Exercise 1 - Creating Posix Threads

In this exercise there will be written a program that creates two threads. The threads must be given an ID, which they'll print to stdout, every second, along with the amount of times the thread has been printed. When each thread has been written 10 times, they will terminate.

Implementation

```
//Exercise 1 - Create two threads

#include <iostream>
#include <pthread.h>
#include <unistd.h>

using namespace std;

// stuct data
struct threadData{
    int threadId_;
};

void* print_message(void* data) {            //voidpointer = Trashcan
    threadData *tdata = static_cast<threadData *>(data);
    for(unsigned int i = 0;i < 10;i++){          // Write 10 times each
        std::cout << "ThreadID: "<< tdata->threadId_ << " Count: " << i<< std::endl;

        sleep(1); // sleep 1 sec
    }
    return nullptr;
}
int main() {
    // create two stuck instances
    threadData *th1,*th2 = new threadData;
    // assign id's
    th1->threadId_ = 0;        //First thread ID
    th2->threadId_ = 1;        //Second thread ID

    // define threads
    pthread_t t1,t2;

    // create thread instance
    pthread_create(&t1, NULL, &print_message,  (void *)th1);
    pthread_create(&t2, NULL, &print_message,  (void *)th2);
    // join threads to parent.
    pthread_join(t1,nullptr);
    pthread_join(t2,nullptr);

    return 0;
}
```

Results

```
stud@stud-virtual-machine:~/Desktop/ISU/Threads$ g++ threads.cpp -o ex1thread -lpthread
stud@stud-virtual-machine:~/Desktop/ISU/Threads$ ls
bin  build  ex1thread  Makefile  threads.cpp
stud@stud-virtual-machine:~/Desktop/ISU/Threads$ ./ex1thread
ThreadID: ThreadID: 1 Count: 0 Count: 00

ThreadID: 1 Count: 1
ThreadID: 0 Count: 1
ThreadID: 1 Count: 2
ThreadID: 0 Count: 2
ThreadID: 1 Count: 3
ThreadID: 0 Count: 3
ThreadID: 1 Count: 4
ThreadID: 0 Count: 4
ThreadID: 1 Count: 5
ThreadID: 0 Count: 5
ThreadID: 1 Count: 6
ThreadID: 0 Count: 6
ThreadID: 1 Count: 7
ThreadID: 0 Count: 7
ThreadID: 1 Count: 8
ThreadID: 0 Count: 8
ThreadID: 1 Count: 9
ThreadID: 0 Count: 9
stud@stud-virtual-machine:~/Desktop/ISU/Threads$
```

When both threads hit a count of 10, the program terminates.

Discussion

What happens if function main() returns immediately after creating the threads? Why?
If the main function returns immediately after only creating the threads, but not joining them, the terminal will return 0 right away. That means the threads wont be executed before the program finishes.

The seemingly easy task of passing the ID to the thread may present a challenge; In your chosen solution what have you done? Have you used a pointer or a scalar?

For this solution a pointer is used. This solution was the most obvious approach since function is a void pointer.

Filer

threads1.png	60,2 KB	2018-09-12	Mie Nielsen