

## ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ЗАПРОСОВ СИСТЕМ УПРАВЛЕНИЯ ИДЕНТИФИКАЦИОННЫМИ ДАННЫМИ

П.П. ПИНЮТА<sup>1</sup>, В.В. БАХТИЗИН<sup>2</sup>

*Белорусский государственный университет информатики и радиоэлектроники*

*ул. П. Бровки, 6, г. Минск, 220013, Республика Беларусь*

*<sup>1</sup>pinyuta-p@yandex.ru; <sup>2</sup>bww@bsuir.by*

Рассмотрен ряд методов планирования. Произведён их сравнительный анализ с точки зрения адаптации к системам управления идентификационными данными. Разработан алгоритм распределения задач на базе кооперационного метода планирования. Приведены результаты экспериментального исследования.

*Ключевые слова:* идентификационные данные, кооперационный метод планирования, распределения задач.

Одним из главных элементов информационной инфраструктуры служит Система управления идентификационными данными (СУИД). Она обеспечивает ролевое управление доступом к ресурсам на основе единой политики безопасности и должностных обязанностей сотрудника. Безопасность любого крупного бизнеса опирается на централизованное управление идентификационными данными совместно с корпоративной стратегией безопасности, обеспечением защиты корпоративных сетей и межсетевых взаимодействий.

Внедрение на предприятие СУИД позволяет в значительной степени минимизировать перечисленные риски. С ростом числа корпоративных приложений некоторые задачи СУИД способны заблокировать работу целевой системы на некоторое время. Под целевой системой следует понимать систему, дающую авторизованному пользователю определённые права и привилегии. Так же целевая система делегирует СУИД механизмы управления идентификационными данными пользователей. Процесс передачи этих механизмов называется интеграцией целевой системы в СУИД. Таким образом, при использовании современных СУИД риск простоя оборудования и персонала остаётся высоким. Для его уменьшения необходимо разработать метод планирования запросов к целевой системе.

Каждый запрос, пришедший от пользователя, делится на две категории: краткосрочный и долгосрочный. Под краткосрочным понимается такой запрос, время выполнения которого незначительно, а приоритет выполнения низкий. Примером могут служить следующие типы запросы: добавление, удаление, блокирование учётной записи пользователя, выделение ресурсов пользователю и т.п. Приоритет долгосрочного запроса выше. Он занимает больше ресурсного времени целевой системы. Это связано с тем, что для обработки запроса требуется опрос большого количества целевых систем со значительным количеством пользователей. Примерами таких запросов могут являться аудит целевой системы, генерация различных отчётов и другое.

В настоящее время планирование широко применяется в различных областях. Для операционных систем все разработанные методы планирования основываются на следующих трёх алгоритмах: невытесняющий, вытесняющий, кооперационный. Они изначально разрабатывались с целью автоматического управления процессами в компьютерах. Рассмотрим возможность адаптации данных алгоритмов к СУИД.

Суть работы невытесняющего алгоритма планирования заключается в следующем. Один процесс загружается на выполнение. Ресурс освобождается только после полного выполнения данного процесса. В контексте СУИД это видится как недостаток, так как запросы часто требуют согласования с пользователями. Следовательно, все остальные запросы из очереди будут вынуждены ожидать, и общая скорость работы снизится.

Альтернативным алгоритмом планирования является вытесняющий алгоритм. При его использовании операционная система сама передает управление от одной выполняемой программы другой в случае появления некоторых событий. Отдельно стоит отметить, что время передачи ресурсного времени от одного процесса другому не зависит от состояния задачи, что означает возможность её прерывания в любое время. При попытке адаптации к СУИД было отмечено, что указанная особенность рассматриваемого алгоритма крайне негативно влияет на работу с целевой системой. При поступлении нового запроса выполняющийся может быть прерван. А это противоречит требованию непрерывности работы долгосрочных запросов.

Основное отличие кооперационного алгоритма от предыдущего заключается в том, что следующая задача выполняется только после того, как текущая явно объявит себя готовой отдать ресурсное время. Таким образом, следующая задача из очереди выполняется только после того, как текущая либо завершится, либо приостановится в связи с синхронизацией с каким-либо объектом. Применение данного алгоритма в классическом виде связано с монополизацией процессора, что, в общем, является недостатком. Однако с точки зрения СУИД это становится достоинством. Это связано с тем, что управляющий процесс блокирует работу всех остальных запросов к целевой системе. Таким образом, удовлетворяется требование о непрерывности работы долгосрочных запросов. Для создания метода управления запросами СУИД на основе кооперационного алгоритма, уменьшающего риски простоя оборудования и персонала, необходимо ввести некоторые дополнительные требования. В частности одним из таких требований является уменьшение количества блокировок во время выполнения долгосрочных запросов.

Для реализации указанного требования время запуска долгосрочных запросов необходимо смещать. При этом оно должно определяться наименьшим количеством заблокированных задач за время от начала до завершения работы долгосрочного запроса. Это значит, что время запуска должно быть выбрано такое, при котором число кратковременных запросов, запущенных ранее в аналогичный период, минимально. Для этих целей предлагается алгоритм подсчёта наилучшего времени запуска длительных задач на основе истории запросов к целевой системе.

Внедрение в СУИД разработанного метода планирования задач положительно влияет на работу системы. Пиковые значения функции количества заблокированных запросов резко уменьшаются. Это способствует уменьшению времени ожидания запрошенного сотрудником ресурса. Данное обстоятельство увеличивает производительность труда на предприятии, так как персоналу больше не требуется ожидать окончания работы длительных запросов. Также следует отметить и другое достоинство применения предложенного метода. Оно заключается в снижении уровня загрузки целевой системы, что облегчает пользователям работу с ней.