

SISTEMAS OPERATIVOS
PRIMER PARCIAL

APUNTES SOBRE PROCESOS

Carlos González García

Marzo 2017

DESCRIPCIÓN Y CONTROL DE PROCESOS

Todos los sistemas operativos multiprogramados se construyen en torno al concepto de proceso. La mayor parte de los requisitos que un sistema operativo debe cumplir se pueden expresar con referencia a procesos:

- El sistema operativo debe intercalar la ejecución de múltiples procesos para maximizar la utilización del procesador mientras se proporciona un tiempo de respuesta razonable.
- El sistema operativo debe reservar recursos para los procesos conforme a una política específica (por ejemplo, ciertas funciones o aplicaciones son de mayor prioridad) mientras que al mismo tiempo evita interbloqueos.
- Un sistema operativo puede requerir dar soporte a la comunicación entre procesos y la creación de procesos, mediante los cuales ayuda a la estructuración de las aplicaciones.

¿QUÉ ES UN PROCESO?

Existen diferentes definiciones del concepto de proceso:

- Un programa en ejecución.
- Una instancia de un programa ejecutado en un computador.
- La entidad que se puede asignar y ejecutar en un procesador.
- Una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual, y un conjunto de recursos del sistema asociados.

También es común pensar en un proceso como una entidad que consiste en un número de elementos: el **código de programa** (que puede compartirse con otros procesos que estén ejecutando el mismo programa) y un **conjunto de datos** asociados a dicho código.

En cualquier instante de tiempo, mientras un proceso se encuentra en ejecución, se puede caracterizar por una serie de elementos:

- Identificador: un identificador único para distinguirlo del resto de procesos.
- Estado: estado del proceso, en este caso: "ejecución".
- Prioridad: nivel de prioridad relativo al resto de procesos.
- Contador de programa: la dirección de la siguiente instrucción del programa que se ejecutará.
- Punteros a memoria: incluye los punteros al código de programa y los datos asociados a dicho proceso, además de cualquier bloque de memoria compartido con otros procesos.
- Datos de contexto: datos que están en los registros del procesador cuando el proceso está en ejecución.
- Información de estado de E/S: incluye las peticiones de E/S pendientes, los dispositivos de E/S asignados a dicho proceso, una lista de los ficheros en uso por el mismo, etc.
- Información de auditoría: puede incluir la cantidad de tiempo de procesador y de tiempo de reloj utilizados, así como los límites de tiempo, registros contables, etc.

Toda la información mencionada anteriormente se almacena en una estructura de datos llamada **bloque de control de proceso** (BCP) que el sistema operativo crea y gestiona.

Lo más significativo que ofrece el BCP es que contiene la suficiente información, de forma que es posible interrumpir el proceso cuando está en ejecución y posteriormente restaurar su estado como si no hubiera habido interrupción alguna.

El BCP es la herramienta clave que permite al sistema operativo dar soporte a múltiples procesos y proporcionar multiprogramación.

Con esto, podemos concluir que un proceso está compuesto por el código de programa, los datos asociados y por el BCP. Un computador monoprocesador en un determinado instante solo puede tener un proceso en ejecución.

ESTADOS DE LOS PROCESOS

Para que un programa se ejecute, se debe crear un proceso o tarea para dicho programa.

Podemos caracterizar el comportamiento de un proceso, listando la secuencia de instrucciones que se ejecutan para dicho proceso. A esta lista la llamamos **traza del proceso**. De la misma manera, se puede caracterizar el comportamiento de un procesador mostrando como las trazas de varios procesos se entrelazan.

Un modelo de proceso de dos estados

Como hemos dicho, la responsabilidad principal del sistema operativo es controlar la ejecución de los procesos, esto incluye determinar el patrón de entrelazado para la ejecución y asignar recursos a los procesos. Esto conlleva que el primer paso para diseñar un sistema operativo para el control de procesos es describir el comportamiento que se desea que tengan los procesos.

El modelo más simple que se puede construir surge observando que, en un determinado instante, un proceso está siendo ejecutado o no. Con esto, en este modelo tenemos dos estados: **Ejecutando** y **No Ejecutando**.

Cuando el sistema operativo crea un nuevo proceso, crea el BCP e inserta el proceso en el sistema en estado de "No Ejecutando". De cuando en cuando, el proceso actualmente en ejecución se interrumpirá y una parte del sistema operativo, el activador, seleccionará otro proceso a ejecutar. El proceso saliente pasará de estado "Ejecutando" a "No Ejecutando" y un nuevo proceso pasará al estado de "Ejecutando".

Creación y terminación de procesos

Antes de seguir describiendo los diferentes modelos de estados de procesos, vamos a hablar de la creación y terminación de procesos, algo común a todos los modelos.

Creación de un proceso

Cuando se va a añadir un nuevo proceso a aquellos que se están gestionando en un determinado momento, el sistema operativo construye las estructuras de datos que se usan para manejar el proceso y reserva el espacio de direcciones en memoria principal para el proceso. Estas acciones constituyen la creación de un nuevo proceso.

