

Sorteio Distribuído

Gubi

20 de novembro de 2010

Deve ser implementado um algoritmo *distribuído* que muda o estado de uma fração dos processos. O número de processos transformados é dado de entrada. Ao longo de várias execuções, deve-se garantir que a distribuição dos processos modificados deve ser uniforme. Pequenas variações da uniformidade são aceitáveis, desde que explicadas no código.

1 Condições

O programa deve satisfazer as seguintes condições:

1. Deve ser escrito em *MPI*.
2. Deve possuir *simetria textual* e não pode haver um processo centralizador.
3. Inicialmente todos os processos se encontram no estado “Neutro”.
4. Ao final o número de processos indicados, e apenas estes, se encontram no estado “Ativado”.

2 Entrada

O programa será disparado por uma linha padrão do *MPI*:

```
mpirun -np procs programa frac
```

O número de processos (*procs*) que deverá mudar de estado é o teto da proporção (*frac*): $\lceil \textit{procs} \times \textit{frac} \rceil$.

3 Outras considerações

O número de mensagens deve ser econômico, deve-se evitar mensagens superfluas ou algoritmos complicados demais.

Tanto o algoritmo como sua implementação devem estar **bem documentados**, especialmente no que diz respeito à clareza.