Задача - Создать гибкую и потокобезопасную систему логирования на C++ с асинхронной обработкой сообщений, поддержкой разных уровней логирования, пользовательских шаблонов и вывода в консоль и файл.

**Архитектура многопоточного логгера**

* Основной поток приложения вызывает методы логгера для записи сообщений, которые помещаются в потокобезопасную очередь messageQueue под защитой мьютекса queueMutex.
* Уведомление о новых сообщениях происходит через std::condition\_variable cv.
* Отдельный рабочий поток workerThread ожидает уведомления, извлекает сообщения из очереди, форматирует и выводит их в консоль и/или файл.
* При завершении работы поток получает сигнал через флаг exitFlag и обрабатывает оставшиеся сообщения.

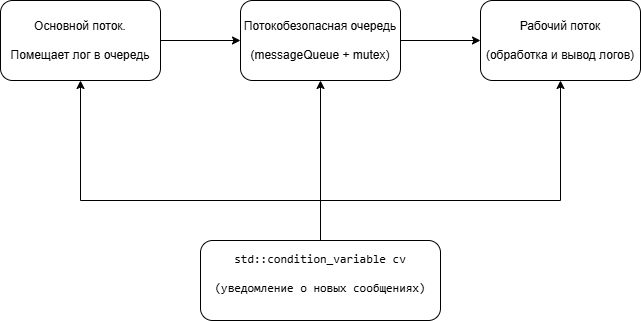
**Обеспечение потокобезопасности**

* Доступ к очереди и настройкам логгера защищён мьютексом, что исключает гонки данных.
* Условная переменная позволяет рабочему потоку эффективно ждать новых сообщений без активного опроса.
* Асинхронная обработка снижает задержки в основном потоке и повышает производительность.

**Преимущества**

* Разделение логики записи и обработки повышает отзывчивость приложения.
* Гарантируется корректность и целостность логов при параллельном доступе.
* Обеспечивается гибкая настройка формата и места вывода.

**Схема многопоточности логгера**



* Основной поток помещает лог-сообщения в очередь под защитой мьютекса и уведомляет рабочий поток.
* Рабочий поток ожидает уведомления, извлекает сообщения и записывает их в консоль и/или файл.
* При завершении приложения рабочий поток обрабатывает оставшиеся сообщения и корректно завершается.

1. **Основной поток**  
   Выполняет вызовы методов логгера для записи сообщений. Формирует лог-сообщения и помещает их в очередь.
2. **Потокобезопасная очередь**  
   Хранит лог-сообщения, поступающие из разных потоков. Доступ к очереди защищён мьютексом (std::mutex), что исключает гонки данных.
3. **Условная переменная (std::condition\_variable)**  
   Используется для уведомления рабочего потока о появлении новых сообщений в очереди. Позволяет рабочему потоку эффективно ждать без постоянного опроса.
4. **Рабочий поток**  
   Отдельный поток, который ожидает уведомления, извлекает сообщения из очереди и записывает их в консоль и/или файл согласно настройкам.

Основной поток → помещает сообщение → потокобезопасная очередь → уведомляет → рабочий поток → обрабатывает и выводит сообщение.