Existen cuatro eventos comunes que llevan a la creación de un proceso. En un entorno por lotes, un proceso se crea como respuesta a una solicitud de trabajo. En un entorno interactivo, un proceso se crea cuando un nuevo usuario entra en el sistema. En ambos casos, el sistema operativo es responsable de la creación de procesos. Un sistema operativo puede, a petición de una aplicación, crear nuevos procesos. Por ejemplo, si un usuario solicita que se imprima un documento, el sistema operativo puede crear un

proceso que se ocupe de gestionar la impresión. De esta manera, el proceso solicitado puede operar independientemente del tiempo requerido para completar la tarea de impresión.

En ocasiones puede ser útil que un proceso cree otro proceso. Cuando un proceso lanza otro, al primero se le denomina proceso padre y al creado se le denomina proceso hijo. Habitualmente la relación entre proceso requiere de comunicación y cooperación entre ellos.

Terminación de procesos

Todo sistema debe proporcionar los mecanismos mediante los cuales un proceso indica su finalización o que ha completado su tarea. Un trabajo por lotes debe incluir una instrucción *HALT* o una llamada a un servicio del sistema operativo específica para su terminación. La instrucción *HALT* generará una interrupción para indicar al sistema operativo que dicho proceso ha finalizado. En contraposición, para una aplicación interactiva, las acciones del usuario indicarán cuando el proceso ha terminado.

Adicionalmente, un número de error o una condición de fallo puede llevar a la finalización de un proceso. A continuación, se muestra una lista con las condiciones más habituales de finalización de un proceso:

- Finalización normal: el proceso ejecuta una llamada al sistema operativo para indicar que ha completado su ejecución.
- Límite de tiempo excedido: el proceso ha ejecutado más tiempo del especificado.
- Memoria no disponible: el proceso requiere más memoria de la que el sistema puede proporcionar.
- Violaciones de frontera: el proceso trata de acceder a una posición de memoria a la cual no tiene acceso permitido.
- Error de protección: el proceso trata de usar un recurso, por ejemplo, un fichero, al que no tiene permitido acceder o trata de utilizarlo de una forma incorrecta (escribir en un fichero de solo lectura).
- Error aritmético: el proceso trata de realizar una operación de cálculo no permitida como, por ejemplo, una división por cero.
- Límite de tiempo: el proceso ha esperado más tiempo que el especificado en un valor máximo para que se cumpla un determinado evento.
- Fallo de E/S: se ha producido un error durante una operación de entrada o salida, por ejemplo, la imposibilidad de encontrar un fichero.
- Instrucción no válida: el proceso intenta ejecutar una instrucción inexistente (habitualmente el resultado de un salto a un área de datos y el intento de ejecutar dichos datos).
- Instrucción privilegiada: el proceso intenta utilizar una instrucción reservada al sistema operativo.
- Uso inapropiado de datos: una porción de datos es de tipo erróneo o no se encuentra inicializada.
- Intervención del operado por el sistema operativo: por alguna razón, el operador o el sistema operativo ha finalizado el proceso (por ejemplo, se ha dado una condición de interbloqueo).
- Terminación del proceso padre: cuando un proceso padre termina, el sistema operativo puede automáticamente finalizar todos los procesos hijos descendientes de dicho padre.

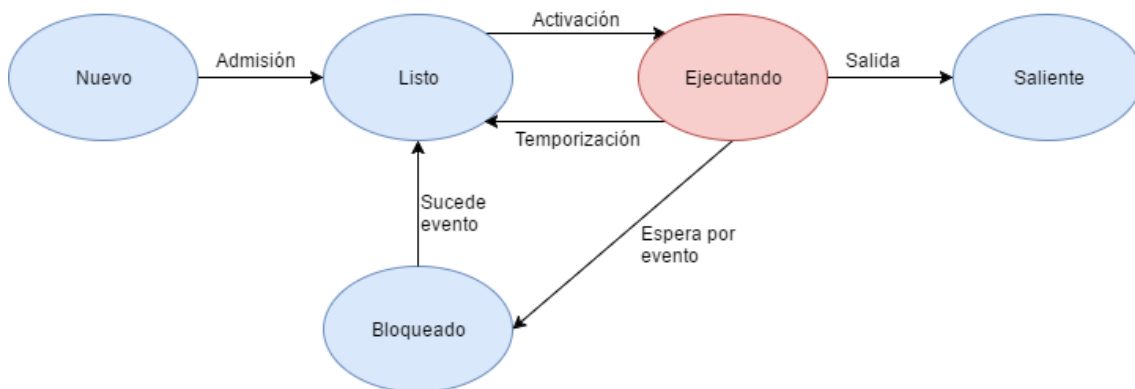
- Solicitud del proceso padre: un proceso padre habitualmente tiene autoridad para finalizar sus propios procesos descendientes.

Modelo de proceso de cinco estados

Si todos los procesos estuviesen siempre preparados para ejecutar, la gestión de colas sería efectiva. La cola es una lista de tipo FIFO y el procesador opera siguiendo una estrategia cíclica sobre todos los procesos disponibles. Sin embargo, esta implementación es inadecuada, algunos procesos que están en “No Ejecutando” están listos para ejecutarse, mientras que otros están bloqueados, esperando a que se complete una operación de E/S. Por lo tanto, utilizando una única cola el activador no puede seleccionar únicamente los procesos que lleven más tiempo en la cola. En su lugar, debería recorrer la lista buscando los procesos que no se encuentran bloqueados y que lleven en la cola más tiempo.

