

Arquitectura de Sistemas

Práctica 7: Cooperación entre hebras

Gustavo Romero López

Actualizado: 2 de mayo de 2018

Arquitectura y Tecnología de Computadores

Objetivos

- ⊙ Supondremos al alumno familiarizado con las **hebras tipo POSIX o pthreads**.
- ⊙ Vamos conocer el nuevo modelo de hebras de **C++11**.
- ⊙ Como material de referencia podemos consultar las páginas **cppreference.com** y **cplusplus.com**.
- ⊙ Descubriremos su funcionamiento mediante una serie de ejemplos.
- ⊙ Al final el alumno debe crear un programa para comprobar los conocimientos adquiridos.
- ⊙ Hebras C++11 explicadas para usuarios de pthreads.

- ⦿ Clase que representa una hebra de ejecución.
- ⦿ Miembros:
 - tipos: native_handle_type: tipo subyacente (pthread).
 - clases: id: representa la identificación de una hebra.
 - funciones:
 - (constructor)
 - (destructor)
 - operator=
 - joinable
 - get_id
 - native_handle
 - hardware_concurrency
 - join
 - detach
 - swap

```
#include <iostream>
#include <thread>

void hola() { std::cout << "hola"; }

int main()
{
    std::thread t1(hola);
    t1.join();

    std::thread t2([]{ std::cout << " mundo!\n"; })
        ;
    t2.join();
}
```

- ⊙ función lambda = función anónima.
- ⊙ declaración completa: `[capture-list] (params)mutable (optional)exception attribute -> ret { body }`
- ⊙ uso habitual: `[] { cuerpo }`
- ⊙ documentación de **funciones lambda**.
- ⊙ ejemplo:

```
int main()
{
    auto f = []{ cout << "hola , mundo!\n"; };
    f();
}
```

Trabajo a realizar

- ⊙ Estudie la tres formas de implementar un servidor web mostradas al final del **tema 7, activación**.
 - Basada en **procesos**.
 - Basada en **hebras**.
 - Basada en un **conjunto de hebras**.
- ⊙ La misma idea se ha aplicado en estos 4 ejemplos:
 - **server0.cc** (1 proceso/secuencial).
 - **server1.cc** (N procesos/paralelo).
 - **server2.cc** (nº óptimo de procesos /paralelo).
 - **server3.cc** (N hebras/paralelo).
 - **server4.cc** (nº óptimo de hebras/paralelo).
- ⊙ Existen 3 cargas de trabajo, ficheros `work.h`, cada una de las cuales ejercita el uso del procesador o de la E/S.

Trabajo a realizar

- ⊙ Dado que no puede modificar las implementaciones, intente explicar la diferencia de rendimiento entre ellas.
- ⊙ En la práctica anterior nos centramos en estudiar el rendimiento con `perf`.
- ⊙ Ahora vamos a intentar buscar otro tipo de cuellos de botella que puedan ralentizar la ejecución de los programas de prueba:
 - Llamadas al sistema: `strace`
 - Funciones de biblioteca: `ltrace`
- ⊙ ¿Los programas de prueba funcionan bien?
- ⊙ ¿Cuál cree que es mejor?
- ⊙ Proponga usted una nueva carga de trabajo y verifique qué modelo de procesamiento se adapta mejor a la misma.