УДК 004.072.4

Описание технологий защиты памяти и анализ проблем понижения производительности при ее использовании.

А.С. Цветкова[1,2]

[1] Московский физико-технический институт (государственный университет)

[2] АО «Интел А/О»

В языках Си и C++ широко используется доступ к памяти с использованием указателей. Данная концепция необходима для низкоуровневого программирования, но она приводит к появлению множества уязвимостей. Одна из самых известных атак такого типа -- атака на переполнение буфера[4]. Выход за границы массива может привести к порче данных или к нарушению безопасности системы. Для обеспечения безопасности системы используются технологии защиты памяти[2]. Одна из технологий называется fat pointers и заключается в том чтобы вместе с указателем на область памяти хранить границы области, в пределах которых он может изменяться. При использовании такого подхода необходима дополнительная память для хранения допустимых границ каждого указателя и дополнительное время для каждого его использования, поскольку каждый раз нужно проверять выход за границы массива.

В данной работе рассмотрены основные преимущества и недостатки, возникающие в результате работы с защищенными указателями, и представлено краткое описание технологии Intel® MPX[1,3].

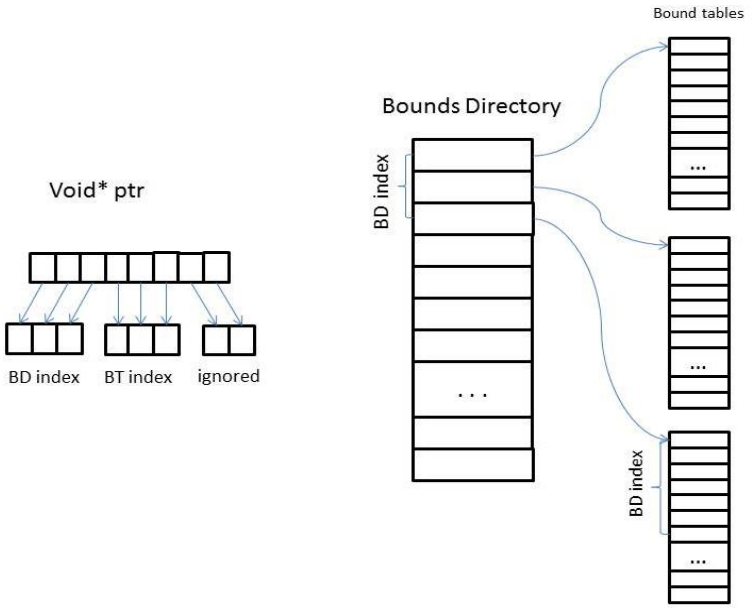

Рис. 1. Адресация в случае Intel® MPX

Наибольшее внимание уделено часто встречающейся задаче копирования памяти, дающей наибольшие накладные расходы. В копируемом массиве могут находиться указатели, ассоциированные со своими границами. Чтобы указатель остался защищенным и после переноса в новую область памяти, необходимо с его новым местоположением ассоциировать границы в процессе копирования, что замедляет работу программы. Описывается разработанный алгоритм копирования памяти, позволяющий сократить накладные расходы при работе программы.

Литература

1. Intel® Memory Protection Extensions (Intel® MPX) support in the GCC compiler // *https://gcc.gnu.org/wiki/Intel%20MPX%20support%20in%20the%20GCC%20compiler*

2. Introduction to Intel® Memory Protection Extensions // *https://software.intel.com/en-us/articles/introduction-to-intel-memory-protection-extensions*
3. Intel® Architecture Instruction Set Extensions Programming Reference // *https://software.intel.com/sites/default/files/managed/07/b7/319433-023.pdf*
4. *Greg Hoglund, Gary McGraw* Exploiting Software: How to Break Code