

CAPITULO 1: INTRODUCCION

1. INTRODUCCION

Reseña Histórica.

El lenguaje C fue desarrollado por Dennis Ritchie en 1972 en los laboratorios de Bell mejorando el lenguaje B de Thompson, el cual, a su vez, deriva del lenguaje BCPL de Martin Richards.

Pero pronto, empezaron a surgir muchas implementaciones del lenguaje C, a raíz de su popularidad. Por este motivo, se hizo necesario definir un estándar que está representado hoy por el ANSI C.

Características de C.

- El lenguaje de programación C está caracterizado por ser de uso general, con una sintaxis sumamente corta y un juego de operadores muy potente.
- C es un lenguaje de "medio-bajo nivel". Esto significa "profundo", en el sentido que explota los recursos Hardware así como el manejo de las direcciones y los bits.
- No posee operaciones de entrada-salida, ni métodos de ficheros, y tampoco maneja los objetos compuestos como las cadenas de caracteres ...etc. que en general dependen del hardware. Estas operaciones se hacen por medio de funciones contenidas en librerías externas al lenguaje, lo que le confiere un alto grado de portabilidad.
- C es un lenguaje estructurado aunque no puede ser formalmente denominado así, por la razón de que no se puede declarar funciones dentro de otros. La estructura básica de un programa C se hace alrededor de la función; ésta puede tener sus propias variables locales, acceder a las variables globales del programa, y finalmente devolver valores o ninguno.
- C es un lenguaje que permite una compilación separada. Se puede partir un archivo en varios archivos más pequeños y que cada uno sea compilado por separado. Luego se enlazan entre sí junto con las rutinas de biblioteca para formar el ejecutable completo. Su ventaja reside en que cambios de código en uno de los archivos no necesita una nueva compilación del programa entero.
- Un programa en C durante su ejecución crea y usa 4 regiones de memoria lógicas diferentes; La primera región contiene el código del programa. La segunda contiene las variables globales. La tercera, es la pila (Stack) que almacena temporalmente las direcciones de las funciones, el estado de CPU,...etc. Y la cuarta llamada el montón (HEAP), que es la región libre, que sirven para reservación dinámica de memoria.
- La descripción del lenguaje se realiza siguiendo las normas del ANSI C, por lo tanto, todo lo expresado será utilizable con cualquier compilador que se adopte.

2. ANATOMIA DE UN PROGRAMA C

```
#include <stdio.h>      /* inclusión de un fichero */
main()
{
printf("Bienvenido");   /* una sentencia */
return 0;               /*termina el programa */
}
```

Función main()

main() es la función principal, indicando donde empieza el programa. El cuerpo es un conjunto de sentencias delimitadas por dos llaves. "{Cuerpo }". Cada sentencia termina con el punto y coma ";". La sentencia (return 0) termina el programa y devuelve un valor al Sistema.

Encabezado.

Las líneas anteriores a la función main() se denominan ENCABEZAMIENTO (HEADER) y son informaciones que se le suministran al Compilador.

La directiva: **#include** es una orden para compilador de leer un archivo de texto especificado en el nombre que sigue a la misma, **<stdio.h>**. El compilador reemplaza la directiva por el contenido de dicho archivo. Este fichero contiene las declaraciones de las funciones llamadas por el programa.

La directiva **#include** no es una sentencia de programa, por lo que no es necesario terminarla con el punto y coma ';'. El compilador tiene varias maneras de buscar un archivo.

#include <stdio.h>	1) Directorio de instalación C:\BC\INCLUDI
#include "stdio.h"	2) Directorio activo en el momento.
#include "a:\milibreria.h"	3) Directorio indicado.

Comentario.

Para el compilador, los comentarios son inexistentes, por lo que no generan líneas de código, permitiendo abundar en ellos tanto como se desee. En C se toma como comentario todo carácter interno a los símbolos: /* */ y pueden ocupar uno o más renglones.

```
/* este es un comentario */
/* Este es otro comentario más
    largo que el anterior */
```

3. VOCABULARIO BASICO.

Para formar los elementos básicos de un programa (constantes, variables, operadores, etc.) C utiliza los siguientes conjuntos:

Las letras: a, b,...z, y A, B, ... Z.;

Los dígitos: 0,1,...9.;

Caracteres especiales. +, -, =, (), [], &, *, %, ...etc.

Secuencia de escape: \n, \t, \r, \\....

Palabras reservadas: **main, if, while, int, for**, .etc.

son las palabras que no se pueden usar como identificador. El ANSI C cuenta con 32 palabras reservadas. Muchos compiladores de C han añadido palabras claves para explotar mejor el sistema operativo de su entorno.

Por ejemplo, varios compiladores de C incluyen nuevas palabras reservadas para gestionar la organización de memoria y las interrupciones de los microprocesadores (8088/8086) como **asm, far, near, interrupt**, .tc. **LOS IDENTIFICADORES.**

En C, los nombre de las variables, funciones y otros objetos definidos por el usuario se conocen como identificadores. La longitud de un identificador puede variar de un carácter a varios, siendo el primero una letra un símbolo de subrayado y los siguientes una combinación de letras, números y subrayado.

<code>Cont, prueba10, balance_total.</code>	1) Correctos.
<code>1cont, hola!, balance..total.</code>	2) Incorrectos.

El ANSI establece que no hay un limite en el numero de caracteres que componen la definición de los identificadores, sin embargo no necesariamente todos son significativos.

Normalmente, los ocho primeros son significativos.

Ningún identificador debe coincidir con las palabras reservadas ni con un nombre de alguna función de librería.

En C, las minúsculas y las mayúsculas se tratan como distintas;

<code>Cuenta, cuenta, CUENTA.</code>	3) identif. distintos.
--------------------------------------	------------------------