**Ejemplo de hilos en c++ - Christian Gallego Chaverra**

#include <iostream>

#include <thread> //Librería para el manejo de los hilos en c++

using namespace std;

void function\_1(){

cout<<"Beuty is only skin-deep"<<endl;

}

int main()

{

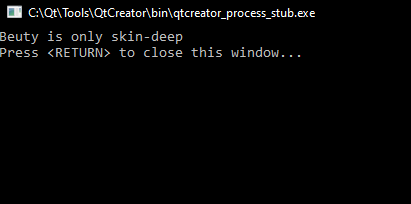
thread t1(function\_1);//t1 empieza su ejecucion

t1.join(); //el hilo principal espera a que t1 finalice

return 0;

}

En este caso hay dos hilos en ejecución, el main thread y t1. El hilo t1 ejecuta la función 1 e imprime el mensaje en pantalla y luego finaliza el programa.



Pero ahora probemos con otra instrucción

#include <iostream>

#include <thread>

using namespace std;

void function\_1(){

cout<<"Beuty is only skin-deep"<<endl;

}

int main(){

thread t1(function\_1);//t1 empieza su ejecucion

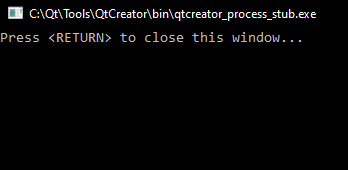
//t1.join(); //el hilo principal espera a que t1 finalice

t1.detach();// t1 se ejecutara libre por su propia cuenta (daemon process)

return 0;

}

Al poner t1.detach(), el hilo t1 se desliga del hilo principal y se ejecuta por su cuenta.



Al ejecutar el programa no aparece el mensaje de la función debido a que el hilo principal termina su ejecución antes de que t1 pueda imprimir el mensaje. Cuando se tienen hilos corriendo por su cuenta en un programa, no siempre se tiene seguro cual se ejecutará primero; a no ser que exista algún tipo de sincronización para la ejecución de los hilos. En este caso en particular, el hilo principal crea t1, pero para crearlo igual necesita procesamiento, y como este esta por su cuenta, el programa finaliza antes de que se ejecute.