

Realizar un microcódigo en un sistema operativo puede mejorar el desempeño de los sistemas de desarrollo de programas y mejorar la seguridad de los sistemas dinámicos y frecuentemente.

Microprogramas.- Se ejecutan en una memoria de control de alta velocidad y están formados por microinstrucciones individuales de naturaleza más elemental y de función menos densa que las instrucciones convencionales de lenguaje máquina.

Microcódigo.- Una importante decisión de diseño es cuales funciones deben realizarse en microcódigo. El microcódigo representa una oportunidad real para mejorar el rendimiento de la ejecución de un sistema. Mediante la codificación cuidadosa de secuencias de instrucciones de ejecución frecuente en firmware, en lugar de software, los diseñadores han logrado extraordinarias mejoras en el rendimiento.

Microdiagnóstico.- Los microprogramas tienen acceso a más hardware que los programas en lenguaje de máquina, por lo que es posible efectuar una detección y una corrección de errores más amplia y con mayor grado de precisión. Algunos sistemas alternan el microdiagnóstico con las instrucciones en lenguaje de máquina. Esto permite evitar errores potenciales y lograr una operación más confiable, por lo que la microprogramación puede ser efectiva en el diseño de sistemas tolerantes a fallas.

Emulación.- Es una técnica para lograr que un computador se comporte como si fuera otro. El conjunto de instrucciones en lenguaje de máquina del computador que se va a emular se microprograma en la máquina anfitriona. Así, los programas en lenguaje de máquina de la máquina emulada pueden ejecutarse directamente en la anfitriona. Los fabricantes de computadores utilizan muy a menudo la emulación cuando presentan nuevos sistemas. Los usuarios acostumbrados a los computadores antiguos pueden ejecutar directamente sus antiguos programas en los sistemas nuevos sin alteración alguna, esto requiere el proceso de conversión.

Algunas funciones del Sistema Operativo que suelen realizarse en Microcódigo son:

- Manejo de interrupciones
- Mantenimiento de varias estructuras de datos
- primitivas de sincronización que controlan el acceso a datos compartidos y a otros recursos
- "Commutación contextual"; esto es, la commutación rápida de un procesador entre varios usuarios en un sistema multiusuario.
- Secuencias de llamada a procedimientos y retorno.

Bloque de control del proceso (PCB) **Unidad II** **CONCEPTOS DE PROCESOS**

Objetivo de esta unidad.- Durante el desarrollo de esta unidad, se analizarán las principales técnicas para la administración del CPU y de los procesos en su secuencia de ejecución.

El alumno deberá comprender qué es un proceso, cuáles son las transiciones de estado de un proceso, cómo se logra la comunicación entre estos conjuntamente con los problemas que enfrenta el Sistema Operativo para lograr la comunicación; cómo decide el sistema operativo que procesos debe efectuar cuando dos o más procesos son ejecutables lógicamente.

Definiciones de "proceso"

El término "proceso" ha sido utilizado a veces como sinónimo de tarea, ha tenido muchas definiciones:

- Es una actividad asíncrona.
- Un programa en ejecución.
- El "espíritu animado" de un procedimiento.
- El "centro de control" de un procedimiento en ejecución.
- Lo que se manifiesta por la existencia de un "bloque de control de proceso" (PCB) en el sistema operativo.
- La entidad a la que se asignan los procesadores
- La unidad despachable.

Un programa es una entidad inanimada, solo cuando un procesador le infunde vida, se convierte en la entidad activa que se denomina proceso.

Estados de un Proceso

Se dice que un proceso está en estado de ejecución si tiene asignado el CPU. Se dice que un proceso está en estado listo si pudiera utilizar un CPU en caso de haber uno disponible. Un proceso está en estado bloqueado si está esperando algún evento antes de proseguir su ejecución.

