

## Разработка кроссплатформенной программы удаления старых записей из архива ПТК «Систел» в рамках программы импортозамещения

А.В. Кочеткова, Т.Н. Кульман, А.О. Макаров

ООО «Систел»

Государственный университет «Дубна», филиал «Протвино»

### Аннотация

Программа импортозамещения требует развития кроссплатформенных приложений, особенно в критически важных областях, к которым относится электроэнергетика. В статье описывается процесс разработки кроссплатформенной программы по очистке архива программно-технического комплекса «Систел» с учётом требований по улучшению функциональных возможностей, надёжности и производительности по сравнению с имеющейся версией для ОС *Windows*. Комплекс предназначен для использования в АСДУ (автоматизированных системах диспетчерского управления). Первоначально программа по очистке архива была реализована в ОС *Windows*, в настоящее время создана кроссплатформенная программа для ОС Линукс.

**Ключевые слова:** импортозамещение, кроссплатформенность, C++, многопоточность, асинхронность, ОС Линукс, СУБД *PostgreSQL*

## Development of a cross-platform program for deleting old records from the «Systel» PTK archive as part of the import substitution program

A.V. Kochetkova, T.N. Kulman, A.O. Makarov

LLC «Systel»

Dubna State University, Protvino Branch

### Abstract

The import substitution program requires the development of cross-platform applications, especially in critical areas, which include the electric power industry. The article describes the process of developing a cross-platform program for cleaning the archive of the «Systel» software and hardware complex, taking into account the requirements for improving functionality, reliability and performance compared to the existing version for Windows OS. The complex is intended for use in ASDA (automated dispatch control systems). Initially, the archive cleanup program was implemented in Windows OS, currently a cross-platform program for Linux OS has been created.

**Keywords:** import substitution, cross-platform, C++, multithreading, asynchrony, Linux OS, PostgreSQL DBMS

## 1. Введение

Данная работа выполнена в рамках программы по импортозамещению в соответствии с Указом Президента РФ № 166 от 30.03.2022 на базе предприятия ООО «Систел», которое было основано в 1992 году и специализируется на выполнении работ в области телемеханики, автоматизации и диспетчеризации процессов передачи, распределения и учета электроэнергии.

За истекшие годы коллективом предприятия был разработан программно-технический комплекс (ПТК) [1], внедренный на большом количестве объектов энергетики. Комплекс предназначен для использования в АСДУ (автоматизированных системах диспетчерского управления).

Комплекс был реализован в ОС *Windows*. После объявления программы по импортозамещению в 2022 году в ООО «Систел» были начаты работы по созданию аналогичного комплекса в среде ОС Линукс [2].

При работе комплекса создаётся большая архивная БД (база данных), которую периодически надо освобождать от устаревших данных.

## 2. Цель и задачи работы

Целью данной работы являлась разработка программы для очистки архива от устаревших записей в ОС Линукс. Также необходимо было устранить определенные недостатки, выявленные в процессе эксплуатации версии под ОС *Windows*.

Очистка должна происходить таким образом, чтобы не создавать слишком высокую нагрузку на сервер с базой данных, где хранится архив, поскольку помимо очистки, в архив, с различными для каждой таблицы периодами, производится запись новых данных, объём которых за сутки может достигать до нескольких десятков миллионов.

Для достижения поставленной цели нужно было решить следующие задачи:

1. Создать кроссплатформенную программу. Для этого надо было использовать среду разработки, позволяющую создавать кроссплатформенные программы и применять только кроссплатформенные библиотеки.

2. Обеспечить непрерывную и бесперебойную работу программы, чтобы не требовался «ручной» перезапуск. Предусмотреть возможность различных нестандартных ситуаций при возникновении системных ошибок (например, при пропадании сети). Сделать все сообщения о различных исключениях информативными и понятными для обслуживающего персонала с записью подробной информации в лог-файл.

3. Разработать эффективную по производительности программу с использованием алгоритма многопоточной очистки архивов.

Языком программирования для создания программы был выбран C++, а именно C++17, поскольку его возможности наиболее полно реализованы в компиляторах под Линукс в отличие от более нового стандарта C++20.

