Laboratório 8
Sincronização com semáforos

Programação Concorrente (ICP-361)
Profa. Silvana Rossetto

[1]Instituto de Computação/UFRJ

Introdução

O objetivo deste Laboratório é praticar o uso de **semáforos** para implementar exclusão mútua e sincronização condicional em programas concorrentes, explorando o padrão produtor/consumidor.

Atividade 1

Objetivo: Projetar e implementar um programa concorrente em C — usando apenas semáforos como mecanismo de sincronização — que usa uma variação do padrão produtor/consumidor.

Descrição: Implemente um programa concorrente onde UMA thread PRODUTORA gera uma sequência consecutiva de N **números inteiros** e os deposita no **canal de inteiros** de tamanho M ($M << N$) que será compartilhado com VÁRIAS threads CONSUMIDORAs. Os valores de N, M devem ser lidos da entrada do programa.

A thread PRODUTORA insere todos os elementos do buffer de uma vez (precisa esperar o buffer ficar vazio). As threads CONSUMIDORAS retiram os números — um de cada vez — e avaliam sua primalidade, usando a seguinte função:

```
int ehPrimo(long long int n) {
   int i;
      if (n<=1) return 0;
      if (n==2) return 1;
      if (n%2==0) return 0;
      for (i=3; i<sqrt(n)+1; i+=2)
         if(n%i==0) return 0;
      return 1;
```

Ao final do programa (depois que os N números foram processados), deverá ser retornado: **a quantidade de números primos encontrados (para avaliar a corretude do programa); e (ii) a thread consumidora VENCEDORA (aquela que encontrou a maior quantidade de primos).**

Entrega do laboratório:

Disponibilize o código implementados na **Atividade 1** em um ambiente de acesso remoto (GitHub ou GitLab). Use o **formulário de entrega** desse laboratório para enviar o link do repositório do código implementado e responder as questões propostas.