

## TEMA 1

### INTRODUCCION.

- Administración de procesos.
- Concurrencia de procesos.
- Protección y seguridad informática.
- Sistemas distribuidos.
- Sistemas operativos reales.

**Proceso.**– Es un programa en ejecución. Mientras que el programa es una entidad pasiva, el proceso es una entidad activa. Un proceso es la entidad que puede ser asignada al procesador y ejecutada por este.

Se pueden conseguir mejoras en el rendimiento si en lugar de ejecutar un programa dentro de otro, introducimos diferentes procesos concurrentes en el procesador (Programación concurrente o multitarea).

La **multiprogramación** (además de lo anterior supone gestión de memoria y archivos). Fue diseñada para mantener ocupados a la vez al procesador como a los dispositivos de E/S incluidos los dispositivos de almacenamiento de modo que se alcance el máximo rendimiento.

La clave de este mecanismo es que, como respuesta a las señales que indiquen que ha terminado una operación de E/S el procesador cambie entre los diversos procesos en memoria.

El SO es el encargado de suministrar el software que ayude al hardware a la gestión de procesos :

- Creación / Eliminación.
- Control de avance y retroceso.
- Interrupciones.
- Gestión de errores.
- Asignación de recursos hardware.
- Proveer de método de comunicación entre diferentes procesos (mensajes u señales).

Un proceso, es un conjunto de ráfagas de CPU y E/S teniendo siempre en cuenta que la primera y la última ráfaga van a ser siempre de CPU.

- Desde el punto de vista del usuario .– Un proceso es un conj. De operaciones e instrucciones específicas destinadas a obtener un resultado a partir de una entrada de datos.
- Desde el punto de vista del operador del sistema.– Se trata de informar al SO del conj. De las actividades que deben ser planificadas para conseguir ejecución concurrente.

Residir en memoria.

Atributos Recursos que necesita, Ficheros, Prioridades.

El que un mismo proceso pueda dividirse en tareas independientes requiere, código complejo, introducción de señales. El programador de sistema tendrá que tener todo esto en cuenta a la hora de diseño.

- Desde el punto de vista del SO .– Es la entidad mínima individualmente planificable. Va a constar de código y datos y se caracteriza por sus atributos y su estado dinámico.

- Un proceso puede a su vez estar formado por varias tareas o hilos que pueden ejecutarse en modo concurrente. La unidad mínima ejecutable es la **tarea o hilo**.

Diferencias entre proceso e Hilo :

- Los procesos tienen asignados diferentes secciones de memoria mientras que los hilos no.
- A los procesos se les asigna recursos mientras que los hilos disponen de los recursos asignados al proceso.
- A la hora de planificar si se pueden planificar los hilos de un mismo proceso.

Un proceso a lo largo de su vida en el sistema va a pasar por varios estados :

- Nuevo
- Listo.– Estructuras de datos.
- Ejecución.– CPU.
- Bloqueo.– Espera de E/S o Evento.
- Suspendido.– Suspender proceso por requerimientos del sistema.
- La concurrencia supone , que a un proceso al se le asigna CPU sale de ejecución y se le asigna la CPU a otro proceso, este proceso en espera puede estar esperando evento o en cola de espera.
- Esto supone que los procesos entren y salgan de la CPU sin finalizar su ejecución. Por tanto, debe de haber algo donde se guarda la información del proceso y que informe del estado del nuevo proceso.
- Toda la información relativa a estos se guarda en el BCP. Que consiste en una serie de posiciones contiguas de memoria donde se guarda el identificador del proceso, puntero a la siguiente instrucción, ficheros abiertos, recursos asignados, lista de recursos solicitados, prioridades, etc.
- En el momento de creación del proceso hay que crear un bloque de control de procesos y hay que ver cual es el sistema de gestión de memoria del sistema.

Si la memoria se gestiona estática se reserva un espacio para contener TBCP (numero de procesos limitados al numero de bloques de la tabla).

Gestión dinámica.– Donde no existe limitación de procesos.

Cuando un proceso comienza a ejecutarse vuelca el contenido del BCP en los registros de la CPU y cuando sale de CPU se realiza la operación contraria (**Cambio de contexto**) Para que el rendimiento de un sistema con multiproceso sea optimo el tiempo empleado en el Cambio de contexto tiene que ser despreciable frente al tiempo de CPU.

- Los procesos van a provenir de 3 sistemas.–
- Procesos por lotes.
- Tiempo compartido.– Se asigna tiempo (cuanto) a cada proceso. Si todos los procesos tienen la misma prioridad el cuanto es el mismo, pero los procesos de mayor prioridad tienen un cuanto mayor.  
**Procesos en Background** .– Se ejecutan en tiempos de CPU libre, estos no entran en competición por cuantos de tiempo sino que van a la cola de Background.
- Tiempo real.– Precisan tiempo de respuesta del orden de mili o microseg., que desalojan el proceso que está en CPU.
- En estos tipos de procesos hay que tener en cuenta 4 principales causas de error :
- Sincronización incorrecta.– Mal diseño del mecanismo de señales. Puede dar lugar a pérdida de señales o recuperación duplicada de estas

