



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): Moreno Razo Laura Mildred

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: No aplica

Semestre: 2021-1

Fecha de entrega: Lunes 11 de Enero del 2021

Observaciones:

CALIFICACIÓN: _____

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

La experiencia ha mostrado que la mejor forma de desarrollar y mantener un programa grande es construirlo a partir de piezas menores o módulos, siendo cada uno de ellos más fácil de manipular que el programa original. Esta técnica se conoce como divide y vencerás.

C fue diseñado para hacer funciones eficientes y fáciles de usar. Los programas C consisten generalmente de varias funciones pequeñas en vez de pocas grandes.

Una función es un fragmento de código que realiza una tarea bien definida. Las funciones se invocan mediante una llamada de función. La llamada de función especifica el nombre de esta y proporciona información (en forma de argumentos) que la función llamada necesita a fin de llevar acabo su tarea.

El nombre de la función es cualquier identificador valido. El tipo de regreso es el tipo del resultado que se regresa al llamador. Se coloca como tipo void cuando la función no regresa un valor. Si la función regresa un resultado el enunciado return expresión devuelve el valor de expresión al llamador y termina la ejecución de la función.

Todas las variables declaradas en las definiciones de función son variables locales (son conocidas solo en la función en la cual están definidas).

Actividad 1

En este código se hace uso de dos funciones, la principal en la cual se obtiene la cadena de caracteres que queremos imprimir y la función "imprimir" que es invocada dentro de main para realizar el trabajo de voltear la palabra, observamos que uno de los parametros solicitados por "imprimir" es una cadena de caracteres para poder ejecutarse, y no solicita ningun valor de retorno, por lo que se coloca "void".

```
C:\Users\mildr\Desktop\C>p12.exe  
aireinegnI ed datlucaF  
  
C:\Users\mildr\Desktop\C>
```

Actividad 2

Se hace uso de dos funciones, main e incremento, en main se declara una variable local llamada cont, la cual sirve para invocar cierta cantidad de veces la función incremento dentro de un ciclo for (en este caso va de 0 a 4 por lo que se repite 5 veces), en incremento se declara una variable local llamada enteraLocal, la cual se mantiene constante en 5 y se suma a enteraGlobal, que se va incrementando de 2 en 2 cada que se usa la función incremento.

```
C:\Users\mildr\Desktop\C>p12.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\mildr\Desktop\C>
```

Actividad 3

Como se menciona en el primer lugar del arreglo esta el nombre del programa, y a continuación los demás argumentos introducidos, como sabemos que las cadenas de caracteres son arreglos, se accede a los valores guardados para imprimirlos mediante un ciclo for, cuya repetición esta dada por las variables `cont` y `argc`, que nos indica el largo de la palabra.

```
C:\Users\mildr\Desktop\C>p12.exe noche holap adios
Los elementos del arreglo argv son:
argv[0] = p12.exe
argv[1] = noche
argv[2] = holap
argv[3] = adios
```

De lo contrario, si no se introduce ningún argumento se muestra el siguiente mensaje.

```
C:\Users\mildr\Desktop\C>p12.exe
El programa no contiene argumentos.
```

Actividad 4

Se utilizan dos funciones, `main` y `llamarFunción`, en `main` se invoca cierta cantidad de veces la función "llamarFuncion" mediante un ciclo for, debido a que la variable local `numValor` de `llamarFuncion` se ha declarado como estática, permanece ahí mediante toda la ejecución y mantiene su valor en memoria. Cada vez que es invocada, se imprime aumentando su valor en 1, por lo que ya no es necesario escribir `numValor = numValor + 1` pues el valor se conserva.

```
C:\Users\mildr\Desktop\C>gcc p12.c -o p12.exe

C:\Users\mildr\Desktop\C>p12.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.
```

Actividad 5

Este programa es un poco más avanzado, ya que consta de dos archivos, que son compilados al mismo tiempo, esto permite acceder a las funciones que definen las operaciones dentro de la calculadora invocandolas en la funcion main del programa “calculadora”, cumplen su utilidad y finaliza su ejecución,excepto en la operación resta y cociente, pues las funciones fueron declaradas como estáticas lo cual no permite se acceda a ellas desde fuera del archivo donde se encuentran.

El programa contenía un error para compilar,pues faltaba un corchete de cierre en la función “producto”.

```
C:\Users\mildr\Desktop\C>gcc funEst.c calcul.c -o exe
C:\Users\mildr\Desktop\C>exe
5 + 7 = 12
6 * 8 = 48
```

Como observamos al intentar invocar las funciones estáticas desde fuera, marca error al compilar.

```
C:\Users\mildr\Desktop\C>gcc funEst.c calcul.c -o exe
C:\Users\mildr\Desktop\C>gcc funEst.c calcul.c -o exe
calcul.c:8:12: warning: 'resta' used but never defined
   8 | static int resta(int,int);
     |               ^~~~~~
calcul.c:10:12: warning: 'cociente' used but never defined
  10 | static int cociente (int,int);
     |               ^~~~~~
c:/mingw/bin/../lib/gcc/mingw32/9.2.0/../../../../mingw32/bin/ld.exe: C:\Users\mildr\AppData\Local\Temp\ccTwt2T0.o:calcul.c:(.text+0x42): undefined reference to `resta'
c:/mingw/bin/../lib/gcc/mingw32/9.2.0/../../../../mingw32/bin/ld.exe: C:\Users\mildr\AppData\Local\Temp\ccTwt2T0.o:calcul.c:(.text+0x8a): undefined reference to `cociente'
collect2.exe: error: ld returned 1 exit status
C:\Users\mildr\Desktop\C>_
```

Conclusión

El tema visto en la práctica nos ayuda a tener herramientas para la solución de problemas mediante el uso de C, donde como vimos, podemos establecer varias funciones cortas que dividen programas extensos e implementarlas en el momento necesario para que cumplan un propósito bien definido.

Aprendimos su sintaxis y uso general, además de como utilizarla y parámetros que podemos solicitar, lo que también nos llevó a comprender ciertas funciones de programas anteriormente realizados donde se utilizaba la función main y un valor de retorno 0 para salir de ella, que eran instrucciones las cuales tal vez anteriormente no comprendíamos del todo su significado y funcionamiento.

Bibliografía

- INCO. (2019). El lenguaje C [Ebook] (pp. 1,3). Montevideo, Uruguay: fing.edu.uy. Recuperado de:
<https://www.fing.edu.uy/tecnoinf/mvd/cursos/prinprog/material/teo/prinprog-teorico06.pdf>