Programación I

Práctica #1 Curso 2024-2025 1º curso

Empezando a programar en C

Introducción

En esta práctica continuaremos practicando con las herramientas disponibles en el sistema operativo Linux.

También seguiremos aprendiendo a utilizar las herramientas básicas para codificar nuestros programas en lenguaje C: `emacs` y `gcc`.

Además, empezaremos a manejar los mecanismos básicos que nos proporcionan las librerías del lenguaje C para:

- la entrada de información por teclado, que es la función `scanf()`, y
- la salida de información a través de la pantalla, que es la función `printf()`.

Por último, empezaremos a trabajar con algunos de los tipos de datos de C: los tipos enteros, reales, caracteres y las cadenas de caracteres.

Objetivos

Esta práctica tiene como objetivos fundamentales:

- Seguir familiarizándose con el entorno de trabajo del laboratorio.
- Reforzar el aprendizaje de los comandos básicos del sistema operativo Linux.
- Reforzar el aprendizaje de los pasos necesarios para editar, compilar y ejecutar un programa en C.
- Empezar a trabajar con los mecanismos de entrada y salida de C
- Empezar a trabajar con los tipos de datos de C.

Planificación

Sesión 1 de laboratorio (6 horas):

1. Realización de los ejercicios 1 y 2.
2. Realización de los ejercicios 3 y 4.
3. Realización de los ejercicios 5 y 6.
4. Realización del ejercicio 7.
5. Realización de los ejercicios 8 y 9.
6. Realización del ejercicio 10.

Trabajo fuera del aula:

- Terminar los ejercicios no completados en el aula.
- Realización de los diagramas de flujo indicados en el anexo I.
- Diseño y codificación de los programas indicados en el anexo II.

Codificando un programa básico en C

Modificando un programa

Para modificar el comportamiento de un programa debemos modificar el fichero que contiene su código fuente con el editor y, a continuación, volver a compilarlo.

Ejercicio 1:

Crea un directorio `pr01` en tu cuenta (si no lo has creado ya previamente).

Entra en ese directorio y utiliza el comando *cp* de la *shell* para hacer una copia del código del `ej3.c` de la práctica anterior:

```
usuario@maquina:~/pr01$ cp ../pr00/ej3.c ej1.c
```

Modifica el programa del ejercicio anterior eliminando el "\n".

Ejecútalo y comprueba qué significado tienen estos caracteres.

También podemos utilizar el código fuente de un programa como base para hacer otro.

Ejercicio 2:

```
usuario@maquina:~/pr01$ cp ../pr00/ej3.c ej2.c
```

Modifica tu programa para que el mensaje que sale por pantalla tenga dos líneas.

En la primera aparecerá "Adios" y en la segunda "Hasta luego"

Errores de compilación

Cuando cometamos un error de codificación en nuestro código fuente, no se generará el fichero ejecutable y el compilador nos indicará el error o errores que ha encontrado, y que han impedido la compilación.

Ejercicio 3:

```
usuario@maquina:~/pr01$ cp ej2.c ej3.c
```

Elimina el ";" al final de una instrucción del código fuente del ejercicio anterior. Comprueba lo que ocurre cuando intentas compilar el programa.

Conversión de tipos

En estos ejercicios comenzaremos a practicar con las conversiones automáticas entre los tipos de datos numéricos (`double`, `float`, `int` y `short`) y sus implicaciones: ¿qué pasa cuando a una variable de un tipo le asignamos el valor de una variable de otro tipo?

Ejercicio 4:

Realiza lo siguiente:

1. Escribe un programa:

```
usuario@maquina:~/pr01$ emacs ej4.c &
```

con el siguiente código:

```
#include <stdio.h>

int main (void) {

    int entero;
    float real;

    entero = 4;
    real = entero/5;
    printf ("Cociente = %.2f\n", real);

    real = (1.0*entero)/5;
    printf ("Division 1 = %.2f\n", real);

    real = entero/5.0;
    printf ("Division 2 = %.2f\n", real);

    real = 4.7;
    entero = real;
    printf ("Entero = %d\n", entero);

    return 0;
}
```

2. Ejecuta el programa y explica los resultados obtenidos.

Ejercicio 5:

Realiza lo siguiente:

1. Escribe un programa:

```
usuario@maquina:~/pr01$ emacs ej5.c &
```

con el siguiente código:

```
#include <stdio.h>

int main (void) {

    short enteroCorto = 30000;
    int entero = 3000000;
    float real = 3e9;

    printf ("enteroCorto: %d\n", enteroCorto);
    printf ("entero: %d\n", entero);
    printf ("real en punto flotante: %.2f\n", real);
    printf ("real en notación cientifica: %.2e\n\n", real);

    enteroCorto = entero;
    printf ("enteroCorto: %hd\n\n", enteroCorto);

    entero = real;
    printf ("entero: %d\n\n", entero);

    entero = 3000000;
    real = entero;
    printf ("entero: %d\n", entero);
    printf ("real: %.2f\n\n", real);

    entero = 1234567892;
    real = entero;
    printf ("entero: %d\n", entero);
    printf ("real: %.2f\n", real);
    printf ("real: %.9e\n", real);

    return 0;
}
```

2. Ejecuta el programa y explica los resultados obtenidos.

Ejercicio 6:

Realiza lo siguiente:

1. Escribe un programa:

```
usuario@maquina:~/pr01$ emacs ej6.c &
```

con el siguiente código:

```
#include <stdio.h>

int main (void) {

    float real;
    double realDoble;

    real = 25.50;
    realDoble = 25e+100;
    real = realDoble * real;
    printf ("Usando float: %.2f\n\n", real);

    real = 25.50;
    realDoble = realDoble * real;
    printf ("Usando double: %.2f\n", realDoble);
    printf ("En notación científica: %.4e\n", realDoble);

    return 0;
}
```

2. Ejecuta el programa y explica los resultados obtenidos.

Desarrollando programas sencillos

Lectura y escritura de datos

En estos ejercicios avanzaremos en la utilización de las funciones básicas de entrada de datos por teclado, `fscanf()`, y de salida de datos por pantalla, `fprintf()`.

Veremos cómo la utilización de funciones de biblioteca facilita la realización de muchas operaciones.

Ejercicio 7:

Escribe un programa:

```
usuario@maquina:~/pr01$ emacs ej7.c &
```

Este programa, primero, solicita al usuario/a que introduzca un número **entero**, para lo que hará uso de la función `scanf()` o de `fscanf()`:

```
Dame un entero: 75
```

Y, después, calcula la raíz cuadrada del número (que será un número real), y muestra el resultado por pantalla, con el formato que a continuación se muestra, para lo que hará uso de `printf()` o de `fprintf()`:

```
RAIZ:      8.66
```

Es decir, alineado a la derecha, en un campo de 10 posiciones de anchura, y con 2 decimales.

Para resolver este ejercicio necesitamos utilizar la función `sqrt()` (raíz cuadrada) de la biblioteca de funciones matemáticas. Comprueba qué te dice la página de manual de esta función:

```
usuario@maquina:~/pr01$ man sqrt
```

Ejercicio 8:

Escribe un programa:

```
usuario@maquina:~/pr01$ emacs ej8.c &
```

Este programa, primero, solicita al usuario/a que introduzca un **carácter**:

```
Dame un caracter: w
```

Y, después, convierte dicho carácter a mayúsculas y lo muestra por pantalla:

```
La letra w en mayusculas es W
```

Para resolver este ejercicio necesitamos utilizar la función de biblioteca `toupper()`

```
usuario@maquina:~/pr1$ man toupper
```

Ejercicio 9:

Escribe un programa:

```
usuario@maquina:~/pr01$ emacs ej9.c &
```

Este programa, primero, solicita al usuario/a que introduzca una **cadena** de caracteres:

```
Dame una cadena: Programacion
```

Y, después, calcula y muestra por pantalla la longitud de dicha cadena:

```
La cadena tiene 12 caracteres
```

Para resolver este ejercicio necesitamos utilizar la función `strlen()` de la biblioteca de cadenas:

```
usuario@maquina:~/pr1$ man strlen
```

Ejercicio 10:

Escribe un programa (`ej10.c`) que, primero, solicite al usuario/a que introduzca la base y la altura de un **triángulo** rectángulo:

```
Dame la base: 15
Dame la altura: 9
```

Y, después, calcule su área y perímetro y muestre el resultado por pantalla con el siguiente formato:

```
Altura:       9    cm
Base:        15    cm
Area:        67.50 cm x cm
Perimetro:   41.49 cm
```

Haz que todos los resultados queden correctamente alineados. La base y la altura deben ser números enteros positivos.