- Fases de exclusión múltiple.– Un proceso puede dividirse en hilos, procesos de código independiente, esto es inviable, porque hay secciones de código críticas que hace volver al proceso a la ejecución secuencial, solo un proceso de los que esta operando puede seguir ejecutando y a los mecanismos empleados para que esa ejecución no sea concurrente se los llama **mecanismos de exclusión mutua**. Es habitual el caso en que dos usuarios o dos procesos intenten acceder a la vez a un recurso compartido. Si no se controlan estos casos puede producirse un error. Debe existir un mecanismo que permita que solo un proceso pueda realizar una transacción sobre una zona de datos.
- Funcionamiento no determinista del programa.– Los resultados del programa debe depender solo de las entradas de ese programa y no de otros procesos en ejecución paralela, por ello no se deja acceder a memoria asignada a otro proceso y se da la canalización de datos de entrada.
- Interbloqueo.– Es posible que dos o más procesos estén suspendidos uno a la espera del otro. Cada uno está deseando que el otro libere un recurso deseado. La única forma de resolver interbloqueo es suspender uno de los procesos para liberar los recursos y que el resto pueda finalizar y así el proceso suspendido pueda terminar también.
- **Protección y seguridad de la información.** (Publicación de la oficina de Estandares).

El crecimiento de la utilización de sistemas en tiempo compartido y de redes de computadoras ha traído consigo un aumento por la preocupación por la protección de la información

- Intentos organizados y deliberados de obtener información de las organizaciones del sector privado.
- Intentos organizados y deliberados de obtener información de las oficinas del gobierno.
- Adquisición inadvertida de información económica y mercantil.
- Adquisición inadvertida de información sobre personas.
- Intromisión del gobierno en los Derechos Individuales.
- Intrusión en Bases de Datos. (En la adquisición de datos financieros, económicos, y personales.)
- Atropello de derechos individuales por la comunidad de información.
  - Pueden construirse algunas herramientas de propósito general dentro de las computadoras y de los SO para dar soporte a varios mecanismos de protección y seguridad.
  - En general, interesan las problemas de control de acceso a las sistemas informaticos y a la información almacenada en ellos.
  - Se han definido cuatro clases de políticas generales de protección:
- No compartición.– Los procesos están completamente aislados unos de otros y cada proceso tiene control exclusivo sobre recursos que le fueron asignados. Los procesos suelen compartir programas o archivos de datos haciendo copias y pasandolas a su propia memoria virtual.
- Compartiendo originales de programas o archivos de datos.– Con el uso de una unica copia reentrante, una unica copia fisica de un programa puede aparecer en varios espacios de memoria virtual, como archivos de solo lectura. Se requieren mecanismos especiales de bloqueo para compartir archivos de datos en los que se puede escribir y evitar que usuarios simultaneos interfieran unos con otros.
- Subsistemas confinados o sin memoria.– En este caso los procesos se agrupan en subsistemas para cumplir una politica de protección en particular. Por ejemplo, un proceso `cliente' llama a un proceso `servidor' para llevar a cabo cierta tarea con datos. El servidor se protegerá de que el cliente no descubra el algoritmo con el cual realiza su trabajo y el cliente se protegerá de que el servidor no descubra que tipo de trabajo esta realizando.
- Diseminación controlada de información .– A los usuarios y a las aplicaciones se les da credenciales de seguridad en un cierto nivel, mientras que a los datos y recursos se les dota de clasificaciones de seguridad. La politica de seguridad hace cumplir las restricciones relativas a que usuarios tiene acceso a que clasificaciones.
- Gran parte del trabajo realizado en cuanto a protección y seguridad se puede organizar en 3

categorías :

- Control de acceso (En cuanto al acceso a la información).– Regulación de acceso al usuario al sistema completo, a los subsistemas y a los datos, así como regular el acceso a recursos del sistema.
- Control del flujo de información, regulando flujo de información que se suministra al usuario.
- Certificación.– Demostración de que los mecanismos de seguridad funcionan. Demostración de que el acceso y los mecanismos de control se llevan a cabo de acuerdo a las especificaciones y a que estas cumplen las políticas de protección y seguridad deseadas.

**Capas del Sistema Operativo** (Clasificación genérica págs 73–75). .–

<b>Capa 13</b>	SHELL
<b>Capa 12</b>	Procesos de usuarios.
<b>Capa 11</b>	Directorio
<b>Capa 10</b>	Dispositivos
<b>Capa 9</b>	Sistemas de archivos.
<b>Capa 8</b>	Comunicaciones
<b>Capa 7</b>	Memoria virtual.
<b>Capa 6</b>	Almacenamiento secundario.
<b>Capa 5</b>	Procesos primitivos
<b>Capa 4</b>	Interrupciones
<b>Capa 3</b>	Procedimientos
<b>Capa 2</b>	Conjunto de instrucciones
<b>Capa 1</b>	Circuitos electrónicos

- Capa 1.– Circuitos electrónicos. Objetos tratados son registros, celdas de memoria, puertas lógicas.
- Capa 2.– Conjunto de instrucciones del procesador. Las operaciones de este nivel son las permitidas por el conjunto de instrucciones del lenguaje máquina.
- Capa 3.– Añade concepto de procedimiento o subrutina. Operaciones de llamada y retorno.
- Capa 4.– Introduce interrupciones, las cuales hacen que procesador salve el contexto actual e invoque rutina de tratamiento de interrupción.

--- Las cuatros primeras capas anteriormente citadas no pertenecen al SO sino al hardware de la máquina.