Como consecuencia de este problema, surge un modelo que cuenta con cinco estados, los cuales se describen a continuación:

- **Ejecutando:** el proceso está actualmente en ejecución.
- **Listo:** un proceso que se prepara para ejecutar en cuanto tenga la oportunidad.
- **Bloqueado:** un proceso que no se puede ejecutar hasta que se cumpla un evento determinado o se complete una operación de E/S.
- **Nuevo:** un proceso que se acaba de crear y que aún no ha sido admitido en el grupo de procesos ejecutables por el sistema operativo.
- **Saliente:** un proceso que ha sido liberado del grupo de procesos ejecutables por el sistema operativo, debido a que ha sido detenido o que ha sido abortado por alguna razón.



Los estados “Nuevo” y “Saliente” son útiles para construir la gestión de procesos. Mientras un proceso está en el estado de “Nuevo”, la información relativa al proceso que necesite el sistema operativo se mantiene en tablas de control de memoria principal, sin embargo, el proceso en sí mismo no se encuentra en memoria principal, el programa permanece en almacenamiento secundario, normalmente en disco.

De forma similar, un proceso sale del sistema en dos fases. Cuando ocurre la terminación del proceso, su estado pasa a ser “Saliente”. En este punto, el proceso no puede volver a pasar nunca a ejecución. Las tablas y otra información relacionada con el trabajo se encuentran temporalmente preservadas por el sistema operativo, el cual proporciona tiempo para que programas auxiliares o de soporte extraigan la información necesaria. Una vez que la información ha sido extraída, el sistema operativo borra el proceso del sistema.

Este modelo de estados cuenta con dos colas: la cola de listos y la cola de bloqueados. Cada proceso admitido por el sistema se coloca en la cola de listos. Cuando el proceso en ejecución deja de utilizar el procesador, o bien finaliza o bien termina en la cola de listos o en la de bloqueados, dependiendo de las circunstancias. Por último, cuando sucede un evento, cualquier proceso de la cola de bloqueados que únicamente esté esperando a dicho evento, pasa a la cola de listos.

Si lo pensamos bien, cuando sucede un evento, el sistema operativo tiene que recorrerse la cola entera de bloqueados e ir sacando a los correspondientes, lo cual es ineficiente, así que resulta más útil tener una cola para cada evento.

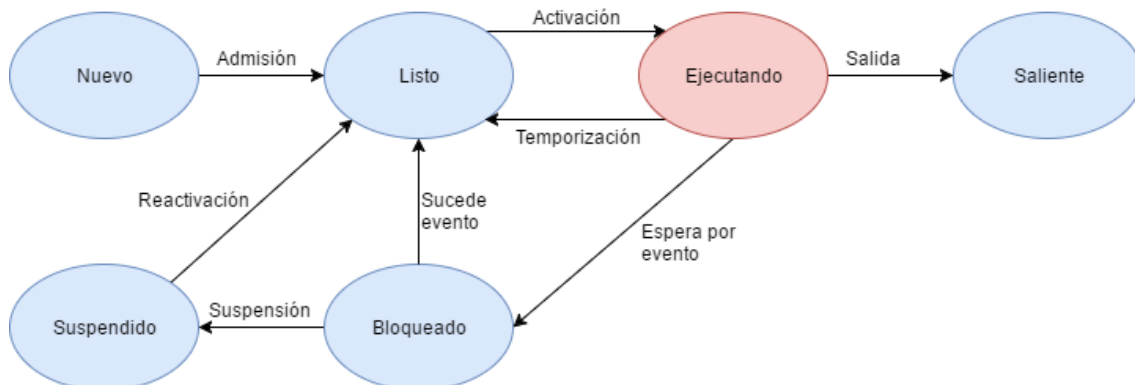
Procesos suspendidos

Los tres principales estados descritos anteriormente (“Listo”, “Ejecutando” y “Bloqueado”) proporcionan una forma sistemática de modelizar el comportamiento de los procesos y diseñar la implementación del sistema operativo. Sin embargo, existe una buena justificación para añadir otros estados al modelo.

Sabemos que las operaciones de E/S son mucho más lentas que los procesos de cómputo y, por tanto, el procesador en un sistema monoprogramado estaría ocioso la mayor parte del tiempo. Es verdad que, en este caso, la memoria almacena múltiples procesos y el procesador puede asignarse a otro proceso si el que lo usa está bloqueado, pero la diferencia de velocidad entre el procesador y la E/S es tal que es probable que todos los procesos de memoria se encontrasen a esperas de dichas operaciones.

Una solución para este problema es que la memoria principal puede expandirse para acomodar más procesos, pero esto presenta dos fallos: primero, hay un coste asociado a la memoria principal, segundo, cada vez ocupan más memoria los programas, luego, las grandes memorias han llevado a ejecutar procesos de mayor tamaño, no más procesos.

