INE5418 Distributed Systems - Mutual Exclusion Policy

Bruno Marques do Nascimento

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 IPC Class Reference

Class that supports the user with the IPC API usage.

```
#include "ipc.h"
```

Public Member Functions

- IPC (int id)

  Constructor.
- void init ()

  Call the file descriptor initialization.
- void **start** ()
- void start_listening ()

  Start listening for other process messages.
- message_t receive_msg ()

  Receive a message from other process.
- void send_msg (message_t)

  Send a message to another process.

Private Member Functions

- void create_process_fd ()

  Create process socket file descriptor.
- void configure_process_fd ()

  Configure process socket file descriptor.

Private Attributes

- int id_

  Process id attached to this IPC.
- int opt_

  Always 1, relative to socket operation.
- int rcv_process_fd_

  File descriptor, to received messages.
- int send_process_fd_

  File descriptor, to send messages.
- int address_length_

  Length of the address.
- int socket_port_

  Socket port that Process will be bind.
- struct sockaddr_in address_

  Struct responsible to configure sockets.

4.1.1 Detailed Description

Class that supports the user with the IPC API usage.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 IPC (int *id*)

Constructor.

Set initial attributes values.

Parameters

id	the id of the process that this IPC will be responsible for.

4.1.3 Member Function Documentation

4.1.3.1 void configure_process_fd ( ) [private]

Configure process socket file descriptor.

4.1.3.2 void create_process_fd ( ) [private]

Create process socket file descriptor.

4.1.3.3 void init ( )

Call the file descriptor initialization.

4.1.3.4 message_t receive_msg ( )

Receive a message from other process.

Returns

the received message.

4.1.3.5 void send_msg (message_t *msg*)

Send a message to another process.

Parameters

msg	message to be sent.

4.1.3.6 void start_listening ( )

Start listening for other process messages.

4.1.4 Member Data Documentation

4.1.4.1 struct sockaddr_in address_ [private]

Struct responsible to configure sockets.

4.1.4.2 int address_length_ [private]

Length of the address.

4.1.4.3 int id_ [private]

Process id attached to this IPC.

4.1.4.4 int opt_ [private]

Always 1, relative to socket operation.

4.1.4.5 int rcv_process_fd_ [private]

File descriptor, to received messages.

4.1.4.6 int send_process_fd_ [private]

File descriptor, to send messages.

4.1.4.7 int socket_port_ [private]

Socket port that Process will be bind.

The documentation for this class was generated from the following files:

- include/ipc.h
- src/ipc.cpp

4.2 message_t Struct Reference

Public Attributes

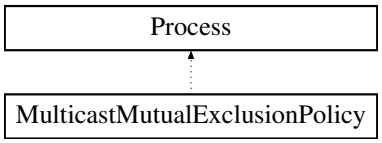
- char **source** [4]
- char **destination** [4]
- char **type** [4]
- char **data** [100]

The documentation for this struct was generated from the following file:

- include/message.h

4.3 MulticastMutualExclusionPolicy Class Reference

Inheritance diagram for MulticastMutualExclusionPolicy:



Public Member Functions

- **MulticastMutualExclusionPolicy** (int id, int n_process)
- void **run** ()

Private Member Functions

- void **process_message** ()
- int **destination** ()
- void **send_token** ()
- void **server_resource_request** ()
- void **server_resource_release** ()
- void **client_resource_request** ()
- void **client_resource_release** ()
- void **start_broadcast** ()

Private Attributes

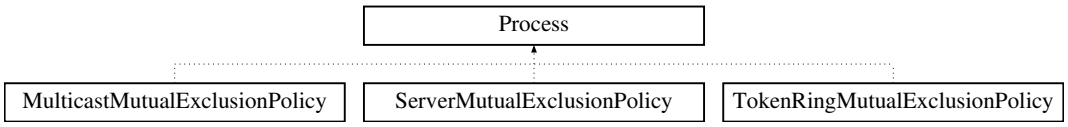
- std::queue< int > **work_queue**

The documentation for this class was generated from the following files:

- include/multicast_policy.h
- src/multicast_policy.cpp

4.4 Process Class Reference

Inheritance diagram for Process:



Public Member Functions

- **Process** (int id, int n_process)

Protected Member Functions

- void **compute** ()
- int **random_computation_time** ()

Protected Attributes

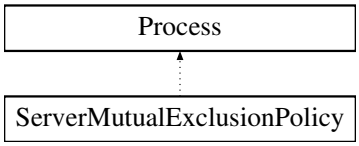
- int **id_**
- int **n_process_**
- int **computation_time_**
- IPC **ipc_**
- message_t **received_message_**

The documentation for this class was generated from the following files:

- include/process.h
- src/process.cpp

4.5 ServerMutualExclusionPolicy Class Reference

Inheritance diagram for ServerMutualExclusionPolicy:



Public Member Functions

- **ServerMutualExclusionPolicy** (int id, int n_process)
- void **run** ()

Private Member Functions

- void **process_message** ()
- int **destination** ()
- void **send_token** ()
- void **server_resource_request** ()
- void **server_resource_release** ()
- void **client_resource_request** ()
- void **client_resource_release** ()
- void **start_broadcast** ()

Private Attributes

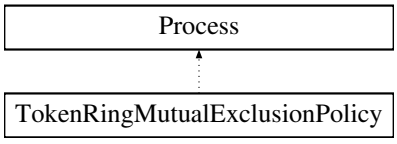
- std::queue< int > **work_queue**

The documentation for this class was generated from the following files:

- include/server_policy.h
- src/server_policy.cpp

4.6 TokenRingMutualExclusionPolicy Class Reference

Inheritance diagram for TokenRingMutualExclusionPolicy:



Public Member Functions

- **TokenRingMutualExclusionPolicy** (int id, int n_process)
- void **run** ()

Private Member Functions

- void **process_message** ()
- int **destination** ()
- void **send_token** ()

Additional Inherited Members

The documentation for this class was generated from the following files:

- include/token_ring_policy.h
- src/token_ring_policy.cpp

Chapter 5

File Documentation

5.1 include/ipc.h File Reference

High level approach to the low level IPC API.

```
#include <netinet/in.h>
#include "message.h"
```

Classes

- class IPC

  Class that supports the user with the IPC API usage.

5.1.1 Detailed Description

High level approach to the low level IPC API.

Author

Bruno Marques do Nascimento

Date

29/04/2018

Version

1.0

This file is a header file that contains the functions and variables of a high level approach to the IPC API.

Index