---

- Capa 5.– Se introduce noción de proceso como programa en ejecución. Entre requisitos fundamentales de un SO que ofrezca soporte para varios procesos incluye capacidad de suspender y reanudar un proceso (Exige guardar cambios de contexto Capa 4). Si un proceso necesita cooperación será necesario un sistema de sincronización. (Técnica sencilla Semáforo).
- Capa 6.– Dispositivos de almacenamiento secundario del ordenador. Se incluyen procesos para la planificación de estos dispositivos.
- Capa 7.– Crea un espacio de direcciones lógicas para procesos. Organiza el espacio de direcciones virtuales en bloques que se pueden mover entre memoria principal y secundaris.

3 esquemas.–

- Paginación de longitud fija.
- Segmentación de longitud variable.
- Las dos anteriores.

- Capa 8.– comunicación de información y mensajes entre procesos (Nivel 5 proporciona mecanismos de señalización). Una de las herramientas más potentes Tubo Canal lógico para flujo de datos entre procesos.
- Capa 9.– Da soporte a almacenamiento a largo plazo de archivos con nombre. Los datos de almacenamiento secundario se contemplan en terminos de entidades abstractas de longitud variable en contraste con el enfoque a hardware del Nivel 6.
- Capa 10.– Acceso a dispositivos externos.
- Capa 11.– Mantiene asociación entre identificadores externos de recursos y objetos del sistema Estos se mantiene en un directorio. Las entradas no incluyen solo lo anterior sino también otras características.
- Capa 12.– Servicio completo de soporte a proceso. Se da soporte para la gestión ordenada de toda la información asociada a la gestión de proceso. Esto incluye, espacio de direcciones virtuales, lista de objetos y procesos con los que puede interactuar y sus restricciones, los parametros pasados al proceso.
- Capa 13.– Interface de usuario con sistema operativo.

### **Sistemas Operativos Reales.–**

- WINDOWS NT Sistema monousuario multitarea diseñado para trabajar con PC y Estaciones de trabajo. Incorpora los últimos adelantos.
- UNIX Sistema operativo multiusuario dirigido a todo tipo de computadoras. Siempre en grandes redes. Compatibilidad absoluta.
- OS/2 Sistemas de oficinas.

## **TEMA 2.**

### **ADMINISTRACION DEL PROCESADOR. PLANIFICACION DE MULTIPROCESADORES.**

Los requisitos principales que debe satisfacer un SO estan expresados haciendo referencia a los procesos :

- Un SO debe intercalar la ejecución de un conjunto de procesos para maximizar la utilización del procesador ofreciendo a la vez un tiempo de respuesta razonable.
- El SO debe asignar los recursos a los procesos en conformidad con una politica especifica, evitando al mismo tiempo el interbloqueo.
- El SO podría tener que dar soporte a la comunicación entre procesos y la creación de procesos por parte del usuario, etc.

#### **– Descripción y control de procesos.**

Para poder diseñar el SO de una forma efectiva, se necesita tener un modelo claro de comportamiento de un proceso.

El modelo más sencillo que puede construirse tiene en cuenta que un proceso puede estar ejecutandose en el procesador o no.

Un proceso puede estar en ejecución o en espera.

El proceso entra, y esta en un estado de no ejecución hasta que es seleccionado por administrador.

El administrador del sistema operativo, selecciona un proceso para que se ejecute. El proceso que estaba ejecutandose pasa del estado de ejecución al estado de Espera y uno de los demás procesos pasará al estado de Ejecución.

Del estado de EJECUCION puede pasar a PAUSA o TERMINAR.

Incluso en este modelo tan simple se aprecia la necesidad de utilizar una serie de estructuras de datos para almacenar el estado actual de cada proceso.

Aquellos procesos que no están ejecutándose tienen que guardarse en algún tipo de cola, para que esperen su turno.

La cola consiste en una lista enlazada de bloques de datos en la que cada bloque representa un proceso.

El comportamiento del distribuidor se podrá describir como diagrama de colas. Cuando un proceso se crea hay que construir una estructura de datos para administrar ese proceso y se le asigna unas posiciones de memoria.

### **Creación y Terminación de procesos.**

Independientemente del modelo utilizado para su gestión los procesos tienen su vida limitada por el momento de creación y terminación.

- **Creación** Cuando se añade un proceso, hay que construir las estructuras de datos que se utilizan para administrar el proceso y asignar el espacio de direcciones que va a utilizar dicho proceso.

Hay diferentes razones por las que se puede crear un proceso.

- **Nueva tarea en proceso por lotes.**– Un proceso se crea como respuesta a la emisión de un trabajo.
- **Nueva conexión iterativa.**– Se crea un proceso cuando un nuevo usuario intenta conectarse.
- **Proceso del SO para dar un servicio.** – El SO crea un proceso como parte de una aplicación (Tercera solicitud de impresión el SO crea un proceso para gestionar la solicitud).
- **Generado por otro proceso ya existente.**– Tradicionalmente todos los procesos eran creados por el SO de una forma transparente para el usuario o para el programa de aplicación. Sin embargo, puede ser útil permitir que un proceso pueda originar la creación de un proceso. El nuevo proceso se ejecutará de forma paralela con la aplicación.

Cuando un proceso es creado por el SO bajo la solicitud de otro proceso, este mecanismo se conoce como **Generación de procesos**. Donde el generador se conoce como **proceso Padre** y el proceso generado es el **proceso Hijo**.

