
Taller de Programación Paralela

Departamento de Ingeniería en Informática

Lab 4: Concurrencia y Orientación a Objetos

1 Objetivos

El objetivo de este laboratorio es experimentar con tareas y monitores de *uC++*, implementando un reconocedor paralelo de expresiones regulares.

2 Descripción

En este laboratorio, usted implementará un programa paralelo en *uC++* que determine si un conjunto de strings pertenece o no una expresión regular. El reconocedor debe estructurarse de acuerdo a las siguientes indicaciones:

1. Una tarea *productora* se encargará de leer de un archivo los strings y depositarlos en un buffer compartido. El largo del buffer será indicado con opción en línea de comando. Se deposita un string a la vez.
2. Un número de tareas *reconocedoras* leerán un string a la vez del buffer compartido y ejecutarán el algoritmo reconocedor.
3. Cada tarea reconocedora depositará el string en un segundo buffer compartido, con alguna indicación de pertenencia a la expresión regular.
4. Una tarea *escritora* leerá un string a la vez, del segundo buffer compartido y escribirá el string y su condición de pertenencia, en el archivo de salida.
5. La tarea del `uMain::main` debe crear a todas las otras tareas y esperar su término.

El comando de ejecución del programa debe ser el siguiente

```
$ ./exp -i in.txt -o out.txt -h número_de_tareas -L largo_buffer1 -l largo_buffer2
```

El archivo de entrada contendrá un string por línea. El número de tareas indicado es el número de tareas reconocedoras. Las opciones `-L` y `-l` son opcionales e indican los largos de los buffers. Estos largos se refieren al número máximo de strings que pueden almacenar, no al número máximo de caracteres. Por defecto, los buffers tendrán largo 10. Ambos buffers deben ser implementados como monitores.

3 Condición de término

Usted debe proponer e implementar una estrategia para la finalización limpia de las tareas. Es decir, debe introducir un mecanismo que permita a las tareas reconocedoras saber cuándo no hay más strings por procesar y a la tarea escritora cuándo no hay más strings por escribir.

4 El reconocedor

La expresión regular a reconocer es:

$$(A + C + G + T)^*GT^+CT^*(A + C + G + T)^*$$

Para este laboratorio usted puede usar una corutina para implementar el AFD si lo desea. Pero note que el string será leído completamente por otra tarea. También es posible No usar corutinas si así lo desea.

El archivo de salida debe contener el string y una indicación (si, o no) de pertenencia.

El siguiente es una ejemplo de salida:

```
GATCTTATATAACTGTGAGATTAATCTCAGATAATGACACAAAATATAGTGAAGTTGGTA no
AGTTATTTAGTAAAGCTCATGAAAATTGTGCCCTCCATTCCCATATAATTTATTAATTGT no
CTAGGAAC TTCCACATACATTGCCTCAATTTATCTTTCAACAACTTGTGTGTTATATTTT no
GGAATACAGATACAAAGTTATTATGCTTTCAAAATATTCTTTTGCTAATTCTTAGAACAA no
AGAAAGGCATAAATATATTAGTATTTGTGTACATCTGTTCCCTTCCTGTGTGACCCTAAGT si
TTAGTAGAAGAAAGGAGAGAAAATATAGCCTAGCTTATAAATTTAAAAAAAAAATTTATTT no
GGTCCATTTTGTGAAAAACATAAAAAAAGAACTGTCACATCTTAATTTAAAAAATATATG si
CTTAGTGGTAAGGAGATATATGTCAACTTTTAAGAGGTTGAAAAACAAACGCCTCCCATT si
ATAAGTTTATACTTCACCTCCCACCACTATAACAACCCAGAATCCATGAGGGCATTATCA no
```

**Fecha de entrega (soft-deadline):
martes 23 de junio antes de las 24:00 hrs.**