



Valgrind



Como encontrar errores de uso de memoria con Valgrind

Albert Astals Cid



Valgrind



¿Quién es Albert Astals Cid?

- Soy Yo!
- Ingeniero en Informática por la UPC
- Programador en KDE desde 2003
- Mantenedor de KPDF/Okular desde 2004
- Miembro de KDE eV desde 2005
- Presidente de KDE España
- NO soy un experto en Valgrind



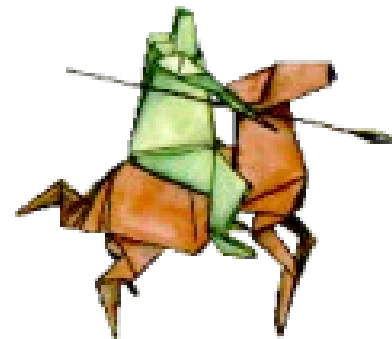


Valgrind



¿Qué es Valgrind?

- Valgrind es un conjunto de herramientas para hacer debugging y profiling
- Herramientas:
 - memcheck: detector de errores de memoria
 - cachegrind: profiler de cache
 - callgrind: profiler de llamadas
 - massif: profiler de uso de heap
 - Helgrind/DRD: debugger de threads
- Funciona en X86/Linux, AMD64/Linux, PPC32/Linux, PPC64/Linux, X86/Darwin
- Licencia GPL



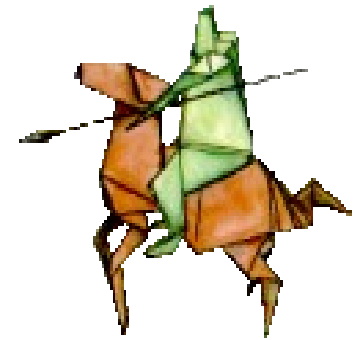


Valgrind



¿Porqué usar Valgrind?

- No necesita recompilación
- No es un juguete
- Tiene mantenedores activos
- Es libre





Valgrind



Memcheck

- Detecta errores de gestión de memoria
- Dirigido a programas en C y C++
- Intercepta las lecturas y escrituras de memoria y llamadas a malloc/new/free/delete
- Detecta:
 - Accesos incorrectos a memoria
 - Usos de variables no inicializadas
 - Leaks de memoria
 - free/delete incorrectos
- Ejecuta los programas entre 10 y 30 veces más lento





Valgrind



¿Como usar Memcheck?

\$ **valgrind** *nombreDeMiPrograma*

--trace-children=no|yes Valgrind-ise child processes? [no]

--log-socket=ipaddr:port log messages to socket ipaddr:port

--num-callers=<number> show <number> callers in stack traces [12]

--suppressions=<filename> suppress errors described in <filename>

--gen-suppressions=no|yes|all print suppressions for errors? [no]

--db-attach=no|yes start debugger when errors detected? [no]

--leak-check=no|summary|full search for memory leaks at exit? [summary]





Valgrind



¿Cuándo usar Memcheck?

- Cuando quieres debugar un problema
- Cuando quieres buscar leaks
- Siempre!
 - Lento...
 - ...pero posible, Julian Seward ejecuta KDE en un 2 PPC970 2.5Ghz y 4GB
 - Encuentra errores que no siempre son visibles





Valgrind



Entendiendo a Valgrind

==25832== Invalid read of size 4

==25832== at 0x8048724: BandMatrix::ReSize(int, int, int) (bogon.cpp:45)

==25832== by 0x80487AF: main (bogon.cpp:66)

==25832== Address 0xBFFFF74C is not stack'd, malloc'd or free'd



???





Valgrind



Ejemplos simples

- error1.c - variable no inicializada
- error2.c - acceso a memoria no alocataada
- error3.c - free incorrecto
- error4.c - delete incorrecto
- error5.c - leak de memoria
- error6.c - leak de memoria
- error7.c - uso de memoria ya liberada

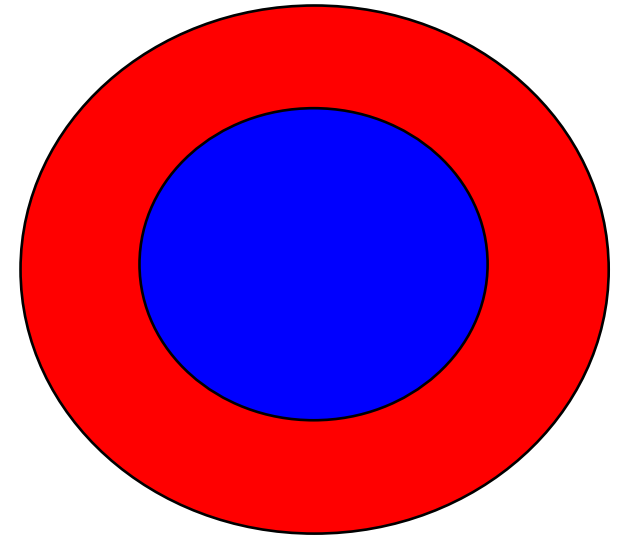


Valgrind



Tu programa no es tu código

- A veces ninguno de los errores es nuestro
- La solución es usar suppressions
- `--gen-suppressions=no/yes/all`
- `--suppression=ficheroSuppressions`





Valgrind



Preguntas?

