Universidad de Barcelona

Quim Lagunas

Arthur Font Gouveia

Sistemas Operativos I

Sessió 9 - Punteros

Barcelona

2020

1. Ejercicio 1

Valgrind:

==4877== Invalid write of size 4

==4877== at 0x1091D9: main (codi_vector1.c:13)

==4877== Address 0x4a51068 is 0 bytes after a block of size 40 alloc'd

==4877== at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)

==4877== by 0x10919E: main (codi_vector1.c:9)

==4877== ERROR SUMMARY: 10 errors from 1 contexts (suppressed: 0 from 0)

El problema és que estamos escribiendo fuera de la zona de memoria dinámica que hemos reservado.

Más específicamente, reservamos 40 bytes para la variable ‘a’ pero escribimos 80 bytes. Por lo tanto, se han pasado 10 errores (escribimos un bloque de 4 bytes fuera del rango 10 veces) como se puede observar en el valgrind.

2. Ejercicio 2

Valgrind:

==5101== Invalid write of size 4

==5101== at 0x1091DD: main (codi_vector2.c:16)

==5101== Address 0x4a51054 is 20 bytes inside a block of size 40 free'd

==5101== at 0x483CA3F: free (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)

==5101== by 0x1091C8: main (codi_vector2.c:13)

==5101== Block was alloc'd at

==5101== at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)

==5101== by 0x10919E: main (codi_vector2.c:8)

==5101== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)

El problema és que estamos escribiendo en una zona de memoria dinámica que está disponible para siguientes alocaciones de memória. Por lo tanto, el valor escrito puede ser sobrescrito a cualquier momento.

El valgrind nos dice que hay 1 error que es la escribir un bloque (de tamaño de 4 bytes) en la zona memoria dinámica que acaba de ser desalojada.

3. Ejercicio 3

Valgrind:

==5624== Conditional jump or move depends on uninitialised value(s)

==5604== at 0x48D686E: __vfprintf_internal (vfprintf-internal.c:1687)

==5604== by 0x48C0EBE: printf (printf.c:33)

==5604== by 0x1091BF: main (codi_vector3.c:10)

==5624== ERROR SUMMARY: 5 errors from 5 contexts (suppressed: 0 from 0)

El problema és que el salto para cambiar de posición en el array depende de las cincos primeras posiciones del array que no están inicializadas. Por lo tanto, el valgrind nos dice que hay 5 errores.

4. Ejercicio 4

Valgrind:

==5725== HEAP SUMMARY:

==5725== in use at exit: 40 bytes in 1 blocks

==5725== total heap usage: 2 allocs, 1 frees, 1,064 bytes allocated

==5725==

==5725== LEAK SUMMARY:

==5725== definitely lost: 40 bytes in 1 blocks

==5725== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

El problema és que olvidamos que llamar el metodo free() para liberar la zona de memoria reservada, así esta zona de memoria seguirá ocupada ya que el lenguaje C no dispone de un ‘garbage collector’ como en el Java.

El valgrind nos dice que no hay ningún error, pero también nos dice que perdemos 40 bytes de la memoria dinámica.

5. Ejercicio 5

Valgrind:

==6080== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

El problema és que estamos escribiendo fuera del vector. Por lo tanto no sabemos donde estamos escribiendo y/o si estamos estamos sobreescribiendo algo en la pila. El valgrind nos dice que no hay ningún error.