

Simulação de uma Fábrica de Canetas

Este trabalho visa desenvolver uma aplicação para simular uma fábrica de canetas em C com *Pthreads*, com semáforos e variáveis de condição, no Linux. A aplicação deve ser executada com 06 (seis) *threads* ao todo, as quais interagem nas atividades de: fabricação de canetas, armazenamento da matéria prima/canetas produzidas e na compra/venda. Às 06 *threads* serão atribuídos identificadores, aqui chamados de *ranks*: Criador: *rank 0*, Depósito de Matéria Prima: *rank 1*, Célula Fabricação de Canetas: *rank 2*, Controle: *rank 3*, Depósito de Canetas: *rank 4* e Comprador: *rank 5*. A função *main()*, ponto de entrada da aplicação, será executada pela *thread* de *rank 0* que recuperará os argumentos de entrada e chamará a função criador. As demais *threads*, com *rank > 0*, começam a executar diretamente nas suas respectivas funções.

Os argumentos de entrada configuram a execução da aplicação desta aplicação são:

- 01 - int - qtde matéria prima existente no depósito para fabricação das canetas;
- 02 - int - qtde unidades enviadas por interação do Dep. Mat. Prima à célula de fabricação das canetas;
- 03 - int - tempo em segundos entre cada envio de Matéria Prima às células de fabricação;
- 04 - int - tempo em segundos para fabricar UMA caneta;
- 05 - int - qtde máxima de canetas que podem ser armazenadas no depósito de canetas simultaneamente;
- 06 - int - qtde de canetas compradas a cada interação pelo comprador; e
- 07 - int - tempo em segundos de espera entre as compras de canetas.

Após suas criações, todas as *threads* estarão em execução independentemente umas das outras e serão sincronizadas por semáforos e variáveis de condição onde necessário. A *thread rank 0*, criador, será a responsável por criar as demais *threads* (vide (1) na Rede de Petri). A *thread rank 4*, Depósito de Canetas, determina para o controle, a cada iteração, a quantidade de slots disponíveis para receber canetas produzidas (vide (2) na Rede de Petri). O depósito de canetas possui capacidade limitada de armazenamento. Quando sua capacidade máxima é atingida, o envio de matéria prima, a fabricação e o envio de canetas ao depósito de canetas devem ser suspensos. A *thread rank 3*, Controle, determina se o Depósito de Matéria Prima na *thread rank 1*, pode transferir matéria prima para a Célula de Fabricação de Canetas (*thread rank 2*) e também determina se tal Célula de Fabricação (*rank 2*) pode produzir canetas (vide (3) e (4) na Rede de Petri). Cada caneta produzida pela Célula de Fabricação de Canetas é “enviada” ao Depósito de Canetas (vide (5) na Rede de Petri), o qual as armazena até que a *thread* comprador faça pedidos de compra de canetas. A *thread rank*, comprador, iterativamente solicita a compra de canetas (vide (6) na Rede de Petri) e o depósito, se possível, as “envia” na quantidade especificada nos argumentos ao comprador (vide (7) na Rede de Petri). Se houver no depósito de canetas a quantidade de canetas solicitada, então esta quantidade solicitada é simplesmente despachada ao comprador. Se houver uma quantidade inferior em estoque, são enviadas apenas as canetas disponíveis. Se não houver caneta armazenada, então é enviada uma mensagem ao comprador explicando que o seu pedido não pôde ser atendido.

Quando o comprador receber as canetas do depósito de canetas (ou informação de que a compra não pôde ser feita), o comprador avisa ao criador este recebimento (vide (8) na Rede de Petri). Apenas o criador tem a permissão para imprimir na saída padrão. Quando todo o estoque de matéria prima acabar E não houver mais canetas produzidas no depósito de canetas, a aplicação é encerrada por determinação do criador. As impressões na saída padrão devem permitir verificar a execução da aplicação de maneira clara e objetiva.

