

Práctica 13 de C: 19-05-2009

Objetivos:

- Familiarizarse con las funciones de C para la gestión dinámica de memoria.
- Asentar los conceptos de manejo de ficheros ya vistos.

Enunciado:

- Descarga el fichero *Gestión_Memoria_Dinámica_en_C.pdf* de la página web de la asignatura.
- Revisa el primer apartado y realiza en una hoja de papel el primer ejercicio. ¿Observas la importancia de mantener un equilibrio entre las funciones *obtener()* y *liberar()*? (10 minutos)
- Revisa ahora el apartado siguiente, donde se describen la sintaxis y la semántica de las funciones C equivalentes a *obtener()* y *liberar()*. ¿Cuáles son las equivalencias? ¿Qué diferencias existen entre *malloc()*, *calloc()* y *realloc()*? (10 minutos)
- (30 minutos) Revisa el ejemplo que se propone para la función *malloc()*. Realiza un nuevo programa, *practica13.c*, que lea una cadena de caracteres de teclado (máximo 80 caracteres, incluyendo espacios en blanco). A continuación debes hacer una copia de esa cadena en otra cadena de caracteres, definida inicialmente como un puntero a carácter. Previamente debes solicitar memoria al sistema mediante la función *calloc()*, reservando únicamente la memoria necesaria. Muestra la copia por pantalla y comprueba que ambas son iguales.
 - Repite ahora el mismo procedimiento (leer, reservar memoria y copiar) con la función *malloc()*. ¿Hay alguna diferencia en el comportamiento del programa?
 - Modifica nuevamente el programa y repite la lectura de frases del teclado hasta que la cadena que se lea esté vacía.
 - ¿Ha sido necesario utilizar la función *free()*? Si no la has utilizado, ¿qué ha ocurrido con la memoria que has ido reservando?
- (30 minutos) En el ejemplo anterior has ido reservando memoria y realizando las copias sobre un mismo puntero. De esta forma vas perdiendo la información que habías leído en el paso anterior. Si quisiésemos guardar las frases leídas, ¿qué tipo de dato tendríamos que definir (imagínate que no imponemos un máximo de 10 cadenas leídas)? ¿Cómo tendrías que modificar el programa *practica13.c* para guardar las frases leídas y poder mostrarlas cuando el usuario decida terminar (introduzca una línea vacía) o llegue al límite de 10 cadenas introducidas? El programa se llamará *practica13_2.c*

- (45 minutos) Revisa ahora la práctica 8 y recupera uno de los ficheros creados donde esté guardada una matriz de dos dimensiones. Realiza ahora una nueva versión del mismo donde la matriz del programa principal asociada al fichero que se lee, no tenga un tamaño fijo. Debe ser un puntero a numérico (entero o real). La nueva versión del programa, *practica13_3.c*, debe leer el tamaño real de la matriz y solicitar la memoria necesaria utilizando el puntero. ¿Cómo recorrerás ahora el puntero para simular que tienes una matriz de dos dimensiones? ¿Sería conveniente definir un nuevo tipo abstracto, *Matriz*, para facilitar su manipulación? ¿Cómo sería? ¿Cómo quedarían las operaciones de lectura (de fichero) y escritura de la matriz (en fichero o en pantalla)?

NOTA: Esta práctica contiene la labor de dos sesiones.