

APUNTES SOBRE PROCESOS Y DEADLOCK

Alejandro Bia

CONCEPTO DE PROCESO¹

PROCESO

- En 1960 (proyecto "Multics") surge concepto de proceso.

Definiciones breves:

- Programa en ejecución.
- Entidad lógica a la que se le asignan procesadores.

ESTADOS DE UN PROCESO²

1. ACTIVO: Esta en marcha. Tiene asignado un procesador que esta usando.
2. LISTO: No tiene procesador, pero podría usarlo si lo tuviera.
3. BLOQUEADO: Proceso en espera de un acontecimiento (generalmente. E/S)

TRANSICIÓN ENTRE ESTADOS:

- * ACT->LISTO: Si su quantum (tiempo asignado) expira antes de que libere el procesador voluntariamente. Tiempo excedido: una interrupción de Hardware indica que su tiempo expiro. [T.Excedido]
- * ACT->BLOQ: Si inicia una operación de E/S. (U otro evento por el que deba esperar). Es la única transición iniciada por el propio proceso y no por entidades externas como en los demás casos. [Bloqueo]
- * BLOQ->LISTO: Cuando finaliza operación de E/S. (U otro evento). El proceso vuelve a estar listo para ejecutar tan pronto se le asigne un procesador. [Despertar]
- * LISTO->ACT: Cuando el despachador (Dispatcher) le asigna un procesador. Esto se hace según estricto orden de prioridad de ejecución. Hay para ello una lista de procesos "listos para ejecutar". [Despacho]

PCB - Proceso Control Block (Bloque de Control de Proceso):

Es una estructura de datos que contiene la siguiente información sobre un proceso:

- ESTADO actual
- IDENTIFICACIÓN
- PRIORIDAD

¹ Ver DEITEL - Introducción a los Sistemas Operativos - cap.3

² ver diagrama DEITEL p.58

- apuntadores para localizar la MEMORIA que usa - apuntadores para asignar RECURSOS
- area para preservar los valores de los REGISTROS
- etc.

Contiene toda la información necesaria para controlar el proceso, y en especial para sacarlo de ejecución y poder volverlo a la ejecución sin problemas.

Los S.O. pueden realizar varias operaciones sobre los procesos:

- Crear (carga de programa e inicio de ejecución)
- Destruir (fin de la ejecución)
- Suspender
- Reanudar
- Cambiar prioridad
- Bloquear (pasaje de ACT a BLOQ)
- Despertar (pasaje de BLOQ a LISTO)
- Despachar (pasaje de LISTO a ACT)

CREACION DE PROCESOS POR PROCESOS³:

Un proceso puede crear a otro proceso: Proceso PADRE -> Proceso HIJO

De esto resulta una estructura jerárquica de procesos.

SUSPENSION y REANUDACION⁴:

Se usa la suspensión de procesos para:

- Equilibrar la carga del sistema (ante sobrecargas)
- En lugar de abortar un proceso que puede estar funcionando mal. (debido a causas propias o del sistema)

Un proceso suspendido no puede proseguir hasta que otro lo reanude.

INTERRUPCIONES⁵:

Las interrupciones son importantes en un sistema con concurrencia de procesos sobre dispositivos.

Cuando se produce una interrupción, el S.O. le presta atención en forma inmediata. Existen rutinas llamadas "manejadoras de interrupciones". Cuando ocurre una interrupción:

1. El S.O. obtiene el control.
2. El S.O. salva el estado del proceso interrumpido.
3. El S.O. analiza la interrupción y pasa el control a la rutina de manejo de interrupción apropiada. "Cambio de contexto".
4. Finalizada la ejecución de dicha rutina, reestablece el proceso interrumpido.

EI NÚCLEO o KERNEL de un S.O.:

- Es pequeño
- Contiene código altamente utilizado
- Reside en la memoria principal

³ ver diagrama DEITEL p.61

⁴ ver diagrama DEITEL p.63

⁵ ver diagrama DEITEL p.64

- Se encuentra en un nivel superior, pero muy próximo al Hardware; es el Software de mas bajo nivel.
- Funciones:
 - o Gestión de memoria principal. (Asig/Desasig)
 - o Manejo de interrupciones
 - o Gestión de procesos (Creación/Destrucción, Cambio de estados)
 - Manejo de los PCB
 - Despacho de procesos
 - Sincronización entre procesos
 - Comunicación entre procesos
 - Soporte de mecanismo de Llamada/Retorno a procedimientos.
 - o Soporte de actividades de E/S
 - o Soporte del sistema de archivos
 - o Ciertas funciones contables/estadísticas del sistema.

