12/11/2017

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
Laboratorio de Algoritmos y Estructuras III
Trimestre Septiembre-Diciembre 2017

Javier Vivas - Carnet: 12-11067
Daniel Francis - Carnet: 12-10863

Reporte del Proyecto 1: Manejador y Lista

1. Introducción

El proyecto se desarrolló a través de tres actividades: La primera era diseñar e implementar un manejador de espacio RAM para la arquitectura RISC de un CPU MIPS. La segunda fue utilizar el manejador para administrar la memoria utilizada de una estructura de datos abstracta correspondiente a una lista enlazada simple. Finalmente, la tercera consistía en la entrega del código fuente de las primeras actividades.

Decidimos comenzar con el manejador para después trabajar sobre la lista. Realizamos un esquema mental de cómo sería más sencillo implementar un manejador de espacio con los recursos disponibles en el simulador MARS.

2. Desarrollo y diseño

Para el manejador consideramos una lista enlazada simple que funcionase como "directorio" de direcciones que permitiese administrar la dirección solicitada a la memoria heap de MIPS. La inicialización del manejador resultaría en la creación de atributos de operación y la cabeza de la lista.

Cada nodo de la lista tendría tres componentes de 4 bytes: Dirección, tamaño y siguiente. Dirección contendría la dirección de memoria correspondiente al espacio o bloque reservado por la función "malloc", tamaño almacena el tamaño de dicho bloque y siguiente apunta al siguiente nodo del directorio.

Para calcular espacios disponibles, restamos direcciones y tamaños entre bloques. Esta información es proporcionada por cada nodo del directorio. Así logramos implementar las funciones malloc, reallococ y free sin mayores problemas.

Para la segunda actividad nos basamos en el enunciado, que sugería reservar tres atributos para la cabeza de la lista enlazada y dos atributos para cada nodo. El orden de la lista lo obtenemos iterando sobre la misma.

Al evaluar los casos de prueba, nos dimos cuenta de que no implementamos la función perror del manejador como era debido. Inicialmente habiamos entendido a través de horas de consulta que perror debía permitirle al usuario arreglar el error que había cometido.

Este inconveniente alteró toda la estructura del manejador, y por lo tanto del proyecto, haciendo imposible que los casos de prueba oficiales funcionasen con nuestro proyecto.

3. Conclusión

Se hizo aparente la importancia de seguir las convenciones de programación, en este caso las convenciones de lenguaje ensamblador de MIPS. Esto asegura que nuestro programa funcione con cualquier otro.

No obstante, notamos las dificultades de no conocer con claridad las pautas de la entrega del proyecto y del comportamiento de las funciones. Hubo una gran diferencia entre cómo probamos la correctitud del proyecto y en cómo iba a ser la evaluación realmente. Por esto no fue posible verificar los casos de prueba en presencia del tutor.

También vimos que comentar y probar el código a medida que se programa y se verifica cada paso con el tutor es de suma importancia para asegurar que el proyecto cumple los objetivos y funciona correctamente.