

Ejercicio.

Modifique el código del ejemplo anterior, de tal modo que cree dos hilos, que corran en paralelo, a partir de la función “hilo”.

Investigue como mandar parámetros a los hilos y presente un ejemplo el cual ejecute 2 hilos:

- La función **hilo01** tome como parámetro una palabra e imprima si esta es palíndromo o no.
- La función **hilo02** tome como parámetro un número entero e imprima si este es un número perfecto o no.
- Un palíndromo es una palabra, número o frase que se lee igual hacia adelante que hacia atrás
- Número perfecto: todo número natural que es igual a la suma de sus divisores propios (es decir, todos sus divisores excepto el propio número). Por ejemplo, 6 es un número perfecto ya que sus divisores propios son 1, 2, y 3 y se cumple que $1+2+3=6$.

EJECUCIÓN CÓDIGO PROPORCIONADO POR EL DOCENTE:

```
Terminal - alejandro@alejandro-VirtualBox: ~/Escritorio/Laboratorio#4
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ gcc Sincronizacion.c -o Sincronizacion#1 -l pthread
gcc: error: Sincronizacion.c: No existe el archivo o el directorio
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ gcc Sincronizacion#1.c -o Sincronizacion#1 -l pthread
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./Sincronizacion#1
Espera union de hilos (join)
Hilo #0
Hilo #1
Hilo #2
Hilo #3
Hilo #4
Hilo #5
Hilo #6
Hilo #7
Hilo #8
Hilo #9
Fin de union de hilos (join)
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$
```

EJECUCION DE CODIGO MODIFICADO QUE CORRE LOS HILOS EN PARALELO

```
Terminal - alejandro@alejandro-VirtualBox: ~/Escritorio/Laboratorio#4
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
Hilo #6
Hilo #7
Hilo #8
Hilo #9
Fin de union de hilos (join)
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ gcc paralelo.c -o paralelo -l pthread
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./paralelo
Esperando la unión de hilos (join)
Hilo #1: 0
Hilo #2: 0
Hilo #2: 1
Hilo #1: 1
Hilo #2: 2
Hilo #1: 2
Hilo #2: 3
Hilo #1: 3
Hilo #2: 4
Hilo #1: 4
Hilo #2: 5
Hilo #1: 5
Hilo #2: 6
Hilo #1: 6
Hilo #2: 7
Hilo #1: 7
Hilo #2: 8
Hilo #1: 8
Hilo #2: 9
Hilo #1: 9
Fin de la unión de hilos (join)
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$
```

EJECUCION DE CODIGO DEL EJERCICIO

```
Terminal - alejandro@alejandro-VirtualBox: ~/Escritorio/Laboratorio#4
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ gcc hilo.c -o hilo -l pthread
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
introduzca la palabra a evaluar:
oso
introduzca el numero a evaluar:
28
28 es un número perfecto.
La palabra "oso" es un palíndromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
introduzca la palabra a evaluar:
hoja
introduzca el numero a evaluar:
96
96 no es un número perfecto.
La palabra "hoja" no es un palíndromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$
```

```
Terminal - alejandro@alejandro-VirtualBox: ~/Escritorio/Laboratorio#4
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ gcc hilo.c -o hilo -l pthread
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
introduzca la palabra a evaluar:
oso
introduzca el numero a evaluar:
28
28 es un número perfecto.
La palabra "oso" es un palíndromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
introduzca la palabra a evaluar:
hoja
introduzca el numero a evaluar:
96
96 no es un número perfecto.
La palabra "hoja" no es un palíndromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ls
hilo hilo.c paralelo paralelo.c Sincronizacion#1 Sincronizacion#1.c
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$
```

“Un palíndromo es una palabra, número o frase que se lee igual hacia adelante que hacia atrás”

```
Terminal - alejandro@alejandro-VirtualBox: ~/Escritorio/Laboratorio#4
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
introduzca la palabra a evaluar:
oso
introduzca el numero a evaluar:
28
28 es un número perfecto.
La palabra "oso" es un palindromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
introduzca la palabra a evaluar:
hoja
introduzca el numero a evaluar:
96
96 no es un número perfecto.
La palabra "hoja" no es un palindromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ls
hilo hilo.c paralelo paralelo.c Sincronizacion#1 Sincronizacion#1.c
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
introduzca la palabra a evaluar:
85
introduzca el numero a evaluar:
85
La palabra "85" no es un palíndromo.
85 no es un número perfecto.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
introduzca la palabra a evaluar:
99
introduzca el numero a evaluar:
99
99 no es un número perfecto.
La palabra "99" es un palíndromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$
```