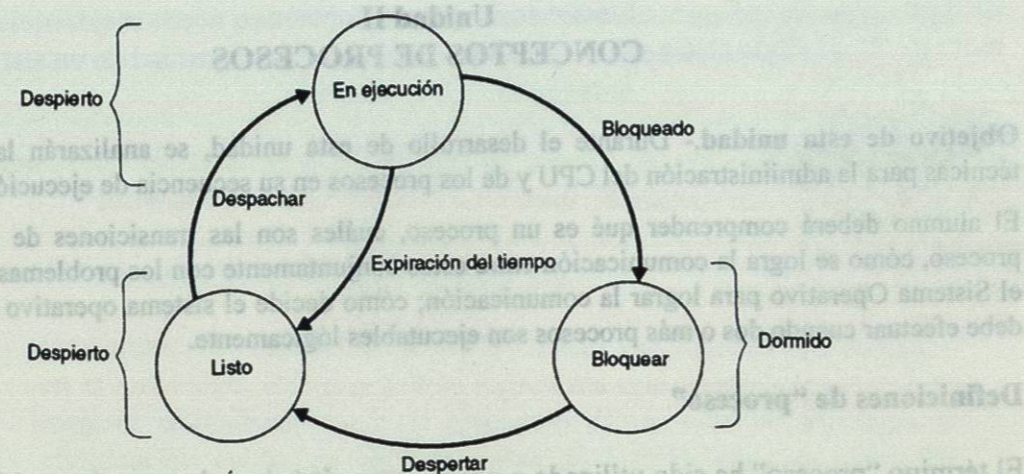
Transiciones de Estado de los Procesos

Cuando se admite una tarea en el sistema, se crea el proceso correspondiente y se inserta normalmente al final de la lista de procesos listos. El proceso se desplaza poco a poco hacia el frente de la lista de procesos listos, a medida que los procesos que se encuentran antes que él completan su turno de uso de el CPU.

Cuando el proceso llega al principio de la lista, se le asigna el CPU cuando este queda disponible y entonces se dice que hay una *transición de estado* del estado listo al estado de ejecución. A la asignación del procesador al primer proceso de la lista de procesos listos se le llama despacho.

1. Eliminarlo de la lista de procesos, permanentemente del sistema.
2. Los recursos que estaba utilizando se devuelven al sistema.
3. El PCB se borra.

La destrucción de un proceso es más difícil cuando este ha creado a otros procesos.



Despachar(x_proceso):Listo → Ejecución

Para evitar que un proceso monopolice el sistema, en forma accidental o intencional, el sistema operativo utiliza un reloj de interrupción por hardware (también llamado cronómetro de intervalos) para que las tareas de ese usuario se ejecuten durante un intervalo específico de tiempo o cuanto (quantum). Si el proceso no libera voluntariamente el CPU antes de que expire el cuanto, el reloj genera una interrupción, haciendo que el sistema operativo retome el control. El sistema operativo envía al proceso que se estaba ejecutando a la lista de procesos listo y procede a ejecutar el primero de la lista de procesos listos.

Tiempo Expirado(x_proceso):Ejecución → Listo

Si el proceso que se está ejecutando inicia una operación de entrada/salida antes de que expire su cuanto, libera voluntariamente al CPU, es decir el proceso se bloquea a sí mismo esperando que se complete la operación de entrada/salida.

Bloquear(x_proceso):Ejecución → Bloqueado

Cuando se completa una operación de entrada/salida o algún otro evento que espere un proceso. El proceso realiza la transición del estado bloqueado al estado listo.

Despertar(x_proceso):Bloqueado → Listo

Como se observará la única transición de estado iniciada por el proceso de usuario es el bloqueo; las otras tres transiciones son iniciadas por entidades externas al proceso.

Bloque de control del proceso (PCB)

La forma en que se manifiesta un proceso en un sistema operativo es mediante el PCB (Bloque de Control de Proceso). EL PCB es una estructura de datos que contiene información importante acerca de un proceso incluyendo:

- El estado actual del proceso.
- Un identificador único del proceso.
- Un apuntador hacia el padre del proceso (hacia el proceso que lo creó).
- Apuntadores a los hijos del proceso en el caso de tener procesos creados por él.
- La prioridad del proceso.
- Apuntadores hacia las zonas de memoria del proceso.
- Apuntadores a los recursos asignados del proceso.
- Un área para salvaguarda de los registros.
- El procesador en que se está ejecutando el proceso (en el caso de múltiples procesadores).

El PCB es un almacén central de información que permite al sistema operativo localizar toda la información importante acerca de un proceso. PCB es la entidad que define un proceso al sistema operativo.

Cuando el sistema operativo conmuta el CPU entre varios procesos activos utiliza las áreas de salvaguarda de los PCB para guardar la información que necesita para reiniciar un proceso cuando obtenga el CPU.

Operaciones sobre procesos

Los sistemas que administran procesos deben ser capaces de realizar ciertas operaciones sobre procesos:

- Crear un proceso que implica:
 1. Darle un nombre al proceso.
 2. Dar de alta en la lista de procesos conocidos del sistema o de la tabla de procesos (insertar)
 3. Determinar la prioridad inicial del proceso.
 4. Crear el PCB de ese proceso.
 5. Asignar recursos al proceso.

Un proceso puede crear un nuevo proceso y al proceso creador se le va a llamar proceso padre y al proceso creado se le va a llamar proceso hijo, lo cual nos va a originar la estructura jerárquica de procesos.

