Clasificación de la estructura de datos

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.

Datos dinámicos: su tamaño y forma es variable (o puede serlo) a lo largo de un programa, por lo que se crean y destruyen en tiempo de ejecución. Esto permite dimensionar la estructura de datos de una forma precisa: se va asignando memoria en tiempo de ejecución según se va necesitando.

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 fragmentarán 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 din´amica (asignación y liberación).

2. Algún tipo de dato con el que podamos acceder a esos datos din´amicos (ya que con los tipos vistos hasta ahora sólo podemos acceder a datos con un tamaño y forma ya determinados).

Bibliografia

1-Pablo López (2006), Memoria Dinámica, Universidad de Málaga, España.