

Lenguajes de Programación III

Examen final

Prof.: Ing. Gustavo Sosa Cataldo

Planteamiento del T. P.:

Este trabajo involucra el desarrollo de dos implementaciones utilizando conceptos vistos en clase y laboratorio.

Cada tema vale 50% del primer final, por tanto en total corresponde al 100%.

Máxima cantidad de integrantes por grupo: 2 (se puede hacer individual)

Todos los integrantes del grupo deben estar presentes el día de la entrega/defensa, el que no esté presente lleva 0.

La fecha de entrega/defensa será en el día lunes 14 de junio mediante meet. Les enviaré unos días antes el horario asignado a cada grupo. Además se debe compartir el código fuente con el profesor mediante GITHUB o GITLAB (que será revisado si se considera oportuno).

Tema 1

Implementar un programa en C que calcule el factorial de los números que se le pasan como argumentos. El programa creará tantos hilos con Pthreads como argumentos se introduzcan y cada uno de los hilos recibirá uno de esos argumentos y calculará su factorial de manera concurrente, pero **la salida que imprimen los hilos deberán ser en orden** (utilice las herramientas de sincronización estudiadas para cumplir con este proposito). Cada hilo ha de recibir como argumento el número cuyo factorial debe calcular. Un ejemplo de la ejecución del programa podría ser la siguiente:

```
$ factorial 3 6 4 1 0
```

Salida

Factorial de 3: 6

Factorial de 6: 720

Factorial de 4: 24

Factorial de 1: 1

Factorial de 0: 1

Tema 2

Implementa un programa que genere un vector de MAX numeros enteros aleatorios comprendidos entre 0 y 9. A continuación buscará secuencialmente en dicho vector el número que se le pasa como primer argumento al programa. El programa debe crear tantos hilos como se le indique en el segundo argumento. Y cada hilo tiene que realizar la búsqueda en un fragmento consecutivo del vector de un tamaño aproximadamente igual para todos los hilos. Una vez que un hilo encuentre el número, todos los hilos finalizan la

búsqueda. El hilo principal esperara que acaben todos los hilos y mostrará por pantalla el índice de la posición en la que esté el número con el número de proceso, o el mensaje de que no está en el vector.

\$ buscar 4 5

el programa buscará en el vector el número 4 con 5 hilos ejecutándose concurrentemente.

Ejemplo, para este vector de 20 elementos con 5 hilos de búsqueda

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

el primer primer hilo hace la búsqueda en el sub vector de las posiciones [0,3], el segundo hilo en el subvector [4,7], y así sucesivamente.