



Primeros programas

Dificultad del comienzo

En general, en cualquier materia resulta difícil el comienzo porque falta visión de conjunto. Como es necesario presentar los contenidos linealmente, hasta que no se ha desarrollado una buena cantidad de ellos no se puede ver la relación global que liga todas las partes.

Esto es especialmente cierto en programación. Para crear un programa hay que manejar simultáneamente muchos conceptos distintos. Por ello, antes de comenzar con el estudio detallado de las partes principales de la programación en C, se van a presentar en esta hoja unos pequeños programas, muy sencillos, sobre los que se explicará el proceso global de compilación. Evidentemente, según se está comentando, no se podrá comprender completamente el proceso, eso se hará más adelante, pero el objetivo es comenzar por obtener la visión global antes de estudiar los detalles.

Hola, mundo

Éste es el nombre del primer programa que aparece en el libro clásico de Kernighan y Ritchie. A partir de su publicación, es costumbre comenzar por este programa: se trata sencillamente de mostrar por pantalla un mensaje de saludo a todo el mundo. Éste es el proceso:

1. Con cualquier editor de texto se escribe el código. Se almacena en el fichero **holamundo.c**.
2. Se invoca el compilador con la orden **gcc -o holamundo holamundo.c**. Esto genera el archivo **holamundo**, que es ejecutable.
3. Se ejecuta el programa con la orden **./holamundo**.
4. Si en algún momento hay algún error, bien de compilación o de ejecución, se retoca el código fuente y se repite el proceso.

El código

A continuación se presenta el código del programa, junto con una explicación de cada línea. Es importante hacer notar que C es un lenguaje de formato libre: se pueden colocar los distintos elementos como se desee, aunque es habitual seguir consistentemente alguno de los estilos clásicos.

<code>/* Programa Hola mundo */</code>	En C se indican los comentarios con los caracteres <code>/*</code> al principio y <code>*/</code> al final. El compilador los ignora.
<code>#include <stdio.h></code>	Todas las líneas que comienzan con el carácter <code>#</code> son órdenes para el preprocesador. En ésta se indica que incluya el fichero de cabecera <code>stdio.h</code> , en el se encuentra la definición de la función <code>printf()</code> .
<code>int main (void)</code>	Los programas en C comienzan con la ejecución de la función <code>main()</code> . En esta línea se especifica que la función <code>main()</code> no va a tomar ningún parámetro (<code>void</code>) y que va a devolver un número entero (<code>int</code>).
<code>{</code>	Las instrucciones que forman la función <code>main()</code> deben estar encerradas entre llaves. Ésta es la de comienzo. Se ha metido dos espacios hacia la derecha para mejorar la legibilidad del código.
<code>printf ("Hola, mundo\n");</code>	Se invoca la función <code>printf()</code> para que envíe a la pantalla un texto. La combinación <code>\n</code> se interpreta como un salto de línea. El punto y coma especifica el fin de la sentencia.
<code>return 0;</code>	La función <code>main()</code> devuelve 0 al sistema operativo indicando que todo ha sido correcto.
<code>}</code>	Con esta llave se concluye la función <code>main()</code> y también el programa.

Raíces cuadradas

Ahora se presenta un programa más complejo, pero más útil. Se trata de invocar el nombre del programa con un parámetro. Deberán aparecer los números naturales y sus raíces cuadradas comenzando por el 1 y terminando por el parámetro. Por ejemplo, con la orden `./raices 4` se obtendrá el resultado de la derecha.

```
1 -> 1.0000
2 -> 1.4142
3 -> 1.7321
4 -> 2.0000
```

Código

A continuación se ve el código del programa, acompañado de unos comentarios:

```
/* Cálculo de raíces cuadradas */
#include <stdio.h> /* printf() */
#include <stdlib.h> /* atoi() */
#include <math.h> /* sqrt() */

int main (int argc, char *argv[])
{
    int i, Max;

    if ( argc != 2 )
        printf ("Uso: raices Número\n");

    else
    {
        Max = atoi (argv[1]);
        for ( i=1 ; i<=Max ; i++ )
            printf ("%d -> %6.4f\n", i,
                sqrt(i));
    }

    return 0;
}
```

- ◆ Las tres primeras líneas son órdenes al precompilador para incluir los ficheros de cabecera necesarios para definir las funciones que se indica en los comentarios.
- ◆ Se han dejado líneas en blanco para mejorar la legibilidad y la línea que incluye la llamada a `printf()` está dividida en dos, para acomodar el código al cuadro.
- ◆ La función `main()` admite dos parámetros: `argc` es el número de parámetros con que se invoca el programa y `argv` son sus textos.
- ◆ Además de las variables `argc` y `argv`, el programa declara las variables `i` y `Max`.
- ◆ Si el número de parámetros no es exactamente dos (el nombre del programa y el número), el programa emite un sencillo mensaje de ayuda al usuario.
- ◆ Si el número de parámetros es dos (numerados 0 y 1), se convierte el 1 en un entero usando la función `atoi()` y mediante un bucle `for` se van imprimiendo en pantalla los números y sus raíces cuadradas.

Compilación

Como en el programa se usa la función `sqrt()`, es necesario compilar añadiendo la librería matemática; por tanto, la orden correcta para compilar el programa es

```
gcc -o raices raices.c -lm
```

Ejecución

Es posible ejecutar el programa variando el parámetro que se le pase, para ir viendo cómo cambia la salida del programa; por ejemplo, `./raices 100`, `./raices 200`, etc.

Algunas ideas importantes

Después de estos dos ejemplos se pueden resaltar algunas ideas básicas:

- ◆ Las líneas que debe atender el precompilador comienzan con el carácter `#`.
- ◆ Las sentencias deben finalizar con punto y coma, pero se pueden escribir con formato libre.
- ◆ Los bloques de sentencias se encierran entre llaves.
- ◆ La ejecución del programa siempre comienza en la función `main()`, que puede recibir parámetros desde la línea de órdenes del sistema operativo.
- ◆ Existen multitud de funciones disponibles que realizan tareas útiles.
- ◆ Para usar una variable, antes hay que declararla.