- Destruir un proceso que implica:
 1. Eliminarlo de la lista de procesos, no solamente del sistema.
 2. Los recursos que estaba utilizando se devuelven al sistema.
 3. El PCB se borra.

La destrucción de un proceso es más difícil cuando este ha creado a otros procesos.

- Suspender un proceso

Un proceso suspendido no puede proseguir hasta que lo reanuda otro proceso. La suspensión es una operación importante y ha sido puesta en práctica en diferentes formas en diversos sistemas. La suspensión dura por lo regular breves momentos o breves periodos de tiempo. El sistema efectúa suspensiones para eliminar temporalmente ciertos procesos y así reducir la carga del sistema durante situaciones de carga máxima. Cuando hay suspensiones largas se deben liberar los recursos del proceso.

- Reanudar un proceso
- Reanudar o activar un proceso implica reiniciarlo a partir del punto en el que se suspendió.

- Bloquear un proceso
- Despertar un proceso
- Despachar un proceso
- Permitir que un proceso se comunique con otros (comunicación entre procesos)
- Cambiarle la prioridad a un proceso.

Razones por las que se puede suspender un proceso:

- Si un sistema esta funcionando mal y es probable que falle se puede suspender los procesos activos para reanudarlos cuando se haya corregido el problema.
- Un usuario que desconfie de los resultados parciales de un proceso puede suspenderlo en lugar de abortarlo hasta que verifique si el proceso funciona correctamente o no.
- Algunos procesos se pueden suspender como respuesta a las fluctuaciones a corto plazo de la carga del sistema y reanudarse cuando las cargas regresen a niveles normales.

Estados de Suspensión

Se ha añadido dos nuevos estados denominados: suspendido_listo y suspendido_bloqueado, por lo que ahora contaremos con estados activos que vimos con anterioridad, y estados de suspensión.

Tenemos que, una suspensión puede ser iniciada por el propio proceso o por otro. En un sistema con un solo procesador, el proceso en ejecución puede suspenderse a sí mismo, ningún otro proceso podría estar en ejecución para realizar la suspensión.

Solamente otro proceso puede suspender un proceso listo. La transición sería:

Suspender(x_proceso):Listo → Suspendido_listo

Un proceso puede hacer que otro proceso que se encuentre en estado suspendido_listo pase al estado listo. La transición sería:

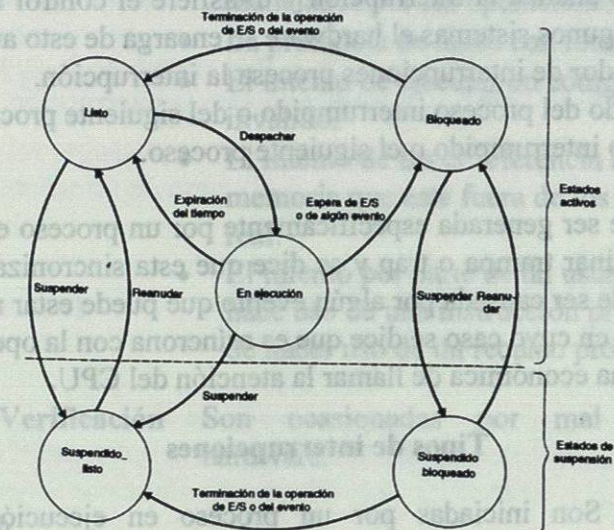
Reanudar(x_proceso):Suspendido_Listo → Listo

Un proceso puede suspender a otro que este bloqueado. La transición sería:

Suspender(x_proceso):Bloqueado → Suspendido_Bloqueado

Un proceso puede reanudar otro proceso que este suspendido_bloqueado. La transición sería:

Reanudar(x_proceso):Suspendido_Bloqueado → Bloqueado



Transacciones

Suspender(x_proceso):Listo → Suspendido_Listo

Despachar(x_proceso):Listo → Ejecución

Suspender(x_proceso):Ejecución → Suspendido_Listo

Reanudar(x_proceso):Suspendido_Listo → Listo

Tiempo excedido(x_proceso):Ejecución → Listo

Suspender(x_proceso):Bloqueado → Suspendido_bloqueado

Completar(x_proceso):Bloqueado → Listo

Reanudar(x_proceso):Suspendido_Bloqueado → Bloqueado

Completar(x_proceso):Suspendido_Bloqueado → Suspendido_Listo

Esperar(x_proceso):Ejecución → Bloqueado

Procesamiento de interrupciones

Interrupción.- Es un evento que altera la secuencia en que el procesador ejecuta las instrucciones. La interrupción es generada por el hardware del sistema.