MODIFICACIÓN PARA QUE EL PROGRAMA SOLO RECIBA ENTRADAS

VÁLIDAS:

```
Terminal - alejandro@alejandro-VirtualBox: ~/Escritorio/Laboratorio#4
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ gcc hilo.c -o hilo -l pthread
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$ ./hilo
Ingresa una cadena: dskjad213113
La entrada no es una cadena válida. Intenta nuevamente.
Ingresa una cadena: ???
La entrada no es una cadena válida. Intenta nuevamente.
Ingresa una cadena: ADjkadj987
La entrada no es una cadena válida. Intenta nuevamente.
Ingresa una cadena: OJO
Cadena válida ingresada: OJO
Ingresa un número entero: dihjadasd
La entrada no es un número entero válido. Intenta nuevamente.
Ingresa un número entero: 87tasddadad
La entrada no es un número entero válido. Intenta nuevamente.
Ingresa un número entero: ???
La entrada no es un número entero válido. Intenta nuevamente.
Ingresa un número entero: 28
Número entero ingresado: 28
28 es un número perfecto.
La palabra "OJO" es un palíndromo.
alejandro@alejandro-VirtualBox:~/Escritorio/Laboratorio#4$
```

CÓDIGO:

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

```
#include <pthread.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

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

```
#include <stdbool.h>
```

```
#include <ctype.h>
```

```
// Estructura para pasar parámetros al hilo01
```

```
struct ParametrosHilo01 {  
  
    char palabra[100]; // Almacena la palabra a verificar  
  
};
```

// Estructura para pasar parámetros al hilo02

```
struct ParametrosHilo02 {  
  
    int numero; // Almacena el número a verificar  
  
};
```

```
bool esCadena(const char *input) {  
  
    // Verifica si la entrada contiene solo caracteres alfabéticos o espacios  
  
    for (int i = 0; i < strlen(input); i++) {  
  
        if (!isalpha(input[i]) && !isspace(input[i])) {  
  
            return false;  
  
        }  
  
    }  
  
    return true;  
  
}
```

```
bool esEntero(const char *cadena) {  
  
    int i = 0;  
  
    if (cadena[0] == '-' || cadena[0] == '+') {  
  
        i = 1;  
  
        }  
  
        for (; cadena[i] != '\0'; i++) {  
  
            if (!isdigit(cadena[i])) {  
  
                return false;  
  
                }  
  
                }  
  
                return true;  
  
            }  
}
```

```
void *hilo01(void *arg) {  
  
    struct ParametrosHilo01 *params = (struct ParametrosHilo01 *)arg;
```

```
char palabra[100];

strcpy(palabra, params->palabra);

int len = strlen(palabra);

int esPalindromo = 1;


for (int i = 0; i < len / 2; i++) {

    if (palabra[i] != palabra[len - i - 1]) {

        esPalindromo = 0;

        break;

    }

}


if (esPalindromo) {

    printf("La palabra \"%s\" es un palíndromo.\n", palabra);

} else {

    printf("La palabra \"%s\" no es un palíndromo.\n", palabra);

}
```



```
        return NULL;

    }

void *hilo02(void *arg) {

    struct ParametrosHilo02 *params = (struct ParametrosHilo02 *)arg;

    int numero = params->numero;

    int sumaDivisores = 0;


    for (int i = 1; i < numero; i++) {

        if (numero % i == 0) {

            sumaDivisores += i;

        }

    }


    if (sumaDivisores == numero) {

        printf("%d es un número perfecto.\n", numero);

    } else {

        printf("%d no es un número perfecto.\n", numero);

    }

}
```

```
}
```

```
return NULL;
```

```
}
```

```
int main(void) {
```

```
    pthread_t hiloPalindromo, hiloPerfecto;
```

```
    char input[100], entradaUsuario[100];
```

```
    bool cadenaValida = false;
```

```
    bool entradaValida = false;
```

```
    // Parámetros para hilo01 (palíndromo)
```

```
    struct ParametrosHilo01 paramsHilo01;
```

```
    do {
```

```
        printf("Ingresa una cadena: ");
```

```
        fgets(input, sizeof(input), stdin);
```

```
// Elimina el salto de línea que agrega fgets

size_t len = strlen(input);

if (len > 0 && input[len - 1] == '\n') {

    input[len - 1] = '\0';

}

cadenaValida = esCadena(input);

if (!cadenaValida) {

    printf("La entrada no es una cadena válida. Intenta nuevamente.\n");

}

} while (!cadenaValida);

printf("Cadena válida ingresada: %s\n",input);

strcpy(paramsHilo01.palabra, input);
```

```
// Parámetros para hilo02 (número perfecto)
```

```
struct ParametrosHilo02 paramsHilo02;
```

```
do {
```

```
    printf("Ingresa un número entero: ");
```

```
    fgets(entradaUsuario, sizeof(entradaUsuario), stdin);
```

```
    // Elimina el salto de línea que agrega fgets
```

```
    size_t longitudEntrada = strlen(entradaUsuario);
```

```
    if (longitudEntrada > 0 && entradaUsuario[longitudEntrada - 1] == '\n') {
```

