



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 07

Integrante(s): Ramirez Garcia Diego Andres

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

No. de Lista o Brigada: 35

Semestre: Primero

Fecha de entrega: 20/Noviembre/2020

Observaciones: Desarrollo y explicación adecuados para la comprensión
de la sintaxis básica del lenguaje C.

CALIFICACIÓN: _____

Fundamentos de Lenguaje C

Objetivo

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo *secuencia*, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

Introducción

Una vez estructurado un digrama de flujo y un pseudocódigo podemos proceder a la codificación, es decir, la generación de nuestro código fuente en algun lenguaje de programaión; para posteriromente crear el ejecutable y finalmente correrlo en la arquitectura correspondiente a la maquina.

Se considero el lenguaje de programación C, ya que es un lenguaje de propósito general que ofrece como ventajas economía de expresión, control de flujo y estructuras de datos y un conjunto de operadores. Y ademas esta basado en el paradigma estructurado, es decir, que todo programa puede desarrollarse utilizando únicamente tres instrucciones de control: Secuencia, Selección e Iteración.

El lenguaje C considerado como un lenguaje de Nivel Medio por algunos autores por su simultaniedad de lenguaje de Alto Nivel y de Bajo Nivel tiene una cierta estructura y sintaxis. Principalmente dentro de esta nos encontramos que todo tiene que estar declarado dentro de la función principal `main()`, y a su vez la función principal puede contener sentencias, estructuras de control y comentarios. Dentro de las sentencias se encuentran la declaración y/o asignación de variables, la realización de operaciones básicas, y las llamadas a funciones.

```
#include <stdio.h>

int main() {

    [modificadores] tipoDeDato identificador [= valor];

    printf("[texto] : %[identificador de formato] / [carácter de escape]", [variable]);

    return 0;
}
```

La biblioteca estándar en C

Estructura y elementos de una variable

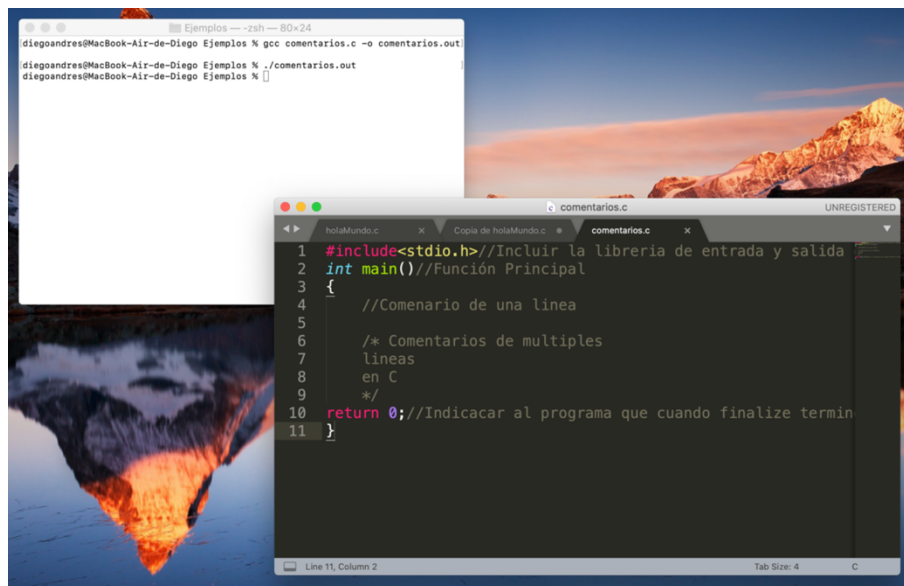
Función para imprimir con formato en el monitor.

Desarrollo de Actividades

Actividad 1.

Es una buena práctica en cualquier lenguaje de programación realizar comentarios para documentar el programa. La primer actividad es generar un programa cuyo contenido incluya únicamente comentarios y posteriormente comentar el programa de la práctica número seis.

