Valgrind приобрел популярность в первую очередь за свои возможности по нахождению утечек памяти в программах. За этот реализацию этих функций отвечает модуль memcheck, для которого определены отдельные опции, управляющие процессом проверки.

**--leak-check**

включает (значение yes, summary или full) или отключает (значение no) функцию обнаружения утечек памяти. Стоит отметить, что при использовании значения summary, memcheck выдает лишь краткую информацию об утечках памяти, тогда как при других значениях, кроме сводной информации, будет выдаваться еще и информация о месте, в котором происходит эта утечка памяти.

**--leak-resolution**

(возможные значения low, med или high) указывает способ сравнения стека вызовов функций. При значениях low и med, в сравнении используются два или четыре последних вызова, соответственно, а при high, сравнивается полный стек вызова. Эта опция влияет лишь на способ представления результатов поиска ошибок.

**--undef-value-errors**

(yes или no) определяет, будут ли показывать ошибки об использовании не инициализированных значений.

Прочие опции используются значительно реже, и в случае необходимости вы можете найти их описание в руководстве пользователя.

**Нахождение утечек памяти**

При окончании работы программы valgrind выдает сводную таблицу, описывающую количество найденных ошибок, а также выделение памяти в программе, например:

ERROR SUMMARY: 2569904 errors from 493 contexts (suppressed: 17962 from 9)

malloc/free: in use at exit: 85,066,939 bytes in 313,004 blocks.

malloc/free: 10,552,914 allocs, 10,239,910 frees, 565,747,810 bytes allocated.

For counts of detected errors, rerun with: -v

searching for pointers to 313,004 not-freed blocks.

checked 117,623,772 bytes.

И в самом конце отчета, выдается сводная таблица по каждому из типов ошибок работы с памятью:

LEAK SUMMARY:

definitely lost: 2,260 bytes in 47 blocks.

indirectly lost: 1,680 bytes in 66 blocks.

possibly lost: 2,703,124 bytes in 13,791 blocks.

still reachable: 82,359,875 bytes in 299,100 blocks.

suppressed: 0 bytes in 0 blocks.

*Definitely lost* означает, что valgrind нашел область памяти, на которую нет указателей, т.е. программист не освободил память, при выходе указателя за область видимости. *Possibly lost* показывает, что найден указатель, указывающий на часть области памяти, но valgrind не уверен в том, что указатель на начало области памяти до сих пор существует (это может происходить в тех случаях, когда программист вручную управляет указателями). *Still reachable* обычно означает, что valgrind нашел указатель на начало не освобожденного блока памяти, что во многих случаях связано с выделением глобальных переменных и т.п. вещей. Обычно эта информация показывается только при указании опции --show-reachable со значением yes.

Между двумя этими таблицами выдаются данные по каждой из найденных ошибок работы с памятью, вида:

756 bytes in 27 blocks are definitely lost in loss record 1,077 of 1,267

at 0x4022AB8: malloc (vg\_replace\_malloc.c:207)

by 0x7C485DA: \_\_libc\_res\_nsend (res\_send.c:425)

by 0x7C47276: \_\_libc\_res\_nquery (res\_query.c:171)

by 0x7C47B5B: \_\_res\_nquery (res\_query.c:223)

by 0x834A618: LookUp::LookUp(std::string&) (LookUp.cpp:83)

by 0x8637C29: ClientThread::Connect(LogFileEntry&) (ClientThread.cpp:359)

by 0x86394D5: ClientThread::Run() (ClientThread.cpp:215)

by 0x80FD839: Thread::StartRunLoop(void\*) (Thread.cpp:315)

by 0x4163FD9: start\_thread (pthread\_create.c:297)

by 0x43843AD: clone (in /usr/lib/debug/libc-2.7.so)

Первой строкой идет описание ошибки, вместе с указанием номера блока в списке потенциально потерянных блоков памяти, а также размером "потерянного" блока памяти. "Важность" ошибки соответствует описанию в итоговой таблице. После строки описания, приводится стек вызовов функций, которые привели к возникновению "потерянного" блока памяти. Этот список достаточно подробен для того, чтобы обнаружить точное место возникновения данной утечки памяти.