

## • DEFINICIÓN DEL SO COMO GESTOR DE RECURSOS.

Es un programa que controla todos los recursos del ordenador:

- Traductor lenguaje maquina–humano y al revés.
- Traslado, almacén y proceso de datos.
- Ejecución de programas con sus correspondientes instrucciones que almacena en memoria.
- Traslado, almacenamiento y proceso de datos.
- Control de todas las funciones descritas arriba.

Es ejecutado por el procesador, por tanto es un programa que da instrucciones al procesador y a la vez lo dirige para usar otros recursos y para llevar un control del tiempo de ejecución de la aplicaciones. Así pues conoce que recursos se están empleando y como distribuirlos.

## • DEFINICIÓN DEL SO COMO INTERFAZ HOMBRE–MAQUINA.

El SO oculta al hombre los detalles de hardware y actúa como mediador entre maquina–programador para el uso de programas que faciliten la tarea de programar un determinado hardware o realizar aplicaciones. Para esto ofrece:

- Editores y Depuradores que ayudan al programador en la Creación de programas. Por tanto también ofrece el SO la Ejecución de los mismos.
- Acceso a dispositivos de E/S así como a los archivos y al sistema.
- Detección de errores y respuesta a errores tanto en dispositivos hardware como en aplicaciones software, ante los cuales debe solucionar el problema terminando la aplicación, reintentando, etc.
- Recoge estadísticas del uso de cada recurso así como del rendimiento del sistema.

## • DESCRIBIR LAS FUNCIONES Y OBJETIVOS DEL SO.

Los objetivos principales de un SO son:

- Eficacia: control de todos los recursos de la computación para ofrecer la base sobre la cual puedan ejecutarse los programas de aplicaciones.
- Comodidad: facilitar el uso del ordenador, con una interfaz entendible para el usuario.
- Capacidad de Evolución: el SO debe construirse de manera que permita posteriormente alguna modificación de nuevas funciones, como las actualizaciones, correcciones a los fallos, versiones mas mejoradas, etc.

Las principales funciones de un SO son:

- ◆ Gestión de memoria.
- ◆ Gestión de errores.
- ◆ Gestión de interrupciones.
- ◆ Gestión de E/S.
- ◆ Gestión de las tareas.
- ◆ Gestión de procesos, es decir, de movimientos de la CPU.

## • DIFERENCIAR LOS DISTINTOS TIPOS DE EXPLOTACIÓN DEL SO.

Según el numero de usuarios tenemos:

- **Sistemas Monousuario:** un solo usuario puede trabajar con el PC, cuando éste termine, queda libre para otro usuario.
- **Sistemas Multiusuario:** varios usuarios pueden trabajar con el ordenador de forma simultanea.

Según el numero de procesos a ejecutar de manera simultanea:

- **Sistemas Monotarea:** solo puede ejecutarse un proceso, hasta que no finalice no se puede ejecutar otro.
- **Sistemas Multitarea:** varios procesos pueden ejecutarse simultáneamente.

Según el numero de procesadores que disponga el sistema:

- **Sistemas Monoproceso:** un solo procesador.
- **Sistemas Multiproceso:** varios procesadores, bien se ejecuta un trabajo en varios procesadores o bien varios trabajos en diferentes procesadores.

Según el tiempo que se tarde en obtener un resultado de un proceso ejecutado:

- **Sistemas a Tiempo Real:** el tiempo de respuesta es inmediato.
- **Sistemas a Tiempo Compartido:** cada proceso consume una porción de tiempo de CPU, cuando se agota la porción de CPU se ejecutará otro proceso.
- **Sistemas de Procesos por Lotes:** procesos cuyos resultados no se necesitan de manera inmediata.

• **DEFINIR PROCESO Y ENUMERAR LOS DISTINTOS ESTADOS EN QUE PUEDE ENCONTRARSE Y LAS CAUSAS QUE PROVOCAN LA TRANSICIÓN DE UN ESTADO A OTRO.**

**Proceso:** actividad resultante de la ejecución de un programa secuencial (con sus datos y la pila del programa, su contador de programa, apuntador de pila y otros registros) por un procesador.

**Estados de un Proceso:**

- ◆ **Ejecución:** el proceso esta usando el procesador.
- ◆ **Listo:** el proceso está preparado para ejecutarse
- ◆ **Bloqueado:** el proceso no se puede ejecutar o no se puede continuar ejecutando porque ha de esperar datos de entrada que todavía no están disponibles.
- ◆ **Nuevo:** proceso que se acaba de crear, pero no ha sido admitido aun por el sistema en el grupo de procesos ejecutables.
- ◆ **Terminado:** proceso que ha sido excluido por el SO del grupo de procesos ejecutable bien porque se detuvo o porque fue abandonado.
- ◆ **Suspendido:** proceso que se ha sacado temporalmente de la memoria principal y se ha llevado a disco.

