

# **SOPER: Memoria práctica 2**

Darío Adrián Hernández Barroso  
Ángel Manuel Martín Canto

2016-03-18

## Hilos

### Ejercicio 3: Comparación de velocidad entre procesos e hilos

Creamos dos programas(ejercicio3a y ejercicio3b) que usan, respectivamente, procesos e hilos. Los imprimen por pantalla en paralelo 100 números aleatorios.

El resultado de la ejecución en varias maquinas es que la versión con procesos es ligeramente más rápida que la versión con hilos. En un principio esto nos resulto sorprendente dado que los procesos suponen más sobrecarga de memoria. Las explicaciones que hemos encontrado son:

- El ejercicio no hace un uso intensivo de memoria
- Midiendo el tiempo con el comando de shell “time” hemos descubierto que la utilización de CPU de la versión con procesos ronda el 400 %(esto es, usa 4 nucleos) mientras que la de hilos esta en el 200 %. Estas medidas se han realizado en una maquina con kernel Linux 3.16 y procesador Intel i7 de 8 núcleos lógicos. Es posible que el planificador del sistema operativo trate distinto a hilos y procesos, especialmente cuando tienen una duración tan reducida.

### Ejercicio 4: Paso de parámetros en hilos

La estrategia que hemos seguido en este ejercicio es crear una estructura con todo la información relevante para cada hilo y pasar punteros a esa estructura como argumento a los hilos.

## Señales

### Ejercicio 6: Terminación mediante señales

Poco hay que reseñar de este ejercicio excepto que no hace falta proveer de un manejador específico para la señal de terminación.

### Ejercicio 8: Envío de señales por un árbol de procesos

En este ejercicio usamos las señales SIGUSR1 y SIGUSR2 para las que proveemos manejadores propios. Hemos intentado separar claramente el código del padre raíz, de los hijos intermedios y del último hijo. También hemos comprobado que no se generen procesos huérfanos. Una ejecución normal del programa da la siguiente salida:

```
Ultimo hijo envia SIGUSR1
Hijo@5244, recibe SIGUSR1
Hijo@5243, recibe SIGUSR1
Hijo@5242, recibe SIGUSR1
Hijo@5241, recibe SIGUSR1
Padre raiz recibe SIGUSR1
Padre raiz envia SIGUSR2
Hijo@5241, recibe SIGUSR2
Hijo@5242, recibe SIGUSR2
Hijo@5243, recibe SIGUSR2
Hijo@5244, recibe SIGUSR2
Ultimo hijo recibe SIGUSR2
Ultimo hijo envia SIGUSR2
Padre raiz recibe SIGUSR2
Hijo@5241, recibe SIGTERM
Hijo@5242, recibe SIGTERM
Hijo@5243, recibe SIGTERM
```

```
Hijo@5244 , recibe SIGTERM
Ultimo hijo recibe SIGTERM
```

### Ejercicio 10:

Este ejercicio combina técnicas de todos los anteriores. Hacemos notar que a la hora de realizar cualquier espera usamos la señal de alarma en vez de la función sleep. Una salida normal del programa es:

```
Creando proceso
Leyendo LEE
Leyendo HASTA
Leyendo QUE
Leyendo PROCESO
Leyendo UN
Leyendo UN
Leyendo CADENA
Leyendo LEE
Leyendo FICHERO
Leyendo LEE
Leyendo EN
Leyendo PROCESO
Leyendo ESCRIBE
Leyendo QUE
Leyendo ESCRIBE
Leyendo CADENA
Leyendo PROCESO
Leyendo PROCESO
Leyendo FIN
Recreando proceso
Leyendo HASTA
Leyendo UN
Leyendo A
Leyendo PROCESO
Leyendo ESCRIBE
Leyendo HASTA
Leyendo EL
Leyendo LEE
Leyendo EL
Leyendo PROCESO
Leyendo ESCRIBE
Leyendo HASTA
Leyendo LEE
Leyendo UN
Leyendo FIN
Recreando proceso
Leyendo A
Leyendo A
Leyendo PROCESO
Leyendo LA
Leyendo FICHERO
Leyendo FIN
Recreando proceso
```