

## AUX 2 – CC41B

Prof.: Luis Mateu Aux.: Juan Manuel Barrios

### Pregunta 1 (P1, Control 1, primavera 2000)

Se propone la siguiente solución con semáforos para múltiples productores y múltiples consumidores con un solo buffer:

```
nSem mutex = nMakeSem(1);
nSem vacio = nMakeSem(1);
nSem lleno = nMakeSem(0);

Productor() {
    for(;;){
        c = producir();
        nWaitSem(mutex);
        nWaitSem(vacio);
        buffer = c;
        nSignalSem(lleno);
        nSignalSem(mutex);
    }
}

Consumidor() {
    for(;;) {
        nWaitSem(mutex);
        nWaitSem(lleno);
        c = buffer;
        nSignalSem(vacio);
        nSignalSem(mutex);
        consumir(c);
    }
}
```

Revise si está correcta, explique porqué y proponga mejoras.

### Pregunta 2

Suponiendo la siguiente estructura de datos de un árbol binario, y utilizando nSystem implemente:

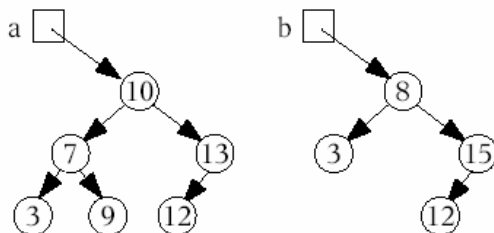
int buscarSeq(Node \*node, int inf); Creando tantas tareas como nodos para recorrer el árbol en paralelo.

int buscarSeq(Node \*node, int inf, int level); Creando tareas mientras la profundidad del árbol no supere algún nivel, y luego recorriendo el árbol recursivamente.

int buscarSeq(Node \*node, int inf, int level); Como el anterior, pero con una optimización: La ejecución de todas las tareas termina cuando alguna encuentra el elemento buscado.

### Pregunta 3 (P2, Control 1, primavera 2000)

Se tienen dos árboles de búsqueda binaria referenciados por las variables globales a y b. Se desea desplegar en pantalla en orden ascendente el contenido de ambos árboles.



Por ejemplo para los 2 árboles de la izquierda, la salida de su programa debe ser:

3 3 7 8 9 10 12 12 13 15

La estructura de cada nodo está dada por:

```
typedef struct Nodo {
    struct Nodo *izq, *der;
    int valor;
} Nodo;
```

Restricciones: la solución debe tomar tiempo proporcional al número de nodos de ambos árboles y ocupar a lo más un espacio en memoria proporcional a la máxima altura de ambos árboles.

Resuelva este problema en nSystem utilizando 2 tareas adicionales. Cada tarea recorre recursivamente uno de los árboles, enviando ordenadamente los enteros almacenados a la tarea principal por medio de mensajes. No olvide terminar adecuadamente ambas tareas con nWaitTask.