Universidad Autónoma De Baja California

Grupo-941

23/09/2025

Alumno:

Gastelum fierro Brian de Jesus

Ejercicio 1

Analiza el siguiente código en C y determina la salida del programa. Explica qué tipo de memoria (estática, dinámica o automática) se usa en cada caso.

ej1.cpp U

memoria > ej1.cpp > main()

```c
#include <stdio.h>

int global_var = 10;

void func() {
int local_var=20;
static int static_var=30;
local_var++;
static_var++;
printf("local_var: %d, static_var: %d",local_var,static_var);
}
int main()
{
    func();
    func();
    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\paradigmas\memoria> ./ej1
local_var: 21, static_var: 31local_var: 21, static_var: 32
PS C:\paradigmas\memoria>
```

Se utiliza

En la variable global_var y static_var se usa memoria estatica

En la variable local_var se usa memoria automatica

Ejercicio 2

Analiza el siguiente código y encuentra el error relacionado con la memoria dinámica.

ej1.cpp U ej2.cpp U

memoria > ej2.cpp > main()

```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   void allocate_memory()
5   {
6   int *arr=(int *)malloc(5 * sizeof(int));
7   for(int i=0; i<5; i++)
8   {
9       arr[i]=i * 2;
10  }
11  printf("arr[2]:%d\n",arr[2]);
12  }
13
14  int main()
15  {
16      allocate_memory();
17      return 0;
18  }
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\paradigmas\memoria> ./ej2
arr[2]:4
PS C:\paradigmas\memoria>
```

1. ¿Qué problema de manejo de memoria presenta este código?

Se reserva memoria dinámica con malloc dentro de allocate_memory().

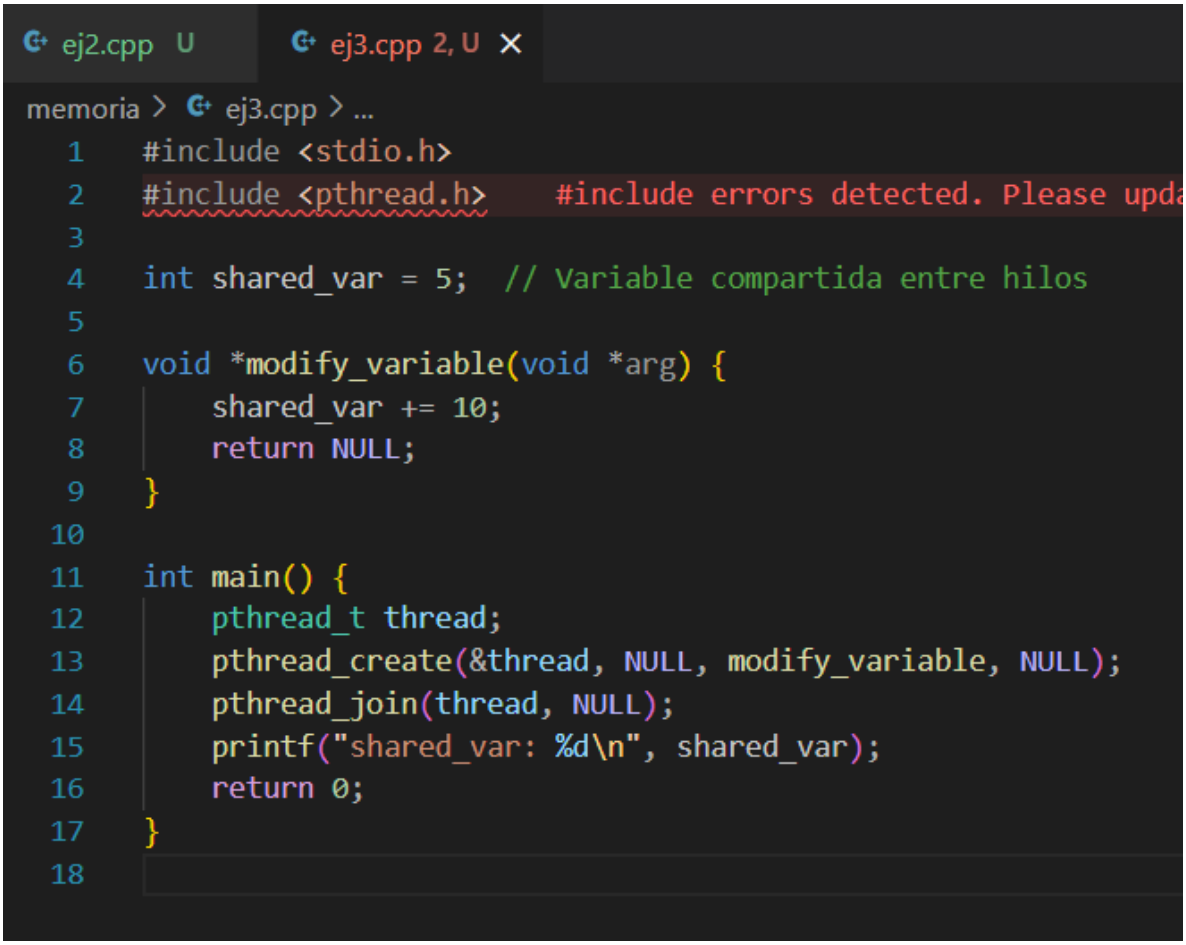
Esa memoria nunca se libera con free.

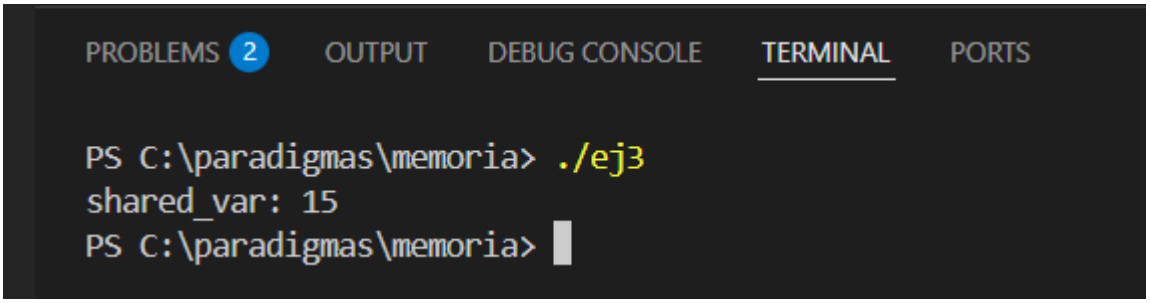
2. ¿Cómo podríamos solucionarlo?

Liberando la memoria con free(arr) una vez que ya no la necesitamos

Ejercicio 3

El siguiente código usa pthread para crear un subproceso que modifica una variable global. Analiza la ejecución y predice la salida.





¿Cual es la salida esperada del programa?

Shared_Var : 15

¿Qué pasaría si removemos pthread_join(thread,Null)?

Me imprime 5