

CC3002 – Examen 2

Nombre: _____ Carné: _____

- I. (22 puntos) Considere los archivos `fork` y `readFile`. `fork` crea un proceso hijo con el programa de `readFile`. `readFile` simplemente lee un archivo. Durante el proyecto 1, se utilizó la system call `sleep` para sincronizar el proceso `fork` para que este espere a que se ejecute el proceso hijo. Este tipo de espera es del tipo *busy waiting*.
- ¿Cómo puede relacionar el concepto de *busy waiting* con el de costo de oportunidad?
 - Modifique los programas `fork` y `readFile` para que en lugar de utilizar *busy waiting*, se sincronicen (`fork` espere a que `readFile` termine de ejecutarse) utilizando semáforos.
- II. (22 puntos) Imagine una versión multi hilos del juego "Memoria" (Ver `memoria.java`). El juego está compuesto por un arreglo de cartas que están volteadas hacia abajo. Cada jugador toma dos cartas y si son iguales, el jugador se queda con las cartas. El juego finaliza cuando ya no queden cartas. Imagine que varias threads ejecutan el método `play()` de forma simultánea.
- ¿Puede ocurrir un deadlock? ¿Cómo? (En qué líneas deben ir ambas threads)
 - En el caso que se presente el escenario donde puede ocurrir el deadlock ¿Cuál es la probabilidad de que este ocurra?
 - ¿Qué mecanismo puede incluir en el código de `play()` para que no ocurran deadlocks?
- III. (22 puntos) Calendarización
- Realice un diagrama de Gantt que ilustre la ejecución de los siguientes procesos utilizando un algoritmo Preemptive por Prioridades. (Utilizar FCFS para procesos con la misma prioridad)

Process	Arrival time	Burst time	Priority
P1	0	10	3
P2	2	1	1
P3	3	2	3
P4	4	1	4
P5	6	5	2

- Realice un diagrama de Gantt que ilustre la ejecución de los siguientes procesos utilizando un algoritmo de Multilevel Feedback Queue.

Process	Arrival time	Burst time
P1	0	17
P2	12	25
P3	28	8
P4	36	32
P5	46	18

La configuración de la Multilevel Feedback Queue está dada por:

```
<sys>
  <scheduling startQueue="1">
    <queue id="1" name="Queue1" type="FCFS" quantum="8" preemptive="1" queueOnQuantumEnd="2" queueOnIOEnd="1" />
    <queue id="2" name="Queue2" type="FCFS" quantum="16" preemptive="1" queueOnQuantumEnd="3" queueOnIOEnd="2" />
    <queue id="3" name="Queue3" type="FCFS" quantum="0" preemptive="1" queueOnQuantumEnd="3" queueOnIOEnd="3" />
  </scheduling>
</sys>
```

IV. (22 puntos) Algoritmo del Banquero. Considere las siguientes matrices que representan el estado de asignación de recursos.

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	0	0	0	0	4	2	3	1	5	2	4	4
P2	0	1	2	2	2	1	4	4				
P3	5	1	1	1	5	1	2	2				

- ¿Está el sistema en un estado seguro?
 - El proceso P1 solicita 3 unidades del recurso D. ¿Es segura esta asignación?
 - El proceso P3 solicita 1 unidades del recurso C. ¿Es segura esta asignación?
- V. (22 puntos) Realice un diagrama de estados para los procesos que realizó en el proyecto. El diagrama debe incluir todas las posibles transiciones de estados y las operaciones que producen cada transición. No olvide mencionar estados iniciales y finales.