

SISTEMAS OPERATIVOS

Un sistema operativo es un conjunto de programas que controla los recursos del ordenador y sirve para la ejecución de los programas e general convirtiéndose en el intermediario entre el hardware y el usuario.

Tiene dos objetivos:

- **Seguridad**: el sistema operativo debe actuar contra cualquier manipulación extraña ya sea accidentada o premeditada que pudiera dañar la información, perjudicar a otros usuarios o funcionar de forma inadecuada el sistema.

Para ello tiene dos estados:

–**Protegido**: se ejecuta el sistema operativo:

–**No protegido**: destinado a la ejecución de los programas de usuario de aplicación. Así se impide que los programas de usuario puedan tener contacto con el hardware del sistema.

- **Abstracción**: permite ocultar lo más posible los detalles de más bajo nivel intentando dar a los niveles superiores una visión más sencilla, global y abstracta ofreciendo operaciones para manipular dichas estructuras ocultas desconocidas por completo para la gestión interna de la estructura.

Gracias a ella, los sistemas operativos enmascaran los recursos físicos permitiendo su manejo con funciones mas generales que ocultan las básicas constituyendo verdaderos recursos ficticios o virtuales que mejoran o son más potentes que los físicos.

FUNCIONES ESPECÍFICAS DE LOS STMAS. OP.

- Comunicación con el usuario.
- Control y gestión de elementos físicos.
- Utilización de varios usuarios.
- Ejecución de varios programas.
- Compilación de la información.
- Compilación de los periféricos.
- Comunicación entre sistemas.

COMPONENTES DE UN STMA. OP

Están formados por el conjunto de programas que utiliza el ordenador para producir un trabajo útil. Hay 4 tipos:

- **Aplicación**: su función es resolver los problemas particulares de los usuarios. A veces forman parte de paquetes que suministran ciertas empresas de software y que pueden ser particularizadas. Hay 2 tipos:

–**de gestión**: se destinan a los diversos aspectos de gestión de las empresas. Manejan gran cantidad de datos y utilizan cálculos muy sencillos por lo que necesitan gran capacidad de almacenamiento y poca memoria.

–**de cálculo**: se destinan a funciones científicas y técnicas. Tienen gran rapidez de CPU y poca gestión de entrada y salida.

- **Utilidad:** destinados a realizar funciones específicas. Destinado a un hardware específico y elaborado para la empresa constructora del ordenador, realizan funciones de tipo repetitivo. Hay dos tipos:

–organizadores del stma: se ejecutan bajo la supervisión del stma os. Y permiten manejar las librerías de programas y proteger la INF.

–organizadores de archivos: se encargan de manipular los datos clasificando o reorganizando los registros de los diferentes archivos.

- **Servicio:** destinados a gestionar y poner a punto los programas de aplicación. dentro están:

–P. Traductores: realizan el paso del lenguaje de usuario al lenguaje máquina. El programa original e el programa fuente y el programa que queda tras la traducción el programa objeto. Según la traducción se pueden clasificar en:

◇ **Compiladores**: traducen todo el programa fuente a máquina dejándolo dispuesto para su ejecución.

◇ **Intérprete**: traduce el programa fuente paso a paso ejecutando cada instrucción traducida.

◇ **Reprocesador**: modifica ligeramente el programa fuente traduciéndolo a otro lenguaje intermedio que aún comprende el usuario permitiéndolo mayor potencia antes de la traducción completa.

◇ **Ensamblador**: compilador sencillo que permite la traducción de una instrucción del lenguaje fuente a otra del lenguaje máquina.

–P. Gestión de librerías: destinado a conservar y organizar el conjunto de los programas de aplicación. Entre sus funciones están anulación y borrado de programas, incorporación de programas y gestión de programas. Como los programas pueden estar en lenguaje e alto nivel en módulos o preparados par la ejecución las librerías se pueden clasificar en:

- **L. Fuente**: contiene los programas en lenguaje de alto nivel.

- **L. Objeto**: contiene los módulos traducidos de los diversos programas.

- **L. Ejecutable**: contiene programas en código máquina, una vez enlazados los módulos que lo componen.

- **Control**: conjunto de programas que facilitan y mejoran el rendimiento del uso de los ordenadores. Son los encargados de llevar a cabo la gestión de los recursos del stma facilitando el acceso a los recursos disponibles e impidiendo el interbloqueo de actividades. Suelen estar formados por módulos cada uno de los cuales gestiona uno de los grupos fundamentales: el hardware, el software y los trabajos.

- **Gestionador hw**: se encarga de controlar y gestionar la CPU, la memoria y las rutinas de entrada y salida.

- **Gestionador de archivos**: maneja los datos externos moviéndolos entre la memoria y los periféricos.

- **Gestionador de trabajos**: se encarga de asignar recursos a las diversas tareas y permite la comunicación con el exterior.

ARQUITECTURA Y ESTRUCTURA DE LOS S.O

Existen varios procedimientos para la implementación de los s.o:

- **Stmas. Monolíticos:** consisten en escribir el s.o como un conjunto de procedimientos cada uno de los cuales tiene una interfaz que le permite llamar a otro y otros procedimientos. Presenta una estructura mediante una instrucción que permite tener a la máquina estar trabajando en dos modos:
 - **Modo usuario:** no se pueden manejar todas las instrucciones del s.o lo que permite la protección del stma. Cuando es necesario realizar una función que afecte al stma hay que hacer una llamada al ordenador de modo que actúe en modo supervisor y después devolver el control al usuario para que continúe con la ejecución del programa.
 - **Modo supervisor:**
- **Stmas. Jerárquicos:** consisten en organizar el s.o como una serie de niveles cada uno de los cuales interactúa con su centro a través de interfaces bien definidas. Cada problema complejo se divide en módulos simples, pequeños y fáciles de comprender, diseñar y cambiar en caso necesario. En cada nivel de la jerarquía se ignoran los detalles de actuación de los niveles inferiores provocando tareas compartidas con independencia de los demás.

Los procesos se agrupan en niveles. Cada procedimiento puede invocar a los del nivel inferior. Cada parte del s.o se encarga de una función. Cuando una función necesita de otra pedirá ayuda al s.o, evitando así la interdependencia entre funciones.

- **Stmas. Máquinas virtuales:** consiste en crear varias máquinas independientes, cada una de las cuales es utilizada por uno o varios usuarios, aunque solo existe una máquina real. Para ello se comparten los recursos de forma que cada usuario cree estar utilizando su propio ordenador.

El centro del stma es el monitor de la máquina virtual u se implementa en el hw realizando multiprogramación con la creación de un estrato superior formado por máquinas virtuales que son copia exacta de la máquina real.

Estos s.o simulan hw y permiten que con la misma unidad de proceso estén trabajando varios usuarios con s.o diferentes.

- **Stmas. Clientes/servidor:** desplazan una serie de rutinas del núcleo a niveles superiores. Para ello se implementan funciones del stma en proceso de usuario. Para solicitar un servicio el proceso de usuario envía una solicitud al proceso servidor que realiza el trabajo y da respuesta al usuario.

Se caracterizan por constar de un núcleo reducido, procesos capaces de interactuar como clientes y servidores.

- **Stmas. Orientados a objetos:** ven el stma como una colección de objetos.

