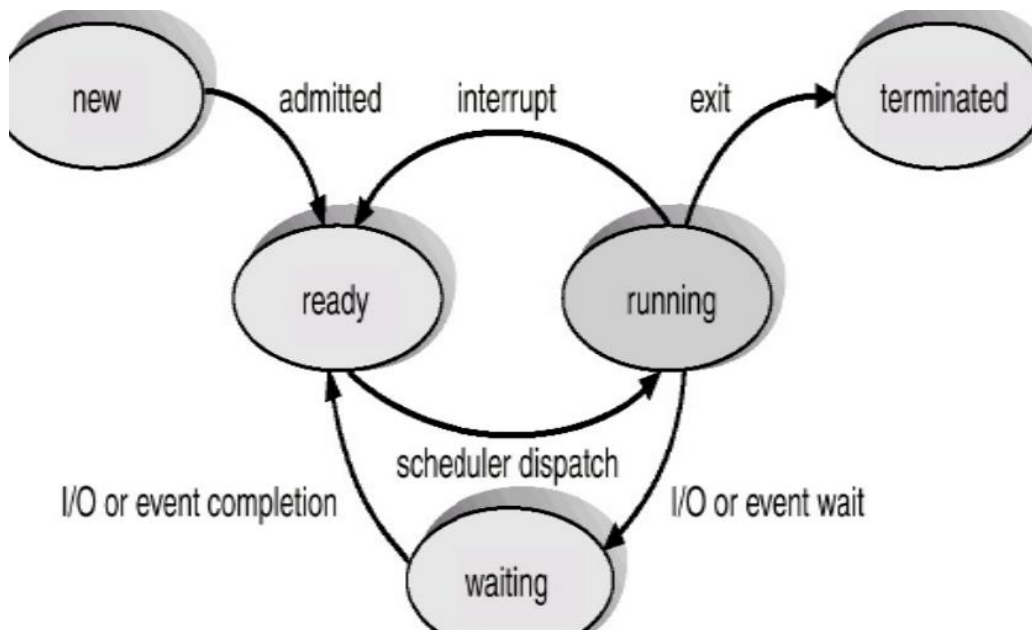


INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS – TEMA 2: PROCESOS

Estados de un proceso:

- Tiene un ciclo de vida por el cuál pasa por:
 - Nacimiento (new).
 - Listo para su ejecución (ready).
 - Ejecutándose (Running).
 - En espera (waiting).
 - Terminado (terminated).



Siempre tiene que haber un proceso, porque ahí está el Kernel.

Colas de planificación de procesos:

- Para realizar la planificación, el SO utiliza el PCB de cada proceso, los relacionan entre sí en función del estado de ese proceso.
- Cuáles procesos están en X ciclo de vida.
- Las PCB se enlazan en Colas siguiendo un orden determinado. Es decir, una PCB tiene un puntero a otra PCB.
- Tenemos varias colas de planificación para cada proceso:
 - Cola que representa que procesos están en espera. Optimización tener varias para saber cual de todos está esperando lo que se consigue.
 - Cola que representa cuáles están en ejecución.
 - Cola que representa que procesos están listos para ejecutarse.
 - Etc.
- Cola de planificación no significa que se maneje como una Cola (estructura de datos).

Módulos de planificación:

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS – TEMA 2: PROCESOS

- Pedazos de software del kernel que realizan distintas tareas asociadas a la planificación.
- Ejecutados ante determinados eventos que lo requieren.
 - Creación / terminación de procesos.
 - Eventos de sincronización o de E/S.
 - Finalización de lapso de tiempo.
 - Etc.
- Principales módulos:
 - Scheduler Long Term:
 - Realiza la actividad de admitir procesos del ciclo new, al estado (ciclo) ready.
 - Ligado al loader: carga el espacio de direcciones de new a ready.
 - Controla el grado de multiprogramación, es decir, la cantidad de procesos en memoria
 - Determina cual pasa.
 - Scheduler Short Term:
 - Planificador que selecciona entre los procesos que están en estado de listo cuál va a tomar la CPU (running, cuál va a correr).
 - Trabaja sobre los ready. Selecciona a quien le toca.
 - Ligado al dispatcher: realiza la tarea de cambio de contexto necesaria para que el proceso pase a ejecutarse.
 - Scheduler Medium Term:
 - Baja nivel de multiprogramación, es decir, saca de RAM un proceso, al área de SWAP (no la que define Linux).
 - Como un S.O es multiprogramado (muchos procesos listos para ejecutarse) → ligado a un grado.
 - En determinadas circunstancias es necesario bajar ese grado de multiprogramación (cant. De procesos listos para ejecutarse).
 - Para la CPU un alto grado de multiprogramación implica más ocupación.
 - Pero para la memoria → puede causar problemas (se verá después).
 - El Scheduler Medium Term baja los procesos de memoria, los que sean necesarios para mantener el equilibrio del sistema.
 - Terminos que asocian:

