

# UT 4: Gestión de recursos de un S.O.

## PROCESOS

Prof.: Baldomero Sánchez Pérez

## 1.2. Procesos de los Sistemas Operativos

- La interfaz entre el S. O. y los programas del usuario se define como el conjunto de **“instrucciones ampliadas”** que proporciona el S. O. y son las **“llamadas al sistema”**:
  - Crean, eliminan y utilizan **objetos del software** controlados por el S. O.:
    - Los mas importantes son **procesos** y **archivos**.
- **Procesos**
- **Archivos**
- **Llamadas al sistema (trap)**

## 1.2. Procesos de los Sistemas Operativos (2)

- **Procesos:** Es el concepto central de todos los S. O.
  - Es básicamente un programa en ejecución.
  - Consta del programa ejecutable, sus datos y pila, contador y otros registros, además de la información necesaria para ejecutar el programa.
  - La información de control relacionada con los procesos se almacena en la **tabla de procesos**:
    - Es administrada por el S. O.
    - Posee un arreglo de estructuras, una por cada proceso existente en ese momento.

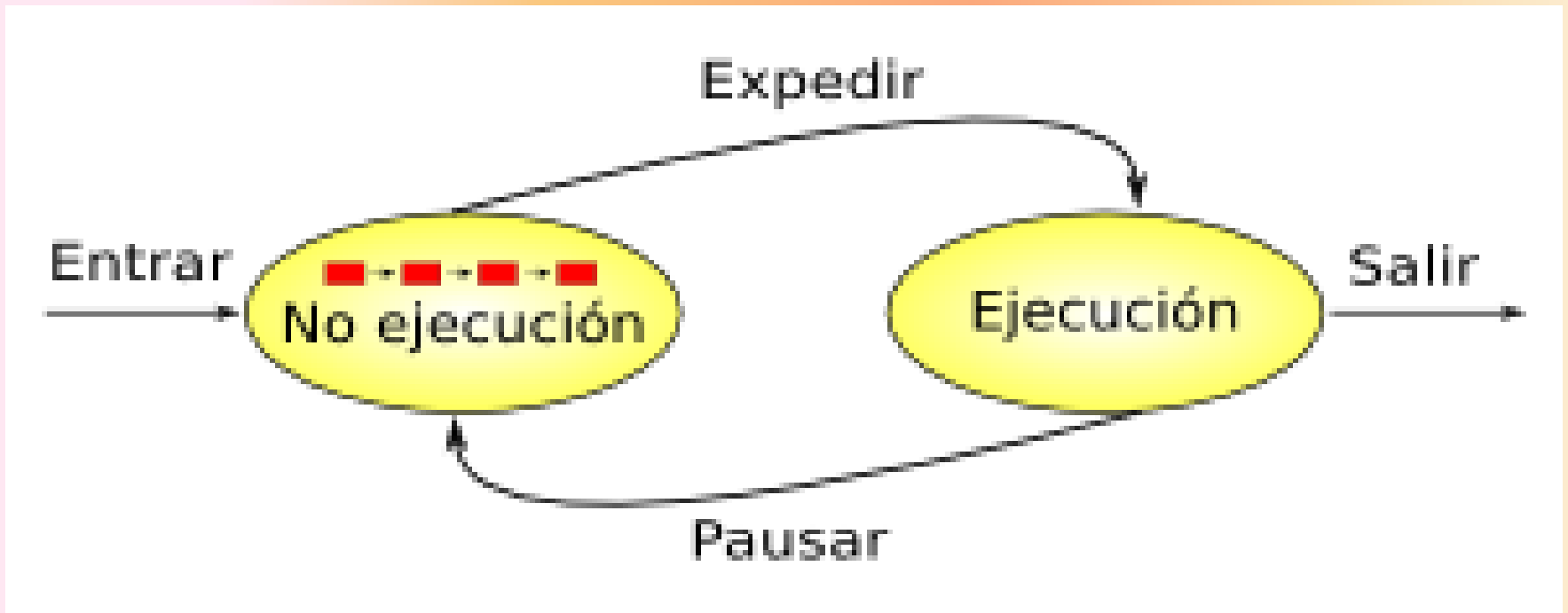
# 1.3. Concepto Proceso de los Sistemas Operativos

(3)