ESTRUCTURA DE LOS STMAS OPERATIVOS

Se organizan en capas entorno al núcleo principal. Cada una de estas tiene un nivel de prioridad siendo el núcleo la capa más prioritaria. Sus partes son:

- **NUCLEO:** es la capa mas interna, en contacto con el hw.
- **NIVEL EJECUTIVO:** se encarga de la gestión del stma de archivos y de los procesos.
- **NIVEL SUPERVISOR:** realiza la interpretación de los mandatos del stma.
- **NIVEL INTERFAZ DE USUARIO:** se encarga de la presentación y gestión de las órdenes del

usuario al stma. Estas ordenes pueden estar en:

- **Modo texto:** se forma con un símbolo o letra. En caso de utilizar la letra, suele ser la unidad de disco que está utilizando.
- **Modo gráfico:** es un stma de ventanas y menús despegables, que incluyen iconos, que proporcionan un acceso inmediato al programa que se quiere ejecutar. Hay dos tipos:
- S.o independiente.
- S.o no independiente.

MODOS DE EXPLOTACIÓN DE UN S.O

- **Offline:** el mayor problema al ejecutar un programa es la diferencia de velocidad entre los elementos de entrada y salida y la CPU.

Los primeros ordenadores utilizaban tarjetas perforadas para la entrada de datos. Eran muy lentos. Uno de los primeros intentos para solventar el problema fue la sustitución de las tarjetas lectoras por cintas magnéticas; se pretende acelerar el proceso utilizándolas como instrumento intermedio entre los periféricos y la CPU. Los procesos de la lectora eran cargados en la cinta en un procedimiento fuera del principal y las salidas de impresora eran volcadas sobre cintas que posteriormente serían impresas. Este proceso permitía agrupar varios procesos en una cinta acelerando el stma de acceso.

Este stma tiene una doble orientación en función de su funcionamiento. Tiene un primer enfoque dado por la utilización de un hw diseñado para hacer cargas y descargas a cintas en el ordenador independiente que prepara los procesos de la cinta al ordenador principal.

- **Buffering:** realiza el papel de intermediario entre dispositivos. Consiste en establecer un dispositivo de almacenamiento entre la CPU y los dispositivos externo produciéndose una serie de interrupciones al controlador cada vez que al buffler se le acaben los datos.

Los bufflers pueden contener los registros que han sido leídos, pero todavía no han sido procesados o viceversa. De esta forma un dispositivo de entrada puede leer por anticipado varios registros antes de que la CPU los procese. Si el buffer se llena, el dispositivo de entrada deberá esperar hasta que la CPU procese los registros

- **Spooling:** es una técnica utilizada solo en los stmas de disco. Trata el disco como un gran buffler para leer tanto como sea posible a partir de los dispositivos de entrada y almacenar los ficheros de salida hasta que el dispositivo de salida los acepte. Puede solapar la entrada y salida de varios trabajos diferentes.

Proporciona una estructura de datos que es el **pool** de trabajos, que es un conjunto de trabajos que permite al stma operativo seleccionar que trabajo ejecutará en siguiente lugar para aumentar la utilización de la CPU.

TIPOS DE STMAS OPERATIVOS

Hay una gran variedad, en función de sus características, y cada uno se caracteriza por las técnicas de acceso a la información y el modo de su proceso. Técnicas:

- **Monoprogramación:** la ejecución de un programa se realiza de forma lineal. Cuando un programa empieza a procesarse se le asigna todos los recursos hasta su finalización sin la interferencia de otros.

La ventaja es la asignación completa de los recursos a un solo usuario con la consiguiente velocidad de este.

El inconveniente es que obtenemos un retraso en la ejecución por el resto de los usuarios que se encuentran en lista de espera.

Con esta técnica, el paso de entrada a ejecución se realiza mediante un stma de asignación de prioridades.

- **Multiprocesamiento**: se da en los stmas que poseen CPU y en cada una de ellas se ejecuta un proceso. Puede ser:
- **Acoplado**: hay una única memoria y varias CPU o dispositivos de e/s dependiendo de ella.
- **Memoria dividida en varias**: a cada una de estas memorias se le asigna una CPU y un dispositivo de e/s, actuando independientemente.
- **Multiprogramación**: consiste en el aprovechamiento de los tiempos muertos liberando al procesador de la espera y destinándolos mientras tanto a avanzar la ejecución de otros procesos.

En ella, el s.o toma uno de los procesos residentes en memoria y comienza a ejecutarlo. Si tiene que esperar, la CPU cambia a otro proceso y empieza a ejecutar el otro proceso hasta que finalice la espera del primer proceso; así, mientras exista algún proceso que ejecutar la CPU no estará parada.

Estos procesos tienen que compartir memoria por lo que exigen memoria de mayor capacidad y técnicas de protección de memoria.

Su problema es el interbloqueo por el acceso de diferentes programas a un mismo recurso o a una misma posición de memoria por lo que deberán utilizarse criterios de asignación de la CPU para evitar estos posibles fallos.

Esta técnica utiliza diferentes técnicas:

- **Ficheros batch**: mediante los cuales, se agrupan varios procesos en un lote y se ejecutan en tiempo en los que no esté el operador. En ellos no participa el usuario y el tiempo no es un factor crítico.
- **Tiempo compartido**: el usuario trabaja con el ordenador de forma interactiva. En estos stmas es muy importante el tiempo de respuesta, ya que se trata de obtener la compartición de un solo stma entre varios usuarios de forma que cada uno pueda relacionarse con los recursos del stma en condiciones óptimas de servicio que le den sensación de tener un sistema. Su funcionamiento es:

Se dispone de un reloj interno que actúa como temporizador. A cada programa se le otorga un quantum de tiempo máximo. Si dentro del tiempo otorgado a un programa se produce una interrupción se actúa según la multiprogramación, pero si el programa consume por completo su quantum de tiempo el s.o genera una interrupción por espiración del tiempo y el control de ejecución pasa a otro programa concurrente.

- **Sistemas en tiempo real**: es utilizado como dispositivo de control en aplicaciones especializadas. Su funcionamiento es:

Un conjunto de sensores proporcionan los datos al ordenador, que los almacena e intenta ajustar los controles para modificar las entradas del sensor.

Un stma de tiempo real está sometido a fuertes restricciones de tiempo fijas y bien definidas. El procesamiento tiene que realizarse dentro del intervalo temporal establecido, sino, el stma no muestra los datos correctos para el fin pretendido. Tienen dos posibilidades de funcionamiento:

- Que el procesador esté parado mientras no se le requiera para tener

rapidez cuando sea necesario.

- Que el procesador realice otras tareas pero cuando se precise la función en tiempo real tenga máxima prioridad.

SEGURIDAD DEL SISTEMA Y DE LA INFORMACION

En los sistemas se establecen tres niveles de protección en los que hay que establecer unas medidas de seguridad para evitar el enfrentamiento y acceso indebido a determinadas posiciones de memoria e información. Los niveles son:

1-.Protección de memoria Consiste en que un trabajo no puede acceder a otro. Una de las técnicas para evitar esto es la utilización de registros frontera y detección por hw.

Cada intento de emplear la memoria por hw comparará si la dirección de memoria es menor o igual al registro frontera. Si es así dará un mensaje de error. Este sistema operativo exige un hw especial y unas funciones del sistema especiales

2-.Protección del uso de la CPU: Para que todas las tareas cuenten con el mismo tiempo se utilizan temporizadores que aseguran una cantidad de tiempo para cada usuario activo.

3-.Protección de e/s: Se establece una protección a la hora de compartir recursos para operaciones de e/s.

La compartición de recursos en sistemas de multiprogramación la establece el S.O. Se hace una asignación en modo privilegiado mediante un bit adicional.