1. Comentarios.c



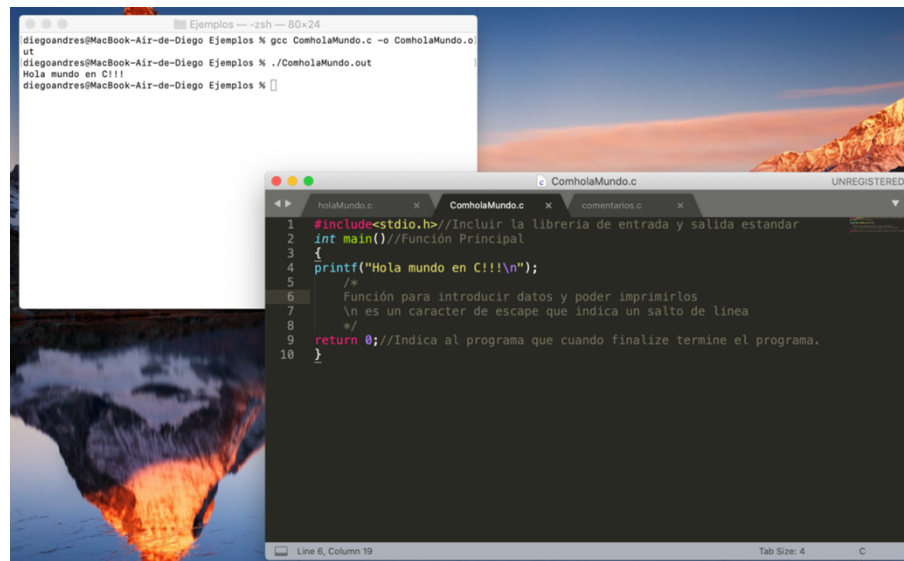
The screenshot shows a terminal window and a code editor. The terminal window displays the following commands and output:

```
diegoandres@MacBook-Air-de-Diego Ejemplos % gcc comentarios.c -o comentarios.out
diegoandres@MacBook-Air-de-Diego Ejemplos % ./comentarios.out
diegoandres@MacBook-Air-de-Diego Ejemplos %
```

The code editor shows the content of 'comentarios.c':

```
1 #include<stdio.h>//Incluir la libreria de entrada y salida
2 int main()//Función Principal
3 {
4     //Comentario de una linea
5
6     /* Comentarios de multiples
7     lineas
8     en C
9     */
10    return 0;//Indicacar al programa que cuando finalize termin
11 }
```

2. HolaMundo. C (con comentarios)



The screenshot shows a terminal window and a code editor. The terminal window displays the following commands and output:

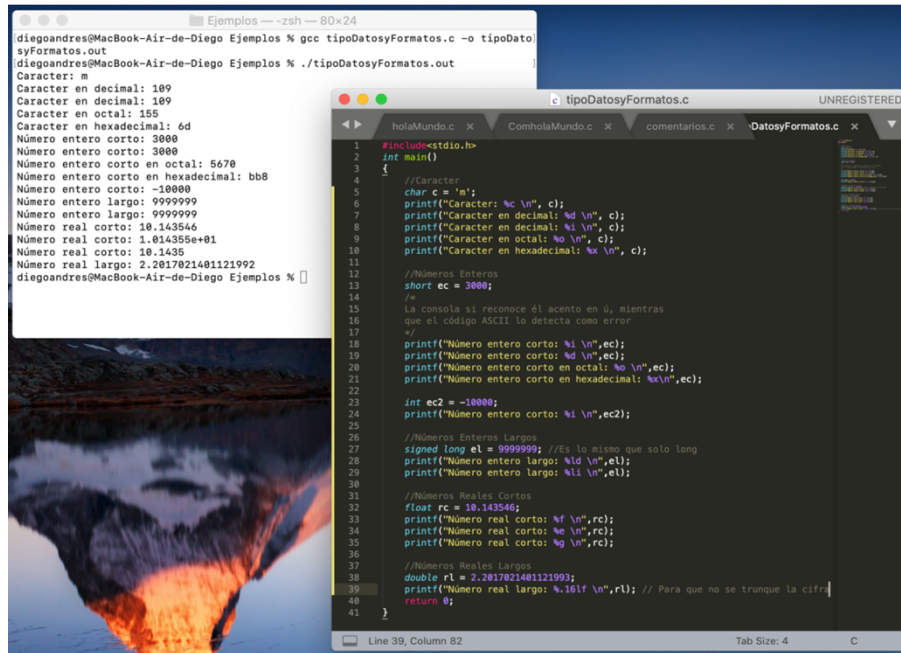
```
diegoandres@MacBook-Air-de-Diego Ejemplos % gcc ComholaMundo.c -o ComholaMundo.out
diegoandres@MacBook-Air-de-Diego Ejemplos % ./ComholaMundo.out
Hola mundo en C!!!
diegoandres@MacBook-Air-de-Diego Ejemplos %
```

