

PRÁCTICA 1: Introducción al curso

Javier Masero Delgado y Celia Flores Alcón

Estudio y comentarios sobre el archivo timer.h

Vamos a comentar la implementación mediante un timer sin el uso de Qt. Se ha utilizado la librería std de C++ en relación con la clase std::thread.

En la primera imagen podemos observar varias funciones con la librería std:

```
template <class callable>
void connect(callable&& f)
{
    std::thread([=] ()
    {
        while(true)
        {
            if(go.load())
                std::invoke(f);
            std::this_thread::sleep_for(std::chrono::milliseconds(period.load()));
        }
    }).detach();
};
```

- **std::invoke(f):** Esta función utiliza una llamada para un objeto, en este caso f (*callable*).
- **std::this_thread::sleep_for(...):** Se encarga de bloquear la secuencia de ejecución durante una duración especificada. Es similar a la función *sleep*.
- **detach():** Igual que remove, pero mantiene los datos jQuery asociandos con elementos eliminados.

En esta segunda imagen, seguimos observando varias funciones de la librería:

```
void start(int p)
{
    period.store(p);
    go.store(true);
};

void stop() { go.store(!go); };
void setPeriod(int p) { period.store(p); };

private:
    std::atomic_bool go = false;
    std::atomic_int period = 0;
```

- **std::atomic_bool go = false:** Se utiliza esta función para el correcto acceso a la variable go, ya que al utilizar threads, se pueden originar problemas de concurrencia. Así nos aseguramos de que la variable tiene un estado atómico.
- **std::atomic_int period = 0:** Al igual que la función anterior, se utiliza para la atomicidad de la variable con relación al uso de los threads.

Además de la utilización de la librería *std* en la clase *timer.h*, también nos encontramos en el archivo *ejemplo.cpp* la siguiente función:

```
ejemplo1::ejemplo1() : Ui_Counter()  
{  
    setupUi(this);  
    show();  
    connect(button, SIGNAL(clicked()), this, SLOT(doButton()));  
  
    mytimer.connect(std::bind(&ejemplo1::cuenta, this));  
    mytimer.start(500);  
}
```

- **mytimer.connect(std::bind(&ejemplo1::cuenta, this))**: Se trata de una plantilla de función que genera un contenedor de desvío de llamadas, es decir, equivale a invocar una función con argumentos asociados a args.