

khuttun /
PolyM

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security



PolyM is a very simple C++ message queue intended for inter-thread communication

MIT license

162 stars 40 forks 16 watching Activity

Public repository

master



Branches Tags



littleWhitePoplar and khuttun Add a size interface for the Que... on Oct 17, 2022 20

[View code](#)

README.md

PolyM

PolyM is a very simple C++ message queue intended for inter-thread communication. There are three major requirement driving the design of PolyM:

- The design should be simple and lightweight
- The design should support delivering any kind of data as the message payload
- There should be no copies made of the message payload data when passing the message through the queue

PolyM (**P**olymorphic **M**essage **Q**ueue) fulfills these requirements by usage of C++ move semantics.

Moving Messages Through the Queue

When a message with payload data (`PolyM::DataMsg<PayloadType>`) is created, the payload is allocated from the heap. When the message is put to the queue, it is moved there with so called "virtual move constructor". The virtual move constructor is what enables the polymorphism of the message queue: any message type derived from the base `PolyM::Msg` type can be moved to the queue. When a message is read from the queue, it is moved out from the queue to be the responsibility of the receiver.

All the message types are movable, but not copyable. Once the message is put to the queue, the sender cannot access the message data anymore. A message always has a single owner, and on send-receive scenario, the ownership is transferred

Sender -> Queue -> Receiver



Messaging Patterns

PolyM supports two kinds of messaging patterns:

- One way
 - Messages are put to the queue with `PolyM::Queue::put()` . `put()` will return immediately.
 - Messages are read from the queue with `PolyM::Queue::get()` . `get()` will block until there is at least one message available in the queue (or until timeout occurs, if specified).
- Request-response
 - A request is made with `PolyM::Queue::request()` . `request()` will block until a response is given. The request message can be read from the queue just as any other message with `get()` .
 - A response is given with `PolyM::Queue::respondTo()` . `respondTo()` will return immediately.

Releases

No releases published

Packages

No packages published

Contributors 3



khuttun Kalle Huttunen



lp35 LouisP



littleWhitePoplar

Languages

● C++ 96.5% ● CMake 3.5%