**P3 SISTEMAS OPERATIVOS**

Práctica 3: Servicios POSIX para la  
gestión de hilos en Linux.

Marcos Bonilla de las Morenas

Miguel de Sande Moreno

1. **Como aplicación de los servicios POSIX básicos de gestión de hilos, complete en el programa las líneas que a continuación se indican:**

**1.1 Línea 53 indicada mediante el comentario CODIGO 1. Inicialice la estructura de cada coche e invoque el servicio POSIX que considere adecuado.**

Coches[i].id = i;  
 Coches[i].cadena = "Coche";

if (pthread\_create(&hilosCoches[i], NULL,  
 (void \*(\*)(void \*))funcion\_coche,  
 (void \*)&Coches[i]) != 0) {

perror("Error creando hilo");  
 exit(EXIT\_FAILURE);

}

Este código se ejecuta tantas veces como N\_COCHES valga, ya que está dentro de un bucle for. Por cada iteración del mismo se definirá un nuevo coche, asignándole como id lo que valga ien ese momento, y en cadena se le asigna “Coche”. Después se crea un hilo para ese coche, que ejecutará la función funcion\_coche(), con el servicio. pthread\_create().

**1.2 Línea 38 indicada mediante el comentario CODIGO 2. Invoque el servicio POSIX que considere adecuado.**

pthread\_exit(NULL);

Simplemente se necesita que el hilo termine cuando acaba de ejecutar la función, por lo que se ejecuta este servicio POSIX al final de la misma.

**1.3 Línea 63 indicada mediante el comentario CODIGO 3. Invoque el servicio POSIX que considere adecuado.**

if (pthread\_join(hilosCoches[i], NULL) != 0) {

perror("Error esperando hilo");  
 exit(EXIT\_FAILURE);

}

Este código hace que el programa principal espere a que todos los coches hayan terminado (se haya ejecutado pthread\_exit(NULL); en todos los hilos) Está dentro de un bucle for, por cada iteración se ejecuta el servicio pthread\_join() para cada hilo, estando sus identificadores guardados en el array hilosCoches[].

**2. ¿Qué efecto produce la función rand\_r() en la función funcion\_coche() que ejecuta cada hilo?**

rand\_r(), en función de una semilla que se le pase como parámetro, genera un número aleatorio. En el código, al resultado de rand\_r() se le aplica %10 para obtener un número aleatorio entre 0 y 9. Este número se guarda en una variable aleatorio, para después utilizarla como parámetro en la función sleep(aleatorio). De esta manera cada coche tardará un tiempo diferente y aleatorio en llegar a la meta.

**3. Qué ocurre si el hilo inicial (que ejecuta la función main()) no espera la finalización del resto de hilos?**

Si no espera al resto de hilos, además de imprimir incorrectamente que todos los coches han llegado a la meta, el proceso termina. Cuando el hilo principal finaliza, el proceso completo termina y el sistema operativo limpia todos los recursos dedicados a él. Como no son independientes entre ellos como los procesos pesados, sino que comparten todos los recursos, los hilos de los coches también terminan en medio de su ejecución.