- **Terminación** En cualquier sistema informático, debe existir alguna forma de que un proceso pueda indicar que ha terminado.

En un proceso por lotes debe indicarse al final la instrucción HALT (Terminación) y en un entorno iterativo debe ser el usuario quien da la orden de terminación, etc.

Razones para la terminación de un proceso :

- **Normal** .– El proceso ejecuta llamada al servicio del SO que notifica su terminación normal.
- **Por tiempo excedido.**– Una tarea tiene asignada la CPU ejecuta el tiempo máximo y le es requisada la CPU y pasa a la cola para competir por la nueva asignación.

- **Violación de límites.**– Proceso trata de acceder a una posición de memoria que no le esta permitida acceder.
- **No memoria disponible.**– El proceso necesita más memoria de la que el sistema puede proporcionar.
- **Error de protección.**– El proceso intenta utilizar un recurso o archivo que no le esta permitido utilizar, o trat de utilizarlo de forma incorrecta.
- **Error aritmético.**– Si el proceso intenta hacer un calculo prohibido, como la división por cero, o trata de acceder a un número mayor del que el hardware acepta.
- **Tiempo máximo de espera de recurso.**– El proceso ha esperado más alla del tiempo máximo especificado para que se produzca cierto suceso.
- **Fallo de dispositivo de E/S.**– Se produce un error en una operación de E/S
- **Instrucción no valida.**– El proceso intenta ejecutar una instrucción inexistente ( a menudo como resultado de un salto para ejecutar datos en la zona de datos)
- **Intento de acceso a una instrucción privilegiada.**– El proceso intenta utilizar una instrucción reservada para el SO.
- **Finalización del padre.**– Cuando un proceso padre finaliza, el SO puede diseñarse para teminar automaticamente con todos sus descendientes.
- **Mal uso de los datos .**– Un elemento de dato, no esta inicializado o es de un tipo equivocado.
- **Intervención del operador o del SO.**– Por alguna razon el operador o el SO termina con un proceso (ej : Interbloqueo).
- **Solicitud del padre.**– Un proceso padre tiene normalmente autoridad para terminar con cualquiera de sus hijos.

### Estados de un proceso.

Algunos procesos en el estado ejecución estan listos para ejecutar, mientras que otros están bloqueados, esperando a que termine una operación de E/S . (Por lo que el modelo de dos estados no sirve).

Asi pues utilizando una cola sencilla el distribuido busca el proceso más antiguo de la cola y que no este bloqueado

Una forma mñas natural de afrontar esta situación es dividir el estado de No ejecución en dos : Lista y Bloqueado. Pro tanto los 5 estados de este nuevo modelo son :

- **Ejecución** El que esta actualmente en ejecución.
- **Listo Preparado** para ejecutar en cuanto se le de la oportunidad.
- **Bloqueados** No puede ejecutar hasta que se produzca un cierto suceso. ( Terminación de una operación de E/S)
- **Nuevo Proceso** que se acaba de crear pero que todavia no ha sido admitido por el SO en el grupo de procesos ejecutables.
- **Terminado** Un proceso que ha sido excluido por el SO del grupo de procesos ejecutables.

Los estados Nuevo y Terminado son construcciones muy utiles para la gestión de procesos

- El estado de Nuevo corresponde a los procesos que acaban de ser definidos.

El SO define un proceso nuevo en dos pasos :

- Primero, lleva a cabo tareas de gestión intera.
- Se le asocia un identificador al proceso y se construyen y asignan algunas tablas necesarias para gestionar el proceso. (En este punto el proceso llega al estado de nuevo. El SO operativo ha llevado a cabo las operaciones necesarias para crear proceso pero no se ha comprometido a su ejecución).
- Un proceso sale del SO en dos pasos :
  - El proceso llega al punto normal de terminación de un proceso, cuando se abandona debido a un error irreuperable o cuando otro proceso hace que este abandone. La terminación pasa el proceso al estado de Terminado. (En este punto, el SO ya no eligira más este proceso para ejecutar aunque conserva las tablas asociadas al proceso durante un tiempo para la posible utilización de la información pro otros procesos)
  - Finalmente el SO ya no necesita mantener los datos relativos al proceso y estos se borran del sistema.
- Estos cinco estados descritos anteriormente dan lugar a una serie de combinaciones :
  - **Nulo Nuevo** : Se crea un nuevo proceso.
  - **Nuevo Listo** se supone que el sistema esta preparado para aceptar un proceso más (Hay memoria disponible) Cambio de estado **Admitir**.
  - **Listo Ejecución (Ejecución)**. – Cuando es hora de seleccionar un nuevo proceso para ejecutar, el Sistema operativo elige uno de los procesos en estos de Listo.
  - **Ejecución Terminación ( Liberar)**. (Liberar CPU).El proceso que se esta ejecutando es finalizado por el SO si este indica que ha terminado o si se abandona el proceso.
  - **Ejecución Listo (Fin de plazo o Tiempo excedido)**. – El proceso que esta en ejecución ha alcanzado el tiempo maximo permitido de ejecución ininterrumpida. Existen otros dos casos :
    - Por establecimiento de prioridades .– Entra en el sistema un proceso con prioridad mayor que el que se está ejecutando. (**Estado de Suspensión**).
    - El proceso cede voluntariamente el control del procesador.
  - **Ejecución bloqueado (Bloqueo)** Espera de asignación de recurso y cuando evento se produce pasa a la cola de espera de Listos.
  - **Bloqueado Listo** .– Un proceso en estado de bloqueo pasará al estado de Listo cuando se produzca el evento que estaba esperando.
  - **Bloqueado Terminado** .
- Debido a que un proceso puede ser suspendido (Debido a requerimientos del sistema O por prioridades ). (Ha sido expulsado de la memoria principal). Se generan nuevos estados :
  - Listo Proceso esta en memoria principal preparado para ejecutar.
  - Bloqueado Esta en meoria principal a la espera de un rproceso.
  - Bloqueado y suspendido Esta en memoria secundaria esperando un suceso.
  - Listo y Suspendido El proceso esta en memoria secundaria pero esta disponible para ejecutar tan pronto como se cargue en memoria principal.
- Va a observarse ahora el modelo de estados desarrollado :