The code editor shows the content of 'ComholaMundo.c':

```
1 #include<stdio.h>//Incluir la libreria de entrada y salida estandar
2 int main()//Función Principal
3 {
4     printf("Hola mundo en C!!!\n");
5
6     /*
7     Función para introducir datos y poder imprimirlos
8     \n es un caracter de escape que indica un salto de linea
9     */
10    return 0;//Indica al programa que cuando finalice termine el programa.
11 }
```

Actividad 2.

Realizar una prueba para visualizar el comportamiento de cada uno de los especificadores de formato. (%i, %d, %o, %x, %ld, %li, %e, %f, %g, %lf)

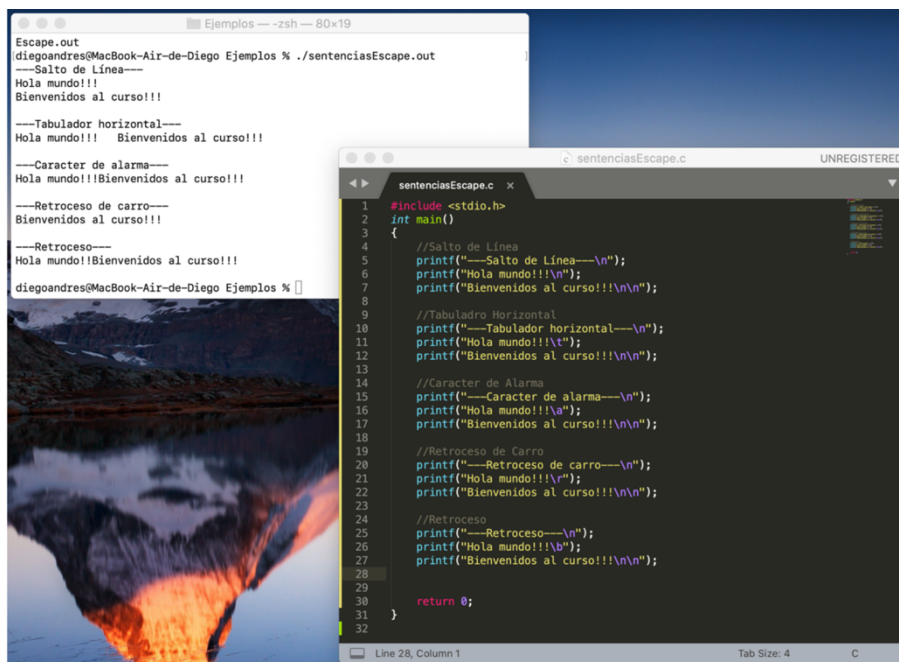


```
diegoandres@MacBook-Air-de-Diego Ejemplos % gcc tipoDatosFormatos.c -o tipoDatosFormatos.out
diegoandres@MacBook-Air-de-Diego Ejemplos % ./tipoDatosFormatos.out
Caracter: m
Caracter en decimal: 109
Caracter en decimal: 109
Caracter en octal: 155
Caracter en hexadecimal: 6d
Número entero corto: 3000
Número entero corto: 3000
Número entero corto en octal: 5670
Número entero corto en hexadecimal: bb8
Número entero corto: -10000
Número entero largo: 9999999
Número entero largo: 9999999
Número real corto: 10.143546
Número real corto: 1.014355e+01
Número real corto: 10.1435
Número real largo: 2.2017021401121992
diegoandres@MacBook-Air-de-Diego Ejemplos %
```

```
1 //Caracter
2 char c = 'm';
3 printf("Caracter: %c\n", c);
4 printf("Caracter en decimal: %d\n", c);
5 printf("Caracter en octal: %o\n", c);
6 printf("Caracter en hexadecimal: %x\n", c);
7
8 //Números Enteros
9 short ec = 3000;
10 printf("Número entero corto: %i\n", ec);
11 printf("Número entero corto: %d\n", ec);
12 printf("Número entero corto en octal: %o\n", ec);
13 printf("Número entero corto en hexadecimal: %x\n", ec);
14
15 int ec2 = -10000;
16 printf("Número entero corto: %i\n", ec2);
17
18 //Números Enteros Largos
19 signed long el = 9999999; //Es lo mismo que solo long
20 printf("Número entero largo: %ld\n", el);
21 printf("Número entero largo: %li\n", el);
22
23 //Números Reales Cortos
24 float rc = 10.143546;
25 printf("Número real corto: %f\n", rc);
26 printf("Número real corto: %e\n", rc);
27 printf("Número real corto: %g\n", rc);
28
29 //Números Reales Largos
30 double rl = 2.2017021401121992;
31 printf("Número real largo: %lf\n", rl); // Para que no se trunque la cifra
32 return 0;
```

