**Introdução**

Este trabalho tem como objetivo implementar uma aplicação produtor-consumidor. Aplicações produtor-consumidor se caracterizam por múltiplas threads utilizando os mesmos objetos, de modo que instruções críticas sejam realizadas de forma sincronizada entre as threads.

Criamos uma aplicação chamada Stalker, que realiza dois tipos de tarefas ao mesmo tempo: baixa páginas do site URI Online Judge (produtor) e extrai informações relevantes das páginas (consumidor). A aplicação foi implementada utilizando a linguagem de programação Python.

**Descrição do problema**

O problema do produtor-consumidor consiste em múltiplos processos que compartilham o mesmo buffer. O produtor insere dados no buffer, enquanto o consumidor retira dados do buffer.

Quando múltiplos processos acessam os mesmos dados simultaneamente, existe a possibilidade de ocorrer uma troca de contexto durante a atualização dos dados. Como instruções de escrita em memória podem não ser atômicas, uma eventual troca de contexto poderia fazer com que processos diferentes utilizem cópias diferentes dos mesmos dados, gerando inconsistência nos dados.

Cada linguagem de programação resolve o problema do acesso concorrente de forma diferente. Internamente, as linguagens chamam uma instrução atômica do sistema operacional, durante a qual é garantido que não haverá troca de contexto. Para o programador, isso pode ser abstraído de diferentes formas: mutex, semáforos, métodos sincronizados, etc.

**Descrição das implementações**