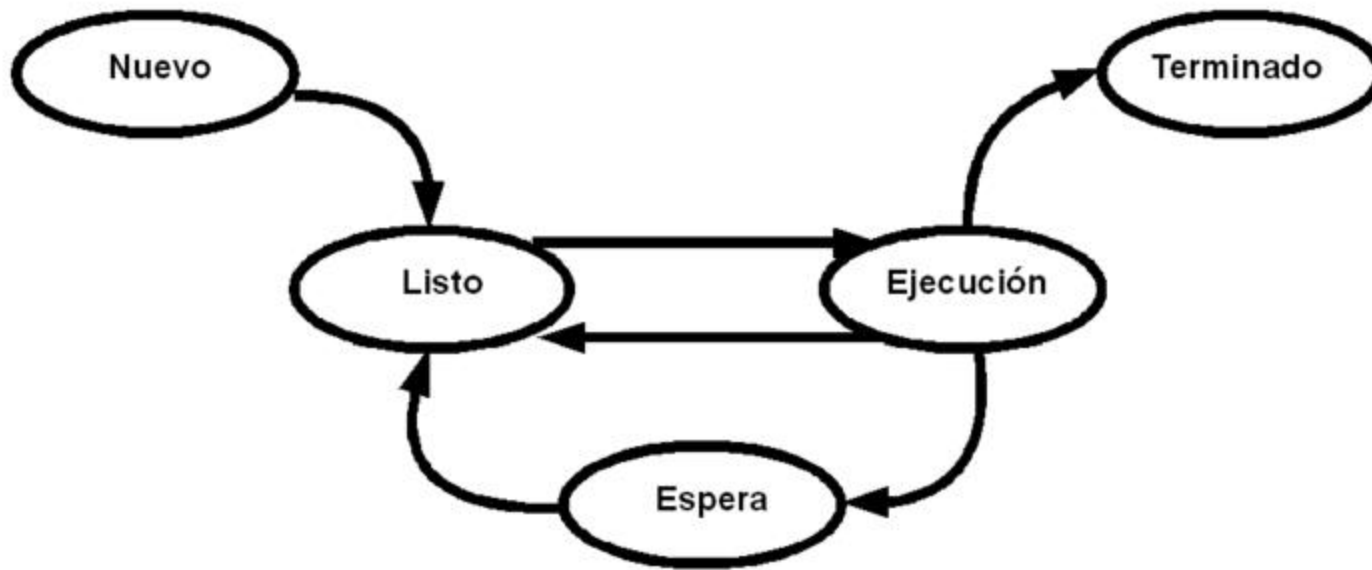
- PROCESO

- Un proceso (suspendido) consta de:
  - Un espacio de dirección.
  - Los datos pertinentes de la tabla de procesos.
- Un proceso (**proceso padre**) “PPID” puede crear **procesos hijo** “PID” y estos nuevos procesos hijo, conformando un **árbol de procesos**.