- **Bloqueado Bloqueado y Suspendido.**– Si no hay procesos listos al menos un proceso bloqueado se suspende para dar cabida a otro que no este bloqueado.
- **Bloqueado y Suspendido Listo y Suspendido.**– Un proceso bloqueado y suspendido pasa al estado de Listo y suspendido cuando ocurre el suceso que estaba esperando.
- **Listo y Suspendido Listo .**– No hay procesos Listos en la memoria principal, el sistema Operativo tendrá que traer uno para continuar con la ejecución. (Puede darse el caso de que el proceso Listo y Suspendido tiene mayor prioridad que el proceso que esta en la cola de Listos ).
- **Listo Listo y Suspendido .**– el Sistema Operativo prefiere suspender un proceso bloqueado que uno Listo, ya que el proceso Listo podría ejecutarse de inmediato, mientras que el proceso bloqueado esta ocupando espacio en memoria principal. Puede ser necesario suspender un proceso Listo si esta es la única manera de liberar un bloque de memoria lo suficientemente grande.
- Otras transacciones :
- **Nuevo – Listo, Suspendido y Nuevo Listo .**– cuando se crea un nuevo proceso, se le puede añadir a la cola de Listos o a la cola de Listos y Suspendidos.
- **Bloqueado y Suspendido Bloqueado .**– Un proceso termina de ejecutar y deja un espacio en memoria principal. Hay un proceso Suspendido y Bloqueado que tiene una prioridad mayor que cualquiera de la cola de Listos y Suspendidos.
- **Ejecución Listo y Suspendido .**– Un proceso en ejecución pasa al estado de Listo cuando expira su cuanto de tiempo asignado. Sin embargo si se esta liberando un proceso de la cola de Bloqueados y suspendidos puede que el SO pase el proceso directamente a la cola de Listos Suspendidos liberando así una porción de memoria principal.
- **Varios Terminado .**– Terminación ( en algunos SO) por causas ajenas al proceso.
- Otros usos de la suspensión.–

Se define proceso suspendido como aquel que tiene las características siguientes :

- No está disponible de inmediato para su ejecución.
- El proceso puede estar esperando o no un suceso. Si lo está, la condición de Bloqueado es independiente de la condición de Suspendido y el acontecimiento del suceso bloqueando no lo habilita para la ejecución.
- El proceso fue situado en el estado de suspendido por un agente con el fin de impedir su ejecución.
- El proceso no puede apartarse de este estado hasta que el agente lo ordene explícitamente.
- Razones para la suspensión de un proceso :
- **Intercambio .**– Liberar memoria principal para otro proceso que está listo para ejecutarse.
- **Suspensión por parte del sistema operativo** por sospechar que el proceso es causante de problema
- **Solicitud del usuario.**– Un usuario puede querer suspender la ejecución de un programa con fines de depuración o en conexión con el uso de un recurso.
- **Por tiempo** (Procesos que se ejecutan a intervalos).– Puede ser suspendido mientras espera el siguiente intervalo de tiempo.
- **Solicitud del proceso padre** (Puede solicitar suspensión del hijo).– Un proceso padre puede suspender la ejecución de un hijo para examinar o modificar el proceso suspendido o para coordinar la actividad de varios descendientes.

### **Estructuras de Control del Sistema Operativo.**

El SO tiene que disponer de información sobre cada proceso o recurso.

Para ello construye unas tablas con información sobre cada entidad de las que administra. (Memoria, Dispositivos, ficheros, recursos y procesos).

Tablas de memoria.

De E/S

De archivos.

De Proceso.

