Parcial de PRED

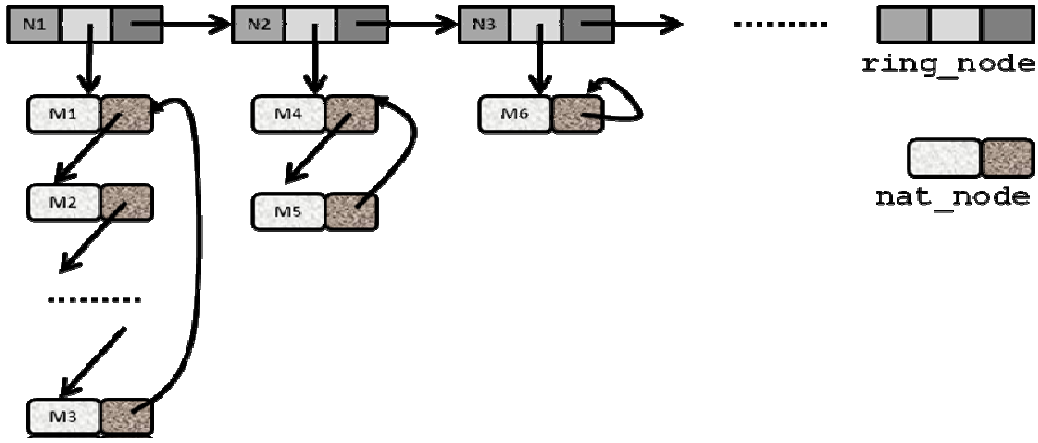
FIB-UPC

4 de noviembre de 2010

---

1) (6 puntos)

Se quiere definir la estructura de datos **`lista de anillo`**. Un **`anillo`** es una lista circular de naturales. Adicionalmente, cada anillo tiene un identificador representado por un número natural. Una representación gráfica de la estructura es la siguiente:



La lista lineal formada por nodos de tipo **`ring_node`** ha de estar ordenada de mayor a menor. No hace falta que los anillos, es decir, las listas circulares formadas por nodos de tipo **`nat_node`**, estén ordenadas. La estructura **`lista de anillo`** ha de incluir una operación **`borrar`** que, dados dos números naturales N y M, elimine un **`nat_node`** con el valor M de la lista circular encabezada por el **`ring_node`** con identificador igual a N. Además, si este anillo quedara vacío, también tendría que eliminar el **`ring_node`** con identificador N. Concretamente, se pide lo siguiente:

a) Definir la estructura de clases que implemente **`lista de anillo`**.
b) Dar el algoritmo correspondiente a la operación **`borrar`** en lenguaje natural.
c) Dar una implementación de la operación **`borrar.`**

2) (2 puntos) Describir en menos de 15 líneas, utilizando adicionalmente gráficos si queréis, el algoritmo de marcado y búsqueda (mark & scan) de recolección de basura (*garbage collection)*.

3) (2 puntos) Considere el siguiente segmento de código en C:

```
int *q;
q = new int;
void p() {
     int x;
     q  = &x;
     *q = 1;
}
```

Cuál sería la salida después de ejecutarse las siguientes instrucciones:

```
*q = 0;
p();
r();
printf(*q);
```

donde `r` es un procedimiento que no menciona `q`. Recordad que C tiene alcance estático y que los operadores '&' y '*' permiten tomar la dirección y el contenido de la variable argumento, respectivamente.