SERVICIOS DE UN SISTEMA OPERATIVO

1.Gestión de procesos: Un proceso es un programa que se está ejecutando y consta también de los datos como de los registros necesarios para llevar a cabo la ejecución. Es un programa que se está ejecutando con unos datos y unos registros.

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 S.O. 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. Procesos:

Activos Inactivo

En ejecución1 Suspendido listo4

Listo 2 Suspendido bloqueado5

Bloqueado3

1El proceso se encuentra activo utilizando la CPU.

2El proceso es capaz de estar activo si tiene la CPU disponible.

3En espera que suceda algún evento para recuperar su actividad.

4En el momento en que lo activemos puede pasar a ejecución.

5No está activo y tiene que esperar a que suceda algo para poder activarse.

Las diferentes operaciones que se pueden realizar sobre un proceso son:

–**Creación de un bloque de control**

–**Destruir un proceso:** Elimina el bloque de control

–**Suspender un proceso:** Pasa de estado bloqueado o ejecución a un estado de suspendido

–**Activar un proceso:** La inversa de la anterior. El hecho de pasarlo a la lista consiste en que en función de la prioridad asignada por el usuario el sistema operativo crea la lista de prioridades para ejecutar el proceso.

Una de las técnicas utilizadas a la hora de establecer prioridades es la de envejecimiento que consiste en el que el S.O. va aumentando la prioridad de aquellos procesos que llevan mas tiempo en la lista de espera.

TIPOS DE PROCESOS

Podemos establecer diferentes clasificaciones:

–**En función de la utilización de memoria:**

Procesos residentes: Durante su vida activa tiene que estar cargado en memoria.

Procesos intercambiables: Cuando se realiza una operación que genera la espera, permite meter su proceso tratado por la CPU en un fichero del sistema.

–**En función de las características de su código:**

Procesos reutilizables: Programas que pueden emplear muchos usuarios. Están escritos con variables genéricas y cada uno introducirá los valores de las variables genéricas que precise.

Procesos reentrantes: Programas que apuntarán a la misma zona de memoria mientras que el proceso tratado por la CPU tendrá asignados unos datos concretos correspondientes a los valores que la memoria da.

Procesos de excepciones: Se trata de acontecimientos que pueden ocurrir en un proceso, condicionados por un suceso que no se debería de haber dado y por ello el sistema tratará de evitarlo. Este tipo de sucesos es a los que denominamos errores. Se establecen diferentes categorías:

- ◆ **Catastróficos:** No permiten continuar al S.O.
- ◆ **Recuperables:** Hacen que el proceso no pueda continuarse lo que conlleva la cancelación del mismo.
- ◆ **No recuperables:** Permiten que el proceso continúe con ciertos ajustes. Al producirse un

error el hw lo detecta. Produce una interrupción del S.O. y lo manda a una posición fija, en esta posición habrá una rutina que se encarga de tratar ese error. El S.O. tiene rutinas específicas para cada tipo de error y están situadas en posiciones de memoria fijas.

PLANIFICADOR DE PROCESOS

Es un conjunto de mecanismos incorporados al sistema operativo que establecen el orden en que se van a realizar los diferentes procesos.

En un sistema multiproceso el S.O. almacenará en un fichero todos los trabajos de los usuarios. Todas estas tareas acumuladas no son procesos puesto que todavía no están iniciadas. Para poder convertirse en procesos deben someterse a la planificación del procesador.

El usuario deberá de ir viendo en cada momento que trabajos se van convirtiendo en procesos y en que momento.

La parte del sistema operativo dedicada a la planificación es el **planificador a largo plazo**, esta planificación la hará en función de prioridades.

El **planificador a corto plazo** se encarga de definir que proceso preparado en la cola pasa a ejecución. Algunos sistemas aprovechan el tiempo de espera para sacar a un disco externo los procesos bloqueados.

El **planificador a medio plazo** decidirá que procesos bloqueados deben sacarse al fichero del sistema para liberar la memoria. Mas tarde es el sistema operativo el que se encarga de volver a introducirlos en la memoria para que sigan ejecutándose. Con la planificación del procesador se pretende conseguir una justicia en el reparto de la CPU entre los diferentes usuarios, lograr la máxima capacidad de ejecución con el mayor número de usuarios interactivos y predecir los límites y capacidad del hw y del propio sistema operativo.

ALGORITMOS DE PLANIFICACIÓN

1-FIFO: Se atiende al primer proceso que llega y el resto entran a la cola de espera. El inconveniente es que el tiempo de respuesta es impredecible y se penalizan los procesos cortos.

2-ROUND-ROBIN: Se define un intervalo o quantum de tiempo. Mismo quantum para todos los procesos de la cola y se atiende a la lista de espera de forma circular.

La diferencia con el FIFO es que este acapara la CPU hasta que finaliza, mientras que con el round-robin cada proceso tiene su quantum.

Cuando se finaliza el tiempo se asocia la CPU a otro; si el proceso inacabado necesita mas CPU se le coloca al final de la lista. Su rendimiento depende del valor del quantum.

3-SHORT JOB FIRST: Se caracteriza por elegir de la cola de trabajos el mas corto. El proceso que tiene asignada la CPU no dejara el proceso hasta que finalice.

4-SHORT REMAINDER TIME: Es igual que el anterior, pero este si entra un proceso mas corto que está en ejecución quita la CPU al que se está ejecutando y se la da al que acaba de llegar.

5-PRIORIDAD MULTINIVEL: Los procesos pueden asignarse por su prioridad en diferentes colas. El sistema operativo aplicará 1determinado algoritmo a cada cola.

6-HIGH REMAINDER NEXT: La prioridad la marca el tiempo que está esperando en la cola y el tiempo para la ejecución.

En la ejecución de los procesos se pueden dar dos circunstancias:

1-.CONVIVENCIA DE PROCESOS–Serán aquellas situaciones en las que los procesos se ejecutan en paralelo sin que existan interferencias

2-.CONCURRENCIA DE PROCESOS–Existe un paralelismo dependiente.

Para evitarla se utiliza el **sistema de exclusión mutua** que es una sincronización por la cual una actividad impide a otra que pueda tener acceso a un dato mientras se encuentra realizando una operación sobre el mismo. Una actividad impide a la otra el acceso a un momento dado; la zona de código perteneciente a cada programa o proceso destinada a utilizar los recursos o datos compartidos es la sección crítica y debe de cumplir:

- ◆ Exclusión mutua Mientras esté en su sección crítica ningún otro podrá hacer uso de los recursos utilizados por el primero.
- ◆ Progreso Ningún proceso puede bloquear a otros cuando esté fuera de su sección crítica.
- ◆ Espera limitada Un proceso no espera indefinidamente para entrar en su sección crítica.
- ◆ No realizar suposiciones sobre la velocidad de los procesos ni el número de procesadores Podría dar lugar a que no se mantuviera la exclusión mutua en el caso de existir variación en las condiciones de procesamiento y entorno permitiendo que varios procesos utilicen a la vez los procesos compartidos.

Con la **exclusión mutua** se consigue asegurar que nunca los dos procesos

van a ejecutar su sección crítica a la vez. Se consigue mediante la sincronización de procesos que se pueden realizar mediante sw con los algoritmos de **espera activa** o **espera no activa** y mediante hw.

–**Espera activa:** El proceso paralelo va ejecutando sentencias que preguntan a la CPU si se ha concluido el trabajo para entrar.

–**Espera no activa:** El proceso queda bloqueado en la cola y cuando se encuentra libre la CPU el sistema avisa.