- Las **Tablas de memoria** se encargan de la gestión de la memoria principal y virtual . Parte de la memoria principal esta reservada para el uso del SO y el resto esta disponible para el uso de los procesos. Los procesos se mantienen en memoria secundaria mediante alguna forma de memoria virtual o por un simple mecanismo de intercambio. Las tablas de memoria deben incluir la información siguiente :
- La asignación de memoria principal a los procesos.
- La asignación de memoria secundaria a los procesos.
- Cualesquiera atributos de protección de segmentos de memoria principal o virtual, tales como que procesos pueden acceder a que zonas de memoria compartidas.
- Cualquier información secundaria para la gestión de memoria virtual.
- Las **Tablas de E/S** son utilizadas por el sistema operativo para administrar los dispositivos y los canales de E/S del sistema informatico. En un momento dado, un dispositivo de E/S puede estar disponible o estar asignado a un proceso en particular. El SO necesita conocer el estado de la operación de E/S y la posición de memoria principal que se esta utilizando como origen o destino de la transferencia de E/S.
- Las **Tablas de Archivos** ofrecen información sobre la existencia de archivos, su posición en la memoria secundaria, su estado actual y otros atributos.
- Las **Tablas de proceso** que el SO debe mantener para administrarlos.
- **Datos de usuario.**– Parte modificable del espacio de usuario. Puede guardar datos del programa, una zona para una pila de usuario y programas que pueden modificarse.
- **Programas de usuario.**– El programa a ejecutar.
- **Pila del sistema.**– Cada procesos tiene una o más pilas asociadas a él. Una pila se utiliza para almacenar los parametros y las direcciones de retorno.
- **BCP.**– Información necesaria para que el SO controle el proceso.

### Cambio de contexto

Para cambiar la CPU de un proceso a otro se reuiere guardar el estado anterior y cargar el estado nuevo para el proceso (Cambio de contexto).

El tiempo de cambio de contexto es un gasto más y depende de la máquina.

Los tiempos de cambio de contexto dependen en gran medida del apoyo del hardware.

- En procesadores que ofrecen varios conjuntos de registros, un cambio de contexto implica cambiar el apuntador el conjunto actual de registros.
- Si hay más procesos activos que registros, el cambio de contexto se realiza utilizando la memoria.

- Cuanto más complejo es el SO, más trabajo se realiza en un cambio de contexto.

### **Bloque de control de Procesos (BCP)**

Bloque o registro de datos que contiene diversa información relacionada con un proceso concreto.

- Estado del proceso .– En el estado en que se encuentra el proceso (Nuevo, Listo, Espera ...)
- Contador de programa.– Contador indica la dirección de la siguiente instrucción a ejecutar.
- Reg. De la CPU .– Varían en número y tipo según la arquitectura del computador. Información que debe guardarse junto con el contador de programa cuando ocurre una interrupción.
- Información de planificación de la CPU .– Incluye una prioridad de proceso, apuntadores a colas de planificación y otros parámetros de planificación.
- Información de administración de memoria .– Incluye registros límites o tablas de páginas.
- Información contable .– Cantidad de tiempo real y de la CPU utilizado, límites de tiempo, números de cuenta, nº de proceso...
- Información del estado de E/S .– La información incluye solicitudes de E/S pendientes, dispositivos de E/S asignados, etc.

### **Hilo.**

Un **`Hilo'** es una unidad básica de utilización de la CPU y tiene poco estado compartido. Un grupo de hilos semejantes comparten código, espacio de direcciones y recursos del SO.

El entorno en el cual se ejecuta un hilo se llama **`Tarea'** .

Un proceso tradicional equivale a una tarea con un solo hilo. Una tarea no hace nada si no tiene hilos.

Un hilo posee un registro de estados y generalmente su propia pila.

- El coste de comunicación de la CPU entre hilos del mismo grupo y de la creación de hilos es el mismo que el de procesos pesados. De esta forma, una solución razonable para el problema de como un servidor puede manejar eficientemente solicitudes es bloquear un hilo y pasar a otro.
- Existen varias alternativas a la implementación de hilos :
- Los hilos pueden ser apoyados por el núcleo. Se proporcionan un conjunto de llamadas al sistema similares a las de los procesos.
- Pueden apoyarse por encima del núcleo, a través de bibliotecas de usuario.

### **Tipos de planificación.**

La planificación del microprocesador consiste en asignar los procesos de forma que se cumplan determinados objetivos. El tiempo de respuesta sea mínimo, productividad máxima (Rendimiento óptimo del sistema).

En el transcurso de su vida, un proceso transita por las diferentes colas del SO y es este quien debe seleccionar los procesos que entran en esas colas. (Tarea realizada por el planificador.)

- **Planificador a Largo Plazo** .– Selecciona procesos de un depósito y los carga en memoria para su ejecución. (Planificador de procesos.)
- **Planificador a Corto Plazo** .– selecciona uno de los procesos Listos para ejecución y le asigna la CPU. (Planificador de la Cola de Nuevos).

La principal diferencia entre los planificadores es su frecuencia de ejecución :

El planificador a corto plazo debe seleccionar con mucha frecuencia un nuevo proceso (Como mínimo una vez cada 10 miliseg).

El planificador a largo plazo se ejecuta con una frecuencia menor (Pueden transcurrir minutos entre la creación de procesos).

- Si el grado de multiprogramación es estable la tasa de creación debe ser igual a la tasa de salida de procesos del sistema.
- En algunos sistemas es posible que no exista planificador a largo plazo o que su función sea mínima.

Algunos sistemas como los de tiempo compartido deberían presentar un nivel intermedio de planificación o **Planificador a Medio Plazo** .– Puede ser a veces ventajoso eliminar un proceso de la memoria y de ese modo reducir el grado de multiprogramación. (Colas de suspensión)

A este esquema se le conoce como Swaping, y el planificador a medio plazo se encarga del intercambio de procesos, sacandolo de la memoria y volviendolo a introducir más tarde (Pueden ser necesarios para mejorar la mezcla de procesos o para liberar memoria

Existen tambien otros planificadores :

- **Planificador de E/S.**– Gestiona las diferentes colas de bloqueo. (Colas de Dispositivo).