Otra solución para el problema es el llamado **swapping** (memoria de intercambio), que implica mover parte o todo el proceso de memoria principal al disco. Como ninguno de los procesos de memoria principal se encuentra en estado de “Listo”, el sistema operativo intercambia uno de los procesos bloqueados a disco, en la cola de “Suspendidos”, la cual es una lista de procesos existentes que han sido temporalmente expulsados de la memoria principal. A cambio, el sistema operativo trae otro proceso de la cola de “Suspendidos” o responde a una solicitud de un nuevo proceso.



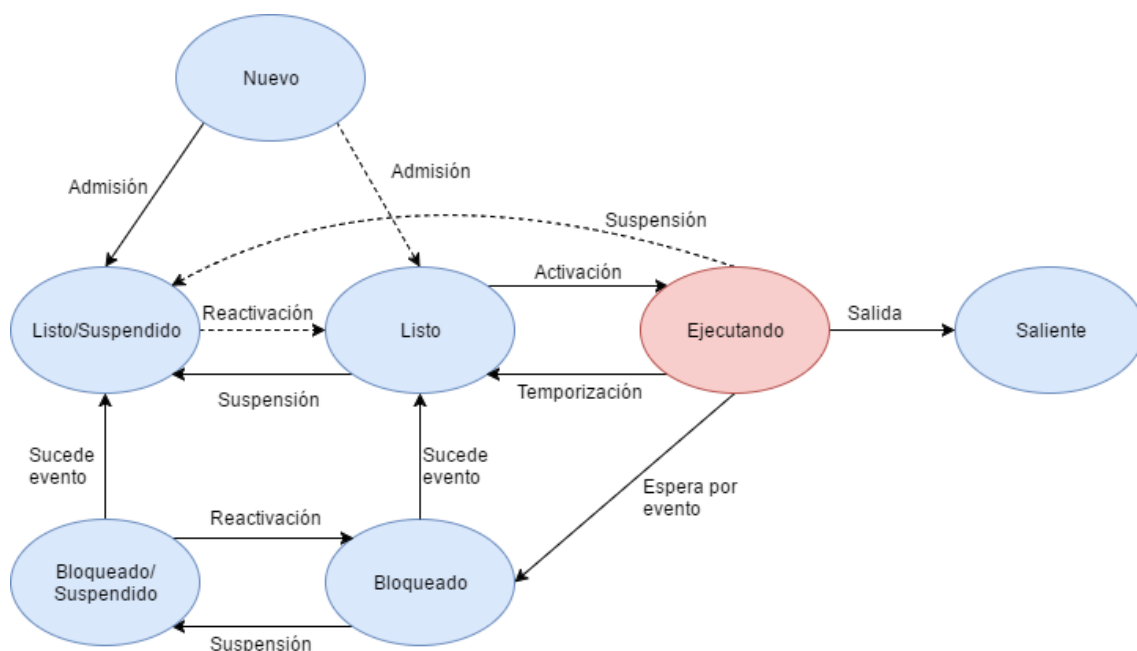
El swapping es una operación de E/S lo que podría hacer que el problema empeore, pero debido a que la E/S de disco es habitualmente más rápida que la E/S de otros sistemas, el swapping normalmente mejora el rendimiento del sistema.

Con el uso del swapping nos vemos obligados a añadir un nuevo estado a nuestro diagrama de comportamiento de procesos. Cuando todos los procesos en memoria principal se encuentran en el estado de “Bloqueado”, el sistema operativo puede suspender un proceso.

Cuando el sistema operativo ha realizado la operación de swap tiene dos opciones para seleccionar un nuevo proceso para traerlo a memoria principal: puede admitir un nuevo proceso que se haya creado o puede traer un proceso que se encontrase anteriormente en el estado de “Suspendido”, parece preferible traer un proceso que se encuentre en la cola de suspendidos, en lugar de incrementar la carga total del sistema.

Pero este razonamiento presenta una dificultad: todos los procesos suspendidos se encontraban previamente bloqueados, lo que hace que traer un proceso suspendido de nuevo a memoria, podría conllevar que este aún no estuviese en el estado de “Listo”. Eso nos hace replantearnos el diseño, modificando los estados que teníamos antes:

- **Listo:** un proceso está en memoria principal listo para ser ejecutado.
- **Bloqueado:** el proceso está en memoria principal esperando a un evento.
- **Bloqueado/Suspendido:** el proceso se encuentra en almacenamiento secundario esperando a un evento.
- **Listo/Suspendido:** el proceso está en almacenamiento secundario, pero está disponible para su ejecución tan pronto como se cargado a memoria principal.



La explicación que hemos dado a todos los sucesos anteriores asume que no se utiliza memoria virtual y que un proceso se encuentra totalmente en memoria principal o completamente fuera de la misma. Aunque los procesos estén parcialmente en memoria virtual, la técnica del swapping es igualmente necesaria.

DESCRIPCIÓN DE PROCESOS

El sistema operativo controla los eventos dentro del computador, planifica y activa los procesos para su ejecución por el procesador, reserva recursos para los mismos y responde a las solicitudes de servicios básicos de los procesos de usuario. Fundamentalmente se piensa en el sistema operativo como en la entidad que gestiona el uso de recursos del sistema por parte de los procesos.