Existe una tendencia hacia migrar partes del KERNEL a microcódigo para mayor seguridad y velocidad de ejecución: (Ej. ROM-BIOS del PC)

SISTEMAS OPERATIVOS DESDE EL PUNTO DE VISTA DE LA MULTITAREA.

CONCEPTOS PREVIOS.

Programa - archivo ejecutable.

Proceso - es una instancia de ejecución de un programa. (cada vez que un programa se corre se genera un proceso)

Muchos procesos pueden ejecutarse a la vez, entonces hablamos de multitarea o multiprogramación.

Dos procesos son concurrentes si existen al mismo tiempo.

Dos procesos concurrentes pueden ejecutarse independientemente uno del otro o pueden interactuar.

Exclusión mutua - Se refiere a que cuando varios procesos acceden a la vez a datos o recursos compartidos, solo accede uno de ellos, mientras que los otros esperan.

Deadlock - Muerte por bloqueo, espera indefinida por un recurso compartido que no es liberado nunca.

SISTEMA OPERATIVO (mas definiciones)

- Conjunto de programas.
- Conjunto de procesos concurrentes que:
 - o contienen tanto datos como procedimientos necesarios para realizar la asignación de recursos compartidos.
 - o permiten la comunicación entre si (asincrónicos)
 - o son controlados por un núcleo (o Kernell).
- Manejador de recursos que resuelve la exclusión mutua, la concurrencia y el deadlock.
- Presenta al usuario una Maq.Virtual mas amigable que la Real.

Estudio de las funciones de un sistema operativo: (por áreas)

- MEMORIA:

- Asignación/desasignación de memoria.
- Memoria virtual:
 - o Implementa mecanismos Swapping permitiendo que se ejecuten muchos más programas de los que realmente caben en la misma, donde el Kernell copia los programas desde memoria a disco.
 - o Swapping paginado bajo demanda. Copia parte del programa a disco no reside totalmente en memoria.
- Carga/descarga de programas
- Discos virtuales en memoria
- Cache para lectura/grabación de disco
- Spooler de impresión

- ENTRADA/SALIDA:

- Soporte de actividades de E/S.
- Redireccionamiento
- Filtros
- Dispositivos estandar (PRN, COM, NUL, CON)
- Comunicaciones:
 - o procesamiento remoto
 - o transmisión de datos
 - o servidor de datos
- Discos virtuales
- Cache de disco (buffering)
- Spooler

- ARCHIVOS:

- Soporte de sistema de archivos.
- Nombres simbólicos y estructura de directorios
- Atributos de archivo
- Funciones de manejo de archivos de alto nivel (Open, Close, Read, Write, Append, ...)

- TAREAS y RECURSOS:

- Asignación de recursos a procesos
- Planificación de tareas (scheduling) (prioridades)
- Interpretación de lenguaje de control de tareas (JCL - Procesos batch)
- Gestión de interrupciones.
- Creación y destrucción de procesos.
- Cambio de estado de los procesos.
- Suspensión y reanudación de procesos.

