



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martinez Quintana

Asignatura: Fundamentos de Programacion

Grupo: 03

No de Práctica(s): 12

Integrante(s): Lopez Martinez Diana

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 28

Semestre: 1

Fecha de entrega: 24/01/2020

Observaciones:

CALIFICACIÓN: _____

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.

Introducción

Como ya se mencionó, un programa en lenguaje C consiste en una o más funciones. C permite tener dentro de un archivo fuente varias funciones, esto con el fin de dividir las tareas y que sea más fácil la depuración, la mejora y el entendimiento del código. En lenguaje C la función principal se llama main. Cuando se ordena la ejecución del programa, se inicia con la ejecución de las instrucciones que se encuentran dentro de la función main, y ésta puede llamar a ejecutar otras funciones, que a su vez éstas pueden llamar a ejecutar a otras funciones, y así sucesivamente.

Código (funciones)



fun: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>
#include <string.h>

/*
    Este programa contiene dos funciones: la función main y la función
    imprimir. La función main manda llamar a la función imprimir. La función
    imprimir recibe como parámetro un arreglo de caracteres y lo recorre de fin a
    inicio imprimiendo cada carácter del arreglo.
*/

// Prototipo o firma de las funciones del programa
void imprimir(char[]);

// Definición o implementación de la función main
int main (){
    char nombre[] = "Facultad de Ingeniería";
    imprimir(nombre);
}

// Implementación de las funciones del programa
void imprimir(char s[]){
    int tam;
    for ( tam=strlen(s)-1 ; tam>=0 ; tam-- )
        printf("%c", s[tam]);
    printf("\n");
}
```

Código (Ámbito de las variables)



funB: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>
```

```
/*
```

```
Este programa contiene dos funciones: la función main y la función incremento. La función main manda llamar a la función incremento dentro de un ciclo for. La función incremento aumenta el valor de la variable enteraGlobal cada vez que es invocada.
```

```
*/
```

```
void incremento();
```

```
// La variable enteraGlobal es vista por todas
```

```
// las funciones (main e incremento)
```

```
int enteraGlobal = 0;
```

```
int main(){
```

```
    // La variable cont es local a la función main
```

```
    for (int cont=0 ; cont<5 ; cont++){
```

```
        incremento();
```

```
    }
```

```
    return 999;
```

```
}
```

```
void incremento(){
```

```
    // La variable enteraLocal es local a la función incremento
```

```
    int enteraLocal = 5;
```

```
    enteraGlobal += 2;
```

```
    printf("global(%i) + local(%i) = %d\n", enteraGlobal, enteraLocal, enteraGlobal+enteraLocal);
```

```
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc funB.c -o funB.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>funB.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
```

ARGUMENTOS PARA LA FUNCIÓN MAIN

la firma de una función está compuesta por tres elementos: el nombre de la función, los parámetros que recibe la función y el valor de retorno de la función.

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. La firma completa de la función main es:

```
int main (int argc, char ** argv);
```

Código (argumentos función main)



funC: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>
#include <string.h>

/*
    Este programa permite manejar los argumentos enviados al ejecutarlo.
*/

int main (int argc, char** argv){
    if (argc == 1){
        printf("El programa no contiene argumentos.\n");
        return 88;
    }

    printf("Los elementos del arreglo argv son:\n");
    for (int cont = 0 ; cont < argc ; cont++){
        printf("argv[%d] = %s\n", cont, argv[cont]);
    }

    return 88;
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc funC.c -o funC.exe

C:\Users\paust\Downloads\Lenguaje c\Ejemplos>funC.exe
El programa no contiene argumentos.
```

Estático

Lenguaje C permite definir elementos estáticos. La sintaxis para declarar elementos estáticos es la siguiente:

```
static tipoDato nombre;  
static valorRetorno nombre(parámetros);
```

Código (variable estática)



funD: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>
```

```
/*
```

```
Este programa contiene dos funciones: la función main y la función  
llamarFuncion. La función main manda llamar a la función llamarFuncion dentro  
de un ciclo for. La función llamarFuncion crea una variable estática e imprime  
su valor.
```

```
*/
```

```
void llamarFuncion();
```

```
int main (){
```

```
    for (int j=0 ; j < 5 ; j++){
```

```
        llamarFuncion();
```

```
    }
```

```
}
```

```
void llamarFuncion(){
```

```
    static int numVeces = 0;
```

```
    printf("Esta funci%cn se ha llamado %d veces.\n",162,++numVeces);
```

```
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc funD.c -o funD.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>funD.exe
```

```
Esta función se ha llamado 1 veces.
```

```
Esta función se ha llamado 2 veces.
```

```
Esta función se ha llamado 3 veces.
```

```
Esta función se ha llamado 4 veces.
```

```
Esta función se ha llamado 5 veces.
```

Código (función estática)



*FunE: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
//##### funcEstatica.c #####
```

```
#include <stdio.h>
```

```
/*
```

```
Este programa contiene las funciones de una calculadora básica: suma, resta, producto y cociente.
```

```
*/
```

```
int suma(int,int);
```

```
static int resta(int,int);
```

```
int producto(int,int);
```

```
static int cociente (int,int);
```

```
int suma (int a, int b){
```

```
    return a + b;
```

```
}
```

```
static int resta (int a, int b){
```

```
    return a - b;
```

```
}
```

```
int producto (int a, int b){
```

```
    return (int)(a*b);
```

```
}
```

```
static int cociente (int a, int b){
```

```
    return (int)(a/b);
```

```
}
```



*calcula: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
//##### calculadora.c #####
```

```
#include <stdio.h>
```

```
/*
```

```
Este programa contiene el método principal, el cual invoca a las funciones  
del archivo funcEstatica.c.
```

```
*/
```

```
int suma(int,int);
```

```
//static int resta(int,int);
```

```
int producto(int,int);
```

```
//static int cociente (int,int);
```

```
int main(){
```

```
    printf("5 + 7 = %i\n",suma(5,7));
```

```
    //printf("9 - 77 = %d\n",resta(9,77));
```

```
    printf("6 * 8 = %i\n",producto(6,8));
```

```
    //printf("7 / 2 = %d\n",cociente(7,2));
```

```
}
```


Conclusion

La funciones son de suma importancia ya que nos ayudan a abstraer datos de entrada, procesos y salidas.

Bibliografia

Solano Gálvez, García Cano, Sandoval Montaña, Nakayama Cervantes, Arteaga Ricci, Castañeda Perdomo, I., 2020. Laboratorio Salas A Y B. [online] Lcp02.fi-b.unam.mx. Available at: <<http://lcp02.fi-b.unam.mx/>> [Accessed 6 April 2020].