

Tutoría 5 - Estructuras de datos y algoritmos

Cristóbal Ganter y Felipe Vera

8 de octubre de 2012

¡Aviso! ¡Recuerde dejar la máquina virtual en orden una vez que termine su trabajo! ¡Pregunte al ayudante apenas tenga una duda!

Si tiene problemas de *segmentation fault* o *stack overflow* en su programa, una buena opción es *debuggear* su programa con el comando `ddd nombre-de-programa`. Estos errores son difíciles de encontrar, por lo que llenar su programa de `printfs` no lo ayudará mucho.

Lea la guía completa antes de empezar a trabajar. Esto le permitirá desarrollar los programas teniendo en cuenta lo que viene más adelante.

No se espera que los alumnos sepan usar las funciones `fread` y `fwrite`. Si tiene problemas, lea la documentación en *libc.pdf* y luego pregunte a su ayudante.

Contenido de esta semana

- punteros, punteros dobles y operador `&`
- Asignación dinámica de memoria: `stack`, `heap`, `malloc` y `free`.
- Estructuras de datos dinámicas: árboles binarios de búsqueda. Creación y lectura.
- Empleo de funciones recursivas para los contenidos anteriores.
- Empleo de funciones para guardar, leer y obtener información de archivos binarios y de texto.

Actividad

1. **Creación de un árbol binario de búsqueda** El árbol binario de búsqueda, es un método efectivo para ordenar elementos de una lista y acceder a ellos usando memoria dinámica. En este caso, se desea ordenar un grupo de personas según su edad.
 - a) Cree una estructura que permita construir un árbol binario de búsqueda. La estructura además debe poder guardar el nombre y la edad de una persona.
 - b) En su función `main` permita al usuario ingresar nombres de personas y sus respectivas edades. Cada par nombre edad debe guardarse en una estructura y cada estructura debe insertarse en un árbol binario de búsqueda (ABB). *El ABB debe estar ordenado según edad.*
2. **Acceso a los ítemes del árbol binario. Recorrer el árbol en orden ascendente** Mediante una función recursiva, se puede obtener todos los nodos de árbol de búsqueda de forma ordenada.
 - a) Escriba una función recursiva que recorra el ABB ascendentemente imprimiendo el **nombre** de cada persona.
 - b) Escriba una función idéntica a la anterior que, en vez de imprimir cada nodo, lo guarde en un archivo binario utilizando la función `fwrite`.
 - c) Llame a la función del punto 2.b. desde su función `main`.
3. **Manejo de archivos: Cargar el árbol guardado**
 - a) Reserve suficiente memoria como para cargar al archivo generado en el punto 2. Puede utilizar la función `stat` para encontrar el tamaño del archivo.

- b) Lea el archivo generado en el punto 2 utilizando la función `fread`. Los datos leídos desde el archivo deben ser almacenados en el espacio de memoria reservado en el punto anterior.
- c) Utilizando la función escrita en 2.a. intente de recorrer el ABB. ¿que ha pasado? ¿cual es el problema? ¿qué alternativas de solución se le ocurren?