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS – TEMA 2: PROCESOS

- Swap out (sacar de memoria).
 - Swap in (meter en memoria).
 - Proceso listo, pero no hay memoria, lo suspendo.
 - Los nombres provienen de la frecuencia de ejecución.
 - Otros módulos:
 - Dispatcher: hace cambios de contexto, cambio del modo de ejecución, despacha el proceso elegido en Short Term → salta a la instrucción a ejecutar.
 - Loader: carga en memoria el proceso elegido por el long term.
-
-

ESTADOS:

NEW:

- Estado inicialización.
- Un usuario dispara el proceso. Un proceso se crea por otro proceso: su proceso padre.
- En new se crean las estructuras asociadas al proceso, y el proceso queda en la cola de procesos, esperando ser cargado en memoria.

READY:

- Cuando el “Scheduler Long Term” elige al proceso para cargarlo en memoria, dicho proceso pasa al estado Ready.
- Esta en memoria pero no se esta ejecutando.
- Los procesos compiten para obtener la CPU.
- El proceso ahora solo necesita que se le asigne CPU.
- En cola de procesos listos (ready queue).
- Se selecciona y pasa a running.

RUNNING:

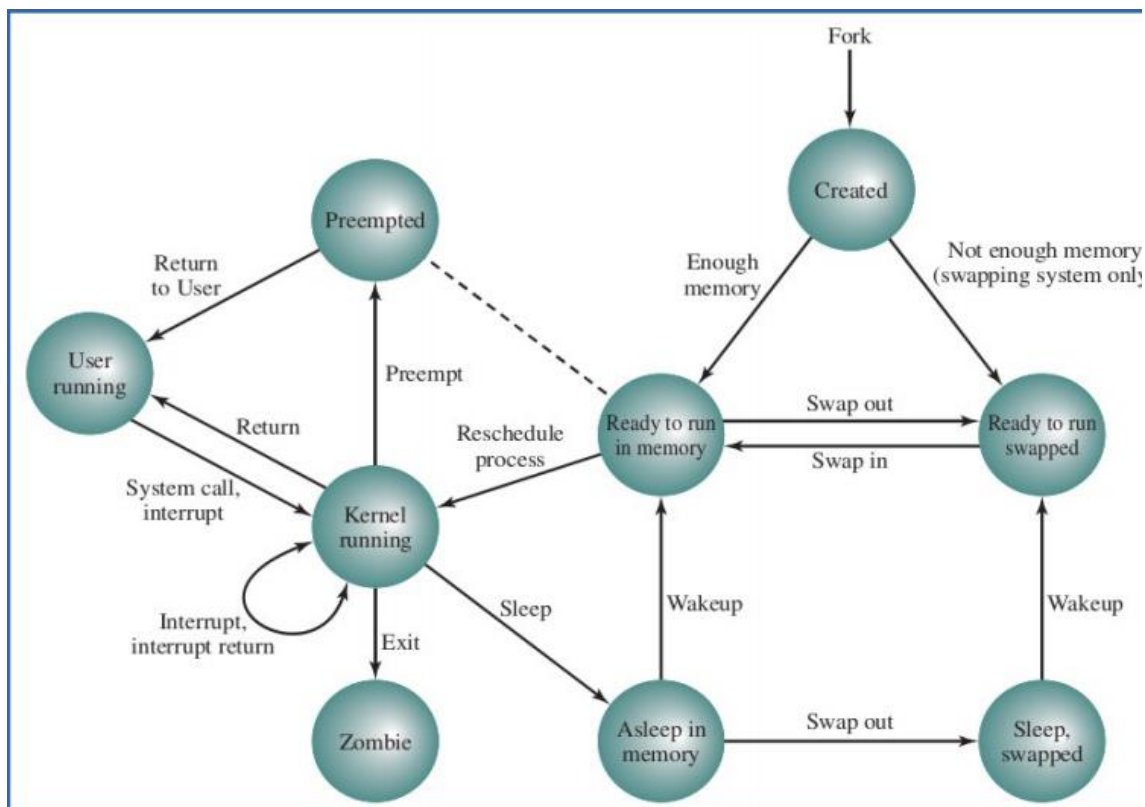
- El “Scheduler Short Term” eligió al proceso para asignarle CPU.
- Hay cambio de contexto.
- Dicho proceso tendrá la CPU a su disposición hasta que se termine el período de tiempo asignado (quantum o time slice), termine el proceso, o hasta que necesite una operación de E/S.

