

МНОГОПОТОЧНОСТЬ, АСИНХРОННОСТЬ И КОНКУРЕНТНОСТЬ В C++

Многопоточность играет важную роль в современной разработке программного обеспечения. Современные компьютерные системы обычно имеют несколько ядер процессора или даже многопроцессорные системы, поэтому эффективное использование многопоточности позволяет распараллеливать выполнение задач и улучшать производительность приложений.

Функционально многопоточность позволяет:

- многопоточность позволяет параллельно выполнять несколько задач, распределяя их между различными потоками или ядрами процессора. Это может значительно ускорить выполнение программы и повысить общую производительность системы;

- путем распараллеливания задач и использования нескольких потоков можно более эффективно использовать доступные ресурсы компьютера, такие как процессорное время, память и ввод-выводные операции;

- многопоточность позволяет разделить сложные задачи на более мелкие подзадачи, которые могут выполняться параллельно. Это упрощает программирование и обеспечивает более структурированное и модульное решение задачи;

В исследовании были рассмотрены стандартные средства работы с потоками в C++ такие как:

- библиотека `thread`;
- библиотека `mutex`;
- библиотека `condition variables`;
- библиотека `future`.