

EP2

Sistemas Operacionais

Bruno Rafael Aricó

Nícolas Nogueira Lopes da Silva

17 de Outubro de 2016

1. Estrutura do EP
2. Experimentos
3. Resultados

Estrutura do EP

Como requisito o programa é composto por $2n$ threads ciclistas, um vetor compartilhado pista de tamanho d onde cada posição do vetor corresponde a $1m$ da pista. Com as duas formas de execução:

- u - Todos os ciclistas andam a 60 km/h constantemente;
- v - Primeira volta com os ciclistas a 30 km/h e nas outras a velocidade é decidida de forma aleatória (entre 30km/h e 60km/h);

Os $2n$ ciclistas estão separados em duas equipes e cada equipe possui um vetor ordenado de integrantes. A struct integrantes guarda informações sobre o id da thread correspondente, velocidade, distância, tempo e quantidade de voltas.

Usamos o método `wait` para sincronizar as threads onde cada interação simula o movimento de um trecho inteiro com tempo real.

A pista é composta por d trechos e cada trecho possui um semáforo e um vetor de ID de threads que ocupam as duas faixas da pista e uma posição livre deste trecho. As threads operam na pista através de duas funções, **chega** e **sai** que indicam quando uma thread sai ou entra em um trecho da pista.

A cada troca de posição de uma thread a função **evento** é chamada para testar a quebra de um ciclista e a mudança de velocidade (no caso da opção 'v'). Uma thread não ciclista externa verifica quando existe a ultrapassagem e printa a pista a cada 60ms.

Experimentos

Experimentos

Observar o consumo de memória com variação do número de threads e do tamanho da pista. Para obter os valores executamos `/usr/bin/time -v ./ep2` e na saída pegamos o valor do Maximum resident set size (kbytes). Feitos 5 testes com a opção 'u':

- Poucas threads (5) em uma pista de 250
- Valor médio de threads (25) em uma pista de 250
- Muitas threads (60) em uma pista de 250
- Pista pequena 250 com 25 threads
- Pista média 500 com 25 threads
- Pista grande 750 com 25 threads

Para cada métrica calculou-se a média e o intervalo de confiança de 30 medições com nível de 95%.

Resultados

Resultados

Consumo de memória comparação para quantidade de threads (esquerda) e tamanho da pista (direita).