### **Planificación de la CPU.**

- **Ciclo de rafagas de CPU y E/S** La ejecución de un proceso consiste en un ciclo de ejecución de CPU y espera de E/S. La ejecución de un procesos comienza con una rafaga de CPU, a esta el siguen rafagas de CPU y E/S. La última rafaga de CPU, terminará con solicitud al sistema de terminación ejecución.
- Un programa limitado por E/S tendrá muchas rafagas de CPU breves.
- Un programa limitado por la CPU tendra pocas rafagas de CPU muy largas.
- **Planificador de CPU** El proceso de selección de procesos se realiza por el planificador a corto plazo. La cola de listos, tendra diferentes implementaciones, conceptualmente se encuentran en fila esperando una oportunidad para ejecutarse.
- **Colas de planificación** Conforme los procesos entran en el sistema se colocan en la colas de procesos (Residen en almacenamiento secundario). Los procesos que residen en memoria prncipal se colocan en la cola de procesos listos. (Cada PCB tiene un apuntador que indica cul es el siguiente proceso en la cola de listos).

Existen otras colas en el sistema :

- La lista de procesos que esperan a un dispositivo de E/S se colocan en la Cola de dispositivo :
- Si se trata de un dispositivo dedicado nunca tendrá más de un procesos en la cola.
- Si por el contrario es un dispositivo compartido, en la cola podrá haber varios procesos.

**Diagrama de colas.**– Un proceso nuevo se coloca inicialmente en un cola de procesos Listos y allí espera hasta que es entregada a la CPU, una vez asignado pueden ocurrir unos de estos sucesos :

- Proceso puede emitir solicitud de E/S y colocarse en la cola de dispositivo.
- Procesos puede crear nuevo procesos y esperar a que este termine.

- Ser extraído de la CPU por la fuerza.

### Algoritmos de Planificación

- La planificación se hace teniendo en cuenta determinados criterios :
- **Tiempo de respuesta.**– En un sistema interactivo puede que el mejor criterio no sea el tiempo de retorno , por eso se utiliza esta otra medida : Tiempo que transcurre desde que se emite solicitud hasta que se recibe la primera respuesta respuesta.
- **Tiempo de retorno.**– Desde el punto de vista de un unico proceso el criterio más improtante es el tiempo que tarda en ejecutarse. El intervalo de tiempo entrada de un proceso en el sistema hasta su finalización.
- **Plazos.**– Si se especifica un plazo de terminación de un proceso la planificación debe planificar otras metas a la maximización de plazos cumplidos.
- **Previsibilidad.**– Una tarea que se ejecute reiteradamente se deberá ejecutar en el mismo tiempo y con el mismo coste.
- **Productividad.**– N° de procesos finalizados por unidad de tiempo.
- **Utilización de procesador.**– Porcentaje de tiempo que procesador esta ocupado.
- **Equidad.**– Todos los procesos deben ser tratados de igual forma impidiendo inanición de un proceso.
- **Prioridades.**– Se debe favorecer a los procesos con mayor prioridad.
- **Recursos equilibrados.**– Se deben mantener todos los recursos del sistema ocupados.
- **Tiempo de espera.**– Total de tiempo que un proceso espera por su ejecución completa. Se obtiene sumando los tiempos de espera de todas las ráfagas con el tiempo de espera de la primera ráfaga.

### Otras políticas de planificación.

Hay sistemas cuya política de planificación se basa en prioridades :

- **Estáticas.**– Se mantiene en ejecución.
- **Dinámicas.**– Varían en ejecución.

El resto de políticas apropiativas / No apropiativas.–

- **No apropiativas** Una vez que el proceso obtiene CPU no le puede ser requisada hasta que acaba la ráfaga.
- **Apropiativas** Permiten que un proceso con mayor prioridad requiese de CPU al proceso que la tenia asignada sin finalizar la ejecución.

### Diferentes algoritmos de planificación.

- FCFS
- RR
- SJF / SPN
- SJF apropiativo o SRT
- Colas multinivel
- Colas multinivel realimentadas.
- HRRN
- **FCFS** .– El proceso que primero solicita la CPU es el primero al que se le asigna. Esta política se implementa facilmente utilizando un cola FIFO.

Cuando un proceso entram el PCB apunta al último proceso en la cola FIFO. Cuando la CPU esta libre se

asigna al proceso que esta situado el primero en la cola. En la politica FCFSm el tiempo promedio de espera es bastante largo.

- El algoritmo FCFS es no apropiativo, una vez que se ha asignado la CPU a un proceso, este la conserva hasta que desee liberarla ya sea por terminación o por E/S
- Especialmente problemática.

PROCESOS	I LLEGADA	T SERVICIO	I COMIENZO	FINAL	T RET	Tq/Ts
A	0	1	0	1	1	1
B	1	100	1	101	100	1
C	2	1	101	102	100	100
D	3	100	102	202	199	1.99

$$T_m = 201 + 199 / 4 = 100$$

- **SJF** (Tiempo restante más corto) Este algoritmo asocia a cada proceso la longitud de su siguiente rafaga de CPU. Cuando la CPU esta disponible es asignada al proceso que tiene la siguiente rafaga de CPU menor. Si dos procesos tiene la misma longitud para la siguiente rafaga de CPU se utiliza la planificación FCFS para romper el empate.