## 3. Создание кроссплатформенного варианта программы

В качестве интегрированной среды разработки (*IDE*) была использована *MS Visual Studio Community 2019/2022* на ОС *Windows*. Эта среда предлагает широкий выбор инструментов, упрощающих процесс разработки, к примеру, технологию автодополнения *IntelliSense*. Поддерживается удалённая отладка проектов под ОС Линукс, которую также можно осуществлять, используя виртуальную машину с данной операционной системой.

Архив представляет собой БД, состоящую из нескольких таблиц. В качестве СУБД используется *PostgreSQL* [3] потому, что она обладает рядом важных свойств:

- Статус свободного программного обеспечения.
- Поддержка как реляционного, так и объектно-ориентированного подходов.
- Расширяемость.
- Возможность работы в ОС Линукс и *Windows*.
- Поддержка индексирования, механизма *ACID*-транзакций, функций, триггеров и так далее.

Имеются две кроссплатформенные библиотеки для взаимодействия с сервером *PostgreSQL*: *libpq* [4], написанная на языке Си, и *libpqxx* [5], написанная на C++. Для использования в программе была выбрана вторая библиотека, так как она предлагает более интуитивный интерфейс для общения с базой данных за счёт применения классов и шаблонов, включённых в стандарт языка C++. Кроме того, она позволяет

использовать *Unicode*. В дополнение к этому создатели *libpqxx* недавно анонсировали, что последняя версия их библиотеки оказалась быстрее в некоторых местах, чем *libpq* [6].

Для реализации оконного интерфейса была применена кроссплатформенная библиотека *wxWidgets* [7]. Библиотека *wxWidgets* является свободно распространяемым программным обеспечением с открытым исходным кодом.

Она предоставляет возможность построения интерфейса пользователя, который использует различные графические элементы (кнопки, текстовые поля и т. д.). Код, написанный с использованием *wxWidgets*, может быть откомпилирован без каких-либо изменений как на *Windows*, так и на *Linux*, при этом поведение программы будет гарантированно одинаковым в обеих операционных системах. На рисунке 1 показана диаграмма компонентов программы, показывающая взаимодействие описанных выше продуктов.



Рисунок 1: Диаграмма компонентов

#### 4. Развитие функциональных возможностей программы

В архивной БД имеется несколько таблиц, в которые записывается разная информация. Самой большой является таблица, содержащая суточную ведомость (диспетчерское название таблицы в электроэнергетике). В нее регулярно пишутся все величины, имеющиеся в системе. Регулярность записи устанавливается на каждом конкретном объекте, и, как правило, составляет 2 минуты. Но, иногда ставят одну минуту или даже чаще. Рассмотрим типичное количество записей за одни сутки в эту таблицу.

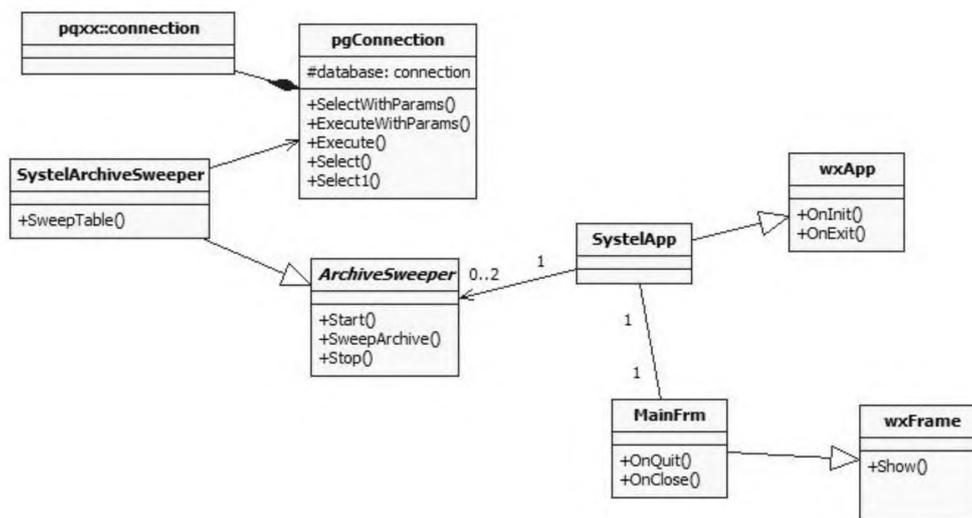
Предположим, что частота записи равна 2 минутам и имеется 100 тыс. величин. Тогда суммарное количество записей за одни сутки будет равно  $100000 * 24 * 30 = 72\,000\,000$  записей. При этом надо учитывать, что в таблицу постоянно идет добавление новых текущих записей. Поэтому нельзя блокировать таблицу непрерывным удалением старых записей.