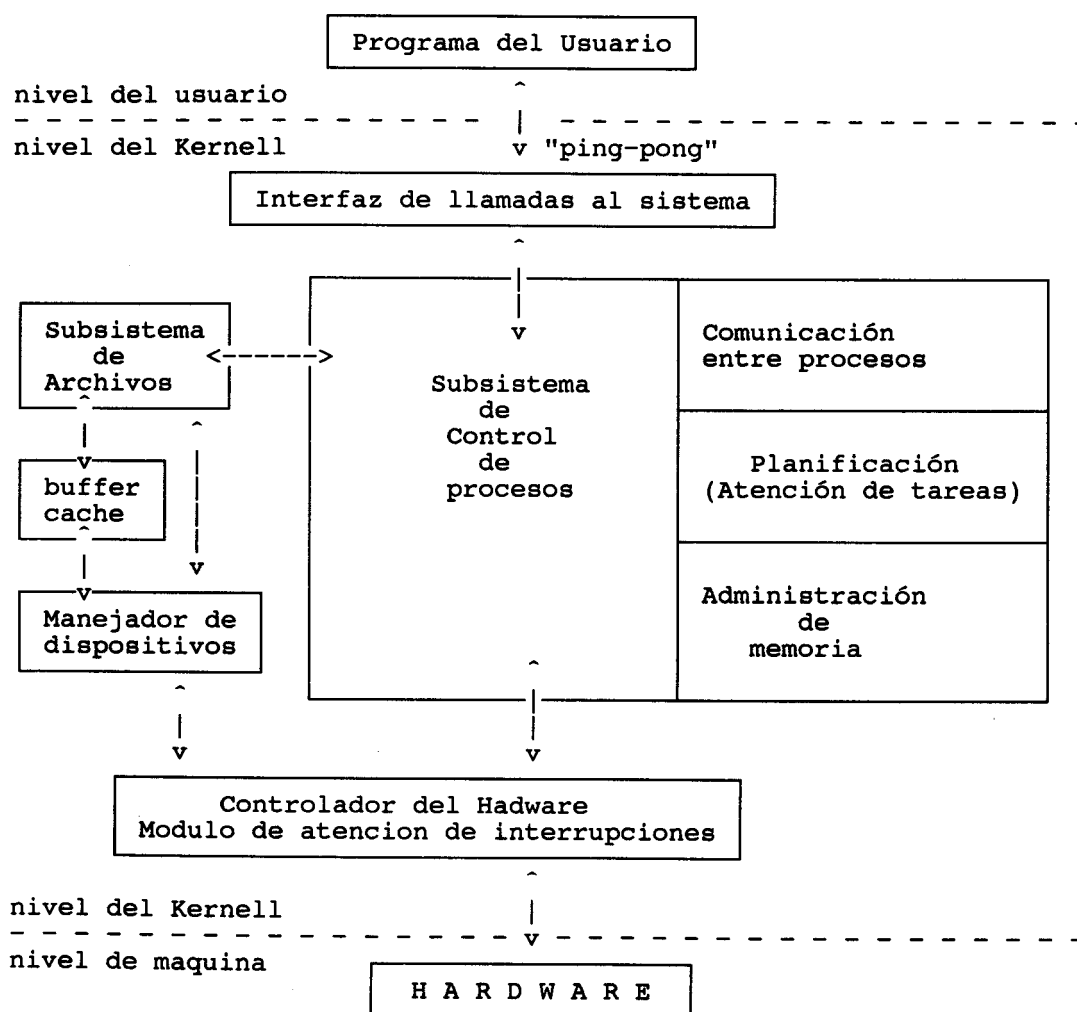
- Función de despacho, operación que alterna distintas tareas por TIME-SLICE (rebanada de tiempo).
- Comunicación entre procesos.
- SEGURIDAD:
 - Protección de archivos y seguridad del sistema
 - Confiabilidad
 - Manejo y recuperación de errores
- OPERADOR y USUARIOS:
 - Fácil manejo (entorno amigable)
 - Permitir control sobre distribución de recursos
 - Gestión de terminales y transacciones (procesos interactivos)

Para la ejecución de procesos tenemos 2 niveles:

- A) modo usuario
- B) modo kernell (Cuando toma el control el S.O. a través de un system call desde el programa del usuario o a través de interrupciones externas por ej.)

Es como un ping-pong entre el programa y el S.O., el programa le pasa el control al S.O. cada vez que necesita usar una de sus funciones, o el sistema le quita el control cuando se produce una interrupción externa, luego, al finalizar le devuelve en control, y así sucesivamente.

Estructura de un Sistema Operativo Ej. Kernell del UNIX



El subsistema de archivos

- Ubica espacio para los archivos.
- Administra el espacio libre
- Controla el acceso a los archivos.
- Recupera datos para los usuarios.

Los procesos interactúan con el subsistema de archivos a través de un conjunto de System calls (LECTURA, GRABACIÓN, ABRO, CIERRO)

Accede a los datos usando un mecanismo de buffering (Buffer Cache) que regula el flujo de datos entre el Kernell y los dispositivos. El mecanismo de buffering interactúa con los manejadores de dispositivos para iniciar la transferencia hacia y desde el kernell.

El subsistema de control de procesos

Es responsable de la comunicación entre procesos, administración de memoria y administración de procesos. El módulo de administración de memoria controla el uso de la memoria, si en determinado momento no hay espacio suficiente para todos los procesos, el Kernell los mueve entre memoria y

disco, esto es realizado usando 2 políticas fundamentales de administración de memoria, el swapping que es cuando son movidos todos los procesos y la otra es el llamado paginado a demanda cuando solo es movido una parte. El planificador (scheduler) hace la función de despacho reasignando la CPU entre los procesos.

