



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

Marco Antonio Martínez Quintana

Asignatura:

Fundamentos de programación

Grupo:

3

No de Práctica(s):

12

Integrante(s):

Pineda Cruz Tania

*No. de Equipo de
cómputo empleado:*

No aplica

No. de Lista o Brigada:

34

Semestre:

1er semestre

Fecha de entrega:

18/01/2021

Observaciones:

CALIFICACIÓN: _____

Guía práctica de estudio 12: Funciones

Objetivo

Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

Actividades

- Implementar en un programa en C la solución de un problema dividido en funciones.
- Elaborar un programa en C que maneje argumentos en la función principal.
- En un programa en C, manejar variables y funciones estáticas.

Introducción

Un programa en lenguaje C consiste en una o más funciones. C permite tener dentro de un archivo fuente varias funciones. El nombre de la función se refiere al identificador con el cual se ejecutará la función. El valor de retorno de este indica el tipo de dato que va a regresar la función al terminar el bloque de código de la misma.

Ámbito o alcance de las variables

Las variables declaradas dentro de un programa tienen un tiempo de vida que depende de la posición donde se declaren. En C existen dos tipos: variables locales (cuando son declaradas dentro de cada función) y variables globales (se declaran fuera de cualquier función).

Argumentos para la función main

La función main también puede recibir parámetros. Debido a que la función main es la primera que se ejecuta en un programa, los parámetros de la función hay que enviarlos al ejecutar el programa.

Estático

El atributo static en una variable hace que ésta permanezca en memoria desde su creación y durante toda la ejecución del programa, lo que quiere decir que su valor se mantendrá hasta que el programa llegue a su fin.


Resultados

Código (funciones)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica12.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

subMenu.c | practica10.c | practica11.c | arreglos.c | arreglos2.c | inventario.c | practica12.c

1  #include <stdio.h>
2  #include <string.h>
3
4  /*
5   * Este programa contiene dos funciones: la función main y la función
6   * imprimir. La función main manda llamar a la función imprimir. La función
7   * imprimir recibe como parámetro un arreglo de caracteres y lo corre de fin a
8   * inicio imprimiendo cada carácter del arreglo.
9   */
10
11 // Prototipo o forma de las funciones del programa
12 void imprimir(char[]);
13
14 // Definición o implementación de la función main
15 int main ()
16 {
17     char nombre[] = "Facultad de Ingenieria";
18     imprimir(nombre);
19 }
20
21 // Implementación de las funciones del programa
22 void imprimir(char s[])
23 {
24     int tam;
25     for ( tam=strlen(s)-1 ; tam>=0 ; tam-- )
26         printf("%c", s[tam]);
27     printf("\n");
28 }
```

 Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica12.exe
aireinegnI ed datlucaF


C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

Código (Ámbito de las variables)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica12.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

subMenu.c | practica10.c | practica11.c | arreglos.c | arreglos2.c | inventario.c | practica12.c

1  #include <stdio.h>
2
3  /*
4   * Este programa contiene dos funciones: la función main y la función incremento. La
5   * función main manda llamar a la función incremento dentro de un ciclo for. La función
6   * incremento aumenta el valor de la variable enteraGlobal cada vez que es invocada.
7   */
8
9  void incremento();
10
11 // La variable enteraGlobal es vista por todas
12 // Las funciones (main e incremento)
13 int enteraGlobal = 0;
14 int main()
15 {
16     // La variable cont es local a la función main
17     for (int cont=0 ; cont<5 ; cont++)
18     {
19         incremento();
20     }
21     return 999;
22 }
23
24 void incremento()
25 {
26     // La variable enteraLocal es local a la función incremento
27     int enteraLocal = 5;
28     enteraGlobal += 2;
29     printf("global(%i) + local(%i) = %d\n", enteraGlobal, enteraLocal,
30           enteraGlobal+enteraLocal);
31 }
```

 Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc practica12.c -o practica12.exe

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica12.exe
global(2) + local(5) = 7
global(4) + local(5) = 9
global(6) + local(5) = 11
global(8) + local(5) = 13
global(10) + local(5) = 15

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

Código (argumentos función main)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica12.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

subMenu.c | practica10.c | practica11.c | aneglos.c | aneglos2.c | inventario.c | practica12.c