На этапе запуска программа читает конфигурационный файл, в котором записана информация для подключения к базе данных и количество дней, в течение которых нужно оставить не удаленными записи для каждой таблицы. Количество дней может быть разным на разных предприятиях и задается администратором комплекса.

После чтения конфигурационного файла устанавливается соединение с архивной СУБД и проверяется, успешно ли оно было открыто.

Затем запускается поток, в котором в цикле проверяется условие продолжения очистки (наличие дней, информация за которые должна быть удалена), и если оно истинно, то вызывается функция, реализующая процесс очистки архива для каждой таблицы.

Наглядно отношения между классами программы демонстрируется на следующей диаграмме (рисунок 2).



**Рисунок 2:** Диаграмма классов

В таблице 1 помещено описание представленных классов. Жирным шрифтом выделены собственные классы, остальные – это классы из библиотек.

Далее перейдём к рассмотрению задач, непосредственно связанных с процессом разработки.

Одним из требований к программе была её непрерывная работа без вмешательства обслуживающего персонала. Для выполнения этого требования предпринято несколько мер, а именно:

- Главным классом программы является класс *wxApp* библиотеки *wxWidgets*. После прохождения этапа инициализации программа входит в бесконечный цикл, выйти из которого для завершения работы может только пользователь, нажав на кнопку закрытия программы.
- Для отслеживания состояния соединения была разработана функция *ConnectionGuard*. Её цель – отменять запрос, выполняющийся по отслеживаемому соединению, если оно прерывалось. Данный алгоритм призван решить следующую специфическую проблему: при переходе на новые сутки программа «Сервер SCADA» проводит некоторые манипуляции с архивом, которые могут прервать все остальные соединения к нему.

**Таблица 1.** Описание классов

Класс	Описание
<i>wxApp</i>	Класс библиотеки <i>wxWidgets</i> для реализации программы
<b><i>SystemApp</i></b>	Класс, реализующий программу, перегруженные методы <i>OnInit</i> и <i>OnExit</i> выполняются при запуске программы и выходе из него соответственно
<i>wxFrame</i>	Класс библиотеки <i>wxWidgets</i> , представляющий окно с графическим интерфейсом
<b><i>MainFrm</i></b>	Главное окно программы, в котором отображаются сообщения
<b><i>ArchiveSweeper</i></b>	Абстрактный класс, который предоставляет интерфейс для очистки архива
<b><i>SysTelArchiveSweeper</i></b>	Класс, реализующий очистку архива ОИК
<i>pqxx::connection</i>	Класс библиотеки <i>libpqxx</i> , представляющий собой подключение к базе данных
<i>pgConnection</i>	Инкапсулирует работу с объектом класса <i>pqxx::connection</i> , реализуя интерфейс для исполнения запросов к базе данных

Процесс очистки архива начинается в три часа ночи, чтобы не мешать работе «Сервера SCADA», которая идет интенсивно на границе суток.

Описанные выше действия обеспечивают бесперебойную работу программы.

При создании программы была также поставлена и решена задача сделать сообщения, записываемые в лог-файл в процессе работы программы, более информативными. Для этого было сделано следующее:

- При выполнении всех запросов перехватываются все возможные исключения, ге-

нерируемые библиотекой *libpqxx*, и заносятся в лог-файл. Каждое исключение сопровождается сообщением с описанием возникшей проблемы. При этом вместе с библиотечным сообщением об исключении записывается дополнительная информация.