## 2.1 Procesos de 2 estados



## 2.2. Listado de procesos



## 2.3. Estados del proceso

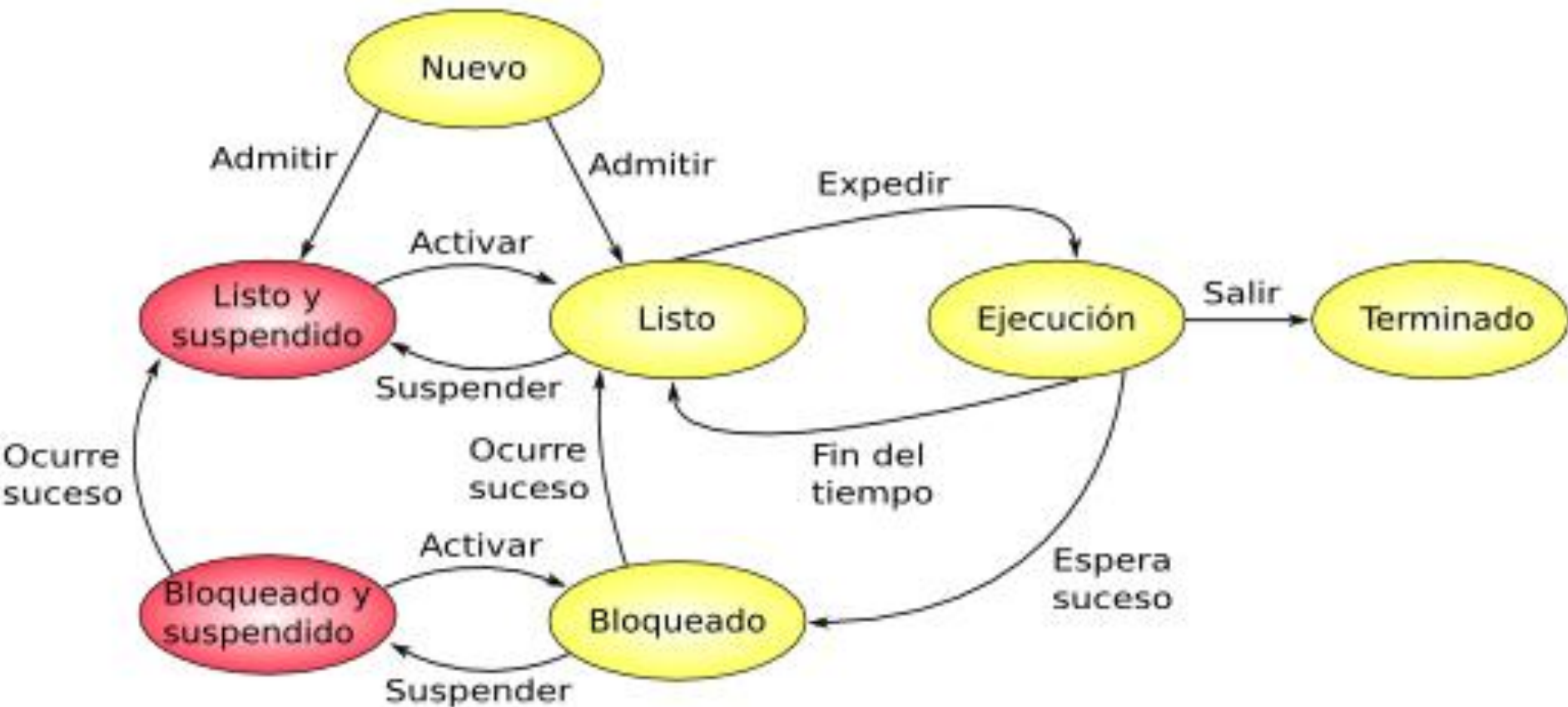
- Los cinco estados de este diagrama son los siguientes:
  - **Ejecución:** el proceso está actualmente en ejecución.
  - **Listo:** el proceso está listo para ser ejecutado, sólo está esperando que el **planificador** de corto plazo así lo disponga.
  - **Espera:** el proceso no puede ejecutar hasta que no se produzca cierto suceso, como la finalización de una operación de Entrada/Salida solicitada por una llamada al sistema operativo.

## 2.3. Estados del proceso (2)

- **Nuevo:** El proceso recién fue creado y todavía no fue admitido por el sistema operativo. En general los procesos que se encuentran en este estado todavía no fueron cargados en la memoria principal.
- **Terminado:** El proceso fue expulsado del grupo de procesos ejecutables, ya sea porque terminó o por algún fallo, como un error de protección, aritmético, etc.



### 3. Procesos suspendidos



## 3.1. Procesos Suspendidos (Hold)

- Dos o más procesos pueden cooperar mediante **señales** de forma que uno obliga a detenerse a los otros hasta que reciban una señal para continuar.
  - Se usa una variable de tipo **Semáforo** para sincronizar los procesos.
  - Si un proceso está esperando una señal, se suspende (Hold) hasta que la señal se envíe (SIGNAL).
  - Se mantiene una cola de procesos en espera en el semáforo.
  - La forma de elegir los procesos de la cola en ESPERA es mediante una política **FIFO** (First In First Out) también llamada **FCFS** (First Come First Served), Round Robin, etc.

## 3.1. Procesos Suspendidos (Hold) (2)

- La sincronización explícita entre procesos es un caso particular del estado "bloqueado".
  - En este caso, el suceso que permite desbloquear un proceso no es una operación de entrada/salida, sino una señal generada a propósito por el programador desde otro proceso.
- Utiliza las siguientes señales.
  - `SIGNAL()`
  - `WAIT()`
- Los semáforos pueden ser:
  - Binarios. 0, 1.
  - Generales. Múltiples valores positivos

## 3.2. Estados de un proceso UNIX



## 3.2. Estados de un proceso Unix (2)

- **Preparado (R).**-Proceso que está listo para ejecutarse. Simplemente está esperando a que el **sistema operativo** le asigne un **tiempo** de **CPU**.
- **Ejecutando (O).** Sólo uno de los **procesos** *preparados* se está ejecutando en cada momento (monoprocesador).
- **Suspendido (S).** Un proceso se encuentra *suspendido* si no entra en el reparto de CPU, ya que se encuentra esperando algún tipo de evento (por ejemplo, la recepción de una señal **software** o **hardware**). En cuanto dicho evento se produce, el proceso pasa a formar parte del conjunto de procesos *preparados*.

