



Estructura de datos en assembly del MIPS R2000

“Listas enlazadas categorizadas”

Asignatura: Instalación y Reemplazo de Componentes Internos 2021

Curso: 6to Informática

Alumnos:

- Binaghi, Julio
- Cosenza, Teo
- Diez, Laureano
- Maiza, Tomás Agustín
- Juan Cruz Villena

Profesor: Alejandro Rodríguez Costello

Agosto 2021

En primer lugar, partiendo de los ejemplos de listas simplemente enlazadas que se muestran en el pdf, fuimos construyendo la estructura de una lista doblemente enlazada circular apoyándonos en las subrutinas *newnode* y *smalloc* propuestas. La primera permite agregar un nuevo nodo a una lista (o iniciar la lista si esta está vacía), y la segunda pide un lugar en memoria (4 palabras) para cada nodo que se ingresa utilizando la syscall 9 (*sbrk*). Comenzamos a modificar la lista enlazada simple para adaptarla a lo que se pedía y construimos así la estructura de una lista de elementos (que contendría todos los elementos de una categoría en específico). Cada nodo está compuesto de 4 palabras: la primera apunta al nodo anterior (o al último si es el nodo más reciente de la lista), la segunda contiene la numeración del elemento dentro de la categoría, la tercera contiene el elemento que se ingresó y la cuarta contiene la dirección del siguiente nodo (o del nodo más reciente si es el último nodo de la lista).

Luego hicimos el menú, creando etiquetas para mostrar mensajes indicando cada posible acción del programa: cada una va acompañada de un número, y al ingresar el número deseado se accede a la función pedida. Las opciones posibles son:

1. Crear una categoría
2. Pasar a la categoría siguiente
3. Pasar a la categoría anterior
4. Mostrar todas las categorías
5. Borrar una categoría
6. Agregar un objeto a la categoría seleccionada
7. Borrar un objeto de la categoría seleccionada
8. Listar todos los objetos de la categoría seleccionada
9. Finalizar programa

Crear una categoría

Tomamos como base la función *newnode* y la adaptamos a los requerimientos de la lista de categorías expresados en la consigna. Cada categoría se compone de 4 palabras: la primera apunta al elemento anterior (o último elemento de la lista si es la primera categoría), la segunda apunta a la lista de objetos de la categoría, la tercera contiene el valor (nombre de la categoría) y la cuarta apunta a la siguiente categoría (o a la primera si es el último elemento de la lista).

Pasar a la categoría siguiente y anterior

Cargamos en un registro la dirección de la categoría seleccionada en curso y avanzamos a la anterior o posterior según corresponda a partir de la dirección almacenada en la primera o cuarta palabra del nodo respectivamente y actualizamos el puntero *wclist*. Actualizamos también los punteros al inicio y final de la lista de objetos de la nueva categoría seleccionada, sabiendo que la dirección al primer elemento de la misma se almacena en la segunda palabra de la categoría y la dirección del último elemento se almacena en la primera palabra del primer elemento de la lista.

Listar todos los objetos de la categoría seleccionada

Mostramos cada elemento de la lista partiendo del primero (al que apunta *slist*) y avanzando de a uno considerando la dirección guardada en la cuarta palabra de cada nodo. Si la dirección del elemento que se mostró por última vez es igual a la dirección contenida en el puntero *final* entonces se corta el loop y volvemos al menú.

Listar todas las categorías

De forma similar a lo anterior, recorrimos con un loop todas las categorías y fuimos mostrando sus nombres hasta que se haya mostrado la última. En cada iteración, comparamos si la dirección de la categoría coincide con la almacenada en el puntero *wclist*. En caso de que lo sea, se muestra un asterisco (*) al lado del nombre de la categoría.

Eliminar nodos

Para eliminar nodos, pedimos que el usuario ingrese el ID del nodo a eliminar. El mismo es comparado con el primer y último nodo. Si resulta que no es ninguno de los dos, actualizamos los punteros del nodo anterior y del siguiente al seleccionado de forma en que ambos se conecten. Luego, actualizamos el id de los consiguientes nodos al eliminado dentro de un loop, hasta encontrar el último. Una vez encontrado, derivamos a una función que carga el nodo eliminado en una lista, para luego volver al menú. El mismo procedimiento resulta en los casos de ser el primer o último nodo a borrar.

Eliminar categorías

En un desarrollo conceptual similar al anterior, la función permite eliminar la categoría seleccionada. De la misma forma, se evalúa su posición relativa con las otras categorías. Una vez ubicada, borramos todos los objetos o nodos contenidos en esa categoría utilizando nuestra función de eliminar nodos, para así luego volver a la categoría y liberarla del resto con el mismo procedimiento, actualizando la lista y los punteros necesarios.