



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Pimentel

Asignatura: Fundamentos de programación

Grupo: 3

No de Práctica(s): 12

Integrante(s): Colonia Montero Sonia

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 7455

Semestre: 2020-1

Fecha de entrega: 04/11/2019

Observaciones:

CALIFICACIÓN: _____

Introducción

Cuando una acción se quiere repetir en diversas ocasiones dentro de un código se usan las funciones. Dichas funciones deben ir fuera de `int main` para que al momento de elaborar el código solo se llamen y se relize la operación programada en dicha función. Sin embargo, no es necesario que vayan antes del `int main`, ya que se pueden utilizar simplemente firmas o prototipos de una función antes de `int main` y poner el resto de la función al final del código. A lo largo del presente documento se mostrará la diferencia entre una firma de función y su implementación, la importancia de usar tanto a las funciones como a su firma y la manera correcta de elaborar funciones en el programa C.

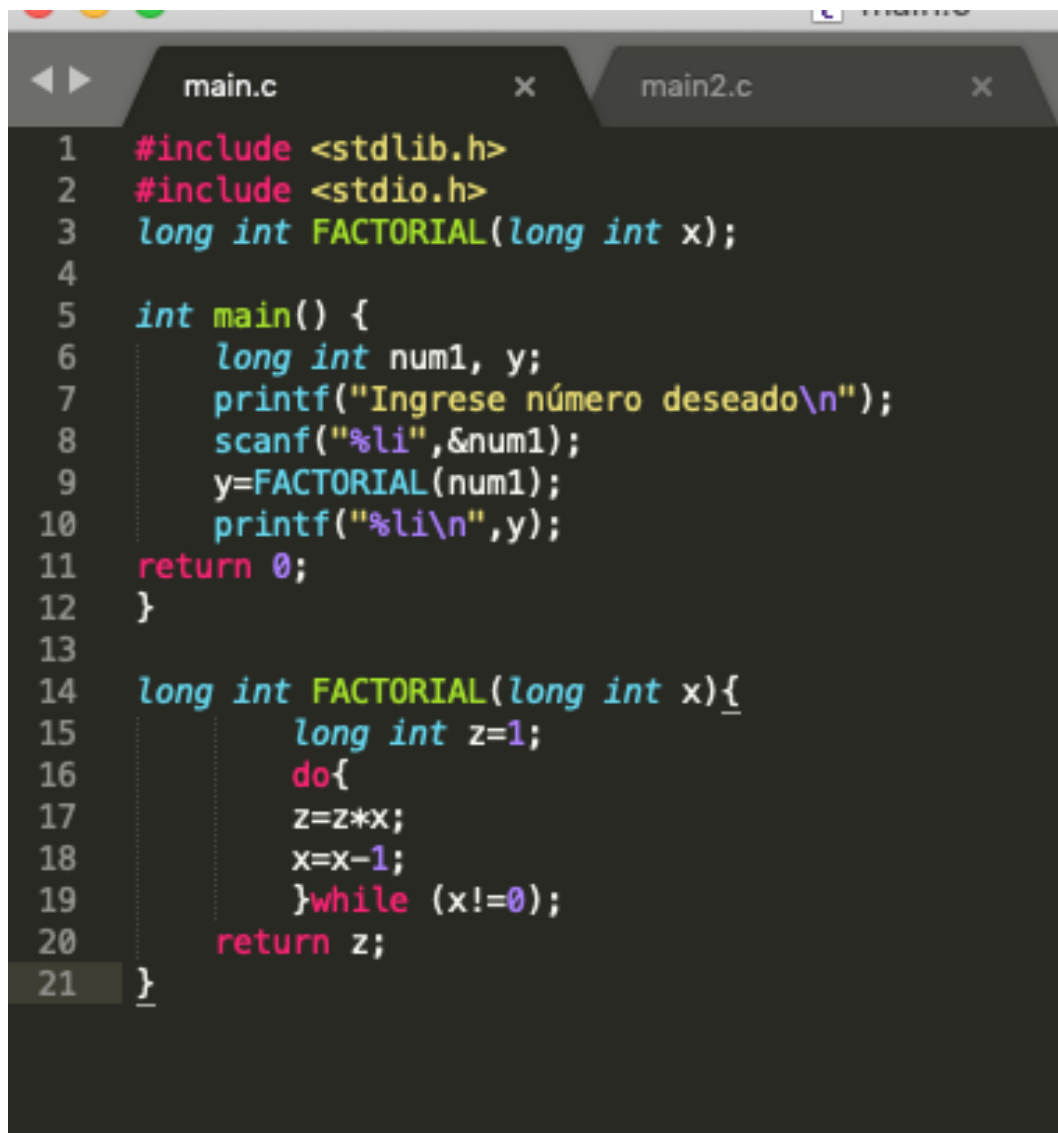
Objetivo

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.

