СПИСОК ЛИТЕРАТУРЫ:

1. *Silverman J. H.* An Introduction to the Theory of Lattices and Applications to Cryptography. 2006. — 76 p.
2. *Peikert C., Rosen A.* Efficient Collision-Resistant Hashing from worst-Case Assumptions on Cyclic Lattices. 2006. — 20 p.
3. *Hoffstein J., Pipher J., Silverman J. H.* NTRU: A ring-based public key cryptosystem // ANTS-III. 1998. P. 267—288.

И. В. Парфилов, П. А. Силин, Ю. Ю. Шумилов

ПОДХОД К РЕШЕНИЮ ПРОБЛЕМЫ ВЗАИМНЫХ БЛОКИРОВОК В МНОГОПОТОЧНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

При разработке сложных многопоточных программных средств важно учитывать ряд возможных проблем, которые в дальнейшем могут привести к их некорректной работе. Одним из признаков некорректной работы многопоточной программы является возникновение ситуации взаимной блокировки потоков в процессе ее выполнения, вызванное некорректным использованием средств синхронизации. Возникновение ситуаций взаимных блокировок может приводить к полной или частичной потере функциональности программных средств, в частности к появлению уязвимостей в системе безопасности. Проявление ситуаций взаимных блокировок сильно зависит от динамики выполнения программы в конкретной программной и аппаратной среде. Это свойство не позволяет создать сколько-нибудь эффективные эмпирические алгоритмы выявления взаимных блокировок на этапе тестирования программного обеспечения. В связи с этим на этапе проектирования многопоточной программы возникает проблема поиска потенциальных ситуаций взаимных блокировок.

Для выявления ситуаций взаимных блокировок на этапе детализированного проектирования используется метод проверки на модели. Такой подход имеет ряд преимуществ:

- он полностью автоматизирован, от пользователя лишь требуется провести моделирование проектируемой системы;
- если в проектируемой системе возможны ситуации взаимных блокировок, то в процессе проверки на модели всегда будет показана структура средств синхронизации, являющаяся результатом их некорректного использования и приводящая к возникновению ситуации взаимной блокировки.

В работе используется описательная модель многопоточного программного обеспечения. В качестве понятий, которыми оперирует модель, используется:

- объект — разделяемый ресурс (информационный или функциональный объект, к которому возможен доступ из разных потоков);
- субъект — поток, выполняющий доступ к разделяемому ресурсу;
- исключающий семафор — средство синхронизации, ограничивающее доступ к разделяемому ресурсу.

Основные преимущества описательной модели многопоточного программного обеспечения по сравнению с моделями, приведенными в [1, 2]:

- строгость, позволяющая в формальных математических терминах определять структуры, являющиеся результатом некорректного использования средств синхронизации;
- наглядность, позволяющая относительно легко исправлять некорректные структуры с целью избавления системы от ситуаций взаимной блокировки;
- полнота, позволяющая обрабатывать потоки произвольного вида (в терминах структурной теоремы Э. Дейкстры это означает, что субъекты могут содержать операторы ветвления и цикла [3]).

При анализе системы на отсутствие в ней потенциальных ситуаций взаимных блокировок приходится проверять большое количество порожденных при декомпозиции вспомогательных систем. Кроме того, исходная система может состоять из достаточно большого числа субъектов, активно взаимодействующих со средствами синхронизации. Данное обстоятельство делает анализ модели достаточно трудоемким (независимо от того, проводится он в ручном или автоматическом режиме).

В связи с этим был разработан алгоритм, позволяющий сводить проблему поиска структур некорректного использования средств синхронизации к проблеме поиска сильно связных компонент в ориентированном графе. Данный алгоритм позволяет проверить отсутствие в исходной модели потенциальных ситуаций взаимной блокировки.

Сложность алгоритма составляет $O(n^5)$, где n — число вершин общего графа системы, что является его несомненным достоинством, поскольку сложность прямого подхода, основанного на поиске ситуаций взаимной блокировки путем анализа относительных динамик выполнения различных потоков, является экспоненциальной.

На основе представленного подхода разработано программное средство анализа моделей многопоточного программного обеспечения, которое предоставляет возможность создания, редактирования, сохранения, загрузки моделей многопоточного программного обеспечения, а также позволяет показывать отсутствие потенциальных ситуаций взаимных блокировок в модели. Программное средство представляет собой совокупность совместно используемых библиотек, в которых непосредственно содержатся компоненты приложения, и ядра, организующего работу и взаимодействие компонентов. Такая компонентная архитектура обладает рядом преимуществ, таких как: четкая инкапсуляция деталей реализации за интерфейсами; возможность легкого добавления, замены, удаления компонент; облегчение процесса тестирования программного средства.

СПИСОК ЛИТЕРАТУРЫ:

1. *Bensalem S., Havelund K.* Dynamic Deadlock Analysis of Multi-threaded Programs // Haifa Verification Conference / Shmuel Ur, Eyal Bin, and Yaron Wolfsthal, ed. Vol. 3875 of LNCS. Springer, 2005. P. 208–223.
2. *Engler D., Ashcraft K.* RacerX: Effective, Static Detection of Race Conditions and Deadlocks // Proceedings of the 19th ACM Symposium on Operating Systems Principles. Oct. 2003. P. 237–252.
3. *Дал У., Дейкстра Э., Хоор К.* Структурное программирование. 1-е изд. М.: Мир, 1975.

Т. М. Пестунова, З. В. Родионова

ОБ ОДНОМ ПОДХОДЕ К УПРАВЛЕНИЮ ПРАВАМИ ДОСТУПА

Эффективность функционирования современных автоматизированных информационных систем (АИС) предприятия во многом зависит от того, насколько соответствуют полномочия пользователя системы его должностным функциям. Признанным фактом является то, что расширение полномочий сверх необходимых приводит к увеличению случайных ошибок, росту рисков, связанных с несанкционированным доступом к данным. При недостаточных полномочиях возникают затруднения в выполнении сотрудником своей работы.

Формализованные полномочия в виде прав доступа получают свое отражение в настройках систем разграничения доступа (СРД) АИС, безопасное построение которых определяется формальной моделью. Несмотря на достаточно высокий уровень теоретических исследований в