



Entrada y salida

La interfaz de usuario

La creación de interfaces de usuario es una materia compleja, sobre la que existe mucha literatura técnica. Sin embargo, en estas hojas no se hará hincapié en este aspecto de la programación, para concentrarse en el proceso de transformación de datos.

Ahora se muestran las tres maneras más sencillas de relacionarse con el usuario: recibir parámetros por la línea de órdenes, escribir todo tipo de datos en la pantalla y recibir datos del teclado.

Parámetros

Cuando en un sistema operativo se ejecuta un programa, siempre se puede acompañar su nombre de más texto. El nombre del programa y el texto adicional reciben el nombre de parámetros.

En C los parámetros se numeran a partir de 0 y se reciben como cadenas de texto, pero se pueden convertir en números usando las funciones `atoi()`, `atof()` y similares.

Ejemplo

Si se invoca un programa con la línea `MiProg -p -a MiFich.c -m 4`, los seis parámetros que recibe el programa en C son:

Parámetro 0: "MiProg"; parámetro 1: "-p"; parámetro 2: "-a"; parámetro 3: "MiFich.c"; parámetro 4: "-m"; parámetro 5: "4".

Definición de main()

Para poder usar los parámetros dentro del programa hay que declarar la función `main()` de modo que su primer parámetro sea el número de parámetros y su segundo parámetro sea el conjunto de cadenas de texto de los parámetros. Es costumbre llamar `argc` (del inglés *argument counter*) al primer argumento de `main()` y `argv` (del inglés *argument values*) al segundo, pero se les puede poner otros nombres.

Un programa

El pequeño programa que se ve a la derecha escribe en pantalla todos los parámetros que recibe.

```
/* Impresión de parámetros */
#include <stdio.h> /* printf() */
int main (int argc, char *argv[])
{
    int i;

    for ( i=0 ; i<argc ; i++ )
        printf ("Parámetro %d: %s\n",
                i, argv[i]);

    return 0;
}
```

La función "printf()"

Es una función fundamental para presentar información en pantalla, ya que es muy versátil. Permite representar todo de tipo de datos, con muchas opciones.

El primer parámetro de la función es la cadena de formato. En ella se escribe cómo se desea que aparezcan en pantalla el resto de los parámetros. Todo el texto de la cadena de formato se escribirá en pantalla tal cual aparezca, salvo los códigos reservados, que se usan para indicar el tipo de las variables y sus modos de representación. Los restantes parámetros, tantos como sea necesario, son las variables que se imprimirán en el lugar de los códigos en la cadena de formato.

```
/* Ejemplo de printf() */
#include <stdio.h> /* printf() */
int main (void)
{
    float a;
    int b;
    char c;

    a = 1.0 / 4.0;
    b = 3 * 4;
    c = 'G';

    printf ("%s a = %f. b = %d. c = %c\n",
            "Resultado:", a, b, c);

    return 0;
}
```

Los códigos

Existen muchos códigos disponibles, y con muchas variantes, que permiten un ajuste muy fino. Éstos son los códigos más importantes:

- ◆ **%d** Sirve para imprimir un número entero.
- ◆ **%f** Para imprimir un número en coma flotante.
- ◆ **%c** Para imprimir un carácter.
- ◆ **%s** Para imprimir una cadena.

Ejemplo

Cuando se ejecuta el programa de más arriba se obtiene esta salida por pantalla:

Resultado: a = 0.250000. b = 12. c = H

La función "scanf()"

Esta función lee desde el teclado las entradas del usuario y las asigna a las variables que se indique. Se suele usar para leer una variable cada vez.

El primer parámetro de **scanf()** es una cadena de formato, en la que especifica el tipo de variable que se va a leer usando los mismos códigos explicados para **printf()**. El segundo parámetro es la dirección de memoria donde se está almacenando la variable. Para obtener la dirección de memoria de una variable se antepone a su nombre el carácter **&**.

Un ejemplo

El siguiente programa pide al usuario un número entero y luego escribe su cuadrado:

```
#include <stdio.h> /* printf() scanf() */

int main (void)
{
    int Numero, Cuadrado;

    printf ("Escribe un número: ");
    scanf ("%d", &Numero);

    Cuadrado = Numero * Numero;
    printf ("Su cuadrado es %d\n", Cuadrado);

    return 0;
}
```

