

- 1- Un proceso de sistema es aquel creado por el sistema operativo para ejecutar el código del sistema periódicamente, o bien cuando se desea proveer de algún servicio (proceso demonio). Por otro lado, el proceso de usuario es aquel que es creado para ejecutar instrucciones y tareas requeridas por el usuario.
- 2- Porque es el que requiere de todos los recursos de un procesador para un determinado periodo de tiempo.
- 3- `fork()`: Permite la creación de un proceso hijo.
  - No recibe parámetros esta función.
  - Se le denomina proceso hijo al nuevo proceso creado, y el proceso que hizo la llamada al sistema es el proceso padre.
  - El proceso hijo es un duplicado del proceso padre, por lo que tendrán varios atributos similares, los cuales pueden ser modificados posteriormente en la ejecución.
  - En caso de éxito, retorna el PID (Process ID) del hijo al padre, y en el hijo se retorna 0. En caso de fallo, retorna -1 al padre y no se crea ningún proceso hijo.

`CreateProcess()`: Permite la creación de un nuevo proceso y su hilo principal. El nuevo proceso ejecuta el archivo especificado.

- Recibe los siguientes parámetros: `lpApplicationName`, `lpCommandLine`, `lpProcessAttributes`, `lpThreadAttributes`, `bInheritHandles`, `dwCreationFlags`, `lpEnvironment`, `lpCurrentDirectory`, `lpStartupInfo`, `lpProcessInformation`.
- De estos parámetros, todos pueden ser nulos a excepción de `bInheritHandles` (TRUE o FALSE), `dwCreationFlags`, `lpStartupInfo` y `lpProcessInformation`.
- Todos estos parámetros permiten configurar al nuevo proceso, en cuestiones como seguridad, prioridad de ejecución, entorno de ejecución, etc. En los parámetros con valor NULL, entonces se adoptarán configuraciones iguales a las del proceso que realizó la llamada del sistema.
- En caso de éxito, retorna un valor diferente de cero. En caso de error, retorna un valor cero.

### Bibliografía ejercicio 3

- [man7.org/linux/man-pages/man2/fork2.html](http://man7.org/linux/man-pages/man2/fork2.html) consultado el 17/marzo/2021 a las 14:49.
- [cs.nyu.edu/courses/Fall01/os/CreateProcess.html](http://cs.nyu.edu/courses/Fall01/os/CreateProcess.html) consultado el 17/marzo/2021 a las 17:27.



4. En el diagrama se puede observar la estructura básica en la que el planificador de procesos realiza la organización para que el CPU ejecute diversos procesos. Primero, todos los procesos que están listos para ser ejecutados se enfilan en la ready queue, con la finalidad de ir siendo ejecutados por el CPU.

Una vez que el CPU está atendiendo un proceso, este sale del ready queue y se encuentra en ejecución hasta que suceda algo. Dependiendo de ese suceso, el proceso se canalizará a algún otro de los queues que aparecen debajo del CPU.

- I/O request, I/O queue e I/O: Aquí se canalizará al proceso cuando este requiere de alguna salida u entrada de datos externa. Una vez que la entrada/salida es proporcionada, el proceso vuelve a enfilarse en la ready queue para esperar a su ejecución en CPU.
- Time slice expired: Aquí se canalizará al proceso cuando este exceda su tiempo límite de ejecución en CPU. Posteriormente el proceso vuelve a enfilarse en el ready queue.
- Fork a child: Aquí se canalizará al proceso cuando este quiera crear otro proceso (subproceso hijo). Una vez atendida esta petición, se tiene child executes y finalmente se enfilan estos procesos en el ready queue.
- Wait for an interrupt: Aquí se canalizará al proceso cuando este defenga su ejecución porque está esperando una secuencia de interrupción. Una vez que suceda la interrupción, se tiene interrupt occurs y finalmente se enfila este proceso en el ready queue.

En caso de que el proceso logre terminar durante su ejecución del CPU, entonces este sale del diagrama, que sería lo equivalente al planificador de procesos.

5. Un proceso conformado por varios hilos compartirá elementos en común con los hilos como el código del programa, la información y los archivos o recursos que se utilicen.

En cada hilo existirán elementos independientes, es decir, dependen de cada hilo. Estos son los registros y el stack, en donde se lleva la secuencia de instrucciones.