Puede comprobarse que el algoritmo SJF es óptimo, ya que ofrece el menor tiempo promedio para un conjunto de procesos dados.

El problema principal del algoritmo es el conocimiento de la longitud de la siguiente rafaga de CPU por tanto no puede implantarse a nivel de la planificación de la CPU, aunque se utilizan aproximaciones.

Aunque no se conocen los valores de las siguientes rafagas de la CPU podemos predecir su valor esperando que sea más o menos del mismo tamaño que las anteriores ( Por ello se elige el proceso con rafaga previa de CPU más breve)

El algoritmo SJF puede ser apropiativo(SRT) o no apropiativo. La alternativa se plantea cuando cuando un nuevo proceso llega a la cola de procesos listos mientras se esta ejecutando otro proceso. El nuevo proceso puede tener una rafaga de CPU menor que lo que resta del proceso que se ejecuta en ese momento.

Un algoritmo SJF apropiativo desplazará al proceso que se ejecuta.

La politica SJF apropiativa es más comunmente conocida como SRT.

PROCESOS	I LLEGADA	T SERVICIO	I COMIENZO	FINAL	T RET	Tq/Ts
A	0	1		1	1	
B	1	100		102	101	
C	2	1		3	1	
D	3	100		202	199	

$$T_m = 302 / 4 = 75 \text{ aprox.}$$

TAREAS	T ENTRADA	CPU	PRIORIDAD
P1	0	5	1
P2	0	2	0

<b>P3</b>	1	4	2
<b>P4</b>	2	3	1
<b>P5</b>	3	10	2
<b>P6</b>	20	4	1
<b>P7</b>	25	20	2
<b>P8</b>	30	2	0
<b>P9</b>	31	6	2
<b>P10</b>	60	1	0

**SJF**

**SRT**

Tr (SJF) = (2+14+8+3+31+8+23+20+25+1) / 10 = 12.5 ms.

Tr /SRT) = (2+14+8+3+21+8+31+2+7+1) / 10 = 9.7 ms.

**ROUND ROBIN (Algoritmo de planificación circular) .-** Diseñado especialmente para sistemas en tiempo compartido. Se define una pequeña cantidad de tiempo o **cuanto de tiempo**, que generalmente varía entre los 10 y 100 milisegundos. La cola de procesos Listos se trata como una cola circular, el planificador de CPU la recorre asignando a cada proceso un cuanto de tiempo.

Se mantiene la cola de Listos en una cola FIFO. Los nuevos procesos se agregan al final de la cola. El planificador toma el primer proceso de la cola de Listos y programa el cronometro para que provoque una interrupción y despacha el proceso.

Pueden suceder una de estas dos cosas :

- El proceso tenga una rafaga de CPU menor que el cuanto , el proceso se libera voluntariamente.
- Si la rafaga es más grande que el cuanto el cronometro se activará y provocará una interrupción para el sistema operativo. Se ejecuta cambio de contexto y el proceso se colocará al final de la cola de Listos.

El rendimiento del RR depende en gran medida del cuanto de tiempo. Si el cuanto es muy grande la politica es la misma que la del FCFS. Si es muy pequeño el enfoque se llama Compartir el procesador, y para los usuarios parece que cada proceso tiene su propio procesador.

**RR (Cuanto = 4)**

Rendimiento = 60% (Trabajos que acaban en su primera rafaga).

Rendimiento optimo 70% , se tiene que subir el Cuanto a 5

Rendimiento = 80 % . Calcular tiempo de ret medio.

**Colas con prioridad**

Se han creado otros algoritmos de planificación en aquellos sistemas donde los procesos pueden clasificarse facilmente en diferentes grupos.

Un algoritmo de Planificación de colas de multiples niveles divide la cola de Listos en varios niveles o colas. Los procesos se asignan de forma permanente a un cola, generalmente dependiendo de una prioridad. Cada cola tiene su propio algoritmo de planificación.

Debe existir además una planificación entre colas la cual es generalmente una planificación apropiativa de prioridad fija.

- Se gestiona la cola de más prioridad hasta que esta vacía y se pasa a la siguiente.

Q=5

Q=10

Q=15

Tiempo de retorno medio.– Se calcula separadamente para cada cola y luego se hace la media aritmética.

Nivel 0 =  $2+10+ /3 =4.3$

Nivel 1 =  $24+8+3+8 / 4 = 10.7$

Nivel 2 =  $19+31+15 /3 =21.6$

Tiempo medio de ret = 12.2 ms.

### **Colas multinivel.**

La planificación de colas multinivel (con realimentación ) permiten al proceso moverse de una cola a otra. La idea es separar a los procesos con diferentes características en cuanto a rafagas de CPU. Si n procesos utiliza demasiado tiempo la CPU pasará a una cola de menor prioridad.

Este esquema deja a los procesos interactivos y a los limitados por E/S en las colas de mayor prioridad.

Por lo general un planificador de colas multinivel se define por los siguientes parametros :

- Número de colas.
- Algoritmo de planificación para cada cola.
- El método utilizado para promover proceso a colas de mayor prioridad.
- El método utilizado para degradar proceso a colas de menor prioridad.
- Método utilizado para determinar a cual cola entrará un proceso cuando necesite un servicio.

Q=5

Q=10

Q=15

\* Algoritmo entre colas SRT