



**Sistemas Operativos**  
**Laboratorio**

**Tema:** Implementación de hilos y sincronización de hilos en Java

**Objetivos:**

Este taller tiene como objetivos principales que los alumnos:

- Comprendan los conceptos de procesos livianos
- Experimenten sobre la implementación de hilos en Java.
- Modele con Redes de Petri el funcionamiento de procesos.
- Comprendan la utilidad de los semáforos y monitores para la coordinación y sincronización de procesos.

**Actividades.**

**Actividad 1.** Implemente una simulación de la fábula que cuenta la carrera entre la liebre y la tortuga. Para hacerlo más interesante la carrera será cuesta arriba por una pista resbaladiza, de modo que a veces podrán resbalar y retroceder algunas posiciones. Habrá un thread que implementará la tortuga y otro la liebre. Cada uno se suspenderá durante un segundo y luego evaluará lo que ha pasado según unas probabilidades:

Animal	Suceso	Probabilidad	Movimiento
Tortuga	Avance rápido	50%	3 casillas a la derecha
	Resbaló	20%	6 casillas a la izquierda
	Avance lento	30%	1 casilla a la derecha
Liebre	Duerme	20%	
	Gran salto	20%	9 casillas a la derecha
	Resbalón grande	10%	12 casillas a la izquierda
	Pequeño salto	30%	1 casilla a la derecha
	Resbalón pequeño	20%	2 casillas a la izquierda

Calcule la probabilidad con random, de 1 a 100 y determine con dicho número que ha hecho cada animal. Considere que hay 70 casillas, de la 1 a la 70, la 1 de salida y la 70 de llegada. Si resbala al principio vuelve a la 1, nunca por debajo. Tras cada segundo y después de calcular su nueva posición imprima una línea por cada animal, con blancos de 1 la posición-1 y luego una letra T para la tortuga y una L para la liebre. Imprima al comienzo de la carrera un mensaje. Después de imprimir las líneas determine si alguno ha llegado a meta y ha ganado, imprimiendo un mensaje. Si ambos llegan a la vez declare un empate.

**Actividad 2.** Suponga tener  $n$  usuarios que comparten 2 impresoras. Antes de utilizar la impresora, el  $U[i]$  invoca requerir(impresora). Esta operación espera hasta que una de las impresoras esté disponible, retornando la identidad de la impresora libre. Después de utilizar a impresora,  $U[i]$  invocan a liberar(impresora).

- Resuelva este problema considerando que cada usuario es un thread y la sincronización se realiza utilizando semáforos.
- Resuelve este problema considerando que cada usuario tiene una prioridad única. Se otorga la impresora al usuario con mayor prioridad que está esperando.

**Actividad 3.** Considere un sistema con tres Fumadores y un proceso Proveedor. Cada fumador está continuamente armando y fumando cigarrillos. Sin embargo, para armar un cigarrillo, el fumador necesita tres ingredientes: tabaco, papel y fósforos. Uno de los procesos fumadores tiene papel, otro tiene tabaco y el tercero los fósforos. El proveedor tiene una cantidad infinita de los tres materiales. El proveedor coloca dos de los ingredientes sobre la mesa. El fumador que tiene el ingrediente restante arma un cigarrillo y se lo fuma, avisando al proveedor cuando termina. Entonces, el proveedor coloca dos de los tres ingredientes y se repite el ciclo. Escriba un programa para coordinar al proveedor y los fumadores utilizando semáforos.

**Actividad 4.** El problema del barbero dormilón (Dijkstra 1971). Una barbería se compone de una sala de espera con  $n$  sillas y la sala de barbería donde se encuentra la silla del barbero. Si no hay clientes a quienes atender, el barbero se pone a dormir. Si un cliente entra y todas las sillas están ocupadas, el cliente sale de la barbería. Si el barbero está ocupado, pero hay sillas disponibles, el cliente se sienta en una. Si el barbero está dormido, el cliente lo despierta. Escriba un programa para coordinar al barbero y sus clientes utilizando semáforos.

**Actividad 5.** Una cuenta de ahorros es compartida entre distintas personas (procesos). Cada persona puede sacar o depositar dinero en la cuenta. El balance actual de la cuenta es la suma de los depósitos menos la suma de las cantidades sacadas. El balance nunca puede ser negativo. Se pide:

- Construir un monitor en pseudocódigo para resolver este problema con las operaciones depositar(cantidad) y devolver(cantidad).
- El cliente que deposita debe despertar a los clientes que están esperando para sacar
- El cliente que llega para sacar dinero lo saca si existe saldo, independientemente de que haya algún otro proceso esperando porque no hay suficiente dinero para él.

**Actividad 6.** Implementar el problema de la tribu con monitores. Los miembros de la tribu cenar en comunidad de una gran olla que contiene  $M$  misioneros. Cuando un miembro de la tribu quiere comer, él mismo se sirve de la olla, a menos que esté vacía. Si la olla está vacía, entonces despierta al cocinero y espera a que éste llene la olla. Sólo se debe despertar al cocinero cuando la olla está vacía.

### **Criterio de Evaluación:**

- Adecuada documentación de las prácticas propuestas.
- Presentación (la documentación entregada deberá ser clara, libre de errores de ortografía, ordenada, concisa y acotada a lo que se le solicita)

### **Fecha de Entrega:**

24 de mayo

### **Escala de Valoración:**

La escala de valoración a emplear será cualitativa (Excelente – Muy Bueno – Bueno - Regular - Desaprobado)