**Causas para la Transición de un Estado a otro:** una interrupción es la indicación de existencia de otra actividad pendiente. En caso de que se produzca, el procesador atiende la interrupción y decide en función de unos criterios preestablecidos cambiar de proceso.

Para poder hacer esta operación guarda información de lo que estaba haciendo y libera memoria o dedica otra parte a la nueva actividad. Esto es a lo que se denomina cambio de contexto. Al hacer un cambio de contexto, el procesador deberá de anotar todos los datos importantes que había antes de la interrupción para poder reanudar mas tarde la actividad. El SO creará un bloque de control para identificar cada proceso. Se establecen tantos bloques de control como procesos existan. Durante la existencia de un proceso puede pasar

por diversas situaciones algunas de las cuales suponen una actividad y otras representan un estado inactivo.

- **COMPARAR LAS DISTINTAS FORMAS DE PLANIFICACIÓN DE PROCESOS QUE PUEDE LLEVAR A CABO EL SO. CALCULAR MEDIANTE UN GRAFICO EL TIEMPO DE EJECUCIÓN TOTAL DE DISTINTOS PROCESOS SEGÚN DISTINTAS FORMAS DE PLANIFICACIÓN DE PROCESOS.**

**Planificación de Procesos:**

- Planificación por Torneo, tiempo paralelo o turno rotatorio: Se crea una lista con los procesos a ejecutar, a cada uno se le asigna un intervalo de tiempo fijo llamado quantum. Cuando un proceso termina el planificador pasa el control al siguiente proceso y así sucesivamente. El contador de la CPU que actúa como cronometro ve que ha transcurrido un tiempo de ejecución para el proceso igual al quantum, provoca una interrupción de la CPU y da el turno al siguiente proceso. Si antes de que termine el quantum el proceso ha terminado o se ha bloqueado, el procesador se conmuta en el momento del bloqueo. Cuando se acaba el quantum del proceso, éste pasa al final de la lista.
- Planificación por Prioridad: a cada proceso se le asigna una prioridad y así se procesa en el procesador el proceso de más alta prioridad (perdón por la redundancia!!!). El planificador decrementa la prioridad a medida que se esta ejecutando el proceso para evitar que se ejecute por tiempo indefinido y cuando su prioridad es menor que la que tiene otro proceso, se realiza un cambio de proceso a ejecutarse.
- Lista de espera con Intervalos Múltiples: Para procesos que necesitan usar mucho procesador. Se le asigna a cada proceso un intervalo de tiempo de ejecución diferente según el proceso. El proceso empieza a ejecutarse con su tiempo, cuando lo ha agotado se reestablece el tiempo para ser usado la siguiente vez.
- Prioridad al mas Corto: Para sistemas por lotes en los que se conoce de antemano el tiempo de ejecución de un proceso.
- Tiempo Compartido: Cuando hay varios terminales , cada usuario en su terminal ejecuta sus procesos como si fuera el único proceso que se está ejecutando.

- **IDENTIFICAR LOS REQUISITOS DEL SISTEMA PARA GESTIONAR LA MEMORIA.**

Son 5 requisitos:

- Reubicación: El programador no sabe qué demás programas residen en memoria en el momento en que se ejecuta uno. El SO carga y descarga los procesos activos en la memoria principal para aumentar el uso del procesador y así reservando procesos listos para ser ejecutados. Esto se hace por técnicas de intercambiabilidad swapping. Para esto debe saber donde ubicar un programa en la memoria. El procesador y el sistema deben traducir las referencias a memoria para saber donde esta ubicado un programa o todas las partes que componen un programa.
- Protección: cada proceso ha de protegerse de interferencias no deseadas de otros procesos. La protección la realiza el procesador, asignando al proceso un determinado registro de base y registro de limite que las carga en la dirección de memoria donde esta el proceso y así se le suma al valor del proceso el valor del registro base. Los valores de las direcciones se comparan también con las del registro limite y así se asegura que no existe direccionamientos de memoria.
- Compartición: Acceso a varios procesos a una zona de memoria compartida.
- Organización Física: Mueve los procesos desde la memoria interna a la secundaria según la disponibilidad de espacio en una u otra.
- Organización Lógica: Los programas se organizan en módulos de memoria lo cual hace que mientras se ejecutan los módulos el sistema pueda acceder a la información de un modulo a otro.