Actividad 3

Realizar una prueba con base en la práctica holaMundo.c aplicando cada uno de las sentencias de escape.



```
diegoandres@MacBook-Air-de-Diego Ejemplos % ./sentenciasEscape.out
—Salto de Línea—
Hola mundo!!!
Bienvenidos al curso!!!

—Tabulador horizontal—
Hola mundo!!! Bienvenidos al curso!!!

—Caracter de alarma—
Hola mundo!!!Bienvenidos al curso!!!

—Retroceso de carro—
Bienvenidos al curso!!!

—Retroceso—
Hola mundo!!Bienvenidos al curso!!!
diegoandres@MacBook-Air-de-Diego Ejemplos %
```

```
1 //Salto de Línea
2 printf("—Salto de Línea—\n");
3 printf("Hola mundo!!!\n");
4 printf("Bienvenidos al curso!!!\n\n");
5
6 //Tabulador Horizontal
7 printf("—Tabulador horizontal—\n");
8 printf("Hola mundo!!!\t");
9 printf("Bienvenidos al curso!!!\n\n");
10
11 //Caracter de Alarma
12 printf("—Caracter de alarma—\n");
13 printf("Hola mundo!!!\a");
14 printf("Bienvenidos al curso!!!\n\n");
15
16 //Retroceso de Carro
17 printf("—Retroceso de carro—\n");
18 printf("Hola mundo!!!\b");
19 printf("Bienvenidos al curso!!!\n\n");
20
21 //Retroceso
22 printf("—Retroceso—\n");
23 printf("Hola mundo!!!\b");
24 printf("Bienvenidos al curso!!!\n\n");
25 return 0;
```

Conclusiones

El lenguaje C representa una gran herramienta para los ingenieros en computación y la industria, por la versatilidad que tiene todos los programas basados en C de correr prácticamente en la mayoría de los sistemas operativos, así como su factibilidad para poder programar en cualquier entorno siempre y cuando el sistema tenga un compilador para C.

Los comentarios son en la mayoría de las veces son un apoyo para la comprensión y lectura del programa, ya que en un principio nos ayuda a repasar como desarrolladores la sintaxis y estructura de un lenguaje de programación, y por el otro lado genera un material de apoyo para quienes apenas van incursionando en la programación.

Para culminar, el tipo de dato de una variable es imprescindible para el resultado del programa, ya que a partir de su definición, especificamos en la impresión del dato el especificador del formato, puesto que cada tipo de dato posee sus respectivos especificadores.

Referencias

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.

Carlos Guadalupe (2013). Aseguramiento de la calidad del software (SQA). [Figura 1]. Consulta: Junio de 2015. Disponible en:
<https://www.mindmeister.com/es/273953719/aseguramiento-de-la-calidad-del-software-sqa>