ALTERNANCIAS

Se basa en que los dos procesos compartan una variable que indique cual puede entrar en la sección crítica. A esta variable la denominaremos turno pudiendo alcanzar dos valores 0 ó 1, indicando si puede acceder a su sección crítica el P1 o el P2. Debido al uso del turno dos procesos no pueden estar en la sección crítica simultáneamente.

Su inconveniente es que esta estricta alternancia puede dar lugar a que no se cumpla el requisito de progreso, puesto que en el caso de el proceso se abortara dentro de la región crítica no se le pasaría el turno al P2 con lo que estaría esperando indefinidamente.

SEMAFOROS

Es una solución planteada para la espera no activa. Se basan en la utilización de una variable como

contador de peticiones de entrada a una sección crítica. Esta variable es compartida por todos los procesos del sistema y tiene un identificador único.

De esta forma cuando un proceso intente entrar en una sección crítica mientras otro está accediendo a los datos compartidos se bloqueará de manera similar a cuando un proceso acceda a un recurso que está ocupado.

El semáforo tiene un valor inicial que indica el número de procesos que pueden acceder simultáneamente a la sección crítica. Cada vez que entra un proceso el valor del semáforo disminuirá en una unidad y al salir se incrementará. Cuando el semáforo alcance el valor 0 no podrá entrar ningún proceso en la sección crítica protegida.

–PROPIEDADES Y PROBLEMAS DE UN SEMÁFORO–

Garantiza la exclusión mutua de cualquier n° de actividades en cualquier n° de procesadores solo si hacen referencia al mismo semáforo.

Cuando se bloquea una actividad al entrar en su sección crítica nunca realizará una espera activa sino que se insertará en la cola de procesos bloqueados.

Pueden darse problemas de espera indefinida de procesos salvo que la cola de procesos bloqueados se gestione con una política FIFO.

–Un semáforo tiene tres operaciones:

1-.INICIALIZACIÓN:

Semáforo = valor

2-.PROCESO P: entrar en sección crítica

Si (Semáforo > 0)

Semáforo = Semáforo - 1

Sección crítica

Sino esperar

3-.OPERACIÓN V: salir sección crítica

Si (Semáforo = 0)

Semáforo = Semáforo + 1

Siguiente proceso (desbloquear)

Sino (Semáforo = Semáforo + 1)

Este tipo de operaciones plantea como ventaja el ahorro de consumo de CPU, pero su inconveniente es que el método de ejecución del semáforo es realizado por un programador conllevando un índice de error.

Para evitarlo, los sistemas crearon una región crítica de tal manera que el compilador creará las operaciones P y V y el programador solo hará sentencias que las llamen. A esto se llama **sincronización implícita**.

Otro tipo de exclusión mutua son:

MONITORES

Surgieron para resolver los problemas de los semáforos y evitar al programador encargarse de la gestión de los recursos.

Se crea un programa monitor que será el encargado de gestionar los recursos y cada programa pedirá permiso a este para disponer de un recurso. El monitor hará todas las operaciones necesarias para asignar ese recurso al programa que lo llama. El monitor asegura que se dispone del recurso exclusivamente. El monitor permite compartir datos entre varias actividades ocultando la composición de los mismos y garantizando la exclusión mutua sin que el programador tenga que subministrarla. Los monitores se basan en:

- ◆ **Abstracción de datos**: Técnica de programación que separa las operaciones que se van a realizar con los datos de los detalles de diseño propios de los mismos. De esta forma los distintos procesos que tengan que operar con datos compartidos solo podrán acceder a ellos a través de las operaciones que les ponga a disposición el monitor correspondiente.
- ◆ **Exclusión implícita**: La finalidad de los monitores es reunir las opciones que operan sobre un conjunto de datos compartidos en un solo módulo, de manera que todos los accesos a estos datos estarán forzados a utilizar dichas funciones, ya que los datos por si mismos estarán ocultos para el mundo exterior.

Para llevar a cabo la sincronización el monitor se basa en un tipo de datos llamado **condición**, dando lugar a los **monitores condicionales** de tal manera que los programadores que prefieran sincronizar los procesos puedan utilizar una o mas variables de condición.

Las funciones que se suelen utilizar sobre una variable de condición son:

1- **.WAIT**: Bloquea el proceso hasta que no exista otro dentro del monitor.

2- **.SIGNAL**: Permite la entrada de otro proceso al monitor. Si no hay otro bloqueado por wait no tiene efecto.

Estas funciones son encargadas internamente por el procesador y no estarán accesibles al programador lo que hace que los monitores sean mas fáciles y seguros que los semáforos.

Para cada condición construiremos una cola para las actividades bloqueadas.

Para que los monitores ofrezcan un grado de servicio aceptable ningún proceso deberá permanecer durante mucho tiempo dentro del mismo.

Otro problema son las llamadas anidadas en monitores.

COMUNICACIÓN DE PROCESOS

Los procesos tienen necesidad de comunicarse; normalmente trabajan relacionados entre si y necesitan intercambiar información.

Esquema:

- Programa emisor Hace la llamada
- Programa receptor Procesa la llamada
- Mensaje Memoria compartida

Dentro de la comunicación hay dos tipos:

COMUNICACIÓN DIRECTA

♦ Síncrona: Se realiza llevando a cabo un conjunto de operaciones:

E(p, m).-Enviar mensaje al proceso P

R(q, m).-Recibir mensaje del proceso Q

♦ Asíncrona: Se llevan a cabo un conjunto de operaciones:

E(p, m).-Envía mensaje al proceso P

R(id, m).-Recibe mensaje de cualquier proceso

COMUNICACIÓN INDIRECTA

Los mensajes se envían y reciben a buzones o desde buzones. Este método soluciona la rigidez de la comunicación directa ya que los buzones simbolizan un emisor o receptor simbólico. Los buzones son recursos y los gestiona el S.O.

E(a,m).-Enviar mensaje al buzón a.

R(a,m).-Recibe el mensaje del buzón a.

Los procesos cliente y servidor se interrelacionan mandándose mensajes desde varios ordenadores teniendo un ordenador cliente que realiza una petición y un ordenador servidor que devuelve los resultados.

INTERBLOQUEO

Cuando un proceso bloquea a otro y este al anterior. Esta situación solo puede arreglarse de una manera externa que es parando la ejecución. Pasos para utilizar un recurso:

- 1-.Solicita un recurso
- 2-.Lo utiliza
- 3-.Libera el recurso

En el concepto de recurso englobamos hw y sw. Se produce inter-bloqueo porque hay un numero finito de recursos y procesos. Las operaciones sobre recursos se realizan sobre llamadas al S.O. para evitar que sea el usuario el que los gestione. Para que se de un interbloqueo tiene que darse la exclusión mutua.

-Condiciones para que se de el interbloqueo:

1-Exclusión mutua

2-Espera y retención-Los procesos retienen los recursos adquiridos mientras esperan por otro.

3-Apropiación-Un recurso no puede ser quitado a un proceso que lo posee hasta que lo ha utilizado.

4-Espera circular-Debe de existir un grupo de procesos en el que cada uno retiene uno o mas recursos que son solicitados por el siguiente proceso de la cadena.

Para el interbloqueo tienen que darse todas las condiciones anteriores.

Se establece un stma de prevención de interbloqueo que se hará asegurándose que al menos una de las 4 condiciones no se pueda dar.

La exclusión mutua no siempre se puede evitar porque existen recursos no compartibles.

Para evitar la espera y retención el s.o fuerza a que un proceso antes de comenzar deba declarar todos los recursos que necesite. So todos están disponibles el s.o le deja arrancar el proceso y si no le hace esperar hasta que estén sin retener los recursos.

