

Asynchronous Task Library exercise for Pix4D

Author: Eduardo Paz Orduña (edupaz2@gmail.com)

Description:

The exercise has two components:

1. Write a C++ library that allows the user to manage an arbitrary number of asynchronous tasks.

The library should allow users to schedule tasks to be run asynchronously and pause, resume and stop them. The library should also allow the user to query the status of a given task.

2. Write an example command line application that allows the user to launch and control tasks. The tasks run by the example can be hard-wired.

The library needs to be able to launch an arbitrary number of asynchronous tasks. For the sake of simplicity, due to time constraints, I will take the following assumptions:

- The tasks don't need any type of synchronization between themselves in any way (e.g, access to same file).
- The job done will be, initially, the same for all of the tasks, but the library is built to support different types of tasks in the future.
- Tasks will run concurrently if the machine allows it.
- Tasks will need some sort of priority mechanism to avoid inanition.

My initial thought about this library was using the traditional approach of threads. As I do some research I found out about the concept of fibers, known as user-land threads.

The idea behind the fibers is that the thread context switch is very expensive and under kernel terms (preemptive), so the fibers brings speed and controlled switching to the table of concurrent tasks.

A typical approach will be one thread per CPU, and a pool of fibers for each thread. The context switch for the fibers are three orders of magnitude faster than the threads', plus there is no need for mutual-lock mechanisms as one and only one fiber is executing in the thread at the same time (controlled switch).¹

Starting from this idea, I checked Boost::Fiber will be the candidate to help me implementing the library.

The code for this exercise is available at my **Asynctasklib**² project in github.

¹ "Parallelizing the Naughty Dog engine using fibers" @
"http://twvideo01.ubm-us.net/o1/vault/gdc2015/presentations/Gyrling_Christian_Parallelizing_The_Naughty.pdf"

² <https://github.com/edupaz2/asynctasklib>

This first version v0.1 has the following features:

- Command line application reading commands from the user (stdin).
- Shared library application allowing:
 - Launching an arbitrary number of tasks, all of them doing the same work.
 - Start, Pause, Resume and Stop tasks are allowed.
 - Status of all the tasks is allowed.
 - All tasks are launched in one single worker thread.
 - Tasks priority mechanism is working.

Future improvements not included in this release:

- Launch several worker threads to maximize CPU use.
- Implement a work-stealing scheduling system between the threads to maximize CPU use.
- Unit tests.
- Different types of tasks.
- Include % of task done.
- GUI Application with same functionality as the Command line app.
- Python bindings using Boost::Python to allow Python apps use the library.