La resolución de este ejercicio se simplifica utilizando la función `sqrt()` (raíz cuadrada) de la biblioteca de funciones matemáticas.

Resumen

Los principales resultados esperados de esta práctica son:

- Ser capaces de editar, compilar y ejecutar programas en C.
- Conocer los mecanismos de entrada por teclado y salida por pantalla de C.
- Empezar a trabajar con los tipos básicos de datos del lenguaje C.

Como posibles líneas de trabajo adicional (opcional) del alumno, se proponen las siguientes:

- Experimentar con las diferentes funcionalidades que nos proporciona *emacs* a través de sus menús.
- Practicar la elaboración de diagramas de flujo y la codificación de programas sencillos, como son los que aparecen en los Anexos I y II,

Anexo I. Ejercicios de algoritmos y diagramas de flujo

Indica los pasos del algoritmo y representa el diagrama de flujo[1] para los siguientes problemas **(para todos los casos, los datos se piden por teclado al usuario/a y el resultado se muestra por pantalla):**

1. Media artitmética de dos números positivos.
2. Cálculo del área y el perímetro de un triángulo rectángulo.
3. Suma de 10 números.
4. Multiplicación de dos números positivos a base de sumas.
5. División de dos números positivos a base de restas. Cálculo del cociente y el resto.
6. Escritura de los N primeros números pares. El valor N es el dato de entrada del problema.
7. Cálculo de 2^N, siendo N un número ≥ 0.
8. Cálculo del máximo de un conjunto indeterminado de números positivos. El/La usuario/a indica la finalización introduciendo un número negativo.
9. Dados tres números, mostrarlos ordenados de mayor a menor.
10. Cálculo del porcentaje de números pares de un conjunto de N números positivos.
11. Media aritmética de un número indeterminado de números positivos. El/La usuario/a indica la finalización introduciendo un número negativo.
12. Lectura de 10 números y cálculo del producto de los impares.
13. Dado un año indicar si es o no bisiesto.
14. Dado un número con tres dígitos indicar si es capicúa.
15. Cálculo del precio con IVA de un artículo dado, a partir del precio sin IVA. Escribir el resultado con 2 decimales.
16. Dado un carácter, determinar si se trata de una letra, de un dígito, o de ninguno de los dos.
17. Cálculo de la hipotenusa de un triángulo dados los dos catetos.
18. Dada la nota de las pruebas parciales (NPP) y la del examen teórico final (ETF) de Programación I, calcular la nota mínima que es necesario obtener en el examen final de laboratorio (EFL) para superar la materia.
19. Dados dos números indicar si el primero es múltiplo del segundo.

[1] Puedes utilizar alguna herramienta como DIA (http://live.gnome.org/Dia/), o incluso PowerPoint, o bien hacer el ejercicio "a mano".

Anexo II. Ejercicios de programación

Escribe programas en C que realice las operaciones necesarias para resolver los siguientes problemas (en cualquiera de ellos, **antes de ponerte a programar, dibuja el diagrama de flujo)**:

1. Pedir al usuario/a un número real y mostrar por pantalla su doble y su mitad, con los mensajes correspondientes.
2. Pedir al usuario/a un carácter y mostrar el carácter anterior y el siguiente, con los mensajes correspondientes.
3. Dados dos puntos (pedir los pares (x1,y1) e (x2,y2)) calcular la distancia que hay entre ellos. Para calcular la distancia entre dos puntos se puede hacer uso de la función `sqrt()` de la biblioteca de funciones matemáticas. Mostrar el resultado siguiendo el formato del ejemplo:

```
(X1,Y1) = (5,6)
(x2,y2) = (5,4)

       Distancia = 2
```

4. Pedir al usuario/a que introduzca un número entre 1 y 10 y mostrar la tabla de multiplicar de dicho número correctamente alineada.
5. Pedir una cantidad en euros y calcular su equivalente en pesetas, dólares y libras. Hacer que todos los resultados queden correctamente alineados.