DEADLOCK (muerte por bloqueo)

El problema ocurre al alternar varias tareas compartiendo recursos en forma simultánea, pudiendo anularse entre si, lo cual es algo que depende del azar.

Aplazamiento indefinido: Se espera en forma indefinida por la liberación de un recurso.

Ejemplo de DEADLOCK:

Sean dos procesos: P1 y P2, y dos recursos: C y D

Lo programado:

```

P1      _____      P2      _____
      \  -----
      /  -----
      /  -----
      /  Solicita(D);
      /  Solicita(C);
      /  -----
      /  -----
      /  Libera(D);
      /  Libera(C);

P2      -----
P2      -----
P2      Solicita(C);
P2      -----
P2      Solicita(D);
P2      -----
P2      Libera(C);
P2      Libera(D);
```

Lo que sucede:

```

P1      -----
P1      -----
P1      Solicita(D);
P1      Solicita(C);
P1      Espera por (C)

P2      -----
P2      Solicita(C);
P2      -----
P2      Solicita(D);
P2      Espera por (D)

-----
D E A D L O C K - Espera indefinida
```

En un sistema de multiprogramación varios procesos, pueden competir por una cantidad de recursos. Un proceso requiere recursos y si los recursos no están disponibles, el proceso entra en estado de espera. Puede suceder que el proceso que espera nunca cambie su estado debido a que los recursos que requiere, están asignados a otros procesos, que a su vez están en estado de espera. También surge el problema del aplazamiento indefinido en el que el proceso, aunque no este en deadlock puede esperar largo tiempo, por un evento debido a la predisposición de los controles de planificación de recursos del sistema. Hay 4 condiciones que tienen que estar presentes para que se produzca el deadlock:

- 1) Mutua exclusión
- 2) Retención y espera
- 3) No remoción
- 4) Espera circular

1) Mutua exclusión : Se refiere a procesos que reclaman control exclusivo de los recursos que solicitan. Al menos un recurso esta asignado, si otro proceso requiere este recurso deberá esperar hasta que sea liberado.

2) Retención y espera: Existe un proceso que tiene asignado recursos y que esta esperando por otros recursos adicionales que están siendo usados por otros procesos.

3) No remoción: Recursos que no pueden ser removidos de los procesos que los poseen hasta que sean usados completamente, es decir un recurso puede ser solamente liberado voluntariamente por el proceso que lo tiene asignado, y no existe opción de quitárselo al sistema mientras este siendo usado.

4) Espera circular: Se refiere a que existe un conjunto de procesos (PO, P1, P2, ... PN) tal que PO está esperando por un recurso asignado a P1, ..., PN-1 está esperando por un recurso asignado a PN y PN esta esperando por un recurso asignado a PO. Tiene asignado un recurso y esta esperando por un recurso.

Areas para la investigación del DEADLOCK

- (1) Prevención
- (2) Evasión
- (3) Detección
- (4) Recuperación

1. Prevención:

Significa imponer reglas o restricciones que me aseguren que no se va a producir el DEADLOCK.

2. Evasión:

Es darse cuenta que una serie de requerimientos pueden provocar DEADLOCK y en consiguiente no permitir que esa serie de requerimientos se produzca; por ejemplo, no pasar a estados de espera con recursos asignados.

Métodos para manejar el DEADLOCK: Podemos usar algún protocolo para asegurar que el sistema nunca entre en deadlock. Podemos dejar que el sistema entre en deadlock y luego tratar de recuperarlo.

Estado seguro: Es aquel que satisface los requerimientos de recursos a cada proceso (hasta su máximo) impidiendo el deadlock, es decir es tal que a todos los procesos les será posible finalizar y por consiguiente no habrá deadlock. El estado inseguro es lo contrario.

4. Recuperación

Las formas de recuperar pueden ser:

- 1) Cancelo la ejecución del sistema y recomienzo.
- 2) Individualizar los procesos que causan DEAD-LOCK y cancelarlos.
- 3) Individualizar los procesos que causan DEAD-LOCK, desasignar los recursos y posponer su ejecución.

La solución más común es cancelar la ejecución de uno o más procesos, otra solución es la de remover algunos recursos de uno o más procesos en DEAD LOCK. Si se usa el criterio de remoción entonces hay que seleccionar la víctima, o sea a que proceso le saco y que recursos saco.

Los elementos a tener en cuenta son:

- Prioridad de procesos.
- Tiempo que ha ejecutado cada proceso.
- Cantidad de recursos asignados.
- Cantidad de recursos adicionales que el proceso necesita para finalizar.