## 3.2. Estados de un proceso Unix (3)

- **Parado (T).** Un proceso *parado* tampoco entra en el reparto de CPU, pero no porque se encuentre suspendido esperando algún evento. En este caso, sólo pasarán a estar *preparados* cuando reciban una señal determinada que les permita continuar.
- **Zombie (Z).** Todo proceso al finalizar avisa a su proceso padre, para que éste elimine su entrada de la tabla de procesos. En el caso de que el padre, por algún motivo, no reciba esta **comunicación** no lo elimina de la tabla de procesos. En este caso, el proceso hijo queda en **estado zombie**, no está consumiendo CPU, pero sí continua consumiendo **recursos** del **sistema**.

## 3.3. Hilos de un proceso

- Un hilo es básicamente una tarea que puede ser ejecutada en paralelo con otra tarea.
- Los hilos de ejecución de un mismo proceso comparten los recursos hace que cualquiera de estos hilos pueda modificar éstos.
- Hilo o hebra.
- Tamaño del hilo:
  - **Hilo grueso:** tamaño es grande, quantum grande.
  - **Hilo fino:** tamaño pequeño, quantum pequeño.

## 3.4. Formas de multihilos

- Los SO generalmente implementan hilos de dos maneras:
  - ***Multihilo apropiativo***: permite al sistema operativo determinar cuándo debe haber un **cambio de contexto**.

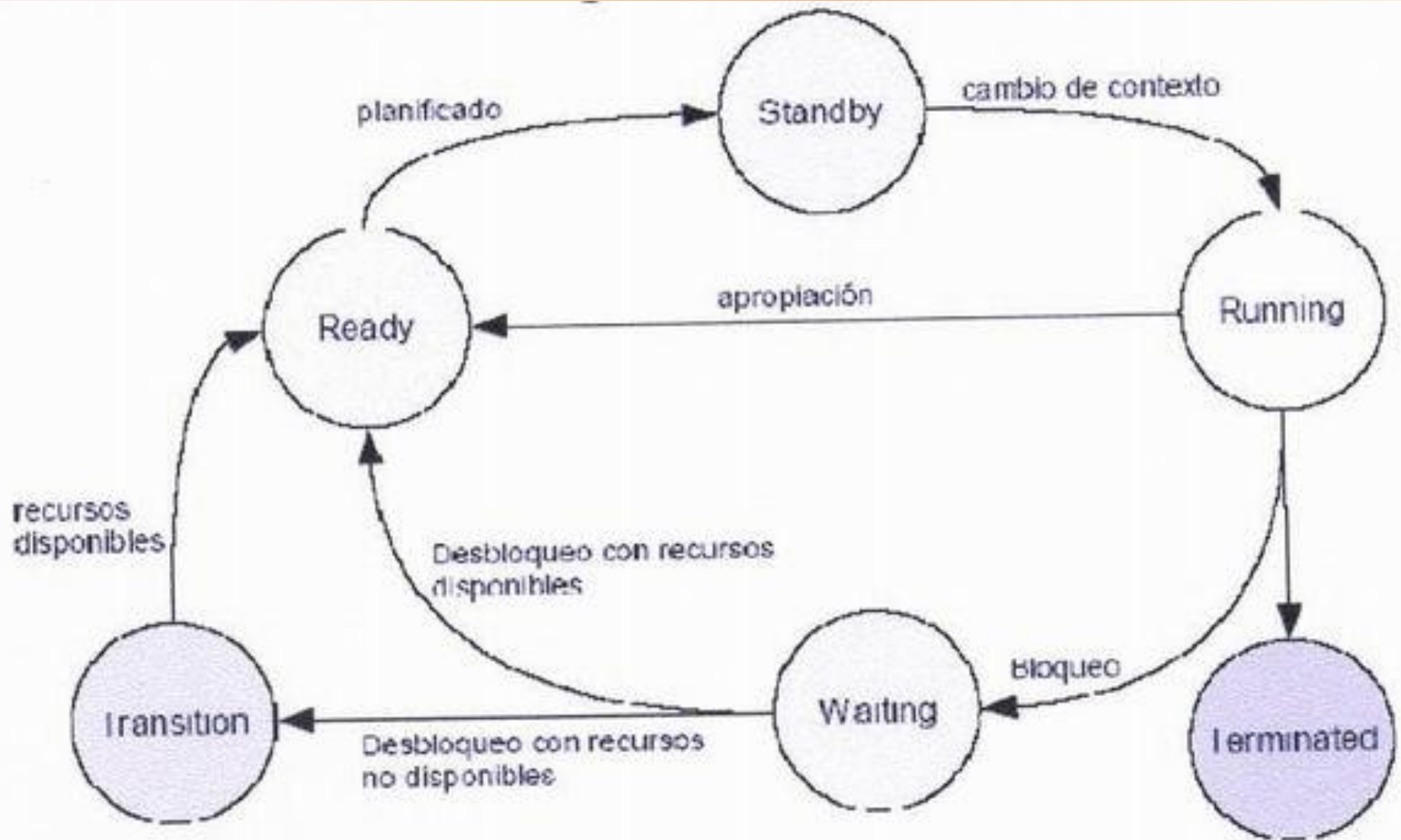
La desventaja de esto es que el sistema puede hacer un cambio de contexto en un momento inadecuado, causando un fenómeno conocido como **inversión de prioridades** y otros problemas.
  - ***Multihilo cooperativo***: depende del mismo hilo abandonar el control cuando llega a un punto de detención, lo cual puede traer problemas cuando el hilo espera la disponibilidad de un recurso.