Para evitar la apropiación, el s.o revisará que si un proceso A con recursos asignados pide otro no disponible deberá ser colocado en espera y liberado todos sus recursos.

Para evitar la espera circular el s.o establece que asignando un número de orden a cada tipo de recurso un proceso puede solicitar únicamente los recursos en orden ascendente.

GESTIÓN DE MEMORIA

La memoria es una sucesión de palabras o bytes con dirección propia. Cuando se desarrolla en hw del ordenador se define el esquema de direccionamiento que podrá utilizar. Para poder procesar un programa en un ordenador es necesario que previamente él y los datos manejados estén en la memoria.

El hw impone la unidad mínima de información direccionable; si esa unidad es la palabra las posiciones sucesivas de memoria corresponderán a palabras sucesivas y el ordenador será direccionable por palabra y si la unidad mínima es el byte entonces es direccionable por byte.

Las características de la memoria son:

- ◆ Velocidad de acceso: tiempo que tarda el ordenador en detener de esta la información desde que se hace una operación de escritura o lectura.
- ◆ Tiempo de ciclo de memoria: tiempo transcurrido entre la petición de dos órdenes consecutivas.

♦ **ASIGNACIÓN DE DIRECCIONES**

Los programas deben estar cargados en memoria para su ejecución pero no saben en que posición o dirección.

El compilador traduce el programa hecho por el programador a módulos y asigna a las instrucciones y campos el desplazamiento respecto a la sentencia inicial. Una vez enlazado el cargador podrá colocarlo en memoria asignándole al cero relativo una dirección de carga real y al resto los desplazamientos correspondientes.

♦ **GESTIÓN DE MEMORIA REAL**

El s.o dedica una parte importante del sw a la gestión de memoria, que es el gestor de memoria. Gestor de memoria en monoprogramación:

- Memoria dedicada: la usaban los primeros ordenadores. El programador disponía por completo del hw y utilizaba la memoria como quería. La utilización de la memoria era ineficaz y el programador tenía que conocer muy bien el hw y dedicar mucho esfuerzo a su gestión.
- División de memoria: surge como consecuencia de la introducción de los s.o. Por esto se divide en dos zonas.
 - ♦ Utilizable por el usuario.
 - ♦ Reservada por la parte residente del s.o.

Tiene problemas:

- ♦ Como asegurar la independencia de ambas zonas.
- ♦ Como asegurar que los programas de usuario no se vean afectados.
- ♦ **PROTECCIÓN DE MEMORIA**

Ya que el monitor y el programa de usuario van a compartir la memoria, es necesario proteger la zona del s.o contra cualquier intento de acceso ilegal del programa que podría afectar al funcionamiento del monitor.

Se establece una dirección de frontera que limita la zona del stma. Cualquier dirección que genere el programa de usuario se comparará dicha frontera permitiendo el acceso si es mayor o igual, degenerándolo si es menor. El control lo realiza el hw. Solo será necesario controlar las direcciones del programa. El monitor debe poder acceder a cualquier punto de la memoria y por lo tanto no es necesario controlar sus direcciones.

A consecuencia de la división de la memoria tenemos que hacer una reasignación de direcciones que consiste en:

Una vez inicializado el stma el contenido del registro frontera modificará el punto a partir del cual puede cargarse el programa de usuario. Para ello, es necesario reasignar las direcciones del programa en función de la frontera. Hay dos maneras de hacerlo:

- ♦ **Estática**: se realiza en la compilación o en la carga del programa en memoria. Así, si hay variación del tamaño del s.o exigirá una nueva compilación o carga del programa.
- ♦ **Dinámica**: se asignan las direcciones reales durante la ejecución del programa. Un dispositivo hw especial interceptará cada dirección lógica generada y le sumará el contenido del registro frontera; el resultado será la dirección real.

Cualquiera que sea la técnica elegida impedirá que el usuario maneje direcciones reales. Su programa utilizará direcciones relativas que variarán entre cero y el máximo valor permitido, y este será el espacio físico de dirección.

Después el stma con el hw establece la correspondencia con el espacio lógico de dirección, limitado entre la dirección frontera y el tamaño real de la memoria.

◇ **INTERCAMBIO DE ALMACENAMIENTO**

Lo impulsó la necesidad de atender a varios usuarios en los stmas de tiempo compartido.

Consiste en liberar espacio de memoria extrayendo un programa instalado en ella para poder ser ocupado por otro que necesite ejecutarse.

Cuando el programa cargado en memoria quede a la espera el stma lo descargará copiándolo exactamente como está en memoria en dispositivos rápidos de almacenamiento secundario en su lugar cargará el de otro usuario traído de dichos dispositivos que esté preparado para usar el procesador.

El stma deberá encargarse de todas las interrupciones que tengan pendientes los programas sacados.

◇ **MULTIPROGRAMACIÓN**

Para poder repartir el procesador entre varios procesos o usuarios, necesitamos tenerlos en memoria, por ello la memoria se divide en trozos llamados **particiones** o **regiones** donde se cargarán los procesos. El número de particiones indicará el grado de multiprogramación.

Para proteger la memoria debemos proteger las particiones de accesos indeseados. El mecanismo elegido tendrá que actuar ante cada referencia de memoria generada y deberá de ser especialmente rápido para evitar que se produzca degradación en el funcionamiento del stma.

Se pueden utilizar dos registros hw (límite) de forma que sus contenidos apunten al límite superior en interior de la partición; cada dirección generada deberá estar entre estos valores.

Esta técnica exige que las direcciones generadas por los procesos sean absolutas asignadas durante la compilación o la carga del programa en ambos casos será una asignación estática, ya que la variación del contenido de los registros límite exigirá una nueva compilación o carga,

Otra solución es cargar un registro con la dirección de comienzo de la partición y otro con el tamaño de la misma denominándose **registro base** y **límite**.

Así es posible asignar las direcciones, ya que bastará actualizar el contenido del registro base.

Cada dirección generada por el proceso deberá ser temor que el contenido del registro límite y se le sumará el contenido del registro base para obtener la dirección absoluta correspondiente.

A la hora de partir la memoria podemos hacer:

◇ **Particiones contiguas de tamaño fijo:** el numero de particiones y su tamaño se definirá al iniciar el stma y quedarán fijados para toda la sesión. Cuando tenga que iniciar un programa el stma le asignará una partición que pueda contenerlo. Será necesario que los programas declares sus necesidades al stma.

Todos los programas se colocan en la cola única de espera y será el stma el encargado de seleccionar cual cargar en la partición disponible.

El planificador de trabajos es el encargado de hacer la selección y lo puede hacer respetando el orden de llegada.

Este criterio de selección es simple de programar pero puede provocar infrautilización de la memoria; el planificador puede lograr mayor eficacia si analiza la cola de espera buscando el trabajo de tamaño más adecuado a la partición libre o el primero que quepa en la partición libre.

En algunos casos se mejora el rendimiento añadiendo técnicas de intercambio a la gestión de particiones. Los programas pendientes de un evento se pueden llevar a almacenamiento secundario cargando en su lugar otro programa.

La eficacia de esta técnica queda condicionada por la elección del tamaño y número de particiones en relación con las características de los programas a trabajar.

Da lugar a dos problemas:

- ◆ **Fragmentación interna**: cuando al asignar la partición sobra memoria que no se utiliza.
- ◆ **Fragmentación externa**: cuando las particiones libres no pueden ejecutar los trabajos por ser estas más pequeñas.