Cuando se genera una interrupción:

1. El sistema operativo toma el control, es decir, el hardware pasa el control al sistema operativo.
2. El sistema operativo guarda el estado del proceso interrumpido, en muchos sistemas esta información se guarda en el PCB del proceso interrumpido.
3. El sistema operativo analiza la interrupción y transfiere el control a la rutina apropiada para atenderla. En algunos sistemas el hardware se encarga de esto automáticamente.
4. La rutina del manejador de interrupciones procesa la interrupción.
5. Se restablece el estado del proceso interrumpido o del siguiente proceso.
6. Se ejecuta el proceso interrumpido o el siguiente proceso.

Una interrupción puede ser generada específicamente por un proceso en ejecución en cuyo caso se le suele denominar trampa o trap y se dice que esta sincronizada con la operación del proceso o bien puede ser causada por algún evento que puede estar relacionado o no con el proceso en ejecución en cuyo caso se dice que es asíncrona con la operación en proceso. Interrumpir es una forma económica de llamar la atención del CPU.

Tipos de interrupciones

Interrupción SVC.- (Supervisor Call) Son iniciadas por un proceso en ejecución que ejecute una instrucción SVC. Una interrupción SVC es una petición generada por el usuario de un servicio en particular del sistema. Puede ser como ejecutar una operación de E/S, obtener más memoria, o comunicarse con el operador del sistema.

Un usuario no puede entrar arbitrariamente al sistema operativo, debe solicitar permiso por medio de una SVC. El sistema operativo esta enterado de los usuarios que intentan rebasar sus límites y puede rechazar ciertas peticiones si el usuario no cuenta con los privilegios necesarios.

Interrupciones de E/S.- Son iniciadas por hardware de E/S. Este tipo de interrupciones indican al CPU el cambio de estado de algún canal o un dispositivo. Se producen cuando finaliza una operación de entrada/salida o cuando un dispositivo pasa al estado listo.

Interrupciones Externas.- Son causadas por diversos eventos, incluyendo la expiración de un cuanto (Quantum del reloj) que interrumpe, cuando pulsamos la tecla de interrupción de la consola o bien cuando recibimos la señal de otro procesador, esto en el caso de un sistema de múltiples procesadores.

Interrupciones de Reinicio.- Cuando se oprime el botón de reinicio de la consola o si llegara desde otro procesador una señal de reinicio.

Interrupciones de Verificación del Programa.- Son causadas por una amplia gama de problemas que pueden ocurrir cuando se ejecutan las instrucciones en lenguaje de máquina de un programa, dichos problemas incluyen:

- Una división entre cero.
- El exceso o defecto de los números que pueden ser manejados por las operaciones aritméticas.
- La presencia de datos con formato erróneo.
- El intento de ejecutar un código de operación inválido.
- El intento de hacer referencia a una localidad de memoria que este fuera de los límites de la memoria real.
- El intento por parte de un usuario no privilegiado de hacer uso de una instrucción privilegiada o el intento de hacer uso de un recurso protegido

Interrupciones por Verificación de Máquina.- Son ocasionadas por mal funcionamiento del hardware.

Cambio de Contexto

En el esquema de las interrupciones de los procesadores a alta escala se maneja mediante el sistema operativo rutinas denominadas manejadores de interrupciones de primer nivel o FLIH (First Level Interrupt Handlers) para procesar las diferentes clases de interrupciones. Existe un manejador de interrupciones de primer nivel por cada tipo de interrupción.

Cuando ocurre una interrupción, el sistema operativo guarda el estado del proceso interrumpido y transfiere el control al manejador de interrupciones del primer nivel apropiado. Esto se logra mediante una técnica llamada Cambio de Contexto.

Los manejadores de primer nivel deben distinguir entre interrupciones de la misma clase; el procesamiento de estas interrupciones es realizado entonces por alguno de los manejadores de interrupciones de segundo nivel.

Palabras de Estado o PSW (Program Status Word)

Las palabras de estado del programa controlan el orden de ejecución de las instrucciones y contienen información acerca del estado de un proceso. Hay tres tipos de PSW: nuevas, actuales y antiguas.

La dirección de la siguiente instrucción que se debe ejecutar se almacena en la PSW actual que indica también los tipos de interrupciones habilitadas e inhabilitadas en ese momento. El CPU permite que ocurran las interrupciones habilitadas; las interrupciones inhabilitadas permanecerán pendientes y solo en algunos casos se pasaran por alto.

El procesador nunca puede estar inhabilitado para una interrupción SVC o para una de reinicio o para ciertas interrupciones de verificación del programa.

En un sistema de un procesador solo hay una PSW actual, seis nuevas y seis antiguas, una por cada tipo de interrupción.