## 3.5. Estados de un hilo en Unix

- **Listo:** el hilo puede ser elegido para su ejecución.
- **Standby:** el hilo ha sido elegido para ser el siguiente en ejecutarse en el **procesador**.
- **Ejecución:** el hilo está siendo ejecutado.
- **Espera:** un hilo pasa a este estado cuando se bloquea por un suceso
- **(E/S):** se realiza una espera voluntaria de sincronización o alguien suspende al hilo.
- **Transición:** después de una espera el hilo pasa a este estado si está listo para ejecutar pero alguno de sus recursos no está disponible aún.
- **Terminado:** un hilo llega a este estado cuando termina normalmente cuando su proceso padre ha terminado.

## 3.6. Estados de un hilo en Windows 2000



## 3.6. Estados de un hilo en Windows 2000 (2)

- **Ready:** El hilo listo para ejecutarse
- **Running:** El hilo está ejecutando
- **Standby:** El hilo ha sido seleccionado para ser ejecutado en un procesador particular.
- **Waiting:** Se encuentra bloqueado esperando un evento.
- **Transition:** Hilo bloqueado-suspendido, en espera de recursos
- **Terminated:** Finalización del hilo.

## 3.6. Estados de un proceso en Windows 2000 <sup>(3)</sup>

- En Windows 2000, la estructura original de los procesos y de los **servicios** que brinda el núcleo es relativamente simple y de propósito general.
- Las características más importantes de los procesos en W2K son:
  - Los procesos se implementan como objetos.
  - Un proceso ejecutable puede tener un hilo o más.
  - Los objetos proceso e hilo tienen capacidades de sincronización.

## 3.7. Estados de un proceso en Windows 2000 <sup>(4)</sup>

- **Un proceso** es una entidad correspondiente a un trabajo de usuario o a una aplicación, que dispone de sus propios recursos, tales como **memoria** y archivos.
- **Un hilo** es una unidad de trabajo que se puede expedir para su ejecución secuencial y que es interrumpible, de forma que el procesador puede pasar de un hilo a otro.

## 4. Conceptos de los Sistemas Operativos

- **Archivos**: Una de las funciones principales del S. O. es brindar **independencia de dispositivo**.
  - Muchos S. O. soportan el concepto de **directorio** como una forma de agrupar archivos.
  - Los directorios se estructuran jerárquicamente, por lo que a cada archivo le corresponde una **ruta de acceso**.
  - Existen distintos esquemas de seguridad de archivos en los distintos S. O.

## 4 . Conceptos de los Sistemas Operativos (2)

- **Llamadas al sistema:** Permiten a los programas comunicarse con el S. O. y solicitarle servicios.
  - A cada llamada le corresponde un procedimiento:
    - Pone los parámetros de la llamada en un lugar específico para luego ejecutar una instrucción tipo “trap” de llamada a procedimiento protegido para iniciar el S. O.
    - Luego de “trap” el S. O. recupera el control , examina los parámetros y si son válidos ejecuta el trabajo solicitado.
    - Luego de terminar, el S. O. coloca un código de estado en un registro indicando si tuvo éxito o fracaso y ejecuta una instrucción del tipo “return from trap” para regresar el control al procedimiento.
    - El procedimiento regresa al programa llamador con un código de estado como un valor de función; dentro de los parámetros pueden regresar valores adicionales.

## 4.1. Criterios de planificación y rendimiento

- **Utilización del procesador:**
  - ( $t_p$ ) Tiempo promedio, que el procesador esta ocupado:
    - Ejecución de programas.
    - Ejecución S.O.
  - Al aumentar los:
    - $T_e$ : tiempos de espera promedios.
    - Longitudes de la cola promedios.