UNIVERSIDAD DE COSTA RICA

ECCI Escuela de Ciencias de la Computación e Informática

CI-0122 Sistemas Operativos

Grupos 2 y 5

Enunciado tarea programada I

Fecha de entrega: 2026/Set/16
Modalidad: individual

Juego de la “papa” caliente

Vamos a construir un anillo de procesos (n) simulando la ronda en el juego de la **papa caliente** (paso de token). Se podrá jugar en el sentido de las manecillas del reloj, entonces el proceso ***i*** le pasa la papa (un mensaje) al proceso ***i+1***, el proceso ***n-1*** se la pasará al ***0***; también será posible jugar en el sentido contrario a las manecillas del reloj, donde el proceso ***i*** le pasa la papa (un mensaje) al proceso ***i-1***, el proceso ***0*** se la pasará al ***n - 1.*** La papa, que tiene un valor inicial (v), es movida entre los participantes, siguiendo la ronda, cada participante **activo** podrá modificar su valor, compruebar si la papa estalló, en cuyo caso “sale” del juego, se convierte en un jugador **pasivo** y escoge al azar un nuevo valor inicial para la papa (v_i), para luego pasar la papa al siguiente participante. Cada participante **pasivo** pasa el valor de la papa sin modificarlo hasta que el juego termine.

Cada participante estará representado por un **proceso** que tendrá un identificador único que utilizará para comunicarse con sus vecinos: 0, 1, 2, 3, ..., n-1. Al inicio el programa podrá recibir tres parámetros: primero que indica la cantidad de miembros de la ronda (n), el segundo un número inicial para la papa (v) y el tercero la dirección de giro de la papa (**clockwise** = 0 -> 1 -> 2 -> ... -> n-1 -> 0, etc o **counter-clockwise** = 0 -> n-1 -> n-2 ... -> 2 -> 1 -> 0). Si no se especifica alguno de los parámetros el programa deberá generarlos convenientemente.

Cada proceso recibe un mensaje con el valor de la papa, aplica las reglas de **Collatz** a ese valor, si el resultado es uno, será el indicativo de que la papa **explotó**, entonces ese proceso “sale” del juego y se convierte en un comunicador **pasivo**. Gana el último proceso en “salir” del juego, al que le tocará avisar a los demás procesos que el juego terminó, pasando para ello un mensaje con un valor de la papa negativo, para que los demás puedan finalizar su ejecución. El programa debe desplegar el valor del proceso ganador.

Para simular este juego, construya un programa que genere un proceso (fork) para cada participante de la ronda (n) y establezca los elementos de sincronización. Elija al azar el participante que va a arrancar con el valor indicado como parámetro para la papa (v). Utilice **buzones** para intercambiar mensajes entre los participantes. Despliegue información para determinar el estado del juego. El procedimiento principal (“main”) no participará en las decisiones de sincronización (excepto para iniciar el juego), ni en la ronda, una vez creados los recursos, solo debe esperar a que todos los participantes terminen, para eliminar los elementos de sincronización y finalizar la ejecución del programa.

Además, debe crear otro proceso que se encargará de generar mensajes mal intencionados para confundir a los participantes de la ronda. Este proceso generará mensajes al azar cada cierto tiempo y los enviará a los distintos participantes de la ronda, para tratar de alterar su funcionamiento. Cada participante deberá saber cuáles mensajes son válidos y cuáles no, de acuerdo con el identificador del proceso. Por la burocracia de este proceso **invasor**, le va a costar arrancar un poco y podemos garantizar que los mensajes de la primera ronda provienen de emisores válidos, por lo que cada proceso puede registrar esta fuente para saber recharzar los mensajes inválidos.