```
        entradaUsuario[longitudEntrada - 1] = '\0';
```

```
    }
```

```
    entradaValida = esEntero(entradaUsuario);
```

```
    if (!entradaValida) {
```

```
printf("La entrada no es un número entero válido. Intenta nuevamente.\n");
```

```
}
```

```
} while (!entradaValida);
```

```
int input1 = atoi(entradaUsuario); // Convierte la cadena en un número entero
```

```
printf("Número entero ingresado: %d\n", input1);
```

```
paramsHilo02.numero = input1;
```

```
if (pthread_create(&hiloPalindromo, NULL, hilo01, (void *)&paramsHilo01) != 0) {
```

```
printf("Error creando el hilo 01.\n");
```

```
exit(1);
```

```
}
```

```
if (pthread_create(&hiloPerfecto, NULL, hilo02, (void *)&paramsHilo02) != 0) {
```

```
printf("Error creando el hilo 02.\n");
```

```
exit(1);
```

```
}
```

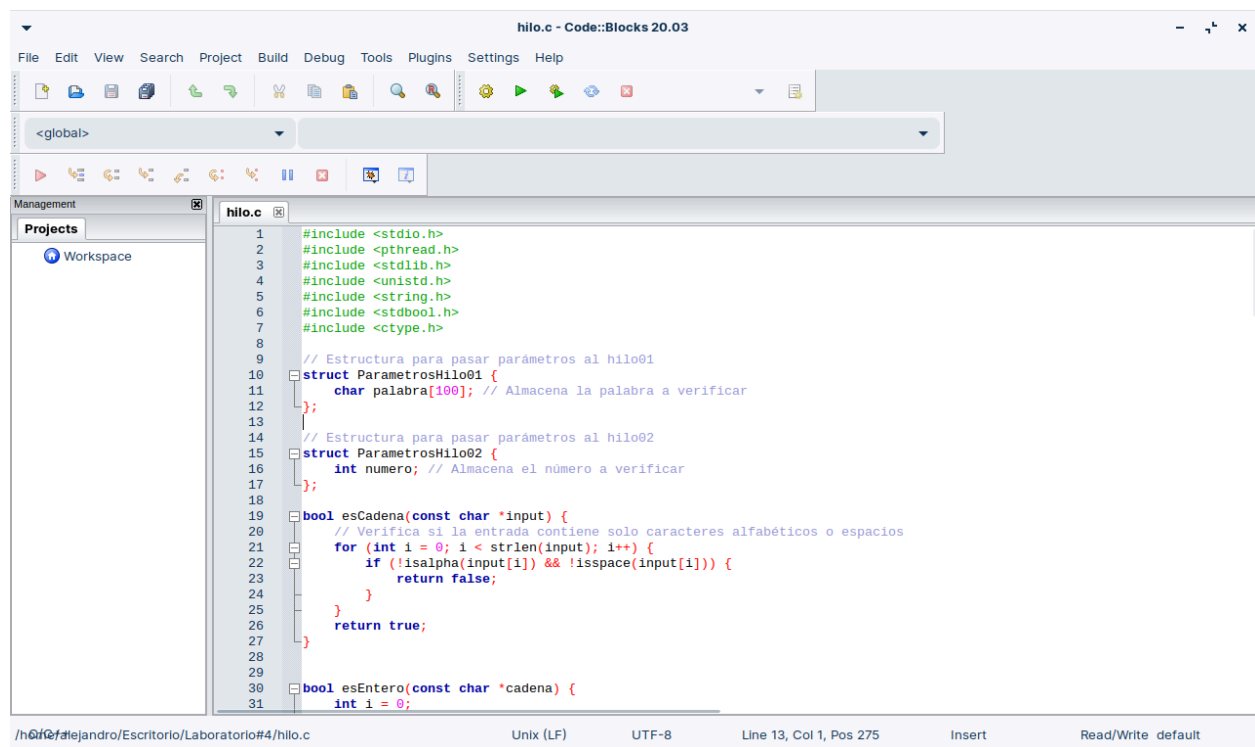
```
pthread_join(hiloPalindromo, NULL);
```

```
pthread_join(hiloPerfecto, NULL);
```

```
return 0;
```

```
}
```

CÓDIGO EN IMÁGENES:



hilo.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Tools Plugins Settings Help

<global>

Management

Projects

Workspace

hilo.c

