

CC3002 – Tarea 3

Entrega: miércoles 07/04/2010 antes de clase, a través de sakai.

- I. El barbero dormilón, Una peluquería tiene un barbero, una silla de peluquero y n sillas para que se sienten los clientes en espera, si es que los hay. Si no hay clientes presentes, el barbero se sienta en su silla de peluquero y se duerme. Cuando llega un cliente, este debe despertar al barbero dormilón. Si llegan más clientes mientras el barbero corta el cabello de un cliente, estos deben esperar sentados (si hay sillas desocupadas) o salirse de la peluquería (si todas las sillas están ocupadas). El problema consiste en programar al barbero y los clientes sin entrar en condición de competencia.

Implemente las funciones del barbero y de los clientes con las siguientes variables de sincronización:

```
semaforo Cliente = 0
semaforo Barbero = 0
semaforo mutex = 1 // Provee exclusión mutua a la variable SillasLibres
int SillasLibres = N
```

- II. Indique si la siguiente implementación de una cola es segura para utilizarse en un ambiente multi threaded (No ocurren deadlocks y los datos no se corrompen).

```
#define QUEUE_LEN 32

typedef struct{
    void *buf[QUEUE_LEN];
    int start;
    int end;
    mutex_t lock;
} queue_t;

/* This function adds data to the queue
 * returns 0 on success, -1 on failure
 */
int enqueue(queue_t *queue, void *data) {
    /* If the queue is full, return failure */
    if (queue->start == (queue->end + 1) % QUEUE_LEN)
        return -1;
    /* lock the queue while we set the data and increment the length */
    mutex_lock(&queue->lock);
    queue->buf[queue->end] = data;
    queue->end = (queue->end + 1) % QUEUE_LEN;
    mutex_unlock(&queue->lock);
    return 0;
}
```

III. Algoritmo del banquero

- Explique por qué este algoritmo recibe el nombre del algoritmo del banquero. ¿Cuál sería la analogía de allocation, max, available, cada proceso y cada recurso?
- ¿Está el sistema en un estado seguro?
- ¿Puede la solicitud de P1 (0,4,2,0) ser completada de forma inmediata?
- ¿Puede la solicitud de P4 (0,3,2,0) ser completada de forma inmediata?

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P0	0 0 1 2	0 0 1 2	1 5 2 0
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	