Conector Java-C++

P1 - Raúl Nozal

He preparado este conector base sobre el que tenéis que realizar unas ligeras modificaciones para dar soporte a la tarea opcional "Conector Java-C" de la Práctica 1.

Os proporciono varios ficheros:

- `Makefile`: a modo de ejecutor de tareas principales.
- `p1.java`: main principal de Java. Incluye el conector base a nivel de Java [modificar].
- `p1bridge.cpp`: Conector base a nivel C++ [modificar].

Probablemente necesitéis el JDK de Java, pero con el `openjdk` (Linux) os valdrá.

Lo he probado en mi sistema (Linux) y funciona correctamente. Lo que os proporciono compila y ejecuta sin generar ningún problema:

- `make` o `make rebuild`: construirá el mock para JNI (header de C/C++), compilará a bytecode el código Java y a librería dinámica el código C++.
- `make run`: ejecutará el main de la clase P1 de Java, quien usa el Bridge entre Java y C para ejecutar un código de ejemplo (mensaje TODO) de C++.

Os facilito los únicos cambios que tenéis que hacer sobre el conector:

- Java procesará los argumentos que antes procesaba el código C.
- Java debe enviar a C sólo 3 parámetros: longitud del array (entero), operación (cadena de texto) y número de hilos (entero), entendiéndose éste último como single-thread cuando es 1, multi-thread cuando es entre 2 y 10. El código C deberá consumir esos 3 parámetros y ejecutar el programa.
- Java debe recibir de C el valor final de la operación (suma o xor) e imprimirlo por pantalla.

La comparación entre Java y C debe ser justa, y ya que queremos medir overheads, deberemos obtener los tiempos de todo el programa, incluyendo el propio lanzamiento y finalización (puedes usar `time` <`program`> o `/usr/bin/time` <`program`>).

Modo de actuación:

- Extrae el código común de C que va a ser utilizado tanto desde el frontend de C como el frontend de Java.
- Proporciona una misma función de C (firma: parámetros y retorno) que será llamada por ambos frontend. La función hará: reservar el espacio del array e inicializarlo, generar/unir hilos, computar sum/xor en función del tipo de operación y devolver el valor. No es necesario que incluyas el resto de funcionalidades (registro, logger, etc) durante la sección "Conector" y su medición de overheads.
- Cada frontend mostrará por pantalla el valor resultante y finalizarán el programa.