



CI-0117 Programación paralela y concurrente

Grupos 1 y 2

Enunciado tarea programada I

Fecha de entrega: 2020/May/15

a) Collatz (40%)

Considere las siguientes operaciones para un número entero positivo N arbitrario:

- Si el número es par, entonces lo reemplazamos por el resultado de dividirlo entre 2: $n/2$
- Si el número es impar, entonces lo reemplazamos por el resultado de multiplicarlo por tres y sumarle 1: $3n + 1$

La conjetura de **Collatz**: si aplicamos las reglas anteriores sobre un número entero positivo, el proceso eventualmente llega al número 1, sin importar el valor del número elegido.

Ejemplo, si $N = 7$:

7 es impar, debe ser reemplazado por $7 \times 3 + 1 = 22$
22 es par, debe ser reemplazado por $22 / 2 = 11$
11 es impar, debe ser reemplazado por $11 \times 3 + 1 = 34$
34 es par, debe ser reemplazado por $34 / 2 = 17$
17 es impar, debe ser reemplazado por $17 \times 3 + 1 = 52$
52 es par, debe ser reemplazado por $52 / 2 = 26$
26 es par, debe ser reemplazado por $26 / 2 = 13$
13 es impar, debe ser reemplazado por $13 \times 3 + 1 = 40$
40 es par, debe ser reemplazado por $40 / 2 = 20$
20 es par, debe ser reemplazado por $20 / 2 = 10$
10 es par, debe ser reemplazado por $10 / 2 = 5$
5 es impar, debe ser reemplazado por $5 \times 3 + 1 = 16$
16 es par, debe ser reemplazado por $16 / 2 = 8$
8 es par, debe ser reemplazado por $8 / 2 = 4$
4 es par, debe ser reemplazado por $4 / 2 = 2$
2 es par, debe ser reemplazado por $2 / 2 = 1$
1 fin

Haga un programa en C o C++ que reciba un número como parámetro. Suponga que el usuario indica el límite superior de un conjunto de valores que quiere calcular, por ejemplo 1000, el conjunto sería entonces $[2, 1000]$, su programa debe determinar cuál



es la mayor cantidad de pasos que Collatz realiza sobre todos los elementos y en cuál elemento del conjunto ocurrieron.

De la misma manera que se programó en PiPorSeries, haga que algunos hijos/hilos estén a cargo de calcular algunos de los términos y luego que el padre obtenga el resultado total de la operación.

- Para facilidad del usuario el límite siempre será una potencia de 10, en la siguiente lista se muestra los valores de algunas potencias de 10
 - 10, 9 con 19 pasos
 - 100, 97 con 118 pasos
 - 1000, 871 con 178 pasos
 - 10000, 6 171 con 261 pasos
 - 10^5 , 77 031 con 350 pasos
 - 10^6 , 837 799 con 524 pasos
 - 10^7 , 8 400 511 con 685 pasos
 - 10^8 , 63 728 127 con 949 pasos
 - 10^9 , 670 617 279 con 986 pasos
 - 10^{10} , 9 780 657 630 con 1132 pasos
- Su programa debe indicar el tiempo que tarda en realizar las operaciones solicitadas, la cantidad de pasos y el número que los generó
- Debe construir un programa en C o C++ que cumpla con las especificaciones indicadas.

b) Ascensor – sincronización (60%)

Un edificio de 10 plantas tiene un único ascensor

- **Ascensor**
 - Es representado por un hilo
 - Tiene una capacidad para 8 personas y esta capacidad no puede ser excedida
 - Se detiene cuando nadie lo utiliza
 - Tiene memoria, de modo que pueden solicitarse varias subidas y bajadas al mismo tiempo, que deben acumularse en una cola, para que el ascensor las pueda atender cuando le sea posible
 - Tiene una política de no cambiar de sentido si no es estrictamente necesario
 - Cuando va bajando atenderá primero las peticiones que impliquen desplazarse hacia abajo hasta que ya no tenga otras peticiones pendientes más abajo y hará lo mismo en sentido ascendente



- Si una persona quiere subir de una planta a otra cuando el ascensor desciende, esta solicitud se posterga hasta que el ascensor comienza a subir de nuevo. Caso equivalente cuando el ascensor asciende.
- Para cambiar de sentido el ascensor debe estar vacío
- Si el ascensor está lleno solo se detendrá en el siguiente piso en el que una persona quiera bajar
- **Personas**
 - Es representada por un hilo
 - Cuando una persona quiere utilizar el ascensor, desde la planta en que se encuentra pulsa el botón de llamado y especifica hacia cuál planta se dirige por medio del llamado a un método "Moverse(sube, baja)", des esta manera avisa al ascensor (hilo) la intención de este usuario. La persona entonces debe esperar hasta que el ascensor llegue a su planta y se pueda montar
 - Cuando una persona llega a su planta de destino baja del ascensor
- **Programar** en C++ el comportamiento del ascensor y las personas utilizando dos tipos de hilos: **Persona(e, s)** y **Ascensor**
 - La sincronización, concurrencia y eficiencia son importantes
 - Debe crear al menos una clase para el "Ascensor" y otra para las "Personas"
 - Para seguir lo que está ocurriendo en cada momento es menester que tanto el ascensor como las personas emitan mensajes por pantalla de vez en cuando para saber qué es lo que está ocurriendo, por ejemplo:
 - 'Persona 9: Pulso el botón en la planta 3 para subir al piso 10'
 - 'Persona 4: El ascensor está lleno, no puedo subir'
 - 'Ascensor: Voy bajando por la planta 3'
 - 'Ascensor: Me detengo para que baje un viajero en planta 5'
 - etc...
 - Para las pruebas su programa debe funcionar correctamente creando al menos 100 hilos que representen personas
 - Debe soportar cargas alternas, esto es que podemos generar secuencias de personas (hilos) y permitir que el ascensor comience a trabajar y realizar la atención de personas, luego de un tiempo (sleep x), sin que la primera carga haya terminado arrancar una segunda carga de personas, que en este caso deben acomodarse de acuerdo con la situación actual del ascensor
 - Al final de la tarea el ascensor debe quedar ocioso y todas las personas atendidas
 - La generación de las plantas de salida y llegada de las personas debe generarse de manera aleatoria y tener sentido