Extended Message Queues

A Subscriber-Broker-Publisher model has been implemented using Message Queues that extend over a network.

Design

A publisher makes a TCP connection to a broker in order to add any message to the queue. The messages are sorted topic wise and the list of topics is maintained by the subscriber in a simple text file. The broker need not know of all topics - as the implementation of the broker does not require it.

Similarly, a subscriber also makes a TCP connection to a broker in order to retrieve any message from the queue. The subscriber sets a topic for which it wants messages, and then can query the broker for any new messages in that topic.

Most of the core logic of the extended message queue interface is implemented by the broker. In order to store messages on disk, the broker creates a temporary directory in `/tmp` using `mkdtemp`. Then the broker forks in order to handle the publisher-broker and subscriber-broker connections separately.

When the broker receives a message from a publisher, the broker first checks the topic of the message. If it is a new topic, it creates a subdirectory in the temporary directory it first created with the topic as the name of the directory. Then it stores the message as a file with filename being the time it received the message.

When the subscriber requests the broker for a message, it sends the timestamp of the last message it has seen (0 if it hasn't seen any). The broker then compares this timestamp received with the files stored on disk, and sends the next message according to the timestamp value. This method of using a timestamp ensures FIFO. The message structure has a special way to indicate if there are no new messages.

In order to ensure that the broker stores messages only for `MESSAGE_TIME_LIMIT` seconds, it is setup to receive `SIGALRM` at fixed intervals. If while waiting for a request it receives this signal, it cleans up all older files, using the timestamp in the filename.