

Desenvolver uma versão paralela do algoritmo para encontrar o maior elemento em um vetor, seguindo estas orientações:

- 1) gere os números usando o valor 1 para todos os elementos e em seguida substitua o valor da posição do meio do vetor ($tamanho_vetor / 2$) pelo valor de $tamanho_vetor$;
- 2) não use a diretiva **for do omp** (não confundir com o **for** do C);
- 3) calcule os limites das iterações dos loops de cada *thread* no seu código explicitamente;
- 4) paralelize a geração dos nrs no vetor e também a busca do maior valor;
- 5) sincronize com uma diretiva **barrier**, entre a geração dos nrs e a busca do maior valor;
- 6) na paralelização dos **for's**, distribua as iterações estaticamente por blocos contínuos:

Thread 00 – gera elems do bloco 0 (0 até $tam/num_threads$);

Thread 01 – gera elems do bloco 1 ($tam/num_threads$ até $(tam/num_threads)*2$);

...

- 7) não use a cláusula **reduce** e, sim, otimize o uso da memória compartilhada;
- 8) imprima na tela o maior valor encontrado no vetor, como feito em sala de aula.

Itens a entregar :

- 1) Projeto do algoritmo paralelo usando a Metodologia PCAM.
- 2) Código fonte em C com boas práticas de programação.
- 3) *Makefile* conforme exemplo.
- 4) Compacte os três arquivos acima em arquivo **.zip** e entregue este arquivo **.zip** apenas. Não troque o compactador.

Observação importante:

Desenvolva e teste o seu algoritmo no Cluster do LaSDPC. O exercício será corrigido nele.

Coloque os nomes dos integrantes do grupo que participaram de fato do desenvolvimento no começo do projeto do algoritmo e do código fonte em C.