Utilización de valgrind

Valgrind (www.valgrind.org) es una herramienta de SW libre que ayuda a detectar problemas de memoria.

Si queremos utilizar valgrind para detectar errores en nuestros programas debemos hacer lo siguiente:

- Compilar nuestro programa de la siguiente manera:

```
gcc -Wall -g fichero1.c fichero2.c -o miejecutable
```

- Ejecutar nuestro programa poniendo en línea de comandos:

```
valgrind --leak-check=full miejecutable param1 param2 ...
```

- Si nuestro programa tiene errores de memoria, valgrind nos mostrará las líneas donde ha detectado el error.

Ejemplo:

➢ Valgrind detecta zonas de memoria mal reservadas. Por ejemplo, imaginad que tenemos el programa siguiente en un fichero llamado `mio.c`:

```
1.  #include <stdlib.h>
2.
3.  void testMem(int * a)
4.  {
5.      a[11]=0;        //inicializar el elemento numero 11 del array (no esta reservada)
6.  }
7.
8.  int main(void)
9.  {
10.
11.     int *array;
12.     array=malloc(10*sizeof(int)); //reservar un array de 10 enteros
13.     testMem(array);
14.     return 0;       //no se ha liberado la memoria array)
15. }
```

y ejecutamos el programa con valgrind, nos mostrará el siguiente mensaje de error:

```
==12059== Invalid write of size 4
==12059== at 0x8048351: testMem (mio.c:5)
==12059== by 0x8048383: main (mio.c:13)
==12059== Address 0x401D054 is 4 bytes after a block of size 40 alloc'd
==12059== at 0x4004639: malloc (vg_replace_malloc.c:149)
==12059== by 0x8048372: main (mio.c:12)
```

significa que estamos escribiendo cuatro bytes en una zona de memoria inválida. Esto ocurre dentro de la función `testMem` en la línea 5 de nuestro fichero `mio.c`, al que se llama desde el programa `main` de `mio.c` en la línea 13. Esta memoria se ha reservado con un `malloc`, dentro del programa `main` en la línea 12 del fichero `mio.c`.

➢ También, valgrind detecta memorias sin liberar. Por ejemplo, para nuestro programa `mio.c`, valgrind nos da:

```
==12087== malloc/free: in use at exit: 40 bytes in 1 blocks.
==12070== malloc/free: 1 allocs, 0 frees, 40 bytes allocated.
```

diciendo que hemos hecho una reserva de memoria (1 malloc en nuestro caso) y cero instrucciones de liberación (0 free), es decir falta un `free`. Y al final mostrará:

```
==12087== 40 bytes in 1 blocks are definitely lost in loss record 1 of 1
==12087==    at 0x4004639: malloc (vg_replace_malloc.c:149)
==12087==    by 0x8048372: main (mio.c:12)
```

significa la memoria perdida (sin liberar) se ha reservado dentro del programa `main` en la línea 12 del `mio.c`.

NOTA IMPORTANTE: Algunas veces, en una máquina puede que nuestro programa funcione bien y no de "`violación de segmento`" a pesar de que intenta acceder a memoria no reservada. Por eso, siempre es recomendable probar el programa con valgrind. Aunque todo funcione bien en una máquina determinada, puede que en otra máquina no tengamos la misma suerte. Para una ayuda más completa ver:

www.valgrind.org/docs/manual/QuickStart.html