Particiones contiguas de tamaño variable: técnica de gestión que asigna la memoria a los trabajos de acuerdo con su tamaño.

El s.o. mantendrá una tabla interna donde registrará las zonas de memoria disponible. A cada trabajo le asignará una partición del tamaño solicitado descontando esa cantidad de la tabla. En este caso si queda una parte de la memoria libre que no quepa el siguiente trabajo se produce una fragmentación externa. Cada trabajo irá terminando en un tiempo determinado e irán entrando otros trabajos.

Cuando se creen fragmentaciones el gestor de memoria unificará todos los huecos con el fin de disponer de huecos mayores.

La técnica utilizada para combatir los procesos de fragmentación es la compactación. La idea es recolocar las zonas de memoria utilizadas para lograr un único hueco de memoria disponible. Los inconvenientes son:

–Coste de tiempo muy alto y exige que los trabajos sean reubicables y solo será utilizable en sistemas en los que las direcciones de memoria se asignen durante la ejecución.

- ◇ **Recubrimiento**: Permite la ejecución de un programa por partes. El programador durante el desarrollo del programa debe establecer las partes del mismo que son recubribles y definir al enlazador.

Durante la ejecución del programa determinadas partes solo se cargarán en memoria cuando se referencien y lo harán recubriendo partes ya cargadas del mismo que no necesitan.

Los programas deberán estructurarse para utilizar esta técnica. Esta estructura dependerá de la relación que exista entre las secciones. Cuando dos secciones no requieren estar cargadas en memoria pueden recubrirse mutuamente. Estas son independientes, ninguna hace referencia a la otra, por esto se les puede asignar la misma dirección de carga en la memoria real que usará cuando sean llamadas.

En general la ejecución de un programa requerirá que una o varias secciones estén permanentemente en memoria siendo desde estas desde donde se llaman las demás cuando se necesita. Este tipo permite que la carga de sección de un nivel pueda realizarse cubriendo otra sección de mismo nivel.

Todo programa que pretenda utilizar el recubrimiento deberá estructurarse teniendo en cuenta que solo se pueden recubrir entre sí las secciones independientes que son del mismo nivel ya que la ejecución de cualquier sección requiere que permanezcan en memoria todas las secciones entre ella y la raíz.

La gestión del recubrimiento necesita los servicios del stma que es el encargado de realizar la carga dinámica de los programas según la estructura. El componente de gestión de memoria de los stmas que utilizan el recubrimiento debe poseer una parte especializada en realizar la carga con recubrimiento de las secciones.

El programador mediante sentencias de control específicas define al enlazador la estructura del programa y los criterios de recubrimiento que debe seguir el stma en su carga. De esta manera el enlazador le asignará al origen de la sección raíz la dirección de memoria relativa cero y a partir de esta el origen de la sección quedará definido por cero más la longitud de las secciones entre ella y la raíz.

Todas las secciones del mismo nivel recubribles entre sí tendrán igual dirección relativa origen y serán reubicadas en las mismas direcciones reales.

◊ **Gestión de la memoria virtual:** es la técnica que combinando hw y sw permite la ejecución de programas parcialmente cargadas en memoria real. Las ventajas son:
–Si los programas se pueden ejecutar por partes la memoria lógica puede ser mayor que la real disponible.

–Como cada programa ocupa menos memoria real se puede elevar el índice de multiprogramación y la eficiencia del stma.

–Al cargar menos cantidad de programa se necesitan menos operaciones de e/s para las operaciones de carga.

Esta técnica surgió con la finalidad de permitir que pudiesen trabajar muchos usuarios en el ordenador a pesar de contar con escasa memoria.

El s.o toma solo la parte de instrucciones y datos realmente necesarios los copia en la memoria dejando el resto en el disco. Solo cuando necesite las partes del disco las traerá a la memoria y devolverá al disco lo necesario.

◊ **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 s.o internamente mantiene una tabla de páginas donde relaciona cada página cargada en memoria principal con el frame que la contenga. Cada dirección que genere el procesador será interceptada y dividida en dos componentes: un número de página y un desplazamiento en la página que indica la posición que ocupa el byte indicado contando desde el principio de la página.

Utilizando el número de página el stma 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. La arquitectura del ordenador establece el tamaño de las páginas y de los frames que suelen ser potencia de 2.

El stma 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

Los programas se desarrollan en torno a un cuerpo central desde el que se bifurca a otras partes, rutinas o se accede a zonas de datos.

Desde este punto de vista el programa es un conjunto de componentes lógicos de tamaño variable, es decir, un conjunto de segmentos; de esta manera, el espacio lógica de dirección se puede considerar como un conjunto de segmentos 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 se 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. Su funcionamiento es:

–El usuario diseña su programa modular, con módulos de tamaño variable referidos por su nombre. Los módulos o segmentos se numeran cada dirección de un segmento se expresa como segmento y desplazamiento y la realiza el compilador considerando que cada segmento tiene una dirección relativa cero en su comienzo. Los segmentos del programa se definen en tiempo de compilación y cada uno es variable en longitud debiéndose definir previamente la longitud.

El hw necesario para la segmentación es: al cargar el programa en memoria el stma crea una tabla de segmentos con la dirección real base y el límite de cada segmento (longitud). Un hw especial mediante la tabla de segmentos transforma la dirección lógica en la física.

GESTIÓN ENTRADA SALIDA

Es cualquier transferencia de INF desde o hacia la memoria o procesador. Comprende la transferencia entre niveles de memoria como la comunicación con el exterior.

Un dispositivo de e/s es el capaz de transferir INF entre la memoria o procesador y cualquier soporte de INF externa. Toda la transferencia es ordenada por el procesador central pero para evitar que el procesador lo haga de forma directa se han diseñado los dispositivos de e/s que se llaman unidades de acceso directo a memoria y en general canales de transferencia.

HW E/S

Las características que agudizan el problema de e/s son:

◇ **Operación asíncrona**: en ella los tiempos de transacciones de CPU con la memoria y

los dispositivos de e/s son muy diferentes. La memoria informa a la CPU con un tiempo fijo dado por ciclos de reloj del procesador, pero los dispositivos de e/s son muy variables con respecto a los tiempos de respuesta a las peticiones de la CPU y actúan asíncronamente respecto a la unidad central.

Como la operación asíncrona es impredecible el hw de e/s ha de utilizar señales de permiso de tráfico para indicar los momentos de la transferencia.

◇ **Diferencia de velocidades entre los dispositivos de e/s, la CPU y la memoria.**

◇ **Conversiones de formato necesarias entre los periféricos y el stma:** mientras unos requieren transferencia en serie, los otros lo transmiten en paralelo.

Los dispositivos de e/s se pueden clasificar en:

- ◆ **Dispositivos de bloque:** suelen estar asociados a dispositivos de alta velocidad que transmiten datos mediante técnicas de acceso directo a memoria.
- ◆ **Dispositivos de carácter:** están controlados por la CPU y transmiten carácter a carácter.

Controladores de dispositivos

Las unidades de e/s tienen una parte mecánica que es el dispositivo, y otra electrónica que es el dispositivo, y otra electrónica llamada controlador de dispositivos y actúa como intermediario entre el ordenador y los dispositivos.

El hecho de ser separar el controlador del periférico es para poder conectar varios periféricos de igual tipo a un acoplador único. Así, aunque un solo periférico puede transmitir información, los demás, de forma paralela, pueden hacer operaciones sin e/s.