- В процессе очистки в лог-файл заносится информация обо всех важных её этапах, например, о поиске записей для очистки или об их отсутствии в определённой таблице, с целью сделать ход выполнения программы более прозрачным.

## 5. Реализация многопоточности процесса очистки

Важным свойством программы является многопоточная очистка, заметно ускоряющая работу программы.

Ее суть заключается в том, что для каждой таблицы запускается свой поток, производящий удаление записей. Асинхронный запуск реализован с помощью функции *std::async*, которая является частью стандартной библиотеки C++ [8] и поддерживает многоплатформенность. Эта функция запускается с политикой *std::launch::async*, что гарантирует выполнение в отдельном потоке [9].

Важным моментом является то, что в каждом потоке запросов на удаление создаётся свой экземпляр класса для подключения к базе данных. Поэтому каждый поток будет работать со своим подключением к БД. Это было сделано по причине того, что библиотека *libpqxx* не допускает параллельного выполнения транзакций в рамках одного подключения.

После вызова функция *std::async* возвращает объект класса *std::future*, нужный для получения возвращаемого значения асинхронно вызванной функции. Чтобы его получить, необходимо с помощью метода *wait* данного класса дождаться, пока возвращаемое значение будет доступно.

В каждом потоке независимо друг от друга из процесса очистки исключаются таблицы, в которых на данный день нет устаревших записей.

## Выводы

В рамках данной работы было сделано следующее:

1. Создана кроссплатформенная программа очистки архива для ПТК «Систел» в среде ОС Линукс и ОС *Windows*.
2. Обеспечена непрерывная работа программы при различных исключительных ситуациях. Все сообщения о различных исключениях сделаны информативными и понятными для обслуживающего персонала с записью подробной информации в лог-файл.
3. Разработан и реализован алгоритм многопоточной очистки архива.

В настоящее время программа проходит тестирование на стендах ООО «Систел».

## Библиографический список

- [1] Рыкованов С.Н, Кульман Н.Ю., Ухов В.И. Развитие программного обеспечения верхнего уровня ПТК «Систел». Современные системы сбора, передачи, обработки и отображения информации на объектах электроэнергетики. Сборник докладов. Москва НЦ ЭНАС, 2006.
- [2] Debian [Электронный ресурс] : The Universal operating system / URL : <https://www.debian.org/intro/index#community> (дата обращения: 03.05.2023).
- [3] PostgreSQL [Электронный ресурс] : The World's Most Advanced Open Source Rela-

- tional Database / URL : <https://www.postgresql.org> (дата обращения: 04.05.2023).
- [4] PostgreSQL [Электронный ресурс] : Documentation: 13: Chapter 33. Libpq : The World's Most Advanced Open Source Relational Database / URL : <https://www.postgresql.org/docs/current/libpq.html> (дата обращения: 13.05.2023).
- [5] The C++ connector for PostgreSQL [Электронный ресурс] : libpqxx: the official C++ language binding for PostgreSQL / URL : <https://pqxx.org/libpqxx> (дата обращения: 13.05.2023).
- [6] 2023-01-12: At last! Faster than C [Электронный ресурс] : libpqxx: the official C++ language binding for PostgreSQL : URL : <https://pqxx.org/libpqxx/#2023-01-12-at-last-faster-than-c> (дата обращения: 13.05.2023).
- [7] Smart J., Hock K., Csomor St. Cross-Platform GUI Programming with wxWidgets / Crawfordsville : R.R. Donnelley & Sons, 2006. 744 с.
- [8] Гримм Р. Параллельное программирование на современном языке C++ / пер. с англ. В. Ю. Винника. Москва : ДМК Пресс, 2022. 618 с.
- [9] Concurrency support library [Электронный ресурс] : C++ reference : URL: <https://en.cppreference.com/w/cpp/thread/launch> (дата обращения: 03.05.2023).

#### **Об авторах:**

Кочеткова Анна Викторовна – ведущий инженер-программист ООО «Систел».

Кульман Татьяна Николаевна – к.т.н., доцент кафедры Информационных технологий филиала «Протвино» государственного университета «Дубна».

Макаров Артём Олегович – студент 3 курса кафедры Информационных технологий филиала «Протвино» государственного университета «Дубна».