

MC705 - Programação de Sistemas Distribuídos

Projeto 1 - Ordenação Total de Mensagens em um Sistema Distribuído

Glauber de Oliveira Costa e Tatiana Al-Chueyr Pereira Martins

1. Introdução

Este projeto consiste na implementação do algoritmo proposto no artigo *Reliable Broadcast Protocols*, por J. M. Chang e N. F. Maxemchuk em [I].

2. Detalhes de implementação

Para a elaboração desse projeto utilizamos a linguagem de programação Python, versão 2.4.1.

Consistem nos módulos principais do projeto: [node.py](#), [receiver.py](#), [reformer.py](#) e [sender.py](#). Não menos importantes, foram utilizados módulos auxiliares: [broadcast.py](#), [defines.py](#), [QueueNorris.py](#) e [XmlMessage.py](#). A seguir, há uma breve descrição de cada um deles.

[node.py](#)

É o módulo principal do programa, de onde são iniciadas as *threads* de recebimento, envio e reforma. Ele é responsável pelo controle do programa de modo geral.

[receiver.py](#)

Implementa os métodos de recebimento do projeto, dando tratamento adequado de acordo com a mensagem recebida.

[reformer.py](#)

Módulo que trata os métodos associados à fase de reforma, possibilitando a mesma. É utilizado para implementar os procedimentos descritos na fase 2 presente em [I].

[sender.py](#)

Trata os métodos de envio de mensagens.

[broadcast.py](#)

Responsável pela simulação de comunicação broadcast. Isso se fez necessário dada a especificação do projeto.

[defines.py](#)

Faz a abstração da aplicação, possuindo estruturas auxiliares, utilizadas pelos demais módulos.

[QueueNorris.py](#)

Estrutura de dados utilizada para o armazenamento de mensagens. Utiliza as estruturas lista e dicionário, presentes em Python, agregando as funcionalidades necessárias pelo programa para controle de informações.

XmlMessage.py

Módulo que cria mensagens XML e/ou as interpreta. Esse módulo trata as mensagens declaradas na especificação do projeto. É utilizado o parser `xml.dom.minidom`, presente na distribuição padrão de Python.

3. Execução

Os parâmetros de configuração, assim como o conjunto de nós participantes, será definido no arquivo `broadcast.conf`, que deve estar de acordo com a especificação do projeto.

Para se rodar o programa, deve-se utilizar a seguinte linha de comando em cada nó participante:

```
$ python node.py broadcast.conf
```

4. Referências

[1] CHANG, Jo-Mei; MAXEMCHUK, N. F. - *Reliable Broadcast Protocols* - ACM Transactions on Computer Systems, Vol. 2, No. 3, agosto de 1984, páginas 251-273.