El controlador suele ser una tarjeta de circuitos impresos que posee un cable que va al dispositivo y a su vez todos los cables del dispositivo se unen en el bus del stma, que va unido a los componentes del ordenador.

Un controlador convierte en flujo de bytes en serie transmitidos desde un dispositivo en un bloque de bytes para la CPU y realiza las correcciones de los errores que se puedan cometer en la transmisión. El controlador dispone de 3 capas:

- ◆ **interfaz con el bus.**
- ◆ **controlador.**
- ◆ **interfaz del dispositivo.**

La parte intermedia del controlador se compone de un conjunto de registro denominado puerto de e/s. las categorías funcionales más importantes encontradas en los puertos de e/s son:

- **Registro de entrada y Registro de salida:** guardan los datos hasta que la CPU o el dispositivo destino esté preparado para recibirlos.
- **Registro de estado:** nos sirven para controlar el estado de los diferentes dispositivos.
- **Registro de órdenes:** se encargan de transferir las órdenes de e/s entre la CPU y los dispositivos de e/s.

ACCESO DIRECTO A MEMORIA

Dispositivo que permite la transferencia de información entre la memoria y los periféricos o

viceversa sin canal intermedio. Para actuar necesita elementos hw:

- ◊ Registro que almacene la dirección de memoria desde o hacia donde se produce la transferencia.
- ◊ Registro contador para guardar la longitud del bloque a transferir.
- ◊ Un byte que indique si la operación es de lectura o escritura.
- ◊ Un bloque de control que regula el funcionamiento del stma.

Para realizar una transferencia el procesador iniciará el acceso directo a memoria facilitándole la dirección de memoria, la longitud del bloque, el identificativo del periférico y el sentido de la transferencia.

Después, el acceso directo a memoria, actuará de forma autónoma preparando la transferencia y solicitando el bus para el acceso a memoria.

Cuando la transferencia se completa se produce una interrupción para informar al procesador del final. Las transferencias se pueden hacer por:

- ◊ **Ráfagas:** el acceso directo a memoria toma el control del bus en modo exclusivo y hasta su finalización.
- ◊ **Robo de ciclo:** el acceso directo a memoria toma el bus durante un ciclo enviando una palabra cada vez.
- ◊ **Transparente:** se aprovechan los ciclos en los que el procesador está parado con respecto al bus.

CANALES DE E/S

Un canal es una vía de transmisión de información en sentido lógico y físico. Es un pequeño ordenador con capacidad para ejecutar sus propios programas llamados programas de canal, almacenados en la memoria del ordenador principal permitiendo liberar a la CPU de las operaciones de e/s. una vez que han comenzado a ejecutarse y aprovechando el tiempo para ejecutar un programa diferente hasta que el canal haya terminado su operación de e/s ejecutada en paralelo con el trabajo de la unidad central.

El uso del canal se debe a que no toda la información está almacenada dentro de la memoria. La información contenida en los soportes y leída o escrita por los periféricos ha de ser transmitida hacia o desde el ordenador utilizando los canales. La transferencia consta de estas operaciones:

–El canal es lanzado por una instrucción especial del procesador, mediante la que se inicializan la palabra de estado que contiene un contador e indicadores de estado de la operación y se indica la dirección donde se encuentra la primera dirección a realizar. También se señala la dirección de periférico. El canal selecciona el periférico y detecta el estado en que se encuentran para ver si la transferencia es posible. El canal se encarga también de enviar órdenes de control al periférico, definir registros en memoria y en periférico para la transferencia y definir las acciones a tomar a su término.

TIPOS DE CANALES

◊ **Selectores:** su misión es enlazar los dispositivos de e/s rápidos con la CPU. Estos canales trabajan en la modalidad de transmisión de información llamada a ráfagas consistente en dedicar la actividad del canal en forma de colas de espera de información de manera que hasta que no termina la operación de transferencia de un bloque de información de un periférico no comienza con la transferencia de otro.

- ◊ **Multiplexores:** enlazan los dispositivos de e/s lentos con la CPU. Estos, trabajan en la modalidad de transmisión de información multiplex, consistente en repartir la actividad del canal en el tiempo transfiriendo en uno u otro sentido un solo carácter de cada unidad periférica lenta.

PRINCIPIOS DEL SW DE E/S

La idea básica del sw de e/s consiste en organizarlo como una serie de estratos que oculten las peculiaridades del hw y presenten interfaz sencillo.

Un concepto importante en el diseño del sw de e/s es la independencia del dispositivo, es decir servir para cualquier dispositivo sin distinción.

También es importante el control de errores que ha de hacerse lo más próximo a hw posible. Al la hora del diseño de e/s debemos distinguir entre los dispositivos dedicados a un solo propósito y dispositivos compartidos por varios. Para lograr todo esto el sw e/s se estructura en 4 niveles:

- ◆ **manejador de interrupciones:** es el estrato inferior más oculto de la estructura. Cuando se emite un comando y se espera una interrupción el manejador de dispositivo queda bloqueado y se desbloquea cuando se presenta la interrupción.

Los procedimientos de interrupciones comienzan con una orden de grabar para salvar los registros del proceso de ejecución, después se llama a un manejador que controla la interrupción y devuelve el control o petición que provocó la interrupción.

- ◆ **manejadores de dispositivos:** su trabajo consiste en aceptar las solicitudes de sw y comprobar que cumplen los requisitos para ser atendidas.

En general, cada manejador se hace cargo de un dispositivo o de un grupo de dispositivos relacionados. Cuando se emite una solicitud el manejador puede iniciar la ejecución si el dispositivo está libre si no debemos guardarlo en una lista de espera.

- ◆ **sw e/s independiente del dispositivo:** existen un conjuntote funciones comunes para todos los dispositivos y estas se muestran en un interfaz uniforme de cara al usuario. Entre ellas destacan la información de errores, asignación y desasignación de dispositivos, protección de dispositivos y designación del tamaño de los bloques.
- ◆ **sw del usuario:** una pequeña parte del sw de e/s no reside en el s.o, sino que está grabado en bibliotecas que los usuarios manejan mediante programas enlazados a las mismas.

La idea consiste en grabar una directorio especial con los programas que van al dispositivo y un programa manejador de dicho directorio que selecciona el siguiente proceso cada vez que el dispositivo queda libre.

ENTRADA SALIDA CONTROLADA POR PROGRAMA

Los dispositivos de E/S sincronizan sus actividades con la CPU mediante protocolos de intercambio. Mediante el bus de sistemas y los registros de estado y de órdenes se proporcionan las señales para dicha sincronización.

Para comenzar con una operación de E/S es imprescindible tener conocimiento sobre el estado del dispositivo; cuando la CPU quiere transmitir o recibir datos lanza una orden de arranque, entonces el punto de E/S establece un byte de estado en la posición de ocupado;

cuando el dato ha sido transmitido el byte cambia a modo de preparado para indicar que el dispositivo puede ser utilizado para una nueva operación.

En general a cada canal se le conectan varios dispositivos periféricos y aunque las formas de conexión son muy variadas hay dos que se utilizan con mucha frecuencia:

- ♦ **conexión en serie**: cada canal tiene asignada una conexión fija a un grupo de periféricos determinados; para conectar con un periférico, hay que hacerlo por su canal y si está ocupado por otro periférico deberá de esperar, por lo que se establecen prioridades.
- ♦ **conexión en red**: cada canal se conecta con todos los periféricos permitiendo transferencias alternativas para los periféricos. Si un canal está ocupado se transmite por otro; como cada canal está conectado a varios periféricos ha de repartir el tiempo entre ellos.