- **ESTUDIAR LAS DIFERENTES TÉCNICAS DE GESTION DE MEMORIA: PARTICIONES**

## **ESTATICAS, PARTICIONES DINAMICAS, SEGMENTACIÓN, PAGINACIÓN Y MEMORIA VIRTUAL. IDENTIFICAR LAS DIFERENCIAS ENTRE ELLOS.**

**Particiones Fijas:** La forma más sencilla de dividir la memoria es en  $n$  partes (que podrían ser de distintos tamaños). Esta partición podría hacerla el operador en forma manual, al iniciar una sesión con la máquina(ahora esto lo hace en forma automática el sistema operativo). Cuando llega un trabajo, se le puede colocar en la cola de entrada de la parte de tamaño más pequeño de forma que lo pueda contener. Puesto que las particiones están fijas en este esquema, cualquier espacio que no sea utilizado por una tarea se pierde.

**Particiones Dinámicas:** el número y tamaño de los procesos en la memoria varía en forma dinámica a medida del uso. El hecho de no estar sujeto a un número fijo de particiones que pudieran ser muy grandes o demasiado pequeñas mejora el uso de la memoria pero también hace más compleja la asignación y re-asignación de memoria, así como también complica el mantener un registro de esto.

**Paginación:** técnica de gestión que permite asignar la memoria de forma discontinua. Con este fin se divide la memoria fija en trozos llamados frames, y la lógica en bloques de igual tamaño llamados páginas. El SO internamente mantiene una tabla de páginas donde relaciona cada página cargada en memoria principal con el frame que la contenga.

Utilizando el número de página el sistema recorrerá toda la tabla de páginas hasta localizarla, sumará el desplazamiento a la dirección de carga y obtendrá la dirección real.

Cada programa se subdivide en páginas, que se cargan en frames libres que no tienen porque ser seguidos.

El sistema analizará cada nuevo trabajo para conocer el número de página que ocupa y buscará en la lista de frames libres un número igual de frames; si encuentra suficientes cargará en ellas las páginas del programa y construirá la tabla de páginas.

**Segmentación:** un programa se divide segmentos o componentes lógicos(cada uno definido por su tamaño y numero; toda dirección lógica se expresará mediante dos valores, el numero de segmento y desplazamiento dentro del segmento). La segmentación del programa la realiza el compilador. Puesto que la memoria física es direccional linealmente con un numero será necesario transformar cada dirección lógica en una dirección real unidimensional. Esta conversión la realiza un dispositivo especial de hw consultando la tabla de segmentos.

**Memoria Virtual:** La idea de la memoria virtual se basa en que el tamaño combinado del programa, los datos y la pila pueden exceder la cantidad de memoria física disponible para él. El sistema operativo mantiene aquellas partes del programa que se utilicen en cada momento en memoria principal y el resto permanece en el disco duro. Usa para ellos la segmentación y la paginación. El procesador traduce las direcciones virtuales a físicas y genera las interrupciones cuando la pagina o segmento referenciado no está en memoria principal.

### **• EXPLICAR COMO LLEVA A CABO EL SISTEMA LA GESTION DE LA INTERRUPCIONES.**

Mecanismo hardware para sincronizar la CPU con los sucesos externos que son asíncronos; la idea consiste en que el interfaz fuerce a la CPU a dejar su actividad para atender a una rutina de E/S y luego haga que la CPU retorne al proceso en curso en el punto donde lo había dejado; para implementar dicha actividad, es necesario un registro que guarde el contador del programa en el instante en que se produce la interrupción para que la CPU pueda reanudar el proceso en el punto en el que fue detenido.

Cuando existen varios dispositivos se puede realizar un escrutinio después de la petición de interrupción. En el caso de existir varios dispositivos preparados habría que tener en cuenta su prioridad para ser atendidos.

### **• ENUMERAR Y DESCRIBIR LAS DISTINTAS TÉCNICAS DE GESTION DE LAS E/S.**

- **E/S Programada:** Cuando el micro está ejecutando un programa y se encuentra con una instrucción de E/S envía la orden al dispositivo para que se realice la acción, el dispositivo no interrumpe al programa que se está ejecutando, por lo que es el micro quien tiene que comprobar periódicamente el estado del modulo de E/ S hasta que éste haya terminado, lo cual provoca tiempos de espera en el micro.
- **E/S Dirigida por Interrupciones:** Cuando el micro envía una orden de E/S al dispositivo, éste solo lo interrumpirá cuando acabe de realizar la acción para avisarle de la situación, sigue consumiendo tiempo de micro pero en menor medida que el anterior.
- **DMA:** Se usa cuando hay que trabajar con grandes cantidades de información. Muchos controladores de dispositivos tienen acceso directo a la memoria, la función DMA se lleva a cabo mediante un modulo del Bus del Sistema. Cuando el micro ha de enviar una acción emite una orden para el modulo DMA y le envía la información. Así el micro continua con cualquier trabajo mientras el modulo DMA se ocupa de la E/S.

