**5.8. Punteros a funciones**

Un puntero a función es una variable que almacena la dirección de una función. Esta función puede ser llamada más tarde, a través del puntero. Este tipo de construcción es útil pues encapsula comportamiento, que puede ser llamado a través de un puntero. Veamos cómo funciona mediante un ejemplo sencillo que crea un puntero a una función de imprimir y lo invoca:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | #include <stdio.h>  void imprime()  {  printf("Imprimiendo un message\n");  }  int main()  {  void (\*ptr\_funct)(void)=imprime;  ptr\_funct(); //Llama a imprime  return 0;  } |

Los punteros a funciones también se usan para realizar funciones de retrollamada (callback, en inglés). El siguiente trozo de código muestra un ejemplo donde una función recibe como paramétro un puntero a función, el cual invoca cuando ha terminado su ejecución. También muestra las direcciones de memoria que ocupan las funciones, que se imprime con la opción %p de la función printf.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | #include <stdio.h>  void imprimiendo\_hola(int x)  {  printf( "[\*HOLA\_\_] Hola número %d\n", x);  }  void imprimiendo\_adios(int x, void (\*ptr\_func)())  {  printf( "[\*ADIOS\_] Adios número %d\n", x);  if(ptr\_func!=NULL)  {  ptr\_func(); //Usando el puntero para retrollamar  }  }  void func\_call\_back()  {  printf( "[\*RETRO\_] Retrollamada llamada\n");  }  int main()  { // Definimos dos punteros a funciones  void (\*ptr\_func\_1)(int)=NULL;  void (\*ptr\_func\_2)(int, void (\*call\_back\_func)() )=NULL;  //Usamos el primero  ptr\_func\_1 = imprimiendo\_hola;  printf("[\*MAIN\_\_] La dirección del primer puntero is %p\n",ptr\_func\_1);  ptr\_func\_1(3);  //Usamos el segundo  ptr\_func\_2 = imprimiendo\_adios;  printf("[\*MAIN\_\_] Using a callback función \n",ptr\_func\_2);  ptr\_func\_2(3,func\_call\_back);  return 0;  } |