```
31 int i = 0;
32 if (cadena[0] == '-' || cadena[0] == '+') {
33     i = 1;
34 }
35
36 for (; cadena[i] != '\0'; i++) {
37     if (!isdigit(cadena[i])) {
38         return false;
39     }
40 }
41 return true;
42 }
43
44 void *hilo01(void *arg) {
45     struct ParametrosHilo01 *params = (struct ParametrosHilo01 *)arg;
46     char palabra[100];
47     strcpy(palabra, params->palabra);
48     int len = strlen(palabra);
49     int esPalindromo = 1;
50
51     for (int i = 0; i < len / 2; i++) {
52         if (palabra[i] != palabra[len - i - 1]) {
53             esPalindromo = 0;
54             break;
55         }
56     }
57
58     if (esPalindromo) {
59         printf("La palabra \"%s\" es un palindromo.\n", palabra);
60     } else {
61         printf("La palabra \"%s\" no es un palindromo.\n", palabra);
```

/home/alejandra/Escritorio/Laboratorio#4/hilo.c Unix (LF) UTF-8 Line 13, Col 1, Pos 275 Insert Read/Write default

hilo.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Tools Plugins Settings Help

<global>

Management

Projects

Workspace

hilo.c

```
61     printf("La palabra \"%s\" no es un palindromo.\n", palabra);
62 }
63
64 return NULL;
65 }
66
67 void *hilo02(void *arg) {
68     struct ParametrosHilo02 *params = (struct ParametrosHilo02 *)arg;
69     int numero = params->numero;
70     int sumaDivisores = 0;
71
72     for (int i = 1; i < numero; i++) {
73         if (numero % i == 0) {
74             sumaDivisores += i;
75         }
76     }
77
78     if (sumaDivisores == numero) {
79         printf("%d es un número perfecto.\n", numero);
80     } else {
81         printf("%d no es un número perfecto.\n", numero);
82     }
83
84     return NULL;
85 }
86
87 int main(void) {
88     pthread_t hiloPalindromo, hiloPerfecto;
89     char input[100], entradaUsuario[100];
90     bool cadenaValida = false;
```

/home/alejandra/Escritorio/Laboratorio#4/hilo.c Unix (LF) UTF-8 Line 13, Col 1, Pos 275 Insert Read/Write default

hilo.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Tools Plugins Settings Help

<global>

Management

Projects

Workspace

hilo.c

```
91     bool cadenaValida = false;
92     bool entradaValida = false;
93
94     // Parámetros para hilo01 (palíndromo)
95     struct ParametrosHilo01 paramsHilo01;
96
97
98     do {
99         printf("Ingresa una cadena: ");
100         fgets(input, sizeof(input), stdin);
101
102         // Elimina el salto de línea que agrega fgets
103         size_t len = strlen(input);
104         if (len > 0 && input[len - 1] == '\n') {
105             input[len - 1] = '\0';
106         }
107
108         cadenaValida = esCadena(input);
109
110         if (!cadenaValida) {
111             printf("La entrada no es una cadena válida. Intenta nuevamente.\n");
112         }
113     } while (!cadenaValida);
114
115     printf("Cadena válida ingresada: %s\n", input);
116
117     strcpy(paramsHilo01.palabra, input);
118
119     // Parámetros para hilo02 (número perfecto)
120
121     struct ParametrosHilo02 paramsHilo02;
```

/home/alejandro/Escritorio/Laboratorio#4/hilo.c Unix (LF) UTF-8 Line 13, Col 1, Pos 275 Insert Read/Write default

hilo.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Tools Plugins Settings Help

<global>

Management

Projects

Workspace

hilo.c

```
121     struct ParametrosHilo02 paramsHilo02;
122
123     do {
124         printf("Ingresa un número entero: ");
125         fgets(entradaUsuario, sizeof(entradaUsuario), stdin);
126
127         // Elimina el salto de línea que agrega fgets
128         size_t longitudEntrada = strlen(entradaUsuario);
129         if (longitudEntrada > 0 && entradaUsuario[longitudEntrada - 1] == '\n') {
130             entradaUsuario[longitudEntrada - 1] = '\0';
131         }
132
133         entradaValida = esEntero(entradaUsuario);
134
135         if (!entradaValida) {
136             printf("La entrada no es un número entero válido. Intenta nuevamente.\n");
137         }
138     } while (!entradaValida);
139
140     int input1 = atoi(entradaUsuario); // Convierte la cadena en un número entero
141
142     printf("Número entero ingresado: %d\n", input1);
143
144     paramsHilo02.numero = input1;
145
146     if (pthread_create(&hiloPalindromo, NULL, hilo01, (void *)&paramsHilo01) != 0) {
147         printf("Error creando el hilo 01.\n");
148         exit(1);
149     }
150
151     if (pthread_create(&hiloPerfecto, NULL, hilo02, (void *)&paramsHilo02) != 0) {
```

/home/alejandro/Escritorio/Laboratorio#4/hilo.c Unix (LF) UTF-8 Line 13, Col 1, Pos 275 Insert Read/Write default

