



# Sistemas Operativos

## Examen Recuperatorio – 6 de noviembre de 2017



Registro: ..... Nombre: ..... Cant. De Hojas: .....

### Indicaciones

- Coloque el nombre, número de libreta y cantidad de hojas entregadas sin incluir el enunciado.
- Realice **UN EJERCICIO** por hoja. Identifique el ejercicio en la hoja.
- No utilice lápiz. Firme la última hoja

### Ejercicio 1

Conteste en forma completa y breve las siguientes preguntas:

- a) Existen varios objetivos de diseño para un sistema operativo, por ejemplo, utilización de recursos, amigabilidad, eficiencia, robustez, tiempo de respuesta, etc. Brinde un ejemplo de dos objetivos de diseño que se contradigan entre sí.
- b) ¿Cuál es la diferencia entre modo kernel y modo de usuario? Explique cuáles son los beneficios en el diseño de los sistemas operativos.

### Ejercicio 2

- a) Describa ventajas y desventajas de entre las planificaciones: SJF apropiativa y colas multinivel con retroalimentación.
- b) En un sistema con planificador de colas multinivel, en donde a cada nivel se le asigna un algoritmo de planificación round-robin con diferente quantum y teniendo mayor prioridad las colas con quantum más chico, ¿qué tipos de procesos (limitados por CPU o limitados por E/S) serían más adecuados para colas con quantum más bajo y cuáles otros para colas con quantum más alto?

### Ejercicio 3

Consideré los siguientes procesos p1, p2 y p3, ejecutando de manera asincrónica la siguiente secuencia de código:

| P1        | P2        | P3        |
|-----------|-----------|-----------|
| ....      | ....      | ....      |
| wait(x)   | wait(y)   | wait(z)   |
| ....      | ....      | ....      |
| wait(z) ← | wait(y) ← | wait(x) ← |
| ....      | ....      | ....      |
| signal(x) | signal(y) | signal(z) |
| ....      | ....      | ....      |
| signal(z) |           | signal(x) |

Las flechas en cada línea indican qué instrucción se está ejecutando en cada proceso. Todos los semáforos se inicializan en 1.

- a) ¿Cuáles condiciones que se deben cumplir para que un sistema entre en deadlock?
- b) Dibuje un grafo de alocación de recursos que describa la situación. Represente cada semáforo como un recurso y las operaciones wait() y signal() significan requerir y liberar el recurso.
- c) ¿El grafo representa un estado de deadlock? Justifique.

#### Ejercicio 4

El problema de los fumadores de cigarrillos (Patil 1971). Considere un sistema con tres procesos fumadores y un proceso agente. 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 el tabaco y el tercero los fósforos. El agente tiene una cantidad infinita de los tres materiales. El agente coloca dos de los ingredientes sobre la mesa. El fumador que tiene el ingrediente restante armaría un cigarrillo y se lo fuma, avisando al agente cuando termina. Entonces, el agente coloca dos de los tres ingredientes y se repite el ciclo. Los ingredientes son tomados de a uno por vez.

- Resuelva en pseudocódigo el problema para sincronizar al agente y los fumadores utilizando semáforos. Para la evaluación se considerará el uso eficiente de recursos, el grado de concurrencia y el sistema no deberá quedar interbloqueado.
- Explique las operaciones de las herramientas que utilizó para la sincronización.

#### Ejercicio 5

Se tiene un sistema con un procesador sobre el que ejecuta un sistema operativo con un planificador Round Robin con un quantum de 20ms.

El siguiente programa comienza su ejecución en el sistema planteado:

| Programa  |
|---|
| Ejecuta durante 10ms<br>pid1 = fork()<br>Ejecuta durante 15ms<br>Se bloquea durante 10ms<br>pid2 = fork()<br>Ejecuta durante 25ms<br>if pid1 <> 0 and pid2 <> 0 then<br>wait()<br>wait()<br>Ejecuta durante 15ms<br>else if pid2 <> 0 then<br>wait()<br>Ejecuta durante 10ms<br>end if<br>Ejecuta durante 5ms<br>Termina. |

Las operaciones `fork()` y `wait()` son estilo Unix, donde `wait()` espera por la finalización de un único hijo. Sabiendo que este es el único programa en ejecución en el sistema, y que el tiempo de ejecución de las funciones `fork()` y `wait()` es de 5ms cada una. Considere que las operaciones del SO son despreciables.

- Realice un diagrama de planificación (tiempo vs procesos) de la ejecución del programa, comenzando en tiempo  $t=0$ , indicando el estado de cada uno de los procesos (listo/ejecutando/bloqueado/terminado) en cada intervalo de tiempo.
- Calcule el tiempo de espera y de retorno de cada proceso.