Desarrollo

Todas las actividades tienen los prototipos de sus funciones, y sus funciones implementadas después del `main`.

En primer lugar, se creó un programa que tiene una función que regresa el factorial de un número de entrada.

A screenshot of a code editor with two tabs: 'main.c' and 'main2.c'. The 'main.c' tab is active and shows C code for calculating a factorial. The code includes headers for `stdlib.h` and `stdio.h`, declares a `long int FACTORIAL(long int x);` function, and implements it in `main()` and a separate function block. The `main()` function prompts the user for a number, reads it, calls `FACTORIAL`, and prints the result. The `FACTORIAL` function uses a `do-while` loop to calculate the factorial. The code is as follows:

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  long int FACTORIAL(long int x);
4
5  int main() {
6      long int num1, y;
7      printf("Ingrese número deseado\n");
8      scanf("%li",&num1);
9      y=FACTORIAL(num1);
10     printf("%li\n",y);
11     return 0;
12 }
13
14 long int FACTORIAL(long int x){
15     long int z=1;
16     do{
17         z=z*x;
18         x=x-1;
19     }while (x!=0);
20     return z;
21 }
```

Posteriormente, creó un programa que tiene una función que regresa el resultado de la serie

$$\sum_{x=1}^n \frac{x!}{x}$$

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  long int FACTORIAL(long int x);
4  long int SERIE(long int x);
5
6  int main() {
7      long int num1, res;
8      printf("Ingrese número deseado\n");
9      scanf("%li",&num1);
10     res=SERIE(num1);
11     printf("%li\n",res);
12     return 0;
13 }
14
15 long int FACTORIAL(long int x){
16     long int z=1;
17     do{
18         z=z*x;
19         x=x-1;
20     }while (x!=0);
21     return z;
22 }
23
24 long int SERIE(long int x){
25     long int y;
26     long int s=1;
27     long int w=0;
28     do{
29         y=FACTORIAL(s)/s;
30         w=w+y;
31         s++;
32     }while (s<=x);
33     return w;
34 }
```

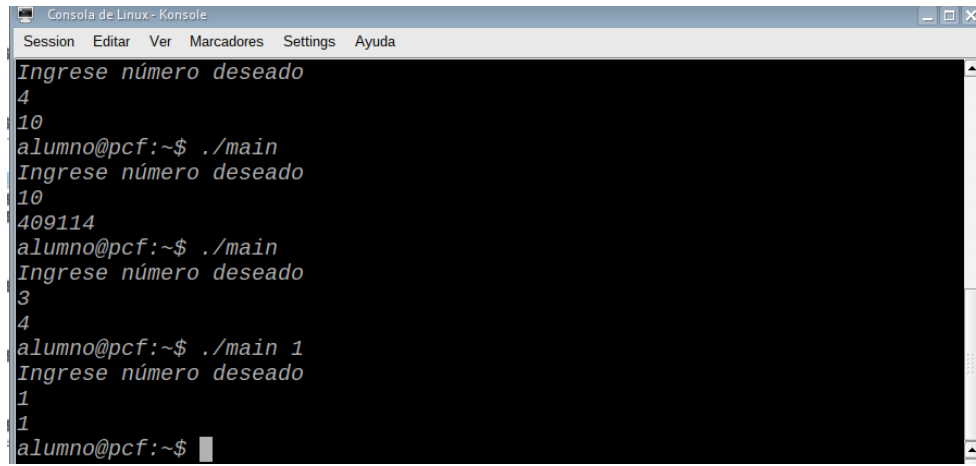
Resultados

Todos los programas se compilaron y se leyeron.

Factorial

```
Escritorio — -bash — 80x14
[Lituania09:desktop fp03alu08$ gcc main.c -o main
[Lituania09:desktop fp03alu08$ ./main
Ingrese número deseado
3
6
[Lituania09:desktop fp03alu08$ ./main
Ingrese número deseado
5
120
[Lituania09:desktop fp03alu08$ ./main
Ingrese número deseado
10
3628800
Lituania09:desktop fp03alu08$
```

Serie



```
Consola de Linux - Konsole
Session  Editar  Ver  Marcadores  Settings  Ayuda

Ingrese número deseado
4
10
alumno@pcf:~$ ./main
Ingrese número deseado
10
409114
alumno@pcf:~$ ./main
Ingrese número deseado
3
4
alumno@pcf:~$ ./main 1
Ingrese número deseado
1
1
alumno@pcf:~$
```

Conclusión

Usar funciones dentro de los códigos computacionales permite no tener que volver a escribir la misma operación o acción a realizar, sino que simplemente llamar a la función programada con anterioridad. Por otra parte, usar la firma o prototipo de función es muy útil para tener un código mucho más organizado. En lugar de tener en la parte de arriba del código mucho texto, solo se verán las funciones que se tienen y en la parte de abajo estará todo su desarrollo, lo que hace mucho más práctico saber que funciones se tienen y cuales conviene usar.