Estructuras de control del sistema operativo

Si el sistema operativo se encarga de la gestión de procesos y recursos, debe disponer de información sobre el estado actual de cada proceso y cada recurso. El mecanismo universal para proporcionar esta información es el siguiente: el sistema operativo construye y mantiene tablas de información sobre cada entidad que gestiona.

El sistema operativo suele manejar tablas de 4 tipos: memoria, ficheros, E/S y procesos.

Las **tablas de memoria** se usan para mantener un registro tanto en de la memoria principal (real) como de la secundaria (virtual). Parte de la memoria principal está reservada para el uso del sistema operativo, el resto está disponible para el uso de los procesos. Las tablas de memoria deben incluir la siguiente información:

- Las reservas de memoria principal por parte de los procesos.
- Las reservas de memoria secundaria por parte de los procesos.
- Todos los atributos de protección que restringe el uso de la memoria principal y virtual, de forma que los procesos pueden acceder a ciertas áreas de memoria compartida.
- La información necesaria para manejar la memoria virtual.

El sistema operativo debe utilizar las **tablas de E/S** para gestionar los dispositivos de E/S y los canales del computador. Pero, en un instante determinado, un dispositivo de E/S puede estar disponible o asignado a un proceso en particular. Si una operación de E/S se está llevando a cabo, el sistema operativo necesita conocer el estado de la operación y la dirección de memoria principal del área usada como fuente o destino de la transferencia de E/S.

El sistema operativo también puede mantener las **tablas de ficheros**. Estas tablas proporcionan información sobre la existencia de ficheros, su posición en almacenamiento secundario, su estado actual y otros atributos. La mayoría de esta información se puede gestionar mediante el sistema de ficheros.

Por último, el sistema operativo debe mantener **tablas de procesos** para poder gestionar los procesos.

Cabe destacar que todas las tablas mencionadas se entrelazan entre sí. Memoria, E/S y ficheros se gestionan por parte de los procesos, de forma que debe haber algunas referencias en estos recursos, directa o indirectamente, desde las tablas de procesos.

Estructuras de control de proceso

La información que debe conocer el sistema operativo si quiere manejar y controlar los procesos es la siguiente: primero, debe conocer donde están localizados los procesos, y segundo, debe conocer los atributos de los procesos que quiere gestionar.

Localización de los procesos

Un proceso debe consistir en una cantidad suficiente de memoria para almacenar el programa y datos del mismo. Adicionalmente, es típico que la ejecución de un programa incluya una pila para guardar las llamadas a procedimientos y los parámetros pasados por dichos procedimientos. Por último, cada proceso está asociado a un número de atributos que son utilizados por el sistema operativo para controlar el proceso. El conjunto de estos atributos es el BCP. Nos referimos al conjunto de programa, datos, pila y atributos como la **imagen del proceso**.

Normalmente esta imagen de proceso se mantiene como un bloque de memoria contiguo, que se mantiene en memoria secundaria (normalmente en disco). Para que el sistema operativo pueda gestionar el proceso, al menos una pequeña porción de su imagen debe mantenerse en memoria principal. Para que el proceso pueda ejecutarse, toda su imagen debe cargarse en memoria principal, o al menos, en memoria virtual.

Los sistemas operativos modernos cuentan con un hardware de paginación que permite el uso de memoria física no contigua.

Existe una tabla primaria de procesos con una entrada para cada proceso. Cada entrada contiene, al menos, un puntero a la imagen del proceso.

Atributos de proceso

Todo sistema operativo multiprogramado requiere de una gran cantidad de información para manejar cada proceso. Esta información reside en el BCP.

Existen distintas categorías de información que el sistema operativo requiere para cada proceso. Podemos agrupar la información del BCP en tres categorías generales:

- Identificación del proceso: **Identificadores**: identificadores numéricos que se pueden guardar dentro del BCP: identificadores del proceso, identificador del proceso padre, identificador del usuario.
- Información de estado del procesador: **Registros visibles por el usuario**: aquellos registros a los que se puede hacer referencia por medio del lenguaje máquina que ejecuta el procesador cuando está en modo usuario. Normalmente existen de 8 a 32 de estos registros. **Registros de estado y control**: registros del procesador que se utilizan para el control de operaciones. Estos incluyen: contador de programa (contiene la dirección de la siguiente instrucción a analizar), códigos de condición (resultan de la operación lógica o aritmética más reciente) e información de estado (incluyen los *flags* de interrupciones habilitadas/deshabilitadas, modo ejecución). **Puntero de pila**: cada proceso tiene una o más pilas LIFO. El puntero de pila apunta a la parte más alta de la pila.
- Información de control de proceso: **Información de estado y de planificación**: información que el sistema operativo necesita para analizar las funciones de planificación. Elementos típicos de esta información son: estado del proceso (indica si el proceso está listo o no para ser planificado para su ejecución), prioridad (prioridad de planificación del proceso), información relativa a planificación (dependerá del algoritmo de planificación utilizado) y evento (identificación del evento al cual el proceso está esperando para continuar su ejecución). **Estructuración de datos**: un proceso puede estar enlazado con otro en una cola o anillo. También un proceso puede tener una relación padre-hijo con otro proceso. El BCP puede tener punteros a otros procesos para dar