Las formas de repartir el tiempo son:

- **multiplexor**: el canal atiende a los periféricos repartiendo el tiempo en forma circular.
- **canal selector**: un periférico comienza una transferencia retiene el canal correspondiente hasta el final de la misma.

En el método de E/S controlada por programa la CPU repite continuamente un ciclo de comprobación de estado del dispositivo, hasta que se encuentre preparado para una nueva transferencia. Se repite un mismo bucle tanto para efectuar una salida como para efectuar una entrada, a este bucle se le llama espera activa.

Un método alternativo es el escrutinio, que consiste en utilizar el bucle de comprobación de la CPU a los diferentes dispositivos y pasar al programa de E/S el primer dispositivo preparado. El escrutinio se puede hacer comenzando cada vez con el primer dispositivo o de forma cíclica para evitar prioridades.

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.

GESTION de archivos y dispositivos

Un stma de archivos es la parte del stma operativo encargada de administrar el almacenamiento secundario. Es el responsable de confeccionar los medios para el almacenamiento de la información y para que esta sea compartida por los usuarios de forma controlada.

Las funciones más importantes que debe realizar un stma de gestión de archivos son:

- ◇ **Creación de archivos**: consiste en dar un nombre y reservar un espacio en disco.

- ◇ **Borrar archivos:** es liberar el espacio que ocupa.
- ◇ **Abrir archivos:** para que sean utilizados con las características específicas.
- ◇ **Cerrar archivos:** finalizar un proceso.
- ◇ **Consultar archivos:** en modo lectura, escritura y modificación.

A demás de las operaciones básicas el Stma de archivos debe proporcionar otras ayudas a la gestión como puede ser:

- ◇ Posibilidad de compartir archivos entre usuarios.
- ◇ Transferencia de datos entre archivos.
- ◇ Estructurar los archivos de la forma más adecuada a cada aplicación concreta.
- ◇ Gestión de los soportes físicos y dispositivos que los manejan.
- ◇ Seguridad y protección garantizando la integridad de los datos, derechos de acceso y condiciones de utilización

En general los datos están jerarquizados según un modelo general:

El primer nivel está formado por los bits o dígitos binarios. Su agrupación constituye los bytes o caracteres.

A un nivel superior estarán los campos que cuando se agrupan según una relación común forman los registros.

El archivo será una agrupación de registros relacionados entre sí y el conjunto de archivos constituye una base de datos.

Los componentes de un stma de archivos se organizan de forma jerárquica dependiendo de los niveles inferiores.

La estructura básica consta de:

- ◆ **rutina del stma de archivos simbólicos:** que utilizando el nombre del archivo localiza la entrada de dicho archivo en el directorio.
- ◆ **la rutina de archivos básicos:** utiliza un identificador que contiene las informaciones sobre el tamaño, nº de registros, tipo de acceso y dirección de comienzo del archivo
- ◆ **rutina de control de acceso:** controla el acceso al archivo. Si la petición es valida pasa al siguiente nivel y si no dará un error.
- ◆ **rutina del stma de archivos lógicos:** convierte el registro lógico en un byte obteniéndose la longitud y una dirección.
- ◆ **rutina del stma de archivos físicos:** utiliza la dirección del byte más una dirección del comienzo del archivo para determinar el bloque físico requerido.
- ◆ **rutina de asignación:** se encarga de llevar el control de los bloques no utilizados de cada dispositivo.
- ◆ **rutina de estrategia para dispositivos:** convierte el número del bloque físico al formato requerido por el dispositivo donde se encuentra almacenado.
- ◆ **rutina de planificación de e/s:** encargada de realizar la lectura del bloque solicitado.

FORMAS DE PLANIFICACIÓN

–Existen muchos algoritmos de planificación unos tratan de reducir el tiempo de búsqueda y otros mejoran el tiempo de rotación de las cabezas. Tipos:

1-.FCFS

La primera petición que llega es la primera que se atiende. Ninguna petición puede ser desplazada por la llegada de otra prioritaria. Es la más sencilla pero la menos eficiente.

2-.SS FT

Se sirve la petición que se encuentra a la distancia mas corta de la pista que se acaba de servir.

3-.SCAN

Selecciona la petición que da una distancia de búsqueda más corta pero en una dirección preferida. La cabeza de lectura/escritura empieza en un extremo del disco y se mueve hacia el otro sirviendo las solicitudes a medida que llega a cada pista. Cuando llega al extremo opuesto se invierte la dirección del movimiento y continúa con el proceso. Esta es la base de la mayoría de las planificaciones actuales.

4-.C-SCAN

Es una variante de la anterior que proporciona un tiempo de espera mas uniforme. Se mueve la cabeza de un extremo a otro sirviendo las solicitudes según se van alcanzando. Cuando llega al otro extremo se vuelve inmediatamente al principio sin servir ninguna solicitud para comenzar un nuevo recorrido.

5-.LOOK y C-LOOK

Son planificaciones scan en las cuales la cabeza de lectura escritura no llega hasta los extremos del disco sino solo hasta que no quedan solicitudes en la dirección actual.

GESTIÓN DEL ESPACIO LIBRE

En cualquier sistema informático los ficheros se crean y borran frecuentemente. Es necesario reutilizar el espacio que ocupaban los ficheros borrados y por tanto tener un sistema adecuado de control de los espacios libres. Una forma de tener localizados los espacios libres consiste en confeccionar una lista que contiene los bloques del disco que no están asignados. Cuando se va a crear un fichero se consulta dicha lista y una vez asignado dicho espacio se borra de la misma. Por el contrario, si el fichero se borra se añade a la lista. La lista de bloques libres se puede implementar de diversas formas. Una consiste en representar cada bloque con un bit. A los libres se les asigna un 0 y a los bloques ocupados un 1. Otra forma consiste en formar una lista en la que cada bloque libre mantiene un puntero al siguiente bloque no asignado.

ASIGNACION DEL ALMACENAMIENTO

La estrategia de asignación del almacenamiento secundario está relacionada con la eficacia de los accesos a los ficheros. La velocidad de realización de un proceso varía en función del tiempo que se requiere para acceder a los bloques que contienen la información necesaria para realizar dicho proceso. Existen dos estrategias básicas de asignación:

--Asignación contigua

Dicho método establece que cada fichero ocupe un conjunto de bloques contiguos en el disco. Teniendo la dirección de comienzo y el tamaño del fichero se puede acceder a cualquier bloque del mismo. Tanto el acceso secuencial como el acceso directo son fáciles en dicha asignación, sin embargo la asignación de espacio a un nuevo fichero puede ocasionar

problemas de fragmentación. Por otra parte tenemos que calcular previamente el tamaño del fichero.

◇ **Asignación no contigua**

Tiene dos métodos:

-Asignación enlazada

Cada fichero es una lista enlazada de bloques. Cada bloque puede encontrarse en cualquier parte del disco. Unos cuantos bytes de cada bloque se reservan para señalar al bloque siguiente. El primer bloque está en una dirección marcada por un puntero de encabezamiento grabado en el directorio.

Ventajas:

-Facilidad de eliminar bloques

-Inserción de nuevos bloques

Inconvenientes:

-El acceso directo es lento

-La pérdida de un puntero hace inaccesibles muchos bloques

-Asignación indexada

Es una mejora de la asignación enlazada. También mantiene punteros a los bloques asignados pero además reúne los punteros en bloques de índices para facilitar el acceso directo. El directorio contiene la dirección del bloque de índices donde están contenidos los bloques del archivo

DIRECTORIOS

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.

33