Universidade do Estado do Rio de Janeiro
Instituto de Matemática e Estatística
Departamento de Informática e Ciência da Computação

Relatório sobre o Sistema Publicadores e Assinantes

Disciplina: Sistemas Operacionais I
Docente: Hélio do Nascimento Cunha Neto
Discente: Camila Catalão de Oliveira

Rio de Janeiro, 23 de Outubro de 2024

Objetivo:

Criar um modelo de publicação/assinantes (**pub/assin**) utilizando a linguagem python3. O modelo é um padrão de mensagens no qual os editores enviam mensagens para um gerenciador de mensagens e os assinantes expressam interesse em receber determinadas mensagens. O gerenciador de mensagens é responsável por entregar as mensagens aos clientes inscritos. Os publicadores e os assinantes se comunicam sem saberem da existência um do outro.

Bibliotecas:

1. threading
2. queue
3. time

Análise:

A implementação está composta por três classes, Publisher, Subscriber e Broker. Cada classe tem suas responsabilidades nesse sistema. A classe Publisher é responsável por enviar as mensagens para o Broker. Ela usa o método __init__() que inicializa seus atributos com seus respectivos parâmetros (linhas 57 até 62) e no método run(self) utiliza esses mesmos parâmetros para dizer quem publicou alguma mensagem e os detalhes dessa mensagem. Além disso ele também chama o método divulgação, que pertence a classe *Broker*, para para fazer a divulgação dessa mensagem caso haja inscritos no respectivo tópico (linha 68).

A classe *Broker* é composta pelo métodos *__init__(), divulgacao(), inscrição()* e *notify_inscritos()*. O primeiro inicializa as fila de mensagem e de inscritos por tópico. O segundo faz a divulgação das tópicos publicados, ele verifica se o tópico está presente no dicionário e se há inscritos para esse tópicp, se for verdade ele irá criar uma fila de tópicos, vai colocar a mensagem na fila e vai notificar cada inscrito presente nela, senão ele irá avisar que essa mensagem será descartada pois não há assinantes para ela. O método de inscrição cria um novo tópico caso ele não esteja no dicionário, e adiciona um novo assinante ao tópico específico. O último método é o de notificação que notifica para cada assinante do tópico enquanto a fila não estiver vazia e depois marca essa tarefa como completa. A classe Subscriber fica num loop infinito esperando por mensagens na fila. Cada inscrito tem uma fila própria e ao receber a mensagem o assinante imprime a mensagem e marca essa atividade como concluída.

Agora é colocar o código para funcionar, e para isso para verificar o funcionamento do código é necessário instanciar as três classes. Para esse exemplo foi feito manualmente a inscrição em cada tópico para cada inscrito e a publicação das mensagens para cada publicador, em seguida iniciar as threads de cada publicador.