

Memoria dinámica

Datos estáticos: su tamaño y forma es constante durante la ejecución de un programa y por tanto se determinan en tiempo de compilación. El ejemplo típico son los arrays. Tienen el problema de que hay que dimensionar la estructura de antemano, lo que puede conllevar desperdicio o falta de memoria.

Cuando el sistema operativo carga un programa para ejecutarlo y lo convierte en proceso, le asigna cuatro partes lógicas en memoria principal: texto, datos (estáticos), pila y una zona libre. Esta zona libre (o heap) es la que va a contener los datos dinámicos, la cual, a su vez, en cada instante de la ejecución tendrá partes asignadas a los mismos y partes libres que fragmentaron esta zona, siendo posible que se agote si no se liberan las partes utilizadas ya inservibles. (La pila también varía su tamaño dinámicamente, pero la gestiona el sistema operativo, no el programador): Para trabajar con datos dinámicos necesitamos dos cosas:

1. Subprogramas predefinidos en el lenguaje que nos permitan gestionar la memoria de forma dinámica (asignación y liberación).
2. Algún tipo de dato con el que podamos acceder a esos datos dinámicos (ya que con los tipos vistos hasta ahora solo podemos acceder a datos con un tamaño y forma ya determinados).

Referencias:

http://www.lcc.uma.es/~lopez/lp2/apuntes/03-punteros/memoria_dinamica.pdf