan juntos
    - Los programas vuelven al monitor cuando finaliza la ejecución
    - No hay interacción con el usuario mientras se ejecutan los trabajos