1  #include <stdio.h>
2  #include <string.h>
3
4  /*
5   * Este programa permite manejar los argumentos enviados al ejecutarlo.
6   */
7
8  int main (int argc, char** argv)
9  {
10     if (argc == 1)
11     {
12         printf("El programa no contiene argumentos.\n");
13         return 88;
14     }
15
16     printf("Los elementos del arreglo argv son:\n");
17     for (int cont = 0 ; cont < argc ; cont++)
18     {
19         printf("argv[%d] = %s\n", cont, argv[cont]);
20     }
21
22     return 88;
23 }
```

Simbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc practica12.c -o practica12.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica12.exe
El programa no contiene argumentos.
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

Código (variable estática)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica12.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

subMenu.c | practica10.c | practica11.c | aneglos.c | aneglos2.c | inventario.c | practica12.c

1  #include <stdio.h>
2
3  /*
4   * Este programa contiene dos funciones: la función main y la función
5   * llamarFuncion. La función main manda llamar a la función llamarFuncion dentro
6   * de un ciclo for. La función llamarFuncion crea una variable estática e imprime
7   * su valor.
8   */
9
10 void llamarFuncion();
11 int main ()
12 {
13     for (int j=0 ; j < 5 ; j++)
14     {
15         llamarFuncion();
16     }
17 }
18
19 void llamarFuncion ()
20 {
21     static int numVeces = 0;
22     printf("Esta funcion se ha llamado %d veces.\n",++numVeces);
23 }
```

Simbolo del sistema


```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>gcc practica12.c -o practica12.exe
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica12.exe
Esta funcion se ha llamado 1 veces.
Esta funcion se ha llamado 2 veces.
Esta funcion se ha llamado 3 veces.
Esta funcion se ha llamado 4 veces.
Esta funcion se ha llamado 5 veces.
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>
```

Código (función estática)

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos\practica12.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

practica10.c practica11.c arreglos.c arreglos2.c inventario.c practica12.c

1  #include <stdio.h>
2
3  /*
4   * Este programa contiene las funciones de una calculadora básica: suma, resta, producto y
5   * cociente
6   */
7
8  int suma(int,int);
9  static int resta(int,int);
10
11 int producto(int,int);
12
13 static int cociente (int,int);
14
15
16
17 int suma (int a, int b)
18 {
19     return a + b;
20 }
21
22 static int resta (int a, int b)
23 {
24     return a - b;
25 }
26
27 int producto (int a, int b)
28 {
29     return (int) (a*b);
30 }
31
32 static int cociente (int a, int b)
33 {
34     return (int) (a/b);
35 }
36
37 //##### calculadora.c #####
38 #include <stdio.h>
39
40 /*
41 * Este programa contiene el método principal, el cual invoca a las funciones
42 * del archivo funcEstatica.c.
43 */
44
45 int suma(int,int);
46 //static int resta(int,int);
47 int producto(int,int);
48 //static int cociente (int,int);
49
50 int main()
51 {
52     printf("5 + 7 = %i\n",suma(5,7));
53     //printf("9 - 77 = %d\n",resta(9,77));
54     printf("6 * 8 = %i\n",producto(6,8));
55     //printf("7 / 2 = %d\n",cociente(7,2));
56 }
```

 Símbolo del sistema

```
C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica12.exe
5 + 7 = 12
6 * 8 = 48

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>practica12.exe
5 + 7 = 12
6 * 8 = 48

C:\Users\super_000\Desktop\Lenguaje C\Ejemplos>_
```

Conclusiones

Como bien sabemos, un programa en lenguaje C consta de una o varias funciones en las cuales, se reciben parámetros de entrada y estos hacen que trabaje la función.

El desarrollo de esta práctica se me hizo un poco complicada puesto que, el desarrollo y uso que se le aplica a las funciones es algo nuevo para mí y, por lo tanto, se me complicó un poco poder entender ciertas cuestiones que se empleaban, así como la función static y el uso de este.

Sin embargo, durante el transcurso de la práctica pude ir aclarando mejor mis dudas y poder comprender de manera más sencilla la función de cada uno y llegar al final de la práctica con escasas dudas, esto consiste en seguir llevándolo a práctica hasta poder realizarlo de manera exitosa.

Referencias

Manual de las prácticas de Fundamentos de Programación, UNAM.