WAITING:

- El proceso necesita que se cumpla un evento esperado para seguir su ejecución.
- Dicho evento puede ser:

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS – TEMA 2: PROCESOS

- Terminación de una E/S requerida.
- Llegada de una señal por parte de otro proceso.
- Sigue en la memoria principal, pero no maneja la CPU.
- Cuando se cumpla el evento que requiere el proceso, dicho proceso pasará al estado READY de nuevo.
 - Transiciones:
 - NEW-READY: carga en memoria del proceso por el “Scheduler Long Term”.
 - READY-RUNNING: asignación de CPU para el proceso por el “Scheduler Short Term”.
 - RUNNING-WAITING: el proceso se pone a dormir, esperando por un evento.
 - WAITING-READY: terminó la espera y compete nuevamente por la CPU.
 - Caso especial, RUNNING-READY:
 - Cuando el proceso termina su quantum sin haber necesitado ser interrumpido por un evento, pasa de nuevo al estado de ready para competir por la CPU, ya que no está esperando ningún evento.
 - Es un caso distinto, el proceso es expulsado de la CPU contra su propia voluntad (se te acabó el tiempo).
 - Situación que se da en algoritmos apropiativos.



INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS – TEMA 2: PROCESOS

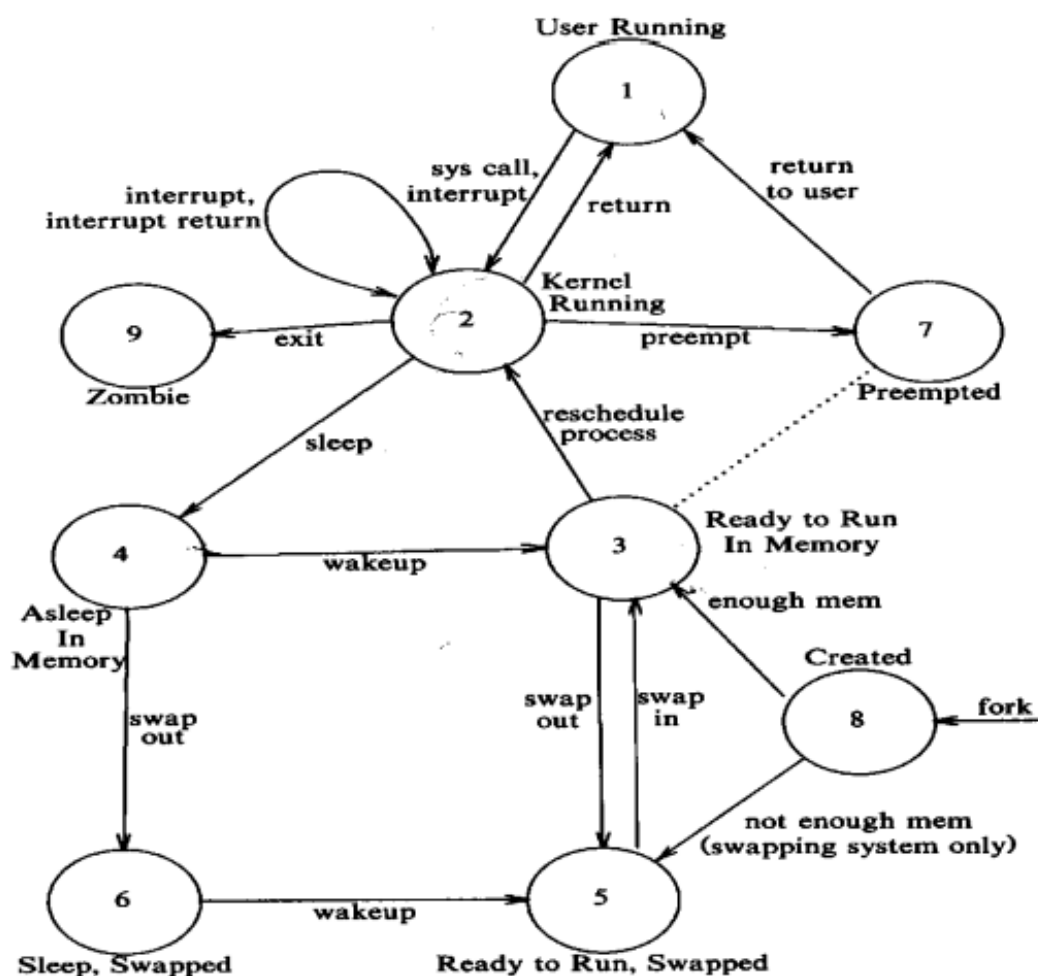
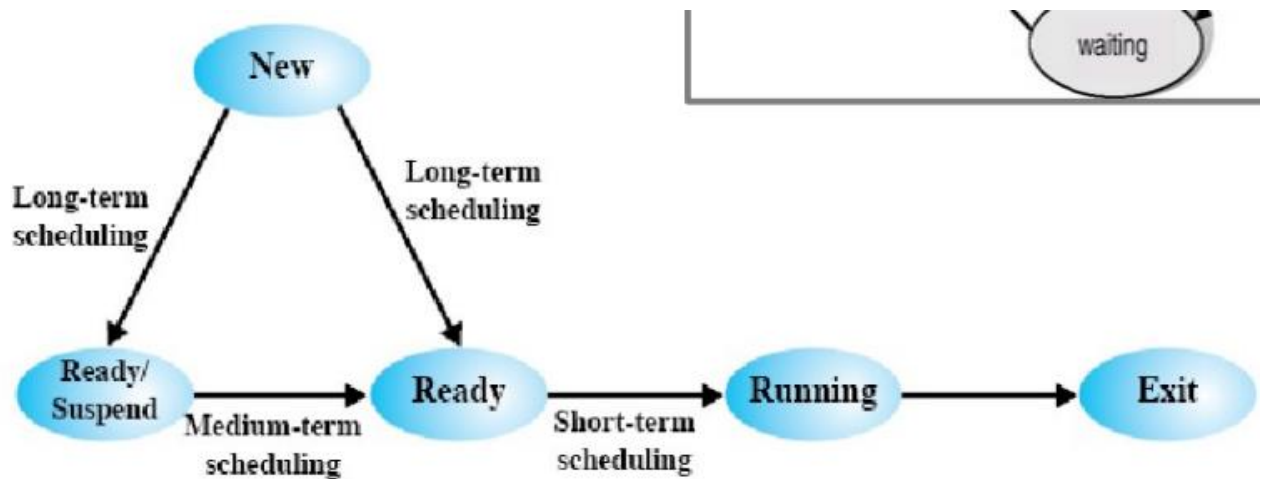


Figure 6.1. Process State Transition Diagram

1. Ejecución en modo usuario
2. Ejecución en modo kernel
3. El proceso está listo para ser ejecutado cuando sea elegido.

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS – TEMA 2: PROCESOS

4. Proceso en espera en memoria principal.
5. Proceso listo, pero el swapper debe llevar al proceso a memoria ppal antes que el kernel lo pueda elegir para ejecutar
6. Proceso en espera en memoria secundaria.
7. Proceso retornando desde el modo kernel al user. Pero el kernel se apropia, hace un context switch para darle la CPU a otro proceso.
8. Proceso recientemente creado y en transición: existe, pero aun no está listo para ejecutar, ni está dormido.
9. El proceso ejecutó la system call exit y está en estado zombie. Ya no existe más, pero se registran datos sobre su uso, codigo resultante del exit. Es el estado final.

