MAC 438 - Programação Concorrente - Primeiro Semestre de 2002

Segundo Exercício-Programa: O Problema da Ponte Com Só uma Pista

Data de Entrega: 13 de maio de 2002

Este exercício deve ser desenvolvido em equipes de duas pessoas, a fim de suscitar discussão. Dúvidas sobre o enunciado devem ser enviadas para `reverbel-mac438@ime.usp.br`.

1 O problema

Carros vindos do norte e carros vindos do sul chegam a uma ponte com só uma pista. Os que viajam no mesmo sentido podem atravessar a ponte ao mesmo tempo, mas os que viajam em sentido oposto não podem. Modele os carros como *threads* e use semáforos para sincronização entre *threads*. Note que os carros devem sair da ponte na mesma ordem em que entraram (não é possível ultrapassagem, pois a ponte só tem uma pista).

2 Suas tarefas

Vimos em classe ema solução que dá preferência aos carros que vão do sul para o norte. Modifique-a para obter uma solução imparcial. Implemente as duas soluções e meça o tempo médio de espera para entrar na ponte em cada direção. Em outras palavras:

1. Implemente a solução que dá preferência aos carros que vão do sul para o norte.
2. Rode esse programa com uma certa população inicial de carros: n carros inicialmente no lado sul e n carros inicialmente no lado norte.
3. Meça o tempo médio de espera para entrar na ponte pelo lado sul e o tempo médio de espera para entrar pelo lado norte.
4. Implemente a solução imparcial.
5. Refaça as etapas 2 e 3 com o novo programa e com a mesma população inicial de carros.
6. Escreva um relatório com os resultados que você obteve.

Faça o tempo de travessia desejado de cada carro (isto é, o tempo que o carro leva para atravessar a ponte na ausência de carros mais lentos à frente dele) ser escolhido aleatoriamente dentro de um certo intervalo. Esse tempo é sorteado no momento que o carro entrar na ponte. O tempo de permanência no mesmo lado da ponte, também aleatório, deve ser sorteado quando um carro começar a rodar ou quando ele sair da ponte.

Seu programa deve receber seis parâmetros: a quantidade inicial de carros de cada lado da ponte, as extremidades do intervalo para sorteio do tempo de travessia desejado, as extremidades do intervalo para sorteio do tempo de permanância do mesmo lado da ponte, e um valor que define quando a execução do programa acaba (após t travessias, ou algo assim).

Você pode ter que experimentar diferentes combinações desses parâmetros para obter situações "interessantes". Por exemplo: com poucos carros, travessia rápida e alto tempo de permanência no mesmo lado a chance de concorrência é pequena. Uma situação assim não é interessante, pois o tempo médio de espera será muito próximo de zero.

3 Seu arsenal

Você deve fazer este EP em C, usando o pacote *LinuxThreads*, que implementa o padrão *pthreads* (*POSIX Threads*) para Linux 2.x. A biblioteca LinuxThreads e sua documentação estão disponíveis na nossa rede Linux. Para mais informações sobre o pacote LinuxThreads, veja

```
http://pauillac.inria.fr/~xleroy/linuxthreads/
```

Todas as funções POSIX Threads tem nome começando com "`pthread_`" ou com "`sem_`" (estas são operações sobre semáforos). Para listar essas funções, diga

```
apropos pthread_
apropos sem_
```

Neste exercício você não precisará de todas as funções do pacote *LinuxThreads*. Você usará funções básicas de gerenciamento de threads (`pthread_create`, por exemplo) e operações sobre semáforos (`sem_init`, `sem_wait`, `sem_post` e `sem_destroy`). Não é permitido o uso de operações sobre mutexes (`pthread_mutex_...`) ou condições (`pthread_cond_...`). (Mutexes e condições servem para se implementar monitores, que serão o assunto do próximo EP.)

Vários tutoriais sobre pthreads estão disponíveis na Internet. Eis um desses tutoriais:

```
http://dis.cs.umass.edu/~wagner/threads_html/tutorial.html
```

Há também um texto sobre pthreads na pasta 05 do xerox do CAMAT. (Embora seja um bom texto, não é a melhor referência para este EP. Ele explica muito bem mutexes e condições, mas não fala sobre semáforos.)

Este é um FAQ com informações variadas sobre programação com threads (incluindo "Microsoft-style threads", caso você precise um dia...):

```
http://www.serpentine.com/~bos/threads-faq/
```

4 Sobre a entrega

Você deverá entregar tres coisas:

- um arquivo tar.gz contendo sua solução (arquivos-fonte, makefile, README, ...);
- uma listagem impressa dos seus arquivos-fonte;
- um relatório impresso.

Por favor, entregue o relatório e a listagem **na secretaria do MAC**, em um saco plástico devidamente fechado, contendo também um disquete com o arquvo tar.gz.

EPs atrasados não serão aceitos!

Bom trabalho!