soporte a estas estructuras. **Comunicación entre procesos:** se pueden asociar *flags*, señales y mensajes relativos a la comunicación entre dos procesos independientes. Parte o toda esta información se puede mantener en el BCP. **Privilegios de proceso:** los procesos adquieren privilegios de acuerdo con la memoria a la que van a acceder y los tipos de instrucciones que van a ejecutar. **Gestión de memoria:** esta sección incluye punteros a tablas de segmentos y/o páginas que describen la memoria virtual asignada al proceso. **Propia de recursos y utilización:** se deben indicar los recursos controlados por el proceso, así como un histórico de uso del procesador u otros recursos.

El identificador de proceso es un valor numérico único. Con él, el sistema operativo puede localizar las tablas apropiadas.

La información de estado de proceso es útil ya que cuando un proceso se interrumpe, toda la información de los registros debe salvaguardarse de forma que en un futuro pueda restaurarse.

Por último, la información de control de proceso es necesaria para el sistema operativo para que este pueda controlar y coordinar varios procesos activos.

A continuación, se muestra la estructura de una imagen de proceso en memoria virtual:

Identificador del proceso
Información de estado del procesador
Información de control del proceso
Pila de usuario
Espacio privado de direcciones de usuario (programas y datos)
Espacio compartido de direcciones

Como vimos antes, el BCP puede contener información estructural, incluyendo punteros que permiten enlazar BCPs entre sí. De esta forma, las colas de las que hablamos al principio pueden implementarse como listas enlazadas de BCPs.

El BCP es la más importante de las estructuras de datos del sistema operativo. Podemos decir que el conjunto de BCPs definen el estado del sistema operativo.

Un gran número de rutinas dentro del sistema operativo necesitan acceder a información de los BCP. Proporcionar acceso directo a estas tablas no es difícil ya que cada proceso lleva un identificador único que permite indexar dentro de una tabla de punteros a BCPs. La dificultad no reside en el acceso sino en la protección. Dos de los posibles problemas que se pueden presentar son:

- Un fallo en una simple rutina, como un manejador de interrupción puede dañar los BCPs, lo que puede destruir la capacidad del sistema para manejar los procesos afectados.
- Un cambio en la estructura o en la semántica de los BCP puede afectar a un gran número de módulos del sistema operativo.

Estos problemas mencionados se pueden tratar obligando a que todas las rutinas del sistema operativo pasen a través de una rutina de manejador, cuyo único trabajo es proteger los BCPs y que es la única que realiza el arbitraje en las operaciones de lectura y escritura de dichos bloques. Los factores a equilibrar en el uso de esta rutina son, por un lado, los aspectos de rendimiento, y por el otro, el grado en el cual el resto del software del sistema puede verificarse para determinar su corrección.

CONTROL DE PROCESOS

Modos de ejecución

La mayoría de los procesadores proporcionan la opción de ejecutar en dos modos. Ciertas instrucciones se pueden ejecutar en modos privilegiados únicamente. Éstas incluirán lectura y modificación de los registros de control e instrucciones relacionadas con la gestión de memoria. Adicionalmente, ciertas regiones de memoria sólo se pueden acceder en los modos más privilegiados.

El modo menos privilegiado se suele denominar **modo usuario**, porque los programas de usuario a menudo se ejecutan en este modo. El modo más privilegiado se llama **modo sistema, modo control o modo núcleo**. Este último término se refiere al núcleo del sistema operativo, que es la parte del sistema operativo que engloba las funciones más importantes del sistema.

A continuación, se listan las funciones que normalmente se encuentran dentro del núcleo del sistema operativo.

En cuanto a la gestión de procesos tenemos:

- Creación y terminación de procesos.
- Planificación y activación de procesos.
- Intercambio de procesos.
- Sincronización de procesos y soporte para comunicación entre procesos.
- Gestión de los bloques de control de proceso.

En cuanto a la gestión de memoria tenemos:

- Reserva de espacio de direcciones para los procesos.
- *Swapping*.
- Gestión de páginas y segmentos.

En cuanto a la gestión de E/S:

- Gestión de *buffers*.
- Reserva de canales de E/S y de dispositivos para los procesos.

Por último, en cuanto a las funciones de soporte:

- Gestión de interrupciones.
- Auditoría.
- Monitorización.

El motivo por el cual existen dos modos es claro: es necesario proteger al sistema operativo y a las tablas clave del sistema, por ejemplo, los BCPs, de la interferencia con programas de usuario. En modo núcleo, el software tiene control completo del procesador y de sus instrucciones, registros y memoria. Este nivel de control no es necesario y por seguridad no es recomendable para los programas de usuario.

El procesador sabe en qué modo está ejecutando gracias a un bit que se encuentra en la palabra de estado de programa (PSW).

