Célula flexível de manufatura

Semáforos POSIX

O padrão POSIX 1003.1B define várias operações envolvendo semáforos, dentre as quais as mais significativas são (para mais informações vide as páginas de manual UNIX):

```
#include <semaphore.h>

sem_t sem ;

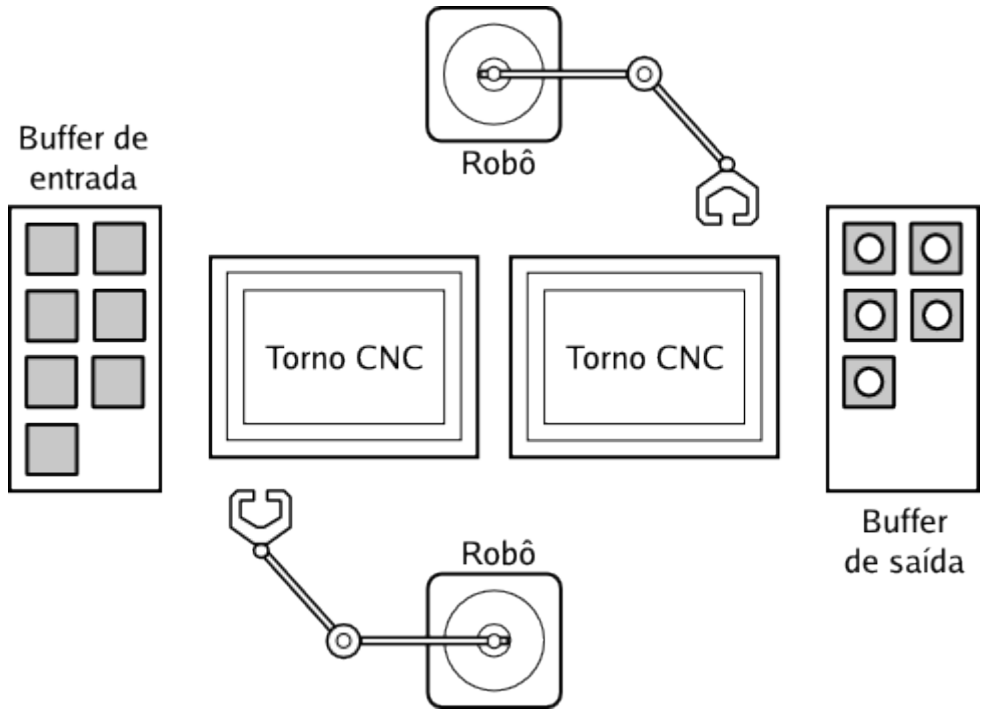
// inicializa o semáforo com o valor init_value
int sem_init (sem_t *sem, int is_shared, unsigned int init_value);

// P(sem) ou down(sem)
int sem_wait (sem_t *sem);

// V(sem) ou up(sem)
int sem_post (sem_t *sem);

// destroi o semáforo, liberando as tarefas bloqueadas
int sem_destroy (sem_t *sem);
```

Usando semáforos e threads POSIX, deve ser construído um programa em C para simular uma célula flexível de manufatura. A célula é composta por tornos automáticos, robôs industriais que continuamente carregam/descarregam os tornos e buffers para a entrada e saída de peças, conforme indicado na figura abaixo.



Existem vários aspectos de sincronização a considerar:

- um torno só pode usinar uma peça de cada vez;
- um robô só pode segurar uma peça de cada vez;

- um buffer só pode ser acessado por um robô de cada vez;
- um torno só pode ser acessado por um robô de cada vez, seja para carga ou descarga;
- qualquer robô pode acessar qualquer torno e qualquer buffer.

Situação 1

Nesta situação, há um torno e um robô.

- As peças a usinar chegam no buffer 1 e as peças já usinadas são retiradas do buffer 2, através de processos externos que não é necessário simular.
- O robô continuamente pega peças do buffer 1 e coloca no torno para usinar; após a usinagem, o robô descarrega a peça do torno e a coloca no buffer 2.
- Cada buffer é implementado por um vetor de inteiros. O robô e o torno são implementados por threads separadas.
- Uma usinagem demora um tempo aleatório entre 1s e 5s; uma operação de carga ou descarga de peça do torno ou de um buffer demora 1s cada.

Situação 2

Nesta situação, há três tornos e dois robôs trabalhando totalmente em paralelo: qualquer robô pode carregar/descarregar qualquer torno. O restante é idêntico à situação anterior.

Avaliação de desempenho

Para medir o desempenho das soluções propostas, sugere-se seguir o seguinte procedimento:

- Definir tempos fixos para os procedimentos de usinagem, carga e descarga.
- Encerrar o programa quando N peças tiverem sido depositadas no buffer 2.

Desta forma, é possível medir o tempo necessário para processar N peças, e por conseqüência, a eficiência da solução implementada. Por exemplo, considerando 20 peças, 1s para cada operação de carga/descarga e 5s para cada usinagem de peça, podem ser calculados os seguintes tempos mínimos **teóricos**:

- para um robô e um torno: 20 * (1s + 1s + 5s + 1s + 1s) = 180s
- para dois robôs e três tornos: 180s / 3 = 60s (considerando que os tornos nunca ficam parados; talvez essa situação não possa ser alcançada)

A entregar

- Os códigos-fonte dos dois programas implementados
- Um relatório (no formato correto) contendo:
  - Pseudo-código dos programas implementados
  - Análise do risco de impasses nos programas e discussão das possibilidades de evitá-los
  - análise do desempenho das soluções implementadas

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=so:celula_flexivel_de_manufatura

Last update: **2010/04/16 10:18**