   Versión 1: Con números enteros.

```
#define PTSENCADAEURO 166
#define PTSENCADALIBRA 213
#define PTSENCADADOLAR 123
```

   Versión 2: Con números reales.

```
#define PTSENCADAEURO 166.386
#define EUROSENCADALIBRA 1.285
#define EUROSENCADADOLAR 0.737
```

6. Pedir dos números reales y mostrar la suma, la resta y el producto de dichos números. Hacer que todos los resultados queden correctamente alineados.
7. Pedir el coste de un artículo vendido y la cantidad de dinero entregada por el cliente y calcular y mostrar el cambio que hay que entregarle. Mostrar el resultado indicando hasta dos decimales.
8. Pedir como dato el radio de un círculo (variable de tipo real), calcular y mostrar tanto su área como la longitud de su circunferencia. Mostrar el resultado con tres decimales. ¿Qué pasa si el/la usuario/a introduce una palabra en lugar de un número?
9. En las olimpiadas de inverno el tiempo que realizan los participantes en la competición de velocidad en pista se mide en minutos, segundos y centésimas. La distancia que recorren se expresa en metros. Calcular la velocidad media de los participantes en kilómetros por hora a partir de los datos indicados anteriormente. Mostrar el resultado indicando hasta dos decimales. ¿Qué pasa si el/la usuario/a introduce más de un espacio entre los números?
10. Resolver un sistema de ecuaciones lineales (pedir los valores de A1, A2, B1, B2, C1 y C2):

$$A_1x + B_1y = C_1$$
$$A_2x + B_2y = C_2$$

    Para ello utilizaremos la regla de Kramer:

$$x = \frac{\begin{vmatrix} C_1 & B_1 \\ C_2 & B_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}} \qquad y = \frac{\begin{vmatrix} A_1 & C_1 \\ A_2 & C_2 \end{vmatrix}}{\begin{vmatrix} A_1 & B_1 \\ A_2 & B_2 \end{vmatrix}}$$

Por lo tanto, hay que resolver las ecuaciones:

$$x = ((C_1 * B_2) - (C_2 * B_1)) \,/\, ((A_1 * B_2) - (A_2 * B_1))$$
$$y = ((A_1 * C_2) - (A_2 * C_1)) \,/\, ((A_1 * B_2) - (A_2 * B_1))$$

11. Pedir como dato un número de días. Calcular e imprimir el número de horas, minutos y segundos que hay en ese número de días. Mostrar el resultado correctamente alineado. ¿Qué pasa si el/la usuario/a introduce una palabra en lugar de un número?
12. Pedir como dato un número de cuatro dígitos y generar una impresión como la que se muestra a continuación (el número introducido es 1234):

    1 2 3 4
      2 3 4
        3 4
          4

    ¿Qué pasa si el/la usuario/a introduce una palabra en lugar de un número?

13. Pedir como dato el lado de un cubo (variable real), calcular el área de la base, el área lateral y el área total. ¿Qué pasa si el/la usuario/a introduce letras en lugar de números?
14. Pedir al usuario/a que introduzca por teclado una fecha (año, mes, día) y calcular el número de segundos transcurridos desde el 1 de enero de 1970. Considerar que en ambos casos la hora de partida son las 00:00.
15. Calcular la media semanal de los ingresos de ventas de unos grandes almacenes. El/La usuario/a debe introducir por teclado las cantidades ingresadas de lunes a sábado.
16. Un/a corredor/a participante en la Vig-Bay (21.5 km) quiere conocer sus estadísticas después de la carrera. El programa deberá pedir los siguientes datos:

    - Hora de salida: hora, minutos, segundos.
    - Hora de llegada: hora, minutos, segundos.

    El programa mostrará como resultado:

    - su velocidad media en km/h, con 1 decimal (ejemplo: 11.3 km/h)
    - su ritmo medio, expresado en minutos por km, con el formato MIN:SEG (ejemplo: 5:19 min/km).