## • ESTUDIO DE LA GESTION DE ALMACENAMIENTO SECUNDARIO: SISTEMAS DE FICHEROS.

Los directorios son tablas simbólicas de los archivos del sistema. En principio cada entrada en un directorio define un archivo. Cada archivo se define con el nombre, sus atributos y el puntero de acceso o comienzo de los bloques de datos. Se pueden jerarquizar en diferentes niveles y así podemos hablar de:

- **Directorios de 1 nivel:** Es la forma mas sencilla y en ella todos los ficheros están en un solo directorio por lo que resulta muy fácil de gestionar pero no está ordenado.
- **Directorios de 2 niveles:** El directorio principal contiene la lista de los directorios subordinados y en cada uno de estos sus ficheros correspondientes.
- **Directorios arborescentes:** Cada usuario posee un subdirectorio o más del principal y a su vez organiza los archivos creando los subdirectorios que considere.

## • IDENTIFICAR LOS DISTINTOS TIPOS DE AMENAZAS A LA SEGURIDAD EN LOS SO Y DESCRIBIR COMO IMPEDIRLAS.

- **Interrupción:** Se destruye un elemento del sistema o se hace inutilizable.
- **Intercepción:** Una parte no autorizada(persona, programa, ordenador...) consigue acceder a un elemento del sistema.
- **Modificación:** Una parte no autorizada además de acceder al sistema, falsifica un elemento.
- **Invencción:** Una parte no autorizada inserta objetos falsos en el sistema.

Para impedir éstas amenazas hay que realizar copias de seguridad del sistema de datos(ficheros, directorios...), copia de reserva o imagen de las versiones software, los programas y usuarios del sistema deben de operar con el mínimo de derechos de acceso o privilegios, poseer para líneas de comunicación un firewall para evitar posibles ataques en la red y un buen antivirus.

## • ESTABLECER METODOS PARA LA SEGURIDAD TANTO DE FICHEROS DEL SISTEMA COMO DE DATOS.

Lo más fundamental es la protección de la memoria principal que en sistemas con procesos segmentados o paginados, el SO debe asegurar que no haya entradas duplicadas en las tablas de paginas o segmentos. Si se va a permitir la compartición, el mismo segmento o pagina puede ser referenciado en mas de una tabla.

Un SO puede ser usado por diversos usuarios, a los cuales se les deben de restringir o conceder determinados

permisos sobre los datos y ficheros, cuantos menos privilegios mas protección de nuestros datos, aunque es necesario a veces dar ciertos permisos.

También es recomendable tener instalado un buen antivirus por si nos infectáramos y poder tener a salvo toda nuestra información.

- **DIFERENCIAR LOS DISTINTOS NIVELES DE LENGUAJES DE PROGRAMACIÓN, ASI COMO LAS DISTINTAS FORMAS DE TRADUCCIÓN: INTERPRETACIÓN Y COMPILACIÓN.**

**Lenguajes de Alto Nivel:** Aquel lenguaje en el que el programa fuente se aproxima mas al lenguaje humano que al código maquina.

Son lenguajes portables, usan ordenes englobantes y necesitan ser traducidos al lenguaje maquina antes de ejecutarlos.

Ejemplos: Fortran, Cobol, Basic, C, C++, Unisys, Linux y Java.

**Lenguajes de Bajo Nivel:** Aquel lenguaje en el que las instrucciones están codificadas en un lenguaje mas parecido al maquina que al humano.

Los programas en lenguajes de bajo nivel son muy rápidos. No poseen portabilidad por lo que es muy raro que puedan ser ejecutados en una maquina distinta. Las operaciones y ordenes deben descomponerse en todos sus pasos y en código binario.

Ejemplos: ENAC, Ensamblador, éste último reagrupa algunas ordenes por código alfanumérico.

**Interpretación:** el interprete coge sentencia por sentencia del programa fuente y la analiza y traduce a código maquina y la ejecuta antes de proceder con la siguiente.

Una orden en un bucle que se ejecuta 100 veces la debe traducir 100 veces.

Un error sintáctico en un bucle puede provocar que el programa se ejecute bien muchas veces y en una ejecución determinada nos dé un error.

Necesitan ejecutarse a través del interprete por lo que si lo trasladamos a otra maquina tenemos que tener un interprete allí.

Las ordenes y funciones son limitadas.

**Compilación:** Consiste en traducir todo el programa fuente en una operación, generando como resultado un fichero que contiene el programa ejecutable, el cual se ejecutará en cualquier maquina que tenga el mismo procesador.

La compilación se hace en dos fases:

- **Compilación del fuente:** revisión léxica, semántica y sintáctica de todas las ordenes y luego las convierte en texto ASCII y posteriormente a código maquina generando el programa objeto que no es ejecutable.
- **Encuadración:** añaden al programa objeto los módulos objeto como librerías, rutinas... y así de un conjunto de programas objeto se obtiene el programa ejecutable.