Creación de procesos

Estos son los pasos que sigue un sistema operativo para crear un proceso:

- **Asignar un identificador de proceso único al proceso:** en este instante se añade una nueva entrada a la tabla primaria de procesos, que contiene una entrada por proceso.
- **Reservar espacio para proceso:** esto incluye todos los elementos de la imagen del proceso. Para ello es necesario que el sistema operativo conozca cuanta memoria se requiere para el espacio de direcciones privado y para la pila de usuario. Si existe una parte del espacio de direcciones compartido por este nuevo proceso, se fijan los enlaces apropiados. Por último, se debe reservar el espacio para el BCP.
- **Inicialización del BCP:** La parte de identificación de proceso contiene el identificador único. En la información de estado de proceso se inicializan la mayoría de las entradas a 0, excepto el contador de programa (fijado en el punto de entrada del programa) y los punteros de pila del sistema (fijados para definir los límites de la pila del proceso). La parte de información de control de procesos se inicializa en base a los valores por omisión, considerando también los atributos que han sido solicitados para este proceso. El estado de proceso se inicializa normalmente a "Listo" o a "Listo/Suspendido". Inicialmente el proceso no debe poseer ningún recurso (dispositivo E/S, fichero) a no ser que exista una indicación explícita para ello o que hayan sido heredados del padre.
- **Establecer los enlaces apropiados:** por ejemplo, si el sistema operativo mantiene cada cola del planificador como una lista enlazada, el nuevo proceso debe situarse en la cola de "Listos" o en la cola de "Listos/Suspendidos".
- **Creación o expansión de otras estructuras de datos:** el sistema operativo puede mantener un registro de auditoría por cada proceso que se puede utilizar posteriormente a efectos de facturación y/o de análisis de rendimiento del sistema.

Cambio de proceso

A primera vista, la operación de cambio de proceso puede parecer fácil, sin embargo, es necesario tener en cuenta algunas consideraciones de diseño: primero, ¿qué evento dispara los cambios de proceso? Otra consideración pasa por conocer la distinción entre cambio de proceso y cambio de modo y, por último, ¿qué debe hacer el sistema operativo con las diferentes estructuras que gestiona para proporcionar un cambio de proceso?

¿Cuándo se realiza el cambio de proceso?

Un cambio de proceso puede ocurrir en cualquier instante en el que el sistema operativo obtiene el control sobre el proceso actualmente en ejecución.

Lo más probable es que el sistema operativo tome el control debido a una interrupción, producidas por causa de algún tipo de evento que es externo e independiente al proceso actualmente en ejecución, por ejemplo, la finalización de una operación de E/S. En una interrupción, el control se transfiere inicialmente al manejador de interrupción, que realiza determinadas tareas internas y que posteriormente salta a una rutina del sistema

operativo, encargada de cada uno de los tipos de interrupciones en particular. Algunos ejemplos son:

- Interrupción de reloj: el sistema operativo determina si el proceso en ejecución ha excedido o no la unidad máxima de tiempo de ejecución, denominada “rodaja de tiempo”, es decir, la máxima cantidad de tiempo que un proceso puede ejecutar antes de ser interrumpido. En este caso, el proceso pasa al estado de “Listo” y se activa otro proceso.
- Interrupción de E/S: el sistema operativo determina que acción de E/S ha ocurrido. Si la acción de E/S constituye un evento por el cual están esperando uno o más procesos, el sistema operativo mueve esos procesos al estado de “Listo” y los que estaban en estado de “Bloqueado/Suspendido” pasan al estado de “Listo/Suspendido”.
- Fallo de memoria: el procesador se encuentra con una referencia a una dirección de memoria virtual, a una palabra que no se encuentra en memoria principal. El sistema operativo debe traer el bloque que contiene la referencia desde memoria secundaria a memoria principal. Después de que se solicita la operación de E/S para traer el bloque a memoria, el proceso que causó el fallo de memoria se pasa al estado de “Bloqueado”. Después de que el bloque de memoria necesitado se haya traído, el proceso pasa a “Listo”.

Un **trap** es una interrupción del sistema asociada a una condición de error o excepción generada dentro del proceso que se está ejecutando. Mediante el trap, el sistema operativo conoce si una condición de error o de excepción es irreversible. De ser así, el proceso en ejecución pasa a estado “Saliente” y se hace un cambio de proceso. De no ser así, el sistema operativo actuará dependiendo de la naturaleza del error.

Por último, el sistema operativo se puede activar por medio de una **llamada al sistema** procedente del programa en ejecución. La llamada al sistema puede conllevar en algunos casos que el proceso que la realiza pase a estado de “Bloqueado”.

Cambio de modo

Si hay una interrupción pendiente, el sistema operativo actúa de la siguiente manera:

- Coloca el contador de programa en la dirección de comienzo de la rutina del programa manejador de la interrupción.
- Cambia de modo usuario a modo núcleo de forma que el código de tratamiento de la interrupción pueda incluir instrucciones privilegiadas.

Justo después, el procesador pasa a la fase de búsqueda de instrucción y busca la primera instrucción del programa de manejo de interrupción, que dará servicio a la misma. En este punto, habitualmente el contexto del proceso que se ha interrumpido se guarda en el BCP del programa interrumpido.

