

MONITORING AND MANAGING LINUX PROCESSES

- Un proceso (unidad activa) es un programa (unidad pasiva) que se está ejecutando
- Para ser ejecutado necesita recursos:
CPU, memoria, dispositivos periféricos,...
- El SO tiene que:
crear, ejecutar y cancelar un proceso
dar mecanismos de comunicación y
sincronización entre los procesos

- A cada proceso se le asocia una estructura de datos Process Control Block (PCB)
- Los datos usados para describir el proceso dependen del SO y de la estructura de la máquina física
- Las informaciones contenidas en el PCB son de vital importancia para el SO: tienen que estar en memoria dentro de un área solo accesible del núcleo del sistema operativo

Información asociada a cada proceso

- Nombre (número) del proceso
- Contexto del proceso: registros de la CPU, estado del proceso, direcciones de las tablas de paginación del proceso
- Información sobre el scheduling de la CPU
- Información sobre la gestión de la memoria
- Información de los recursos asociados: tiempo de uso de CPU, tiempo transcurrido desde que empezó a ejecutarse,...
- Información sobre el I/O: dispositivos asociados al proceso, files abiertos,...

JERARQUÍA EN LOS PROCESOS

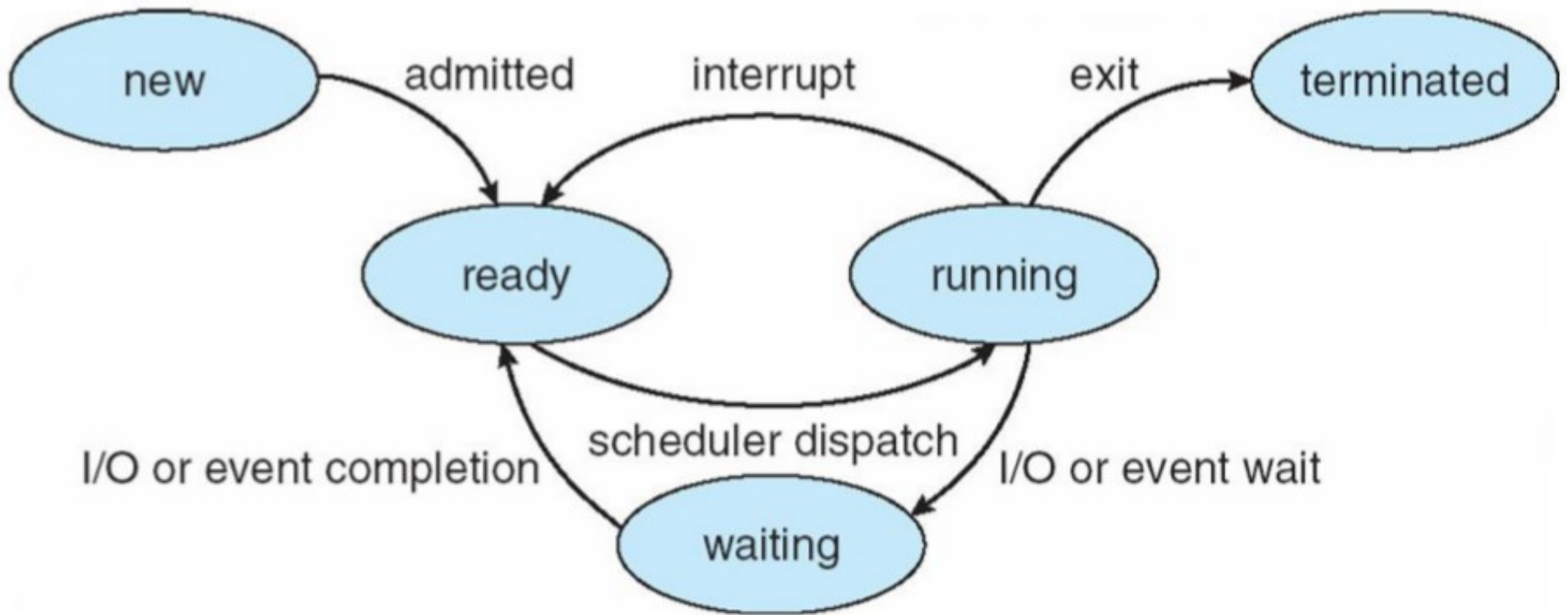
En los sistemas operativos de la familia Unix la creación de un proceso nuevo se realiza principalmente a través de la llamada al sistema `fork()` ==> si la llamada tiene éxito da dos diferentes valores:

- al proceso que ha invocado `fork()` se le da el PID del proceso hijo apenas creado
- al proceso hijo apenas creado se le da el valor 0 (código de error que significa que la llamada se ha ejecutado bien)

- El proceso hijo creado por medio de la llamada `fork()` es un “duplicado” del proceso padre:
 - mismas instrucciones de máquina
 - mismo estado de memoria
 - mismos file abiertos, ...
- El hijo y el padre se diferencian por algunos detalles:
 - los PID son diferentes
 - el hijo memoriza en su PCB el PID del padre
 - el uso de recursos por parte del hijo parte desde cero
 - el hijo no hereda algunos mecanismos de comunicaciones asíncronas de eventos

ESTADO DE LOS PROCESOS:

- **New**: Corresponde a la creación de un nuevo proceso (carga del código de disco a RAM)
- La transición de new a **ready** se da cuando el SO (scheduler) admite el nuevo proceso en la ready queue
- La transición de ready a **running** se da cuando el proceso viene elegido (dispatcher) entre todos los procesos ready para ser ejecutado
- La transición de running a **waiting** se da por un pedido I/O al SO o por la espera de un evento
- La transición de waiting a ready se da cuando se verifica el evento
- El estado **terminated** se da por una llamada al SO para indicar el fin de la actividad, un final anormal o una ejecución de instrucciones no consentida o no válidas



[root@fedora21 ~]# grep 'physical id' /proc/cpuinfo ---> solo una

physical id : 0

physical id : 0

physical id : 0

physical id : 0

physical id : 0

physical id : 0

physical id : 0

physical id : 0

[root@fedora21 ~]# grep 'core id' /proc/cpuinfo ---> n. cores de la cpu física, cada core 2 threads

core id : 0

core id : 0

core id : 1

core id : 1

core id : 2

core id : 2

core id : 3

core id : 3

[root@fedora21 ~]# grep 'processor' /proc/cpuinfo → cpus lógicas

processor : 0

processor : 1

processor : 2

processor : 3

processor : 4

processor : 5

processor : 6

processor : 7