

# Utilización de bibliotecas escritas en C con C++

Ing José Luis MARTÍNEZ

2 de noviembre de 2021

## Bibliotecas en C

En el estudio de la programación en C, podíamos crear nuestro programa en forma modular con un archivo de encabezado ('header') y un archivo con el cuerpo de las funciones. Si bien los comandos en C pueden ser utilizados en C++, cuando se trata de ejecutar programas en C la compatibilidad no es directa, y debemos indicar explícitamente la porción del código escrita en C.

**Ejemplo.** Sea la siguiente librería creada en C para calcular el área de figuras geométricas, denominada *Areas.h* y el archivo con el cuerpo de las funciones *Areas.c*

```
/* Encabezado Areas.h */
#ifndef AREAS_H_INCLUDED
#define AREAS_H_INCLUDED
#include<math.h>
#define PI 3.141592654

/* Prototipos de funciones */
float rectangulo(float a, float b);

float trianguloBH(float a, float b);

float trianguloABC(float a, float b, float c);

float circulo(float r);

#endif // AREAS_H_INCLUDED
```

El archivo con el cuerpo de las funciones será

```
/* Biblioteca de funciones Areas.c, para el
cálculo de areas de distintas figuras geométricas */

#include "Areas.h"
//#include<math.h>
//#define PI 3.141592654

/* Rectangulo */

float rectangulo(float a, float b)
{
    return a*b;
}

/* Triangulo del que sabemos la altura */
```

```

float trianguloBH(float a, float b)
{
    return a*b/2;
}

/* triangulo del que conocemos la longitud de sus lados, se aplica la formula
de Herón  $A = \sqrt{s(s-a)(s-b)(s-c)}$ ,  $s = (a+b+c)/2$  */
float trianguloABC(float a, float b, float c)
{
    float s;
    s=(a + b + c)/2;
    return sqrt(s*(s-a)*(s-b)*(s-c));
}

/* Circulo */
float circulo(float r)
{
    return PI * r * r;
}

```

Si se desea importar la esta biblioteca en un programa en C++ nos aparecen problemas de compatibilidad y debemos indicar expresamente la porción de código escrita en C, que será utilizada en C++.

## Compatibilidad C a C++

Tomemos el archivo *usoDeAreasCpp.cpp* donde se importa la biblioteca *Areas.h* como se indica a continuación.

```

/* Desde C++ utilizamos la biblioteca areas creada en C
*/
#include<iostream>
#include "Areas.h"

int main(int argc, char **argv)
{
    float a, b, c;
    std::cout << "Inserte los valores de a, b, y c. "; std::cin >> a >> b >> c;

    std::cout << rectangulo(a, b) << std::endl;
    std::cout << trianguloBH(a, b) << std::endl;

    return EXIT_SUCCESS;
}

```

El compilador me devuelve los siguientes mensajes:

```

compatibilidadC/Areas/usoAreasCpp.cpp:11: undefined reference to 'rectangulo(float, float)'
compatibilidadC/Areas/usoAreasCpp.cpp:12: undefined reference to 'trianguloBH(float, float)'
collect2.exe: error: ld returned 1 exit status Process terminated with status 1 (1 minute(s), 39 second(s)) 3
error(s), 0 warning(s) (1 minute(s), 39 second(s))

```

Me indica que no reconoce las funciones escritas en C. Para compatibilizar ambos lenguajes en el archivo de encabezado debo escribir:

```

#ifdef __cplusplus
extern "C"
{
#endif // __cplusplus

// Prototipos de funciones escritas en C

funcion1enC;
funcion2enC;
funcion3enC;

.....

#ifdef __cplusplus
}
#endif // __cplusplus

```

El programa pregunta con `#ifdef` si el código está escrito en C++, en ese caso le indica que las funciones descritas en los prototipos están escritas en C mediante la instrucción *extern "C"*, luego abre una llave { y lo que queda dentro de las llaves serán las funciones a ser compatibilizadas con C++, al final escribe las instrucciones para cerrar la llave.

De esta forma nuestro archivo encabezado *Areas.h*, queda de la siguiente forma.

```

/* Encabezado Areas.h */
#ifndef AREAS_H_INCLUDED
#define AREAS_H_INCLUDED
#include<math.h>
#define PI 3.141592654

#ifdef __cplusplus
extern "C"
{
#endif // __cplusplus
/* Prototipos de funciones */
float rectangulo(float a, float b);

float trianguloBH(float a, float b);

float trianguloABC(float a, float b, float c);

float circulo(float r);
#ifdef __cplusplus
}
#endif // __cplusplus

#endif // AREAS_H_INCLUDED

```

Al ejecutar el programa principal ya puede reconocer las funciones escritas en C.

## 1. Utilizar funciones en C que no se encuentran en un archivo encabezado

Cuando la función a utilizar no se encuentra en un archivo encabezado, debemos incluir en nuestro archivo *main()* los prototipos de función a utilizar y preceder la declaración del comando *extern "C"*.

**Ejemplo.** Se desea calcular el volumen de una esfera cuya función se encuentra en el archivo *Volumen.c*

```
/*Programas para calcular el volumen de algunos cuerpos*/

#define PI 3.141592654
// Devuelve el volumen de un prisma de base rectangular
// a*b y altura h
float volPrismaRect(float a, float b, float h)
{
    return a*b*h;
}

// Devuelve el volumen de una esfera
float volEsfera(float r)
{
    return PI*r*r*r*4/3;
}
```

Para poder utilizar estas funciones debemos modificar el archivo *main()* de la siguiente forma.

```
/* Desde C++ utilizamos la biblioteca areas creada en C
*/
#include<iostream>
#include "Areas.h"

extern "C" float volEsfera(float r);
int main(int argc, char **argv)
{
    float a, b, c;
    std::cout << "Inserte los valores de a, b, y c. "; std::cin >> a >> b >> c;

    std::cout << rectangulo(a, b) << std::endl;
    std::cout << trianguloBH(a, b) << std::endl;
    std::cout << "Volumen de una esfera de radio r= " << a << " ,es "
        << volEsfera(a) << "m3" << endl;

    return EXIT_SUCCESS;
}
```

El compilador encuentra la función y realiza la compatibilidad con C++ para su utilización.

## Problemas

1. Durante la práctica de comunicación con el microcontrolador ATmega328 de la Arduino Uno, se escribió un programa en C que permitía programar el microcontrolador accediendo a sus registros. Tome ese programa y realice la compatibilidad para ejecutarlo en C++
2. Con la biblioteca Termios el alumno puede realizar una comunicación a través del puerto serie, realice un programa en C++ para utilizar esta biblioteca escrita en C.