El contexto que se salvaguarda constituye toda la información que se puede ver alterada por la ejecución de la rutina de interrupción y que se necesitará para la continuación del proceso que ha sido interrumpido. De esta forma se puede guardar la parte del BCP que hace referencia a la información de estado del procesador. Esto incluye el contador de programa, otros registros del procesador y la información de la pila.

Si a esta interrupción le sigue un cambio de proceso, se necesita hacer alguna cosa más, pero en muchos sistemas operativos, la existencia de una interrupción no implica necesariamente un cambio de proceso.

Cambio del estado del proceso

Hemos visto que un cambio de modo puede ocurrir sin cambio del proceso que está en proceso de "Ejecutando". Sin embargo, si el proceso que se está ejecutando va a pasar al estado de "Listo", "Bloqueado", etc., entonces el sistema operativo debe realizar cambios sustanciales en su entorno. Los pasos que se realizan para un cambio de proceso completo son:

- Salvar el estado del procesador, incluyendo el contador de programa y otros registros.
- Actualizar el BCP que está actualmente en "Ejecutando" por uno de los otros estados "Listo", "Bloqueado", "Listo/Suspendido", "Saliente". También hay que actualizar otros campos como el motivo por el cual se ha dejado de ejecutar el proceso y otra información de auditoría.
- Mover el BCP a la cola apropiada (Listo, Bloqueado en el evento X, Listo/Suspendido).
- Selección de un nuevo proceso a ejecutar.
- Actualizar el BCP del proceso elegido, lo que incluye pasarlo al estado de "Ejecutando".
- Actualizar las estructuras de datos de gestión de memoria. Esto se puede necesitar dependiendo de cómo se haga la traducción de direcciones.
- Restaurar el estado del procesador al que tenía en el momento en el que el proceso seleccionado salió del estado de "Ejecutando" por última vez, leyendo los valores anteriores de contador de programa y registros.

Por lo tanto, podemos concluir que un cambio de proceso, que implica un cambio de estado, requiere un mayor esfuerzo que un cambio de modo.

Ejecución del sistema operativo

Un sistema operativo tiene dos aspectos intrínsecos:

- Funciona de la misma forma que cualquier software, en el sentido que un sistema operativo es un conjunto de programas ejecutados por un procesador.
- Con frecuencia, cede el control y depende del procesador para recuperar dicho control.

Núcleo sin procesos

Cuando un proceso en ejecución se interrumpe o invoca una llamada a sistema, el contexto se guarda y el control pasa al núcleo. El sistema operativo tiene su propia región de memoria y su propia pila de sistema para controlar la llamada a procedimientos y sus retornos. El sistema operativo puede realizar todas las funciones que necesite y restaurar el contexto del proceso interrumpido que hace que se retome la ejecución del proceso de usuario afectado. De forma alternativa, el sistema operativo puede realizar la salvaguarda del contexto y la activación de otro proceso diferente. Si esto ocurre o no depende de la causa de la interrupción y de las circunstancias puntuales en el momento.

El punto clave aquí es que el concepto de proceso se aplica únicamente a los programas de usuario. El código del sistema operativo se ejecuta como una entidad independiente que requiere un modo privilegiado de ejecución.

Ejecución dentro de los procesos de usuario

Una alternativa que es muy común en los sistemas operativos de máquinas pequeñas es ejecutar virtualmente todo el software de sistema operativo en el contexto de un proceso de usuario. Con esta visión el sistema operativo se percibe como un conjunto de rutinas que el usuario invoca para realizar diferentes funciones, ejecutadas dentro del entorno del proceso de usuario.

A continuación, se muestra la imagen de proceso acorde con esta estrategia. Se usa una pila de núcleo separada para manejar llamadas/retornos cuando el proceso está en modo núcleo. El código del sistema operativo y sus datos están en el espacio de direcciones compartidas y se comparten entre todos los procesos.

Identificador del proceso
Información de estado del procesador
Información de control del proceso
Pila de usuario
Espacio privado de direcciones de usuario (programas y datos)
Pila del núcleo
Espacio compartido de direcciones

En esta estrategia, cuando ocurre una interrupción, un trap o una llamada a sistema, el procesador se pone en modo núcleo y el control se pasa al sistema operativo. Para este fin, el contexto se salva y se cambia de modo a una rutina del sistema operativo. Sin embargo, la ejecución continúa dentro del proceso de usuario actual. De esta forma, no se realiza un cambio de proceso, sino un cambio de modo dentro del mismo proceso.

Una de las principales ventajas de esta alternativa es la siguiente: un programa de usuario se interrumpe para utilizar alguna rutina del sistema operativo y luego continúa, todo ello sin necesitar haber hecho un doble cambio de proceso.

Sistemas operativos basados en procesos

Otra alternativa es implementar un sistema operativo como una colección de procesos de sistema. Como en las otras opciones, el software que es parte del núcleo se ejecuta en modo núcleo. En este caso, las principales funciones del núcleo se organizan como procesos independientes.

La implementación del sistema operativo como un grupo de procesos en entornos de multiprocesadores y multicomputadores, en los cuales determinados servicios del sistema operativo se pueden enviar a procesadores dedicados, incrementan el rendimiento.