

# Procesos

## Práctica 1

Escribir un programa que cree una instancia child con la syscall `fork()`.

Luego de `fork()`, la instancia padre ejecuta un `sleep` de 20 segundos y luego termina. La instancia hijo ejecuta un `sleep` de 40 segundos y luego termina.

Para compilar y probar trabajar con dos instancias de consolas, una en cada mitad horizontal de la pantalla. En una consola lanzar el programa y en la otra visualizar el estado de los procesos (usar **`ps -elf | grep [proceso]`**) y observar el estado de cada instancia.

¿Qué pasa con el proceso hijo cuando termina el proceso padre?

¿Terminan normalmente?

**Nota 1:** Incluir `<unistd.h>` para utilizar la función `sleep()`.

**Nota 2:** Puede utilizarse el comando `watch` para ejecutar periódicamente el comando `ps`. Ejemplo:  
**`watch -n 1 "ps -elf | grep [proceso]"`**

**Nota 3:** Usar el nombre del proceso “proceso” y no “./proceso”. Ya que al ser zombie el comando `ps` lo muestra sin el “./”

## Práctica 2

Editar el código del programa anterior e invertir los tiempos utilizados en `sleep()` entre las instancias padre e hijo. Compilar y ejecutar.

¿Cómo es el comportamiento en este caso del proceso hijo? ¿Termina el proceso normalmente?

¿Qué ocurre cuando termina su ejecución el proceso padre?